Continuous Data Home

Continuous Data

Exported on 05/22/2024

Table of Contents

1 Welcome to the Delphix Continuous Data documentation! 75
2 Quick references 93
3 Release notes 94
3.1 New features 94
3.1.1 Release 23.0.0.0 94
3.1.2 Release 22.0.0.0 94
3.1.3 Release 21.0.0.0 95
3.1.4 Release 20.0.0.0 95
3.1.5 Release 19.0.0.0 96
3.1.6 Release 18.0.0.0 97
3.1.7 Release 17.0.0.0 97
3.1.8 Release 16.0.0.0 98
3.1.9 Release 15.0.0.0 98
3.1.10 Release 14.0.0.0 99
3.1.11 Release 13.0.0.0 100
3.1.12 Release 12.0.0.0 101
3.1.13 Release 11.0.0.0 101
3.1.14 Release 10.0.0.0 101
3.1.15 Release 9.0.0.0 102
3.1.16 Release 8.0.0.0 102
3.1.17 Release 7.0.0.0 102
3.1.18 Release 6.0.17.0 103
3.1.19 Release 6.0.16.0 103
3.1.20 Release 6.0.15.0 104
3.1.21 Release 6.0.14.0 105
3.1.22 Release 6.0.13.0 106
3.1.22.1 New in this release 106

3.1.22.2 Certifications 107
3.1.23 Release 6.0.12.0 107
3.1.23.1 New in this release 107
3.1.23.2 Certifications 108
3.1.24 Release 6.0.11.0 108
3.1.24.1 New in this release 108
3.1.24.2 Certifications 108
3.1.25 Release 6.0.10.0 109
3.1.25.1 Virtualization 109
3.1.25.2 Certifications 110
3.1.25.3 Virtualization 110
3.1.26 Release 6.0.9.0 110
3.1.26.1 Virtualization 110
3.1.26.2 Certifications 111
3.1.27 Release 6.0.8.0 111
3.1.27.1 Virtualization 111
3.1.28 Release 6.0.7.0 112
3.1.28.1 Virtualization 112
3.1.29 Release 6.0.6.0 113
3.1.29.1 Virtualization 113
3.1.29.2 Certifications 114
3.1.30 Release 6.0.5.0 114
3.1.30.1 Virtualization 114
3.1.30.2 Certifications 115
3.1.31 Release 6.0.4.0 116
3.1.31.1 Virtualization 116
3.1.31.2 Certifications 116
3.1.32 Release 6.0.3.0 116
3.1.32.1 Virtualization 116

3.1.32.2 Certifications ... 117
3.1.33 Release 6.0.2.0 ... 117
3.1.33.1 Virtualization ... 117
3.1.33.2 Certifications ... 118
3.1.34 Release 6.0.1.0 ... 118
3.1.34.1 Certifications ... 118
3.1.35 Release 6.0.0.0 ... 119
3.2 Fixed issues ... 120
3.2.1 Release 23.0.0.0 Changes ... 120
3.2.2 Fixes that take effect after upgrading and rebooting ... 120
3.2.3 Fixes that take effect immediately after upgrading ... 121
3.2.4 Release 22.0.0.0 Changes ... 122
3.2.5 Fixes that take effect immediately after upgrading ... 122
3.2.6 Release 21.0.0.1 Changes ... 125
3.2.7 Fixes that take effect immediately after upgrading ... 125
3.2.8 Release 21.0.0.0 Changes ... 125
3.2.9 Fixes that take effect after upgrading and rebooting ... 125
3.2.10 Fixes that take effect immediately after upgrading ... 125
3.2.11 Release 20.0.0.0 Changes ... 127
3.2.12 Fixes that take effect after upgrading and rebooting ... 127
3.2.13 Fixes that take effect immediately after upgrading ... 127
3.2.14 Release 19.0.0.0 Changes ... 129
3.2.15 Fixes that take effect after upgrading and rebooting ... 129
3.2.16 Fixes that take effect immediately after upgrading ... 129
3.2.17 Release 18.0.0.1 Changes ... 131
3.2.18 Fixes that take effect immediately after upgrading ... 131
3.2.19 Release 18.0.0.0 Changes ... 131
3.2.20 Fixes that take effect immediately after upgrading ... 131
3.2.21 Release 17.0.0.0 Changes ... 132

3.2.22 Fixes that take effect immediately after upgrading 132
3.2.23 Release 16.0.0.0 Changes 134
3.2.24 Fixes that take effect after upgrading and rebooting (optional) 134
3.2.25 Fixes that take effect immediately after upgrading 134
3.2.26 Release 15.0.0.0 Changes 135
3.2.27 Fixes that take effect after upgrading and rebooting (optional) 135
3.2.28 Fixes that take effect immediately after upgrading 136
3.2.29 Release 14.0.0.0 Changes 137
3.2.30 Fixes that take effect after upgrading and rebooting (optional) 137
3.2.31 Fixes that take effect immediately after upgrading 138
3.2.32 Release 13.0.0.0 Changes 139
3.2.33 Fixes that take effect immediately after upgrading 139
3.2.34 Release 12.0.0.0 Changes 141
3.2.35 Security fixes 141
3.2.36 Fixes that take effect after upgrading and rebooting (optional) 141
3.2.37 Fixes that take effect immediately after upgrading 142
3.2.38 Release 11.0.0.0 changes 143
3.2.39 Security fixes 143
3.2.40 Fixes that take effect after upgrading and rebooting (optional) 144
3.2.41 Fixes that take effect immediately after upgrading 144
3.2.42 Release 10.0.0.1 changes 148
3.2.43 Fixes that take effect after upgrading and rebooting (optional) 148
3.2.44 Release 10.0.0.0 changes 148
3.2.45 Fixes that take effect after upgrading and rebooting (optional) 148
3.2.46 Fixes that take effect immediately after upgrading 149
3.2.47 Release 9.0.0.1 changes 151
3.2.48 Release 9.0.0.0 changes 151
3.2.49 Fixes that take effect after upgrading and rebooting (optional) 151
3.2.50 Fixes that take effect immediately after upgrading 152

3.2.51 Release 8.0.0.0 changes 153
3.2.52 Fixes that take effect after upgrading and rebooting (optional)..... 153
3.2.53 Fixes that take effect immediately after upgrading..... 154
3.2.54 Release 7.0.0.0 changes 157
3.2.55 Fixes that take effect after upgrading and rebooting (optional)..... 157
3.2.56 Fixes that take effect after upgrading and rebooting (optional)..... 157
3.2.57 Fixes that take effect immediately after upgrading..... 158
3.2.58 Release 6.0.17.0 changes 161
3.2.59 Security fixes..... 161
3.2.60 Fixes that take effect after upgrading and rebooting (optional)..... 161
3.2.61 Fixes that take effect immediately after upgrading..... 162
3.2.62 Release 6.0.16.0 changes 165
3.2.63 Release 6.0.15.0 changes 167
3.2.64 Fixes that take effect after upgrading and rebooting (optional)..... 167
3.2.65 Fixes that take effect immediately after upgrading..... 167
3.2.66 Release 6.0.14.0 changes 170
3.2.67 Security fixes..... 170
3.2.68 Fixes that take effect immediately after upgrading..... 170
3.2.69 Fixes that take effect after upgrading and rebooting (optional)..... 173
3.2.70 Release 6.0.13.1 changes 173
3.2.71 Fixes that take effect immediately after upgrading..... 173
3.2.72 Release 6.0.13.0 changes 174
3.2.73 Security fixes..... 174
3.2.74 Fixes that take effect immediately after upgrading..... 174
3.2.75 Release 6.0.12.1 changes 177
3.2.76 Fixes that take effect immediately after upgrading..... 177
3.2.77 Release 6.0.12.0 changes 177
3.2.78 Log4j updates..... 177
3.2.79 Fixes that take effect immediately after upgrading..... 178

3.2.80 Release 6.0.11.0 changes 181
3.2.81 Security fixes 181
3.2.82 Fixes that take effect immediately after upgrading 181
3.2.83 Fixes that take effect after upgrading and rebooting (optional) 184
3.2.84 Release 6.0.10.1 changes 185
3.2.85 Fixes that take effect immediately after upgrading 185
3.2.86 Release 6.0.10.0 changes 185
3.2.87 Fixes that take effect immediately after upgrading 185
3.2.88 Fixes that take effect after upgrading and rebooting (optional) 189
3.2.89 Release 6.0.9.0 changes 190
3.2.90 Fixes that take effect immediately after upgrading 190
3.2.91 Fixes that take effect after upgrading and rebooting (optional) 194
3.2.92 Release 6.0.8.1 changes 194
3.2.93 Fixes that take effect immediately after upgrading 194
3.2.94 Release 6.0.8.0 changes 194
3.2.95 Fixes that take effect immediately after upgrading 194
3.2.96 Fixes that take effect after upgrading and rebooting (optional) 198
3.2.97 Release 6.0.7.0 changes 199
3.2.98 Fixes that take effect immediately after upgrading 199
3.2.99 Fixes that take effect after upgrading and rebooting (optional) 203
3.2.100 Release 6.0.6.1 changes 203
3.2.101 Fixes that take effect immediately after upgrading 203
3.2.102 Release 6.0.6.0 changes 204
3.2.103 Fixes that take effect immediately after upgrading 204
3.2.104 Fixes that take effect after upgrading and rebooting (optional) 207
3.2.105 Fixes that take effect after upgrading and rebooting (optional) 208
3.2.106 Release 6.0.5.0 changes 208
3.2.107 Fixes that take effect immediately after upgrading 208
3.2.108 Fixes that take effect after upgrading and rebooting (optional) 211

3.2.109 Release 6.0.4.2 changes 211
3.2.110 Fixes that take effect immediately after upgrading..... 211
3.2.111 Release 6.0.4.1 changes 212
3.2.112 Fixes that take effect immediately after upgrading..... 212
3.2.113 Release 6.0.4.0 changes 212
3.2.114 Fixes that take effect immediately after upgrading..... 212
3.2.115 Release 6.0.3.1 changes 214
3.2.116 Fixes that take effect after upgrading and rebooting (optional)..... 214
3.2.117 Release 6.0.3.0 changes 215
3.2.118 Fixes that take effect immediately after upgrading..... 215
3.2.119 Fixes that take effect after upgrading and rebooting (optional)..... 219
3.2.120 Release 6.0.2.1 changes 220
3.2.121 Fixes that take effect immediately after upgrading..... 220
3.2.122 Fixes that take effect after upgrading and rebooting (optional)..... 220
3.2.123 Release 6.0.2.0 changes 221
3.2.124 Fixes that take effect immediately after upgrading..... 221
3.2.125 Fixes that take effect after upgrading and rebooting (optional)..... 223
3.2.126 Release 6.0.1.1 changes 224
3.2.127 Fixes that take effect after upgrading and rebooting (optional)..... 224
3.2.128 Release 6.0.1.0 changes 224
3.2.129 Fixes that take effect immediately after upgrading..... 224
3.2.130 Fixes that take effect after upgrading and rebooting (optional)..... 228
3.2.131 Release 6.0.0.0 changes 229
3.2.132 Fixes that take effect immediately after upgrading..... 229
3.2.133 Fixes that take effect after upgrading and rebooting (optional)..... 232
3.2.134 Release 6.0.15.0 changes 234
3.2.135 Fixes that take effect after upgrading and rebooting (optional)..... 234
3.2.136 Fixes that take effect immediately after upgrading..... 234
3.2.137 Release 6.0.14.0 changes 236

3.2.138 Security fixes ... 236
3.2.139 Fixes that take effect immediately after upgrading ... 237
3.2.140 Fixes that take effect after upgrading and rebooting (optional) ... 240
3.2.141 Release 6.0.13.1 changes ... 240
3.2.142 Fixes that take effect immediately after upgrading ... 240
3.2.143 Release 6.0.13.0 changes ... 240
3.2.144 Security fixes ... 240
3.2.145 Fixes that take effect immediately after upgrading ... 241
3.2.146 Release 6.0.12.1 changes ... 244
3.2.147 Fixes that take effect immediately after upgrading ... 244
3.2.148 Release 6.0.12.0 changes ... 244
3.2.149 Log4j updates ... 244
3.2.150 Fixes that take effect immediately after upgrading ... 245
3.2.151 Release 6.0.11.0 changes ... 248
3.2.152 Security fixes ... 248
3.2.153 Fixes that take effect immediately after upgrading ... 248
3.2.154 Fixes that take effect after upgrading and rebooting (optional) ... 251
3.2.155 Release 6.0.10.1 changes ... 252
3.2.156 Fixes that take effect immediately after upgrading ... 252
3.2.157 Release 6.0.10.0 changes ... 252
3.2.158 Fixes that take effect immediately after upgrading ... 252
3.2.159 Fixes that take effect after upgrading and rebooting (optional) ... 256
3.2.160 Release 6.0.9.0 changes ... 257
3.2.161 Fixes that take effect immediately after upgrading ... 257
3.2.162 Fixes that take effect after upgrading and rebooting (optional) ... 261
3.2.163 Release 6.0.8.1 changes ... 261
3.2.164 Fixes that take effect immediately after upgrading ... 261
3.2.165 Release 6.0.8.0 changes ... 261
3.2.166 Fixes that take effect immediately after upgrading ... 261

3.2.167 Fixes that take effect after upgrading and rebooting (optional)..... 266
3.2.168 Release 6.0.7.0 changes 267
3.2.169 Fixes that take effect immediately after upgrading..... 267
3.2.170 Fixes that take effect after upgrading and rebooting (optional)..... 271
3.2.171 Release 6.0.6.1 changes 271
3.2.172 Fixes that take effect immediately after upgrading..... 271
3.2.173 Release 6.0.6.0 changes 272
3.2.174 Fixes that take effect immediately after upgrading..... 272
3.2.175 Fixes that take effect after upgrading and rebooting (optional)..... 275
3.2.176 Fixes that take effect after upgrading and rebooting (optional)..... 276
3.2.177 Release 6.0.5.0 changes 276
3.2.178 Fixes that take effect immediately after upgrading..... 276
3.2.179 Fixes that take effect after upgrading and rebooting (optional)..... 279
3.2.180 Release 6.0.4.2 changes 279
3.2.181 Fixes that take effect immediately after upgrading..... 279
3.2.182 Release 6.0.4.1 changes 280
3.2.183 Fixes that take effect immediately after upgrading..... 280
3.2.184 Release 6.0.4.0 changes 280
3.2.185 Fixes that take effect immediately after upgrading..... 280
3.2.186 Release 6.0.3.1 changes 282
3.2.187 Fixes that take effect after upgrading and rebooting (optional)..... 282
3.2.188 Release 6.0.3.0 changes 282
3.2.189 Fixes that take effect after upgrading and rebooting (optional)..... 282
3.2.190 Release 6.0.2.1 changes 283
3.2.191 Fixes that take effect immediately after upgrading..... 283
3.2.192 Fixes that take effect after upgrading and rebooting (optional)..... 284
3.2.193 Release 6.0.2.0 changes 284
3.2.194 Fixes that take effect immediately after upgrading..... 284
3.2.195 Fixes that take effect after upgrading and rebooting (optional)..... 287

3.2.196 Release 6.0.1.1 changes 287
3.2.197 Fixes that take effect after upgrading and rebooting (optional)..... 287
3.2.198 Release 6.0.1.0 changes 288
3.2.199 Fixes that take effect immediately after upgrading..... 288
3.2.200 Fixes that take effect after upgrading and rebooting (optional)..... 292
3.2.201 Release 6.0.0.0 changes 292
3.2.202 Fixes that take effect immediately after upgrading..... 292
3.2.203 Fixes that take effect after upgrading and rebooting (optional)..... 295
3.3 Known issues 296
3.3.1 Version 23.0.0.0 296
3.3.2 Version 22.0.0.0 297
3.3.3 Version 21.0.0.0 298
3.3.4 Version 20.0.0.0 300
3.3.5 Version 19.0.0.0 303
3.3.6 Version 18.0.0.0 304
3.3.7 Version 17.0.0.0 306
3.3.8 Version 16.0.0.0 307
3.3.9 Version 15.0.0.0 309
3.3.10 Version 14.0.0.0 313
3.3.11 Version 13.0.0.0 316
3.3.12 Version 12.0.0.0 320
3.3.13 Version 11.0.0.0 324
3.3.14 Version 10.0.0.0 328
3.3.15 Version 9.0.0.0 332
3.3.16 Version 8.0.0.0 333
3.3.17 Version 7.0.0.0 334
3.3.18 Version 6.0.17.0 335
3.3.19 Version 6.0.16.0 336
3.3.20 Version 6.0.15.0 337

3.3.21 Version 6.0.14.0 ... 337
3.3.22 Version 6.0.13.0 ... 338
3.3.23 Version 6.0.12.0 ... 339
3.3.24 Version 6.0.11.0 ... 341
3.3.25 Version 6.0.10.0 ... 342
3.3.26 Version 6.0.9.0 ... 343
3.3.27 Version 6.0.8.0 ... 345
3.3.28 Version 6.0.7.0 ... 348
3.3.29 Version 6.0.6.0 ... 349
3.3.30 Version 6.0.2.0 ... 351
3.3.31 Version 6.0.0.0 ... 352
3.4 API changes ... 353
3.4.1 API changes in Delphix 23.0.0.0 ... 354
3.4.1.1 What's new? ... 354
3.4.1.2 What's changed? ... 354
3.4.2 API changes in Delphix 22.0.0.0 ... 355
3.4.2.1 What's changed? ... 355
3.4.2.2 What's new? ... 355
3.4.3 API changes in Delphix 21.0.0.0 ... 356
3.4.3.1 What's new? ... 356
3.4.3.2 What's changed? ... 356
3.4.4 API changes in Delphix 20.0.0.0 ... 357
3.4.4.1 What's New? ... 357
3.4.4.2 What's Changed? ... 357
3.4.5 API changes in Delphix 19.0.0.0 ... 358
3.4.5.1 What's new? ... 358
3.4.5.2 What's changed? ... 358
3.4.6 API changes in Delphix 18.0.0.0 ... 359
3.4.6.1 What's New? ... 359

3.4.6.2 What's Changed? ... 359
3.4.7 API Changes in Delphix 17.0.0.0 ... 359
3.4.7.1 What's New? ... 360
3.4.8 API Changes in Delphix 16.0.0.0 ... 360
3.4.8.1 What's Changed? ... 360
3.4.9 API Changes in Delphix 15.0.0.0 ... 361
3.4.9.1 What's Changed? ... 361
3.4.9.2 What's New? ... 362
3.4.10 API Changes in Delphix 14.0.0.0 ... 362
3.4.10.1 What's Changed? ... 362
3.4.10.2 What's New? ... 363
3.4.11 API Changes in Delphix 13.0.0.0 ... 363
3.4.11.1 What's Changed? ... 364
3.4.11.2 What's New? ... 364
3.4.12 API Changes in Delphix 12.0.0.0 ... 364
3.4.12.1 What's Changed? ... 365
3.4.12.2 What's New? ... 366
3.4.13 API changes in Delphix 11.0.0.0 ... 367
3.4.13.1 What's changed? ... 367
3.4.13.2 What's new? ... 369
3.4.14 API changes in Delphix 10.0.0.0 ... 370
3.4.14.1 What's new? ... 371
3.4.15 API changes in Delphix 9.0.0.0 ... 372
3.4.15.1 What's changed? ... 372
3.4.15.2 What's new? ... 373
3.4.16 API changes in Delphix 8.0.0.0 ... 374
3.4.16.1 What's changed? ... 374
3.4.16.2 What's new? ... 376
3.4.17 API changes in Delphix 7.0.0.0 ... 377

3.4.17.1 What's changed? ... 377
3.4.17.2 What's new? ... 377
3.4.18 API changes in Delphix 6.0.17.0 ... 380
3.4.18.1 What's changed? ... 380
3.4.18.2 What's new? ... 382
3.4.19 API changes in Delphix 6.0.16.0 ... 383
3.4.19.1 What's changed? ... 383
3.4.19.2 What's new? ... 384
3.4.20 API changes in Delphix 6.0.15.0 ... 385
3.4.20.1 What's changed? ... 385
3.4.20.2 What's new? ... 386
3.4.21 API changes in Delphix 6.0.14.0 ... 388
3.4.21.1 What's changed? ... 388
3.4.21.2 What's new? ... 389
3.4.22 API changes in Delphix 6.0.13.0 ... 391
3.4.22.1 What's changed? ... 391
3.4.22.2 What's new? ... 393
3.4.23 API changes in Delphix 6.0.12.0 ... 396
3.4.23.1 What's changed? ... 396
3.4.23.2 What's new? ... 398
3.4.24 API changes in Delphix 6.0.11.0 ... 400
3.4.24.1 What's changed? ... 400
3.4.24.2 What's new? ... 403
3.4.25 API changes in Delphix 6.0.10.0 ... 405
3.4.25.1 What's changed? ... 406
3.4.25.2 What's new? ... 407
3.4.26 API changes in Delphix 6.0.9.0 ... 408
3.4.26.1 What changed ... 408
3.4.26.2 What is new ... 410

3.4.27 API changes in Delphix 6.0.8.0 ... 411
3.4.27.1 What changed ... 411
3.4.27.2 What is new ... 414
3.4.28 API changes in Delphix 6.0.7.0 ... 416
3.4.28.1 What’s changed ... 416
3.4.28.2 What 's new ... 417
3.4.29 API changes in Delphix 6.0.6.0 ... 419
3.4.29.1 What’s changed ... 420
3.4.30 API changes in Delphix 6.0.5.0 ... 420
3.4.30.1 What’s changed ... 420
3.4.30.2 What’s new ... 421
3.4.31 API changes in Delphix 6.0.4.0 ... 423
3.4.31.1 What’s changed ... 423
3.4.31.2 What’s new ... 424
3.4.32 API changes in Delphix 6.0.3.0 ... 425
3.4.32.1 What’s changed ... 425
3.4.32.2 What’s new ... 426
3.5 Support matrices ... 428
3.5.1 Kerberos support matrix ... 429
3.5.1.1 Oracle - Red Hat Enterprise Linux (RHEL) ... 429
3.5.1.2 SAP ASE - Red Hat Enterprise Linux (RHEL) ... 431
3.5.1.3 DB2 support matrix ... 432
3.5.1.4 Red Hat Enterprise Linux (RHEL) ... 432
3.5.1.5 DB2 on Advanced Interactive eXecutive (AIX) ... 433
3.5.2 Data source certifications ... 434
3.5.2.1 Legal notice ... 434
3.5.2.2 Goals ... 434
3.5.2.3 Data sources ... 435
3.5.2.4 Cadence for data source database version changes ... 435

3.5.2.5 Cadence for operating system version changes 435
3.5.2.6 Appendix 436
3.5.3 Select connectors matrix 438
3.5.3.1 Couchbase 439
3.5.3.2 Architecture options 440
3.5.3.3 Mongo 441
3.5.3.4 Architecture options 443
3.5.3.5 Microsoft SQL backup ingestion 444
3.5.3.6 Architecture options 445
3.5.3.7 Oracle backup ingestion 446
3.5.3.8 Architecture Options 447
3.5.3.9 SAPIQ 448
3.5.3.10 Architecture options 449
3.5.3.11 Cockroach 450
3.5.3.12 Cassandra 453
3.6 Upgrade matrix 454
3.7 Tested browser and operating systems 457
3.8 Deprecated and end-of-life features 457
3.8.1 Release 23.0.0.0 457
3.8.1.1 Deprecated features 457
3.8.2 Release 21.0.0.0 458
3.8.2.1 Deprecated features 458
3.8.3 Release 17.0.0.0 458
3.8.3.1 End-of-life feature 458
3.8.4 Release 11.0.0.0 458
3.8.4.1 End-of-life feature 458
3.8.5 Release 10.0.0.0 458
3.8.5.1 Deprecated features 458
3.8.6 Release 9.0.0.0 459

3.8.6.1 Deprecated features ... 459
3.8.7 Release 6.0.17.0 ... 459
3.8.7.1 End-of-life features ... 459
3.8.7.2 Deprecated features ... 459
3.8.8 Release 6.0.12.0 ... 459
3.8.8.1 End-of-life features ... 459
3.8.9 Release 6.0.11.0 ... 460
3.8.9.1 Deprecated features ... 460
3.8.9.2 End-of-life features ... 460
3.8.10 Release 6.0.10.0 ... 460
3.8.10.1 End-of-life features ... 460
3.8.11 Release 6.0.9.0 ... 461
3.8.11.1 End-of-life features ... 461
3.8.12 Release 6.0.8.0 ... 461
3.8.12.1 End-of-life features ... 461
3.8.13 Release 6.0.7.0 ... 461
3.8.13.1 End-of-life features ... 461
3.8.14 Release 6.0.5.0 ... 461
3.8.15 Release 6.0.4.0 ... 462
3.8.15.1 End-of-life features ... 462
3.8.15.2 Deprecated features ... 462
3.8.16 Release 6.0.2.0 ... 462
3.8.16.1 End-of-life features ... 462
3.8.17 Release 6.0.0.0 ... 463
3.8.17.1 End-of-life features ... 463
3.9 Licenses and notices ... 463
3.10 Data source integration (plugin) release notes ... 464
3.10.1 CockroachDB release notes ... 464
3.10.1.1 New features (CockroachDB) ... 464

3.10.1.2 Fixed issues (CockroachDB) ... 465
3.10.1.3 Known Issues (CockroachDB) ... 466
3.10.2 Couchbase release notes ... 466
3.10.2.1 New features (Couchbase) ... 466
3.10.2.2 Fixed issues (Couchbase) ... 467
3.10.3 IBM Db2 release notes ... 470
3.10.3.1 New features (IBM Db2) ... 470
3.10.3.2 Fixed issues (IBM Db2) ... 477
3.10.3.3 Known issues (IBM Db2) ... 482
3.10.4 MongoDB release notes ... 486
3.10.4.1 New features (MongoDB) ... 487
3.10.4.2 Fixed issues (MongoDB) ... 488
3.10.4.3 Known issues (MongoDB) ... 491
3.10.5 MySQL release notes ... 491
3.10.5.1 New features (MySQL) ... 491
3.10.5.2 Fixed issues (MySQL) ... 493
3.10.6 Oracle E-Business Suite (EBS) 12.2 release notes ... 495
3.10.6.1 New features (EBS) ... 495
3.10.6.2 Fixed issues (EBS) ... 499
3.10.6.3 Known issues (EBS) ... 505
3.10.7 PostgreSQL release notes ... 507
3.10.7.1 New features (PostgreSQL) ... 507
3.10.7.2 Fixed issues (PostgreSQL) ... 512
3.10.7.3 Known issues (PostgreSQL) ... 519
3.10.8 SAP HANA release notes ... 521
3.10.8.1 New features (SAP HANA) ... 521
3.10.8.2 Fixed issues (SAP HANA) ... 523
3.10.8.3 Known issues (SAP HANA) ... 526
3.10.9 Yugabyte release notes ... 537

3.10.9.1 New features (Yugabyte) ... 537
4 Overview ... 538
4.1 What is Delphix? ... 538
4.2 What is Delphix Continuous Data? ... 538
4.3 Deployment ... 538
4.4 Environments ... 538
4.5 Data Source ... 539
4.6 Virtual Databases ... 539
4.7 Product information ... 539
4.7.1 Overview ... 539
4.7.2 Packaging ... 539
4.7.3 Updates, upgrades, and versions ... 540
4.7.4 Administration ... 540
4.7.5 Customization ... 541
4.8 Delphix glossary ... 541
4.8.1 Products ... 542
4.8.2 Interfaces ... 542
4.8.3 Core concepts ... 543
4.8.4 Continuous Data ... 544
4.8.5 Architecture ... 545
4.8.6 Datasets ... 547
4.8.7 Data Source Connectors ... 549
4.8.8 Transparent Data Encryption (TDE) definitions ... 549
4.8.9 Users roles ... 550
4.8.10 User privileges ... 551
4.8.11 Types of notification ... 552
4.8.12 Delphix Self-Service terms ... 552
4.9 Product icon reference ... 554
5 Deployment ... 556

5.1 Accessing the Continuous Data Engine 556
5.1.1 Default users 556
5.1.2 Access methods 557
5.1.2.1 Accessing the Delphix Continuous Data Engine using a web browser .. 557
5.1.2.2 Accessing the Delphix Continuous Data Engine using administrative Command Line Interface (CLI) 558
5.1.2.3 Accessing the Delphix Continuous Data Engine using the Delphix web API 559
5.2 Standard deployment architecture 559
5.2.1 Overview 559
5.3 Checklist of information required for installation and configuration 560
5.3.1 Overview 560
5.3.2 Hypervisor specific options 561
5.3.3 Optional information required for initial configuration 564
5.4 Network connectivity requirements 566
5.4.1 Overview 566
5.4.2 General outbound from the Delphix engine port allocation 567
5.4.3 General inbound to the Delphix Engine port allocation 567
5.4.4 Firewalls and intrusion detection systems (IDS) 568
5.4.5 Setting up network access to the Delphix engine 569
5.4.5.1 Overview 569
5.4.5.2 Procedure 569
5.5 Installation and initial system configurations 571
5.5.1 Initial setup 571
5.5.1.1 Overview 571
5.5.1.2 Welcome 572
5.5.1.3 Administrators 572
5.5.1.4 Time 573
5.5.1.5 Network 573
5.5.1.6 Network security 574

5.5.1.7 Storage ... 574
5.5.1.8 Outbound connectivity ... 575
5.5.1.9 Authentication ... 576
5.5.1.10 Kerberos ... 576
5.5.1.11 Registration ... 577
5.5.1.12 Summary ... 577
5.5.1.13 Azure object storage setup ... 578
5.5.1.14 AWS object storage setup ... 586
5.5.1.15 GCP object storage setup ... 588
5.5.1.16 OCI object storage setup ... 592
5.5.1.17 Other S3 compatible object storage setup ... 596
5.5.1.18 Updating object storage endpoints ... 598
5.5.2 Customizing the Delphix engine system settings ... 599
5.5.2.1 Overview ... 599
5.5.2.2 Procedure ... 599
5.5.2.3 Enabling hot-adding ... 601
5.5.2.4 Post-requisites ... 601
5.5.3 Installing an OVA or AMI ... 602
5.5.3.1 Overview ... 602
5.5.3.2 Procedure to install an OVA ... 602
5.5.3.3 Procedure to install an AMI ... 603
5.6 Validating host deployment with host Checker ... 603
5.6.1 Overview ... 603
5.6.2 Procedure ... 603
5.7 Deployment for VMware ... 604
5.7.1 Overview ... 604
5.7.2 Supported ESX versions ... 605
5.7.3 Virtual machine hardware versions ... 605
5.7.4 Virtual CPUs ... 606

5.7.5 Memory ... 606
5.7.6 Network ... 607
5.7.7 SCSI controller ... 609
5.7.8 General storage ... 609
5.7.9 Delphix storage options ... 610
5.7.9.1 Delphix VM configuration storage ... 610
5.7.9.2 Delphix engine system disk storage ... 610
5.7.9.3 Database storage ... 611
5.7.9.4 Option 1: Block Storage for database storage ... 611
5.7.9.5 Option 2: Elastic Data where Object storage is used for database storage and block storage is used for cache ... 613
5.7.10 Additional VMware configuration notes ... 613
5.7.11 Procedure to install an OVA ... 613
5.8 Deployment for KVM ... 615
5.8.1 Overview ... 615
5.8.2 Pre-requisites ... 615
5.8.3 Virtual CPUs ... 616
5.8.4 Memory ... 617
5.8.5 Network ... 617
5.8.6 SCSI controller ... 618
5.8.7 General storage ... 618
5.9 Deployment for Hyper-V ... 619
5.9.1 Overview ... 619
5.9.2 Supported versions ... 620
5.9.3 Virtual CPUs ... 620
5.9.4 Memory ... 620
5.9.5 Network ... 621
5.9.6 SCSI controller ... 622
5.9.7 General storage ... 622

5.9.8 Delphix storage options ... 623
5.9.8.1 Delphix engine system disk storage ... 623
5.9.8.2 Database storage ... 623
5.9.9 Procedure for deploying with Hyper-V ... 624
5.9.9.1 Overview ... 624
5.9.9.2 Creating a Delphix engine VM ... 625
5.9.9.3 Customize the VM by selecting settings ... 625
5.10 Deployment for AWS EC2 ... 625
5.10.1 Overview ... 625
5.10.2 Instance types ... 626
5.10.3 Network configuration ... 626
5.10.4 EBS configuration ... 627
5.10.5 General storage configuration ... 628
5.10.6 Additional AWS configuration notes ... 628
5.10.7 Prerequisites to deploying in AWS ... 629
5.10.7.1 Overview ... 629
5.10.7.2 Prerequisites ... 629
5.10.7.3 Geographic distribution in regions and availability zones ... 630
5.10.8 Procedure for deploying in AWS ... 631
5.10.8.1 Overview ... 631
5.10.8.2 Procedure to install an AMI ... 631
5.10.8.3 Subscribe to the Delphix virtualization engine marketplace image ... 631
5.10.8.4 Launching the Delphix engine ... 632
5.10.8.5 Configuring the Delphix engine ... 635
5.10.8.6 Logging in for the first time ... 635
5.10.8.7 Next steps ... 636
5.10.9 AWS RDS Custom for Oracle and SQL Server ... 636
5.10.9.1 Overview ... 636
5.11 Deployment for Microsoft Azure ... 637

5.11.1 Overview ... 637
5.11.2 Instance types ... 638
5.11.3 Network configuration ... 638
5.11.4 Storage configuration ... 639
5.11.5 Extensions ... 640
5.11.6 Prerequisites to deploying in Azure ... 640
5.11.6.1 Overview ... 640
5.11.6.2 Prerequisites ... 640
5.11.7 Procedure for deploying in Azure ... 642
5.11.7.1 Overview ... 642
5.11.7.2 Deploying the Delphix engine ... 642
5.11.7.3 Configuring the Delphix engine virtual machine ... 643
5.11.7.4 Configuring the Delphix engine ... 644
5.11.7.5 Next steps ... 644
5.11.8 Enabling Azure accelerated networking ... 644
5.11.8.1 Overview ... 644
5.11.8.2 Individual VMs and VMs in an availability set ... 645
5.12 Deployment for Google cloud platform ... 646
5.12.1 Overview ... 646
5.12.2 Machine types ... 646
5.12.3 Network configuration ... 647
5.12.4 Storage configuration ... 648
5.12.5 Prerequisites to deploying in GCP ... 648
5.12.5.1 Overview ... 648
5.12.5.2 Prerequisites ... 649
5.12.5.3 Additional GCP configuration notes ... 650
5.12.6 Procedure for deploying in GCP ... 650
5.12.6.1 Overview ... 650
5.12.6.2 Deployment in Google cloud console ... 650

5.12.6.3 Deployment in Google cloud marketplace 651
5.12.6.4 Configuring the Delphix engine 651
5.12.6.5 Next Steps 651
5.13 Deployment for OCI 651
5.13.1 Overview 651
5.13.2 Compute image types 652
5.13.3 Supported shapes 652
5.13.4 Network configuration 653
5.13.5 General storage configuration 654
5.13.6 Additional OCI configuration notes 654
5.13.7 Prerequisites to deploying OCI 655
5.13.7.1 Overview 655
5.13.7.2 Prerequisites 655
5.13.8 Procedure for deploying in OCI 656
5.13.8.1 Overview 656
5.13.8.2 Download and verify the Delphix engine image 656
5.13.8.3 Upload the Delphix engine image as an object 657
5.13.8.4 Creating a custom compute image from an object 657
5.13.8.5 Launching the Delphix engine 658
5.13.8.6 Create block storage volumes 659
5.13.8.7 Attach block storage volumes 660
5.13.8.8 Configuring the Delphix engine 660
5.13.8.9 Next steps 661
5.14 Deployment for IBM cloud 661
5.14.1 Overview 661
5.14.2 Supported profiles 661
5.14.3 Network configuration 662
5.14.4 General storage configuration 663
5.14.5 Additional IBM configuration notes 663

5.14.6 Prerequisites for deploying in IBM Cloud 664
5.14.6.1 Overview 664
5.14.6.2 Prerequisites 664
5.14.7 Procedure for deploying in the IBM Cloud 665
5.14.7.1 Overview 665
5.14.7.2 Deploying from the IBM software catalog 665
5.14.7.3 Downloading the Delphix image 666
5.14.7.4 Uploading the Delphix engine image as an object 666
5.14.7.5 Creating a custom image 667
5.14.7.6 Launching the Delphix engine 668
5.14.7.7 Creating block storage volumes 668
5.14.7.8 Attaching block storage volumes 669
5.14.7.9 Configuring the Delphix engine 670
5.14.7.10 Next steps 670
5.15 Hotfix information 670
5.15.1 Overview 670
6 Configuration 671
6.1 Configuration 671
6.1.1 Registration management 671
6.1.2 User and authentication management 671
6.1.3 Network and DNS management 671
6.1.4 Capacity and resource management 671
6.1.5 Monitoring and log management 671
6.1.6 Performance analytics management 672
6.1.7 Starting, stopping, and restarting your engine 672
6.1.8 Usage data management 672
6.2 Registration management 672
6.2.1 Retrieving the Delphix engine registration code 672
6.2.2 Regenerating the Delphix engine registration code 673

6.3 User and authentication management ... 673
6.3.1 Users and groups ... 674
6.3.1.1 User types and user management ... 674
6.3.1.2 User privileges for Delphix objects ... 675
6.3.1.3 Managing groups ... 677
6.3.1.4 Authentication mechanisms ... 677
6.3.1.5 Assigning group and object ownership ... 678
6.3.1.6 Adding and deleting groups ... 678
6.3.1.7 Managing system administrators ... 680
6.3.1.8 Managing Delphix users ... 685
6.3.1.9 Managing individual profile information ... 689
6.3.2 Authentication mechanisms ... 689
6.3.2.1 How to setup auto-authentication ... 690
6.3.2.2 Configuring and using LDAP with the Delphix engine ... 692
6.3.2.3 Configuring single sign-on ... 696
6.3.2.4 Configuring and managing kerberos ... 699
6.3.2.5 Configuring OAuth2 authentication for API access ... 712
6.4 Network and DNS management ... 717
6.4.1 General network and connectivity requirements ... 717
6.4.1.1 Overview ... 717
6.4.1.2 General outbound from the Delphix engine port allocation ... 717
6.4.1.3 General inbound to the Delphix engine port allocation ... 718
6.4.1.4 Firewalls and intrusion detection systems (IDS) ... 718
6.4.2 Network performance configuration options ... 719
6.4.2.1 Optimal network architecture for the Delphix engine ... 719
6.4.2.2 Network operations using the Delphix session protocol ... 722
6.4.2.3 Network performance test tool interface ... 725
6.4.2.4 Working with dataset performance ... 730
6.4.2.5 Network performance expectations and troubleshooting ... 731

6.4.3 Determining the Delphix server ID and host name 733
6.4.3.1 Server setup application method 733
6.4.3.2 Delphix admin application 734
6.4.3.3 CLI method 735
6.4.4 Configuring multiple DNS domain names in DNS search list 736
6.4.4.1 Procedure 736
6.4.4.2 To update DNS using the CLI: 738
6.4.5 How to change the IP address of the Delphix engine 739
6.4.5.1 Changing the IP address 739
6.4.5.2 Changing the IP address via CLI 740
6.4.6 How to change the hostname of the Delphix engine 741
6.4.6.1 Changing the hostname 741
6.4.7 How to change the DNS server of the Delphix engine 743
6.4.8 Configuring a second network interface 744
6.4.8.1 Procedure 744
6.5 NFSv4 configuration 746
6.5.1 Overview 746
6.5.2 Dataset status tab NFS reasons 746
6.5.3 NFSv4-Only mode 747
6.6 Capacity and resource management 748
6.6.1 An overview of capacity and performance information 748
6.6.1.1 The performance reservoir and capacity threshold warnings 748
6.6.1.2 Ways to view capacity usage 749
6.6.1.3 Using and understanding the storage capacity screen 750
6.6.1.4 Understanding Delphix disk usage 756
6.6.1.5 Reviewing historical capacity from the CLI 757
6.6.2 Setting quotas 760
6.6.2.1 Quota thresholds 761
6.6.2.2 Disallowed database actions when in quota critical threshold 762

6.6.3 Deleting objects to increase capacity ... 763
6.6.4 Adding, expanding, and removing storage devices ... 764
6.6.4.1 Prerequisites ... 765
6.6.4.2 Adding or increasing storage and/or cache ... 765
6.6.4.3 Expanding a storage/cache device ... 765
6.6.4.4 Removing device storage ... 766
6.6.5 Delphix storage migration ... 768
6.6.5.1 Getting started ... 768
6.6.5.2 Understanding Delphix storage migration ... 769
6.6.5.3 Limitations of Delphix storage migration ... 770
6.6.5.4 User interface ... 770
6.6.5.5 Device removal for storage migration ... 770
6.6.5.6 Getting the UUID of an RDM disk from VMware via the vSphere GUI ... 773
6.6.5.7 Getting the UUID of a VMDK from VMware, via ssh to the ESX server ... 775
6.6.5.8 Getting the UUID of a VMDK from VMware, via VMware PowerCLI ... 775
6.6.5.9 In-place block-to-object storage migration ... 776
6.6.6 Managing source data ... 781
6.6.7 An overview of held space ... 782
6.6.7.1 Scenario description ... 782
6.7 Monitoring and log management ... 784
6.7.1 Configuring SNMP ... 785
6.7.1.1 Prerequisites ... 785
6.7.1.2 Configuring SNMP for v2 ... 786
6.7.1.3 Configuring SNMP for v3 ... 788
6.7.1.4 CLI: viewing SNMP engine ID ... 792
6.7.1.5 Supported MIBs ... 793
6.7.1.6 Examples ... 793
6.7.2 Viewing action status ... 794
6.7.2.1 Action sidebar procedure ... 794

6.7.2.2 Description ... 794
6.7.2.3 Sub-action ... 795
6.7.2.4 Errors ... 796
6.7.3 Viewing jobs ... 796
6.7.4 System faults ... 798
6.7.4.1 Overview ... 798
6.7.4.2 Delphix object-based environment monitor faults ... 799
6.7.4.3 Viewing faults ... 799
6.7.4.4 Addressing faults ... 800
6.7.4.5 Fault lifecycle example ... 801
6.7.4.6 Viewing system faults ... 802
6.7.5 Accessing audit logs ... 804
6.7.5.1 Overview ... 804
6.7.5.2 Procedure ... 805
6.7.5.3 Sorting and filtering ... 806
6.7.5.4 Column resizing and tooltips ... 806
6.7.5.5 Exporting results ... 807
6.7.6 Creating support logs ... 807
6.7.6.1 Using the GUI ... 807
6.7.6.2 Using the CLI ... 809
6.7.7 Setting support access control ... 809
6.7.8 Setting syslog preferences ... 810
6.7.8.1 Severity levels for syslog messages ... 812
6.7.9 Support access audit logs ... 812
6.7.9.1 Overview ... 813
6.7.9.2 Listing the session audit logs ... 813
6.7.9.3 Downloading a session audit log ... 813
6.7.9.4 Reviewing a session audit log ... 814
6.7.9.5 Limitations ... 814

6.7.9.6 Deleting a session audit log ... 814
6.7.10 Diagnosing connectivity errors ... 815
6.7.10.1 Failed actions ... 815
6.7.10.2 Viewing active faults ... 816
6.7.11 Email (SMTP) alert notifications ... 817
6.7.11.1 Overview ... 817
6.7.11.2 Configuring the SMTP gateway ... 817
6.7.11.3 Alert profiles ... 819
6.7.11.4 A simple profile ... 821
6.7.11.5 A compound alert profile ... 821
6.7.11.6 Complex alert profile ... 822
6.7.12 Fluentd plugin service for API modules ... 826
6.7.12.1 Overview ... 826
6.7.12.2 Technical details ... 827
6.7.12.3 Splunk integration ... 835
6.7.12.4 Creating Fluentd plugins ... 860
6.8 Performance analytics management ... 869
6.8.1 Performance analytics ... 870
6.8.1.1 Performance analytics tool overview ... 870
6.8.1.2 Working with performance analytics graphs in the graphical user interface ... 872
6.8.1.3 Performance analytics statistics reference ... 875
6.8.1.4 Performance analytics tool API reference ... 876
6.8.1.5 Performance analytics case study: using a single statistic ... 884
6.8.1.6 Performance analytics case study: using multiple statistics ... 888
6.8.2 Storage performance configuration options ... 896
6.8.2.1 Optimal storage configuration parameters for the Delphix engine ... 896
6.8.2.2 Storage performance test tool (fio) ... 897
6.8.2.3 Storage performance expectations and troubleshooting ... 903

6.8.3 Architecture for performance - hypervisors and host 904
6.8.3.1 Architecture best practices hypervisor host ESX 904
6.8.4 Target host OS and database configuration options 907
6.8.4.1 OS-specific tuning recommendations 907
6.8.4.2 OS-specific tuning recommendations for windows 916
6.9 Usage data management 917
6.9.1 Disabling user-click analytics 917
6.10 Starting, stopping, and restarting your engine 919
6.10.1 Factory reset 919
6.10.2 Restart 919
6.10.3 Reboot 920
6.10.4 Power off 921
6.11 Introduction to privilege elevation profiles 922
6.11.1 Support for privilege elevation profiles 923
6.11.2 How do privilege elevation profiles work? 923
7 Upgrade 925
7.1 Upgrade 925
7.2 Upgrading the Delphix Engine: Overview 925
7.2.1 Types of upgrade 925
7.2.2 Outline of the upgrade process 926
7.3 Upgrade prerequisites 927
7.3.1 Scheduling downtime 927
7.3.2 Verifying the integrity of the downloaded upgrade image 928
7.3.3 Verifying connectivity to datasets and environments 928
7.4 Downloading the upgrade image 929
7.5 Uploading the upgrade image 930
7.5.1 Upgrade verification 932
7.5.1.1 Understanding the verification page 933
7.5.2 Resolving upgrade checks 935

7.5.2.1 Overview ... 935
7.5.2.2 Upgrade severities checks ... 936
7.5.2.3 Upgrade check actions ... 937
7.6 Applying the upgrade ... 940
7.6.1 Failure to quiesce a dataset ... 941
7.7 Post upgrade ... 943
7.7.1 Failures ... 943
8 Security ... 944
8.1 Security principles ... 944
8.2 Product security ... 945
8.2.1 Delphix product security ... 945
8.2.1.1 Overview ... 945
8.2.1.2 Software delivery security ... 946
8.2.1.3 Ancillary components ... 946
8.2.2 Software updates ... 946
8.2.2.1 Patch annually ... 946
8.2.2.2 Subscribe to Delphix notifications ... 946
8.2.3 Network security ... 946
8.2.3.1 General port allocation ... 947
8.2.4 Password vault support ... 948
8.2.4.1 Overview ... 948
8.2.4.2 Configuring password vaults ... 948
8.2.4.3 Using password vaults ... 949
8.2.4.4 Roles and privileges for CyberArk and HashiCorp users ... 949
8.2.4.5 Supported environments and databases ... 949
8.2.4.6 Setting up a vault via GUI ... 949
8.2.4.7 Setting up Vault via API ... 962
8.2.5 Certificate management ... 965
8.2.5.1 Configuring network security settings ... 966

8.2.5.2 Configuring network settings ... 977
8.2.5.3 KeyStore settings ... 987
8.2.5.4 TrustStore settings ... 989
8.3 Replication security ... 991
8.3.1 Choose encrypt and compress when replicating ... 991
8.4 Object security ... 991
8.4.1 User management ... 992
8.4.1.1 Secure user management ... 992
8.4.1.2 Use LDAP for authentication ... 992
8.4.1.3 Create named users ... 992
8.4.1.4 Assign least privileges ... 992
8.4.1.5 Consider Delphix self-service functionality ... 992
8.4.1.6 Disable ADMIN and SYSADMIN ... 993
8.4.2 Source database security ... 993
8.4.2.1 Choose minimum privileges for Delphix DB user ... 993
8.4.2.2 Protect the Delphix DB user password ... 993
8.4.2.3 Use database encryption functionality to encrypt sensitive data at rest ... 993
8.4.2.4 Choose encrypt and compress when linking ... 993
8.4.3 Source and target host security ... 994
8.4.3.1 Oracle on UNIX ... 994
8.4.3.2 Windows ... 994
8.5 System configuration ... 995
8.5.1 Maintain system time with NTP ... 995
8.5.2 Enable phone home ... 996
8.5.3 Register your Delphix engine ... 996
8.5.4 Enable LDAP for authentication ... 996
8.5.5 Enable SMTP and/or SNMP monitoring ... 996
8.6 GUI security ... 996

8.6.1 Overview ... 996
8.6.2 Reduce inactive session timeout to 15 minutes ... 996
8.6.3 Use a URL from your domain and create a signed certificate ... 997
8.6.4 Disable HTTP access ... 997
8.7 Repave Delphix Engine ... 997
8.7.1 Overview ... 997
8.7.1.1 Prerequisites ... 997
8.7.2 How to Repave a Delphix Engine ... 998
8.7.3 Repave API calls ... 998
8.7.3.1 Tips and usage guidelines ... 1000
8.7.4 POST calls ... 1000
8.7.4.1 Establish session ... 1000
8.7.4.2 Login ... 1000
8.7.4.3 Enable feature flag ... 1001
8.7.4.4 Repave prepare ... 1001
8.7.4.5 Repave apply ... 1002
8.7.4.6 Repave status ... 1004
8.7.4.7 Repave preview ... 1005
8.8 Masking sensitive data ... 1006
8.9 Audit logs ... 1006
8.9.1 Review audit logs monthly ... 1006
8.9.2 Forward audit logs to central server via syslog ... 1007
8.10 Support security ... 1007
8.10.1 Delphix operating system (DxOS) ... 1007
8.10.1.1 Disable support access (optional) ... 1007
8.11 Password policies ... 1007
8.11.1 Getting started ... 1007
8.11.2 Understanding password policies ... 1008
8.11.3 Default password policy ... 1008

8.11.4 Changing the password policy 1008
8.11.4.1 Who can change password policy for whom 1008
8.11.5 What operations can be done by administrators 1009
8.11.6 Password policy parameters 1009
8.11.7 Restrictions 1009
8.12 Additional topics 1009
8.12.1 Perform a yearly audit 1009
8.12.2 Port scan 1009
8.12.3 Security testing 1010
8.12.4 Security banner 1010
8.12.5 Virtual database security 1010
9 Datasets 1011
9.1 Getting started with datasets 1011
9.1.1 Datasets overview 1012
9.1.2 Dataset support 1012
9.1.3 General architecture 1014
9.1.3.1 Ingestion overview 1014
9.1.3.2 Staging Pull vs. Staging Push 1016
9.1.3.3 Provisioning 1017
9.1.4 Environment management 1017
9.1.4.1 Overview 1017
9.1.4.2 Environment configuration requirements 1018
9.1.4.3 Adding and discovering an environment 1020
9.1.4.4 Network requirements 1020
9.1.4.5 Environment users 1022
9.1.4.6 Delphix toolkit 1024
9.1.4.7 Java Development Kit (JDK) 1025
9.1.4.8 NFS encryptions 1029
9.1.4.9 Environment operations 1034

9.1.4.10 Validating host environments with HostChecker 1038
9.1.4.11 Manually recovering a database after V2P 1039
9.1.5 dSource management 1040
9.1.5.1 Overview 1040
9.1.5.2 Linking dSources 1040
9.1.5.3 The dSource timeflow 1040
9.1.5.4 The dSource snapshot 1041
9.1.5.5 dSource operations 1041
9.1.6 Virtual database (VDB) management 1046
9.1.6.1 Overview 1046
9.1.6.2 VDB Timeflow 1046
9.1.6.3 VDB Snapshot 1047
9.1.6.4 VDB operations 1047
9.1.7 Hook operations 1056
9.1.7.1 Overview 1056
9.1.7.2 Creating hook operations 1056
9.1.7.3 Hook operations templates 1058
9.1.7.4 Hook operations list 1060
9.1.7.5 Self-Service interaction with hooks 1063
9.1.8 Shell operations 1064
9.1.8.1 RunCommand operation 1064
9.1.8.2 RunBash operation 1064
9.1.8.3 Running processes in the background 1065
9.1.8.4 Examples of running background processes 1065
9.1.9 Other operations 1065
9.1.9.1 RunExpect operation 1066
9.1.9.2 Environment variables 1066
9.1.9.3 Secure credential management 1066
9.1.10 Policies 1068

9.1.10.1 Overview ... 1068
9.1.10.2 Default vs. Custom policies ... 1069
9.1.10.3 Setting different policies for objects in a group ... 1069
9.1.10.4 Create a custom policy ... 1071
9.1.10.5 Policies and timezones ... 1071
9.1.10.6 Configure retention on individual snapshots ... 1072
9.1.11 Basic troubleshooting ... 1073
9.1.11.1 Environment ... 1073
9.1.11.2 Delphix Toolkit ... 1073
9.1.11.3 dSource and virtual databases (VDB) ... 1073
9.1.11.4 Log management ... 1073
9.2 CockroachDB data sources ... 1074
9.2.1 Seed ... 1075
9.2.2 AWS S3 Backup ... 1075
9.2.3 Disk Backup ... 1076
9.2.4 CockroachDB virtualization process ... 1077
9.2.5 CockroachDB requirements and prerequisites ... 1077
9.2.5.1 CockroachDB support matrices ... 1078
9.2.5.2 Requirements for cockroachDB source databases ... 1079
9.2.5.3 Network and connectivity requirements for CockroachDB ... 1081
9.2.5.4 Sudo privileges for the CockroachDB environment OS user ... 1084
9.2.6 CockroachDB installation and upgrade ... 1086
9.2.6.1 Downloading the connector ... 1086
9.2.6.2 Installing the connector ... 1087
9.2.6.3 Upgrading the connector ... 1088
9.2.7 CockroachDB connector operations ... 1089
9.2.7.1 Discovering installations for CockroachDB ... 1089
9.2.7.2 Linking CockroachDB data sources ... 1093
9.2.7.3 Provisioning a CockroachDB virtual database ... 1100

9.2.8 Troubleshooting CockroachDB data sources 1107
9.2.8.1 CockroachDB Plugin logs 1107
9.2.8.2 CockroachDB Cluster logs 1107
9.2.9 CockroachDB developer's guide 1108
9.2.9.1 CockroachDB discovery operations 1109
9.2.9.2 CockroachDB sourceconfig operations 1109
9.2.9.3 CockroachDB linking operations 1109
9.2.9.4 CockroachDB provision operations 1111
9.3 Couchbase data sources 1113
9.3.1 Couchbase virtualization process 1113
9.3.2 Couchbase requirements and prerequisites 1114
9.3.2.1 Requirements for source databases 1114
9.3.2.2 Requirements for staging and target databases 1115
9.3.2.3 Staging prerequisites for ingestion using backups 1115
9.3.2.4 Couchbase support matrices 1115
9.3.2.5 Couchbase network and connectivity requirements 1117
9.3.2.6 Couchbase host environment requirements 1119
9.3.3 Couchbase installation and upgrade 1122
9.3.3.1 Prerequisites 1122
9.3.3.2 Downloading the connector 1122
9.3.3.3 Installing the connector 1123
9.3.3.4 Upgrading the connector 1123
9.3.3.5 Upgrading Couchbase version on source, staging, and target host 1123
9.3.4 Couchbase connector operations 1124
9.3.4.1 Discovering Couchbase installations 1124
9.3.4.2 Linking Couchbase data sources 1127
9.3.4.3 Provisioning Couchbase VDBs 1133
9.3.5 Couchbase troubleshooting 1141
9.3.5.1 Logging 1141

9.3.6 Couchbase developer's CLI guide 1141
9.3.6.1 Discovering a Couchbase environment 1141
9.3.6.2 Couchbase SourceConfig operations 1142
9.3.6.3 Couchbase linking operations 1143
9.3.6.4 Couchbase provisioning operations via CLI 1147
9.3.7 Couchbase data source limitations 1150
9.4 IBM Db2 data sources 1150
9.4.1 Overview 1150
9.4.1.1 IBM Db2 connector architecture 1150
9.4.1.2 Limitations of the IBM Db2 solution 1152
9.4.1.3 Support 1152
9.4.2 IBM Db2 virtualization process 1152
9.4.3 IBM Db2 requirements and prerequisites 1153
9.4.3.1 IBM Db2 support matrix 1153
9.4.3.2 Requirements for IBM Db2 source databases 1161
9.4.3.3 Requirements for IBM Db2 staging and target environments 1164
9.4.3.4 Sudo privileges for the environment users 1170
9.4.4 IBM Db2 installation and upgrade 1176
9.4.4.1 Downloading the connector 1176
9.4.4.2 Installing the connector 1176
9.4.4.3 Upgrading the connector 1177
9.4.5 IBM Db2 connector operations 1179
9.4.5.1 Discovering IBM Db2 installations 1180
9.4.5.2 Linking IBM Db2 data sources 1185
9.4.5.3 Provisioning IBM Db2 virtual databases 1208
9.4.6 IBM Db2 troubleshooting 1212
9.4.6.1 IBM Db2 advanced configuration 1212
9.4.6.2 IBM Db2 logging 1218
9.4.6.3 Managing IBM Db2 log rotation and log level 1218

9.4.7 IBM Db2 additional information 1219
9.4.7.1 Delphix Virtualization glossary 1219
9.4.7.2 IBM Db2 glossary 1220
9.4.8 IBM Db2 FAQS 1225
9.4.8.1 What are the best practices for taking a backup? 1225
9.4.8.2 How to take a full online backup of a db2 database? 1226
9.4.8.3 What is the user-side validation prior to the snapshot? 1226
9.4.8.4 How to use the Delphix Continuous Data Engine key pair? 1226
9.4.8.5 How to update the Db2 version level? 1227
9.4.8.6 How to match the DB2 common directory when the target host is added in multiple Delphix Continuous Data Engines? 1227
9.5 MongoDB data sources 1228
9.5.1 MongoDB virtualization process 1229
9.5.1.1 seed 1229
9.5.1.2 offlinemongodump 1229
9.5.1.3 onlinemongodump 1230
9.5.1.4 extendedcluster 1230
9.5.1.5 shardedsource (using MongoDB Ops Manager backup files) 1231
9.5.1.6 shardedsource (using cluster-to-cluster sync) 1231
9.5.1.7 nonshardedsource (using MongoDB Ops Manager backup files) 1232
9.5.1.8 stagingpush 1232
9.5.2 MongoDB ingestion types 1233
9.5.3 MongoDB requirements and prerequisites 1235
9.5.3.1 MongoDB support matrix 1235
9.5.3.2 Requirements for MongoDB source databases 1238
9.5.3.3 Requirements for MongoDB staging/target environments 1239
9.5.4 MongoDB installation and upgrade 1244
9.5.4.1 Downloading the connector 1244
9.5.4.2 Installing the connector 1245

9.5.4.3 Upgrading the connector ... 1246
9.5.5 MongoDB connector operations ... 1247
9.5.5.1 Discovering MongoDB installations ... 1247
9.5.5.2 Adding a MongoDB environment ... 1248
9.5.5.3 Adding a MongoDB source config ... 1250
9.5.5.4 Linking MongoDB data sources ... 1252
9.5.5.5 Taking a snapshot of the staging DB ... 1267
9.5.6 Provisioning MongoDB virtual databases (VDBs) ... 1268
9.5.6.1 Prerequisites ... 1268
9.5.6.2 Procedure ... 1269
9.5.6.3 Rewinding a VDB ... 1271
9.5.7 MongoDB logging ... 1272
9.5.8 MongoDB developer's guide ... 1272
9.5.8.1 MongoDB discovery operations ... 1273
9.5.8.2 MongoDB link operations ... 1273
9.5.8.3 MongoDB provision operations ... 1278
9.6 MySQL data sources ... 1280
9.6.1 Overview ... 1280
9.6.2 Limitations ... 1281
9.6.3 Support ... 1281
9.6.4 MySQL virtualization process ... 1281
9.6.5 MySQL requirements and prerequisites ... 1282
9.6.5.1 MySQL support matrix ... 1282
9.6.5.2 Requirements for MySQL source databases ... 1286
9.6.5.3 Requirements for MySQL staging and target environments ... 1288
9.6.6 MySQL installation and upgrade ... 1291
9.6.6.1 Downloading the connector ... 1291
9.6.6.2 Installing the connector ... 1292
9.6.6.3 Upgrading the connector ... 1293

9.6.7 MySQL connector operations 1294
9.6.7.1 Discovering MySQL installations 1294
9.6.7.2 Linking MySQL data sources 1301
9.6.7.3 Provisioning MySQL virtual databases 1316
9.6.7.4 Upgrading dSources and VDBs MySQL database versions 1321
9.6.8 MySQL troubleshooting 1327
9.6.8.1 Logs 1327
9.6.8.2 Common MySQL commands 1327
9.6.8.3 MySQL Plugin exit codes 1328
9.6.8.4 Version compatibility 1330
9.6.9 MySQL FAQs 1331
9.6.9.1 How to create and configure an environment (host) key pair? 1331
9.6.9.2 Why is the BLOB limited to 1GB in size? 1331
9.7 Oracle data sources 1332
9.7.1 Delphix architecture with Oracle 1332
9.7.1.1 Overview 1332
9.7.1.2 Environment linking and provisioning architecture 1333
9.7.2 Transparent data encryption and Delphix 1334
9.7.2.1 Overview 1334
9.7.2.2 Oracle TDE support for external key managers 1336
9.7.2.3 Terminology 1336
9.7.2.4 Requirements 1339
9.7.2.5 Wallet location configuration 1341
9.7.2.6 Required O/S permissions for the Delphix user 1342
9.7.3 Oracle environment management 1343
9.7.4 Quick start guide for Oracle on Linux 1343
9.7.4.1 Overview 1343
9.7.4.2 Deploy OVA on VMware 1344
9.7.4.3 Setup network access to Delphix engine 1345

9.7.4.4 Setting up the Delphix engine 1347
9.7.4.5 Requirements for Oracle hosts and databases 1351
9.7.4.6 Oracle hosts and databases 1351
9.7.4.7 Source host requirement 1351
9.7.4.8 Additional requirements for RAC environments 1353
9.7.4.9 Auto-discovery requirements 1353
9.7.4.10 Source database requirements 1354
9.7.4.11 Operating system specific requirements 1355
9.7.4.12 Additional target /staging host requirements 1355
9.7.4.13 Adding Oracle source and target environments 1357
9.7.4.14 Linking an Oracle data source 1358
9.7.4.15 Provisioning an Oracle VDB 1359
9.7.4.16 Next steps 1361
9.7.4.17 Script 1362
9.7.5 Quick start guide for Oracle on Solaris SPARC 1362
9.7.5.1 Overview 1362
9.7.5.2 Deploy OVA on VMware 1363
9.7.5.3 Setup network access to Delphix engine 1364
9.7.5.4 Setting up the Delphix engine 1366
9.7.5.5 Administrators 1367
9.7.5.6 System time 1368
9.7.5.7 Network 1368
9.7.5.8 Storage 1368
9.7.5.9 Outbound connectivity 1368
9.7.5.10 Authentication 1368
9.7.5.11 Summary 1369
9.7.5.12 Requirements for Oracle hosts and databases 1370
9.7.5.13 Oracle hosts and databases 1370
9.7.5.14 Source host requirement 1370

9.7.5.15 Additional requirements for RAC environments 1371
9.7.5.16 Source database requirements 1373
9.7.5.17 Operating system specific requirements 1374
9.7.5.18 Additional target /staging host requirements 1374
9.7.5.19 Deploy hostchecker to validate Delphix requirements 1375
9.7.5.20 Adding Oracle source and target environments 1377
9.7.5.21 Linking an Oracle data source 1378
9.7.5.22 Provisioning an Oracle VDB 1378
9.7.5.23 Next steps 1381
9.7.5.24 Script 1381
9.7.6 Datapatch and Delphix 1381
9.7.6.1 Overview 1381
9.7.6.2 Delphix Implementation of Datapatch Invocation 1382
9.7.6.3 When to select the Invoke Datapatch option in Delphix? 1383
9.7.6.4 Required OS Permissions for the Delphix User 1384
9.7.7 Oracle support and requirements 1384
9.7.7.1 Oracle matrix 1384
9.7.7.2 Requirements for Oracle hosts and databases 1392
9.7.7.3 Network and connectivity requirements for Oracle environments 1399
9.7.7.4 Sudo privilege requirements for Oracle environments 1405
9.7.7.5 Sudo file configuration examples for Oracle environments 1409
9.7.8 Managing Oracle environments and hosts 1413
9.7.8.1 Managing environment users 1414
9.7.8.2 Environment attributes for hosts with Oracle 1414
9.7.8.3 Adding an Oracle single instance or RAC environment 1417
9.7.8.4 Adding Oracle standalone and RAC targets in Kerberos 1419
9.7.8.5 Adding a database installation home to an Oracle environment 1423
9.7.8.6 Adding a database to an Oracle environment 1424
9.7.8.7 Adding a listener to an Oracle environment 1425

9.7.8.8 Enabling linking and provisioning for Oracle databases 1427
9.7.8.9 Linking an Oracle RAC source database using a standalone host environment 1427
9.7.8.10 Listener and JDBC verification..... 1429
9.7.8.11 Changing the hostname or IP address for Oracle source and target environments 1433
9.7.8.12 How to change ORACLE_HOME..... 1435
9.7.8.13 CLI support for partial-full backup in Oracle 1436
9.7.8.14 Managing cluster nodes of an Oracle RAC environment 1436
9.7.9 Linking data sources and syncing data with Oracle 1438
9.7.9.1 Linking data sources with Oracle..... 1438
9.7.9.2 Linking an Oracle pluggable database 1439
9.7.9.3 Staging push implementation for Oracle 1440
9.7.9.4 Data management settings for Oracle data sources 1452
9.7.9.5 Prepare and upgrade a Non-MT Oracle dSource to MT 1456
9.7.9.6 Repository templates for Oracle databases..... 1459
9.7.9.7 Using the PurgeLogs operation 1460
9.7.9.8 Advanced linking..... 1463
9.7.10 Provisioning and managing virtual databases with Oracle..... 1515
9.7.10.1 Overview of provisioning Oracle virtual databases 1515
9.7.10.2 Configuration settings for Oracle virtual databases..... 1517
9.7.10.3 Provisioning an Oracle VDB 1541
9.7.10.4 Provisioning an Oracle virtual pluggable database (vPDB)..... 1546
9.7.10.5 Provisioning a TDE-enabled vPDB 1549
9.7.10.6 A closer look at TDE provisioning 1571
9.7.10.7 Provisioning a vPDB from a non-multitenant source 1576
9.7.10.8 Provisioning a VDB from an encrypted Oracle database 1584
9.7.10.9 Managing Oracle RAC virtual databases with only a subset of configured RAC nodes available 1585
9.7.10.10 Managing multiple virtual PDBs in a virtual CDB 1586

9.7.10.11 Refreshing and rewinding a TDE-enabled vPDB 1589
9.7.10.12 Rotating TDE external key manager password 1593
9.7.10.13 Adding or Editing TDE External Key Manager Credential 1596
9.7.10.14 Adding or Editing the OKV Home 1598
9.7.10.15 Updating the Parent Database TDE Keystore Location 1599
9.7.10.16 Adding or Editing the Target Keystore Password 1600
9.7.10.17 Adding or Editing the TDE Keystores Root 1601
9.7.10.18 Timeflows for RAC provisioning of VDBs 1602
9.7.10.19 Upgrading an Oracle VDB, linked CDB, or a vCDB 1606
9.7.10.20 Customizing RAC instances after provision 1609
9.7.10.21 Migrating an Oracle VDB 1611
9.7.10.22 Migrating a vPDB 1612
9.7.10.23 Migrating a TDE-enabled vPDB 1613
9.7.10.24 Migrating a vPDB (from a linked CDB) to a higher Oracle version (linked CDB) 1622
9.7.10.25 V2P with an Oracle VDB 1624
9.7.10.26 V2P with a snapshot or timeflow-point 1639
9.7.10.27 Provisioning from a replicated Oracle VDB 1641
9.7.10.28 Provisioning a TDE-enabled vPDB to a RAC cluster target 1642
9.7.10.29 Refreshing or rewinding a broken/unusable virtual PDB 1643
9.7.10.30 Managing cluster instances of Oracle RAC virtual databases 1645
9.7.11 Oracle hook operations 1647
9.7.11.1 Oracle RAC 1647
9.7.11.2 Shell operations 1647
9.7.11.3 Other operations 1649
9.7.11.4 Oracle environment variables 1650
9.8 Oracle E-Business Suite (EBS) environments and data sources 1652
9.8.1 Oracle E-Business Suite and Delphix conceptual overview 1652
9.8.1.1 Why use the Delphix Continuous Data Engine with the Oracle E-Business Suite? 1653

9.8.1.2 Overview of linking and provisioning Oracle EBS 1653
9.8.1.3 DBTechStack dataset 1654
9.8.1.4 Database dataset 1654
9.8.1.5 AppsTier dataset 1654
9.8.1.6 How does it work? 1655
9.8.2 Oracle EBS virtualization process overview 1656
9.8.3 Oracle EBS requirements and prerequisites 1656
9.8.3.1 Oracle EBS support matrix 1657
9.8.3.2 Requirements for Oracle EBS source data sources 1664
9.8.3.3 Requirements for Oracle EBS target data sources 1667
9.8.3.4 Sudo privileges for the Oracle EBS environment users 1672
9.8.3.5 Privilege elevation requirements for Oracle EBS 1673
9.8.4 Host Checker for external connectors 1681
9.8.4.1 Overview 1681
9.8.4.2 Structure 1682
9.8.4.3 Workflows 1682
9.8.4.4 Prerequisites 1682
9.8.4.5 Instructions 1682
9.8.5 Oracle EBS installation and upgrade 1686
9.8.5.1 Downloading the connector 1686
9.8.5.2 Installing the connector 1686
9.8.5.3 Upgrading the connector 1687
9.8.6 Oracle EBS connector operations 1689
9.8.6.1 Oracle EBS discovery operations 1690
9.8.6.2 Oracle EBS linking data sources 1693
9.8.6.3 Provisioning Oracle EBS VDBs 1700
9.8.6.4 Refreshing and rewinding a TDE-enabled vPDB for Oracle EBS 1745
9.8.6.5 Starting and stopping a virtual EBS instance 1745
9.8.6.6 Rewinding a virtual EBS instance 1746

9.8.6.7 Refreshing a virtual EBS instance 1748
9.8.6.8 Enabling and disabling a virtual EBS instance 1749
9.8.6.9 Deleting a virtual EBS instance 1751
9.8.6.10 Modifying the AppsTier topology 1752
9.8.6.11 Replicating a virtual EBS instance 1752
9.8.6.12 Upgrading a Delphix Engine hosting a virtual EBS instance 1753
9.8.6.13 Migrating an EBS VDB 1753
9.8.7 Troubleshooting Oracle EBS 1755
9.8.7.1 EBS connector logging 1755
9.8.8 Oracle EBS additional information 1756
9.8.8.1 Oracle central inventory 1757
9.8.8.2 ETCC patch level 1758
9.8.8.3 Clean up before provisioning option 1758
9.8.8.4 Configuring a Delphix self-service data template for EBS 1758
9.8.8.5 Managing the APPS password 1759
9.8.8.6 Refreshing a target DBTier environment 1761
9.8.8.7 Managing the Weblogic password 1761
9.8.8.8 Working with EBS snapshots 1763
9.8.8.9 Monitoring Oracle EBS R12.2 AppsTier provisioning progress 1764
9.8.8.10 Monitoring Oracle EBS R12.2 DBTechstack provisioning progress 1764
9.8.8.11 Unsupported EBS versions and features 1765
9.8.9 Oracle EBS glossary 1765
9.8.9.1 delphix_os user 1765
9.8.9.2 oravis 1766
9.8.9.3 applvis 1766
9.8.9.4 WLS (Web Logic Server) 1766
9.8.10 Oracle EBS FAQs 1766
9.8.10.1 What are important notes that are needed for EBS? 1767

9.8.10.2 How to use the Delphix Continuous Data Engine key pair for Oracle EBS? ... 1767
9.9 PostgreSQL data sources ... 1767
9.9.1 Overview ... 1767
9.9.2 Limitations of the PostgreSQL solution ... 1768
9.9.3 Support ... 1768
9.9.4 PostgreSQL virtualization process ... 1768
9.9.5 PostgreSQL requirements and prerequisites ... 1769
9.9.5.1 PostgreSQL Support matrix ... 1769
9.9.5.2 Requirements for source databases ... 1776
9.9.5.3 Requirements for staging and target environments ... 1780
9.9.6 PostgreSQL installation and upgrade ... 1792
9.9.6.1 Downloading the connector ... 1792
9.9.6.2 Installing the connector ... 1793
9.9.6.3 Upgrading the connector ... 1793
9.9.7 PostgreSQL connector operations ... 1796
9.9.7.1 Discovering PostgreSQL installations ... 1796
9.9.7.2 Linking PostgreSQL data sources ... 1802
9.9.7.3 Provisioning PostgreSQL virtual databases ... 1828
9.9.8 PostgreSQL troubleshooting ... 1838
9.9.8.1 Advanced configuration ... 1838
9.9.8.2 Logging ... 1839
9.9.8.3 Managing log rotation and log level ... 1840
9.9.9 PostgreSQL additional information ... 1840
9.9.9.1 PostgreSQL glossary ... 1841
9.9.9.2 Important PostgreSQL connector settings ... 1842
9.9.9.3 PostgreSQL architecture for streaming replication ... 1842
9.9.10 PostgreSQL FAQs ... 1843
9.9.10.1 How to copy WAL files to the target environment? ... 1843

9.9.10.2 Database authentication methods for PostgreSQL connector datasets ... 1844
9.9.10.3 How to use the Delphix Continuous Data Engine key pair? ... 1845
9.10 SAP ASE data sources ... 1846
9.10.1 Delphix architecture with SAP ASE ... 1846
9.10.1.1 Linking architecture between SAP ASE and Delphix engine ... 1847
9.10.1.2 Environment setup ... 1848
9.10.1.3 Target hosts for ASE ... 1849
9.10.1.4 Validated sync and logSync ... 1850
9.10.2 Overview of ASE database encryption ... 1850
9.10.2.1 Delphix implementation of database encryption ... 1851
9.10.3 TLS security for Sybase ASE ... 1853
9.10.3.1 Implementing TLS support in the Delphix Continuous Data Engine ... 1853
9.10.4 Quick start guide for SAP ASE ... 1855
9.10.4.1 Deploy OVA on VMware ... 1858
9.10.4.2 Setup network access to Delphix engine ... 1859
9.10.4.3 Setting up the Delphix engine ... 1860
9.10.4.4 Requirements for SAP ASE hosts and databases ... 1864
9.10.4.5 SAP ASE source host requirements ... 1864
9.10.4.6 Source database requirements ... 1865
9.10.4.7 Target host requirements ... 1865
9.10.4.8 Target database requirements ... 1867
9.10.4.9 Adding SAP ASE source and target environments ... 1868
9.10.4.10 ASE manual discovery ... 1869
9.10.4.11 Creating an ASE environment ... 1870
9.10.4.12 Manually discovering a repository ... 1870
9.10.4.13 Updating a repository ... 1871
9.10.4.14 Enable linking and provisioning for SAP ASE environments ... 1872
9.10.4.15 Linking an SAP ASE data source ... 1872

9.10.4.16 ASE managed backups ... 1873
9.10.4.17 Delphix managed backups ... 1873
9.10.4.18 Procedure ... 1874
9.10.4.19 Provisioning an SAP ASE VDB ... 1875
9.10.4.20 Procedure ... 1875
9.10.4.21 Configuration settings for ASE virtual databases ... 1876
9.10.4.22 Automatic VDB restart on target server after reboot ... 1876
9.10.4.23 Next steps ... 1877
9.10.5 SAP ASE support and requirements ... 1877
9.10.5.1 SAP ASE matrix ... 1877
9.10.5.2 Requirements for SAP ASE environments and databases ... 1884
9.10.5.3 Network and connectivity requirements for SAP ASE environments ... 1892
9.10.5.4 Sudo privilege requirements for SAP ASE environments ... 1897
9.10.5.5 Sudo file configuration examples for SAP ASE environments ... 1900
9.10.6 Managing SAP ASE environments and hosts ... 1903
9.10.6.1 Managing SAP ASE environments overview ... 1904
9.10.6.2 Adding an SAP ASE environment ... 1905
9.10.6.3 Environment attributes for hosts with SAP ASE ... 1907
9.10.6.4 How to discover SAP ASE instances which use multiple network handlers ... 1909
9.10.6.5 Changing the host name or IP address of an SAP ASE environment ... 1911
9.10.6.6 Using HostChecker to validate SAP ASE source and target environments ... 1912
9.10.6.7 Enabling linking and provisioning for SAP ASE environments ... 1916
9.10.7 Linking data sources and Syncing Data with SAP ASE ... 1917
9.10.7.1 Linking SAP ASE data sources: an overview ... 1917
9.10.7.2 Linking data sources with SAP ASE ... 1918
9.10.7.3 Linking data sources with encrypted SAP ASE database ... 1921
9.10.7.4 Data management settings for SAP ASE dSources ... 1921
9.10.7.5 Upgrading a SAP ASE dSource ... 1924

9.10.7.6 Detaching and Re-attaching SAP ASE dSources 1924
9.10.7.7 Deleting an SAP ASE dSource 1926
9.10.7.8 Enabling and disabling SAP ASE dSources 1927
9.10.7.9 Working with SAP ASE snapshots 1928
9.10.8 Provisioning and managing VDBs from SAP ASE 1929
9.10.8.1 Overview of provisioning SAP ASE virtual databases 1929
9.10.8.2 Provisioning an SAP ASE VDB 1931
9.10.8.3 Provisioning a VDB from an encrypted SAP ASE database 1932
9.10.8.4 Provisioning an SAP ASE VDB from a replicated VDB or dSource 1933
9.10.8.5 Resizing an SAP ASE VDB 1934
9.10.8.6 V2P with an SAP ASE dSource or VDB 1939
9.10.8.7 Additional SAP ASE VDB topics 1940
9.10.9 Backup server best practices 1947
9.10.9.1 Recommendations/best practices 1947
9.10.10 SAP ASE hook operations 1948
9.10.10.1 Shell operations 1948
9.10.10.2 Other operations 1950
9.10.10.3 SAP ASE environment variables 1951
9.10.11 Support for dump history file 1952
9.10.11.1 Overview 1952
9.10.11.2 Prerequisites 1953
9.11 SAP HANA data sources 1953
9.11.1 SAP HANA overview 1954
9.11.1.1 Implications of the HANA column store 1954
9.11.1.2 Select 1955
9.11.1.3 Insert / Update 1955
9.11.2 Delphix architecture 1955
9.11.2.1 HANA plugin with pull architecture 1956
9.11.2.2 SAP HANA plugin with staging push architecture 1957

9.11.2.3 Operational best practice 1957
9.11.2.4 Benefits of HANA plugin 1958
9.11.2.5 Limitations of HANA plugin 1959
9.11.3 SAP HANA 2.0 plugin installation 1959
9.11.4 SAP HANA 2.0 plugin tools 1959
9.11.4.1 Prerequisites 1959
9.11.4.2 Interactive mode 1960
9.11.4.3 Providing the complete list of parameters 1962
9.11.5 Quick start guide for SAP HANA plugin 1963
9.11.5.1 Overview 1964
9.11.5.2 Deploy OVA on VMware 1964
9.11.5.3 Setup network access to the Delphix engine 1965
9.11.5.4 Setting up the Delphix engine 1967
9.11.5.5 Requirements for HANA staging/target host and databases 1971
9.11.5.6 Requirements for HANA target host and databases 1971
9.11.5.7 General target environment requirements 1972
9.11.5.8 Adding a HANA environment 1973
9.11.5.9 Prerequisites 1973
9.11.5.10 Procedure 1973
9.11.5.11 Linking a HANA data source 1976
9.11.5.12 Prerequisites for customer provided backups ingestion mechanism . 1976
9.11.5.13 Procedure 1977
9.11.5.14 Provisioning a HANA VDB 1979
9.11.5.15 Prerequisites 1979
9.11.5.16 VDB creation UI 1980
9.11.5.17 Procedure 1980
9.11.5.18 Next steps 1982
9.11.6 SAP HANA support and requirements 1982
9.11.6.1 SAP HANA matrix 1982

9.11.6.2 Requirements for SAP HANA staging hosts and databases 1988
9.11.6.3 Requirements for SAP HANA target hosts and databases 1989
9.11.6.4 Network and connectivity requirements for SAP HANA environments 1990
9.11.6.5 Sudo requirements for the SAP HANA 2.0 plugin 1990
9.11.7 Managing SAP HANA environments and hosts 1992
9.11.7.1 Workflow for SAP HANA environments 1993
9.11.7.2 Adding a SAP HANA environment 1993
9.11.7.3 Editing SAP HANA environment details 1996
9.11.7.4 Changing the hostname or IP address for SAP HANA staging and target environments 1997
9.11.7.5 Steps to modify SAP HANA dSource properties 1997
9.11.8 Linking data sources and syncing data with SAP HANA 1998
9.11.8.1 SAP HANA dSource ingestion 1998
9.11.8.2 Linking SAP HANA data sources 1999
9.11.8.3 Staging push implementation details 2002
9.11.8.4 Advanced data management settings for SAP HANA dSources 2005
9.11.8.5 Working with SAP HANA dSource snapshots 2006
9.11.8.6 Source sizing implementation for SAP HANA data sources 2008
9.11.9 Provisioning and managing VDBs from SAP HANA dSources 2008
9.11.9.1 Provisioning SAP HANA VDBs: an overview 2009
9.11.9.2 Provisioning SAP HANA VDBs 2010
9.11.9.3 Working with SAP HANA VDB snapshots 2013
9.11.9.4 Failure during provision 2013
9.11.9.5 SAP HANA hook operations 2014
9.12 SQL Server data sources 2020
9.12.1 Delphix architecture with SQL Server 2021
9.12.1.1 Staging push mechanism with SQL server 2021
9.12.1.2 Delphix in multi-domain windows environments 2022

9.12.2 Quick start guide for SQL Server (Microsoft SQL Server on Windows) ... 2028
9.12.2.1 Overview ... 2028
9.12.2.2 Deploy OVA on VMware ... 2029
9.12.2.3 Setting up the Delphix engine ... 2032
9.12.2.4 System time ... 2034
9.12.2.5 SQL server source hosts and databases ... 2036
9.12.2.6 SQL server staging hosts and databases ... 2040
9.12.2.7 SQL server target hosts and databases ... 2044
9.12.2.8 Supported roles for failover cluster instances and always on availability groups ... 2046
9.12.2.9 Add the SQL server source environment ... 2048
9.12.2.10 Linking a SQL server data source (dSource) ... 2049
9.12.2.11 Procedure ... 2049
9.12.2.12 Provisioning a SQL server virtual database (VDB) ... 2050
9.12.2.13 Next steps ... 2052
9.12.3 SQL Server support and requirements ... 2052
9.12.3.1 SQL Server matrix ... 2052
9.12.3.2 Overview of requirements for SQL Server environments ... 2067
9.12.3.3 Requirements for Windows iSCSI configuration ... 2083
9.12.3.4 Network access requirements for SQL Server ... 2089
9.12.4 Managing SQL Server environments and hosts ... 2092
9.12.4.1 Overview of setting up SQL Server environments ... 2092
9.12.4.2 Environment attributes for hosts with SQL Server ... 2096
9.12.4.3 Installing the Delphix connector service on the target database servers ... 2098
9.12.4.4 Upgrading the Delphix connector ... 2105
9.12.4.5 Reducing the number of AD logins to the domain controller ... 2106
9.12.4.6 Manual discovery for SQL Server instances ... 2108
9.12.4.7 Adding a SQL Server source environment ... 2110

9.12.4.8 Adding a SQL Server standalone target environment 2111
9.12.4.9 Adding a SQL Server failover cluster target environment 2113
9.12.4.10 Additional SQL Server environment topics..... 2117
9.12.5 Linking data sources and syncing data with SQL Server 2139
9.12.5.1 Linking a dSource from a SQL server: An overview 2139
9.12.5.2 Linking data sources with SQL server 2144
9.12.5.3 Staging push implementation for SQL server 2145
9.12.5.4 Data management settings for SQL server data sources 2158
9.12.5.5 Additional SQL Server dSource topics..... 2160
9.12.6 Provisioning and managing virtual databases with SQL Server 2180
9.12.6.1 Overview of SQL Server virtual databases 2180
9.12.6.2 Configuration settings for SQL Server virtual databases 2181
9.12.6.3 Provisioning a SQL Server VDB 2183
9.12.6.4 V2P with a SQL Server VDB..... 2187
9.12.6.5 Additional SQL Server VDB topics 2188
9.12.6.6 CDC support in SQL Server..... 2195
9.12.6.7 MSSQL V2P file mapping 2197
9.12.7 Hooks for SQL Server 2202
9.12.7.1 Overview 2202
9.12.7.2 Hook operation templates..... 2204
9.12.7.3 Windows environment variables..... 2205
9.12.7.4 Python script to migrate hooks from PowerShell version 2 to host's default powerShell version. 2205
9.12.7.5 SQL Server hook operation notes 2209
9.12.7.6 Using pre- and post-scripts with SQL Server dSources 2212
9.12.7.7 Using pre- and post-scripts with SQL Server VDBs..... 2214
9.12.8 Using hostChecker to validate target database servers 2217
9.12.8.1 Prerequisites 2217
9.12.8.2 Procedure to validate target environments..... 2217

9.12.8.3 Tests run ... 2218
9.12.8.4 Additional options ... 2219
9.13 Unstructured files and app data ... 2219
9.13.1 Getting started with unstructured files ... 2219
9.13.2 Unstructured files support and requirements ... 2219
9.13.2.1 vFiles matrix ... 2220
9.13.2.2 Unstructured files environment requirements ... 2226
9.13.3 Create an empty VDB for unstructured files in the Delphix Engine ... 2251
9.13.3.1 Prerequisites ... 2251
9.13.3.2 Procedure ... 2251
9.13.4 Provisioning unstructured files as vFiles ... 2252
9.13.4.1 Overview ... 2252
9.13.4.2 Prerequisites ... 2252
9.13.4.3 Post provision/migration ownership rules ... 2253
9.13.4.4 Procedure ... 2253
9.13.5 Managing vFiles ... 2254
9.13.5.1 Overview ... 2254
9.13.5.2 Adding an additional mount ... 2254
9.13.5.3 Post provision/migration ownership rules ... 2254
9.13.6 vFiles best practices and common pitfalls ... 2255
9.13.6.1 Overview ... 2255
9.13.6.2 Best practices and implementation ... 2255
9.13.6.3 Common use cases ... 2255
9.13.6.4 Common pitfalls ... 2255
9.13.6.5 Understanding and implementing scripts and dxtoolkit ... 2256
9.13.6.6 Conclusion ... 2257
9.13.7 Delphix Engine plugin management ... 2257
9.13.7.1 Plugin management ... 2257
9.13.7.2 Plugin types ... 2257

9.13.7.3 Installation of plugin ... 2258
9.13.7.4 Upgrading a plugin ... 2259
9.13.7.5 Notes on replication with objects administered by a plugin ... 2260
9.13.7.6 Upgrading an inactive plugin ... 2261
9.13.8 Unstructured files hook operation notes ... 2263
9.13.8.1 Shell Operations ... 2263
9.13.8.2 Other operations ... 2265
9.14 Yugabyte data sources ... 2266
9.14.1 Yugabyte virtualization process ... 2267
9.14.2 Yugabyte requirements and prerequisites ... 2268
9.14.2.1 Yugabyte support matrix ... 2268
9.14.2.2 Additional Yugabyte staging prerequisites for ingestion using backups ... 2270
9.14.2.3 Yugabyte network and connectivity requirements ... 2271
9.14.2.4 Yugabyte host environment requirements ... 2273
9.14.3 Yugabyte installation and upgrade ... 2276
9.14.3.1 Prerequisites ... 2276
9.14.3.2 Downloading the connector ... 2276
9.14.3.3 Installing the connector ... 2276
9.14.3.4 Upgrading the connector ... 2278
9.14.3.5 Upgrading the Yugabyte version on the source/staging/target host ... 2279
9.14.4 Yugabyte connector operations ... 2279
9.14.4.1 Discovering Yugabyte installations ... 2279
9.14.4.2 Linking Yugabyte data sources ... 2283
9.14.4.3 Provisioning yugabyte virtual databases ... 2293
9.14.5 Yugabyte troubleshooting ... 2299
9.14.5.1 Logging ... 2299
9.14.6 Yugabyte FAQs ... 2299
9.14.6.1 How to use the Delphix Continuous Data Engine key pair? ... 2299

9.14.7 Yugabyte developers guide 2299
9.14.7.1 Yugabyte discovery operations 2300
9.14.7.2 Yugabyte sourceconfig operations 2300
9.14.7.3 Yugabyte linking operations 2301
9.14.7.4 Yugabyte provisioning operations 2303
10 Best practices 2306
10.1 Introduction to Delphix architecture 2306
10.2 Hypervisor and host 2306
10.3 Network 2306
10.4 Storage 2306
10.5 Data protection 2306
10.6 Source DB and OS settings 2307
10.7 Target DB and OS settings 2307
10.8 Staging target 2307
10.9 Architecture checklist (consolidated best practices) 2307
10.10 Architecture checklist FAQ 2307
10.11 Architecture checklist FAQ 2307
10.11.1 Architecture best practices overview FAQ 2307
10.11.1.1 Why go through an architecture or sizing process? 2307
10.11.1.2 Why does Delphix collect IO data? 2308
10.11.1.3 Why is the customer database and inventory (CDI) important? 2308
10.11.1.4 Why does Delphix use a virtual machine? Shouldn't all high-performance applications use physical infrastructure? 2308
10.11.2 Architecture best Practices for hypervisor host and VM guest - ESX FAQ 2308
10.11.2.1 Why does Delphix recommend 8 vCPUs and 128GB of memory per 8 vCPUs? 2308
10.11.2.2 Why does Delphix request reservations for CPU and memory? 2308
10.11.2.3 Why does Delphix request Hyper-Threading (HT) be disabled? 2309

10.11.2.4 Why does Delphix request 4 controllers, and why must the storage be identical between them? 2309
10.11.2.5 Why must virtual disks (VMDKs) be thick provisioned and eager-zeroed? 2309
10.11.2.6 Why is 20% free space required? 2309
10.11.2.7 Why does Delphix request you reserve CPU and RAM for Hypervisor overhead? 2309
10.11.2.8 Why does Delphix generally want VMWare HA enabled, but DRS disabled? 2309
10.11.2.9 Why does Delphix request you set power management to High-Performance Mode? 2310
10.11.3 Architecture best practices for storage FAQ 2310
10.11.3.1 Why does Delphix require 127GB of storage for the OS? 2310
10.11.3.2 Why does Delphix require our LUNS to be uniform and contain an equal quantity and capacity of VMDKs, yet thin provisioning is OK? 2310
10.11.3.3 Why does Delphix require < 10ms latency (95th percentile) storage? . 2310
10.11.3.4 Why does Delphix prefer to extend existing storage rather than simply add more while maintaining equal distribution? 2310
10.11.4 Architecture best practices for network FAQ 2311
10.11.4.1 Why does Delphix request 10GE Ethernet? 2311
10.11.4.2 Why does Delphix require < 1ms latency to TARGET servers and < 50ms to SOURCE servers? 2311
10.11.4.3 Why does Delphix request Jumbo Frames? 2311
10.11.4.4 How does Delphix avoid communication impact with non-jumbo frame hosts when Jumbo Frames are enabled on the Delphix Host? 2311
10.11.4.5 When does Delphix recommend NIC teaming? 2311
10.11.4.6 Why does Delphix recommend logical and physical and co-location? .2312
10.11.5 Architecture best practices for Delphix engine data protection FAQ ... 2312
10.11.5.1 Why does Delphix recommend SAN snapshots or Delphix replication for backup? 2312
10.11.5.2 What is the DXToolkit, and how can it help? 2312

10.11.5.3 Can I use a VMware-based backup solution such as VEEAM to backup my Delphix Engine? ... 2312
10.11.5.4 Can I use a VMware snapshot for backing up Delphix for a small window – for example, during an engine upgrade? ... 2312
10.11.5.5 Can I use a Storage snapshot solution to protect Delphix against Storage and Delphix corruption? ... 2313
10.11.5.6 Can I use RMAN to backup my VDBs just like a Physical database to provide extra protection? ... 2313
10.12 Architecture checklist ... 2313
10.12.1 Overview ... 2313
10.12.2 Architecture best practices for hypervisor host and VM guest - ESX... 2313
10.12.2.1 Hypervisor ... 2313
10.12.2.2 Virtual Machine Guest ... 2315
10.12.3 Architecture best practices for network configuration ... 2316
10.12.4 Architecture best practices for storage ... 2318
10.12.5 Architecture best practices for Delphix engine data protection ... 2319
10.12.6 Architecture best practices for Masking ... 2320
10.12.7 Architecture best practices for source DB and OS settings ... 2322
10.12.7.1 Oracle ... 2322
10.12.7.2 SQL server ... 2323
10.12.8 Architecture best practices for target DB and OS settings ... 2323
10.12.8.1 Target database application settings ... 2323
10.12.8.2 Target Host OS Settings ... 2324
10.12.9 Architecture best practices for Staging Targets ... 2325
10.12.9.1 Memory and CPU ... 2325
10.12.9.2 General guidance for staging servers (Multi-platform) ... 2326
10.12.9.3 Disk / Local storage ... 2326
10.12.9.4 Network requirements ... 2326
10.12.9.5 Windows and MSSQL Specific ... 2327
10.13 Best practices for hypervisor host and VM guest ... 2327

10.13.1 Hypervisor 2327
10.13.2 Virtual machine guest 2328
10.13.3 Best practices hypervisor host and VM guest ESX FAQ 2330
10.13.3.1 Frequently asked questions 2330
10.14 Best practices for network configuration 2331
10.14.1 Best practices network FAQ 2333
10.14.1.1 Frequently asked questions 2333
10.15 Best practices for Delphix engine data protection 2334
10.15.1 Best practices Delphix engine data protection FAQ 2335
10.15.1.1 Frequently Asked Questions 2335
10.16 Best Practices for source DB and OS settings 2337
10.16.1 Best practices 2337
10.16.1.1 Oracle 2337
10.16.1.2 SQL server 2337
10.17 Best practices for target DB and OS settings 2338
10.17.1 Target database application settings 2338
10.17.2 Memory and CPU 2339
10.17.3 Windows and MSSQL specific 2339
10.17.4 Network requirements 2340
10.17.5 Target Host OS Settings 2340
10.17.6 Receive side scaling (RSS) for windows staging target and targets 2341
10.17.6.1 Steps to implement RSS on windows 2341
10.18 Best practices for staging targets 2344
10.18.1 Memory and CPU 2344
10.18.2 General guidance for staging servers (Multi-platform) 2345
10.18.3 Disk / local storage 2345
10.18.4 Network requirements 2345
10.18.5 Windows and MSSQL specific 2346
10.18.5.1 Environment details 2347

10.18.5.2 Setup notes: 2347
10.18.5.3 Performance Scenario 1 2347
10.18.5.4 Performance Scenario 2 2347
10.19 Best practices for storage 2348
11 Data backup and recovery solutions 2350
11.1 Data backup and recovery solutions 2350
11.2 Delphix continuous vault 2350
11.2.1 Overview 2350
11.2.2 Replica continuous vault 2351
11.2.2.1 Advantages 2351
11.2.2.2 Implementation 2351
11.2.2.3 CLI functions 2352
11.2.3 Single engine continuous vault 2356
11.2.3.1 Advantages 2356
11.2.3.2 Implementation 2356
11.2.3.3 CLI functions 2356
11.2.4 Continuous vault alert system 2359
11.3 Backup and recovery strategies for the Delphix engine 2359
11.3.1 Backup and recovery requirements 2360
11.3.1.1 Failure points 2360
11.3.1.2 Recovery objectives 2361
11.3.2 Backup solution implementation 2361
11.3.2.1 Clustering 2361
11.3.2.2 Snapshots 2362
11.3.2.3 Replication 2362
11.3.2.4 Backup 2363
11.3.3 Deployment architecture 2363
11.3.3.1 Architectural components 2363
11.3.3.2 Fault Recovery Features 2364

11.3.4 Mapping requirements to solutions 2365
11.3.4.1 Feature capabilities 2365
11.4 Delphix replication 2367
11.4.1 Replication overview 2368
11.4.1.1 Replication features 2368
11.4.1.2 Replication details 2369
11.4.1.3 Resumable replication 2370
11.4.1.4 Replicating Delphix Self-Service templates 2370
11.4.1.5 Replication of non-data objects 2371
11.4.2 Forward Compatible Replication (FCR) 2371
11.4.2.1 Overview 2371
11.4.2.2 Replication matrix 2372
11.4.3 Replication concepts 2375
11.4.3.1 Received replicas or namespaces 2376
11.4.3.2 Failover and conflict resolution 2376
11.4.3.3 Enabling databases and environments 2376
11.4.4 Replication use cases 2377
11.4.4.1 Replicating to the public cloud 2377
11.4.4.2 Delphix engine deployed in the public cloud 2377
11.4.4.3 Disaster recovery 2378
11.4.4.4 Configuration steps 2378
11.4.4.5 Failover object management 2378
11.4.4.6 Geographically distributed development 2379
11.4.4.7 Data migration 2381
11.4.5 Configuring replication 2381
11.4.5.1 Requirements 2381
11.4.5.2 Configuring the network 2381
11.4.5.3 Configuring the replication source Delphix engine 2382
11.4.5.4 Replication profile options 2383

11.4.5.5 Configuring the target Delphix engine 2384
11.4.6 Controlled failover 2385
11.4.6.1 Prerequisites 2385
11.4.6.2 Procedure 2385
11.4.6.3 After replication failover 2386
11.4.7 Uncontrolled failover 2387
11.4.7.1 Uncontrolled failover procedure 2387
11.4.7.2 Oracle environment and data sources 2387
11.4.7.3 SAP ASE environment and data sources 2388
11.4.7.4 SQL server environment and data sources 2389
11.4.8 Managing replicated objects 2395
11.4.8.1 SSH keys 2395
11.4.8.2 Enabling environments 2396
11.4.8.3 Enabling dSources 2396
11.4.8.4 Enabling VDBs 2396
11.4.9 Replicas and failover 2397
11.4.9.1 Replicas 2397
11.4.9.2 Failover and conflict resolution 2397
11.4.9.3 Enabling databases and environments 2400
11.4.10 Test-failover and failback 2400
11.4.10.1 Test-failover 2400
11.4.11 Provisioning from replicated data sources or VDBs 2403
11.4.11.1 Prerequisites 2403
11.4.11.2 Procedure 2403
11.4.11.3 Post-requisites 2403
11.4.12 Replication user interface 2403
11.4.12.1 Sources for replication 2403
11.4.12.2 Targets for replication 2404
11.4.12.3 Canceling a replication job 2404

11.4.12.4 Replication UI 2404
11.4.12.5 Create new replication profile wizard 2406
11.4.12.6 Viewing and editing an existing replication profile 2407
11.5 Selective data distribution 2408
11.5.1 Support 2408
11.5.2 Selective data distribution overview 2408
11.5.2.1 Selective data distribution features 2409
11.5.2.2 Selective data distribution details 2409
11.5.2.3 Resumable selective data distribution 2410
11.5.2.4 Selective data distribution restrictions 2410
11.5.2.5 Selective data distribution supported platforms 2410
11.5.3 Selective data distribution use cases 2411
11.5.3.1 Geographically distributed development 2411
11.5.3.2 Best practices 2412
11.5.3.3 Migration 2413
11.5.4 Selective data distribution user interface 2413
11.5.4.1 Sources for selective data distribution 2413
11.5.4.2 Replication/selective data distribution section 2413
11.5.4.3 Create new profile 2414
11.5.5 Configuring selective data distribution 2415
11.5.5.1 Prerequisites 2415
11.5.5.2 Configuring the network 2415
11.5.5.3 Configuring the source Delphix engine 2416
11.5.5.4 Configuring the target Delphix engine 2418
11.5.6 Selective data distribution and failover 2418
11.5.6.1 Replicas 2418
11.5.6.2 Failover 2419
12 Delphix self-service 2420
12.1 Delphix self-service admin guide 2420

12.1.1 Getting started with delphix self-service 2422
12.1.1.1 Welcome to Delphix self-service 2422
12.1.1.2 User roles and permissions 2422
12.1.1.3 Login 2423
12.1.2 Delphix self-service concepts 2423
12.1.2.1 Data sources 2423
12.1.2.2 Data templates 2423
12.1.2.3 Data containers 2424
12.1.2.4 Delphix self-service data flow 2424
12.1.3 Navigating the Delphix self-service admin interface 2425
12.1.3.1 Overview screen 2425
12.1.3.2 Data template management 2426
12.1.3.3 User roles in admin App 2427
12.1.3.4 Data operations interface for Delphix administrators 2428
12.1.3.5 Administrator settings 2429
12.1.4 Understanding data templates 2431
12.1.4.1 Data templates: an overview 2431
12.1.4.2 Data template activities 2431
12.1.4.3 Viewing data templates 2437
12.1.5 Understanding how to manage data template details 2439
12.1.5.1 The data template details page 2439
12.1.6 Understanding data containers 2443
12.1.6.1 Delphix Self-Service data container overview 2443
12.1.6.2 Delphix Self-Service data container recovery 2446
12.1.6.3 Preserving independent containers in Delphix Self-Service during replication 2447
12.1.6.4 Creating independent containers 2448
12.1.7 Delphix self-service data container activities 2452
12.1.7.1 Configuring data containers in Delphix self-service 2452

12.1.7.2 Selecting Masked Data Sources for Data Containers 2454
12.1.7.3 Delete a Data Container 2456
12.1.7.4 Data management operations 2456
12.1.8 Using masked data sources with Delphix self-service 2457
12.1.8.1 SDD Overview 2457
12.1.8.2 Configuring Delphix Self-Service with Masked Data Sources 2458
12.1.8.3 Refreshing Masked VDBs in Delphix Self-Service Data Templates 2461
12.1.8.4 Limitations 2462
12.1.9 Understanding Delphix self-service user management 2462
12.1.9.1 User management activities 2462
12.1.9.2 Assigning a user to a data container 2463
12.1.9.3 User details page 2464
12.1.10 Working with multiple container owners 2465
12.1.10.1 How many owners should a container ideally be shared between? 2466
12.1.10.2 How should users handle potentially disruptive operations? 2466
12.1.10.3 Coordinating users 2467
12.1.10.4 What operations could disrupt others using a container? 2467
12.1.10.5 Where can I see which user has performed what operation? 2467
12.1.10.6 Where can I see which containers are unlocked/locked? 2468
12.1.11 Understanding bookmarks 2469
12.1.11.1 Bookmarks overview 2469
12.1.12 Understanding Delphix self-service usage management 2470
12.1.12.1 Usage management dashboard overview 2470
12.1.12.2 Template usage overview 2471
12.1.12.3 Template usage details 2472
12.1.12.4 User usage overview 2473
12.1.12.5 Template usage (Containers) overview 2474
12.1.12.6 Template usage (Bookmarks) overview 2476
12.1.12.7 Container usage (Branches) overview 2478

12.2 Delphix self-service data user guide 2479
12.2.1 Welcome to delphix self-service 2480
12.2.1.1 User roles and permissions 2480
12.2.1.2 Admin user 2480
12.2.1.3 Data user 2480
12.2.1.4 Login 2481
12.2.1.5 Changing your default locale 2481
12.2.2 Delphix self-service data concepts 2482
12.2.2.1 Understanding data sources 2482
12.2.2.2 Understanding data templates 2482
12.2.2.3 Understanding data containers 2482
12.2.2.4 Data flow 2482
12.2.2.5 Understanding branches 2483
12.2.3 Delphix self-service user interface 2484
12.2.3.1 Data container report panel 2485
12.2.3.2 Data container report panel 2485
12.2.3.3 Timeline 2485
12.2.3.4 Data container workspace 2485
12.2.3.5 Data container workspace 2486
12.2.3.6 User login and settings drop down menu 2486
12.2.3.7 Data container view panel 2486
12.2.3.8 Switch container 2486
12.2.3.9 Data container self-service toolbar 2487
12.2.3.10 Branch timeline 2487
12.2.4 Understanding timelines and how to preserve data in a point in time . 2487
12.2.4.1 Branch timeline 2487
12.2.4.2 Container Timeline 2488
12.2.4.3 Understanding the self-service toolbar 2490
12.2.4.4 Understanding how to preserve data in a point in time 2495

12.2.4.5 Understanding bookmarks 2497
12.2.4.6 Bookmarks tab in the data container view panel 2498
12.2.4.7 Bookmarks tile in the data container report Panel 2498
12.2.4.8 Bookmark sharing permissions 2499
12.2.4.9 Bookmark appearance 2499
12.2.4.10 Data container storage and retention for branches and timelines 2500
12.2.5 Data container activities 2500
12.2.5.1 Getting started 2500
12.2.5.2 Activity One: how to start and stop a data container 2500
12.2.5.3 Working with a branch, a branch timeline, and the self-service toolbar 2501
12.2.5.4 Activity Two: using reset from a bookmark to facilitate destructive testing 2501
12.2.5.5 Activity Three: using refresh to get the latest data from a data template 2504
12.2.5.6 Activity Four: using restore to return data back to a point in time 2504
12.2.5.7 Activity Five: restoring to a point on the parent template 2505
12.2.5.8 force option 2506
12.2.5.9 Activity Six: create a new branch and switch between branches 2506
12.2.5.10 Activity Seven: rename and/or delete a branch 2507
12.2.5.11 Activity Eight: restoring a data container to a consistent state with the recovery operation 2508
12.2.5.12 Activity Nine: working with container locks 2509
12.2.5.13 Errors 2510
12.2.6 Containers with multiple owners 2511
12.2.6.1 How many owners should a container ideally be shared between? 2512
12.2.6.2 How should users handle potentially disruptive operations? 2512
12.2.6.3 Conflicting operations 2512
12.2.6.4 Destructive operations 2512
12.2.6.5 Deleting objects 2513

12.2.7 Working with bookmarks in a data container 2513
12.2.7.1 Activity Nine: Share a bookmark with other Delphix self-service users 2513
12.2.7.2 Activity Ten: editing bookmarks 2514
12.2.7.3 Activity Eleven: filter and view bookmarks 2515
12.2.8 Understanding Delphix self-service usage 2516
12.2.8.1 Usage management Dashboard overview 2516
12.2.8.2 Container usage overview 2517
12.2.8.3 Bookmarks usage overview 2518
12.2.8.4 Branches usage overview 2519
13 Developer's guide 2521
13.1 Developer's guide 2521
13.2 Command line interface guide 2521
13.2.1 Command line interface overview 2521
13.2.1.1 Connecting to the CLI 2522
13.2.1.2 CLI contexts 2524
13.2.1.3 Managing objects 2525
13.2.1.4 Managing properties 2526
13.2.1.5 Array properties 2527
13.2.1.6 Untyped object properties 2527
13.2.1.7 CLI automation 2528
13.2.2 Delphix objects 2530
13.2.2.1 Object type hierarchy 2531
13.2.2.2 Object names and references 2531
13.2.2.3 Databases and environments 2533
13.2.2.4 Asynchronous jobs 2535
13.2.3 Command reference 2535
13.2.3.1 CLI help and display commands 2536
13.2.3.2 CLI context commands 2536

13.2.3.3 CLI object commands 2537
13.2.3.4 CLI miscellaneous commands 2537
13.2.3.5 CLI property commands 2538
13.2.4 CLI cookbook: common workflows, tasks, and examples 2538
13.2.4.1 Authentication and users 2539
13.2.4.2 CLI cookbook: system administration 2555
13.2.4.3 CLI cookbook: hosts and environments 2590
13.2.4.4 CLI cookbook: source databases and dSources 2598
13.2.4.5 CLI cookbook: VDBs 2618
13.2.4.6 CLI cookbook: replication 2701
13.2.4.7 CLI cookbook: Delphix self-service actions 2707
13.2.4.8 CLI cookbook: hooks and hook templates 2738
13.2.4.9 CLI cookbook: network performance 2740
13.2.4.10 Kerberos CLIs 2745
13.3 Web services API guide 2752
13.3.1 API version information 2753
13.3.2 Web service object model 2757
13.3.2.1 Object types 2757
13.3.2.2 Object references 2757
13.3.2.3 API operations 2758
13.3.2.4 Database object mModel 2758
13.3.2.5 Asynchronous execution 2759
13.3.3 Web service protocol 2759
13.3.3.1 Introduction 2759
13.3.3.2 Protocol operation 2760
13.3.3.3 Session establishment 2760
13.3.3.4 Authentication 2761
13.3.4 CLI to web services translation 2761
13.3.4.1 CLI translation to HTTP 2762

13.3.4.2 Tracing HTTP calls 2763
13.3.5 GUI API mapping 2764
13.3.5.1 dSource operations 2764
13.3.5.2 VDB operations 2766
13.3.5.3 Environment operations 2767
13.3.6 CLI to Python transition 2769
13.3.6.1 Installation 2769
13.3.6.2 Connecting to the Delphix engine 2770
13.3.6.3 CLI translation to Python 2770
13.3.6.4 Example: creating a group 2771
13.3.6.5 Asynchronous mode 2771
13.3.7 Python cookbook: adding a UNIX host 2772
13.3.7.1 Procedure 2772
13.3.8 Working with Delphix APIs 2774
13.3.8.1 Background information 2774
13.3.8.2 Delphix API reference URLs 2774
13.3.8.3 API prerequisite knowledge 2777
13.3.8.4 Delphix RESTFul APIs command line basics 2784
13.3.8.5 API shell scripts programming language examples 2789
13.3.8.6 JSON parsing 2793
13.3.8.7 API use case commands and scripts 2803
13.3.8.8 API programming language examples 2823
13.3.8.9 API timeflows 2827
13.3.8.10 API Cookbook: common tasks, workflows, and examples 2831

1 Welcome to the Delphix Continuous Data documentation!

This information explains how to deploy Continuous Data Engines for data virtualization, use its features, or tune its configurations for optimal performance. The content has been organized into several categories, available from the lefthand navigation.

List of Continuous Data documentation versions in PDF format.

- 22.0.0.0_ContinuousData.... [1]
- 21.0.0.0_ContinuousData.... [2]

1 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587417%2F22.0.0.0_ContinuousData.pdf
2 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587423%2F21.0.0.0_ContinuousData.pdf

- 20.0.0.0_ContinuousData.... [3]
- 19.0.0.0_ContinuousData.... [4]

[3] https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587420%2F20.0.0.0_ContinuousData.pdf
[4] https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587426%2F19.0.0.0_ContinuousData.pdf

- 18.0.0.0_ContinuousData.... [5]
- 17.0.0.0_ContinuousData.... [6]

[5] https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587435%2F18.0.0.0_ContinuousData.pdf
[6] https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587438%2F17.0.0.0_ContinuousData.pdf

- 16.0.0.0_ContinuousData.... [7]
- 15.0.0.0_ContinuousData.... [8]

7 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587429%2F16.0.0.0_ContinuousData.pdf
8 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587441%2F15.0.0.0_ContinuousData.pdf

- 14.0.0.0_ContinuousData.... [9]
- 13.0.0.0_ContinuousData.... [10]

9 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587444%2F14.0.0.0_ContinuousData.pdf

10 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587603%2F13.0.0.0_ContinuousData.pdf

- 12.0.0.0_ContinuousData.... [11]
- 11.0.0.0_ContinuousData.... [12]

[11] https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587600%2F12.0.0.0_ContinuousData.pdf
[12] https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587597%2F11.0.0.0_ContinuousData.pdf

- 10.0.0.0_ContinuousData.… [13]
- 9.0.0.0_ContinuousData.p… [14]

13 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587594%2F10.0.0.0_ContinuousData.pdf
14 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587591%2F9.0.0.0_ContinuousData.pdf

- 8.0.0.0_ContinuousData.p… [15]
- 7.0.0.0_ContinuousData.p… [16]

15 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587588%2F8.0.0.0_ContinuousData.pdf
16 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587582%2F7.0.0.0_ContinuousData.pdf

- 6.0.17.2_ContinuousData.... [17]
- 6.0.17.0_ContinuousData.... [18]

17 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587576%2F6.0.17.2_ContinuousData.pdf
18 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587573%2F6.0.17.0_ContinuousData.pdf

- 6.0.16.0_ContinuousData.... [19]
- 6.0.15.0_ContinuousData.... [20]

[19] https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587570%2F6.0.16.0_ContinuousData.pdf
[20] https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587567%2F6.0.15.0_ContinuousData.pdf

- 6.0.14.0_ContinuousData.... [21]
- 6.0.13.0_ContinuousData.... [22]

21 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587564%2F6.0.14.0_ContinuousData.pdf
22 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587561%2F6.0.13.0_ContinuousData.pdf

- 6.0.12.0_ContinuousData.... [23]
- 6.0.11.0_ContinuousData.... [24]

23 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587558%2F6.0.12.0_ContinuousData.pdf
24 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587555%2F6.0.11.0_ContinuousData.pdf

- 6.0.10.0_ContinuousData.… [25]
- 6.0.9.0_ContinuousData.p… [26]

25 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587552%2F6.0.10.0_ContinuousData.pdf
26 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587546%2F6.0.9.0_ContinuousData.pdf

- 6.0.8.0_ContinuousData.p… [27]
- 6.0.7.0_ContinuousData.p… [28]

27 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587543%2F6.0.8.0_ContinuousData.pdf
28 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587519%2F6.0.7.0_ContinuousData.pdf

- 6.0.6.0_ContinuousData.p… [29]
- 6.0.5.0_ContinuousData.p… [30]

29 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587510%2F6.0.6.0_ContinuousData.pdf
30 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587501%2F6.0.5.0_ContinuousData.pdf

- 6.0.4.0_ContinuousData.p… [31]
- 6.0.3.0_ContinuousData.p… [32]

31 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587492%2F6.0.4.0_ContinuousData.pdf
32 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587459%2F6.0.3.0_ContinuousData.pdf

- 6.0.2.0_ContinuousData.p… [33]
- 6.0.1.0_ContinuousData.p… [34]

33 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587456%2F6.0.2.0_ContinuousData.pdf
34 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587453%2F6.0.1.0_ContinuousData.pdf

- 6.0.0.0_ContinuousData.p… [35]

[35] https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149587394?preview=%2F149587394%2F149587447%2F6.0.0.0_ContinuousData.pdf

2 Quick references

- Delphix Engine overview[36]
- Standard deployment architecture[37]
- New features[38]
- Fixed issues[39]
- Support matrices[40]

36 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/99494807
37 https://cd.delphix.com/docs/latest/standard-deployment-architecture
38 https://cd.delphix.com/docs/latest/new-features
39 https://cd.delphix.com/docs/latest/fixed-issues
40 https://cd.delphix.com/docs/latest/support-matrices

3 Release notes

This section covers the following topics:

- New features (see page 94)
- Fixed issues (see page 120)
- Known issues (see page 296)
- API changes (see page 353)
- Support matrices (see page 428)
- Upgrade matrix (see page 454)
- Tested browser and operating systems (see page 457)
- Deprecated and end-of-life features (see page 457)
- Licenses and notices (see page 463)
- Data source integration (plugin) release notes (see page 464)

3.1 New features

3.1.1 Release 23.0.0.0

- **Elastic data for GCP**
  Support has been expanded to elastic data for Google Cloud Platform (GCP) object storage. Initially, support was limited to GCP block storage.
- **Sybase TLS/SSL**
  Delphix Continuous Data now supports security protocols for ASE instances that only use TLS/SSL connections authenticating via a client certificate.
- **PostgreSQL**
  Certification has been added for RHEL 9.x.
- **Documentation**
  Broad improvements continue across multiple areas of documentation, including Cassandra, MySQL, and SAP HANA. Documentation revisions have also been made to clarify new upgrade paths for all connectors.

3.1.2 Release 22.0.0.0

- **Enhanced Windows Active Directory log experience**
  The Delphix SQL Server install base is growing fast, and with that growth, our biggest users are experiencing excessive login entries to their Domain Controllers from Windows target hosts, and also from environment monitoring to targets and sources. Improvements have been made in Delphix code to optimize the AD logon calls to the Domain Controller.
- **API activity log shipping**
  Today, users are unable to ship an audit trail of API activity off-engine. This means they cannot

reliably detect vulnerability exploit attempts by analyzing Delphix logs. To respond to this important security requirement, Nginx access logs are now shipped to the Delphix Fluentd service.

- **Cassandra**
  This version is primarily a bug-fix release. Refer to the Cassandra Release Notes[41] for more details.
- **MySQL**
  Certified AWS RDS and RDS Aurora using the Staging Push ingestion mechanism. In addition, we streamlined the Staging Push configuration for all sources.
- **Oracle EBS**
  This version is primarily a bug-fix release. Refer to the Oracle EBS Fixed issues[42] for more details.
- **PostgreSQL**
  In v4.2.0, an issue was resolved where some expected tables were missing after a single database refresh or restore. This had the unintended effect of failing a previously working ingestion. This change has been removed and will be re-implemented in a future version. Refer to the PostgreSQL Fixed issues[43] for more details.
- **YugabyteDB**
  The YugabyteDB database is now supported. Added AWS S3 support.
- **Documentation updates**
  Broad improvements continue across multiple areas of documentation, including Oracle EBS.

3.1.3 Release 21.0.0.0

- **Documentation Updates**
  Broad improvements continue across multiple areas of documentation, including CockroachDB and MySQL.
- **Oracle**
  RedHat Enterprise Linux (RHEL) v9.0 and v9.3 are now supported for Oracle 19c.
- **IBM Db2**
  The connector now supports IBM AIX 7.2 and RHEL 9.2.
- **MySQL**
  Improvements have been made to the UI for linking a MySQL dSource in the wizard, where only properties applicable to the selected linking method are shown.

3.1.4 Release 20.0.0.0

- **Customizable local listeners for Oracle VDBs and Virtual CDBs**
  Database administrators need to configure database listener parameters for multiple reasons, notably for security, performance optimization and resource management. Previous improvements brought support to customize local listeners for Oracle VDBs in single-instance environments. This functionality has now been expanded to encompass the customization of local listeners for VDBs in

41 https://delphix.github.io/cassandra/Release_Notes/1.2.0/Release-v1.2.0/
42 https://cd.delphix.com/docs/latest/fixed-issues-ebs
43 https://cd.delphix.com/docs/latest/fixed-issues-postgresql

cluster environments, in addition to Virtual Container Databases in both cluster and single-instance environments.

  - **Customizable local listeners for Oracle MT RAC VDBs:** Delphix now supports customizing local listeners when provisioning a VDB in an Oracle RAC environment. In addition, this allows you to provision a vPDB into a new vCDB in a non-RAC or RAC environment, update local listeners for a VDB in a RAC environment, and update local listeners for a vCDB in a non-RAC or RAC environment.
- **Support Delphix operations for Oracle RAC VDBs/vPDBs during downtime:** Oracle RAC is used for production databases to ensure their availability 24/7. To comply with business continuity guidelines, Delphix now supports the following VDB/vPDB operations when one or more hosts of the Oracle RAC clusters are down: Provisioning, enable/disable, refresh, start/stop, and rollback. This enhancement includes VDBs/vPDBs in vCDBs and Linked CDBs.
- **Staging Push online DB mode for SQL Server**: Delphix now supports database online mode for Staging Push dSources for SQL Server, keeping the source database open for transactions, as needed. Before this enhancement, Staging Push source databases were required to be in restoring mode only. This enhancement is supported for SQL Server deployed on-premises and in the cloud (for both self-managed databases and SQL PaaS environments).
- **IBM Db2**: RHEL v9.2 is now supported for IBM Db2 v11.5, including IBM Db2 v11.5.9.
- **MySQL**: Added Staging Push support for AWS RDS and AWS RDS Aurora MySQL. Certified Amazon Linux 2 as staging and target host environments.
- **Oracle E-Business Suite (EBS)**: Oracle EBS's DB Tech Stack on IBM AIX v7.2 is now supported. The EBS App Tier is not supported on IBM AIX v7.2.
- **SAP HANA:** HANA SPS 06 and SPS 07 are now supported on the SUSE Operating Systems.
- **Documentation updates:** Broad improvements continue across multiple areas of our documentation, including Couchbase, IBM Db2, MongoDB, and MySQL.

3.1.5 Release 19.0.0.0

- **Documentation updates**
  Broad improvements have been made across multiple areas of documentation, including the Delphix Overview (see page 538), Product information (see page 539), Glossary (see page 541), Getting started in Datasets (see page 1011) section, and the PostgreSQL (see page 1767) dataset section.
- **UI for object storage connectivity modifications**
  Key rotation, endpoint URL, regions, and authentication types may now be modified for existing connected object storage through the UI, in addition to the previously available API and CLI options.
- **Improved migration time to Elastic Data**
  Leveraging Elastic Data with Continuous Data provides significant cost savings by leveraging object storage in addition to block storage. We have decreased the time to execute migrating an engine from a standard Continuous Data Engine to an Elastic Data Engine. A reboot is required to enable this feature enhancement.
- **TDE support for Hardware Security Module (HSM) solutions**
  Integration with HSM solutions such as Oracle Key Vault (OKV) and CipherTrust help enhance security by offering robust TDE capabilities that ensure encryption keys are securely managed and

stored. Delphix can now connect to OKV and CipherTrust to perform all operations within TDE environments managed by any of these solutions.

- **SAP HANA support**
  HANA SPS 06 and SPS 07 are now supported on SUSE Operating Systems.
- **Oracle EBS support**
  EBS on IBM AIX v7.2 is now supported.
- **MongoDB support**
  MongoDB is now supported on RHEL 9.0.

3.1.6 Release 18.0.0.0

- **Decreased VDB downtime**
  There are two types of upgrades for Continuous Data, full upgrades where VDB downtime is incurred during a system reboot, and deferred updates where there is no system reboot and therefore no downtime. Note that some features will require a full upgrade. The full upgrade process has been enhanced to minimize VDB downtime, limiting it to the duration of the system restart. VDBs now remain operational throughout the upgrade, experiencing downtime solely during the system reboot phase. This enhancement will be in effect for upgrades from 18.0.0.0 or greater.
- **Object storage modification**
  Key rotation, endpoint URL, regions, and authentication types may now be modified for existing connected object storage through the API and CLI.
- **NFS encryption**
  Adds support for Repave and Oracle RAC configurations with NFS encryption.
- **IBM Db2 HealthChecker**
  The HealthChecker can now validate staging and target environment configuration to simplify implementation. This joins existing support for Oracle E-Business Suite (EBS).
- **PostgreSQL**
  Provided WAL Logs will no longer automatically roll forward a dSource. This action will now occur only when a dSource snapshot is taken to eliminate unknown changes.
- **SAP HANA certification**
  Certified SAP HANA SPS 06 and 07.
- **ESXi 8.0 U2 certification**
  Certified VMware ESXi 8.0 U2.

3.1.7 Release 17.0.0.0

- **NFS encryption**
  Using encryption over NFS between Continuous Data Engines and the target and staging hosts is now supported. Please consult with your account team as to whether this is an appropriate fit for your needs.
- **MySQL on Linux:**
  ”Manual Ingestion” has been renamed “Staging Push” to standardize on the established Delphix

ingestion architectures. Strengthened the refresh and snapshot workflow to minimize occasional failures. Please review the upgrade path[44] for installation guidance.

- **Oracle E-Business Suite (EBS)**
  Updated the virtual database provisioning hooks to support the Solaris operating system. This release is recommended to all users as it contains various bugs and security fixes.
- **Couchbase support**
  Support has been added to define multiple buckets, configure RAM sizes, and ingest multiple full backups. Additionally, the connector will now report data source sizing.
- **Microsoft SQL server backup**
  Improved how easily you can bring in data from Microsoft SQL Server on Azure by automatically updating the Access Control Lists (ACLs), allowing users of various staging and target environments to better manage their files and folders.

3.1.8 Release 16.0.0.0

- **Bug fixes and performance improvements**
  This release focuses on a number of bug fixes and performance improvements. More details can be found on the Fixed issues[45] page.

3.1.9 Release 15.0.0.0

- **Improved SQL Server Access Control List (ACL) update time for file stream files (for VDB operations)**
  Mitigates delays caused by the longer Access Control List (ACL) update time compared to the transaction log interval, which is achieved by bypassing ACL checks during virtual database operations.
- **Fluentd logs download**
  Adds a capability that allows you to download Fluentd logs for debugging, as well as import the logs into analysis tools.
- **In-place conversion of an engine to Elastic Data (object storage)**
  System administrators can now convert an existing all-block-storage Continuous Data Engine to Elastic Data (with object storage), in-place, without the need to replicate to a new engine. Note that a reboot update is required to use this feature.
- **Certification of Oracle 19c and 21c on SUSE 15 SP5**
  Oracle 19c and 21c on SUSE 15 SP5 are now certified.
- **Datapatch**
  Adds support for invoking Datapatch against Oracle VDBs when they are provisioned, refreshed, rewound, started, or enabled via Delphix UI and CLI.
- **Export an Oracle non multitenant or PDB snapshot to a physical Oracle ASM or Exadata database**
  This feature enables you to export data from an Oracle non multitenant, PDB snapshot, or timeflow

44 https://cd.delphix.com/docs/19.0.0.0/mysql-matrix
45 https://cd.delphix.com/docs/19.0.0.0/fixed-issues

point to a physical Oracle database that uses Oracle Automatic Storage Management (ASM). This feature is especially useful for Oracle target environments running on Exadata or ExaCC systems.

- **IBM Db2**
  - **SSL/TLS support for HADR ingestion**
    The Db2 plugin introduces SSL/TLS support for HADR ingestion, enhancing data security during replication by encrypting data in transit between primary and standby databases.
  - **Dynamic UI for Db2 virtualization**
    With the introduction of various dynamic UI features, you can now experience a streamlined and user-friendly interface that adapts based on your selected choices, simplifying the linking process.
- **Redeploy support (Repave)**
  Continuous Data Engines may now be disconnected from their storage and redeployed, maintaining the previous configuration and data. Redeployment support only works with the same Delphix Engine version. Currently, engine redeployment is not supported on Elastic Data Engines, which is expected to be available in a subsequent release.

3.1.10 Release 14.0.0.0

- **Elastic Data**
  Previously referred to as “Cloud Engines”, Continuous Data with Elastic Data allows you to leverage lower-cost object storage in addition to traditional block storage. This has the effect of dramatically decreasing the operational overhead of Continuous Data while enabling new use cases like long-term archival and retention.
  - **Private data center Elastic Data**
    Previously, Elastic Data was only available for Continuous Data Engines deployed in AWS (using S3) or Azure (using BLOB storage). We now support deploying Elastic Data with on-premises, S3-compatible object storage arrays.
  - **Elastic Data on Oracle Cloud Infrastructure**
    Elastic Data may now be used in OCI, providing decreased operational overhead and enabling new use cases, as mentioned above.
- **Replication Failback**
  In the case of a failure on a primary Continuous Data Engine, failover may be used to swap operations to a secondary Engine. Previously, this was a one-time, terminal action. With failback, you can restore operations back to the primary engine if the failure has been resolved, or if you simply want to test the failover process.
- **PurgeLogs operation**
  The PurgeLogs operation now includes support for Oracle Multitenant dSources.
- **Oracle Staging Push**
  The Staging Push method of ingestion now supports point-in-time provisioning of data.
- **Couchbase**
  Couchbase 7.1.3 is now supported and several bugs have been resolved, such as increasing buffer size in the bucket-list command, backup restore failures, and replication status checks.

- **MySQL/Linux**
  There is now guidance for upgrading Source, Staging, and Target environments from MySQL 5.7 to v8.0. In addition, new guardrails have been introduced to prevent incompatible refresh and rollback operations.
- **PostgreSQL**
  All RHEL v8.x operating systems are now supported. In addition, new protections have been introduced to prevent accidental modification of parameters via VDB Config Template configuration.
- **SAP HANA**
  VDB provisioning has been improved in scenarios when SAP HANA services `scriptserver` and others are missing volume information.
- **Support for upgrading OpenJDK Java**
  Enables the ability to provide your own Adoptium OpenJDK upgrades without upgrading the product. Extends “Bring Your Own Java” to include Adoptium, not just Oracle.
- **ESXi 8.0 U1**
  Continuous Data may now be run on VMware ESXi 8.0 U1.

3.1.11 Release 13.0.0.0

- **Recovery from failed vPDB in a Linked CDB**
  A variety of factors may cause an Oracle virtual pluggable database to fail in a linked container database. It is now possible to recover from a failed Pluggable Virtual Database with a forced refresh or rewind operation.
- **IBM Db2**
  Support for RHEL 8.8 has now been certified for Delphix. This release was originally published in June.
- **MongoDB**
  Additional parameters are now supported in the `mongodump` and `mongorestore` commands. Backups and restores can be customized more flexibly to improve data extraction and ingestion speeds, such as parallel collections processing.
- **PostgreSQL**
  Support has been added for PostgreSQL 15.0 on RHEL 9.0 and Azure Flexible Server for PostgreSQL 14.0. The source sizing calculation has also been updated to more accurately represent the correct database size.
- **SQL Server**
  The following are now supported:
  - SQL Server 2016 on Windows Server Core 2016
  - SQL Server 2019 on Windows Server Core 2019
  - SQL Server 2022 on Windows Server Core 2022
  - SQL Server Developer Edition for all supported SQL Server versions

3.1.12 Release 12.0.0.0

- **User audit for Delphix support**
  In a case where engine access is granted to Delphix Support to debug or fix issues, or any actions take any actions, this is now tracked in its own audit log.
- **SMTP TLS**
  You can now add a certification for SMTP TLS authentication via the API or command line.
- **Refresh/Rewind of Oracle VDBs and vPDBs that have been exported to Oracle ASM Databases**
  Users may now refresh or rewind Oracle VDBs or vPDBs that have been exported to physical ASM databases. The performance of the export to ASM can also be improved with the ability to specify specific RMAN file section sizes.
- **IBM Db2**
  The v4.6.0 release supports Staging Push with the Database Partitioning feature (DPF) and custom mount points.
- **MongoDB**
  The v1.3.0 release supports passing additional parameters to mongodump and mongorestore. This adds parallel processing and flexibility to improve data extraction and ingestion speeds.

3.1.13 Release 11.0.0.0

- **Staging push read-only Oracle databases**
  Supports staging-push ingestion from read-only databases.
- **Partial ingestion via Oracle staging push**
  Partial ingestion enables the ability to ingest only selected Oracle tablespaces with Staging Push. This feature allows you to subset the Oracle database by choosing only specific tablespaces.
- **MongoDB**
  Supports MongoDB Enterprise 6.0 and Mongo Atlas, including cluster-to-cluster sync to enable ingestion of large, sharded databases.
- **MySQL/Linux**
  Supports MySQL v8 across all certified editions.
- **PostgreSQL**
  Supports point-in-time (PIT) provisioning using external logs and ingestion using the “pg_dumpall” utility.

3.1.14 Release 10.0.0.0

- **Azure key vault integration**
  This release adds Azure Key Vault support for storing and accessing secrets, keys, and certificates necessary for Continuous Data operations.
- **NFSv4 only mode**
  - This release adds support for turning off NFSv3 on the engine.

3.1.15 Release 9.0.0.0

- **READ_COMMITTED_SNAPSHOT parameter can now be set during VDB provisioning**
  This feature allows the SQL Server *READ_COMMITTED_SNAPSHOT* database parameter to be defined during VDB Provision operations, either by specifying it as a VDB Configuration Parameter or associating the VDB with a VDB Configuration Template. If defined, the parameter will be automatically set during Provision, Refresh, and Rewind operations for the VDB.
- **Export an Oracle virtual DB or virtual PDB to a physical Oracle ASM or Exadata database**
  This feature enables you to export data from an Oracle Virtual database (VDB) or a Multitenant (MT) vPDB to a physical Oracle database that uses Automatic Storage Management (ASM). This feature is especially useful for Oracle target environments running on Exadata or ExaCC systems.

3.1.16 Release 8.0.0.0

- **Oracle: Multiple virtual PDBs in a virtual CDB**
  This feature enables you to provision and manage multiple Oracle virtual PDBs to a virtual CDB (vCDB).
  - A vCDB is a Delphix-managed, fully compatible Oracle Container Database created and maintained by Delphix as part of the VDB provisioning workflow. This workflow allows organizations to automate their Oracle data lifecycle end-to-end, which results in developer productivity enhancements. Additionally, because container databases allow it, Oracle users benefit from the Oracle licensing agreement (which allows hosting up to three PDBs inside a vCDB).

> This feature is only supported for Oracle versions 12.1.0.2 (with a patch for Oracle bug 18967466) and later.

- **Source sizing for HANA**
  This feature enables you to calculate the source dataset size based on the total allocated space on the mount point provided by the Delphix Engine for the dataset.
- **Support for multiple backup paths in DB2**
  This feature supports multiple backup paths instead of one for dSource ingestion. This allows you to provide a single backup path for the dSource ingestion and add more backup paths with a dynamic UI to accommodate for the extra backup directories where the backup was split into.

3.1.17 Release 7.0.0.0

- **Oracle: Multitenant TDE for VCDBs**
  Use transparent data encryption (TDE) with virtual container databases to secure data-at-rest.

Existing TDE keys can be used or data can be rekeyed upon deployment. Automated KeyStore sanitization ensures production TDE keys are not shared to non-production environments.

- **Oracle: staging push**
  Pushes new data to the Delphix DevOps Data Platform instead of requesting data on a polling interval. Isolate ingests to Delphix from production instances. Gives the ability to leverage third-party backup applications. Full-tested integration with Oracle Data Guard and Active Data Guard allows for staging DB to be the standby DB, enabling tight point-in-time snapshots.
- **Cassandra: new Continuous Data connector**
  Continuously sync to Cassandra databases and deploy data to non-production environments. Deploy data from multiple Cassandra nodes as a single virtual database. Track source changes along a continuous time flow and redeploys data from any time. Quickly deploy data to new ephemeral instances. Branch production data, apply data masking, and deploy to new developer sandbox environments.

> ⚠ Multiple Device Removal for Delphix Engines version 6.0.12.0 to 6.0.17.X contains a breaking kernel module change that requires a reboot in order for the new module to load. With that, a deferred reboot engine upgrade operation will be unable to remove devices until a reboot is performed.

3.1.18 Release 6.0.17.0

- **Source sizing for Postgres and IBM Db2**
  This feature brings improved visibility for data source ingest management and metrics with data source size information logs. The source volume sizes can now be viewed through engine APIs and easily integrate with automation scripts.
- **Environment variables in hooks**
  This feature allows users to customize data deployments with hook scripts using environment variables, defined by plugin connectors in DLPX_DATA_DIRECTORY. Automated hook scripts can be executed during pre and post-deployment.

3.1.19 Release 6.0.16.0

- **Elastic Data engines on Microsoft Azure**
  You can now use Azure Blob object storage in place of block storage to reduce operational costs. Cache only necessary data to engines in order to maintain consistent VDB performance. Allows for elastic expansion of your storage footprint as-you-go in the cloud. Quickly deploy data to engines across regions, combining these tools to achieve potential cost savings of 25%-80% depending on use and workloads.
- **Elastic Data engines cache resize**
  This feature allows your Elastic Data engine cache to meet changing cost-performance demands on-the-fly by fine-tuning cache ratios for data ingestion and distribution use cases. Adapt to different

workloads, saving costs during idle situations and delivering high performance during heavy workloads.

- **Continuous Vault: additional security controls**
  New security controls are introduced for Ransomware Protection when using Delphix Continuous Vault (see page 2350). Alert Profiles, dSources, and LogSync profiles can now be locked on Continuous Data Source Engines. Delphix Continuous Ransomware Protection Solution provides Continuous Data Protection, Recovery, Detection, and Compliance.
- **Non-admin user password reset**
  Non-admin users will be able to request a password reset if an email is associated with the account. The password reset will continue to be controlled through LDAP/SSO for users on IdP platforms. Self-service password reset is available both within the UI and CLI.
- **Zero trust update for SAP HANA connector**
  A “least-privileged” OS user can now be used to run virtualization operations on a Delphix target host. Aligning with Zero Trust initiatives, Delphix no longer requires a high-level OS user for virtualization.
- **Increased mount security**
  This change brings improved security controls that limit connections to Delphix Engines only from a defined host list. NFS mount checks will now run on ephemeral mount points matched to specific host UIDs. Mount checks are used to verify that the Continuous Data Engine is mountable before running virtualization operations.
- **iSCSI parameter warnings**
  New warning to ensure iSCSI parameters on Windows Targets are optimized for Delphix Engines. The check runs for Target environments during Add, Refresh, and Enable operations.
- **Database port customization for EBS plugin**
  This feature supports the provisioning of VDBs in EBS dbTechStack & AppsTier plugins to run on a customized port. You can now provision database binaries(ORACLE_HOME) cloning and listener process to run on a custom DB port and Application tier to connect to that port.

3.1.20 Release 6.0.15.0

- **Oracle MT TDE for system tablespaces**
  The next release enables system tablespaces on Oracle Virtual Databases to be encrypted using transparent data encryption (TDE). Oracle system tables store database table information, index, sequences, and other objects. Encrypting system tables in non-prod environments enforce security controls and prevents database index and metadata tampering.
- **Staging push for Postgres**
  Staging push enables organizations to push data into Delphix without having to directly query production instances. With staging push, teams can use 3rd party solutions to recover data to staging instances that automatically sync to Continuous Data. This also enables logical replication workflows (common for PaaS sources) and the use of tools like pgbackrest.
- **Staging Push for IBM Db2 HADR**
  Staging push will be added in the next release for IBM Db2 in a High Availability Disaster Recovery (HADR) configuration. This gives teams an alternative way to sync data into the Continous Data.
- **Cache analysis for Elastic Data engines**
  We plan to add a UI for teams to better understand the effectiveness of allotted Delphix Elastic Data

engine cache size. This enables users to optimize DCE cache sizes to meet required cost-performance levels for VDBs.

- **Storage device removal UI**
  The team will be able to remove disks on engines backed by block storage through the UI in addition to the CLI in 6.0.15.
- **Zero trust for Postgres**
  6.0.15 supports custom “least privilege” OS user roles to be assigned to the “Delphix_OS” user to perform virtualization operations from Postgres sources.
- **TDE source database support for EBS plugin**
  6.0.15 supports TDE source databases for the EBS plugin.

Certifications

- Oracle 19c on RHEL 8.6 (6.0.15.0+)
- Oracle 21c on RHEL 8.6 (6.0.15.0+)
- AppData on RHEL 8.6 (6.0.15.0+)

3.1.21 Release 6.0.14.0

- The **Delphix DevOps data platform** will feature some product name changes to better align with our offered solutions:
  - **Continuous Data** (Virtualization)
  - **Continuous Compliance** (Masking)
  - **Continuous Vault** (Data vault)
- **Single engine Continuous Vault**
  The Single Engine Continuous Vault provides effective protection against ransomware attacks in a standalone Delphix Engine. This option may be preferable for deployments where maintaining two separate engines is not architecturally necessary. See Delphix Continuous Vault for more information.
- **AWS Elastic Data engines**
  This offering will support the use of AWS S3 object storage in place of traditional block storage, making it easier and more efficient to store more data for longer with Delphix. For this release, only engines deployed on AWS are supported, with support for other infrastructures to come.
- **Microsoft SQL Server Virtual-to-Physical (V2P) improvements**
  With these new improvements, users can specify a separate drive for SQL Server transaction log files, since these often need to split across multiple locations due to size. This ensures that DB files rehydrate in the manner that is most efficient during initial V2P, which helps avoid unnecessary downtime; this works in tandem with our ransomware solution. An additional enhancement to improve transfer performance involves replacing XCopy with Robocopy multithreading capabilities.
- **Oracle MT TDE support for encrypted system tablespaces**
  Every Oracle database has a system tablespace that is always used to store system data, including information about database tables, indexes, sequences, and other objects. Oracle TDE can encrypt, in addition to the data itself, the system tablespace for maximum security. This feature allows Delphix to provision VDBs with TDE-enabled encrypted system tablespaces to maintain the overall database security in non-production environments.

- **Staging push for SAP HANA via CommVault**
  CommVault is the most popular backup application for SAP HANA. We have certified that the backup application can be used as expected by our customers with our new Staging Push capability.
- **Salesforce data protect and version performance improvements**
  The March release will allow the user to configure up to 30 parallel upload threads (a 10x increase), improving our restore performance to ensure a fast recovery. We've also added several improvements towards faster and leaner retries for any records that fail during restore, plus increased the efficiency and reuse of API calls while downloading data (leading to faster backups and better resource utilization).
- **Ingest and restore for PostgreSQL**
  This feature allows you to ingest logical backups (pg_dump/pg_restore) from PostgreSQL, which enables workflows for direct ingestion from PaaS data sources (AWS Aurora & RDS), deployment to IaaS-hosted targets, and selective ingestion of individual databases from multiple database instances of PostgreSQL.

Certifications

- Oracle 19c on RHEL 8.5 (6.0.14.0+)
- Oracle 21c on RHEL 8.5 (6.0.14.0+)
- AppData on RHEL 8.5 (6.0.14.0+)

3.1.22 Release 6.0.13.0

3.1.22.1 New in this release

- **Support for AWS object storage**
  This release adds support for the use of AWS S3 object storage in place of traditional block storage. This release currently supports only those engines that are deployed on AWS.
- **Staging push for SQL Server**
  This release supports staging push for SQL Server. Staging push allows you to push data to Delphix if you have tools, configurations, or security concerns that do not work with the existing pull model. Staging push enables, for example, compatibility with data stored in backup systems and gives you control over the staging database. For more information, see Staging Push Implementation for SQL Server (see page 2145).
- **Support for TLS 1.3**
  This release provides support for TLS 1.3 for connections from the virtualization engine to the host and engine-to-engine communication.
- **Auto vPDB restart for single instance linked CDBs**
  This release supports the auto-restart feature for single-instance vPDBs in a linked CDB. The auto-restart feature allows the virtualization engine to detect and restart VDBs if the remote host has been restarted.
- **Transparent Data Encryption (TDE) for Oracle Multitenant RAC (Oracle 12.2 and 21c)**
  The Oracle TDE feature encrypts the sensitive data (database tables and tablespaces) stored on the disk. This prevents misuse of the data if the disks or storage mediums are lost or stolen. The data is transparently decrypted for authorized users when they access the data. Our large enterprise

customers leverage Oracle RAC configurations for their business-critical applications. This release adds support for Oracle 12.2 and 21c. For more information, see Provisioning a TDE-enabled vPDB to a Cluster Target (see page 1642).

- **FluentD monitoring**
  FluentD is a popular open-source data logging layer for DevOps and Site Reliability Engineering (SRE) teams[46] to stream telemetry to their chosen monitoring solution to improve systems observability. Delphix already provides the ability to forward data events and performance metrics to Splunk[47]. Delphix 6.0.13 enables additional Fluentd configurations to be developed and added to the engine. extending telemetry output[48] to new tools like ELK, New Relic, and DataDog.
- **Db2 plugin version 4.1.3**
  This plugin release resolves some issues that are listed in the Db2 Release Notes (see page 470).
- **EBS plugin version 3.0.3**
  This plugin release resolves some issues that are listed in the EBS Release Notes (see page 495).

3.1.22.2 Certifications

- **Virtualization**
  - Microsoft Windows Server 2022
  - Oracle E-Business Suite (EBS) 12.2.11 in OCI
- **Hypervisors/Clouds**
  - ESXi 7.0 U3c

3.1.23 Release 6.0.12.0

3.1.23.1 New in this release

- **HANA staging push**
  The HANA 2.0 plugin introduces the Staging Push feature. This feature provides a new data ingestion mechanism that helps users to push data into the Delphix-provided mount point on their own. For more information, see Delphix Architecture for HANA (see page 1955).
- **Multiple device removal**
  You can now remove multiple storage devices at a time when engines are over-provisioned or when moving to new storage. This is done only after ensuring sufficient resources to support the removal.
- **Transparent Data Encryption (TDE) for Oracle Multitenant RAC (Oracle 18c and 19c)**
  The Oracle TDE feature encrypts the sensitive data (database tables and tablespaces) stored on the disk. This prevents misuse of the data if the disks or storage mediums are lost or stolen. The data is transparently decrypted for authorized users when they access the data. Our large enterprise customers leverage Oracle RAC configurations for their business-critical applications. We have added support for provisioning a TDE-encrypted vPDB to a linked (Physical CDB) target RAC environment on

46 https://www.delphix.com/solutions/sre-teams
47 https://splunkbase.splunk.com/app/4373/
48 https://www.fluentd.org/dataoutputs

Oracle 18c and 19c in this release. For more information, see Provisioning a TDE-enabled vPDB to a Cluster Target (see page 1642).

3.1.23.2 Certifications

- Oracle 21c on SLES 15 SP3 (6.0.11.0+)
- Oracle 19c on SLES 15 SP3 (6.0.11.0+)
- Oracle 21c on SLES 15 SP2 (6.0.11.0+)
- Oracle 19c on SLES 15 SP2 (6.0.11.0+)
- AppData on RHEL 8.4 (6.0.11.0+)

3.1.24 Release 6.0.11.0

3.1.24.1 New in this release

- **OAuth2 API support**
  The Virtualization and Masking engine APIs are now accessible via OAuth2 tokens that improve Delphix's security offerings. For more information, see Configuring OAuth2 Authentication for API Access (see page 712).
- **Oracle support for Exadata, Exadata Cloud, or Exadata Cloud-at-Customer (ExaCC) cluster**
  This release adds support for Exadata, Exadata Cloud, or Exadata Cloud-at-Customer (ExaCC) Cluster for Oracle databases. For more information, see Oracle Support Matrix[49].
- **Apply Hotfixes using the self-service UI**
  You can now apply hotfixes using the self-service upgrade UIs.
- **TDE for Oracle Multitenant - support for rekey**
  Oracle Advanced Security Transparent Data Encryption (TDE) provides the ability to create virtual pluggable databases with a new key (independent from the source database). This is to facilitate another layer of security - ensuring that different keys are used in the production and non-production systems. For more information, see Provisioning a TDE-enabled vPDB (see page 1549).
- **Modify DBID for non-multitenant Oracle VDBs after provisioning**
  You can now generate a new DBID after a VDB is provisioned and refreshed. For more information, see Generate a New DBID for Oracle VDBs (see page 1527). Support for multitenant databases is currently not available.
- **New wizard for creating replication profile**
  A new multi-step wizard is now available that replaces the previous in-place editing option to manage Replication Profiles. For more information, see Replication User Interface (see page 2403).

3.1.24.2 Certifications

- Oracle 21c on RHEL 8.4, RHEL 8.3, SLES 15 SP1 (6.0.11.0+)
- Oracle 19c on RHEL 8.4 (6.0.10.0+)

---

49 https://cd.delphix.com/docs/latest/oracle-matrix

- Exadata Cloud-at-Customer(ExaCC)/Exadata support for Oracle 12.2 on OEL 7.8 and OEL 7.9

3.1.25 Release 6.0.10.0

3.1.25.1 Virtualization

- **SQL Server-support for Azure storage backups**
  SQL native backups can now be read directly from Azure Cloud storage. For more information, see Restoring SQL Backups Stored in Azure Cloud Storage (see page 2174).
- **Support dSource upgrades from Non-MT to MT in Oracle**
  Once a dSource is converted to a multitenant PDB, you will be able to share its storage blocks with its non-multitenant predecessor. Delphix will only store the incremental changes to the database. For more information, see Prepare and Upgrade a Non-MT Oracle dSource to MT (see page 1456).
- **Oracle-provision to the latest Point-In-Time on RAC**
  Provision to the latest point-in-time is now supported.
- **Flexible ORACLE_HOME permissions configuration**
  Removed the need to set permissions of the "$ORACLE_HOME/dbs" subdirectory using STARTUP SPFILE syntax to simplify Oracle operations. For more information, see Requirements for Oracle Hosts and Databases (see page 1392).
- **Support for manually starting an Oracle VDB**
  An Oracle VDB can now be manually started. For more information, see Manually Starting a VDB (see page 1539).
- **Attach and detach Oracle CDB containers**
  Detaching, attaching, and linking of the Oracle CDB containers is now supported via CLI. For more information, see CLI Cookbook: Attaching, Detaching, or Linking a CDB (see page 2620).
- **Password vaults and remote hooks for UI**
  In the 6.0.9.0 release, we introduced the ability to use password vaults with hooks. This allows our customers to ensure a high level of security with all operations with external systems. This can now be configured via the user interface. For more information, see the Secure credential management[50] section in the linked page.
- **NFSv4**
  NFSv4 is now enabled by default. For more information, see NFSv4 Configuration (see page 746).
- **12-month support for upgrades and Forward Compatible Replication (FCR)**
  To better align with the Delphix support program, Engine upgrades, and FCR operations must be within versions no more than 12 months apart. For example, upgrading to version 6.0.10.0 will require the previous version to be at least 6.0.4.0. For more information, see Upgrade Matrix[51] and Replication Overview (see page 2368).
- **HANA plugin port control**
  You can now keep the port numbers consistent throughout the HANA VDB life cycle so that the

50 https://cd.delphix.com/docs/latest/other-operations
51 https://delphixdocs.atlassian.net/wiki/pages/createpage.action?fromPageId=149533690&linkCreation=true&spaceKey=CD&title=%2823.0.0.0%29+Upgrade+matrix

connections made to the VDBs are not disrupted during their life cycle. For more information, see Provisioning HANA VDBs: An Overview (see page 2009).

- **User-specified mount path For Db2 dSource**
  You can now specify a mount path of your choice to host the dSource dataset on the target host. For more information, see the Db2 connector operations[52] section.
- **Staging push automation for Db2 dSource**
  You can now use a set of scripts that can be used to automate the restore and roll forward operations on the dSource.

3.1.25.2 Certifications

3.1.25.3 Virtualization

- **OCI VM.Standard.E4** (see page 651)
- **AWS r5n.24xlarge** (see page 625)
- **Oracle 21C RH 8.3**[53]

3.1.26 Release 6.0.9.0

3.1.26.1 Virtualization

- **Delphix data vault - additions**
  In continuation to the **Delphix data vault for ransomware protection** released in 6.0.8.0, this release enables you to manage this feature from the Delphix Engine user interface and also lets you monitor the regular valid replication received. For more information, see Delphix Data Vault (see page 2350).
- **TLS 1.3**
  The TLS 1.3 support is added as an available secure connection option at the Engine Admin Console to be used between engines.
- **Phonehome data collection frequency**
  The default collection period for the Phonehome users is now changed to daily. For more information, see System Configurations - Enable Phone Home[54].
- **Password vaults for remote hooks**
  With this release, the hooks running on environments can now obtain credentials from the engine and its configured password vaults. These credentials can be used to perform custom authentication tasks in a secure manner. Currently, this feature is supported via the command line interface only. For more information, see Passing Credentials Securely to Hook Operations[55].

---

52 https://cd.delphix.com/docs/latest/ibm-db2-connector-operations
53 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/4817620/Oracle+support+matrix
54 https://delphixdocs.atlassian.net/wiki/pages/createpage.action?fromPageId=149533690&linkCreation=true&spaceKey=CD&title=%2823.0.0.0%29+System+configuration
55 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/6262834/Passing+credentials+securely+to+hook+operations

- **Delphix integrations (dxi) docker image**
  A new dockerized version of the dxi library is now available. For more information, see Docker Image[56].

3.1.26.2 Certifications

- **Virtualization**
  - EBS 12.2 on Oracle 19c
  - Db2 11.1.4.6 Fix pack
- **Hypervisors/Clouds**
  - ESXi 7.0 U2

3.1.27 Release 6.0.8.0

3.1.27.1 Virtualization

- **Delphix data vault**
  The Delphix Data Vault for ransomware protection (accessible via CLI) enables organizations to recover access to their application data much faster than traditional backup solutions after malicious attacks. It relies on the new **data vault** Replication feature, which replicates critical business DB data stored on Delphix engines to a new target engine called Data Vault. Once securely stored on the Data Vault, the replicated DB data can be used to recover business applications upon a ransomware attack with very low RTO and RPO.
- **Dxi executable and support for encrypted credentials**
  We will be distributing the dxi CLI, a Delphix solution built to facilitate simpler and seamless integration of Delphix Platform Operations into existing workflows, such as Windows, macOS, and RHEL binaries. This will simplify adoption and remove the requirement on Python. Delphix has also added encryption for the login credentials.
- **Expansion of retention period on replicated objects**
  At present, when snapshots on the replication source engine are deleted (either due to retention policies or user action), the next replication job will delete those snapshots on the replication target engine. This improvement will allow you to extend the retention period of replicated objects on target engines while keeping the original retention at the source. Once the object in the target engine reaches its retention period, it will be flagged and deleted by the policy agent based on a daily schedule.
- **Db2 staging push**
  Delphix now supports Staging Push for non-DPF Db2 databases. The Staging Push architecture will allow organizations to bring their data to Delphix, with their own tools and standard processes. This facilitates the use of any backup tool, a major ongoing source of requests. This should dramatically increase the volume of data that can be easily managed by Delphix.

---

56 https://hub.docker.com/r/delphix/dxi

- **HANA plugin staged architecture**
  Delphix introduces a staging architecture for HANA virtualization. This will make it consistent with other virtualized data sources. This new architecture will build a foundation for future staging push capabilities, as Delphix has begun to introduce other platforms. These changes, together, will allow us to support more prospective organizations with various SAP-certified, 3rd-party backup applications for HANA. Delphix will continue to support the pull ingestion method with HANA native backups and logs.
- **ASE native encryption support**
  SAP ASE version 16.0 introduces the ability to fully encrypt databases and provides protection for all the data, indexes, and transaction logs in a database. This offers full database protection while allowing the user to query and manage the data as usual, as the encryption is transparent to existing functions. In response to customer demand, we have added a security enhancement to support encrypted ASE databases. This allows customers to maintain the ASE encryption that is active on their sources and propagate that through to their VDBs.
- **System tunable interface**
  A new web service API is introduced which allows you to set or receive the values of a system tunable via Delphix CLI. A Support engineer can now provide context via a Support case to modify these values.
- **Oracle customized full backups**
  There is a rarely-seen bug in Oracle that results in some blocks not being written to the datafiles during an Oracle SnapSync operation. When this happens, the datafiles can become incomplete and provisioning/refreshing from that snapshot might fail. We are providing a SnapSync option via CLI that you can customize to accept all datafiles during an Oracle SnapSync operation to prevent this error.

3.1.28 Release 6.0.7.0

3.1.28.1 Virtualization

- **Simplified connection management for Oracle databases**
  This feature streamlines the way that Delphix communicates and interacts with Oracle databases by simplifying the connection management infrastructure. Prior to this release, connections were established to Oracle databases using two different methods (remote connections from the Delphix Engine and local connections from the Delphix toolkit) and communication was performed with Oracle databases using two different users (a Delphix OS user and a Delphix DB user). Starting with this release, all communication with Oracle databases will be performed locally on the Oracle host and all connections to Oracle databases will be established using OS authentication. Existing Oracle dSources and VDBs will continue to function with no user intervention required. This feature results in several key benefits for Oracle DB customers such as the elimination of the requirement for a Delphix DB user when linking, automated PDB discovery, elimination of Delphix interaction with any network listener, and many more.
- **Virtualization SDK support for password vaults**
  Building off of the existing CyberArk and Hashicorp support for Oracle, SQL Server, and SAP ASE database user credentials, Continuous Data will extend password vault coverage to the virtualization

SDK (vSDK). This will enable data sources that are connected via a vSDK plugin to incorporate this more secure method of authentication.

- **SAP ASE device mapping improvements**
  The Continuous Data experience with SAP ASE heavily relies on and mirrors a database's device allocation from the initial load (creating the dSource) to provision (creating VDBs). As these source device allocations shift over time, Delphix maps these changes and propagates them to their associated Delphix objects. However, dramatic device layout changes can negatively impact performance. This enhancement provides a quality-of-life (QoL) improvement to the overall SAP ASE experience by providing better error handling and escape valves should a dSource get into a bad state due to a major device layout shift.
- **Improved storage utilization for large pools**
  Up through the 6.0.6 release, Continuous Data has enforced a storage usage limit of 85%. Once met, this limit will cause certain API operations to be disabled to ensure engine data integrity. In the 6.0.7 release, this threshold is relaxed significantly. The new thresholds are as follows:
  - "Warning": when 85% of the total storage quota is reached or 1536GB of free storage is remaining (whichever is less), which can be resolved/ignored, with no impact on system behavior.
  - "Critical": when 90% of the total storage quota is reached or 1024GB of free storage is remaining(whichever is less), which cannot be resolved/ignored, with some impact on system behavior.
  - "Minimum": when 95% of the total storage quota is reached or 512GB of free storage is remaining (whichever is less). In this case, a critical fault is raised and cannot be resolved/ignored, with a substantial impact on system behavior (stop policies, VDB operations, etc).
- **PVSCSI support**
  In addition to LSI Logic, with the 6.0.7 release, Delphix has added support for the VMware Paravirtual vSCSI controller (aka PVSCSI). While VMware designed PVSCSI to support very high throughput with minimal processing cost, the performance improvements on Delphix engines can vary from case to case. In 6.0.7, we also support manual changes from LSI Logic to PVSCSI for currently deployed engines.

3.1.29 Release 6.0.6.0

3.1.29.1 Virtualization

- **Solaris x86 to Linux x86 Oracle DB provisioning**
  This feature allows the provisioning of Oracle Virtual Databases from Solaris x86 dSources to Linux x86 target environments.
- **TDE for Oracle Multitenant**
  Oracle Advanced Security Transparent Data Encryption (TDE) provides the ability to encrypt sensitive application data on storage. Delphix will now support TDE for Oracle 12cR2, 18c, and 19c multitenant. This release introduces support for single-instance linked container databases (CDBs) using software keystores. Virtual Container Database (vCDBs), RAC, and rekeying of the TDE encryption keys are not supported in this release.

Note:
Please note the following important restrictions for **TDE for the Oracle Multitenant** feature:

- TDE-enabled vPDBs must be provisioned to a linked CDB, not a vCDB.
- RAC dSources and target CDBs are not supported.
- The Oracle version must be 12.2 or higher (12.1 is not supported).
- System tables or tablespaces either in the PDB or CDB must not be encrypted.
- Oracle Key Vault is not supported.
- Hardware keystores are not supported.
- Keystores must not be on ASM storage.
- The dSource from which the initial provision is done must be encrypted when it is linked. Existing dSources cannot be encrypted without unlinking and creating a new dSource.
- Encrypting an already-provisioned unencrypted vPDB (with clear data) which is managed by Delphix is not supported

- **Single to Multitenant VDBs**
  Oracle announced the end of support for non-multitenant databases in their 20c release, and as such, Oracle DB customers are planning their upgrade and migration programs. Delphix will now support provisioning a virtual pluggable database from a non-multitenant virtual database.
- Added support for **HashiCorp namespaces**.

3.1.29.2 Certifications

- **Virtualization**
  - ASE 15.7/16 on RHEL 7.9
  - Oracle 12.1 on RHEL 7.9 and SLES 12 SP5
  - Oracle 12.2 on RHEL 7.9 and SLES 12 SP5
  - Oracle 19c on RHEL 7.9 and SLES 12 SP5
- **Hypervisors/Clouds**
  - ESXi 7.0 U1

3.1.30 Release 6.0.5.0

3.1.30.1 Virtualization

- **NFSv4 support:**
  Support has been added for Oracle and ASE on AIX.
- **Expanded Replication: Replication of non-data objects**
  Our customers are increasingly using replication to facilitate moves of data across network boundaries, to the cloud, and for DR purposes. We've had long-standing requests to replicate more than just the data, and in this release, we will support the replication of users, roles, permissions, policies, and configuration templates.

- **Upgraded Windows Connector:**
  The Windows Connector will now support newer versions of Microsoft's .NET framework (4.x), which encompasses myriad higher security standards, new functionality, etc. Previously, the connector relied on .NET 3.5 due to two dependencies: SQL Server and Powershell, both have since been removed with SQL Server 2016+ and Powershell update in 6.0.3.0.
- **Db2 Extensible Ingestion:**
  We will now support an extensible model for ingesting Db2 data. In this new, additive, model, we will support customers manually performing a restore & roll forward of their staging database to Delphix from native backups or arbitrary third-party backup tools which integrate directly with Db2. This will allow customers to bring data from whatever system or backup they have and restore it to an exact point in time, as needed.

3.1.30.2 Certifications

- **Virtualization**
  - ASE 16.0 on RHEL8.1 and RHEL8.2 on 6.0.4+
  - ASE 16.0 on SLES12.4 on 6.0.4+
  - Oracle 19.7 on RH7.8 and RH8.0 on 5.3.9+ and 6.0.3+
  - ESX 7.0
  - NFS v4 support on AIX
  - IBM Cloud Catalog. Delphix is now available in the IBM Cloud Catalog, a private marketplace for trusted IBM Technology partners that is offered to large IBM enterprise customers. In 6.0.5 we will start with a few certified instances for virtualization and masking and will grow our presence as more the business justifies the cost and efforts. Specifically, we support the following instances:
    - mx2-8x64
    - mx2-16x128
    - mx2-32x256
    - mx2-48x38
  - Oracle Cloud: The following are newly supported instance types:
    - VM.Standard2.8
    - VM.Standard2.16
    - VM.Standard2.24

3.1.31 Release 6.0.4.0

3.1.31.1 Virtualization

- **HashiCorp and expanded CyberArk support:**
  Delphix has extended both CyberArk and HashiCorp Vault support to Oracle Database Users in addition to previously supported ASE and MSSQL domain users. GUI support for HashiCorp Vault has been added during setup to authenticate host users and database users.
  Note:
  The HashiCorp namespace Enterprise feature is supported starting 6.0.6.0.
- **NFSv4 support:**
  Support has been added for SuSE and Db2 on AIX.

3.1.31.2 Certifications

- **Virtualization**
  - OCI Support
  - NFSv4 support for Db2 on AIX and SuSE
  - SQL Server Instances with a Managed Service Account

3.1.32 Release 6.0.3.0

3.1.32.1 Virtualization

- **CyberArk and Hashicorp vault support for virtualization**:
  Delphix is introducing password vault support to authenticate environment and database linking and will support both CyberArk with Oracle and Hashicorp with CLI only.
- **Capacity management**:
  Understanding where and how storage is used on Virtualization Engines can be a challenge, in particular, understanding how and where space is held and how to recover it. In this release, Delphix provides better details of held space, particularly around locked objects, and provides clear instructions about what steps are required to free up space.
- **Diagnosability**:
  Additional performance health-check analytics in phone-home have been added to better troubleshoot and understand customer problems.
- **Powershell upgrade**:
  Delphix is reducing our requirements for Windows hosts running PowerShell by allowing you to use any PowerShell version from 2.0 to 5.1. Delphix will now use the default available PowerShell version on each host. When specifying hooks (such as “configureClone”), users may specify whether to use 2.0 or whatever PowerShell version is installed on the host.

- **Support for Oracle read-only homes:**
  Delphix is introducing support for Oracle read-only homes, which is a new Oracle feature starting with Oracle 18c. In a read-only Oracle home, all the configuration data and log files reside outside of the read-only Oracle home. This feature allows you to use the read-only Oracle home as a software image that can be distributed across multiple servers.
- **Replication performance:**
  Delphix will continue to improve replication performance for replication specifications that include multiple objects and single-object replication throughput.
- **SAP ASE support for VDB upgrade:**
  SAP ASE Customers will now be able to validate DBMS Upgrades with this feature that enables provisioning VDBs to a higher version than the source DB (e.g. ASE 15.7 > ASE 16).
- **Shared NFS for toolkits**
  With this release, Delphix introduces shared NFS for clustered environments. Customers wish to use a common NFS mount point, in which the Delphix toolkit for each cluster node can be deployed. The product today only creates a directory with appliance UUID and OS user in the folder name and uses this for detection to determine if a host is already managed by that Engine. As such, this prevents the customer from utilizing common NFS storage due to name conflict.
  When a new environment is created, upgraded or if the toolkit path is changed, a new toolkit is created with naming convention Delphix_COMMON_ for common directory and Delphix_ for user directory.
  With this change, the customer can use mounts on shared file systems (like NFS) as a toolkit path for clustered environments without any naming conflict.This change is not intended for windows environments.

3.1.32.2 Certifications

- **Virtualization**
  - EBS 12.2 with RHEL 7.6
  - PostgreSQL 12.1 & 12.2 with RHEL 7.8
  - Oracle 11g R2 and Oracle 19c with RHEL 7.8 on 5.3.9.0 and 6.0.2+

3.1.33 Release 6.0.2.0

3.1.33.1 Virtualization

- **Support for Db2 Database Partition Feature (DPF)**:
  Delphix has long supported distributed Db2 (running on Unix/Linux Systems). However, Db2 supports partitioned databases as a means of scaling to larger, more complex systems. With this release, Delphix will now support Db2 DPF allowing you to scale to an increasing number of your Db2 databases.
- **Windows authentication for SQL Server**:
  You will now be able to use Windows Authentication to link SQL Server databases. Rather than providing both a database user and a Windows user to ingest data, you can leverage one set of

credentials (a Windows OS user) to perform all source operations. This capability will simplify SQL Server deployments and reduce Delphix's security requirements on source databases.

- **Smart failover**:
  Smart Failover allows the Delphix Administrator to simplify failover processes by automating object conflict resolution. By selecting a new option "Automate Object Conflict Resolution" before the failover process starts, the failover process will rename all conflicting objects and show a report of all object changes at the end.
- **NFSv4 support:** In 6.0.2 Delphix will start providing NFSv4 for data sources running on RedHat 7.0 or later. NFSv4 can be enabled using the CLI. Support for additional host OS versions will be added in subsequent releases. Delphix will consider enabling NFSv4 by default for those supported configurations in a future release.
- **Support bundles not required for upgrade**: When upgrading from 6.0.0 or greater to a release 6.0.2 or greater, we no longer require support bundles to be sent to Delphix. This allows you to execute more self-service upgrades.

3.1.33.2 Certifications

- ASE 16.0 on AIX 7.1
- AWS r5n Instance Support: r5n.2xlarge, r5n.8xlarge, r5n.16xlarge
- Azure E Series Instance Support: E8s_v3, E16s_v3, E32s_v3
- Masking support of Oracle 19c.

3.1.34 Release 6.0.1.0

- **Masking extended connectors**: A very common request for masking has been to support additional data sources, outside of the currently supported list[57]. Thus, the next step in the strategy is the release of Masking Extended Connectors, which will allow our customers to add JDBC drivers to the masking engine to facilitate the masking of additional data sources. This will allow masking to be used for other common databases that can be accessed via JDBC, like SAP HANA, Informix, etc.
- **SQL server CDC support**: We have expanded our support for SQL Server databases using Change Data Capture[58] (CDC), a SQL Server feature that captures all the change information that is applied to the databases and stores it in change tables. Now, users will have the ability to preserve CDC data and enable CDC for SQL Server VDBs.

3.1.34.1 Certifications

- **Virtualization**
  - ASE 16 and 15.7 with Solaris SPARC 11U3 and SPARC 11U4
  - ASE 16 and 15.7 with RHEL 7.7

57 https://masking.delphix.com/docs/latest/data-source-support

58 https://docs.microsoft.com/en-us/sql/relational-databases/track-changes/about-change-data-capture-sql-server?view=sql-server-ver15

    - SQL Server 2019 Support with Windows 2016 and Windows 2019
    - Oracle 19c with SUSE SLES 15 SP1
    - Oracle 19c with Solaris 11 U4 and U3 x86
- **Hypervisors**: The following hypervisors have been certified in 6.0.1.
    - VMware ESX 6.5 U1, U2, U3
    - VMware ESX 6.7 U3

3.1.35 Release 6.0.0.0

Google cloud support: Delphix now supports running in Google Cloud Platform for existing supported databases.

Enhanced Networking Adapter (ENA) support: Delphix supports networking on AWS instances with the Elastic Network Adapter (ENA). This offers our customers enhanced networking capabilities and more economical options. Notably, this includes the AWS R4 instance types.

- **Masking NFS/CIFS mount**: Our customers increasingly are masking files alongside their databases. The masking engine has classically supported this via FTP/SFTP but now to make things easier Delphix has introduced the ability to directly mount and mask a file system - over NFS and CIFS. This should dramatically simplify the process of file masking.
- **Oracle quality**: Continued focus on Oracle quality and have introduced several quality improvements with our 6.0 release.
- **Masking API updates**: 6.0 introduces a significant number of new endpoints, including mainframe control, as well as updates for existing endpoints. This release also introduces versioning for the masking API, allowing our customers to upgrade without risk of breaking their integrations.
- **AdoptOpenJDK 8 for the Delphix toolkit**: Delphix has changed the Java Development Kit (JDK) that is included with the toolkit, and is sent to all Delphix connected environments. Customers who require using Oracle Java may continue to do so with the feature to provide their own Java, which shipped in 5.3.5.
- **Removed instance check**: When running in AWS or Azure, the product will no longer raise a fault when it detects that it is running on an unsupported instance. This enables Delphix to certify previously released software on new instances without having to modify the software.
  The product will still detect what instance it is running on and include this information in the user interface and phone home bundles. We will also continue to publish a matrix of supported instances for Azure, AWS, and GCP in the product documentation. Delphix provides no guarantee of performance or support for unsupported instance types.
- **Upgrade process**: The upgrade to 6.0 will be an in-place upgrade like other Delphix releases, there are a few changes that will improve the process overall for 6.0:
    - We will require an upgrade to an interim release first (either 5.3.6). This can be done at the same time as the customer upgrades to 6.0 or in the months prior.

- We will be introducing new upgrade checks to ensure that customers are not using features that have been removed. For a list of removed features see Deprecated and End-of-Life Features[59].
    - We will provide an upgrade image specific for each platform we support with Virtualization (VMWare, AWS, Azure, GCP). This will allow us to be more precise in customizing the images for each.

3.2 Fixed issues

Multiple Device Removal on engines version 6.0.12.0 to 6.0.17.2
For Data Engine versions between 6.0.12.0 and 6.0.17.2, there is a critical kernel module change that impacts the Multiple Device Removal feature. This change necessitates a reboot to activate the new module. Consequently, if your engine is still on versions within this range and has undergone a deferred reboot upgrade, it will not support Multiple Device Removal without a system reboot. Post-reboot, the feature will function normally.
In versions 7.0.0.0 and above, Multiple Device Removal operates as intended regardless of the upgrade method used, whether a deferred reboot upgrade or a full upgrade.

Block-to-object storage migration
For kernel versions below 15.0.0.0, in-place block-to-object storage migration is not possible. To utilize this feature, you must first complete a full upgrade to version 15.0.0.0 or higher.

3.2.1 Release 23.0.0.0 Changes

3.2.2 Fixes that take effect after upgrading and rebooting

Bug number	Description
DLPX-91010	Fixed an issue where large numbers of masked versions for SDD could occasionally cause a crash during replication.

59 https://delphixdocs.atlassian.net/wiki/pages/createpage.action?fromPageId=149533690&linkCreation=true&spaceKey=CD&title=%2823.0.0.0%29+Deprecated+and+end-of-life+features

3.2.3 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-62559	Added logic so that the notification channel name for a particular tab will persist for the lifetime of that tab.
DLPX-77122	Fixed an issue where the Continuous Data Engine may fail to cleanup timeflows and/or snapshots that are expired or no longer needed. A failure to cleanup after repeated attempts will cause the Delphix Management Service to become unavailable.
DLPX-85091	Fixed an issue which would allow customers to add devices that have different physical block sizes.
DLPX-89424	Added retry attempts for fetching iSCSI session during mounting.
DLPX-89638	Show better error message after Enable operation on a vPDB fails when the linked CDB is not up.
DLPX-89898	Fixed an issue to provide clear action to the user when the hostname or the host IQN is changed without following the recommended steps.
DLPX-89933	Fixed an issue that caused partial networking configuration updates, resulting in address removal when a static address is not passed.
DLPX-90199	Fixed an issue which prevented deletion of a vCDB having a vPDB that did not have a current timeflow.
DLPX-90373	Fixed an issue where if a node is added/deleted for a vCDB in a RAC cluster, enabling one of its vPDBs modified the `wallet_root` location of the vCDB.
DLPX-90612	Fixed an issue where aged cloud engines could experience a `zfs_object_agent` panic loop.
DLPX-90658	Corrected the logic for the calculation of iSCSI LUNs for the target server while linking and provisioning.

Bug number	Description
DLPX-90697	Fixed an issue where a job warning would be thrown when MSSQL database files contain trailing and leading ASCII whitespaces or a trailing ASCII period after the restore operation is performed.
DLPX-90748	Fixed an issue where enabling an Oracle vPDB failed with “Instance already in use on the target host” if there were two or more vCDBs with the same prefix on the target host.
DLPX-90785	Fixed an issue where the SnapSync operation for Oracle staging push dSources would fail with `exception.oracle.dsource.stagingdb.datafile.checkpoint_scn.not.matching` if the production database had read-only tablespaces.
DLPX-90788	Adjusted the NTP configuration to allow `ntpd` to change the system time when getting corrections from pooled servers.
DLPX-90805	The timeflow creator for staging push dSources has been updated, which will default ZFS filesystems to 8k block size.

3.2.4 Release 22.0.0.0 Changes

3.2.5 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-72665	Fixed an issue where email alerts would continue being sent to disabled Admin users.
DLPX-74650	Fixed "Internal Server Error" in all Oracle vPDB operations after a canceled refresh operation.
DLPX-75466	Fixed an issue where provisioning an AppData database with default values, specifically when JSON is empty, throws an error from CLI.

Bug number	Description
DLPX-75493	Fixed `exception.executor.object.missing` in Oracle vPDB enable/refresh operations after a failed provisioning operation.
DLPX-78731	The `StorageTest` `testRegion` has been capped at 1TB.
DLPX-80913	Fixed a timing error where the device initialization monitor job would probe at the same time as initialization completes, which falsely detected and reported an error.
DLPX-83760	When a device removal is attempted with insufficient free space available, the resulting error message now clearly indicates the problem and the amount of free space needed.
DLPX-83856	Better actions are now provided when an environment refresh fails because of antivirus.
DLPX-85644	Fixed an issue which caused an Oracle vPDB delete job to hang.
DLPX-85995	Fixed an issue where a large hook script size may cause LogSync to fail with an `OutOfMemory` exception, causing the Delphix management service to crash.
DLPX-87824	For Oracle vPDBs, the TDE software-based keystore parameters, `parentTdeKeystorePath` and `targetVcdbKeystorePath`, should not be pointing to the same location.
DLPX-87934	Fixed an issue where a ZFS filesystem containing a clone of Oracle archive logs from the parent snapshot was not deleted after the failed provision/refresh of a vPDB, in a linked CDB or a VDB.
DLPX-88916	Added secure CSP header `Object-Src`.
DLPX-89268	Fixed an issue that was preventing Delphix from tracing file deletions over NFS.
DLPX-89324	Worker thread for plugins can now be controlled using a tunable, `APPDATA_INTERVAL_BETWEEN_VALIDATED_SYNC`.

Bug number	Description
DLPX-89562	Fixed an issue where Self-Service refresh could still get an `IllegalStateException` while running an Oracle vPDB reprovision, because of retries for reprovision.
DLPX-89691	For Linux hosts, `bash` is now expected to be available in the path as a prerequisite for the `toolkit` to run. This fixes an issue where disabled `vsyscall` kernel parameter leads to failure of multiple host related workflows including VDB operations.
DLPX-89696	Fixed an issue where the start of an Oracle RAC virtual PDB, in a virtual CDB, fails with an error `exception.oracle.targetscripts.pdb.open`, if the virtual PDB is already running.
DLPX-89930	Fixed an issue where an Oracle vPDB is not disabled first during upgrade, leading to an exception while disabling the vCDB.
DLPX-90181	Added defensive checks to report an error if PDB tempfile directories are created with the wrong permissions, due to an Oracle bug.
DLPX-90207	Converted the data type to float for handling larger network throughput values.
DLPX-90208	Addressed CVE-2024-22243 from `spring-core`.
DLPX-90407	Increased the default timeout for `datapatch` to 12 hours to account for longer time taken by the Oracle `datapatch` command.
DLPX-90371	Delphix will now provide a clear error message and suggested action when the Oracle `datapatch` command is terminated by Delphix due to a timeout.

3.2.6 Release 21.0.0.1 Changes

3.2.7 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-90533	Fix an issue where a customer may see recurring and somewhat persistent errors on the SETUP application.

3.2.8 Release 21.0.0.0 Changes

3.2.9 Fixes that take effect after upgrading and rebooting

Bug number	Description
DLPX-89825	Fixed an issue where CPU usage increases as the system has been running longer since a reboot, causing poor performance.

3.2.10 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-42204	Improved the error message shown when the Continuous Data Engine does not have enough storage to ingest an Oracle database.
DLPX-63594	While navigating back to the Dataset page from another page, the user's previous selected container will be loaded, instead of always loading the first one.
DLPX-74693	Enhanced the diagnosability of Commvault Restore failures as part of the validated sync job.

Bug number	Description
DLPX-76121	Incorporated a timeout mechanism into the script to streamline the process of adding iSCSI Login on the target, enhancing script reliability across its usage.
DLPX-84540 DLPX-89854	Fixed an issue to better highlight the Change CDC flag in the UI while provisioning a VDB.
DLPX-85313	Fixed an issue that causes the management service to restart when an Oracle VDB has a large number of archived logs.
DLPX-86990	Provided a more meaningful error message and action to the end user in case zvol creation times out.
DLPX-88265	Timeflow data is now loaded in batches only for the timeflow dates that are opened by the user, thus, improving load time of the Timeflow page.
DLPX-88831	Fixed an issue that would prohibit the enable/start of an Oracle RAC or multi-tenant virtual source, after a failed hook operation during its provisioning or refresh.
DLPX-89510	Fixed CVE-2023-48795
DLPX-89610	The Region field is no longer required when configuring on-premise Elastic Data engines via the CLI.
DLPX-89820	Fixed an issue where TDE-enabled vPDB provisioning using a dSource created in releases prior to 19.0.0.0 failed after upgrade.
DLPX-89894	Fixed an issue where the object storage API could result in 500 internal error response.
DLPX-90010	Fixed an issue where the engine would incorrectly assume a dSource is TDE-enabled when sqlnet.ora, in an Oracle database, contains the `WALLET_LOCATION` parameter to configure an external password wallet.

3.2.11 Release 20.0.0.0 Changes

3.2.12 Fixes that take effect after upgrading and rebooting

Bug number	Description
DLPX-89834	Fixed an issue that can cause excessive CPU usage and poor system performance after a system has been running for an extended period of time.

3.2.13 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-64990	When a user selects a group for replication but subsequently deselects specific objects, Continuous Data automatically deselects related parent groups. This ensures only the specifically chosen objects are replicated.
DLPX-69276	Verbatim regarding different “apply upgrade” types have been updated for better user experience.
DLPX-72414	Modified the action item UI to accommodate owner info.
DLPX-75574	Added the ability to apply changes to DNS Configuration via the UI.
DLPX-81656	Fixed an "unknown error" in Oracle environment discovery when the `LC_ALL` environment variable is set in the associated host's `bashrc` file.
DLPX-82055	Fixed an error when enabling/starting a Oracle RAC provisioned vPDB in a linked CDB if one of the cluster nodes is down.
DLPX-84710	Improved handling of missing networking devices after instance type migration.

Bug number	Description
DLPX-86069	Fixed an issue where after critical engine storage utilization issue is resolved, an attempt is made to incorrectly resume replicated dSources.
DLPX-86611	Fixed an issue to prevent redundant faults on the Continuous Data Engine caused by failed restores of source database backups on Commvault. This avoids the recreation of the staging database on the dSource, thus saving significant infrastructure resources.
DLPX-88039	Added the actual error message in output and updated the message to be more generic.
DLPX-88414	Fixed an issue where an Oracle vCDB was getting re-registered with the default listener (port 1521) after a host environment refresh.
DLPX-88679	Fixed an internal error in running the `purgeLog` API for a linked CDB after all PDBs are migrated out from it.
DLPX-89115	Fixed an issue where a provisioning job warning is posted because of "ORA-25153: Temporary Tablespace is Empty".
DLPX-89386	Support bundles now contain diagnosable session information for iSCSI.
DLPX-89436	Fixed an issue where a new value for the tunable `ORACLE_MT_PROVISION_SORT_AREA_SIZE` did not come into effect during provision or refresh of Oracle vPDBs.
DLPX-89538	Fixed an issue where storage space recovery did not happen right after a failover commit operation.
DLPX-89597	Addressed HF-1283 & HF-1328.
DLPX-89817	Fixed an issue where TDE software keystore based provisioning fails after upgrading to Continuous Data Engine 19.0.0.0 release.

3.2.14 Release 19.0.0.0 Changes

3.2.15 Fixes that take effect after upgrading and rebooting

Bug number	Description
DLPX-88884	Fixed kernel deadlock issue that may occur during block-to-object storage migration.

3.2.16 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-22606	Added better messaging in the event of Windows Host reaching the Maximum LUN Limit.
DLPX-72967	Fixed an issue where if CRS Home is on a different version than Oracle Home, the export job tried to run the SQLPlus utility from the CRS HOME instead of Oracle Home, leading to job failure.
DLPX-74892	Clarified a user error that did not provide exact details when missing `gtar` causes Solaris Environment discovery failure.
DLPX-77378	Labels for `autoVDBRestartValue` throughout in the app has been made uniform to Yes/No.
DLPX-77621	Added relevant error message when the LDAP server name is not resolved.
DLPX-82158	Fixed an issue when discovering an Oracle RAC cluster, instance numbers starting with 1 were automatically assigned, resulting in SnapSync failures.
DLPX-82754	Fixed wrong environment name shown in summary and progress page when running Network Performance tool throughput test and DSP test.

Bug number	Description
DLPX-84115	Fixed an issue where Java core analysis, when generating support bundles, can sometimes get stuck without timing out.
DLPX-86028	Fixed inconsistent behavior of capacity historical data calculation when an ownerless container is encountered.
DLPX-87822	Fixed an issue to provide better action in the event of Windows Environment creation or refresh failure caused by invalid characters in the database name.
DLPX-87965	Fixed internal error for Data Vault Replication spec when spec schedule's Cron expression has a single white space at the end of expression.
DLPX-88322	Fixed an issue where if a datafile is added when a SnapSync job is running, provision or refresh to the snapshot generated by the SnapSync job will fail.
DLPX-88753	Fixed an issue where refreshing or provisioning a RAC vPDB may fail on Oracle 12.x with the error, “failed to apply logs in database recovery” if a PDB is open in some RAC nodes and mounted in others. This could also happen for a non-RAC database in some situations, but has been resolved.
DLPX-88925	Upgrades during block to object storage migration are now allowed.
DLPX-89004	Fixed an issue that could occur when converting instance types on a Cloud Engine.
DLPX-89068	Fixed an issue where a RAC vCDB is provisioned with only 2 online log groups with 4MB each, per thread. This is the minimum allowed size for online logs. Now, the user can specify the desired number and size of log groups for a RAC or non-RAC vCDB.
DLPX-89300	Fixed an issue related to MSSQL full backup hangs by logging the `robocopy` command output to the log file, at the Delphix connector logs folder, during sync operation.
DLPX-89392	Fixed a race condition during an 'Apply now' upgrade that may cause the boot loader to become corrupted.

3.2.17 Release 18.0.0.1 Changes

3.2.18 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-89392	Removed race condition during an "Apply now" upgrade where the engine's boot loader may get corrupted.

3.2.19 Release 18.0.0.0 Changes

3.2.20 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-69788	Fixed an issue where the audit log did not tell the reason of login failure when the same username is defined under DOMAIN & SYSTEM.
DLPX-71024	Fixes an issue where device links were not being generated correctly.
DLPX-78465	Fixed an issue where the Delphix management stack could fail to start when finalizing a deferred upgrade that was initiated while the stack was down.
DLPX-79498	Fixed an error message when the DelphixConnector service is already installed with the previous version.
DLPX-80623	Fixed an issue where starting an Oracle vPDB in a linked CDB failed after one of the RAC nodes is rebooted.
DLPX-83344	Fixed an issue where the Delphix Fluentd service would silently accept invalid Splunk configurations without testing the connection first.

Bug number	Description
DLPX-84824	Fixed a failure in Oracle dSource snapshot operation due to the `traceroute` command timing out.
DLPX-85466	Fixed an issue where an event type that does not match the defined regex pattern lead to failure in creating an alert profile.
DLPX-86212	Fixed an issue where an invalid hec_host value can be saved to the Splunk configuration without actionable feedback being provided.
DLPX-87844	Fixed an issue where the SSH connectivity test did not detect expired passwords.
DLPX-88114	Fixed an issue where a replication job on a target engine could cause an upgrade to fail.

3.2.21 Release 17.0.0.0 Changes

3.2.22 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-50012	Fixed missing timeflow error when enabling VDBs post replication failover.
DLPX-57673	Fixed an issue where Support bundle generation can be time consuming if the engine has a large number of core files to process.
DLPX-78962	Updated the suggested error action to offer improved guidance to users in the event of a SQL server hook script failure caused by disabled or uninstalled Windows PowerShell 2.0.
DLPX-81559	Fixed an issue after upgrading an Oracle VDB from 12c to 19c where the VDB refresh fails with, "failed to mount database instance" due to ORA-01130 or ORA-00201 errors.

Bug number	Description
DLPX-82143	Fixed an issue causing unneeded CPU utilization that occurred due to excessive reads of Fluentd configuration.
DLPX-84565	Fixed an issue with the `telegraf.service` restarting in a loop after an upgrade.
DLPX-85494	Fixed a deadlock between a replication lock and the metadata store connection pool.
DLPX-86966	Fixed an issue that was leaving stale metadata after the deletion of `tlog` backup snapshots in the Staging Push environment.
DLPX-87401	MSSQL File mapping for V2P operations now supports regex with `?` (i.e. `?i`) for case insensitive search.
DLPX-88040	Clarifications have been added to provide better insight to the reason for disabling some datasets during SDD Replication.
DLPX-88056	Prevented database file deletion for new VDBs when the old VDB is dropped after replication failover.
DLPX-88113	Added a banner to `move-to-asm.sh` warning users about deprecation and impending EOL in Continuous Data 23.0.0.0 (May 2024)
DLPX-88143	Increase disk removal threshold to 90% used space from the previous limit of 80%.
DLPX-88351	Fixed an issue where the support bundle redactor runs out of memory when too many alerts are loaded at once into memory, which resulted in all metadata being missed from support bundles.
DLPX-88480	Fixed an issue when provisioning, refreshing, or rewinding an Oracle vPDB that has a different minor version than the target Oracle Home, and datapatch is enabled, it may fail with, "ORA-01652: unable to extend temp segment by 128 in tablespace TEMP".
DLPX-88488	Fixed an issue where not enough samples are collected for performance analysis.

Bug number	Description
DLPX-88518	Fixed an issue where the `undo_tablespace` parameter was considered restricted and not allowed to be changed by the user after an Oracle VDB is provisioned.

3.2.23 Release 16.0.0.0 Changes

3.2.24 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-87910	Fixed a delay that may occur when an Elastic Data Engine has had devices removed and added, and is then upgraded from 11.0.0.0 or earlier to 12.0.0.0 - 15.0.0.0 (inclusive).

3.2.25 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-78318	Fixed an issue where the first SnapSync of an Oracle DoubleSync snapshot incorrectly showed missing logs.
DLPX-84253	Fixed an issue preventing users from being able to delete Delphix-generated CA certificates if no dependencies exist.
DLPX-87163	Fixed an issue where the Environment refresh job failed with "The object OraclePDBConfig does not exist on the system".
DLPX-87320	Fixed an issue where exporting a VDB or vPDB to ASM would fail with ORA-32771 when the database has a temporary bigfile tablespace.

Bug number	Description
DLPX-87516	SQL server database files will be dropped from the staging database DATA\db directory when the source database drops the files. All files and folders not part of the MSSQL database will be removed from the DATA\db folder.
DLPX-87702	Fixed an issue where provisioning an Oracle VDB/vPDB fails with "ORA-03214: File Size specified is smaller than minimum required" when temp files have a non-default local uniform size.
DLPX-87885	Fix provided to raise a fault for replicated Python Plugins with an unsupported vSDK build API version and mark them as inactive.
DLPX-87908	Fixed an issue where fixed_vdb_parameters (uncustomizable parameters) are incorrectly updated and resulted in the inability to perform VDB snapshot operations.
DLPX-88250	Fixed an issue where a missing NTP configuration could prevent upgrade image verification from succeeding.

3.2.26 Release 15.0.0.0 Changes

3.2.27 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-86450	Fixes a hang in the I/O subsystem that can cause the Delphix Engine to become unresponsive.
DLPX-86595	Increased the limit of masked timeflows per snapshot from 32 to 16000 for SDD.

3.2.28 Fixes that take effect immediately after upgrading

Bug Number	Description
DLPX-72342	Environment discovery will not fail if Ag Listeners are not reachable from the Engine. If a listener is unreachable, it will be reported during linking or sync operation.
DLPX-79973	Fixed an issue where the Splunk HEC Token could leak in Support Bundle through the fluent.conf file.
DLPX-80093	Removed fluent.conf file from Support Bundles.
DLPX-83609	The 'Type' column filter in the admin UI Jobs page now contains the DB_ROLLBACK job type.
DLPX-85082	Browser Tab title changed from Delphix System to "engineName - Delphix Setup".
DLPX-86892	Fixed an issue where the vCDB TDE refresh failed if "Target vCDB TDE KeyStore Location" is set with "tde" trailing path.
DLPX-86894	Fixed an issue where after a canceled refresh operation, VDB refresh or delete may fail due to error code `exception.oracle.vdb.no.virtual.datafiles.found`. If a workaround is still needed, disable the VDB and try the operation again.
DLPX-86964	Fixed an issue where if a `SOURCE_DISABLE` job of a TDE-enabled VPDB is interrupted after it's unplugged from the CDB, but before it's dropped from the CDB, any subsequent disable jobs against the vPDB would fail.
DLPX-86965	Escaped constructs in Regex Patterns (for example, \Q, \E) are now allowed for MSSQL V2P File Mappings.
DLPX-87224	Fixed an issue where the Privilege section of the User Management wizard was momentarily hanging for Engines with multiple datasets or groups.
DLPX-87454	Updated MSSQL JDBC driver to version 7.4.1.0. This fixes the issue that created multiple faults with `RejectedExecutionException`.

Bug Number	Description
DLPX-87522	Fixed an issue where the `move-to-asm.sh` script could fail with, "move-to-asm.sh: Failed to move online log files", while attempting to move online log files to ASM.
DLPX-87547	Analytics timestamps are now accurate when the collection interval is customized via the CLI.
DLPX-87623	Fixed an issue related to provisioning failures from a snapshot taken after opening the staging database with the `resetlogs` option.
DLPX-87624	Fixed an issue related to provisioning failures from a snapshot taken after opening the staging database with the `resetlogs` option.
DLPX-87649	Fixed an issue preventing retries of upgrades after quiesce failures.
DLPX-87650	Fixed an issue regarding upgrade verification error version 13 (stack startup verification failed) due to `StreamReadConstraints` limit when deserializing event bundles.

3.2.29 Release 14.0.0.0 Changes

3.2.30 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-86764, DLPX-87360	Added a mechanism to automatically restart internal service if I/O to object storage is not making progress.
DLPX-86962	Eliminated deadlock scenario introduced by upstream changes to ZFS.

3.2.31 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-67911	Fixed issues related to having stale mounts on staging and target host.
DLPX-71914	Capacity table UI for received replica now shows aggregated size for virtual and source datasets.
DLPX-79098	Seconds is also shown as part of the snapshot point in time on the Summary of Refresh Wizard.
DLPX-83250	Support for SSL for Fluentd/Insight data consumers.
DLPX-84003	Fixed an issue where provisioning an Oracle vPDB into a new vCDB when the source PDB$SEED has a TEMP tablespace, where the initial allocation plus file headers is greater than 100M (for 8k blocksize) fails with “ORA-03214: File Size specified is smaller than minimum required”.
DLPX-85770	Fixed a failure reported by a start operation on a self-service container or an enable operation for a Oracle virtual source that is already enabled.
DLPX-86195	Fix ensures the `is_encrypted` flag is inherited during provisioning via snapshot. Activate with the tunable `CREATE.ENCRYPTED.SEED.DB` set to true (the default is false).
DLPX-86620	Instead of raising faults for `java.util.concurrent.RejectedExecutionException`, changes have been made to log the issue in debug logs and fail silently, to reduce noise.
DLPX-86644	Fixed an issue where a TDE-enabled vPDB provision or refresh operation fails with a misleading error if the parent CDB or parent PDB’s primary key is missing in the parent TDE keystore on the target host.
DLPX-86685	Fixed an issue causing VDB operations to fail after upgrade verification is run.
DLPX-86805	Updated Spring Framework dependency version to 5.3.28.

Bug number	Description
DLPX-86854	Fixed an issue that caused post-upgrade cleanup to fail.
DLPX-86906	Fixed an issue to prevent linking an Oracle PDB that is added manually in a virtual CDB.
DLPX-87006	Fixed an issue in refreshing self-service containers with ordered sources, post upgrade to 12.0.0.0.
DLPX-87038	Fixed a size discrepancy in the OVA to fix certain deployment issues.
DLPX-87219	Post-refresh Snapsync of an Oracle VDB/vPDB fails with `exception.oracle.vdb.uncustomizable.parameters.changed` after the PDB is detached/attached following a standby switchover.

3.2.32 Release 13.0.0.0 Changes

3.2.33 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-61403	If an Oracle CDB/vCDB has a quota policy configured, Delphix will ensure that PDB/vPDBs are disabled/enabled properly when quota policy disables/enables the CDB/vCDB.
DLPX-71066	Fixed an issue to provide better action when a plugin upload fails due to the existence of a badly provisioned VDB.
DLPX-74553	Fixed an issue where the `Target Database Exists` error may be thrown during VDB provisioning, if a VDB was present during the last Environment Refresh.

Bug number	Description
DLPX-78987	A RAC node can only be disabled if no virtual sources are in ‘running’ status on the node.
DLPX-81268	Fixed an issue where executing `expect` commands logged sensitive information.
DLPX-84624	Removed build date from the Upgrade Images page UI.
DLPX-85298	Created new Oracle faults for quota policy violation with more details for error and action text.
DLPX-85747	AWS Cloud engines now support IMDSv2.
DLPX-86109	Fixed an issue where Oracle VDB unquiesce/enable may fail after a failed quiesce/disable on 10.0.0.X or 11.0.0.X.
DLPX-86163	Fixed an issue where if TNS_ADMIN for an Oracle vCDB was wrongly set during provision or refresh, it would cause older clients such as SQL*Plus 9.2.0.8 and Oracle Application Server 10.1.0.3 to fail with `ORA-28040: No matching authentication protocol`.
DLPX-86272	Fixed an issue for secondary nodes with different MSSQL instance owners than the primary node, where it now works with Delphix Managed backups.
DLPX-86344	Fixed an issue related to failed Delphix engine upgrades due to plugin operations that were failing.
DLPX-86496	Fixed an issue in the interaction between replication and replica retention policy.
DLPX-86497	Fixed the functionality to relink/attach SQL Server Availability Source Group database using the Link dSource dialog.
DLPX-86563	Fixed an issue where if any detached dSources were present then the performance history page would hang.

Bug number	Description
DLPX-86787	Fixed the GUI un-responsiveness of the Datasets page, after viewing the replication profiles.
DLPX-86842	Fixed a failure in the `verify upgrade` job while upgrading to 12.0.0.0.

3.2.34 Release 12.0.0.0 Changes

3.2.35 Security fixes

Bug number	Bug introduced in	Description	Security bulletin
DLPX-86329	6.0.13.0	Sysadmin can execute shell commands on the underlying Operating System.	TB109

3.2.36 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-86177	Fixed a bug where Accelerated Networking was broken due to missing drivers.

3.2.37 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-44117	Fixed an issue where a null pointer exception is encountered during parsing of an Oracle archive log not present in the Delphix file system.
DLPX-59966	Fixed an issue where the Snapsync of an Oracle vCDB may become unresponsive if the Delphix archive log destination is removed or changed, pointing to an invalid location.
DLPX-72422	Reduced VDB downtime on upgrade.
DLPX-82702	Improved error behavior when attempting to expand storage devices with underlying problems.
DLPX-83430	Fixed an issue where the initial configuration of Syslog breaks most of the pre-existing appenders.
DLPX-84611	Fixed an issue allowing environment users that are NOT in the primary group of the primary OS user to monitor the builtin files VDBs.
DLPX-85578	Replaced the Win32_Volume class output with mountvol output to fetch volumeId for Delphix ISCSI mount points.
DLPX-85647	Added a filter to fetch only the IPv4 address for AG cluster nodes.
DLPX-85748	Now displaying the state of enabled services in the Delphix Startup Screen.
DLPX-85857	Fixed an issue where after a Delphix Continuous Data engine upgrade to 10.0.0.0, faults related to listener registration are thrown during source enable.
DLPX-85883	Fixed an issue where a vPDB provision fails when invoked as a non-instance environment user, with standby source and datafile added in the recovery stream.

Bug number	Description
DLPX-85925	Added a new condition in subquery to uniquely identify filegroup for commvault backup files.
DLPX-86050	Fixed an issue where TDE diagnostics for directory permission fails if the TDE artifact directory has a space in the path or name.
DLPX-86102	Fixed an issue where ASE instance discovery is failing during environment add/refresh when ASE instance is running in multi-process mode with multiple ASE engine.
DLPX-86201	Fixed an issue where users are sometimes unable to generate a complete support bundle.
DLPX-86240	Internal services will now restart during a deferred upgrade, enabling the delivery of fixes without requiring a reboot.

3.2.38 Release 11.0.0.0 changes

3.2.39 Security fixes

Bug number	Bug introduced in	Description	Security bulletin
DLPX-85604, DLPX-85606, DLPX-85608	Product Inception	Several input fields in the Self-Service feature are vulnerable to cross site scripting (XSS).	TB104

3.2.40 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-82702	Improved error behavior when attempting to expand storage devices with underlying problems.
DLPX-85526	Eliminate the possibility that a spurious udev event during drive expansion will cause the pool to suspend.

3.2.41 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-86068	Fixed an issue from release 10.0.0.0 where users could not replace certificate chains when using root certificate authorities from Java's default list.
DLPX-85915	Fixed an issue where users were unable to add an environment when part of the host name contained all numeric elements.
DLPX-85752	The icons for the export functionality have been updated to secondary buttons (along with the icon) for clarity.
DLPX-85732	Fixed an issue where provisioning a TDE-enabled vPDB into a new vCDB with "Mask this vPDB" check fails with "ORA-46636: cannot add second keystore to the target keystore".
DLPX-85718	Starting from 7.0.0.0, the URL for "Help Documentation" has beed updated.

Bug number	Description
DLPX-85621	Fixed a UI message saying, "management stack needs restart after replacing certificate', when it did not.
DLPX-85601	Removed weak CBC ciphers from default list.
DLPX-85493	MSSQL: Linking and AttachSource operations will not require unnecessary permissions of source user on the staging host and staging database.
DLPX-85445	Fixed the software version main window issue so that it now updates if a previous version is selected.
DLPX-85205	Improved the error message for environment authentication failure by including username.
DLPX-84877	Fixed an issue where adding HPUX environment failed with an Unsatisfied link error.
DLPX-84792	Added new tunable, MSSQL.ROLLBACK_HANDLING_AFTER_FAILED_RESTORE. If this tunable is set to true, the Delphix engine will perform a zfs rollback on DATA LUN right after a restore db failure.
DLPX-84738	Fixed an issue where the environment user public key is not displayed in the CLI/API.
DLPX-84709	Fixed an issue to disallow CLI or API transactions for an Environment user with a null publickey or privatekey.
DLPX-84610	Improved error message to show correct host name and user name in the case of builtin files virtual source status failure.

Bug number	Description
DLPX-83927	Fixed an issue so that Fluentd/Splunk integration will accept protocol parameter in any case.
DLPX-83897	Fixed an issue with changing port number while configuring Kerberos.
DLPX-83360	Fixed an issue where new authorization would not be granted unless the old one gets removed successfully.
DLPX-81587	DeletionDependency object in API documentation has been updated to reflect that size is Valid for TimeFlowSnapshot and HeldSpace objects.
DLPX-82895	Fixed an upgrade verify issue when fs.inotify.max_user_watches exceeds the limit of 16384.
DLPX-80325	Fixed an internal error that appeared in the Oracle dSource upgrade operation while resuming logsync for the dSource.
DLPX-80221	Resolved issues related to incorrect mount path by normalizing the path.
DLPX-79492	Improved the selection of faults and the toggle behavior of "Hide Resolved/Ignored" from the "Verification Result's Faults List" on the Version Upgrade page.
DLPX-78839	Fixed an issue where the NTP server would not validate when configured.
DLPX-77826	Added timeout functionality to JDBC queries made through DSP. Leaked connections due to network issues will be auto-closed after the timeout period. Timeout can be set using the tunable dsp.jdbc.queryTimeout.

Bug number	Description
DLPX-77386	Fixed a misleading prompted message about host being unavailable.
DLPX-76762	Performance improvement: Enabled unbuffered copy and multithreading options with the robocpy command for vFile operations.
DLPX-75448	Fixed an internal error that appeared when the purgeLogs API was used for timeflows with no snapshots.
DLPX-74905	Fixed an issue where environment refresh and VDB provision/refresh operation will fail with an "internal error" if the target host's filesystem becomes 100% full.
DLPX-73058	Fixed an issue where if an environment is disabled, VDB operations like start, stop, rollback, refresh, undo, disable would not be allowed.
DLPX-63076	Fixed upgrade check result messages to make them consistent with the failure's severity level.
DLPX-63425	Delphix will now throw a job warning and raise a critical fault in the Delphix Admin UI if a failure is encountered while backing up the archived logs and LogSync is disabled. In this case, Delphix will invoke an RMAN script to delete the backups.
DLPX-67347	Fixed an exception encounter (paired with an indefinite loading sign in the UI) when an environment addition task was cancelled in between. The environment got deleted in the next attempt but the toolkit folder did not. Added DUE to show the warning for toolkit deletion failure.
DLPX-84981	Improved the error message for null or whitespace timezone registry.
DLPX-85674	Fixed the issue where snapshots were created with the wrong (older) timeflow that led to a failure to enable the linked sources with internal error.

Bug number	Description
DLPX-85102	Reduced the size of the temporary directory unique name to be backward compatible with the name `_delphix`.

3.2.42 Release 10.0.0.1 changes

3.2.43 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-85760	Addressed an issue that in rare scenarios could lead to data loss during upgrade or reboot of Cloud Engines (Upgrade to a version with the fix is safe).

3.2.44 Release 10.0.0.0 changes

3.2.45 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-85019 (only applicable to initial deployments)	Fixed a bug that causes the management service to fail startup upon initial VM deployment in the cloud, if the DHCP domain name of the VM ends with a period (.).
DLPX-85109 (only applicable to initial deployments)	Fixed a bug that causes the management service to fail startup upon initial VM deployment in the cloud, if the DHCP domain name of the VM contains bad characters (does not belong to [a-zA-Z0-9.-]).
DLPX-84985	Fixed a deadlock which caused iSCSI connections to fail on Windows hosts.

Bug number	Description
DLPX-84995	Fixed an issue where NFS could cause excessive CPU usage when open files from NFSv4 mounts exceeded 16,384.
DLPX-62215	Fixed a nuance in the CAPACITY_RECLAMATION job.

3.2.46 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-51276	Configuration discovery script fixed to handle Windows host timezone more smoothly.
DLPX-53420	Improved error message for failure in deleting CDB sources with no associated PDBs, but with timeflow dependencies from migrated vPDBs.
DLPX-59162	Fixed failure in connecting to a Compliance engine while running a compliance job as a post-provision hook operation for RAC VDBs.
DLPX-61781	Fixed authentication failures past the first node while adding Kerberized RAC environments.
DLPX-66944	Fixed logsync failures caused by commands exceeding shell limit of 1,024 characters.
DLPX-68327	Certificates in the Truststore are considered for making connections to SMTP servers.
DLPX-74134	Fixed an issue where Oracle provisioning/refresh/rewind operations may fail with error "ORA-01169: DATAFILE number 1 not found. Must be present."
DLPX-76200	To handle unusually demanding workloads, the maximum heap size of the management application can now be increased by system administrators via the maxHeapSizeGb property of the CLI at /system.

Bug number	Description
DLPX-78673	Fixed an issue related to a message being ignored during the mount check when provisioning virtual databases. The message is not thrown because the provision completes successfully, but it could still be seen in the logs.
DLPX-80925	Fixed an issue of a raised fault due to inconsistent snapshot creation time, due to manual change of Delphix engine time.
DLPX-81874	Fixed an issue that occurred during the discovery of a cluster when adding or refreshing a Windows FCI environment with multiple NICs. The issue involved the use of the IP address of hosts that were already added.
DLPX-81982	The Delphix Fluentd logs do not automatically rotate when exceeding their expected maximum size of 100MB.
DLPX-82169	Fixed an issue where provisioning an Oracle TDE-enabled vPDB into a linked CDB with a different patch level than the source CDB will fail leaving the vPDB in a broken state.
DLPX-84108	The validation of RSS on the target host will now only occur when the customer uses an IP address for an environment. This update removes the detection of the RSS property on interfaces that are not relevant.
DLPX-84422	You can now use a single script to copy all transaction log backup files, regardless of whether they were created using native backup or Litespeed backup.
DLPX-84647	The syslog pattern is now fully configurable and can be changed to conform to RFC 5424.
DLPX-84655	A false warning message that listener registration was not successful is posted when enabling a VDB or a vCDB, users can ignore this message.
DLPX-84679	Unable to add Oracle Staging push PDB if the Staging Environment has more than one repository.
DLPX-84686	To address the issue of multiple IP addresses, the validation of RSS on the target host will now only occur when the user uses an IP address for an environment. This update removes the need to find the IP address in the code.
DLPX-84898	Fixed clean-up job failure for V2ASM of a RAC VDB when one of the nodes's DB instance is shutdown.

Bug number	Description
DLPX-84929	Improved initial load time of datasets page.
DLPX-84944	The Delphix Continuous Data Engine now considers truststore certificates for more connections, including secure connections to proxy and SMTP hosts.
DLPX-85053	PowerShell scripts code enhancement.
DLPX-85095	Fixed an issue which caused the vPDB Refresh operation to fail due to vPDB unplug operation timing out in 30 mins.

3.2.47 Release 9.0.0.1 changes

Bug number	Description
DLPX-85176	Fixed a bug that can bring down VDBs using NFSv3.

3.2.48 Release 9.0.0.0 changes

3.2.49 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-85019	Fixed handling of domain names with a terminal period (.) in Terminal.
DLPX-68852	Fixed a bug in out of memory situations that could result in service interruptions.

3.2.50 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-41505	Posted a job warning message when listeners are not registered successfully as part of an Oracle VDB/vPDB provision job.
DLPX-46210	Fixes `fault.policy.log.retention.old.snapshot`, which is not raised against a PDB snapshot instead of CDB snapshots.
DLPX-57047	For MSSQL, 'Delphix Copy Only Full Backup' is now excluded when synchronizing to most recent backup files.
DLPX-58762	Provides a better action message in Oracle nologging-related faults.
DLPX-65557	Retention no longer removes the previous timeflow until the current timeflow is successful.
DLPX-68852	Fixed an issue that could cause upgrade failures.
DLPX-69790	Fixed an issue where API login with an ambiguous user name returns an incorrect action.
DLPX-70090	The correct exception is now showing when hostname is not resolvable during Environment Validation.
DLPX-79792	If there are any DB snapshots associated with the VDB during creation of a Self-Service bookmark, there will be a message to drop all associated snapshots with the VDB and try again.
DLPX-81260	The CLI error message that appears due to 'invalid primaryAddress value', has been updated to provide better insight on a resolution.
DLPX-83702	Fixed an issue where the environment discovery path was not checked to ensure it is a valid unix path.

Bug number	Description
DLPX-83905	Fixed an issue causing `HOST_REFRESH` failures when toolkit is on a shared filesystem, with an Oracle RAC configuration.
DLPX-84068	For MSSQL, warning fault raised in case there is a failure while querying instance port.
DLPX-84284	After upgrades, NFSv3 services are automatically disabled if they are no longer required.
DLPX-84339	Fault raised due to inconsistent snapshots creation time, due to manual change of Delphix Engine time.
DLPX-84351	SMTP Test now has a check in the UI to let the user know if the password was unaltered. Save now has a check to not send the password if it was unaltered.
DLPX-84528	Datasets Search bar will now work for the staging push dSource as well.
DLPX-84589	Increased size of ssh key fields to allow for larger key, for environment users.
DLPX-85081	Fixed an NFS issue that could cause NFSv3 services to be disabled even though there were active v3 mounts after an upgrade.

3.2.51 Release 8.0.0.0 changes

3.2.52 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-83859	Fixed a rare deadlock in the kernel that can cause a Delphix Engine to become unresponsive to all management operations.

3.2.53 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-43174	Fixed an issue where Oracle VDB provision job will fail if the "Open database after provision" flag is set to false.
DLPX-48083	Removed the CLI ability to set default user as it was not needed and prevents deletion.
DLPX-52904	After initial server setup, Default Snapshot/Snapsync policies are always in "US/Pacific", regardless of the timezone selected during initial Engine Setup. An admin has to manually change the time zone.
DLPX-53209	Improved error messaging during toolkit preparation (discovery) when there are host problems.
DLPX-59248	Added a fix to throw a critical alert on the last failed attempt to collect the archive log.
DLPX-59308	Added a timeout in the drop database PowerShell script.
DLPX-59689	Fixed an issue where the environment discovery could hang indefinitely when UserLAnd commands hang (i.e. lsnrctl status, ps, etc.).
DLPX-61734	Fixed an issue where the user was unable to provision an Oracle VDB when there is a dollar sign in the Tablespace name.
DLPX-62857	Permits listing users with only the "domainUserType" parameter.
DLPX-64413	Fixed an issue where provisions/refreshes of Oracle VDBs could fail when an obsolete parameter is specified in the VDB config template.
DLPX-72691	Fixed an issue where multiple snapshots were reporting the same time in the Delphix Continuous Data GUI for standby databases.

Bug number	Description
DLPX-72740	Increased users' pubkey support from 10 to 1000.
DLPX-75521	The correct error message will now be displayed in case of a provisioning failure due to waiting for the 'DB STARTUP' process timeout.
DLPX-75605	Fixed an issue where the Security Banner was not displayed after SSH login.
DLPX-77438	Added a fix to disable UNDO operation if any children for the timeflow are present.
DLPX-77792	Fixed a misleading "toolkit inaccessible" error if password expired.
DLPX-78702	Fixed an issue where users with the SYSTEM permission are able to disable any source.
DLPX-78741	For MSSQL, raised a critical fault in case the Validated Sync interval increases to 16 minutes or more.
DLPX-79136	Fixed an issue where canceling an Oracle VDB preprovision job or a vPDB (into linked CDB) provision job leaves the auxiliary database mounted on the target host.
DLPX-79919	Fixed an issue where the Delphix Continuous Data Engine allowed an environment with a duplicate "crs_database_name" to be added.
DLPX-81182	Improved the error message for when the system is out of space.
DLPX-81425	The UI will now show the running jobs up to 60 days old.
DLPX-81497	Added a fix to improve "keystore.merge.required" fault for an Oracle TDE-enabled vPDB.
DLPX-82152	Fixed an issue where navigating resolved faults was slow.

Bug number	Description
DLPX-83575	The port in the connection string for a vPDB in a Linked CDB may be shown incorrectly when it is registered to a non-default listener.
DLPX-83635	Fixed an issue where exporting the TDE encryption keys failed when the keystores root is on ACFS and referenced via a symlink.
DLPX-83788	Suggested action in "exception.ccc.authenticate.failed" no longer references Delphix Connector.
DLPX-83823	The issue where a Oracle PDB dSource snapshot is marked as not provisionable after detaching and re-attaching the PDB to a different CDB using force flag is now resolved
DLPX-83904	Fixed an issue where a provision of an Oracle vPDB may fail during recovery if there are many datafiles that are renamed under a single ORA-01244 error.
DLPX-83954	Package details are no longer revealed during HTTP redirection.
DLPX-83977	Fixed an issue causing inability to add a hook, due to "Duplicate key" error.
DLPX-84103	Persisting total "database_transaction_log_bytes_used" while taking a snapshot, for debugging purposes.
DLPX-84151	The Replication Page performance has been optimized by making network calls efficiently.
DLPX-84255	In a Single Engine Continuous Vault product, adding a new Sybase dSource to a locked group may result in the background environment monitoring process to stop working.
DLPX-84324	Fixed an issue where NFS mounts on a Solaris target could fail after a deferred upgrade.
DLPX-84495	Fixed an issue that causes upgrades from versions < 6.0.17.0 to any version between 6.0.17.0 and 7.0.0.0 on a replication target engine which may fail due to the management services being down, requiring a support call.

3.2.54 Release 7.0.0.0 changes

3.2.55 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-83579	Fixed handling of domain names with a terminal period (.) in Terminal.
DLPX-83611	Fixed a race condition between storage device link creation and the storage pool import process.
DLPX-83701	Added additional diagnosability tracepoints to the kernel unmount code.
DLPX-83697	Fixed a hang in the iSCSI initiator.
DLPX-83675	Fixed an issue that was causing stale entries to be created in a system file.
DLPX-83684	Fixed a crash in the zcache_probe command during upgrades.
DLPX-83916	Fixed cases when zcachedb was opening devices for writing when it should be read-only access.

3.2.56 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-80130	Fixed an issue which may cause the Delphix Engine to hang when doing storage migration.

Bug number	Description
DLPX-83395	Fixed an issue when storage device removal consumes an unexpected high amount of memory upon its completion.
DLPX-83697	Fixed a hang in the iSCSI initiator.

3.2.57 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-66792	Fixed a misleading "toolkit inaccessible" error on VDB Stop/Refresh if password expired.
DLPX-68053	Fixed an issue causing phone home redaction fails when JSON contains management stack errors.
DLPX-71907	Fixed an issue where C drive label gets over-written during mount script execution.
DLPX-74604	For MSSQL, provided a fix to raise warnings in case CommVault restore fails.
DLPX-75278	Fixed display data in Datasets Performance page table.
DLPX-75488	Listeners and Instances for SQL Server AG (Availability Group) will now be shown on the User Interface correctly.
DLPX-77214	Fixed an invalid version status transition when upgrade is cancelled.
DLPX-78014	Automatically disable NFSv3 services when they are no longer required.

Bug number	Description
DLPX-78506	Fixed dSource Database Authentication issue. Now the user is able to edit the credentials of the dSource DB, where authentication is configured using “Domain User with HashiCorp Vault Credentials”.
DLPX-80300	Added more error information for Environment discovery failure when ISCSI target port is blocked on network.
DLPX-80473	Fixed an issue where the vPDB unplug operation timed out during disable operation.
DLPX-80512	Fixed pagination on Snapshots tab of Storage Capacity page. Now the user will be able to see all the snapshots using the pagination control at the bottom of the page.
DLPX-81238	DFE caused by policy schedule with a non-recurring quartz cron string.
DLPX-81750	Added a more descriptive message while handling the condition where VDB is renamed outside.
DLPX-81794	For MSSQL, raised a warning fault during environment monitoring in case dlpxrunas is removed. Also, provided fix to propagate the cause of host unavailability while environment refresh.
DLPX-82288	It is now permissible to remove devices after a deferred upgrade from before 6.0.12.0 (multi-device removal support added) to 7.0.0.0 or after.
DLPX-82316	The issue with the failure of the Oracle vPDB provisioning from a standby source and datafile added in the recovery stream is now resolved.
DLPX-82882	Fixed Script: Replacing Self-signed Certificates, on the Delphix Connector.
DLPX-83002	Cron expression handling has been fixed as per quartz cron expressions.
DLPX-83153	Fixed the issue where an invalid fault "fault.oracle.db.connection.failed" was raised during VDB refresh.

Bug number	Description
DLPX-83372	Profiling script code enhanced.
DLPX-83422	Fixed an issue where Oracle move-to-asm script fails due to missing initialization parameter file init.ora in `$ORACLE_BASE_CONFIG/dbs`.
DLPX-83434	Upgrade Spring framework to 5.3.20.
DLPX-83564	Users can now perform V2P of dSource and VDB snapshot to root of a Windows drive.
DLPX-83608	Users can now V2P an MSSQL dSource or VDB snapshot to a Mounted volume on a Windows path.
DLPX-83622	Upgrade verify will fail from coming from 6.0.15.0 if a Fluentd plugin other than splunkHec is configured. Support help will be needed to upgrade.
DLPX-83706	Fixed network connectivity issues in cases where the MAC address changes.
DLPX-83783	Implemented a fix to prevent invalid transition after a deferred upgrade.
DLPX-83787	Fixed an edge case issue that could cause an engine to be rebooted after a device has been removed from the storage pool.
DLPX-83789	MountLunData.ps1 script code enhanced.
DLPX-83819	Fixed an issue causing script failure output duplication in debug logs.
DLPX-83824	Fixed an issue where datasets would intermittently going inactive with critical faults.
DLPX-83828	While editing an Environment User, the Password field will be empty by default in order to get the correct password from the end user.

Bug number	Description
DLPX-84073	Fixed an issue where existing TDE-configured vPDBs would fail to enable after upgrading the Delphix Engine from 6.0.14.0 to a later Delphix Engine version.

3.2.58 Release 6.0.17.0 changes

3.2.59 Security fixes

Bug number	Bug introduced in	Description	Security bulletin
DLPX-83043	5.2.0.0	Weak DH 1024 bit exchange key detected by security scanner for the Delphix connector.	TB099

3.2.60 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-75209	Fixed an issue that could cause an AWS EC2 Delphix Engine to be left with no network configuration following a change of instance type.
DLPX-80122	Fixed bug that caused disks with write errors in their history to have degraded performance (activated after optional reboot).
DLPX-81081	Fixed bug that would sometimes cause VMs with a lot of MSSQL VDBs to report issues on reboots (activated after optional reboot).
DLPX-81701	Fixed bug that would sometimes cause OS panics when issuing a reboot (activated after optional reboot).
DLPX-82405	Fixed an issue where the NFS server could fail when restarted (activated after optional reboot).

3.2.61 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-46621	The issue with adding concurrent environment that resulted in `exception.executor.object.exists` exception is now resolved.
DLPX-56976	The Delphix engine now returns a descriptive error message if Oracle SnapSync or environment monitor fails to connect to the database in the scenario where the value of `ORACLE_HOME` or `ORACLE_BASE` is set incorrectly in `orabasetab` file.
DLPX-57078	The issue that occurred during Oracle provisioning where if a job is canceled during recovery steps, provisioning will continue until the end of recovery before the job is actually cancelled is now resolved.
DLPX-57934	The issue where large text objects inserted into engine metadata caused errors and potentially OutOfMemory is now resolved.
DLPX-59179	Warning fault will no longer be raised when linking is done to a source host running Enterprise Edition SQL Server and a staging host is running Standard Edition SQL Server.
DLPX-60981	Fixed issue that prevented customers from reusing disks from old engines to new ones.
DLPX-63889	For SAP ASE, pre and post validated sync scripts have been removed. Any existing hooks are converted to pre-hooks list and post-hooks list, respectively.
DLPX-64169	For MSSQL, provided a fix to handle registry exceptions for source environment.
DLPX-70817	Fixed issue that prevented customers from using a disk after running the I/O report card on it.
DLPX-71064	Updated the error action for Lua and Platform plugin upgrade validations.
DLPX-73533	Added fix to prevent multiple connector operations' testing for the presence of the same SCRIPT directory.

Bug number	Description
DLPX-78787	For MSSQL, added a retryer to get the FQDN of the host.
DLPX-81043	Added additional guidance while doing Replace Certificate on the Upload Certificate step.
DLPX-81532	Standby files for MSSQL V2P operations are now created in the DATA/db folder.
DLPX-81842	Fixed an issue of Arithmetic overflow while fetching database size.
DLPX-82050	Fixed an issue in the intersection of extended replica retention, chained replication, and an entire timeflow getting deleted on the source engine.
DLPX-82145	Fixed an issue where TDE-enabled RAC vPDB provisions may fail with “ORA-28374: typed master key not found in wallet” when activating the key in the target CDB.
DLPX-82262	Disabled Apollo client DevTools to avoid vulnerabilities.
DLPX-82371	Delphix Engine improvements.
DLPX-82433	Fixed an issue where provisioning or refresh of TDE-enabled vPDBs could fail with an internal error if the `tdeKeystoresRootPath` contains any special characters.
DLPX-82514	Password field is now initially empty so that users can enter password.
DLPX-82527	Delphix Engine improvements.
DLPX-82686	Rollback during upgrade now works on Cloud Engines.

Bug number	Description
DLPX-82859	Fixed an issue where API clients that close their connections early while getting a list of snapshots could cause an internal resource leak that could make the application unresponsive if a replication job is initiated afterward.
DLPX-82908	Managed Source Data UI page now correctly displays the Type of Data for AppData sources.
DLPX-82972	Provided the capability to override default root squash behavior for mounted filesystem of unstructured files via Tunable: `LUA_VFILE_TOOLKIT_ROOT_SQUASH_DISABLE` .
DLPX-83103	Fixed an issue of environment refresh failure for Windows host after upgrade to 6.0.16.0 occurred due to iSCSI related registry parameters' monitoring.
DLPX-83105	Fixed an issue following upgrades where SSO entityID changes without user intervention.
DLPX-83148	Fixes environment refresh of an Oracle Live Source environment failure reporting an internal error.
DLPX-83149	From Windows Connector side, while handshaking, stop using the cipher suites which uses Diffie-Hellman key exchange with keys less than 2,048 bits in size.
DLPX-83206	Improved the error message that is displayed when RMAN/sqlplus connection to an Oracle virtual database fails.
DLPX-83359	Delphix Engine improvements.
DLPX-83504	Delphix Engine improvements.
DLPX-83595	The default behavior for VDBs is altered, to disable DBCC CHECKTABLE commands for datafile accessibility.

3.2.62 Release 6.0.16.0 changes

Bug number	Description
DLPX-67753	Fixed an issue causing redirect responses to reveal server type and version when HTTP redirection is enabled.
DLPX-72068	Improved the way volumes are fetched while working on mounts.
DLPX-74396	Fixed an issue that occurred when manually adding a database to an environment which has the same unique name as a database in another environment managed by the same Delphix engine. Previously, Delphix reported an incorrect environment containing the same unique name.
DLPX-80172	Updated the Self-Service refresh warning message.
DLPX-80271	changeArchivelogMode now has an associated job event.
DLPX-80387	Fixed an issue where Oracle move-to-asm script would fail while dropping temp due to tempfiles being in use and unable to be dropped after the database was started.
DLPX-81184	For S3 object store, the "Base URL" input is renamed to "endpoint". The "region" input is now a dropdown for the user to select from a list of standard regions. The endpoint corresponding to that region will now be auto-populated.
DLPX-81692	Fixed an issue where Direct NFS was not being detected for Oracle 21.
DLPX-81996	Fixed an issue where an environment refresh after upgrade did not remove outdated/obsoleted toolkit components in 6.0.014.0 and 6.0.15.0.

Bug number	Description
DLPX-82075	Fixed an issue that prevented the creation of a network route whose gateway is reachable through multiple interfaces.
DLPX-82112	Added checks to prevent using NFSv4 with Direct NFS for some Oracle 19 versions that don't support v4 due to an Oracle bug.
DLPX-82236	Fixed an issue where Speculative Logging was being called out of context and could lead to unbounded consumption of rpool.
DLPX-82308	Fixed an issue where the provision/refresh of an Oracle Key Vault vPDB fails with, "ORA-28365: wallet is not open".
DLPX-82329	Action-based alerts now include the success or failure state of the action, including the reason in case of failure.
DLPX-82334	Enabled the creation of on-link network routes; routes whose destination are directly reachable without a gateway.
DLPX-82381	Improved Replication performance on engines with a lot of objects.
DLPX-82392	Fixed an issue where after editing credential environment variables, the create/provision VDB wizard would fail because the environment variables were missing the password field in the payload. The required password field has now been added.

3.2.63 Release 6.0.15.0 changes

3.2.64 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-80760	Increased the inotify limit to address a defect during upgrade.

3.2.65 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-68240	Fixed an issue where LogSync for an Oracle RAC standby does not set max number of backup tasks correctly.
DLPX-73375	Added granularity in restore job events to mark various phases of Restore Backup process. Added events in Restore Backup job to provide information at the start of the Redo phase of a restore command.
DLPX-75677	Fixed an issue causing auto population of the encryption key when no input is given.
DLPX-78913	Fixed a minor typing issue when deleting a MSSQL database.
DLPX-79228	Fixed a VDB Start Fault after a VDB start job is successful.

Bug number	Description
DLPX-79528	Made improvements in environment monitoring to return the NO_DELPHIX_DATABASE status for Staging Push dSources when they are not managed by Delphix, and avoids monitoring other attributes for such databases.
DLPX-79596	Made a change in the enable flow for Staging Push dSources to always make an attempt at unmounting the DATA storage before dropping the staging database.
DLPX-79780	Mount with local_lock=all on Linux, when mounting VDBs with NFSv3.
DLPX-79999	Use the offset in the time_zone column of msdb.dbo.backupset to convert the timestamp correctly for a backup from a source, and then continue using the staging host timezone to display the time of a snapshot on the UI.
DLPX-80206	Updated MD5 checksums for Oracle 12.1.0.2.0 OJDBC jars.
DLPX-80254	Fixed an issue where Oracle JDBC jar checksum checks are being reported as false positives despite underlying database connection issues.
DLPX-80406	The Spring framework has been updated due to the Spring4Shell vulnerability.
DLPX-80487	Fixed an issue where the Delphix UI would sometimes not render, showing waiting for response.
DLPX-80494	Added an action item to check for support Host and Server Type combination while adding MSSQL environments.
DLPX-80619	This change will introduce the ability to add sporadic failures via tunable: ADDITIONAL_SPORADIC_FAILURES.

Bug number	Description
DLPX-80909	Cross-Site Scripting (Reflected) in /resources/json/delphix/session.
DLPX-81048	Removed the requirement for Linux kernel recover-lost-locks setting when using NFSv4 with Oracle dNFS.
DLPX-81090	Can now enable/disable SNMPv3 vs. v1/v2.
DLPX-81100	Updated the command for checking database files accessibility while fetching the VDBs status with a lighter and less intrusive command, to get relief from VDBs being stopped randomly.
DLPX-81242	Fixed an issue preventing the upgrade of a replication Continuous Vault source with automatic replication.
DLPX-81308	Fixed an issue causing Appdata SnapSync to crash with NullPointerException when a virtual database is not successfully refreshed or rollbacked.
DLPX-81358	Fixed the unnecessary alerts of timezone discovery failure that users were facing randomly for the cluster environments.
DLPX-81502	Improved performance of the Replication page.
DLPX-81696	Users should now be able to enable the feature flag AZURE_DATA_BANK.
DLPX-81710	Fixed an issue where an engine could not be setup when objectStorage is enabled.

3.2.66 Release 6.0.14.0 changes

3.2.67 Security fixes

Bug number	Bug introduced in	Description	Security bulletin
DLPX-81059	5.2.2.0	Arbitrary Code Execution may be performed when configuring masking environments	TB098[60]

3.2.68 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-38908	Oracle LogSync should automatically resolve faults for transient issues
DLPX-39193	Null Pointer Exception in Oracle LogSync backup stream handler
DLPX-57078	Job cancel requests during Oracle provisioning are not processed until the end of recovery
DLPX-57405	move-to-asm.sh does not support TDE-enabled databases
DLPX-62343	disable the OPTIONS method for all HTTP(S) requests
DLPX-64386	Oracle LogSync thread may hang when trying to remove temporary RMAN command file from source host toolkit directory

60 https://support.delphix.com/Support_Policies_and_Technical_Bulletins/Technical_Bulletins/TB098_Arbitrary_Code_Execution_May_Be_Performed_When_Configuring_Masking_Environments

Bug number	Description
DLPX-65413	Ensure "source-archive" directory is unmounted at start of Oracle Provision
DLPX-66879	Oracle LogSync can create orphaned logs in certain scenarios
DLPX-69453	Provide tunable for Oracle LogSync client timeout
DLPX-69802	Common Toolkit directory is not removed from a mounted shared NFS location when the environment is deleted
DLPX-76382	Force disable should succeed despite environmental problems
DLPX-77840	Fixed an issue on the Setup pane to allow successful completion of an smtp test against a specific email address
DLPX-78412	SCM/Talaria failure reason should be communicated in the Delphix fault warnings
DLPX-78726	Removing windows environment performs cleanup of iSCSI persistent login target
DLPX-78754	Disable operation is prohibited on replicated sources
DLPX-78986	Prevent DSP connections for disabled Oracle RAC cluster nodes
DLPX-79077	Resolved an issue of an infinite spinner when validating BEQ credentials for Oracle dSource with duplicate unique_name. While linking a dSource, credentials are now required when discovering an unknown CDB.
DLPX-79242	Splunk HEC token logged to debug logs during splunkHec test

Bug number	Description
DLPX-79396	Allow users to unset Oracle database user name and credentials through CLI if Simplified Connection Management is enabled.
DLPX-79502	SnapSync fails if more than 1000 tempfiles exist in the whole CDB
DLPX-79591	Changed the NFSv4 minimum supported target Redhat version to 6.4 (was previously 6.3).
DLPX-79742	Unable to provision PostgreSQL VDB to Linux host with processor type of ppc64le
DLPX-79823	Improved the action item for failure to enable/attach the staging push dSource.
DLPX-79942	Improved the action item for failure to enable/attach the staging push dSource.
DLPX-80137	Limit SNMP configuration access to sysadmin
DLPX-80144	Oracle SnapSync crashes with NullPointerException when a PDB dSource is renamed and replaced with a new PDB of the same name
DLPX-80217	Fixed issue with filename conflicts during source backup restore
DLPX-80302	It was necessary to restart the auxiliary CDB during a TDE provision after recreating the autologin keystore
DLPX-80369	Oracle environment monitor triggers fault.oracle.db.connection.failed fault immediately on dataset stop or disable.
DLPX-80415	Fixes long delay in operations such as VDB start/stop when JDBC Thin connection to database fails due to unable to establish network connection.
DLPX-80439	Provide mount location to upgrade scripts during Lua to Python upgrade process if mount specification has not been provided.

Bug number	Description
DLPX-80440	Fixed spinner issue while provisioning a VDB from a SQL Server staging dSource from Provision VDB option in datasets menu
DLPX-80482	TDE-enabled provisions to a RAC target fail with "ORA-28365: wallet is not open" while attempting to reopen the database in start_database.sh in the auxiliary
DLPX-80483	Fix failing TDE-enabled vPDB provisions to a linked RAC container database due to "ORA-28365: wallet is not open" errors
DLPX-80487	Improved performance across dataset and replication related pages.

3.2.69 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-80078	The issue with removing files with complex file permissions on EBS is now fixed

3.2.70 Release 6.0.13.1 changes

3.2.71 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-80818	libc upgrade necessitates PostgreSQL re-index.

3.2.72 Release 6.0.13.0 changes

3.2.73 Security fixes

Bug number	Bug introduced in	Description	Security bulletin
DLPX-79789	5.3.0.0	Arbitrary Code Execution May Be Performed by Engine System Administrators.	TB096[61]

3.2.74 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-7868	The issue with the confusing vPDB error on a snapshot after resetlogs of a linked CDB is now fixed.
DLPX-41671	You can now update the Oracle cluster home through the user interface.
DLPX-43467	When dSource is an Oracle standby in RTA mode, LogSync was raising the fault.oracle.linkedsource.log.conflict error and getting disabled on its own. This issue is now fixed.
DLPX-50309	Users can now change logSyncInterval for Oracle dsources.
DLPX-59757	The count for masking jobs fetched from the Masking Engine is now configurable. By default, it is set to 500.
DLPX-67604	The manual recovery of a database after V2P from a snapshot of dSource was failing with an error. This issue is now fixed.

61 https://support.delphix.com/Support_Policies_and_Technical_Bulletins/Technical_Bulletins/TB096_Arbitrary_Code_Execution_May_Be_Performed_by_Engine_System_Administrators

Bug number	Description
DLPX-68684	The self-signed certificate is now compliant with the requirements for trusted certificates in MacOS 10.15.
DLPX-74613	Oracle VDB migration check needs to be done against the Oracle target host instead of the source. Furthermore, the Error and Action plan provided should include the target hostname.
DLPX-75467	The CLI now returns correct and descriptive error messages when executing unauthorized requests on uninitialized engines.
DLPX-75646	To diagnose BEQ connection failure, this release adds MD5 checksums for ojdbc*.jar for Oracle release versions up to currently supported release version.
DLPX-75878	The issue with the JDBC connection string for an Oracle vPDB not getting updated after an IP address change is now fixed.
DLPX-75989	The issue with the failure of environment discovery of an Oracle Cluster with a NullPointerException error is now fixed.
DLPX-76956	Previously, the Oracle JDBC test connection with the wrong password was increasing the `LCOUNT` value by more than 1. This issue is now fixed.
DLPX-77140	This release now speeds up metadata that is sent during replication when Extended retention is involved.
DLPX-77231	Previously, when source discontinuity on the dSource was followed by resync on the livesource, one or more livesource workers were failing to start. This prevented livesource status from getting updated and the first snapshot from being taken after resync. This issue is now fixed.
DLPX-77600	This release fixes NPE when Linking dSource with missing backup and unreachable nodes.
DLPX-77880	This release improves scalability for engines with an extremely large number of snapshots that were causing them to run out of memory.
DLPX-78015	Previously, V2P export with absolute data files was failing with an internal error. This issue is now fixed.

Bug number	Description
DLPX-78174	Insecure DES is no longer supported for SNMPv3.
DLPX-78420	This release adds V2P support for Windows server 2022 host machines.
DLPX-78473	This release improves load times for Datasets and Dataset Performance pages for engines with a large number of datasets and containers.
DLPX-78488	Previously, when switching from the backup server to ASE, dump history was not working for dSources configured to use the remote backup server. This issue is now fixed.
DLPX-78594	This release fixes an issue with disabled VDBs not being able to undo a refresh.
DLPX-78688	TLS 1.0 and TLS 1.1 ciphers are no longer available. Any system that is only configured with TLS 1.0 or TLS 1.1 ciphers is switched to use the default cipher set.
DLPX-78693	Invalid sync parameters will not cause DE server unavailability.
DLPX-78696	Switching sync strategy from source sync strategy type to sourceless strategy type is not allowed.
DLPX-79126	VDBs can now be automatically started with the tunables if stopped intermittently because Windows fail to write on the mount (Msg 9001).
DLPX-79292	This release eases restrictions on taking a snapshot of PDBs with encrypted UNDO tablespaces.
DLPX-79344	Previously, Snapsync of a standby PDB in mount mode was failing with the `ORA-01109: database not open` error message. This issue is now fixed.
DLPX-79422	Previously, clicking on a Replication Profile was resulting in the following error message `An error happened while communicating with the server`. This issue is now fixed.

Bug number	Description
DLPX-79789	Under certain conditions, arbitrary code execution may be performed by sysadmins.
DLPX-79808	This release fixes failures if the VDB name is more than 68 characters.

3.2.75 Release 6.0.12.1 changes

3.2.76 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-79151	The issue with remote syslog configurations preventing engine upgrades or virtualization service restarts is now fixed.

3.2.77 Release 6.0.12.0 changes

3.2.78 Log4j updates

Based on detailed testing and analysis, all the currently supported products are not susceptible to known log4j vulnerabilities. Please refer to TB095 Technical Bulletin[62] for more information. All instances of log4j in currently supported Delphix products are updated to **log4j 2.17.1** as of this release.

Delphix keeps you updated on the latest developments and keeps releasing hotfixes, procedures, and workarounds for such critical vulnerabilities. For more information on how Delphix supports our product and customers in such cases, see Delphix Product Security (see page 945)

For more information, refer to the following pages:

- TB095 log4j vulnerabilities[63]

62 https://support.delphix.com/Support_Policies_and_Technical_Bulletins/Technical_Bulletins/TB095_log4j_Vulnerabilities_(CVE-2021-44228%2C_CVE-2021-45046%2C_CVE-2021-45105%2C_CVE-2019-17571%2C_CVE-2021-4104)

63 https://support.delphix.com/Support_Policies_and_Technical_Bulletins/Technical_Bulletins/TB095_log4j_Vulnerabilities_(CVE-2021-44228%2C_CVE-2021-45046%2C_CVE-2021-45105%2C_CVE-2019-17571%2C_CVE-2021-4104)

- Uninstalling the delphix connector service from the target database servers (see page 2102)
- Delphix product lifecycle policies[64]
- Product security (see page 945)

3.2.79 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-23068	Validation for target database parameter 'DB_FILES' for single-tenant databases for the following operations: provisioning, refresh, rewind, and converting a dSource to LiveSource is now added. Furthermore, specific error messages for handling ORA-00059 errors are added.
DLPX-44544	The issue with the SnapSync of an Oracle standby dSource in Real-Time Apply mode calculating the snapshot's timestamp incorrectly is now fixed. This issue was resulting in ORA-01194 or ORA-01152 errors when provisioning to a timestamp after the snapshot.
DLPX-56691	The issue with the data files of a VDB getting unmounted when provisioning, refresh, or rollback job is canceled manually is now fixed.
DLPX-57971	The issue with the latest snapshot of a LiveSource taking a long time to show the SCN/timestamp range on its card in the GUI is now fixed.
DLPX-60320	UI now allows the selection of older dataset repositories (downgrade) in addition to selecting newer ones (upgrade).
DLPX-67069	The issue with the stopped Oracle VDB monitoring by the environment monitor that resulted in connection errors flooding the debug log is now fixed.
DLPX-68132	The issue with the “Copy query to clipboard” SQL copy functionality in the “Managed Source Data” is now fixed to have correct apostrophe characters.
DLPX-72123	The issue with the failure of detaching or deleting an Oracle dSource operation on RAC environments (This issue was occurring due to failure of deletion of RMAN backups on RAC and the operation needs to be retried with a force option) is now fixed.
DLPX-72779	The UI showing enabled or disabled for cluster nodes now uses a grid table. The Enabled column contains a checkmark to show whether the cluster is enabled or disabled. Users are able to select or unselect the checkmark to enable or disable a cluster.

64 https://support.delphix.com/Support_Policies_and_Technical_Bulletins/Support_Policies/Product_Lifecycle_Policies_(KBA1003)

Bug number	Description
DLPX-73975	Critical storage faults should not be ignorable nor manually resolvable
DLPX-74862	This release fixes an issue where the RESUME operation failed without any error thrown to the user on dSources where ENFORCE was still in progress. The fix will make sure that even if RESUME fails, it throws a DUE to the user with suggested actions to resolve the issue.
DLPX-75763	The issue with the failure of refreshing a VDB provisioned as an empty vfiles since there is no parent container to refresh from is now fixed.
DLPX-76266	The issue with the VDB Disable operation that resulted in an error message while connectivity with the host machine can't be established is now fixed.
DLPX-77123	You can now run Upgrade Verify when another upgrade is in progress.
DLPX-77347	This release fixes the difference in time shown for MSSQL snapshot based on different database authentication methods.
DLPX-77638	The issue with the failure of the End Entity Certificate expiration fault is now fixed.
DLPX-77664	The issue with the failure of the Oracle SnapSync with an error message, "RMAN-06183: datafile or datafile copy (file number) larger than MAXSETSIZE" if a datafile resized in the middle of SnapSync is now fixed.
DLPX-77913	The issue with the faults table missing data if there was more than one page of faults is now fixed.
DLPX-77925	The issue with the unsupported Windows release error message is now fixed.
DLPX-78113	MSSQL VDB database size will now be refreshed periodically based on environment_monitor.dynamic_attributes_check_period tunable.
DLPX-78183	The issue with the MSSQL validated sync schedule not getting updated without successful backup restoration is now fixed.

Bug number	Description
DLPX-78244	The issue with the failure of a few operations on self-service containers due to incorrect entries corresponding to the Oracle log metadata on the Delphix engine is now fixed.
DLPX-78258	The issue with the input bug that retained cleared out DB credentials in the dSource linking wizard is now fixed.
DLPX-78263	The issue with the failure of a SnapSync of an Oracle standby dSource in Real-Time Apply mode with an error message, "exception.oracle.snl.linkedsource.current_scn.invalid" if the rate of change in the database is low is now fixed.
DLPX-78265	Offline Oracle bystander PDBs data files can now be optimized leading to improved provision performance.
DLPX-78309	The issue with the CLI being unable to log in to system users when the main virtualization service is down is now fixed.
DLPX-78334	The issue with a large number of missing Oracle archive logs causing an error while viewing dataset is now fixed.
DLPX-78392	Hosts running Windows Server 2022 can now be added as Source and Target environments to the Delphix Engine.
DLPX-78522	SSLv3, TLS 1.0, and TLS 1.1 are no longer configurations options for HTTPS. Any system configured only with these removed options will be automatically set to use TLS 1.2.
DLPX-78791	This release upgrades log4j from 1.2.17 to the latest 2.x in Windows Connector.
DLPX-78938	This release upgrades log4j in virtualization to 2.17.1.

3.2.80 Release 6.0.11.0 changes

3.2.81 Security fixes

Bug number	Bug introduced in	Description	Security bulletin
DLPX-77921	6.0.8.0	Arbitrary Code Execution by Delphix System Administrators may be Performed on Virtualization and Masking Engines	TB094[65]

3.2.82 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-53019	The issue with the missing redo alert raised during the environment monitor check has now been resolved.
DLPX-59299	Discovery and monitoring rely on "Connected" in sqlplus output, which may not be the case if NLS_LANG is set to another language (e.g. Japanese). Downstream operations, like linking or provisioning, may then fail due to missing user privileges.
DLPX-59662	The issue with copy-Only Backups failure with Virtual Service Accounts has now been resolved.
DLPX-62706	The issue with the Hostchecker not properly checking /home/delphix permissions has now been resolved.
DLPX-64082	The issue with Oracle Provisioning scripts having hard-coded timeout issues has now been resolved.

65 https://support.delphix.com/Support_Policies_and_Technical_Bulletins/Technical_Bulletins/TB094_Arbitrary_Code_Execution_by_Delphix_System_Administrators_May_Be_Performed_on_Virtualization_and_Masking_Engines

Bug number	Description
DLPX-65729	Added retry functionality to the 'read backup files' operation during a validated sync to an account for an unstable environment.
DLPX-69778	SAML response is not logged on successful SSO login.
DLPX-72043	The issue where LiteSpeed `xp_restore_headeronly` stored procedure failure message are displayed when validated sync is active for dSources with LiteSpeed backup has now been resolved.
DLPX-72220	A UI issue that occurred while updating the vault when only the private key is changed has now been resolved.
DLPX-72225	Admin user created from management UI is no longer showing as 'non-admin' type.
DLPX-72237	The 'Verify Credentials' button from the DSP Throughput test page is now removed.
DLPX-72369	The dependency on a parent snapshot relying on the latest snapshot is now removed if a parent snapshot does not exist during the VDB enable operation.
DLPX-72778	Oracle dSource attach operation with changed DB ID using 'Force' option is now allowed.
DLPX-74555	Updated the "no Delphix connector" message while provisioning a Windows source environment.
DLPX-74676	Oracle LiveSource LogSync should only catalog valid archive log files.
DLPX-74851	In the Add Environment GUI, the mouseover information for "Set Delphix Session Protocol Options (DSP)" has been currented.
DLPX-74896	The race condition issue when running Oracle VDB refresh and dSource snap sync resulting in incorrect engine metadata entry for the parent snapshot of a VDB in the `dlpx_timeflow` table has now been resolved.

Bug number	Description
DLPX-75335	Added a product name and product version for the Delphix Connector executable so this information can be available before installation.
DLPX-75500	For ag cluster nodes, if the refresh fails due to timezone discovery failure, don't delete the nodes from engine metadata as it doesn't mean we had an issue with the nodes.
DLPX-75952	Database configs will be replicated only if the associated VDB is replicated.
DLPX-75995	The issue causing environment 'Add' or 'Refresh' to fail when PowerShell Transcription is enabled has now been resolved.
DLPX-76244	The issue where TCP fallback connection to database stops responding if the Oracle database instance is down has now been resolved.
DLPX-76290	Databases of UNKNOWN cdb type are now included in the attachment of a non-PDB container.
DLPX-76731	Added Delphix support for `WALLET_ROOT` and `TDE_CONFIGURATION` parameters to manage wallets in 19c instead of sqlnet.ora.
DLPX-76759	Added "Response" to faults along with other details when logged in the Admin App.
DLPX-76777	Remove orphaned Oracle logs resulting from archive log fetch timeouts.
DLPX-76793	Added execution timeout for execution of `UpdateFileACL.ps1`.
DLPX-76974	The issue where a user was unable to change 'from address' of SMTP server to noreply@delphix.com[66] in the GUI has now been resolved.
DLPX-77112	The issue where an Oracle VDB cannot be provisioned between different minor versions if the Source is on higher RU has now been resolved.

66 mailto:noreply@delphix.com

Bug number	Description
DLPX-77284	The issue where after a hotfix was removed due to a successful upgrade, the system would still indicate the hotfix was installed post-upgrade has now been resolved.
DLPX-77345	The issue where provisioning a vVDB fails with `java.lang.OutOfMemoryError` when sqlplus is used to rename the datafiles has now been resolved.
DLPX-77405	Replicated password vaults will no longer be visible in the UI.
DLPX-77676	The issue where provisioning a vPDB from a PDB dSource fails with "ORA-65114: space usage in container is too high" if PDB `max_size/max_pdb_storage` is configured has now been resolved.
DLPX-77708	The issue where a refresh/disable/destroy of a VDB using NFSv3 could cause loss of access to other VDBs that were using NFSv3 has now been resolved.
DLPX-77844	The issue where V2P operations from a VDB snapshot would result in the deletion of any production datafiles that exist on specified V2P target directory has now been resolved.
DLPX-77904	Removed 'Factory Reset' for Delphix Engines that are Data Vaults, as the operation is disabled for those engines.
DLPX-77912	The issue that can cause a VDB stop, refresh, or rollback to fail with an internal error has been resolved.
ORB-2465	Removed the requirement that SAML SSO email addresses must match case-sensitively for SSO logins.

3.2.83 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-77577	Increased nvme I/O timeout to prevent storage issues in EC2 (Activated after optional Reboot).

3.2.84 Release 6.0.10.1 changes

3.2.85 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-77467	Loading the setup app dashboard (as sysadmin) was rendering a server error popup with instruction to contact Delphix Support. This 6.0.10.0 error has been known to impair the ability to configure web proxy, PhoneHome, SMTP servers, and other connectivity settings via the GUI. It has now been resolved.

3.2.86 Release 6.0.10.0 changes

3.2.87 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-18438	The issue with provisioning to the latest available time that resulted in generating the `exception.oracle.target.point.not_provisionable` exception has now been resolved.
DLPX-35480	Previously, static routes were being added using the add command. Now, the same can be added using the create command.
DLPX-57516	The issue with the failure of management service to start after configuring some abbreviated timezones from a picklist in server setup or sysadmin CLI has now been resolved.
DLPX-58133	Previously, the Oracle SnapSync operation was resulting in a warning for BCT usage on editions that do not provide it. This issue has now been resolved.
DLPX-58675	The issue with the deletion of the last snapshot on timeflow by Retention during a failed Oracle DB_SYNC operation has now been resolved.

Bug number	Description
DLPX-63003	The issue with memory being exhausted while reading too many snapshots from engine metadata has now been resolved.
DLPX-63347	If the staging source has the "Use as Staging" flag set as off, the user was seeing a specific exception while trying to enable a linked dSource to point in the direction of what needs to be done. Any compatibility failure will now have a specific exception.
DLPX-63601	Previously, querying the following operation "backupset table" and "whether a database is part of AG or not" was resulting in deadlocks and lock timeouts errors. We have now added retries to resolve the issue.
DLPX-64369	The issue with throwing `fault.oracle.linkedsource.incomplete.tempfile` for physical standby has now been resolved.
DLPX-65949	The issue with misleading status in the progress bar while taking a copy-only backup has now been resolved.
DLPX-69831	Previously, the Oracle dSource SnapSync operation was not displaying a clear failure message if a dSource `db_unique_name` is changed. This issue has now been resolved.
DLPX-70317	The issue with the restarting of the NFS-server by the reaper thread while deleting a vPDB from a linked CDB with Talaria turned on has now been resolved.
DLPX-71018	The issue with UI displaying only the suffix of the device name used by Hyper-V has now been resolved. UI now displays a unique device name for storage in Hyper-V.
DLPX-71292	The issue with the allowance of incremental SnapSync after LogSync throws `fault.oracle.linkedsource.log.conflict` has now been resolved.
DLPX-71639	The NFSv4 is now set as the default option when mounting datasets from OS platforms that support it.
DLPX-71769	The need to set permissions of `$ORACLE_HOME/dbs` subdirectories using STARTUP SPFILE syntax is now removed.
DLPX-72011	The issue with the CLI network setup not configuring the first network interface when multiple interfaces exist has now been resolved.

Bug number	Description
DLPX-72186	The issue with CDB log file retention working incorrectly if a PDB has multiple time flows pointing to the same CDB timeflow has now been resolved.
DLPX-72432	The format of `zpool_iostat_60.log` has been enhanced in this release. A timestamp is recorded for each sample in the log, making it easier to determine the time for each sample.
DLPX-72956	The issue with disabling the Oracle LogSync after running the validated sync job has now been resolved.
DLPX-73575	The timezone monitoring is now added for the Windows hosts.
DLPX-73590	You can now refresh a VDB whose parent dataset is in a different group without needing authorization on the parent or its group.
DLPX-73800	The issue with the failure of olsnodes when run as a non-Oracle user has now been resolved.
DLPX-74504	The issue with throwing a new DUE and NotFoundException when ojdbc libs cannot be read has now been resolved.
DLPX-74945	UI now displays a detailed error message for transaction log-chain break fault.
DLPX-74992	The issue with the failure of SnapSync operation when Database incarnation reset-logs end time is changed from "2021-03-13 22:03:07.0" to "2021-03-13 21:03:07.0" for virtual pluggable database “ " has="" now="" been="">">
DLPX-75389	The issue with the recording of the insufficient details by Logsync when dbid change was detected has now been resolved.
DLPX-75517	The issue with the failure of the Oracle vPDB provisioning with the "ORA-00959: tablespace 'TEMP' does not exist" error has now been resolved.
DLPX-75721	The issue with the failure of an Environment discovery with the "DelphixFatalException: Unknown Oracle Database status: REFRESHING" error has now been resolved.

Bug number	Description
DLPX-75737	The issue with saving unnecessary logs by Retention if bookmark falls exactly on a snapshot end SCN or snapshot end timestamp has now been resolved.
DLPX-75897	Previously, failure to start I/O services after the upgrade operation was resulting in a stack restart loop. This issue has now been resolved.
DLPX-75951	The internal error being signaled during VDB SnapSync by `removeUnneededZFSFiles` when a data file is physically removed during processing by an external cause has now been resolved.
DLPX-76288	The NFS latency for workloads involving a lot of parallel I/O (e.g. Oracle VDBs with concurrent accesses to many data files) is now improved.
DLPX-76388	Previously, entering key pairs directly into hook environment variables, as opposed to via a vault or as passwords were resulting in an internal error. This issue has now been resolved.
DLPX-76406	The issue with NFS-based VDBs becoming unresponsive has now been resolved.
DLPX-76447	The issue with the V2P Functionality to customize target directory structure for exporting database files to separate file systems not working as documented has now been resolved.
DLPX-76613	The issue with the unnecessary accumulation of heap when validated sync is active for dSources using an environment user that eventually can cause out of memory issues has now been resolved.
DLPX-76690	The issue with the removal of extraneous Oracle data files while creating snapshots has now been resolved. The extraneous Oracle data files are now removed during the provisioning operation.
DLPX-76692	For V2P operation, we now use the unbuffered copy method for better performance.
DLPX-76718	The issue with the creation of the extraneous Self-Service branch segments during the replication operation has now been resolved.
DLPX-76760	Environment clusters will now show faults from their child nodes.

Bug number	Description
DLPX-76802	Previously, the engines that are in the DEFERRED upgrade state were resulting in the "Large Receive Offload" option turned off which was leading to performance degradation in network transfers. This issue has now been resolved. Upgrading the engines that are in the DEFERRED state will also resolve the issue. Screen reader support is now enabled.
DLPX-76891	CRON expression labels now ask for Quartz format on the user interface.

3.2.88 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-76619	The NFS reliance on DNS to prevent VDB unresponsiveness related to DNS unreliability is now reduced.
DLPX-76203	The NFS latency for engines with many Oracle dNFS clients is now improved.
DLPX-76119	The issue with the Delphix Engine crashing or becoming unresponsive when canceling a replication job has now been resolved.
DLPX-76991	Optimized in-memory cache eviction by making minor improvements to I/O performance.

3.2.89 Release 6.0.9.0 changes

3.2.90 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-28435	MS SQL instances PatchLevel will be displayed in preference to the version on the UI.
DLPX-40005	Improved the error message that is displayed when a JDBC connection cannot be established or the Oracle database Instance is unavailable.
DLPX-48080	Oracle Home Check may generate spurious faults if Oracle Home entry does not exist in /etc/oratab.
DLPX-55951	Attempting to provision a plugin-based VDB onto an incompatible OS (Windows to Linux or vice versa) is possible in the UI, and would fail with a crash requiring a restart. Now an informative error message is shown after the attempt is made.
DLPX-59613	Fixed creation of retention policies workers on replica objects after failover.
DLPX-64307	Environment refresh should ignore cluster discovery for Oracle VDBs.
DLPX-66191	Fixed the side-effects of the native Windows "Recent Files" behavior, when large numbers of PowerShell operations are being run concurrently.
DLPX-67537	Domain administrators can now create, view, and edit the alert profiles of other domain users.
DLPX-69605	Poor error message when selecting Timeflow range.
DLPX-70502	On detaching a dSource, delete backup server entry from engine metadata and related ones if the backup server is unused.

Bug number	Description
DLPX-71002	In case we have null values coming for recovery_model from msdb.dbo.backupset table, the user will see a generic exception for manual sync and a fault for validated sync.
DLPX-71908	Users will now see a warning when they remove any object from the replication specification list.
DLPX-72012	Prevents the same IP address from being configured on more than one network interface.
DLPX-72411	Environment names for Windows and Oracle Clusters are once again editable by users.
DLPX-72609	When we do MSSQL standalone environment discovery, the user will see a warning for databases attached to AG that are present and will not be discovered unless cluster environment discovery is selected.
DLPX-72695	Improved Oracle SnapSync performance by eliminating unnecessary calls to getTotalHoleBlocks.
DLPX-73409	Duplicate listener entry gets generated in engine metadata if Oracle listener is manually started with a non-uppercase name.
DLPX-73586	Fixed a display error of some snapshot names in the command line interface which showed references instead of actual names.
DLPX-73720	Provisioning an Oracle vPDB fails with "ORA-65149: PDB name conflicts with existing service name in the CDB or the PDB" if the PDB and CDB names are the same.
DLPX-74050	Environment names for Windows and Oracle Clusters are once again editable by users.
DLPX-74078	The Target Directory path is combined with other directories such as Data Directory, Archive Directory, Temp Directory, etc to build the full path for data files, archive logs, temp files, etc. As long as the combined paths are valid the V2P job proceeds.
DLPX-74197	Fixes a misleading warning about insufficient space on a replication target.

Bug number	Description
DLPX-74201	Snapshots created as a result of refresh or rewind operations will now be labeled as just "Snapshot" to avoid confusion. Users are advised to look at the Timeflow markers to know when the Timeflow operation was performed.
DLPX-74367	Delphix Engine repeatedly reports "Failed to parse logfile".
DLPX-74377	Improved diagnostics information for the case when Delphix Engine fails to connect to the Windows host.
DLPX-74387	fix an issue that causes the management service to crash under heavy CPU load.
DLPX-74398	Added handling of dangling nodes during Windows cluster environment add and refresh operations.
DLPX-74486	Delphix OS users cannot provision 12.2 TDE vPDB due to directory permissions in the default wallet location.
DLPX-74495	Enabled more logging in Delphix connector logs for timeouts.
DLPX-74681	During RAC vPDB provision, Oracle 19.9 target CDB crashes with ORA-00600 [krccfl_chunk] when BCT is enabled.
DLPX-74806	Improved error message displayed when the storage device initialization fails unexpectedly.
DLPX-74860	Provisioning the 2nd generation VDB fails if the dSource has imported read-only transportable tablespaces fails.
DLPX-74975	Allow adding invalid or unreachable paths as a shared backup location for dSources.
DLPX-75026	Invalid JDBC connections are not purged from the connection pool when the home is changed.

Bug number	Description
DLPX-75134	Improved performance for the Environment Databases pages when there are a lot of databases.
DLPX-75363	Update exception description is seen when the SnapSync fails for ASE encrypted database.
DLPX-75401	local listener set to null if oracle.lsnr.protocol_registration_order is quoted.
DLPX-75506	Fix a bug that can cause Oracle RAC VDBs to fail with stale NFSv3 mounts if NFSv4 is also enabled.
DLPX-75532	Insufficient heap memory settings on AIX cause connector and SnapSync to hang or crash.
DLPX-75663	FIPS compliant algorithms will be used while merging the old and new toolkit directories during environment refresh.
DLPX-75716	Delphix may remove Oracle VDB temp tablespaces during Snapsync.
DLPX-75735	Fixed creation of retention policies workers on replica objects after failover.
DLPX-75834	Rearranged Syslog configuration dialog inputs to avoid confusion and have a more consistent user experience.
DLPX-75858	BEQ processes can hold on to file descriptors leading to hook scripts hanging after upgrading to 6.0.7.0.
DLPX-76018	Remove hardcoded 5-minute timeout for doDropPDBKeepDatafiles.sh.
DLPX-76140	TDE SnapSync should ignore WARNING plugin violations.

3.2.91 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-71980	Fix a bug that was causing the Delphix Engine storage pool to fail to import on boot under certain circumstances.

3.2.92 Release 6.0.8.1 changes

3.2.93 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-75804	Some Delphix operations may fail if mount and umount commands, on staging or target hosts, are setup to run as sudo and if sudo rules prohibit these commands from running with unrecognized options. The issue is fixed now after removing "-v" added in 6.0.8.0.

3.2.94 Release 6.0.8.0 changes

3.2.95 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-49694	Skip parsing of DBCC when code 0 is not present in the DBCC page output. In case the DBCC page has code 0 but not dbid, use bdbid (present in the buffer section).
DLPX-68764	VDB SnapShot does not progress if ASE database devices are not on Delphix storage, environment monitoring raises a fault. Subsequent VDB operations like enable, disable, start, stop, delete, snapshot, or refresh will fail.

Bug number	Description
DLPX-70793	Delphix Engine should not allow linking Oracle DB with null `db_unique_name`.
DLPX-71300	For newer ASE versions (>= 15.7 SP138 and 16.0 SP02 PL05 and ASE 16.0 SP03), do not run DBCC PAGE anymore, as it was an identifier for DBCC CHECKALLOC that is already not run.
DLPX-71471	Error message asking user to manually perform disable/enable operation or correctly rename the target database back will be displayed during Start VDB, if VDB does not exist.
DLPX-71687	Provide a mechanism to enable VDBs up to filesystem mount point.
DLPX-71875	Fixed a bug that results in a memory reservation not being represented correctly in the Delphix API.
DLPX-72046	Deletion of vPDB in a vCDB shows this warning, "Encountered an error while shutting down and cleaning up Oracle files."
DLPX-72209	Downloading a support bundle is not supported at the same time that an upload of an upgrade image has been initiated by the same Delphix user.
DLPX-72319	Fixed an issue where some error dialogs would freeze in Internet Explorer 11.
DLPX-72705	Connection timeout when deleting remote shipper script can cause a timeout in LogSync client.
DLPX-72757	ASE sync using Dump History fails for large dump history files.
DLPX-72780	Timezone is set incorrectly for snapshots of Solaris 10 dSources and VDBs.
DLPX-72904	Storage capacity now includes usage from all file system objects, not just snapshots.
DLPX-73143	Fixed an issue where the support bundle dialog showed a loading spinner intermittently while jobs were running.

Bug number	Description
DLPX-73354	Traverse all shared backup locations while syncing, even if some of the paths are invalid or not reachable.
DLPX-73489	Fixed bug where adding or editing a parameter using the UI VDB Config Template "Text" tab was truncating the parameter's value.
DLPX-73602	Incorrect mount options used when a single instance RAC is linked as a standalone single instance.
DLPX-73607	Added paging for days with large numbers of snapshots to prevent slowdown.
DLPX-73623	Fixed an out-of-memory condition that occurs in SSH tunneling for encrypted log-syncs when storage latencies are high.
DLPX-73627	The help text on upgrade replication warnings have been updated to avoid confusion between Ignored and Resolved.
DLPX-73668	Fixed Missing security headers.
DLPX-73669	Cross-site request forgery (CSRF) issue in management UI.
DLPX-73727	Fixed an issue where the faults table was unable to navigate to other pages.
DLPX-73797	Fixed VDB refresh failures due to SQL Server Error 924 after setting VDB to single user mode.
DLPX-74025	Implemented logic to retry offline database along with a drop database to overcome deadlocks while off-lining or dropping the database.
DLPX-74029	VMware Hot-Add memory is not immediately reflected in the system API.
DLPX-74057	Fixed a typo in "Download Support Bundle" UI component where the word "suport" was missing a "p".

Bug number	Description
DLPX-74254	Ownership of files inside VDB now matches new owner when VDB owner is changed.
DLPX-74298	Fixed an issue where the user could not upload a keystore with a blank keystore passcode.
DLPX-74362	Fixes an issue with namespace deletion when the replication receive jobs have been cleaned up.
DLPX-74442	VDB Enable with attemptStart=false will now mount the datasets so that VDB can be started.
DLPX-74457	Cluster discovery for Oracle RAC clusters are partially failing on Solaris 10.
DLPX-74529	Fixed a bug so that an upgrade completes even when jobs fail.
DLPX-74542	Fixed a bug so that upgrade completion is properly handled after kernel upgrades.
DLPX-74645	Delphix Engine uses the uptime command to keep track of a target host reboot and auto start VDBs on the host. In some cases, the output of this command is not what is expected and causes unintended restart of a stopped VDB. This issue is now fixed.
DLPX-74656	Oracle errors during doCreateSPFile.sh are not captured.
DLPX-74704	Fixed a bug where the Dataset scroll does not extend to the bottom of a dataset list, thus truncating the status of the last dataset in the expanded group.
DLPX-74883	Prevent support bundle collection from cancelling replication.
DLPX-74911	Talaria TCP fallback fault may be misconstrued if an Oracle RAC node is down.

Bug number	Description
DLPX-74997	Prevent granting replicated roles to users.
DLPX-75083	Post upgrade cleanup task may become unresponsive while attempting a migration from 5.3.x to 6.0.x due to several threads stuck in WAITING state.
DLPX-75095	Provisioning an Oracle VDB fails if change-archivelog-mode.sh takes longer than 5 minutes.
DLPX-75134	Improved performance of the Environment Databases page under certain conditions.
DLPX-75188	Fixed "out of memory" issue when processing a large number of objects on the Target engine.
DLPX-75204	Addressed a performance issue on the Target engine when receiving large number of replicated objects.
DLPX-75208	Snapshot names are incorrectly redacted in the engine metadata dlpx_action table in support bundles.
DLPX-75416	Fixed a replication issue when there are sources with TLS enabled.

3.2.96 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-72065	Fixed a bug that can cause a Windows iSCSI initiator to fail connect to the Delphix Engine.
DLPX-72681	Console Delphix status screen shows a Python stack trace if the system is configured with a static IP address.

Bug number	Description
DLPX-73423	Console Delphix status screen shows a Python stack trace if the system has no default route.
DLPX-74216	Fixed an issue that causes management service failures in low memory situations.
DLPX-74622	Fixed a bug that can cause a replication job to fail with an internal error.
DLPX-75089	Fixed a bug that can cause NFSv3 clients to lose locks during upgrade verification.
DLPX-75524	Fixed a bug that can lead to Oracle data corruption when running VDBs on Oracle 19c with dNFS.

3.2.97 Release 6.0.7.0 changes

3.2.98 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-39006	LogSync failed with "Cannot read archived log due to failure of log shipping script".
DLPX-39245	Fixed a bug that caused the management service to become inaccessible if the storage pool ran out of space.
DLPX-59155	Provisioning a VDB or vPDB failed with unclear error message 'A database with the name "xxx" already exists'.
DLPX-60317	Fixed Out of Memory issue when replicating a large number of objects.

Bug number	Description
DLPX-60947	Replica VDBs will be updated when performing a point-in-time restore.
DLPX-62805	vPDB provision did not raise an error when a non-provisionable target point-in-time was provided.
DLPX-62969	Fixed Out of Memory issue when receiving large number of replicated objects.
DLPX-64600	Skipped connecting to ASE dSources during SnapSync policy runs as it is not applicable for them, hence prevent recurrent faults that the policy throws for connectivity issues.
DLPX-67363	Maximum identify provider authentication time age can be customized for single sign-on.
DLPX-67607	Fix to make Snapsync throw exception if manifest file is missing or of 0 bytes instead of internal error with null pointer exception.
DLPX-67767	Fixed a bug that caused the upgrade to hang, while waiting for running jobs to finish.
DLPX-70821	Allow the entity id for SAML single sign-on to be a URL for compatibility with Azure AD.
DLPX-71783	doRenameDatafiles cleanup of extra files fails due to file permissions mismatch.
DLPX-72010	Fixed an issue that prevents changing the default gateway using the network setup CLI.
DLPX-72075	Maximum SAML response time skew can be customized for single sign-on.
DLPX-72191	Oracle privilege discovery not performed for all homes if an invalid home exists.
DLPX-72351	When a user tries to change credentials for a dSource, validating the credentials before updating them. In case of invalid credentials, showing user an error message about it.

Bug number	Description
DLPX-72545	Initial ORA-65294 error not reported to user when vPDB provision fails due to compatible parameter mismatch.
DLPX-72652	Fix and issue that prevents use of the NFSv4 on some versions of SUSE Linux targets.
DLPX-72698	Patching Oracle 19C vCDB leads to ORA-25153 as described in 2285159.1.
DLPX-72807	Fixed issue with SQL Server 2014 dSources with filestreams where sync failed in merging filestream directories due to long path names.
DLPX-72882	Datasets hooks script editor properly displays multiline scripts instead of as one long line on non-Chrome browsers.
DLPX-72916	Empty string in SNMPv3 USM username creation no longer throws fatal error.
DLPX-73048	Non-sys user credentials for Oracle sources cannot use password vault.
DLPX-73108	Fix a bug that prevents the API from displaying the correct number of CPUs or amount of memory assigned to a Delphix Engine after a hot-add operation.
DLPX-73201	Fix an issue that prevents the configuration of additional NICs on Azure Delphix Engines.
DLPX-73202	Fix a bug that can cause a VDB to fail to mount while other VDBs are being stopped.
DLPX-73424	Fix a bug that prevents the sysadmin from deleting a default route.
DLPX-73527	SnapSync job fails with 'internal error during execution' due to ONS/FanManager errors.
DLPX-73528	Fixed a bug that prevented accessing SDD specs from CLI.

Bug number	Description
DLPX-73611	Kerberos ticket expiration date parsing is incorrect after migration from Illumos to Linux.
DLPX-73742	Provisioning an Oracle TDE-enabled vPDB fails with the error "ORA-28367: wallet does not exist" if the TDE wallet for the target linked CDB is stored on ASM storage.
DLPX-73765	Fix a file descriptor leak that causes the management service to crash over time.
DLPX-73789	Auxiliary CDB instance uses dSource keystore location if WALLET_ROOT is configured in dSource.
DLPX-74030	CDB database password may be leaked as part of environment monitor checks that launch sqlplus command on the source or target host.
DLPX-74043	Delphix OS user cannot provision TDE-enabled vPDB due to directory permissions in the default wallet location.
DLPX-74044	Delphix OS user cannot provision TDE-enabled vPDB in Delphix-writable keystore location due to directory permissions.
DLPX-74119	Drop database fails if default database is set to any other than master.
DLPX-74164	Sync fails with db.aseldb.source.dump_history.incomplete_stripes after dump history file is purged and Use dump history is enabled for the dSource.
DLPX-74233	During failover of a namespace, if there is a collision between an environment in the namespace with one on the target engine, the namespace environment will get renamed if its host does not match that of the environment on the target.

3.2.99 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-73390	Improve replication receive throughput.
DLPX-73393	Improve write performance under extreme disk fragmentation.
DLPX-73280	Improve write performance under extreme disk fragmentation.

3.2.100 Release 6.0.6.1 changes

3.2.101 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-73848	Fixed an issue that can cause the management service to fail to start after upgrade on systems that have had SNMP enabled.
DLPX-73859	Fixed a file descriptor leak triggered by faults and alerts that can cause the management service to fail.

3.2.102 Release 6.0.6.0 changes

3.2.103 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-47065	VDB recovery failed when files other than archive logs were detected by Oracle.
DLPX-47493	Fixed the bug where VDB directory under the DelphixConnector directory was not being removed from the target host on MSSQL VDB deletion.
DLPX-48046	Added sorting parameter to network test APIs.
DLPX-61405	Replication may send more data than expected if masking involves dropping large DBF files.
DLPX-61525	The height of the storage configuration list was limited to show 3 disks at a time. It will now dynamically grow with the number of disks.
DLPX-63603	Increased connector timeout from 10 minutes to 30 minutes to avoid unnecessary faults due to timeout during Validated Sync operation.
DLPX-67368	Delphix Engine hostname change is now immediately reflected in Splunk events.
DLPX-67593	Fixed an issue that caused the management service to remain offline following an out-of-space condition.
DLPX-68531	Introduced better handling of UniversalConnectionPoolException errors during SnapSync.
DLPX-69759	Oracle environment discovery failed due to an unhandled exception occurring at insert into dlpx_faults.
DLPX-69852	Fixed a bug that caused network configuration problems when removing and adding additional NICs.

Bug number	Description
DLPX-70426	Redaction of usernames took forever on tables with millions of entries.
DLPX-70583	move-to-asm.sh fails if timing is set in glogin.sql.
DLPX-70638	Removed Failed Actions section of Actions sidebar, in favor of manually dismissing from Running Actions and falling to Finished Actions section.
DLPX-70653	Removal of all instances in a RAC VDB should not be allowed.
DLPX-70808	Fixed issue related to the creation of empty DisableBroker.sql on the Windows machine in case DisableBroker.sql execution fails in the first attempt.
DLPX-70896	Added more detailed error message for when the Delphix Engine fails to push a script to Windows host.
DLPX-70919	Fixed an issue that causes job progress to not update in Self-Service.
DLPX-70928	Fixed a bug that results in a Delphix Engine remaining powered on following a shutdown from the user interface.
DLPX-71093	For AG databases, a full backup is not required even recovery fork guid changed but the LSN chain didn't break because of transactional log backups.
DLPX-71097	Unable to ignore snl.bct.needed warnings if Block Change Tracking is legitimately disabled on an Oracle dSource.
DLPX-71153	Recovery of PDB should fail if the database is down after offlining datafiles.
DLPX-71370	While deleting initiator in Windows environment deletion operation, delete all the views as well for that initiator.
DLPX-71685	VDB is auto disabled if the hook fails.

Bug number	Description
DLPX-71865	Reduced the size of support bundles.
DLPX-71961	When a PDB is selected for replication, its CDB and all other PDBs in the parent CDB get automatically selected for replication. Going forward, in the above scenario, while the CDB will get selected, its other PDBs will no longer get selected.
DLPX-72031	Fixed VDB refresh operations failures due to 'DB STARTUP' background process spid greater than 50.
DLPX-72066	Migrate VDB verifies against the old configuration, rather than new.
DLPX-72083	Fix an issue that causes a fully-qualified hostname to be changed on upgrade from 5.3 to 6.0.
DLPX-72131	Added namespace support for HashiCorp password vaults.
DLPX-72265	doCreateTempfiles.sh.template exits with code 0 on failure.
DLPX-72340	Incomplete recovery not detected during provisioning.
DLPX-72386	Unlock Solaris x86 Solaris -> Linux x86 provisioning.
DLPX-72452	For clusters with long hostnames, vPDB sync fails with exception.oracle.accessor.instances.missing.
DLPX-72495	Fixed a bug that prevents the application from coming up after an upgrade
DLPX-72686	Delphix no longer logs environment variables in logs on connected hosts since this could leak sensitive information such as passwords that are sometimes stored as environment variables on database hosts such as for the ASE database.

Bug number	Description
DLPX-72730	Fixed a Snapsync performance issue.
DLPX-72790	SnapSync job fails with 'internal error during execution' due to ORA-01652.
DLPX-72862	The scenario which was causing the null pointer has been fixed now.
DLPX-73300	Validation of connection to a container for PDBs should allow connecting to CDB$ROOT.
DLPX-73311	Added platform detection for ESX 7.0u1.
DLPX-73449	Replication of policies between two engines, in a loop, could lead to OOM exceptions.

3.2.104 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-72990	Addressed a minor CVSS 5.9 security issue with no known attack vectors.
DLPX-73067	Fix for CVE-2020-10753.
DLPX-73069	Fix for CVE-2020-12059.
DLPX-73070	Fix for CVE-2020-1760.

3.2.105 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-71924	Fixed a bug that causes support bundle collection to fail with an internal error.
DLPX-72918	Fixed a system crash that can happen when replicating a masked VDB using SDD.
DLPX-73147	Fixed a bug that can cause a replication source to crash if it had run replication while running on 5.0.

3.2.106 Release 6.0.5.0 changes

3.2.107 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-23360	The unistallation of the Delphix Connector installer should succeed even if one of the component connector services doesn't exist.
DLPX-69155	Reduced the time taken to generate support bundles in some cases.
DLPX-67316	Recreating a controlfile against an Oracle source may yield misleading error during snapshot.
DLPX-70766	JDBC driver updated to resolve intermittent JDBC connection failures due to JDBC SSL bug.
DLPX-70785	Options passed to VDB mounts on target AIX hosts did not include read and write size values. This fix adds the rsize and wsize parameters to mount command depending on the maximum values host is configured to support.

Bug number	Description
DLPX-70741	Enabling Validated Sync while SAP ASE SnapSync job is running leaves staging database unrecoverable.
DLPX-69865	Fixed a bug that causes a network interface to become unconfigured if its MAC address changes.
DLPX-71233	If LogSync is suspended when performing SnapSync of a standby database in real-time apply, SnapSync attempts to backup the archived logs which can cause SnapSync to become unresponsive.
DLPX-69800	UEM/Hostchecker directory ownership checks fail on HPUX environment with long usernames.
DLPX-69807	Provided mechanism for the user to bypass corrupted/incomplete jdbc libraries.
DLPX-66585	Bundle ID "fault.environment.configuration.file.owner" reports insufficient host address.
DLPX-65739	createDelphixDBUser.sh fails when "@" used in the password.
DLPX-70973	SAP ASE database provisioning fails if the source database has holes in log fragments.
DLPX-71532	Improved error handling for Oracle memory configuration errors.
DLPX-62987	Allowed assigning privileges over replicated objects through the UI.
DLPX-71593	TIMEFLOW_REPAIR incorrectly skips a log because of "wrong database".
DLPX-71751	Added NFSv4 support on AIX for Oracle and SAP ASE.

Bug number	Description
DLPX-71736	Dynamically disable RPC services if NFSv3 is no longer in use.
DLPX-71772	Network DSP Test between versions 5.3 and (6.0.3, 6.0.4) is fixed.
DLPX-71513	Replicate non-data objects like delphix engine users, authorizations, roles, permissions, policies and DB config templates.
DLPX-71305	Unable to load dummy recovery database dump due to SAP ASE error 15728.
DLPX-71172	Enabling SAP ASE dump history causes IllegalStateException in getDumpListFromLastRestoreDateAndFiles due to timestamp mismatch because of TZ.
DLPX-71178	SAP ASE internal error raised when dump history file is purged using sp_dump_history.
DLPX-65101	Fixed a race condition between a DB_DELETE job and the Oracle retention policy worker for the same container that can lead to a deadlock between the job and the worker.
DLPX-71918	Fixed an issue that causes the Delphix Engine UUID to change upon rebooting in IBM Cloud.
DLPX-71141	Fixed an issue where upgrading an Oracle dSource or changing the environment user in a linked Oracle dSource fails with the error "SOURCE_UPGRADE job for xxx failed due to an internal error during execution."
DLPX-71611	Updated UI time zone library to IANA 2020a.
DLPX-70349	Fixed a memory leak during incremental replication.
DLPX-72038	Fixed an issue that prevents 5.3.x - 6.0.x upgrade if a static route exists that goes over a DHCP interface.

Bug number	Description
DLPX-72148	Fixed a bug of always order hooks alphabetically rather than the running order set by users.
DLPX-71971	Allowed Enable/Disable of VDB if its current Timeflow has at least one snapshot.
DLPX-72115	Changed the Time Point field on a VDB back to reflecting the point on the parent the VDB was created from, but displayed in the timezone of the parent.
DLPX-71995	6.0.4.0 can no longer interact with 5.3.x remote Masking Engines.

3.2.108 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-70675	Fixed a bug that causes the system to become unresponsive after expanding multiple storage devices.

3.2.109 Release 6.0.4.2 changes

3.2.110 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-72155	Fixed an issue that can render a Delphix Engine unbootable if a reboot occurs after upgrade verification but before the upgrade is applied.

Bug number	Description
DLPX-71141	Fixed an issue where upgrading an Oracle dSource or changing the environment user in a linked Oracle dSource fails with the error "SOURCE_UPGRADE job for xxx failed due to an internal error during execution."

3.2.111 Release 6.0.4.1 changes

3.2.112 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-71930	Fix a bug that causes feature flags to be disabled when upgrading to 6.0.4.0.

3.2.113 Release 6.0.4.0 changes

3.2.114 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-68173	Resolved an issue where temporary database backup/device files created for cleaning up the target database were not being deleted.
DLPX-68773	The management stack runs out of memory as the environment monitor does not purge stale objects.
DLPX-69573	Allow linking an Oracle PDB with a lowercase name.

Bug number	Description
DLPX-69634	Allow provisioning an Oracle PDB with a lowercase name.
DLPX-69962	After detaching a PDB, perform unplug/plug, and attach again, if disabled is performed before SnapSync, the PDB can no longer be enabled.
DLPX-66045	Prevent Self-Service Container branches getting into an unusable state by blocking deleting the last segment of branches.
DLPX-77037	Snapsync performs an unnecessary checkpoint.
DLPX-68277	Users will see the detailed error message upon connection failure to Delphix connector during OS user validation and there will also be a "More" button with an error message which will open an error popup with all error details.
DLPX-70288	On the "Add Environment" screen when OS user validation will get fail, they will see the "More" button along with the error message. When the user clicks the button, an error popup opens with all details of the error and suggested action.
DLPX-70832	NFSv4 support for appdata sources running on AIX.
DLPX-68495	Fixed GUI reporting conflict information when creating a Retention Policy.
DLPX-70788	Added Environment User field for MSSQL sources in Datasets -> Configuration -> Source tab -> Staging Environment section.
DLPX-58047	Fixed bug where the sort sequence was incorrect. Fixed in Hook Operation Templates.
DLPX-67931	Provision against VPDB after create/drop a new tablespace failed with exception.oracle.targetscripts.rename.datafiles.
DLPX-59910	Comps.xml associated with Oracle Homes are marked as unparseable if they are longer than 65535 characters.

Bug number	Description
DLPX-55476	CLI provisioning fails when the mount point provided includes quotes around the path.
DLPX-71168	Changed type to text and spaced "Secret Key" and "Username Key".
ORB-3285	Support using `api.delphix.com` as a proxy for verifying the Cloud Agent binary's code signature certificate.
DLPX-71006	Allow provisioning across patch versions for Oracle versions on or after 18.X.
DLPX-71334	Migrate NTP configuration when upgrading between 5.3 and 6.0.
ORB-3286	Communication with Central management servers is now routed through the web proxy when one is configured for the Engine.
ORB-3117	Summary: Increase an action's failure message size to 256 characters so users can view large failure messages.

3.2.115 Release 6.0.3.1 changes

3.2.116 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-71339	Fixed an issue that can cause the Virtualization Management service to become inaccessible when the system memory became highly fragmented.

3.2.117 Release 6.0.3.0 changes

3.2.118 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-63192	More details will be displayed in the error message in case DB_SYNC fail due to missing SELECT permission on database 'msdb'.
DLPX-67708	Removed unnecessary Source Continuity source-archive file system.
DLPX-66878	A meaningful error message will be displayed in case the user is missing 'VIEW ANY DEFINITION' and 'VIEW SERVER STATE' permissions on AG Instance and linking is performed.
DLPX-68668	Fixed the issue when Environment discovery fails with Internal Error when Oracle DB instance name is > 15 characters.
DLPX-68539	Added support for Read-Only Oracle homes.
DLPX-62027	Fixed a bug that causes factory reset to fail when there are provisioned VDBs.
DLPX-67830	Eliminated a virtualization management service crash caused by egregious use of memory by environment monitoring.
DLPX-69067	Enabled NFSv4 support for older RedHat NFS clients.
DLPX-68491	Fixed an issue when SCAN Listener is not discovered for Oracle 19c Cluster Environment.
DLPX-68931	Improved replication throughput by parallelizing data streams.

DLPX-69350	Fixed an issue where in some cases a VDB's Time Point would not appear.
DLPX-62602	Prevented a full snapsync after detecting an incarnation change of and reverting to a previous incarnation of an Oracle database.
DLPX-69579	Resolved the issue of intermittent failure of DB_SYNC for source database full backups containing in-memory tables which were caused due to improper merging of filestream folders.
DLPX-69104	Fixed an issue when Environment Monitor task monitors replicated entities could lead to Out of Memory.
DLPX-58561	Increased online Redo Log size when using VDB Provisioning defaults from 50mb to 1024mb.
DLPX-68831	Storage is removed even when the drop database fails, causing ASE error 823.
DLPX-68323	Linking will not fail in case a slash is used as the path delimiter on the source database.
DLPX-69561	Allowed NoLogging Diagnosis to be shown and edited for Oracle CDBs.
DLPX-69625	Fixed an issue that causes the CLI to hang when deleting an object.
DLPX-65357	Source Environment selection in Attach dSource dialog is now alphabetically sorted.
DLPX-65215	Fixed an issue where Hotfixes aren't listed until after management service restarts.
DLPX-57988	The increased timeout of doShutdownOracleInstance.sh script from 20 seconds to 10 minutes.
DLPX-70018	Resolved the issue where during validated sync, fault "fault.mssql.source.next.backup.missing" was caused due to backupsets with similar first and last lsn.
DLPX-66671	dSource selection in dSource Linking Wizard is now alphabetically sorted.

DLPX-65723	MSSQL server cluster address is now editable through the Environments GUI.
DLPX-69514	Gracefully handle accelerated networking on Azure.
DLPX-68942	Implemented retries with some time delay in case of a failure while switching database user mode.
DLPX-56626	Some orcl_log_info entries have a very large and incorrect end_scn (281474976710655).
DLPX-67579	Deleted users' actions should be included in the action/audit log API results.
DLPX-69863	Enhanced instruction text relating to SSH when editing environment users.
DLPX-70081	Removed excessive debug logging for DSP connections which results in fast rollover of debug logs.
DLPX-66203	CLI / API calls to refresh/rewind vCDB directly should be disallowed.
DLPX-67194	RHEL 7.6 connector log shows Unidentifiable version string: RedHatEnterpriseServer 7.6.
DLPX-66754	When VDB is disabled, environment configuration can now be edited in the UI.
DLPX-62095	A wrong certificate is identified as an issuer of a self-signed certificate in rare cases.
DLPX-69243	Do not require an issuer to be present or keep the full chain intact on Truststore operations.
DLPX-59331	Permit non-CA certs in user Truststore.
DLPX-60779	Changed error message when there are no compatible installations on provisioning.

DLPX-69518	Provisioning failures due to BitLocker encryption will be identified and a proper error message will be displayed.
DLPX-64797	Fixing memory leak in hk2 library.
DLPX-70039	Password vault migration nullifies ASE linked source dump credentials.
DLPX-70089	Protection against a variant of billion laughs attack (XML entity expansion).
DLPX-64207	Added API support to revert from static to DHCP DNS settings.
DLPX-68857	Faults reported for Oracle Home missing where the Central Inventory does not show this Oracle Home present.
DLPX-64435	Exclusively specifying 'required' parameters to discover Oracle cluster via CLI results in an exception.
DLPX-69604	Alerts & Faults are reported for hosts in a namespace that can cause Out of Memory issue.
DLPX-39882	Prevented cloning of Tiimeflow storage for Oracle source continuity.
DLPX-67425	Resolved an issue when validated sync (with full/diff) restored multiple backupsets and a restore failed with a SQL server transient issue after a source continuity reset event resulting in a state where no operations could be performed on the dSource.
DLPX-70433	"DLPX_EXECUTE_SQL_CLEANUP_RETRY" will also print nested SQL error messages in case of command failure.
DLPX-68582	Customers now have access to an API to display the Engine License information.
DLPX-61335	Displayed in confirmation dialog the name of the user being deleted.
DLPX-70639	Resolved output buffer issue while identifying BitLocker encryption during provisioning on Win19.

DLPX-66259	Updated messages on the upgrade page when the operation fails.
DLPX-70782	Bumped up connector version for NET 4x installer as shipped OpenJDK version had been upgraded.

3.2.119 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-68995	Improved performance of dataset deletion.
DLPX-68997 DLPX-68999	Improved single connection replication throughput.
DLPX-70697 DLPX-70703	Addressed an issue that causes long periods of I/O unresponsiveness.
DLPX-69953	Fixed a bug that can cause a Windows iSCSI initiator to fail to connect to the Delphix Engine.
DLPX-70512	Fix a hang in the I/O subsystem that can cause the Delphix Engine to become unresponsive.

3.2.120 Release 6.0.2.1 changes

3.2.121 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-70065	Provisioning a VDB from a dSource or another VDB will fail if the following conditions are met: • Delphix Engine has at least one dSource and a VDB created using a Python plugin prior to the upgrade • Delphix Engine was upgraded to 6.0.2 • Provisioning was attempted from the UI after the completion of the upgrade
DLPX-69350	Fixed an issue that the time point attribute of a VDB is not shown.

3.2.122 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-69864	Fixed an issue that causes MSSQL operations to hang after the reception of an iSCSI LUN reset.

3.2.123 Release 6.0.2.0 changes

3.2.124 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-62806	Fixed an issue where provision against PDB after unplug/replug against the same linked PDB fails with exception.oracle.targetscripts.controlfile.create.
DLPX-67567	Oracle Source Continuity creates an unnecessary source-archive file system on zfs.
DLPX-27807	LogSync may fall behind when connected to an Oracle physical standby database in Real-Time Apply mode.
DLPX-68385	Customer provided Oracle Java missing in the search path for Java on hosts.
DLPX-62782	Reducing the number of nodes for RAC VDB and VDB in NOARCHIVELOG mode may result in ORA-00258 errors during VDB enable operation.
DLPX-62738	Better error message when plugins are uploaded out of sequence.
DLPX-68722	The product now recognizes VMware with BIOS date of 12/12/2018 as VMware 6.7.0u2.
DLPX-68579	SnapSync of Oracle 19c DB with encrypted tablespace fails with fatal exception "Block header 91 is not empty".
DLPX-68689	Fixed the issue where a huge number of error messages from ASE caused OutOfMemory Error.
DLPX-68957	Always On AG discovery will not fail in a multi-subnet environment.
DLPX-63088	Can now recover multiple Self-Service containers at the same time.

Bug number	Description
DLPX-47977	Improved handling of snapshot standby.
DLPX-64125	SnapSync failed with exception.oracle.dsource.sync.no_hosts.rac on RAC clusters with very long hostnames.
DLPX-62584	PDB enable failed after migration if mountBase has a trailing slash.
DLPX-68657	Virtualization can now fetch jobs from Masking engines configured with HTTP redirection.
DLPX-69121	It is no longer mandatory to have at least one enabled system administrator with local credentials.
DLPX-68167	Fixed an issue where too many requests were being sent for Faults from the Datasets pages.
DLPX-69082	Large stderr produced by failed rsync jobs are truncated to prevent Java OutOfMemory errors.
DLPX-58600	Datasets filter updated so that all items within a group that matches the filter string are displayed, even if the items contained in the group do not match the filter string.
DLPX-65896	VDB deletion failed due to the inability to delete LogSync worker.
DLPX-57903	Improved diagnosability of PDB discovery issues.
DLPX-68878	Fixed issue where start/stop buttons were not being displayed in the RAC instances configuration table.
DLPX-69271	Enabled replication smart failover by default.
DLPX-66715	The user-visible name for Oracle cluster objects is being replaced with the Oracle cluster name. For Windows clusters, the user-visible name is being replaced with the cluster address.

Bug number	Description
DLPX-68929	Changed default replication settings for better out of the box performance.
DLPX-68930	Improved replication throughput when sending multiple timeflows.
DLPX-69245	Fixed a memory leak that occurs when experiencing connectivity errors.
DLPX-69377	At least one non LDAP system user should be enabled when the LDAP server is being disabled.
DLPX-68575	LDAP principal fields were not being redacted in phone-home bundles.
DLPX-68528	Self Service Recover operation failed due to missing Timeflow.

3.2.125 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-66808	Re-introduced console splash screen with IP address and service states.

3.2.126 Release 6.0.1.1 changes

3.2.127 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-69203	Improved synchronous write performance over iSCSI.
DLPX-69167	Improved SQL Server data ingestion performance by leveraging asynchronous writes on underlying storage.
DLPX-69298	Eliminated possible data corruption on SQL server and vFiles over iSCSI that can occur when a Delphix Engine reboots.

3.2.128 Release 6.0.1.0 changes

3.2.129 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-60689	For SAP ASE, instead of using the DBCC CHECKALLOC command to fix DBID mismatch issue, the MOUNT command with FIXDBID and ALLOW_DBID_MISMATCH clauses will be used, to improve performance.
DLPX-65831	VDB snapshots need to clean unneeded ZFS datafiles.
DLPX-63949	Improved boot time after 5.3 to 6.0 migration by optimizing metadata indexing.
DLPX-66261	Upgrades to 6.0.0.0 will only be supported from a release greater than or equal to 5.3.6.0.

Bug number	Description
DLPX-66486	Snapshot of a linked database can end up with extra datafiles that do not belong to the database which might cause VDB on VDB provision to fail during rename of datafiles.
DLPX-66558	Cluster environment discovery was incomplete if the host locale was not English.
DLPX-66804	DB_LINK using incorrect user when RAC node also configured as a standalone environment.
DLPX-66768	vPDB save state lead to rollback or child provisioning failures.
DLPX-66823	Unable to link database with CL8MSWIN1251 charset.
DLPX-64538	Fixed a bug causing the timezone selector to only be visible when manually setting the time.
DLPX-66809	Removed the Windows Diagnostics Files and Directories on successful Diagnostics upload.
DLPX-67279	Provision failed when the source was from a RAC Oracle Standard Edition database and the target was Oracle Standard Edition.
DLPX-67451	Fixed an issue that sporadically caused replication to fail with an internal error.
DLPX-67454	Delphix Engine should select the highest version ojdbc driver available at ORACLE_HOME/ojdbc/lib.
DLPX-66077	Ensures child worker threads are gracefully exited when parent linked source sync job has completed/terminated.
DLPX-45983	MSSQL Validated sync will resume when storage usage falls below the threshold if storage threshold enforcement failed in the past.
DLPX-67560	Fixed an issue where MT provision may result in ORA-02058 due to un-purged or inflight 2PC transactions on dSource.

Bug number	Description
DLPX-67594	Old timeflows and snapshots are not getting removed by snapshot retention.
LX-2020	Report the correct amount of memory allocated to EC2 Nitro instances.
DLPX-67413	Fixed an issue where VDB point in time provisioning might fail if Oracle database environment is configured in a non-English locale.
DLPX-67684	PDB provisioning failed if the source had shutdown triggers.
DLPX-67575	Fixed failure during point in time 'Virtual to Physical' provisioning.
DLPX-67668	After setting the database online give it some extra time to startup completely, before doing any further operation on it.
DLPX-67759	Redact sensitive information from phone-home data.
DLPX-64638	Validated sync stops working if Delphix cannot connect to the backup server.
DLPX-65559	Even when the staging instance is down, attempt counter to detect backup files keeps on increasing and eventually, it stops detecting backups.
DLPX-56537	When a target host is used by a large number of dSources for staging or has a large number of objects, the performance of Delphix operations like validated sync, refresh, rewind, etc can be slow due to Powershell processes being serialized.
DLPX-67894	Removing cluster resource without removing its dependency can result in cluster failure. So, added retryer logic while fetching the resource dependencies (Get-ClusterResourceDependency) and ultimately fail the operation after all the retries.
DLPX-67813	Unsupported SQL server backup type gets picked while validated sync and the operation fails while looking for the backup. So, introduced a tunable filter to automatically skip SQL backups taken by backup software not supported by yet Delphix.

Bug number	Description
DLPX-67925	Added env host connectivity toolkit support for SLES on Power9.
DLPX-67934	Retries to fetch image identifiers during Netbackup restore if there is a mismatch between MSDB and Netbackup Master.
DLPX-67655	Fixed an issue where retention enforcement can generate user-visible errors while attempting to delete snapshots with dependencies after PDB migration to new CDB.
LX-1944	EBS NVMe devices can now be used in Delphix Engines.
DLPX-68022	Fixed an issue where hostchecker 'Check Oracle DB Instance' fails on HPUX and AIX.
DLPX-68124	PDBs with lower/mixed case names will not enable after an upgrade.
DLPX-68126	Fixed a bug that limits the number of disks that can be added in GCP.
DLPX-67421	Update the primary db file names in a transaction with the Timeflow creation to make sure whenever a Timeflow is created successfully we have its primary file information.
DLPX-67440	Skip VDBs having its current Timeflow as null from 'PrimaryDbFileAvailabilityCheck' as these VDBs doesn't undergo queisceing and are recoverable by refreshing them.
DLPX-61818	Linking wizard - Target Environment step - Privileged Credentials authenticates on the selected target now.
DLPX-68117	Some non-Admin users, lack all permissions, are unable to login to upgraded engine.
DLPX-67290	A wrong version input by user while manually adding a SQL Server instance, created issues in provisioning VDBs. SQL Server version will now be auto-discovered for manually added instances on adding or refreshing the environment.

Bug number	Description
DLPX-66238	Updated error message to let know user that non discovered CDBs are filtered out from the list when linking a detached source.
DLPX-68457	When a target host is used by a large number of dSources for staging or has a large number of objects, the performance of Delphix operations like validated sync, refresh, rewind, etc can be slow due to Powershell processes being serialized.
DLPX-68484	Fixed the issue where 'lstart' column value of sysusages table was beyond the range of Integer data type by taking the Long data type to store the lstart value.
DLPX-68500	Fixed an issue where the NTP service is not started following a reboot.
DLPX-68290	Support bundle generation can be time-consuming if the engine has a large number of snapshots to process.
DLPX-67792	Fixed issue in grids in which the selection checkbox was unclickable.
DLPX-67555	Provision vPDB/vCDB fails with ORA-45900 if the parameter enable_pluggable_database is omitted when specifying database parameters for new vCDB.

3.2.130 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-67782	Engines running 5.3 on EC2 i3 can now be migrated to 6.0.
DLPX-67961	Fixed an issue that prevents ssh access after switching to a static IP address.
DLPX-65948	Fixed a bug that could cause replication jobs to fail with internal errors

Bug number	Description
DLPX-68025	Improved boot time after 5.3 to 6.0 upgrade by reducing the overhead of setting ZFS properties.
DLPX-67868	Fixed a bug that can cause the management service to run out of memory when disabling the Splunk integration.

3.2.131 Release 6.0.0.0 changes

3.2.132 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-27433	The analytics GUI network graph shows newly added NIC information without requiring a management service restart.
DLPX-33998	If you add a hook script via the CLI, newlines are removed erroneously.
DLPX-40094	Correctly set the default type for the parameters to all operations in the CLI according to the container type.
DLPX-43215	Exclude sybsecurity from the list of auto-discovered databases.
DLPX-48712	Java 6 packages are no longer included in the product image.
DLPX-48280	When a user is set with the Provisioner role the 'provision' button does not appear, meaning anyone set with this role only is unable to provision VDBs.
DLPX-53996	The Delphix Engine does not provide instructions to browsers to avoid caching HTTP responses (pages).

Bug number	Description
DLPX-54740	Ensure Windows mount points are always unmounted as part VDB refreshes to prevent future VDB refreshes from failing due to "ERROR_ASSIGN_MOUNTPATH: failed to assign mount path for disk at="">>, error="">,>
DLPX-55282	In environments where the vPDB has been provisioned using a Delphix provisioned virtual CDB, shutting down the virtual PDB causes it to get into an incorrect "Cannot monitor" state, this has now been fixed to show the correct "Stopped" state.
DLPX-55598	Fixed an issue where vPDB refresh/rollback triggers spurious vCDB restart jobs, after vPDB+vCDB auto-restart.
DLPX-55829	Validated Sync can fail when monitoring ASE backup servers started by using the $DSLISTEN environment variable instead of the "-S" argument. This can be worked around by accessing $DSLISTEN in the RUN_xxxxx script and pass it down as -S.
DLPX-55958	VDBs with no snapshots failed to re-enable after a Delphix Engine upgrade, this has now been fixed.
DLPX-57454	Display underlying ssh error when environment host connections fail.
DLPX-58519	Enable Oracle LiveSource when LiveSource is in RESYNC_NEEDED state currently re-start Oracle Redo Apply. Oracle Redo Apply should not be restarted in this state.
DLPX-58760	Fixed a TCP port leak in the network throughput test feature.
DLPX-58845	Provisioning vFiles to the same host using different OS Environment Users no longer fails.
DLPX-59772	The API to list all snapshots consumes a significant amount of memory when there are more than 100,000 snapshots on the engine.
DLPX-60356	Fixed an issue where Oracle remote listener registration fails if set to empty string.
DLPX-60603	Network settings dialog now displays actual MTU value rather than a checkbox.

Bug number	Description
DLPX-60907	Fixed an issue where the Environment Monitor on Redhat 6.9 and 6.10 might throw unidentified version errors.
DLPX-60979	When user configures connection strings manually, these connect strings can end up connecting to incorrect PDBs/CDBs causing invalid snapshots. Verify that each connection to a PDB/CDB connects to the expected PDB/CDB.
DLPX-60993	Delphix backups create controlfile records; in rare circumstances, these records can cause invalid snapshots. To avoid this problem, remove Delphix backups control file records when using SCN-based SnapSyncs once a SnapSync completes successfully.
DLPX-62094	Allow certificates to expire after issuer certificate expiration.
DLPX-62241	Reduce SSH connections by temporarily preserving and reusing existing Delphix<->host connections where possible.
DLPX-62781	Spurious job event "DISCOVERED_TO_MANUAL_ORACLE_CLUSTER_NODES" no longer shows up for non-Oracle RAC environment refreshes.
DLPX-62892	In Oracle versions 18c and 19c, an Oracle bug can prevent the datafile headers from being updated for a standby database when managed recovery is running, resulting in failed SnapSync operations. Alert the user that an Oracle patch might be needed.
DLPX-62962	Removed unneeded EMPTY_RENEGOTIATION cipher
DLPX-62998	Fixed an issue where stale file mounts may be leftover when vPDB provision fails.
DLPX-63469	Initial setup now fails if the system was not provisioned with enough storage.
DLPX-63600	Network settings dialog now displays actual MTU value rather than a checkbox.
DLPX-64641	Fixed an issue where the last snapshot of a vPDB Timeflow can be deleted after the vPDB has been disabled, thus leaving the vPDB in a state with no provisionable snapshots.

Bug number	Description
DLPX-64711	Allow provisioning to complete when source CDB includes PDBs in a broken state.
DLPX-66020	Provision should remove files present in datafile filesystem that are not part of the database when provisioning a VDB from a VDB.
DLPX-67299	ASE environment discovery will not fail if there is a mismatch of "dataserver name argument" and value of "@@servername".

3.2.133 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-67753	Fixed an issue causing redirect responses to reveal server type and version when HTTP redirection is enabled.
DLPX-72068	Improved the way volumes are fetched while working on mounts.
DLPX-74396	Fixed an issue that occurred when manually adding a database to an environment which has the same unique name as a database in another environment managed by the same Delphix engine. Previously, Delphix reported an incorrect environment containing the same unique name.
DLPX-80172	Updated the Self-Service refresh warning message.
DLPX-80271	changeArchivelogMode now has an associated job event.
DLPX-80387	Fixed an issue where Oracle move-to-asm script would fail while dropping temp due to tempfiles being in use and unable to be dropped after the database was started.
DLPX-81184	For S3 object store, the "Base URL" input is renamed to "endpoint". The "region" input is now a dropdown for the user to select from a list of standard regions. The endpoint corresponding to that region will now be auto-populated.

Bug number	Description
DLPX-81692	Fixed an issue where Direct NFS was not being detected for Oracle 21.
DLPX-81996	Fixed an issue where an environment refresh after upgrade did not remove outdated/obsoleted toolkit components in 6.0.014.0 and 6.0.15.0.
DLPX-82075	Fixed an issue that prevented the creation of a network route whose gateway is reachable through multiple interfaces.
DLPX-82112	Added checks to prevent using NFSv4 with Direct NFS for some Oracle 19 versions that don't support v4 due to an Oracle bug.
DLPX-82236	Fixed an issue where Speculative Logging was being called out of context and could lead to unbounded consumption of rpool.
DLPX-82308	Fixed an issue where the provision/refresh of an Oracle Key Vault vPDB fails with, "ORA-28365: wallet is not open".
DLPX-82329	Action-based alerts now include the success or failure state of the action, including the reason in case of failure.
DLPX-82334	Enabled the creation of on-link network routes; routes whose destination are directly reachable without a gateway.
DLPX-82381	Improved Replication performance on engines with a lot of objects.
DLPX-82392	Fixed an issue where after editing credential environment variables, the create/provision VDB wizard would fail because the environment variables were missing the password field in the payload. The required password field has now been added.

3.2.134 Release 6.0.15.0 changes

3.2.135 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-80760	Increased the inotify limit to address a defect during upgrade.

3.2.136 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-68240	Fixed an issue where LogSync for an Oracle RAC standby does not set max number of backup tasks correctly.
DLPX-73375	Added granularity in restore job events to mark various phases of Restore Backup process. Added events in Restore Backup job to provide information at the start of the Redo phase of a restore command.
DLPX-75677	Fixed an issue causing auto population of the encryption key when no input is given.
DLPX-78913	Fixed a minor typing issue when deleting a MSSQL database.
DLPX-79228	Fixed a VDB Start Fault after a VDB start job is successful.
DLPX-79528	Made improvements in environment monitoring to return the NO_DELPHIX_DATABASE status for Staging Push dSources when they are not managed by Delphix, and avoids monitoring other attributes for such databases.

Bug number	Description
DLPX-79596	Made a change in the enable flow for Staging Push dSources to always make an attempt at unmounting the DATA storage before dropping the staging database.
DLPX-79780	Mount with local_lock=all on Linux, when mounting VDBs with NFSv3.
DLPX-79999	Use the offset in the time_zone column of msdb.dbo.backupset to convert the timestamp correctly for a backup from a source, and then continue using the staging host timezone to display the time of a snapshot on the UI.
DLPX-80206	Updated MD5 checksums for Oracle 12.1.0.2.0 OJDBC jars.
DLPX-80254	Fixed an issue where Oracle JDBC jar checksum checks are being reported as false positives despite underlying database connection issues.
DLPX-80406	The Spring framework has been updated due to the Spring4Shell vulnerability.
DLPX-80487	Fixed an issue where the Delphix UI would sometimes not render, showing waiting for response.
DLPX-80494	Added an action item to check for support Host and Server Type combination while adding MSSQL environments.
DLPX-80619	This change will introduce the ability to add sporadic failures via tunable: ADDITIONAL_SPORADIC_FAILURES.
DLPX-80909	Cross-Site Scripting (Reflected) in /resources/json/delphix/session.
DLPX-81048	Removed the requirement for Linux kernel recover-lost-locks setting when using NFSv4 with Oracle dNFS.
DLPX-81090	Can now enable/disable SNMPv3 vs. v1/v2.

Bug number	Description
DLPX-81100	Updated the command for checking database files accessibility while fetching the VDBs status with a lighter and less intrusive command, to get relief from VDBs being stopped randomly.
DLPX-81242	Fixed an issue preventing the upgrade of a replication Continuous Vault source with automatic replication.
DLPX-81308	Fixed an issue causing Appdata SnapSync to crash with NullPointerException when a virtual database is not successfully refreshed or rollbacked.
DLPX-81358	Fixed the unnecessary alerts of timezone discovery failure that users were facing randomly for the cluster environments.
DLPX-81502	Improved performance of the Replication page.
DLPX-81696	Users should now be able to enable the feature flag AZURE_DATA_BANK.
DLPX-81710	Fixed an issue where an engine could not be setup when objectStorage is enabled.

3.2.137 Release 6.0.14.0 changes

3.2.138 Security fixes

Bug number	Bug introduced in	Description	Security Bulletin
DLPX-81059	5.2.2.0	Arbitrary Code Execution may be performed when configuring masking environments	TB098[67]

67 https://support.delphix.com/Support_Policies_and_Technical_Bulletins/Technical_Bulletins/TB098_Arbitrary_Code_Execution_May_Be_Performed_When_Configuring_Masking_Environments

3.2.139 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-38908	Oracle LogSync should automatically resolve faults for transient issues
DLPX-39193	Null Pointer Exception in Oracle LogSync backup stream handler
DLPX-57078	Job cancel requests during Oracle provisioning are not processed until the end of recovery
DLPX-57405	move-to-asm.sh does not support TDE-enabled databases
DLPX-62343	disable the OPTIONS method for all HTTP(S) requests
DLPX-64386	Oracle LogSync thread may hang when trying to remove temporary RMAN command file from source host toolkit directory
DLPX-65413	Ensure "source-archive" directory is unmounted at start of Oracle Provision
DLPX-66879	Oracle LogSync can create orphaned logs in certain scenarios
DLPX-69453	Provide tunable for Oracle LogSync client timeout
DLPX-69802	Common Toolkit directory is not removed from a mounted shared NFS location when the environment is deleted
DLPX-76382	Force disable should succeed despite environmental problems

Bug number	Description
DLPX-77840	Fixed an issue on the Setup pane to allow successful completion of an smtp test against a specific email address
DLPX-78412	SCM/Talaria failure reason should be communicated in the Delphix fault warnings
DLPX-78726	Removing windows environment performs cleanup of iSCSI persistent login target
DLPX-78754	Disable operation is prohibited on replicated sources
DLPX-78986	Prevent DSP connections for disabled Oracle RAC cluster nodes
DLPX-79077	Resolved an issue of an infinite spinner when validating BEQ credentials for Oracle dSource with duplicate unique_name. While linking a dSource, credentials are now required when discovering an unknown CDB.
DLPX-79242	Splunk HEC token logged to debug logs during splunkHec test
DLPX-79396	Allow users to unset Oracle database user name and credentials through CLI if Simplified Connection Management is enabled.
DLPX-79502	SnapSync fails if more than 1000 tempfiles exist in the whole CDB
DLPX-79591	Changed the NFSv4 minimum supported target Redhat version to 6.4 (was previously 6.3).
DLPX-79742	Unable to provision PostgreSQL VDB to Linux host with processor type of ppc64le
DLPX-79823	Improved the action item for failure to enable/attach the staging push dSource.

Bug number	Description
DLPX-79942	Improved the action item for failure to enable/attach the staging push dSource.
DLPX-80137	Limit SNMP configuration access to sysadmin
DLPX-80144	Oracle SnapSync crashes with NullPointerException when a PDB dSource is renamed and replaced with a new PDB of the same name
DLPX-80217	Fixed issue with filename conflicts during source backup restore
DLPX-80302	It was necessary to restart the auxiliary CDB during a TDE provision after recreating the autologin keystore
DLPX-80369	Oracle environment monitor triggers fault.oracle.db.connection.failed fault immediately on dataset stop or disable.
DLPX-80415	Fixes long delay in operations such as VDB start/stop when JDBC Thin connection to database fails due to unable to establish network connection.
DLPX-80439	Provide mount location to upgrade scripts during Lua to Python upgrade process if mount specification has not been provided.
DLPX-80440	Fixed spinner issue while provisioning a VDB from a SQL Server staging dSource from Provision VDB option in datasets menu
DLPX-80482	TDE-enabled provisions to a RAC target fail with "ORA-28365: wallet is not open" while attempting to reopen the database in start_database.sh in the auxiliary
DLPX-80483	Fix failing TDE-enabled vPDB provisions to a linked RAC container database due to "ORA-28365: wallet is not open" errors
DLPX-80487	Improved performance across dataset and replication related pages.

3.2.140 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-80078	The issue with removing files with complex file permissions on EBS is now fixed

3.2.141 Release 6.0.13.1 changes

3.2.142 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-80818	libc upgrade necessitates PostgreSQL re-index.

3.2.143 Release 6.0.13.0 changes

3.2.144 Security fixes

Bug number	Bug introduced in	Description	Security Bulletin
DLPX-79789	5.3.0.0	Arbitrary Code Execution May Be Performed by Engine System Administrators.	TB096[68]

68 https://support.delphix.com/Support_Policies_and_Technical_Bulletins/Technical_Bulletins/TB096_Arbitrary_Code_Execution_May_Be_Performed_by_Engine_System_Administrators

3.2.145 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-7868	The issue with the confusing vPDB error on a snapshot after resetlogs of a linked CDB is now fixed.
DLPX-41671	You can now update the Oracle cluster home through the user interface.
DLPX-43467	When dSource is an Oracle standby in RTA mode, LogSync was raising the fault.oracle.linkedsource.log.conflict error and getting disabled on its own. This issue is now fixed.
DLPX-50309	Users can now change logSyncInterval for Oracle dsources.
DLPX-59757	The count for masking jobs fetched from the Masking Engine is now configurable. By default, it is set to 500.
DLPX-67604	The manual recovery of a database after V2P from a snapshot of dSource was failing with an error. This issue is now fixed.
DLPX-68684	The self-signed certificate is now compliant with the requirements for trusted certificates in MacOS 10.15.
DLPX-74613	Oracle VDB migration check needs to be done against the Oracle target host instead of the source. Furthermore, the Error and Action plan provided should include the target hostname.
DLPX-75467	The CLI now returns correct and descriptive error messages when executing unauthorized requests on uninitialized engines.
DLPX-75646	To diagnose BEQ connection failure, this release adds MD5 checksums for ojdbc*.jar for Oracle release versions up to currently supported release version.
DLPX-75878	The issue with the JDBC connection string for an Oracle vPDB not getting updated after an IP address change is now fixed.

Bug number	Description
DLPX-75989	The issue with the failure of environment discovery of an Oracle Cluster with a NullPointerException error is now fixed.
DLPX-76956	Previously, the Oracle JDBC test connection with the wrong password was increasing the `LCOUNT` value by more than 1. This issue is now fixed.
DLPX-77140	This release now speeds up metadata that is sent during replication when Extended retention is involved.
DLPX-77231	Previously, when source discontinuity on the dSource was followed by resync on the livesource, one or more livesource workers were failing to start. This prevented livesource status from getting updated and the first snapshot from being taken after resync. This issue is now fixed.
DLPX-77600	This release fixes NPE when Linking dSource with missing backup and unreachable nodes.
DLPX-77880	This release improves scalability for engines with an extremely large number of snapshots that were causing them to run out of memory.
DLPX-78015	Previously, V2P export with absolute data files was failing with an internal error. This issue is now fixed.
DLPX-78174	Insecure DES is no longer supported for SNMPv3.
DLPX-78420	This release adds V2P support for Windows server 2022 host machines.
DLPX-78473	This release improves load times for Datasets and Dataset Performance pages for engines with a large number of datasets and containers.
DLPX-78488	Previously, when switching from the backup server to ASE, dump history was not working for dSources configured to use the remote backup server. This issue is now fixed.
DLPX-78594	This release fixes an issue with disabled VDBs not being able to undo a refresh.

Bug number	Description
DLPX-78688	TLS 1.0 and TLS 1.1 ciphers are no longer available. Any system that is only configured with TLS 1.0 or TLS 1.1 ciphers is switched to use the default cipher set.
DLPX-78693	Invalid sync parameters will not cause DE server unavailability.
DLPX-78696	Switching sync strategy from source sync strategy type to sourceless strategy type is not allowed.
DLPX-79126	VDBs can now be automatically started with the tunables if stopped intermittently because Windows fail to write on the mount (Msg 9001).
DLPX-79292	This release eases restrictions on taking a snapshot of PDBs with encrypted UNDO tablespaces.
DLPX-79344	Previously, Snapsync of a standby PDB in mount mode was failing with the `ORA-01109: database not open` error message. This issue is now fixed.
DLPX-79422	Previously, clicking on a Replication Profile was resulting in the following error message `An error happened while communicating with the server`. This issue is now fixed.
DLPX-79789	Under certain conditions, arbitrary code execution may be performed by sysadmins.
DLPX-79808	This release fixes failures if the VDB name is more than 68 characters.

3.2.146 Release 6.0.12.1 changes

3.2.147 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-79151	The issue with remote syslog configurations preventing engine upgrades or virtualization service restarts is now fixed.

3.2.148 Release 6.0.12.0 changes

3.2.149 Log4j updates

Based on detailed testing and analysis, all the currently supported products are not susceptible to known log4j vulnerabilities. Please refer to TB095 Technical Bulletin[69] for more information. All instances of log4j in currently supported Delphix products are updated to **log4j 2.17.1** as of this release.

Delphix keeps you updated on the latest developments and keeps releasing hotfixes, procedures, and workarounds for such critical vulnerabilities. For more information on how Delphix supports our product and customers in such cases, see Delphix Product Security (see page 945)

For more information, refer to the following pages:

- TB095 log4j vulnerabilities[70]
- Uninstalling the delphix connector service from the target database servers (see page 2102)
- Delphix product lifecycle policies[71]
- Product security (see page 945)

69 https://support.delphix.com/Support_Policies_and_Technical_Bulletins/Technical_Bulletins/TB095_log4j_Vulnerabilities_(CVE-2021-44228%2C_CVE-2021-45046%2C_CVE-2021-45105%2C_CVE-2019-17571%2C_CVE-2021-4104)

70 https://support.delphix.com/Support_Policies_and_Technical_Bulletins/Technical_Bulletins/TB095_log4j_Vulnerabilities_(CVE-2021-44228%2C_CVE-2021-45046%2C_CVE-2021-45105%2C_CVE-2019-17571%2C_CVE-2021-4104)

71 https://support.delphix.com/Support_Policies_and_Technical_Bulletins/Support_Policies/Product_Lifecycle_Policies_(KBA1003)

3.2.150 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-23068	Validation for target database parameter 'DB_FILES' for single-tenant databases for the following operations: provisioning, refresh, rewind, and converting a dSource to LiveSource is now added. Furthermore, specific error messages for handling ORA-00059 errors are added.
DLPX-44544	The issue with the SnapSync of an Oracle standby dSource in Real-Time Apply mode calculating the snapshot's timestamp incorrectly is now fixed. This issue was resulting in ORA-01194 or ORA-01152 errors when provisioning to a timestamp after the snapshot.
DLPX-56691	The issue with the data files of a VDB getting unmounted when provisioning, refresh, or rollback job is canceled manually is now fixed.
DLPX-57971	The issue with the latest snapshot of a LiveSource taking a long time to show the SCN/ timestamp range on its card in the GUI is now fixed.
DLPX-60320	UI now allows the selection of older dataset repositories (downgrade) in addition to selecting newer ones (upgrade).
DLPX-67069	The issue with the stopped Oracle VDB monitoring by the environment monitor that resulted in connection errors flooding the debug log is now fixed.
DLPX-68132	The issue with the “Copy query to clipboard” SQL copy functionality in the “Managed Source Data” is now fixed to have correct apostrophe characters.
DLPX-72123	The issue with the failure of detaching or deleting an Oracle dSource operation on RAC environments (This issue was occurring due to failure of deletion of RMAN backups on RAC and the operation needs to be retried with a force option) is now fixed.
DLPX-72779	The UI showing enabled or disabled for cluster nodes now uses a grid table. The Enabled column contains a checkmark to show whether the cluster is enabled or disabled. Users are able to select or unselect the checkmark to enable or disable a cluster.
DLPX-73975	Critical storage faults should not be ignorable nor manually resolvable

Bug number	Description
DLPX-74862	This release fixes an issue where the RESUME operation failed without any error thrown to the user on dSources where ENFORCE was still in progress. The fix will make sure that even if RESUME fails, it throws a DUE to the user with suggested actions to resolve the issue.
DLPX-75763	The issue with the failure of refreshing a VDB provisioned as an empty vfiles since there is no parent container to refresh from is now fixed.
DLPX-76266	The issue with the VDB Disable operation that resulted in an error message while connectivity with the host machine can't be established is now fixed.
DLPX-77123	You can now run Upgrade Verify when another upgrade is in progress.
DLPX-77347	This release fixes the difference in time shown for MSSQL snapshot based on different database authentication methods.
DLPX-77638	The issue with the failure of the End Entity Certificate expiration fault is now fixed.
DLPX-77664	The issue with the failure of the Oracle SnapSync with an error message, "RMAN-06183: datafile or datafile copy (file number) larger than MAXSETSIZE" if a datafile resized in the middle of SnapSync is now fixed.
DLPX-77913	The issue with the faults table missing data if there was more than one page of faults is now fixed.
DLPX-77925	The issue with the unsupported Windows release error message is now fixed.
DLPX-78113	MSSQL VDB database size will now be refreshed periodically based on environment_monitor.dynamic_attributes_check_period tunable.
DLPX-78183	The issue with the MSSQL validated sync schedule not getting updated without successful backup restoration is now fixed.
DLPX-78244	The issue with the failure of a few operations on self-service containers due to incorrect entries corresponding to the Oracle log metadata on the Delphix engine is now fixed.

Bug number	Description
DLPX-78258	The issue with the input bug that retained cleared out DB credentials in the dSource linking wizard is now fixed.
DLPX-78263	The issue with the failure of a SnapSync of an Oracle standby dSource in Real-Time Apply mode with an error message, "exception.oracle.snl.linkedsource.current_scn.invalid" if the rate of change in the database is low is now fixed.
DLPX-78265	Offline Oracle bystander PDBs data files can now be optimized leading to improved provision performance.
DLPX-78309	The issue with the CLI being unable to log in to system users when the main virtualization service is down is now fixed.
DLPX-78334	The issue with a large number of missing Oracle archive logs causing an error while viewing dataset is now fixed.
DLPX-78392	Hosts running Windows Server 2022 can now be added as Source and Target environments to the Delphix Engine.
DLPX-78522	SSLv3, TLS 1.0, and TLS 1.1 are no longer configurations options for HTTPS. Any system configured only with these removed options will be automatically set to use TLS 1.2.
DLPX-78791	This release upgrades log4j from 1.2.17 to the latest 2.x in Windows Connector.
DLPX-78938	This release upgrades log4j in virtualization to 2.17.1.

3.2.151 Release 6.0.11.0 changes

3.2.152 Security fixes

Bug number	Bug introduced in	Description	Security bulletin
DLPX-77921	6.0.8.0	Arbitrary Code Execution by Delphix System Administrators may be Performed on Virtualization and Masking Engines	TB094[72]

3.2.153 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-53019	The issue with the missing redo alert raised during the environment monitor check has now been resolved.
DLPX-59299	Discovery and monitoring rely on "Connected" in sqlplus output, which may not be the case if NLS_LANG is set to another language (e.g. Japanese). Downstream operations, like linking or provisioning, may then fail due to missing user privileges.
DLPX-59662	The issue with copy-Only Backups failure with Virtual Service Accounts has now been resolved.
DLPX-62706	The issue with the Hostchecker not properly checking /home/delphix permissions has now been resolved.
DLPX-64082	The issue with Oracle Provisioning scripts having hard-coded timeout issues has now been resolved.

72 https://support.delphix.com/Support_Policies_and_Technical_Bulletins/Technical_Bulletins/TB094_Arbitrary_Code_Execution_by_Delphix_System_Administrators_May_Be_Performed_on_Virtualization_and_Masking_Engines

Bug number	Description
DLPX-65729	Added retry functionality to the 'read backup files' operation during a validated sync to an account for an unstable environment.
DLPX-69778	SAML response is not logged on successful SSO login.
DLPX-72043	The issue where LiteSpeed `xp_restore_headeronly` stored procedure failure message are displayed when validated sync is active for dSources with LiteSpeed backup has now been resolved.
DLPX-72220	A UI issue that occurred while updating the vault when only the private key is changed has now been resolved.
DLPX-72225	Admin user created from management UI is no longer showing as 'non-admin' type.
DLPX-72237	The 'Verify Credentials' button from the DSP Throughput test page is now removed.
DLPX-72369	The dependency on a parent snapshot relying on the latest snapshot is now removed if a parent snapshot does not exist during the VDB enable operation.
DLPX-72778	Oracle dSource attach operation with changed DB ID using 'Force' option is now allowed.
DLPX-74555	Updated the "no Delphix connector" message while provisioning a Windows source environment.
DLPX-74676	Oracle LiveSource LogSync should only catalog valid archive log files.
DLPX-74851	In the Add Environment GUI, the mouseover information for "Set Delphix Session Protocol Options (DSP)" has been currented.
DLPX-74896	The race condition issue when running Oracle VDB refresh and dSource snap sync resulting in incorrect engine metadata entry for the parent snapshot of a VDB in the `dlpx_timeflow` table has now been resolved.

Bug number	Description
DLPX-75335	Added a product name and product version for the Delphix Connector executable so this information can be available before installation.
DLPX-75500	For ag cluster nodes, if the refresh fails due to timezone discovery failure, don't delete the nodes from engine metadata as it doesn't mean we had an issue with the nodes.
DLPX-75952	Database configs will be replicated only if the associated VDB is replicated.
DLPX-75995	The issue causing environment 'Add' or 'Refresh' to fail when PowerShell Transcription is enabled has now been resolved.
DLPX-76244	The issue where TCP fallback connection to database stops responding if the Oracle database instance is down has now been resolved.
DLPX-76290	Databases of UNKNOWN cdb type are now included in the attachment of a non-PDB container.
DLPX-76731	Added Delphix support for `WALLET_ROOT` and `TDE_CONFIGURATION` parameters to manage wallets in 19c instead of sqlnet.ora.
DLPX-76759	Added "Response" to faults along with other details when logged in the Admin App.
DLPX-76777	Remove orphaned Oracle logs resulting from archive log fetch timeouts.
DLPX-76793	Added execution timeout for execution of `UpdateFileACL.ps1`.
DLPX-76974	The issue where a user was unable to change 'from address' of SMTP server to noreply@delphix.com[73] in the GUI has now been resolved.
DLPX-77112	The issue where an Oracle VDB cannot be provisioned between different minor versions if the Source is on higher RU has now been resolved.

73 mailto:noreply@delphix.com

Bug number	Description
DLPX-77284	The issue where after a hotfix was removed due to a successful upgrade, the system would still indicate the hotfix was installed post-upgrade has now been resolved.
DLPX-77345	The issue where provisioning a vVDB fails with `java.lang.OutOfMemoryError` when sqlplus is used to rename the datafiles has now been resolved.
DLPX-77405	Replicated password vaults will no longer be visible in the UI.
DLPX-77676	The issue where provisioning a vPDB from a PDB dSource fails with "ORA-65114: space usage in container is too high" if PDB `max_size/max_pdb_storage` is configured has now been resolved.
DLPX-77708	The issue where a refresh/disable/destroy of a VDB using NFSv3 could cause loss of access to other VDBs that were using NFSv3 has now been resolved.
DLPX-77844	The issue where V2P operations from a VDB snapshot would result in the deletion of any production datafiles that exist on specified V2P target directory has now been resolved.
DLPX-77904	Removed 'Factory Reset' for Delphix Engines that are Data Vaults, as the operation is disabled for those engines.
DLPX-77912	The issue that can cause a VDB stop, refresh, or rollback to fail with an internal error has been resolved.
ORB-2465	Removed the requirement that SAML SSO email addresses must match case-sensitively for SSO logins.

3.2.154 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-77577	Increased nvme I/O timeout to prevent storage issues in EC2 (Activated after optional Reboot).

3.2.155 Release 6.0.10.1 changes

3.2.156 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-77467	Loading the setup app dashboard (as sysadmin) was rendering a server error popup with instruction to contact Delphix Support. This 6.0.10.0 error has been known to impair the ability to configure web proxy, PhoneHome, SMTP servers, and other connectivity settings via the GUI. It has now been resolved.

3.2.157 Release 6.0.10.0 changes

3.2.158 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-18438	The issue with provisioning to the latest available time that resulted in generating the `exception.oracle.target.point.not_provisionable` exception has now been resolved.
DLPX-35480	Previously, static routes were being added using the add command. Now, the same can be added using the create command.
DLPX-57516	The issue with the failure of management service to start after configuring some abbreviated timezones from a picklist in server setup or sysadmin CLI has now been resolved.
DLPX-58133	Previously, the Oracle SnapSync operation was resulting in a warning for BCT usage on editions that do not provide it. This issue has now been resolved.
DLPX-58675	The issue with the deletion of the last snapshot on timeflow by Retention during a failed Oracle DB_SYNC operation has now been resolved.

Bug number	Description
DLPX-63003	The issue with memory being exhausted while reading too many snapshots from engine metadata has now been resolved.
DLPX-63347	If the staging source has the "Use as Staging" flag set as off, the user was seeing a specific exception while trying to enable a linked dSource to point in the direction of what needs to be done. Any compatibility failure will now have a specific exception.
DLPX-63601	Previously, querying the following operation "backupset table" and "whether a database is part of AG or not" was resulting in deadlocks and lock timeouts errors. We have now added retries to resolve the issue.
DLPX-64369	The issue with throwing `fault.oracle.linkedsource.incomplete.tempfile` for physical standby has now been resolved.
DLPX-65949	The issue with misleading status in the progress bar while taking a copy-only backup has now been resolved.
DLPX-69831	Previously, the Oracle dSource SnapSync operation was not displaying a clear failure message if a dSource `db_unique_name` is changed. This issue has now been resolved.
DLPX-70317	The issue with the restarting of the NFS-server by the reaper thread while deleting a vPDB from a linked CDB with Talaria turned on has now been resolved.
DLPX-71018	The issue with UI displaying only the suffix of the device name used by Hyper-V has now been resolved. UI now displays a unique device name for storage in Hyper-V.
DLPX-71292	The issue with the allowance of incremental SnapSync after LogSync throws `fault.oracle.linkedsource.log.conflict` has now been resolved.
DLPX-71639	The NFSv4 is now set as the default option when mounting datasets from OS platforms that support it.
DLPX-71769	The need to set permissions of `$ORACLE_HOME/dbs` subdirectories using STARTUP SPFILE syntax is now removed.
DLPX-72011	The issue with the CLI network setup not configuring the first network interface when multiple interfaces exist has now been resolved.

Bug number	Description
DLPX-72186	The issue with CDB log file retention working incorrectly if a PDB has multiple time flows pointing to the same CDB timeflow has now been resolved.
DLPX-72432	The format of `zpool_iostat_60.log` has been enhanced in this release. A timestamp is recorded for each sample in the log, making it easier to determine the time for each sample.
DLPX-72956	The issue with disabling the Oracle LogSync after running the validated sync job has now been resolved.
DLPX-73575	The timezone monitoring is now added for the Windows hosts.
DLPX-73590	You can now refresh a VDB whose parent dataset is in a different group without needing authorization on the parent or its group.
DLPX-73800	The issue with the failure of olsnodes when run as a non-Oracle user has now been resolved.
DLPX-74504	The issue with throwing a new DUE and NotFoundException when ojdbc libs cannot be read has now been resolved.
DLPX-74945	UI now displays a detailed error message for transaction log-chain break fault.
DLPX-74992	The issue with the failure of SnapSync operation when Database incarnation reset-logs end time is changed from "2021-03-13 22:03:07.0" to "2021-03-13 21:03:07.0" for virtual pluggable database “ " has="" now="" been="">">
DLPX-75389	The issue with the recording of the insufficient details by Logsync when dbid change was detected has now been resolved.
DLPX-75517	The issue with the failure of the Oracle vPDB provisioning with the "ORA-00959: tablespace 'TEMP' does not exist" error has now been resolved.
DLPX-75721	The issue with the failure of an Environment discovery with the "DelphixFatalException: Unknown Oracle Database status: REFRESHING" error has now been resolved.

Bug number	Description
DLPX-75737	The issue with saving unnecessary logs by Retention if bookmark falls exactly on a snapshot end SCN or snapshot end timestamp has now been resolved.
DLPX-75897	Previously, failure to start I/O services after the upgrade operation was resulting in a stack restart loop. This issue has now been resolved.
DLPX-75951	The internal error being signaled during VDB SnapSync by `removeUnneededZFSFiles` when a data file is physically removed during processing by an external cause has now been resolved.
DLPX-76288	The NFS latency for workloads involving a lot of parallel I/O (e.g. Oracle VDBs with concurrent accesses to many data files) is now improved.
DLPX-76388	Previously, entering key pairs directly into hook environment variables, as opposed to via a vault or as passwords were resulting in an internal error. This issue has now been resolved.
DLPX-76406	The issue with NFS-based VDBs becoming unresponsive has now been resolved.
DLPX-76447	The issue with the V2P Functionality to customize target directory structure for exporting database files to separate file systems not working as documented has now been resolved.
DLPX-76613	The issue with the unnecessary accumulation of heap when validated sync is active for dSources using an environment user that eventually can cause out of memory issues has now been resolved.
DLPX-76690	The issue with the removal of extraneous Oracle data files while creating snapshots has now been resolved. The extraneous Oracle data files are now removed during the provisioning operation.
DLPX-76692	For V2P operation, we now use the unbuffered copy method for better performance.
DLPX-76718	The issue with the creation of the extraneous Self-Service branch segments during the replication operation has now been resolved.
DLPX-76760	Environment clusters will now show faults from their child nodes.

Bug number	Description
DLPX-76802	Previously, the engines that are in the DEFERRED upgrade state were resulting in the "Large Receive Offload" option turned off which was leading to performance degradation in network transfers. This issue has now been resolved. Upgrading the engines that are in the DEFERRED state will also resolve the issue. Screen reader support is now enabled.
DLPX-76891	CRON expression labels now ask for Quartz format on the user interface.

3.2.159 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-76619	The NFS reliance on DNS to prevent VDB unresponsiveness related to DNS unreliability is now reduced.
DLPX-76203	The NFS latency for engines with many Oracle dNFS clients is now improved.
DLPX-76119	The issue with the Delphix Engine crashing or becoming unresponsive when canceling a replication job has now been resolved.
DLPX-76991	Optimized in-memory cache eviction by making minor improvements to I/O performance.

3.2.160 Release 6.0.9.0 changes

3.2.161 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-28435	MS SQL instances PatchLevel will be displayed in preference to the version on the UI.
DLPX-40005	Improved the error message that is displayed when a JDBC connection cannot be established or the Oracle database Instance is unavailable.
DLPX-48080	Oracle Home Check may generate spurious faults if Oracle Home entry does not exist in /etc/oratab.
DLPX-55951	Attempting to provision a plugin-based VDB onto an incompatible OS (Windows to Linux or vice versa) is possible in the UI, and would fail with a crash requiring a restart. Now an informative error message is shown after the attempt is made.
DLPX-59613	Fixed creation of retention policies workers on replica objects after failover.
DLPX-64307	Environment refresh should ignore cluster discovery for Oracle VDBs.
DLPX-66191	Fixed the side-effects of the native Windows "Recent Files" behavior, when large numbers of PowerShell operations are being run concurrently.
DLPX-67537	Domain administrators can now create, view, and edit the alert profiles of other domain users.
DLPX-69605	Poor error message when selecting Timeflow range.
DLPX-70502	On detaching a dSource, delete backup server entry from engine metadata and related ones if the backup server is unused.

Bug number	Description
DLPX-71002	In case we have null values coming for recovery_model from msdb.dbo.backupset table, the user will see a generic exception for manual sync and a fault for validated sync.
DLPX-71908	Users will now see a warning when they remove any object from the replication specification list.
DLPX-72012	Prevents the same IP address from being configured on more than one network interface.
DLPX-72411	Environment names for Windows and Oracle Clusters are once again editable by users.
DLPX-72609	When we do MSSQL standalone environment discovery, the user will see a warning for databases attached to AG that are present and will not be discovered unless cluster environment discovery is selected.
DLPX-72695	Improved Oracle SnapSync performance by eliminating unnecessary calls to getTotalHoleBlocks.
DLPX-73409	Duplicate listener entry gets generated in engine metadata if Oracle listener is manually started with a non-uppercase name.
DLPX-73586	Fixed a display error of some snapshot names in the command line interface which showed references instead of actual names.
DLPX-73720	Provisioning an Oracle vPDB fails with "ORA-65149: PDB name conflicts with existing service name in the CDB or the PDB" if the PDB and CDB names are the same.
DLPX-74050	Environment names for Windows and Oracle Clusters are once again editable by users.
DLPX-74078	The Target Directory path is combined with other directories such as Data Directory, Archive Directory, Temp Directory, etc to build the full path for data files, archive logs, temp files, etc. As long as the combined paths are valid the V2P job proceeds.
DLPX-74197	Fixes a misleading warning about insufficient space on a replication target.

Bug number	Description
DLPX-74201	Snapshots created as a result of refresh or rewind operations will now be labeled as just "Snapshot" to avoid confusion. Users are advised to look at the Timeflow markers to know when the Timeflow operation was performed.
DLPX-74367	Delphix Engine repeatedly reports "Failed to parse logfile".
DLPX-74377	Improved diagnostics information for the case when Delphix Engine fails to connect to the Windows host.
DLPX-74387	fix an issue that causes the management service to crash under heavy CPU load.
DLPX-74398	Added handling of dangling nodes during Windows cluster environment add and refresh operations.
DLPX-74486	Delphix OS users cannot provision 12.2 TDE vPDB due to directory permissions in the default wallet location.
DLPX-74495	Enabled more logging in Delphix connector logs for timeouts.
DLPX-74681	During RAC vPDB provision, Oracle 19.9 target CDB crashes with ORA-00600 [krccfl_chunk] when BCT is enabled.
DLPX-74806	Improved error message displayed when the storage device initialization fails unexpectedly.
DLPX-74860	Provisioning the 2nd generation VDB fails if the dSource has imported read-only transportable tablespaces fails.
DLPX-74975	Allow adding invalid or unreachable paths as a shared backup location for dSources.
DLPX-75026	Invalid JDBC connections are not purged from the connection pool when the home is changed.

Bug number	Description
DLPX-75134	Improved performance for the Environment Databases pages when there are a lot of databases.
DLPX-75363	Update exception description is seen when the SnapSync fails for ASE encrypted database.
DLPX-75401	local listener set to null if oracle.lsnr.protocol_registration_order is quoted.
DLPX-75506	Fix a bug that can cause Oracle RAC VDBs to fail with stale NFSv3 mounts if NFSv4 is also enabled.
DLPX-75532	Insufficient heap memory settings on AIX cause connector and SnapSync to hang or crash.
DLPX-75663	FIPS compliant algorithms will be used while merging the old and new toolkit directories during environment refresh.
DLPX-75716	Delphix may remove Oracle VDB temp tablespaces during Snapsync.
DLPX-75735	Fixed creation of retention policies workers on replica objects after failover.
DLPX-75834	Rearranged Syslog configuration dialog inputs to avoid confusion and have a more consistent user experience.
DLPX-75858	BEQ processes can hold on to file descriptors leading to hook scripts hanging after upgrading to 6.0.7.0.
DLPX-76018	Remove hardcoded 5-minute timeout for doDropPDBKeepDatafiles.sh.
DLPX-76140	TDE SnapSync should ignore WARNING plugin violations.

3.2.162 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-71980	Fix a bug that was causing the Delphix Engine storage pool to fail to import on boot under certain circumstances.

3.2.163 Release 6.0.8.1 changes

3.2.164 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-75804	Some Delphix operations may fail if mount and umount commands, on staging or target hosts, are setup to run as sudo and if sudo rules prohibit these commands from running with unrecognized options. The issue is fixed now after removing "-v" added in 6.0.8.0.

3.2.165 Release 6.0.8.0 changes

3.2.166 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-49694	Skip parsing of DBCC when code 0 is not present in the DBCC page output. In case the DBCC page has code 0 but not dbid, use bdbid (present in the buffer section).

Bug number	Description
DLPX-68764	VDB SnapShot does not progress if ASE database devices are not on Delphix storage, environment monitoring raises a fault. Subsequent VDB operations like enable, disable, start, stop, delete, snapshot, or refresh will fail.
DLPX-70793	Delphix Engine should not allow linking Oracle DB with null `db_unique_name`.
DLPX-71300	For newer ASE versions (>= 15.7 SP138 and 16.0 SP02 PL05 and ASE 16.0 SP03), do not run DBCC PAGE anymore, as it was an identifier for DBCC CHECKALLOC that is already not run.
DLPX-71471	Error message asking user to manually perform disable/enable operation or correctly rename the target database back will be displayed during Start VDB, if VDB does not exist.
DLPX-71687	Provide a mechanism to enable VDBs up to filesystem mount point.
DLPX-71875	Fixed a bug that results in a memory reservation not being represented correctly in the Delphix API.
DLPX-72046	Deletion of vPDB in a vCDB shows this warning, "Encountered an error while shutting down and cleaning up Oracle files."
DLPX-72209	Downloading a support bundle is not supported at the same time that an upload of an upgrade image has been initiated by the same Delphix user.
DLPX-72319	Fixed an issue where some error dialogs would freeze in Internet Explorer 11.
DLPX-72705	Connection timeout when deleting remote shipper script can cause a timeout in LogSync client.

Bug number	Description
DLPX-72757	ASE sync using Dump History fails for large dump history files.
DLPX-72780	Timezone is set incorrectly for snapshots of Solaris 10 dSources and VDBs.
DLPX-72904	Storage capacity now includes usage from all file system objects, not just snapshots.
DLPX-73143	Fixed an issue where the support bundle dialog showed a loading spinner intermittently while jobs were running.
DLPX-73354	Traverse all shared backup locations while syncing, even if some of the paths are invalid or not reachable.
DLPX-73489	Fixed bug where adding or editing a parameter using the UI VDB Config Template "Text" tab was truncating the parameter's value.
DLPX-73602	Incorrect mount options used when a single instance RAC is linked as a standalone single instance.
DLPX-73607	Added paging for days with large numbers of snapshots to prevent slowdown.
DLPX-73623	Fixed an out-of-memory condition that occurs in SSH tunneling for encrypted log-syncs when storage latencies are high.
DLPX-73627	The help text on upgrade replication warnings have been updated to avoid confusion between Ignored and Resolved.

Bug number	Description
DLPX-73668	Fixed Missing security headers.
DLPX-73669	Cross-site request forgery (CSRF) issue in management UI.
DLPX-73727	Fixed an issue where the faults table was unable to navigate to other pages.
DLPX-73797	Fixed VDB refresh failures due to SQL Server Error 924 after setting VDB to single user mode.
DLPX-74025	Implemented logic to retry offline database along with a drop database to overcome deadlocks while off-lining or dropping the database.
DLPX-74029	VMware Hot-Add memory is not immediately reflected in the system API.
DLPX-74057	Fixed a typo in "Download Support Bundle" UI component where the word "suport" was missing a "p".
DLPX-74254	Ownership of files inside VDB now matches new owner when VDB owner is changed.
DLPX-74298	Fixed an issue where the user could not upload a keystore with a blank keystore passcode.
DLPX-74362	Fixes an issue with namespace deletion when the replication receive jobs have been cleaned up.

Bug number	Description
DLPX-74442	VDB Enable with attemptStart=false will now mount the datasets so that VDB can be started.
DLPX-74457	Cluster discovery for Oracle RAC clusters are partially failing on Solaris 10.
DLPX-74529	Fixed a bug so that an upgrade completes even when jobs fail.
DLPX-74542	Fixed a bug so that upgrade completion is properly handled after kernel upgrades.
DLPX-74645	Delphix Engine uses the uptime command to keep track of a target host reboot and auto start VDBs on the host. In some cases, the output of this command is not what is expected and causes unintended restart of a stopped VDB. This issue is now fixed.
DLPX-74656	Oracle errors during doCreateSPFile.sh are not captured.
DLPX-74704	Fixed a bug where the Dataset scroll does not extend to the bottom of a dataset list, thus truncating the status of the last dataset in the expanded group.
DLPX-74883	Prevent support bundle collection from cancelling replication.
DLPX-74911	Talaria TCP fallback fault may be misconstrued if an Oracle RAC node is down.
DLPX-74997	Prevent granting replicated roles to users.

Bug number	Description
DLPX-75083	Post upgrade cleanup task may become unresponsive while attempting a migration from 5.3.x to 6.0.x due to several threads stuck in WAITING state.
DLPX-75095	Provisioning an Oracle VDB fails if change-archivelog-mode.sh takes longer than 5 minutes.
DLPX-75134	Improved performance of the Environment Databases page under certain conditions.
DLPX-75188	Fixed "out of memory" issue when processing a large number of objects on the Target engine.
DLPX-75204	Addressed a performance issue on the Target engine when receiving large number of replicated objects.
DLPX-75208	Snapshot names are incorrectly redacted in the engine metadata dlpx_action table in support bundles.
DLPX-75416	Fixed a replication issue when there are sources with TLS enabled.

3.2.167 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-72065	Fixed a bug that can cause a Windows iSCSI initiator to fail connect to the Delphix Engine.

Bug number	Description
DLPX-72681	Console Delphix status screen shows a Python stack trace if the system is configured with a static IP address.
DLPX-73423	Console Delphix status screen shows a Python stack trace if the system has no default route.
DLPX-74216	Fixed an issue that causes management service failures in low memory situations.
DLPX-74622	Fixed a bug that can cause a replication job to fail with an internal error.
DLPX-75089	Fixed a bug that can cause NFSv3 clients to lose locks during upgrade verification.
DLPX-75524	Fixed a bug that can lead to Oracle data corruption when running VDBs on Oracle 19c with dNFS.

3.2.168 Release 6.0.7.0 changes

3.2.169 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-39006	LogSync failed with "Cannot read archived log due to failure of log shipping script".
DLPX-39245	Fixed a bug that caused the management service to become inaccessible if the storage pool ran out of space.
DLPX-59155	Provisioning a VDB or vPDB failed with unclear error message 'A database with the name "xxx" already exists'.

Bug number	Description
DLPX-60317	Fixed Out of Memory issue when replicating a large number of objects.
DLPX-60947	Replica VDBs will be updated when performing a point-in-time restore.
DLPX-62805	vPDB provision did not raise an error when a non-provisionable target point-in-time was provided.
DLPX-62969	Fixed Out of Memory issue when receiving large number of replicated objects.
DLPX-64600	Skipped connecting to ASE dSources during SnapSync policy runs as it is not applicable for them, hence prevent recurrent faults that the policy throws for connectivity issues.
DLPX-67363	Maximum identify provider authentication time age can be customized for single sign-on.
DLPX-67607	Fix to make Snapsync throw exception if manifest file is missing or of 0 bytes instead of internal error with null pointer exception.
DLPX-67767	Fixed a bug that caused the upgrade to hang, while waiting for running jobs to finish.
DLPX-70821	Allow the entity id for SAML single sign-on to be a URL for compatibility with Azure AD.
DLPX-71783	doRenameDatafiles cleanup of extra files fails due to file permissions mismatch.
DLPX-72010	Fixed an issue that prevents changing the default gateway using the network setup CLI.
DLPX-72075	Maximum SAML response time skew can be customized for single sign-on.
DLPX-72191	Oracle privilege discovery not performed for all homes if an invalid home exists.

Bug number	Description
DLPX-72351	When a user tries to change credentials for a dSource, validating the credentials before updating them. In case of invalid credentials, showing user an error message about it.
DLPX-72545	Initial ORA-65294 error not reported to user when vPDB provision fails due to compatible parameter mismatch.
DLPX-72652	Fix and issue that prevents use of the NFSv4 on some versions of SUSE Linux targets.
DLPX-72698	Patching Oracle 19C vCDB leads to ORA-25153 as described in 2285159.1.
DLPX-72807	Fixed issue with SQL Server 2014 dSources with filestreams where sync failed in merging filestream directories due to long path names.
DLPX-72882	Datasets hooks script editor properly displays multiline scripts instead of as one long line on non-Chrome browsers.
DLPX-72916	Empty string in SNMPv3 USM username creation no longer throws fatal error.
DLPX-73048	Non-sys user credentials for Oracle sources cannot use password vault.
DLPX-73108	Fix a bug that prevents the API from displaying the correct number of CPUs or amount of memory assigned to a Delphix Engine after a hot-add operation.
DLPX-73201	Fix an issue that prevents the configuration of additional NICs on Azure Delphix Engines.
DLPX-73202	Fix a bug that can cause a VDB to fail to mount while other VDBs are being stopped.
DLPX-73424	Fix a bug that prevents the sysadmin from deleting a default route.
DLPX-73527	SnapSync job fails with 'internal error during execution' due to ONS/FanManager errors.

Bug number	Description
DLPX-73528	Fixed a bug that prevented accessing SDD specs from CLI.
DLPX-73611	Kerberos ticket expiration date parsing is incorrect after migration from Illumos to Linux.
DLPX-73742	Provisioning an Oracle TDE-enabled vPDB fails with the error "ORA-28367: wallet does not exist" if the TDE wallet for the target linked CDB is stored on ASM storage.
DLPX-73765	Fix a file descriptor leak that causes the management service to crash over time.
DLPX-73789	Auxiliary CDB instance uses dSource keystore location if WALLET_ROOT is configured in dSource.
DLPX-74030	CDB database password may be leaked as part of environment monitor checks that launch sqlplus command on the source or target host.
DLPX-74043	Delphix OS user cannot provision TDE-enabled vPDB due to directory permissions in the default wallet location.
DLPX-74044	Delphix OS user cannot provision TDE-enabled vPDB in Delphix-writable keystore location due to directory permissions.
DLPX-74119	Drop database fails if default database is set to any other than master.
DLPX-74164	Sync fails with db.aseldb.source.dump_history.incomplete_stripes after dump history file is purged and Use dump history is enabled for the dSource.
DLPX-74233	During failover of a namespace, if there is a collision between an environment in the namespace with one on the target engine, the namespace environment will get renamed if its host does not match that of the environment on the target.

3.2.170 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-73390	Improve replication receive throughput.
DLPX-73393	Improve write performance under extreme disk fragmentation.
DLPX-73280	Improve write performance under extreme disk fragmentation.

3.2.171 Release 6.0.6.1 changes

3.2.172 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-73848	Fixed an issue that can cause the management service to fail to start after upgrade on systems that have had SNMP enabled.
DLPX-73859	Fixed a file descriptor leak triggered by faults and alerts that can cause the management service to fail.

3.2.173 Release 6.0.6.0 changes

3.2.174 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-47065	VDB recovery failed when files other than archive logs were detected by Oracle.
DLPX-47493	Fixed the bug where VDB directory under the DelphixConnector directory was not being removed from the target host on MSSQL VDB deletion.
DLPX-48046	Added sorting parameter to network test APIs.
DLPX-61405	Replication may send more data than expected if masking involves dropping large DBF files.
DLPX-61525	The height of the storage configuration list was limited to show 3 disks at a time. It will now dynamically grow with the number of disks.
DLPX-63603	Increased connector timeout from 10 minutes to 30 minutes to avoid unnecessary faults due to timeout during Validated Sync operation.
DLPX-67368	Delphix Engine hostname change is now immediately reflected in Splunk events.
DLPX-67593	Fixed an issue that caused the management service to remain offline following an out-of-space condition.
DLPX-68531	Introduced better handling of UniversalConnectionPoolException errors during SnapSync.
DLPX-69759	Oracle environment discovery failed due to an unhandled exception occurring at insert into dlpx_faults.
DLPX-69852	Fixed a bug that caused network configuration problems when removing and adding additional NICs.

Bug number	Description
DLPX-70426	Redaction of usernames took forever on tables with millions of entries.
DLPX-70583	move-to-asm.sh fails if timing is set in glogin.sql.
DLPX-70638	Removed Failed Actions section of Actions sidebar, in favor of manually dismissing from Running Actions and falling to Finished Actions section.
DLPX-70653	Removal of all instances in a RAC VDB should not be allowed.
DLPX-70808	Fixed issue related to the creation of empty DisableBroker.sql on the Windows machine in case DisableBroker.sql execution fails in the first attempt.
DLPX-70896	Added more detailed error message for when the Delphix Engine fails to push a script to Windows host.
DLPX-70919	Fixed an issue that causes job progress to not update in Self-Service.
DLPX-70928	Fixed a bug that results in a Delphix Engine remaining powered on following a shutdown from the user interface.
DLPX-71093	For AG databases, a full backup is not required even recovery fork guid changed but the LSN chain didn't break because of transactional log backups.
DLPX-71097	Unable to ignore snl.bct.needed warnings if Block Change Tracking is legitimately disabled on an Oracle dSource.
DLPX-71153	Recovery of PDB should fail if the database is down after offlining datafiles.
DLPX-71370	While deleting initiator in Windows environment deletion operation, delete all the views as well for that initiator.
DLPX-71685	VDB is auto disabled if the hook fails.

Bug number	Description
DLPX-71865	Reduced the size of support bundles.
DLPX-71961	When a PDB is selected for replication, its CDB and all other PDBs in the parent CDB get automatically selected for replication. Going forward, in the above scenario, while the CDB will get selected, its other PDBs will no longer get selected.
DLPX-72031	Fixed VDB refresh operations failures due to 'DB STARTUP' background process spid greater than 50.
DLPX-72066	Migrate VDB verifies against the old configuration, rather than new.
DLPX-72083	Fix an issue that causes a fully-qualified hostname to be changed on upgrade from 5.3 to 6.0.
DLPX-72131	Added namespace support for HashiCorp password vaults.
DLPX-72265	doCreateTempfiles.sh.template exits with code 0 on failure.
DLPX-72340	Incomplete recovery not detected during provisioning.
DLPX-72386	Unlock Solaris x86 Solaris -> Linux x86 provisioning.
DLPX-72452	For clusters with long hostnames, vPDB sync fails with exception.oracle.accessor.instances.missing.
DLPX-72495	Fixed a bug that prevents the application from coming up after an upgrade
DLPX-72686	Delphix no longer logs environment variables in logs on connected hosts since this could leak sensitive information such as passwords that are sometimes stored as environment variables on database hosts such as for the ASE database.

Bug number	Description
DLPX-72730	Fixed a Snapsync performance issue.
DLPX-72790	SnapSync job fails with 'internal error during execution' due to ORA-01652.
DLPX-72862	The scenario which was causing the null pointer has been fixed now.
DLPX-73300	Validation of connection to a container for PDBs should allow connecting to CDB$ROOT.
DLPX-73311	Added platform detection for ESX 7.0u1.
DLPX-73449	Replication of policies between two engines, in a loop, could lead to OOM exceptions.

3.2.175 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-72990	Addressed a minor CVSS 5.9 security issue with no known attack vectors.
DLPX-73067	Fix for CVE-2020-10753.
DLPX-73069	Fix for CVE-2020-12059.
DLPX-73070	Fix for CVE-2020-1760.

3.2.176 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-71924	Fixed a bug that causes support bundle collection to fail with an internal error.
DLPX-72918	Fixed a system crash that can happen when replicating a masked VDB using SDD.
DLPX-73147	Fixed a bug that can cause a replication source to crash if it had run replication while running on 5.0.

3.2.177 Release 6.0.5.0 changes

3.2.178 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-23360	The unistallation of the Delphix Connector installer should succeed even if one of the component connector services doesn't exist.
DLPX-69155	Reduced the time taken to generate support bundles in some cases.
DLPX-67316	Recreating a controlfile against an Oracle source may yield misleading error during snapshot.
DLPX-70766	JDBC driver updated to resolve intermittent JDBC connection failures due to JDBC SSL bug.
DLPX-70785	Options passed to VDB mounts on target AIX hosts did not include read and write size values. This fix adds the rsize and wsize parameters to mount command depending on the maximum values host is configured to support.

Bug number	Description
DLPX-70741	Enabling Validated Sync while SAP ASE SnapSync job is running leaves staging database unrecoverable.
DLPX-69865	Fixed a bug that causes a network interface to become unconfigured if its MAC address changes.
DLPX-71233	If LogSync is suspended when performing SnapSync of a standby database in real-time apply, SnapSync attempts to backup the archived logs which can cause SnapSync to become unresponsive.
DLPX-69800	UEM/Hostchecker directory ownership checks fail on HPUX environment with long usernames.
DLPX-69807	Provided mechanism for the user to bypass corrupted/incomplete jdbc libraries.
DLPX-66585	Bundle ID "fault.environment.configuration.file.owner" reports insufficient host address.
DLPX-65739	createDelphixDBUser.sh fails when "@" used in the password.
DLPX-70973	SAP ASE database provisioning fails if the source database has holes in log fragments.
DLPX-71532	Improved error handling for Oracle memory configuration errors.
DLPX-62987	Allowed assigning privileges over replicated objects through the UI.
DLPX-71593	TIMEFLOW_REPAIR incorrectly skips a log because of "wrong database".
DLPX-71751	Added NFSv4 support on AIX for Oracle and SAP ASE.

Bug number	Description
DLPX-71736	Dynamically disable RPC services if NFSv3 is no longer in use.
DLPX-71772	Network DSP Test between versions 5.3 and (6.0.3, 6.0.4) is fixed.
DLPX-71513	Replicate non-data objects like delphix engine users, authorizations, roles, permissions, policies and DB config templates.
DLPX-71305	Unable to load dummy recovery database dump due to SAP ASE error 15728.
DLPX-71172	Enabling SAP ASE dump history causes IllegalStateException in getDumpListFromLastRestoreDateAndFiles due to timestamp mismatch because of TZ.
DLPX-71178	SAP ASE internal error raised when dump history file is purged using sp_dump_history.
DLPX-65101	Fixed a race condition between a DB_DELETE job and the Oracle retention policy worker for the same container that can lead to a deadlock between the job and the worker.
DLPX-71918	Fixed an issue that causes the Delphix Engine UUID to change upon rebooting in IBM Cloud.
DLPX-71141	Fixed an issue where upgrading an Oracle dSource or changing the environment user in a linked Oracle dSource fails with the error "SOURCE_UPGRADE job for xxx failed due to an internal error during execution."
DLPX-71611	Updated UI time zone library to IANA 2020a.
DLPX-70349	Fixed a memory leak during incremental replication.
DLPX-72038	Fixed an issue that prevents 5.3.x - 6.0.x upgrade if a static route exists that goes over a DHCP interface.

Bug number	Description
DLPX-72148	Fixed a bug of always order hooks alphabetically rather than the running order set by users.
DLPX-71971	Allowed Enable/Disable of VDB if its current Timeflow has at least one snapshot.
DLPX-72115	Changed the Time Point field on a VDB back to reflecting the point on the parent the VDB was created from, but displayed in the timezone of the parent.
DLPX-71995	6.0.4.0 can no longer interact with 5.3.x remote Masking Engines.

3.2.179 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-70675	Fixed a bug that causes the system to become unresponsive after expanding multiple storage devices.

3.2.180 Release 6.0.4.2 changes

3.2.181 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-72155	Fixed an issue that can render a Delphix Engine unbootable if a reboot occurs after upgrade verification but before the upgrade is applied.
DLPX-71141	Fixed an issue where upgrading an Oracle dSource or changing the environment user in a linked Oracle dSource fails with the error "SOURCE_UPGRADE job for xxx failed due to an internal error during execution."

3.2.182 Release 6.0.4.1 changes

3.2.183 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-71930	Fix a bug that causes feature flags to be disabled when upgrading to 6.0.4.0.

3.2.184 Release 6.0.4.0 changes

3.2.185 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-68173	Resolved an issue where temporary database backup/device files created for cleaning up the target database were not being deleted.
DLPX-68773	The management stack runs out of memory as the environment monitor does not purge stale objects.
DLPX-69573	Allow linking an Oracle PDB with a lowercase name.
DLPX-69634	Allow provisioning an Oracle PDB with a lowercase name.
DLPX-69962	After detaching a PDB, perform unplug/plug, and attach again, if disabled is performed before SnapSync, the PDB can no longer be enabled.
DLPX-66045	Prevent Self-Service Container branches getting into an unusable state by blocking deleting the last segment of branches.

Bug number	Description
DLPX-70 37	Snapsync performs an unnecessary checkpoint.
DLPX-68 277	Users will see the detailed error message upon connection failure to Delphix connector during OS user validation and there will also be a "More" button with an error message which will open an error popup with all error details.
DLPX-70 288	On the "Add Environment" screen when OS user validation will get fail, they will see the "More" button along with the error message. When the user clicks the button, an error popup opens with all details of the error and suggested action.
DLPX-70 832	NFSv4 support for appdata sources running on AIX.
DLPX-68 495	Fixed GUI reporting conflict information when creating a Retention Policy.
DLPX-70 788	Added Environment User field for MSSQL sources in Datasets -> Configuration -> Source tab -> Staging Environment section.
DLPX-58 047	Fixed bug where the sort sequence was incorrect. Fixed in Hook Operation Templates.
DLPX-67 931	Provision against VPDB after create/drop a new tablespace failed with exception.oracle.targetscripts.rename.datafiles.
DLPX-59 910	Comps.xml associated with Oracle Homes are marked as unparseable if they are longer than 65535 characters.
DLPX-55 476	CLI provisioning fails when the mount point provided includes quotes around the path.
DLPX-71 168	Changed type to text and spaced "Secret Key" and "Username Key".
ORB-32 85	Support using `api.delphix.com` as a proxy for verifying the Cloud Agent binary's code signature certificate.

Bug number	Description
DLPX-71006	Allow provisioning across patch versions for Oracle versions on or after 18.X.
DLPX-71334	Migrate NTP configuration when upgrading between 5.3 and 6.0.
ORB-3286	Communication with Central management servers is now routed through the web proxy when one is configured for the Engine.
ORB-3117	Summary: Increase an action's failure message size to 256 characters so users can view large failure messages.

3.2.186 Release 6.0.3.1 changes

3.2.187 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-71339	Fixed an issue that can cause the Virtualization Management service to become inaccessible when the system memory became highly fragmented.

3.2.188 Release 6.0.3.0 changes

3.2.189 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-68995	Improved performance of dataset deletion.

Bug number	Description
DLPX-68997 DLPX-68999	Improved single connection replication throughput.
DLPX-70697 DLPX-70703	Addressed an issue that causes long periods of I/O unresponsiveness.
DLPX-69953	Fixed a bug that can cause a Windows iSCSI initiator to fail to connect to the Delphix Engine.
DLPX-70512	Fix a hang in the I/O subsystem that can cause the Delphix Engine to become unresponsive.

3.2.190 Release 6.0.2.1 changes

3.2.191 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-70065	Provisioning a VDB from a dSource or another VDB will fail if the following conditions are met: • Delphix Engine has at least one dSource and a VDB created using a Python plugin prior to the upgrade • Delphix Engine was upgraded to 6.0.2 • Provisioning was attempted from the UI after the completion of the upgrade
DLPX-69350	Fixed an issue that the time point attribute of a VDB is not shown.

3.2.192 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-69864	Fixed an issue that causes MSSQL operations to hang after the reception of an iSCSI LUN reset.

3.2.193 Release 6.0.2.0 changes

3.2.194 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-62806	Fixed an issue where provision against PDB after unplug/replug against the same linked PDB fails with exception.oracle.targetscripts.controlfile.create.
DLPX-67567	Oracle Source Continuity creates an unnecessary source-archive file system on zfs.
DLPX-27807	LogSync may fall behind when connected to an Oracle physical standby database in Real-Time Apply mode.
DLPX-68385	Customer provided Oracle Java missing in the search path for Java on hosts.
DLPX-62782	Reducing the number of nodes for RAC VDB and VDB in NOARCHIVELOG mode may result in ORA-00258 errors during VDB enable operation.
DLPX-62738	Better error message when plugins are uploaded out of sequence.
DLPX-68722	The product now recognizes VMware with BIOS date of 12/12/2018 as VMware 6.7.0u2.

Bug number	Description
DLPX-68579	SnapSync of Oracle 19c DB with encrypted tablespace fails with fatal exception "Block header 91 is not empty".
DLPX-68689	Fixed the issue where a huge number of error messages from ASE caused OutOfMemory Error.
DLPX-68957	Always On AG discovery will not fail in a multi-subnet environment.
DLPX-63088	Can now recover multiple Self-Service containers at the same time.
DLPX-47977	Improved handling of snapshot standby.
DLPX-64125	SnapSync failed with exception.oracle.dsource.sync.no_hosts.rac on RAC clusters with very long hostnames.
DLPX-62584	PDB enable failed after migration if mountBase has a trailing slash.
DLPX-68657	Virtualization can now fetch jobs from Masking engines configured with HTTP redirection.
DLPX-69121	It is no longer mandatory to have at least one enabled system administrator with local credentials.
DLPX-68167	Fixed an issue where too many requests were being sent for Faults from the Datasets pages.
DLPX-69082	Large stderr produced by failed rsync jobs are truncated to prevent Java OutOfMemory errors.
DLPX-58600	Datasets filter updated so that all items within a group that matches the filter string are displayed, even if the items contained in the group do not match the filter string.
DLPX-65896	VDB deletion failed due to the inability to delete LogSync worker.

Bug number	Description
DLPX-57903	Improved diagnosability of PDB discovery issues.
DLPX-68878	Fixed issue where start/stop buttons were not being displayed in the RAC instances configuration table.
DLPX-69271	Enabled replication smart failover by default.
DLPX-66715	The user-visible name for Oracle cluster objects is being replaced with the Oracle cluster name. For Windows clusters, the user-visible name is being replaced with the cluster address.
DLPX-68929	Changed default replication settings for better out of the box performance.
DLPX-68930	Improved replication throughput when sending multiple timeflows.
DLPX-69245	Fixed a memory leak that occurs when experiencing connectivity errors.
DLPX-69377	At least one non LDAP system user should be enabled when the LDAP server is being disabled.
DLPX-68575	LDAP principal fields were not being redacted in phone-home bundles.
DLPX-68528	Self Service Recover operation failed due to missing Timeflow.

3.2.195 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-66808	Re-introduced console splash screen with IP address and service states.

3.2.196 Release 6.0.1.1 changes

3.2.197 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-69203	Improved synchronous write performance over iSCSI.
DLPX-69167	Improved SQL Server data ingestion performance by leveraging asynchronous writes on underlying storage.
DLPX-69298	Eliminated possible data corruption on SQL server and vFiles over iSCSI that can occur when a Delphix Engine reboots.

3.2.198 Release 6.0.1.0 changes

3.2.199 Fixes that take effect immediately after upgrading

Bug number	Description
DLPX-60689	For SAP ASE, instead of using the DBCC CHECKALLOC command to fix DBID mismatch issue, the MOUNT command with FIXDBID and ALLOW_DBID_MISMATCH clauses will be used, to improve performance.
DLPX-65831	VDB snapshots need to clean unneeded ZFS datafiles.
DLPX-63949	Improved boot time after 5.3 to 6.0 migration by optimizing metadata indexing.
DLPX-66261	Upgrades to 6.0.0.0 will only be supported from a release greater than or equal to 5.3.6.0.
DLPX-66486	Snapshot of a linked database can end up with extra datafiles that do not belong to the database which might cause VDB on VDB provision to fail during rename of datafiles.
DLPX-66558	Cluster environment discovery was incomplete if the host locale was not English.
DLPX-66804	DB_LINK using incorrect user when RAC node also configured as a standalone environment.
DLPX-66768	vPDB save state lead to rollback or child provisioning failures.
DLPX-66823	Unable to link database with CL8MSWIN1251 charset.
DLPX-64538	Fixed a bug causing the timezone selector to only be visible when manually setting the time.

Bug number	Description
DLPX-66809	Removed the Windows Diagnostics Files and Directories on successful Diagnostics upload.
DLPX-67279	Provision failed when the source was from a RAC Oracle Standard Edition database and the target was Oracle Standard Edition.
DLPX-67451	Fixed an issue that sporadically caused replication to fail with an internal error.
DLPX-67454	Delphix Engine should select the highest version ojdbc driver available at ORACLE_HOME/ojdbc/lib.
DLPX-66077	Ensures child worker threads are gracefully exited when parent linked source sync job has completed/terminated.
DLPX-45983	MSSQL Validated sync will resume when storage usage falls below the threshold if storage threshold enforcement failed in the past.
DLPX-67560	Fixed an issue where MT provision may result in ORA-02058 due to un-purged or inflight 2PC transactions on dSource.
DLPX-67594	Old timeflows and snapshots are not getting removed by snapshot retention.
LX-2020	Report the correct amount of memory allocated to EC2 Nitro instances.
DLPX-67413	Fixed an issue where VDB point in time provisioning might fail if Oracle database environment is configured in a non-English locale.
DLPX-67684	PDB provisioning failed if the source had shutdown triggers.
DLPX-67575	Fixed failure during point in time 'Virtual to Physical' provisioning.
DLPX-67668	After setting the database online give it some extra time to startup completely, before doing any further operation on it.

Bug number	Description
DLPX-67759	Redact sensitive information from phone-home data.
DLPX-64638	Validated sync stops working if Delphix cannot connect to the backup server.
DLPX-65559	Even when the staging instance is down, attempt counter to detect backup files keeps on increasing and eventually, it stops detecting backups.
DLPX-56537	When a target host is used by a large number of dSources for staging or has a large number of objects, the performance of Delphix operations like validated sync, refresh, rewind, etc can be slow due to Powershell processes being serialized.
DLPX-67894	Removing cluster resource without removing its dependency can result in cluster failure. So, added retryer logic while fetching the resource dependencies (Get-ClusterResourceDependency) and ultimately fail the operation after all the retries.
DLPX-67813	Unsupported SQL server backup type gets picked while validated sync and the operation fails while looking for the backup. So, introduced a tunable filter to automatically skip SQL backups taken by backup software not supported by yet Delphix.
DLPX-67925	Added env host connectivity toolkit support for SLES on Power9.
DLPX-67934	Retries to fetch image identifiers during Netbackup restore if there is a mismatch between MSDB and Netbackup Master.
DLPX-67655	Fixed an issue where retention enforcement can generate user-visible errors while attempting to delete snapshots with dependencies after PDB migration to new CDB.
LX-1944	EBS NVMe devices can now be used in Delphix Engines.
DLPX-68022	Fixed an issue where hostchecker 'Check Oracle DB Instance' fails on HPUX and AIX.
DLPX-68124	PDBs with lower/mixed case names will not enable after an upgrade.

Bug number	Description
DLPX-68126	Fixed a bug that limits the number of disks that can be added in GCP.
DLPX-67421	Update the primary db file names in a transaction with the Timeflow creation to make sure whenever a Timeflow is created successfully we have its primary file information.
DLPX-67440	Skip VDBs having its current Timeflow as null from 'PrimaryDbFileAvailabilityCheck' as these VDBs doesn't undergo queisceing and are recoverable by refreshing them.
DLPX-61818	Linking wizard - Target Environment step - Privileged Credentials authenticates on the selected target now.
DLPX-68117	Some non-Admin users, lack all permissions, are unable to login to upgraded engine.
DLPX-67290	A wrong version input by user while manually adding a SQL Server instance, created issues in provisioning VDBs. SQL Server version will now be auto-discovered for manually added instances on adding or refreshing the environment.
DLPX-66238	Updated error message to let know user that non discovered CDBs are filtered out from the list when linking a detached source.
DLPX-68457	When a target host is used by a large number of dSources for staging or has a large number of objects, the performance of Delphix operations like validated sync, refresh, rewind, etc can be slow due to Powershell processes being serialized.
DLPX-68484	Fixed the issue where 'lstart' column value of sysusages table was beyond the range of Integer data type by taking the Long data type to store the lstart value.
DLPX-68500	Fixed an issue where the NTP service is not started following a reboot.
DLPX-68290	Support bundle generation can be time-consuming if the engine has a large number of snapshots to process.
DLPX-67792	Fixed issue in grids in which the selection checkbox was unclickable.

Bug number	Description
DLPX-67555	Provision vPDB/vCDB fails with ORA-45900 if the parameter enable_pluggable_database is omitted when specifying database parameters for new vCDB.

3.2.200 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-67782	Engines running 5.3 on EC2 i3 can now be migrated to 6.0.
DLPX-67961	Fixed an issue that prevents ssh access after switching to a static IP address.
DLPX-65948	Fixed a bug that could cause replication jobs to fail with internal errors
DLPX-68025	Improved boot time after 5.3 to 6.0 upgrade by reducing the overhead of setting ZFS properties.
DLPX-67868	Fixed a bug that can cause the management service to run out of memory when disabling the Splunk integration.

3.2.201 Release 6.0.0.0 changes

3.2.202 Fixes that take effect immediately after upgrading

Bug numer	Description
DLPX-27433	The analytics GUI network graph shows newly added NIC information without requiring a management service restart.

Bug numer	Description
DLPX-33998	If you add a hook script via the CLI, newlines are removed erroneously.
DLPX-40094	Correctly set the default type for the parameters to all operations in the CLI according to the container type.
DLPX-43215	Exclude sybsecurity from the list of auto-discovered databases.
DLPX-48712	Java 6 packages are no longer included in the product image.
DLPX-48280	When a user is set with the Provisioner role the 'provision' button does not appear, meaning anyone set with this role only is unable to provision VDBs.
DLPX-53996	The Delphix Engine does not provide instructions to browsers to avoid caching HTTP responses (pages).
DLPX-54740	Ensure Windows mount points are always unmounted as part VDB refreshes to prevent future VDB refreshes from failing due to "ERROR_ASSIGN_MOUNTPATH: failed to assign mount path for disk at="">>, error="">,>
DLPX-55282	In environments where the vPDB has been provisioned using a Delphix provisioned virtual CDB, shutting down the virtual PDB causes it to get into an incorrect "Cannot monitor" state, this has now been fixed to show the correct "Stopped" state.
DLPX-55598	Fixed an issue where vPDB refresh/rollback triggers spurious vCDB restart jobs, after vPDB+vCDB auto-restart.
DLPX-55829	Validated Sync can fail when monitoring ASE backup servers started by using the $DSLISTEN environment variable instead of the "-S" argument. This can be worked around by accessing $DSLISTEN in the RUN_xxxxx script and pass it down as -S.
DLPX-55958	VDBs with no snapshots failed to re-enable after a Delphix Engine upgrade, this has now been fixed.
DLPX-57454	Display underlying ssh error when environment host connections fail.

Bug numer	Description
DLPX-58519	Enable Oracle LiveSource when LiveSource is in RESYNC_NEEDED state currently re-start Oracle Redo Apply. Oracle Redo Apply should not be restarted in this state.
DLPX-58760	Fixed a TCP port leak in the network throughput test feature.
DLPX-58845	Provisioning vFiles to the same host using different OS Environment Users no longer fails.
DLPX-59772	The API to list all snapshots consumes a significant amount of memory when there are more than 100,000 snapshots on the engine.
DLPX-60356	Fixed an issue where Oracle remote listener registration fails if set to empty string.
DLPX-60603	Network settings dialog now displays actual MTU value rather than a checkbox.
DLPX-60907	Fixed an issue where the Environment Monitor on Redhat 6.9 and 6.10 might throw unidentified version errors.
DLPX-60979	When user configures connection strings manually, these connect strings can end up connecting to incorrect PDBs/CDBs causing invalid snapshots. Verify that each connection to a PDB/CDB connects to the expected PDB/CDB.
DLPX-60993	Delphix backups create controlfile records; in rare circumstances, these records can cause invalid snapshots. To avoid this problem, remove Delphix backups control file records when using SCN-based SnapSyncs once a SnapSync completes successfully.
DLPX-62094	Allow certificates to expire after issuer certificate expiration.
DLPX-62241	Reduce SSH connections by temporarily preserving and reusing existing Delphix<->host connections where possible.
DLPX-62781	Spurious job event "DISCOVERED_TO_MANUAL_ORACLE_CLUSTER_NODES" no longer shows up for non-Oracle RAC environment refreshes.

Bug numer	Description
DLPX-62892	In Oracle versions 18c and 19c, an Oracle bug can prevent the datafile headers from being updated for a standby database when managed recovery is running, resulting in failed SnapSync operations. Alert the user that an Oracle patch might be needed.
DLPX-62962	Removed unneeded EMPTY_RENEGOTIATION cipher
DLPX-62998	Fixed an issue where stale file mounts may be leftover when vPDB provision fails.
DLPX-63469	Initial setup now fails if the system was not provisioned with enough storage.
DLPX-63600	Network settings dialog now displays actual MTU value rather than a checkbox.
DLPX-64641	Fixed an issue where the last snapshot of a vPDB Timeflow can be deleted after the vPDB has been disabled, thus leaving the vPDB in a state with no provisionable snapshots.
DLPX-64711	Allow provisioning to complete when source CDB includes PDBs in a broken state.
DLPX-66020	Provision should remove files present in datafile filesystem that are not part of the database when provisioning a VDB from a VDB.
DLPX-67299	ASE environment discovery will not fail if there is a mismatch of "dataserver name argument" and value of "@@servername".

3.2.203 Fixes that take effect after upgrading and rebooting (optional)

Bug number	Description
DLPX-57384	Fixed a system hang caused by a deadlock in ZFS.

3.3 Known issues

3.3.1 Version 23.0.0.0

Key	Summary	Workaround
DLPX-58185	Changing a custom policy while it is running on an object can cause the policy execution to fail.	None
DLPX-88930	An “Apply Now” upgrade from 18.0.0.x to 19.0.0.x+ will log the user out due to the `delphix-mgmt` service restarting. The upgrade may still be in progress (e.g. quiesce jobs). To view the upgrade progress, login into Setup as a sysadmin to see the running actions.	None
DLPX-89128	`/var/crash` cleanup should include `hprof` files in this directory.	None
DLPX-90491	SnapSync of a standby Oracle PDB might fail with `exception.oracle.snl.snapsync.mounted.pdb.not.consistent` during initial load/full backup if PDB SnapSync is not completed without interruption.	Remove the dSource for initial load and ensure the PDB SnapSync completes without interruption. For a full backup after the initial, a new full backup with the “Do Not Resume” option will need to be taken, which should complete without interruption.
DLPX-90568	A sysadmin user cannot view actions initiated by a different sysadmin user.	None
DLPX-90596	Enabling an operation on an Oracle vPDB fails after the vPDB is exported to ASM using an in-place V2ASM operation, followed by a migrate operation.	Provision a new vPDB from the original vPDB's latest point in time and then delete the original vPDB.

3.3.2 Version 22.0.0.0

Key	Summary	Workaround
DLPX-58185	Changing a custom policy while it is running on an object can cause the policy execution to fail.	None
DLPX-88930	An “Apply Now” upgrade from 18.0.0.x to 19.0.0.x+ will log the user out due to the `delphix-mgmt` service restarting. The upgrade may still be in progress (e.g. quiesce jobs). To view the upgrade progress, login into Setup as a sysadmin to view the running actions.	None
DLPX-89128	`/var/crash` cleanup should include hprof files in this directory.	None
DLPX-90491	SnapSync of a standby Oracle PDB might fail with `exception.oracle.snl.snapsync.mounted.pdb.not.consistent` during initial load/full backup if PDB SnapSync is not completed without interruption.	Remove the dSource for initial load and ensure the PDB SnapSync completes without interruption. For a full backup after initial, a new full backup with “Do Not Resume” option will need to be taken, and this should complete without interruption.
DLPX-90568	A sysadmin user cannot view actions initiated by a different sysadmin user.	None
DLPX-90596	Enable operation on an Oracle vPDB fails after the vPDB is exported to ASM using an in-place V2ASM operation followed by a migrate operation.	Provision a new vPDB from the original vPDB's latest point in time and then delete the original vPDB.
DLPX-90748	Failed to proxy-enable a vPDB when its vCDB's name has same prefix as another vCDB.	Enable the vCDB first and then it's corresponding vPDB.

Key	Summary	Workaround
DLPX-90785	SnapSync operation for Oracle Staging Push dSources will fail with `exception.oracle.dsource.stagingdb.datafile.checkpoint_scn.not.matching` if the production database has read-only tablespaces.	No workaround.
DLPX-90805	ZFS filesystems are created with 128k block size for Oracle Staging Push dSources resulting in high CPU usage and NFS latency.	No workaround.

3.3.3 Version 21.0.0.0

Key	Summary	Workaround
DLPX-58185	Changing a custom policy while it is running on an object can cause the policy execution to fail.	None
DLPX-78589	During an upgrade, when the Continuous Data Engine tries to run the ASE `UNMOUNT` command while also trying to quiesce VDBs, the `UNMOUNT` command gets stuck (this command does not run under a timeout). Due to this, the upgrade job is stalled.	When the ASE `UNMOUNT` command is hung, the ASE instance must be restarted.
DLPX-86181	Provision, refresh, rewind, or start operations on a virtual database, on a Solaris host with NFSv4 enabled, may become stuck indefinitely due to the mount process getting stuck on the Solaris NFS client.	Switch the NFS protocol to v3 using the Delphix Continuous Data Engine CLI, manually terminate the stuck mount processes on the Solaris host, then re-attempt the failed operation.
DLPX-88930	An “Apply Now” upgrade from 18.0.0.x to 19.0.0.x+ will log the user out due to the delphix-mgmt service restarting. The upgrade may still be in progress (e.g. quiesce jobs). To view the upgrade progress, login into Setup as a sysadmin to view the running actions.	None

Key	Summary	Workaround
DLPX-89128	`/var/crash` cleanup should include hprof files in this directory.	None
DLPX-89562	Self-Service Refresh can still get an `IllegalStateException` while running the reprovision, because of retries for reprovision.	The default `retryAttempts` is 1 for Self-Service operations. Thus, if a Self-Service operation like deprovision/reprovision failed during a refresh, Delphix will retry the operation (deprovision/reprovision) again. To stop retrying the operation, set the `retryAttempts` to 0 to get the real error, instead of an internal error. This will enable Delphix to analyze the actual error for the failed scenario. `ip-10-110-195-124 selfservice config update *> ip-10-110-195-124 selfservice config update *> set retryAttempts=0 ip-10-110-195-124 selfservice config update *> commit; ip-10-110-195-124 selfservice config>`

Key	Summary	Workaround

DLPX-89696	The start of an Oracle RAC virtual PDB in a virtual CDB, fails with an `exception.oracle.targetscripts.pdb.open` error if the virtual PDB is already running.	Start the vCDB instances manually (one at a time) from the CLI on all the nodes where the vPDB instance is not running as follows:
		1. Stop the vCDB instance on the node first if its already running:
		`source → select vcdb → stop; set instances.0=<instance number>; commit`
		2. Start the vCDB instance on the node:
		`source → select → start; set instances.0=<instance number>; commit`
		3. Start the vPDB from the Delphix UI.

DLPX-90596	Enable operation on a vPDB fails after the vPDB is exported to ASM using an in-place V2ASM operation followed by a migrate operation.	Provision a new vPDB from the original vPDB's latest point in time and then delete the original vPDB.

3.3.4 Version 20.0.0.0

Key	Summary	Workaround
DLPX-58185	Changing a custom policy while it is running on an object can cause the policy execution to fail.	None
DLPX-78589	During an upgrade, when the Continuous Data Engine tries to run the ASE `UNMOUNT` command while also trying to quiesce VDBs, the `UNMOUNT` command gets stuck (this command does not run under a timeout). Due to this, the upgrade job is stalled.	When the ASE `UNMOUNT` command is hung, the ASE instance must be restarted.

Key	Summary	Workaround
DLPX-84598	A sysadmin user cannot view actions initiated by a different sysadmin user.	None
DLPX-86181	Provision, refresh, rewind, or start operations on a virtual database, on a Solaris host with NFSv4 enabled, may become stuck indefinitely due to the mount process getting stuck on the Solaris NFS client.	Switch the NFS protocol to v3 using the Delphix Continuous Data Engine CLI, manually terminate the stuck mount processes on the Solaris host, then re-attempt the failed operation.
DLPX-87213	When relocating multiple terabytes of data either via block to object storage migration or device removal, management of the engine can become unresponsive for a lengthy period of time (e.g. 30+ mins), dependent on the dataset size and workload.	None, the data relocation and concurrent IO workloads will continue in the background. If the engine cannot be accessed after one hour, contact Delphix Support for corrective actions.
DLPX-88930	An Apply Now upgrade from 18.0.0.x to 19.0+ will log the user out due to the delphix-mgmt service restarting. The upgrade may still be in progress (e.g. quiesce jobs). To view the upgrade progress, login into Setup as a sysadmin to view the running actions.	None
DLPX-89128	`/var/crash` cleanup should include hprof files in this directory.	None

Key	Summary	Workaround
DLPX-89562	Self-service refresh can still get an IllegalStateException while running the reprovision, because of retries for reprovision.	The default `retryAttempts` is 1 for Self-Service operations. Thus, if a Self-Service operation like deprovision/reprovision failed during a refresh, Delphix will retry the operation (deprovision/reprovision) again. To stop retrying the operation, set the `retryAttempts` to 0 to get the real error, instead of an internal error. This will enable Delphix to analyze the actual error for the failed scenario. `ip-10-110-195-124 selfservice config update *> ip-10-110-195-124 selfservice config update *> set retryAttempts=0 ip-10-110-195-124 selfservice config update *> commit; ip-10-110-195-124 selfservice config>`

Key	Summary	Workaround

DLPX-89696	Start of an Oracle RAC virtual PDB, in a virtual CDB, fails with an error `exception.oracle.targetscripts.pdb.open` if the virtual PDB is already running.	Start the vCDB instances manually (one at a time) from CLI on all the nodes where vPDB instance is not running as follows:
		1. Stop the vCDB instance on the node first if its already running:
		`source → select vcdb → stop; set instances.0=<instance number>; commit`
		2. Start the vCDB instance on the node:
		`source → select → start; set instances.0=<instance number>; commit`
		3. Start the vPDB from the Delphix UI.

3.3.5 Version 19.0.0.0

Key	Summary	Workaround
DLPX-58185	Changing a Custom Policy while it is running on an object can cause the Policy execution to fail.	None
DLPX-78589	During an upgrade, when the engine tries to run the ASE `UNMOUNT` command while also trying to quiesce VDBs, the `UNMOUNT` command gets stuck (this command does not run under a timeout). Due to this, the upgrade job is stalled.	When the ASE UNMOUNT command is hung, the ASE instance must be restarted.
DLPX-84598	A sysadmin user cannot view actions initiated by a different sysadmin user.	None

Key	Summary	Workaround
DLPX-84710	Improved handling of missing networking devices after instance type migration.	None
DLPX-86181	Provision, refresh, rewind, or start operations on a virtual database on a Solaris host with NFSv4 enabled may become stuck indefinitely due to mount process getting stuck on the Solaris NFS client.	Switch the NFS protocol to v3 using the Delphix Continuous Data Engine CLI, manually terminate the stuck mount processes on the Solaris host, then re-attempt the failed operation.
DLPX-87213	When relocating multiple terabytes of data either via block-to-object storage migration or device removal, management of the engine can become unresponsive for a lengthy period of time (i.e. 30+ mins), dependent on the dataset size and workload.	None, the data relocation and concurrent IO workloads will continue in the background. If the engine cannot be accessed after one hour, contact Delphix Support for corrective actions.
DLPX-88930	An 'Apply Now' upgrade from 18.0.0.x to 19.0.0.x will logout the user due to the delphix-mgmt service restarting. The upgrade may still be in progress (e.g. quiesce jobs). To view the upgrade progress, login into Setup as sysadmin to view the Running Actions.	None
DLPX-89128	`/var/crash` cleanup should include hprof files in this directory.	None
DLPX-89817	Software keystore TDE-enabled provisioning fails after upgrading to 19.0.0.0	Refresh the source environment after the Delphix engine upgrade and re-attempt the provision.

3.3.6 Version 18.0.0.0

Key	Summary	Workaround
DLPX-58185	Changing a Custom Policy while it is running on an object can cause the Policy execution to fail.	None

Key	Summary	Workaround
DLPX-78589	During an upgrade, when the engine tries to run the ASE `UNMOUNT` command while also trying to quiesce VDBs, the `UNMOUNT` command gets stuck (this command does not run under a timeout). Due to this, the upgrade job is stalled.	When the ASE `UNMOUNT` command is stuck, the ASE instance must be restarted.
DLPX-82158	When discovering an Oracle RAC cluster, the engine automatically assigns instance numbers starting with 1, resulting in SnapSync failures.	Manually update the RAC instance number through the Delphix CLI after an environment refresh.
DLPX-84598	A sysadmin user cannot view actions initiated by a different sysadmin user.	None
DLPX-84710	Improved handling of missing networking devices after instance type migration.	None
DLPX-86181	Provision, refresh, rewind, or start operations on a virtual database on a Solaris host with NFSv4 enabled may become stuck indefinitely due to mount process getting stuck on the Solaris NFS client.	Switch the NFS protocol to v3 using the Delphix Continuous Data Engine CLI, manually terminate the stuck mount processes on the Solaris host, then re-attempt the failed operation.
DLPX-87213	When relocating multiple terabytes of data either via block-to-object storage migration or device removal, management of the engine can become unresponsive for a lengthy period of time (i.e. 30+ mins), dependent on the dataset size and workload.	None, the data relocation and concurrent IO workloads will continue in the background. If the engine cannot be accessed after one hour, contact Delphix Support for corrective actions.
DLPX-88322	If a datafile is added when a SnapSync job is running, provision or refresh to the snapshot generated by the SnapSync job will fail.	Take a new source snapshot (incremental is fine) and refresh the vPDB from the new snapshot.

3.3.7 Version 17.0.0.0

Key	Summary	Workaround
DLPX-56944	In Solaris 5.11 environments where the Delphix OS user shell is set to `csh`, host parameters may be improperly parsed, leading to environment refresh and other job failures.	Contact Delphix Support for corrective actions.
DLPX-58185	Changing a custom policy while it is running on an object can cause the policy execution to fail.	None
DLPX-66155	Restarting a Delphix Continuous Data Engine during a network throughput test is not recommended, as it may lead to a system hang.	None
DLPX-78589	During an upgrade, when the Continuous Compliance Engine attempts to run the ASE `UNMOUNT` command while trying to quiesce VDBs, the `UNMOUNT` command hangs up because that command does not run under a timeout. Due to this, the upgrade job stalls.	When the ASE `UNMOUNT` command hangs, the ASE instance must be restarted.
DLPX-82158	When discovering an Oracle RAC cluster, Delphix automatically assigns instance numbers starting with 1, resulting in SnapSync failures	Manually update the RAC instance number through the Delphix CLI after an environment refresh.
DLPX-84598	A sysadmin user cannot view actions initiated by a different sysadmin user.	None
DLPX-84710	Improving the handling of missing networking devices after instance type migration.	None
DLPX-86181	Provision, refresh, rewind, or start operations on a Solaris-hosted VDB with NFSv4 enabled may get stuck indefinitely, due to the mount process getting stuck on the Solaris NFS client.	Switch the NFS protocol to v3 using the Delphix Continuous Data Engine CLI, manually terminate the stuck mount processes on the Solaris host, then re-attempt the failed Delphix operation.

Key	Summary	Workaround
DLPX-87213	When relocating multiple terabytes of data (either via block-to-object storage migration or device removal), management of the engine can become unresponsive for a lengthy period of time (30+ mins), dependent on the dataset size and workload.	None, the data relocation and concurrent IO workloads will continue in the background. If the engine cannot be accessed after one hour, contact Delphix Support for corrective actions.
DLPX-88322	If a datafile is added when a SnapSync job is running, provision or refresh to the snapshot generated by the SnapSync job will fail.	Take a new source snapshot (incremental is fine) and refresh the vPDB from the new snapshot.

3.3.8 Version 16.0.0.0

Key	Summary	Workaround
DLPX-56944	In Solaris 5.11 environments where the Delphix OS user shell is set to `csh`, host parameters may be improperly parsed, leading to environment refresh and other job failures.	Contact Delphix Support for corrective actions.
DLPX-56978	Certain usernames are not available for use, as they are reserved system words (e.g. "root", "postgres", "delphix").	None
DLPX-56979	LDAP server test fails if authentication is set to DIGEST-MD5, but setup still works correctly.	None
DLPX-57142	The Job dashboard does not display the user that invoked each Job.	None
DLPX-57412	CLI parameters must be used exactly as described in documentation, including spelling and capitalization, or they are ignored.	None
DLPX-57673	Support bundle generation can be time consuming if the engine has a large number of core files to process.	None

Key	Summary	Workaround
DLPX-57823	Changing the Compliance Engine used by a VDB's masking job can lead to an internal error.	None
DLPX-58185	Changing a custom policy while it is running on an object can cause the policy execution to fail.	None
DLPX-59473	It is not possible to set a password policy that prevents an administrator from re-using a previous password, though this can be set for other users.	None
DLPX-66155	Restarting a Data Engine during a network throughput test is not recommended, as it may lead to a system stall.	None
DLPX-77986	On a Sybase host with multiple Sybase instances, if access to the first instance picked up for discovery fails due to invalid credentials, the discovery job will exit immediately – preventing discovery of the remaining instances.	Fix any credential failures.
DLPX-78589	During an upgrade, when the Data Engine tries to run the ASE `UNMOUNT` command while trying to quiesce VDBs, the `UNMOUNT` command hangs; this command doesn't run under a timeout and due to this, the upgrade job stalled.	When the ASE `UNMOUNT` command hangs, the ASE instance must be restarted.
DLPX-81478	If a transaction log is taken using the `standby_access` option for a SAP ASE database, the validated sync worker will not be able to restore that log, and will fail with a fatal exception (as Delphix does not currently support this option).	Change the backup script creating transaction logs to ingest FULL backups.
DLPX-81559	After upgrading an Oracle VDB from 12c to 19c, VDB refresh fails with "Failed to mount database instance" due to ORA-01130 or ORA-00201 errors.	Update the VDB parameters (either directly or via a VDB Config Template). See this knowledge base article.

Key	Summary	Workaround
DLPX-84598	A sysadmin user cannot view actions initiated by a different sysadmin user.	None
DLPX-84710	Improved handling of missing networking devices after instance type migration.	None
DLPX-86181	Provision, refresh, rewind or start operations on a virtual database, on a Solaris host with NFSv4 enabled, may become stuck indefinitely due to the mount process getting stuck on the Solaris NFS client.	Switch the NFS protocol to v3 using the Data Engine CLI, manually terminate the stuck mount processes on the Solaris host, and re-attempt the failed Delphix operation.
DLPX-87213	When relocating multiple terabytes of data, either via block to object storage migration or device removal, management of the engine can become unresponsive for a lengthy period of time (e.g. 30+ mins), dependent on the dataset size and workload.	None, the data relocation and concurrent IO workloads will continue in the background. If the engine cannot be accessed after one hour, contact Delphix Support for corrective actions.
DLPX-88040	Steps have been taken to make the users understand the reason of disabling some datasets during SDD Replication.	None
DLPX-88322	Provisioning or refreshing from a snapshot of an Oracle vPDB may fail during the renaming the data files stage if a data file is added during a SnapSync job.	Take a new source snapshot (incremental is fine) and refresh the vPDB from the new snapshot.

3.3.9 Version 15.0.0.0

Key	Summary	Workaround
DLPX-56944	In Solaris 5.11 Environments where Delphix OS user shell is set to csh, host parameters may be improperly parsed leading to Environment refresh and other job failures.	Contact Delphix Support for corrective actions.

Key	Summary	Workaround
DLPX-56978	Certain usernames are not available for use as they are reserved system words (e.g. "root", "postgres", "delphix").	None
DLPX-56979	LDAP server test fails if authentication is set to DIGEST-MD5, but setup still works correctly.	None
DLPX-57142	The Job dashboard does not display the user that invoked each Job.	None
DLPX-57412	CLI parameters must be used exactly as described in documentation, including spelling and capitalization, or they are ignored	None
DLPX-57673	Support bundle generation can be time consuming if the engine has a large number of core files to process.	None
DLPX-57823	Changing the masking engine used by a VDB's masking job can lead to an internal error.	None
DLPX-58185	Changing a Custom Policy while it is running on an object can cause the Policy execution to fail.	None
DLPX-59473	It is not possible to set a password policy that prevents an Administrator from re-using a previous password, though this can be set for other users.	None
DLPX-66155	Restarting a Delphix Engine during a network throughput test is not recommended as it may lead to a system hang.	None

Key	Summary	Workaround
DLPX-66860	For SSO/SAML, ADFS requires explicit rule to transform emailAddress attribute into nameid.	Create an explicit rule in ADFS that transforms the emailAddress attribute into a nameid. The rule type must be "Transform an incoming claim". The incoming claim type must be "Email address" and the outgoing claim type "Name ID". The nameid format must be "Email address".
DLPX-77986	On a sybase host with multiple Sybase instances, If access to the first instance picked up for discovery fails due to invalid credentials, the discovery job will exit immediately preventing discovery of the remaining instances.	Fix any credential failures.
DLPX-78589	During the upgrade, when Delphix was trying to run the ASE "UNMOUNT" command while trying to quiesce VDBs, UNMOUNT command got hung as this command doesn't run under a timeout and due to this upgrade job stalled.	When the ASE UNMOUNT command is hung, the ASE instance must be restarted.
DLPX-81478	If a transaction log is taken using the standby_access option for a SAP ASE database, the validated sync worker will not be able to restore that log, and will fail with Delphix fatal exception (as Delphix currently does not support this option).	Change the backup script creating transaction logs to Ingest FULL backups
DLPX-81559	After upgrading an Oracle VDB from 12c to 19c, VDB refresh fails with "Failed to mount database instance" due to ORA-01130 or ORA-00201 errors.	Update the VDB parameters (either directly or via a VDB Config Template). See this knowledge base article.
DLPX-83643	After engine upgrade or after disable and enable of vPDB, NFS version in GUI/API may be incorrect if - it is a vPDB in a vCDB, it was mounted with NFSv3 and now the host supports NFSv4 mounts. This condition corrects itself on the next refresh operation.	Perform a vPDB refresh operation to update the NFS protocol version to 4.
DLPX-84253	Users should be able to delete Delphix-generated CA certificate if no dependencies exist.	Customer may mark the fault Ignored, but will need to be aware the CA certificate will still appear in Truststore.

Key	Summary	Workaround
DLPX-84598	A sysadmin user cannot view actions initiated by a different sysadmin user.	None
DLPX-84710	Improved handling of missing networking devices after instance type migration	None
DLPX-86181	Provision, refresh, rewind or start operations on a virtual database on a Solaris host with NFSv4 enabled may become stuck indefinitely due to mount process getting stuck on the Solaris NFS client.	Switch the NFS protocol to v3 using the Delphix engine CLI, manually terminate the stuck mount processes on the Solaris host and re-attempt the failed Delphix operation.
DLPX-87213	When relocating multiple terabytes of data either via block to object storage migration or device removal, management of the engine can become unresponsive for a lengthy period of time (e.g. 30+ mins), dependent on the dataset size and workload.	None, the data relocation and concurrent IO workloads will continue in the background. If the engine cannot be accessed after one hour, contact Delphix Support for corrective actions.
DLPX-87320	Exporting a VDB or vPDB to ASM will fail with ORA-32771 when the database has a bigfile temporary tablespace.	If using the move-to-asm script, use an older version of script from Delphix engine v6.0.15.0 or earlier. No workaround if attempting to export a VDB or vPDB to a physical ASM or Exadata database when the database has a bigfile temporary tablespace using the database export CLI.
DLPX-87702	Provisioning an Oracle VDB/vPDB fails with "ORA-03214: File Size specified is smaller than minimum required" when tempfiles have a non-default local uniform size.	Recreate temp files in the source database with size smaller than 50MB and take a new snapshot.

3.3.10 Version 14.0.0.0

Key	Summary	Workaroound
DLPX-56944	In Solaris 5.11 environments where Delphix OS user shell is set to `csh`, host parameters may be improperly parsed, leading to environment refresh and other job failures.	Contact Delphix Support for corrective actions.
DLPX-56978	Certain usernames are not available for use, as they are reserved system words (e.g. "root", "postgres", "delphix").	None
DLPX-56979	LDAP server test fails if authentication is set to DIGEST-MD5, but setup still works correctly.	None
DLPX-57142	The Job dashboard does not display the user that invoked each Job.	None
DLPX-57412	CLI parameters must be used exactly as described in documentation, including spelling and capitalization, or they are ignored.	None
DLPX-57673	Support bundle generation can be time consuming if the engine has a large number of core files to process.	None
DLPX-57823	Changing the Compliance Engine used by a VDB's masking job can lead to an internal error.	None
DLPX-58185	Changing a Custom Policy while it is running on an object can cause the Policy execution to fail.	None
DLPX-58226	Completed cleanup jobs may still show as running, even after an upgrade.	None

Key	Summary	Workaroound
DLPX-59473	It is not possible to set a password policy that prevents an administrator from re-using a previous password, though this can be set for other users.	None
DLPX-66155	Restarting a Delphix Engine during a network throughput test is not recommended as it may lead to a system hang.	None
DLPX-66860	For SSO/SAML, ADFS requires explicit rule to transform `emailAddress` attribute into `nameid`.	Create an explicit rule in ADFS that transforms the `emailAddress` attribute into a `nameid`. The rule type must be "Transform an incoming claim". The incoming claim type must be "Email address" and the outgoing claim type "Name ID". The `nameid` format must be "Email address".
DLPX-77849	An excessive number of connections to the SQL Server instance can cause infrastructure issues, leading the LSASS.exe to crash and the host to reboot.	Rather than using a Windows domain user for authentication, switch the dSource to use a database user. It seems SQL Server may be able to handle the massive number of connections that Delphix is establishing for each dSource better than Windows LSASS.EXE.
DLPX-77986	On a sybase host with multiple Sybase instances, If access to the first instance picked up for discovery fails due to invalid credentials, the discovery job will exit immediately preventing discovery of the remaining instances.	Fix any credential failures.
DLPX-78589	During the upgrade, when Delphix was trying to run the ASE "UNMOUNT" command while trying to quiesce VDBs, UNMOUNT command got hung as this command does not run under a timeout and due to this, the upgrade job stalled.	When the ASE UNMOUNT command is hung, the ASE instance must be restarted.

Key	Summary	Workaroound
DLPX-80193	A vague error message appears in a case where provisioning fails due to a database being in read-only mode.	Change source database to read-write, create a backup and sync to the dSource and provision VDB.
DLPX-81300	Windows Environment Add/Refresh operations may fail if the iSCSI Initiator Name is not a valid IQN.	Set the initiator name to a valid name, e.g. iqn.1991-05.com.microsoft:10-43-47-42.qa-ad.delphix.com. The “Default” button can also be selected in Windows iSCSI configuration to reset to the default (valid) name.
DLPX-81478	If a transaction log is taken using the `standby_access` option for a SAP ASE database, the validated sync worker will not be able to restore that log, and will fail with Delphix fatal exception (as Delphix currently does not support this option).	Change the backup script creating transaction logs to Ingest FULL backups.
DLPX-81559	After upgrading an Oracle VDB from 12c to 19c, VDB refresh fails with "Failed to mount database instance" due to ORA-01130 or ORA-00201 errors.	Update the VDB parameters (either directly or via a VDB Config Template). See this knowledge base[74] article.
DLPX-83643	After engine upgrade or after disable and enable of vPDB, NFS version in GUI/API may be incorrect; for a vPDB in a vCDB, it was mounted with NFSv3 and now the host supports NFSv4 mounts. This condition corrects itself on the next refresh operation.	Perform a vPDB refresh operation to update the NFS protocol version to 4.
DLPX-84075	SQL Server VDB Refresh operations may fail if the PowerShell command `[IO.Path]::GetTempFileName()` returns no value.	Go to the temp directory below and delete all files from it: `C:\Window\ServiceProfiles\NetworkService\appdata\local\temp`
DLPX-84253	Users should be able to delete Delphix-generated CA certificate if no dependencies exist.	Mark related faults as Ignored, but note that the CA certificate will still appear in Truststore.

74 https://support.delphix.com/Continuous_Data_Engine_(formerly_Virtualization_Engine)/Oracle/How_to_Fix_ORA-01130_or_ORA-00201_Errors_When_Provisioning_an_Oracle_VDB_(KBA6234)

Key	Summary	Workaroound
DLPX-84598	A sysadmin user cannot view actions initiated by a different sysadmin user.	None
DLPX-84710	Improved handling of missing networking devices after instance type migration required.	None
DLPX-86181	Provision, refresh, rewind, or start operations on a virtual database on a Solaris host with NFSv4 enabled may become stuck indefinitely due to mount process getting stuck on the Solaris NFS client.	Switch the NFS protocol to v3 using the Delphix engine CLI, manually terminate the stuck mount processes on the Solaris host and re-attempt the failed Delphix operation.
DLPX-86894	After a canceled refresh operation, VDB refresh or delete may fail due to error code `exception.oracle.vdb.no.virtual.datafiles.found`. As a workaround, disable the VDB and try the operation again.	Disable the VDB and re-attempt the failed operation.
DLPX-87320	Exporting a VDB or vPDB to ASM fails with "ORA-32771: cannot add file to bigfile tablespace" when the database has a bigfile temporary tablespace	If using the `move-to-asm` script, use an older version of script from Delphix engine v6.0.15.0 or earlier. No workaround if attempting to export a VDB or vPDB to a physical ASM or Exadata database using the `database export` CLI.

3.3.11 Version 13.0.0.0

Key	Summary	Workaround
DLPX-56944	In Solaris 5.11 environments where the Delphix OS user shell is set to `csh`, host parameters may be improperly parsed, leading to ‘environment refresh’ and other job failures.	Contact Delphix Support for corrective actions.

DLPX-56978	Certain usernames are not available for use, as they are reserved system words (e.g. "root", "postgres", "delphix").	None
DLPX-56979	LDAP server test fails if authentication is set to DIGEST-MD5, but setup still works correctly.	None
DLPX-57142	The Job dashboard does not display the user that invoked each job.	None
DLPX-57412	CLI parameters must be used exactly as described in documentation, including spelling and capitalization, or they are ignored.	None
DLPX-57673	Support bundle generation can be time consuming if the engine has a large number of core files to process.	None
DLPX-57823	Changing the Compliance engine used by a VDB's masking job can lead to an internal error.	None
DLPX-58185	Changing a Custom Policy while it is running on an object can cause the Policy execution to fail.	None
DLPX-58226	Completed cleanup jobs may still show as running, even after an upgrade.	None
DLPX-59473	It is not possible to set a password policy that prevents an Administrator from re-using a previous password, though this can be set for other users.	None
DLPX-66155	Restarting a Delphix Engine during a network throughput test is not recommended, as it may lead to a system hang.	None

DLPX-66860	For SSO/SAML, ADFS requires an explicit rule to transform the emailAddress attribute into nameid.	Create an explicit rule in ADFS that transforms the emailAddress attribute into a nameid. The rule type must be "Transform an incoming claim". The incoming claim type must be "Email address" and the outgoing claim type is "Name ID". The nameid format must be "Email address".
DLPX-77849	An excessive number of connections to the SQL Server instance can cause infrastructure issues, leading the LSASS.exe to crash and the host to reboot.	Rather than using a Windows domain user for authentication, switch the dSource to use a database user. It seems SQL Server may be able to handle the massive number of connections that Delphix is establishing for each dSource better than Windows LSASS.exe.
DLPX-77986	On a Sybase host with multiple Sybase instances, if access to the first instance picked up for discovery fails due to invalid credentials, the discovery job will exit immediately (preventing discovery of the remaining instances).	Fix any credential failures.
DLPX-78589	During an upgrade, when Delphix tried to run the ASE "UNMOUNT" command, while trying to quiesce VDBs, the UNMOUNT command got hung (as this command doesn't run under a timeout); due to this, the upgrade job stalled.	When the ASE UNMOUNT command is hung, the ASE instance must be restarted.
DLPX-80193	Vague error message appears in a case where provisioning fails due to a database being in read-only mode.	Change source the database to read-write, create a backup, sync to the dSource, and provision VDB.
DLPX-81300	Windows Environment Add/Refresh operations may fail if the iSCSI Initiator Name is not a valid IQN.	Set the initiator name to a valid name, eg: iqn.1991-05.com.microsoft:10-43-47-42.qa-ad.delphix.com. The “Default” button can also be selected in the Windows iSCSI configuration to reset to the default (valid) name.

DLPX-81478	If a transaction log is taken using the `standby_access` option for an SAP ASE database, the validated sync worker will not be able to restore that log and will fail with a Delphix fatal exception (as Delphix currently does not support this option).	Change the backup script creating transaction logs to ingest FULL backups.
DLPX-81559	After upgrading an Oracle VDB from 12c to 19c, a VDB refresh fails with, "Failed to mount database instance" due to ORA-01130 or ORA-00201 errors.	Update the VDB parameters (either directly or via a VDB Config Template). See this knowledge base[75] article.
DLPX-83643	After engine upgrade or after disable and enable of a vPDB, the NFS version in the GUI/API may be incorrect if it's a vPDB in a vCDB; this means it was mounted with NFSv3, and now the host supports NFSv4 mounts. This condition corrects itself on the next refresh operation.	Perform a vPDB refresh operation to update the NFS protocol version to 4.
DLPX-84075	SQL Server VDB Refresh operations may fail if the PowerShell command `[IO.Path]::GetTempFileName()` returns no value.	Go to the following temp directory and delete all the files from it: `C:\Window\ServiceProfiles\NetworkService\appdata\local\temp`
DLPX-84253	Unable to delete Delphix-generated CA certificate if no dependencies exist.	Mark the fault Ignored, but the CA certificate will still appear in Truststore.
DLPX-84598	A sysadmin user cannot view actions initiated by a different sysadmin user.	None
DLPX-84710	Improved handling of missing networking devices after instance type migration required.	None

75 https://support.delphix.com/Continuous_Data_Engine_(formerly_Virtualization_Engine)/Oracle/How_to_Fix_ORA-01130_or_ORA-00201_Errors_When_Provisioning_an_Oracle_VDB_(KBA6234)

DLPX-85770	If a Oracle VDB/vPDB is already enabled, a Self-Service container `start` operation or the VDB/vPDB `enable` operation may fail with `exception.oracle.vdb.database.exists.enable.not.allowed` / `exception.oracle.vdb.pdb.exists.enable.not.allowed`.	Retry the failed VDB/vPDB start or self-service container start operation, it should succeed without any error.
DLPX-86181	Provision, refresh, rewind or start operations on a virtual database on a Solaris host with NFSv4 enabled may become stuck indefinitely due to the mount process getting stuck on the Solaris NFS client.	Switch the NFS protocol to v3 using the Delphix engine CLI, manually terminate the stuck mount processes on the Solaris host and re-attempt the failed Delphix operation.
DLPX-87006	Refresh operation fails for the self service containers having ordered sources.	None. Contact Delphix support for corrective actions.

3.3.12 Version 12.0.0.0

Key	Summary	Workaround
DLPX-56944	In Solaris 5.11 environments where the Delphix OS user shell is set to csh, host parameters may be improperly parsed, leading to environment refresh and other job failures.	Contact Delphix Support for corrective actions.
DLPX-56978	Certain usernames are not available for use, as they are reserved system words (e.g. "root", "postgres", "delphix").	None
DLPX-56979	LDAP server test fails if authentication is set to DIGEST-MD5, but setup still works correctly.	None
DLPX-57142	The Job dashboard does not display the user that invoked each Job.	None

Key	Summary	Workaround
DLPX-57412	CLI parameters must be used exactly as described in documentation, including spelling and capitalization, or they are ignored.	None
DLPX-57673	Support bundle generation can be time consuming if the engine has a large number of core files to process.	None
DLPX-57823	Changing the masking engine used by a VDB's masking job can lead to an internal error.	None
DLPX-58185	Changing a Custom Policy while it is running on an object can cause the Policy execution to fail.	None
DLPX-58226	A completed Cleanup Job, after upgrade, still shows as "running".	None
DLPX-59473	It is not possible to set a password policy that prevents an Administrator from re-using a previous password, though this can be set for other users.	None
DLPX-66155	Restarting a Delphix Engine during a network throughput test is not recommended, as it may lead to a system hang.	None
DLPX-66860	For SSO/SAML, ADFS requires explicit rule to transform emailAddress attribute into nameid.	Create an explicit rule in ADFS that transforms the emailAddress attribute into a nameid. The rule type must be "Transform an incoming claim". The incoming claim type must be "Email address" and the outgoing claim type is "Name ID". The nameid format must be "Email address".

Key	Summary	Workaround
DLPX-77849	An excessive number of connections to SQL Server Instances causes infrastructure issues and leads LSASS.exe to crash, and host to reboot.	Rather than using a Windows domain user for authentication, switch the dSource to use a database user. It seems SQL Server may be able to handle the massive number of connections that Delphix is establishing for each dSource better than Windows LSASS.EXE.
DLPX-77986	On a sybase host with multiple Sybase instances, If access to the first instance picked up for discovery fails due to invalid credentials, the discovery job will exit immediately preventing discovery of the remaining instances.	Fix any credential failures
DLPX-78589	During the upgrade, when Delphix was trying to run the ASE "UNMOUNT" command while trying to quiesce VDBs, the UNMOUNT command gets hung (as this command doesn't run under a timeout and due to this, the upgrade job stalled).	When the ASE UNMOUNT command is hung, the ASE instance must be restarted.
DLPX-80193	Provide a proper error message in case provisioning fails due to database being in read-only mode.	Change source database to read-write, create a backup and sync to the dSource, and provision VDB.
DLPX-80920	A failed Delphix Engine upgrade can cause plugin operations to fail with "grpc_status 14".	Verify the upgrade and then apply the upgrade. If the issue occurs after a failed upgrade, restarting the management stack will resolve the issue.
DLPX-81300	Windows environment Add/Refresh operations may fail if the iSCSI Initiator Name is not a valid IQN.	Set initiator name to a valid name, eg: iqn.1991-05.com.microsoft:10-43-47-42.qa-ad.delphix.com The “Default” button can also be selected in the windows iSCSI configuration to reset to the default (valid) name.
DLPX-81478	If a transaction log is taken using the standby_access option for a SAP ASE database, the validated sync worker will not be able to restore that log, and will fail with Delphix fatal exception (as Delphix currently does not support this option).	Change the backup script creating transaction logs to ingest FULL backups.

Key	Summary	Workaround
DLPX-81559	After upgrading an Oracle VDB from 12c to 19c, VDB refresh fails with "Failed to mount database instance", due to ORA-01130 or ORA-00201 errors.	Update the VDB parameters (either directly or via a VDB Config Template). See this knowledge base[76] article.
DLPX-83643	After an engine upgrade or after the disable/enable of a vPDB, the NFS version in the GUI/API may be incorrect if it's a vPDB in a vCDB or it was mounted with NFSv3 (and now the host supports NFSv4 mounts). This condition corrects itself on the next refresh operation.	Perform a vPDB refresh operation to update the NFS protocol version to 4.
DLPX-84075	SQL Server VDB Refresh operations may fail if the PowerShell command [IO.Path]::GetTempFileName() returns no value.	Go to the below mentioned temp directory, and delete all the files from it: C:\Window\ServiceProfiles\NetworkService\appdata\local\temp.
DLPX-84253	Users should be able to delete the Delphix-generated CA certificate, if no dependencies exist.	Users may mark the fault Ignored, but will need to be aware the CA certificate will still appear in Truststore.
DLPX-84598	Sysadmin users cannot view actions initiated by different sysadmin users.	None
DLPX-84710	Improved handling of missing networking devices after instance type migration.	None
DLPX-85469	JSON file masking does not support the use of a multi-column algorithm on (a) Fields in two or more different arrays (b) Fields at different levels in a single multi-dimensional array.	None
DLPX-85770	If an Oracle VDB/vPDB is already enabled, a self-service container start operation or the VDB/vPDB enable operation may fail with: exception.oracle.vdb.database.exists.enable.not.allowed/ exception.oracle.vdb.pdb.exists.enable.not.allowed	Retry the failed VDB/vPDB start or self-service container start operation, it should succeed without any error.

76 https://support.delphix.com/Continuous_Data_Engine_(formerly_Virtualization_Engine)/Oracle/How_to_Fix_ORA-01130_or_ORA-00201_Errors_When_Provisioning_an_Oracle_VDB_(KBA6234)

Key	Summary	Workaround
DLPX-86109	Oracle VDB unquiesce/enable may fail after a failed quiesce/disable on 10.0.0.X or 11.0.0.X.	To get disabled Oracle VDBs back to an enabled state, manually disable sources via the Delphix CLI and enable them back via the Delphix CLI/UI.
DLPX-86181	Provision, refresh, rewind, or start operations on a virtual database on a Solaris host with NFSv4 enabled may become stuck indefinitely due to mount process getting stuck on the Solaris NFS client.	Switch the NFS protocol to v3 using the Delphix engine CLI, manually terminate the stuck mount processes on the Solaris host and re-attempt the failed Delphix operation.
DLPX-86344	A failed Delphix Engine upgrade can cause plugin operation to fail with grpc_status 14.	Verify the upgrade and then apply the upgrade. If the issue is hit after a failed upgrade, Restarting the management stack will resolve the issue.
DLPX-86842	If there is any detached or unlinked Oracle dSource, upgrade failure may occur during `verify upgrade` job while upgrading to 12.0.0.0.	Re-link or delete the dSources before trying to upgrade to 12.0.0.0, OR upgrade to version 11.0.0.0 or 13.0.0.0.
DLPX-87006	Refresh operation fails for the self service containers having ordered sources.	None. Contact Delphix support for corrective actions.

3.3.13 Version 11.0.0.0

Key	Summary	Workaround
DLPX-56978	Certain usernames are not available for use, as they are reserved system words (e.g. "root", "postgres", "delphix").	None
DLPX-56979	LDAP server test fails if authentication is set to DIGEST-MD5, but setup still works correctly.	None
DLPX-57142	The Job dashboard does not display the user that invoked each job.	None

Key	Summary	Workaround
DLPX-57412	CLI parameters must be used exactly as described in documentation, including spelling and capitalization, or they are ignored.	None
DLPX-57673	Support bundle generation can be time consuming if the engine has a large number of core files to process.	None
DLPX-57823	Changing the Continuous Compliance engine used by a VDB's masking job can lead to an internal error.	None
DLPX-58185	Changing a Custom Policy while it is running on an object can cause the Policy execution to fail.	None
DLPX-58226	Completed Cleanup Job, after upgrade, will still show as running.	None
DLPX-59473	It is not possible to set a password policy that prevents an Administrator from re-using a previous password, though this can be set for other users.	None
DLPX-66155	Restarting a Delphix engine during a network throughput test is not recommended, as it may lead to a system hang.	None
DLPX-66860	For SSO/SAML, ADFS requires an explicit rule to transform emailAddress attribute into nameid.	Create an explicit rule in ADFS that transforms the emailAddress attribute into a nameid. The rule type must be "Transform an incoming claim". The incoming claim type must be "Email address" and the outgoing claim type "Name ID". The nameid format must be "Email address".

Key	Summary	Workaround
DLPX-77849	Excessive number of connections to SQL Server instance causes infrastructure issues and leads the LSASS.exe to crash and the host to reboot.	Rather than using a Windows domain user for authentication, switch the dSource to use a database user. SQL Server should be able to handle the massive number of connections that Delphix is establishing for each dSource better than LSASS.EXE.
DLPX-77986	On a Sybase host with multiple Sybase instances, if access to the first instance picked up for discovery fails due to invalid credentials, the discovery job will exit immediately, preventing discovery of the remaining instances.	Fix any credential failures.
DLPX-78589	When Delphix tries to run the ASE "UNMOUNT" command (while trying to quiesce VDBs) during the upgrade, the UNMOUNT command gets hung up (this command does not have a timeout). Because of this, the upgrade job stalled.	When the ASE UNMOUNT command is hung up, the ASE instance must be restarted.
DLPX-80193	Provides a proper error message in case provisioning fails due to database being in read-only mode.	Change the source database to read-write, create a backup, and sync to the dSource, then provision the VDB.
DLPX-80920	A failed Delphix engine upgrade can cause plugin operation to fail with "grpc_status 14".	Verify the upgrade and then apply the upgrade. If the issue occurs after a failed upgrade, restarting the management stack will resolve the issue.
DLPX-81300	Windows Environment Add/Refresh operations may fail if the iSCSI Initiator Name is not a valid IQN.	Set the initiator name to a valid name, eg: iqn.1991-05.com.microsoft:10-43-47-42.qa-ad.delphix.com. The “Default” button can also be selected in the Windows iSCSI configuration to reset to the default (valid) name.
DLPX-81478	If the transaction log is taken using the `standby_access` option for a SAP ASE database, the validated sync worker will not be able to restore that log and will fail with a fatal exception, as Delphix currently does not support this option.	Change the backup script creating transaction logs. Ingest FULL backups.

Key	Summary	Workaround
DLPX-81559	After upgrading an Oracle VDB from 12c to 19c, VDB refresh fails with "Failed to mount database instance", due to ORA-01130 or ORA-00201 errors.	Update the VDB parameters (either directly or via a VDB config template). See this knowledge base article[77]
DLPX-83430	Initial configuration of Syslog breaks most of the pre-existing appenders.	Restart the management stack.
DLPX-83643	After an engine upgrade or after disable and enable of vPDB, the NFS version in the GUI/API may be incorrect if it's a vPDB in a vCDB, it was mounted with NFSv3 and now the host supports NFSv4 mounts. This condition corrects itself on the next refresh operation.	Perform a vPDB refresh operation to update the NFS protocol version to 4.
DLPX-84075	SQL Server VDB Refresh operations may fail if the PowerShell command [IO.Path]::GetTempFileName() returns no value.	Go to the below mentioned temp directory and delete all files from it: C:\Window\ServiceProfiles\NetworkService\appdata\local\temp.
DLPX-84253	Users should be able to delete Delphix-generated CA certificate if no dependencies exist.	User may mark the fault as ignored, but will need to be aware the CA certificate will still appear in TrustStore.
DLPX-84598	A sysadmin user cannot view actions initiated by different sysadmin user	None
DLPX-85578	Replaced the Win32_Volume class output with the mountvol output to fetch volumeId for Delphix iSCSI mount points.	None
DLPX-85770	If an Oracle VDB/vPDB is already enabled, a Self-Service container start operation or the VDB/vPDB enable operation may fail with exception.oracle.vdb.database.exists.enable.not.allowed/ exception.oracle.vdb.pdb.exists.enable.not.allowed.	Retry the failed VDB/vPDB start or Self-Service container start operation, it should succeed without any error.

77 https://support.delphix.com/Continuous_Data_Engine_(formerly_Virtualization_Engine)/Oracle/How_to_Fix_ORA-01130_or_ORA-00201_Errors_When_Provisioning_an_Oracle_VDB_(KBA6234)

Key	Summary	Workaround
DLPX-86109	Oracle VDB unquiesce/enable may fail after a failed quiesce/disable on 10.0.0.X or 11.0.0.X.	To get disabled Oracle VDBs back to Enabled state, manually disable such sources via the Delphix CLI and enable them back via the Delphix CLI/UI.
DLPX-86181	Provision, refresh, rewind, or start operations on a virtual database on a Solaris host with NFSv4 enabled may become stuck indefinitely due to the mount process getting stuck on the Solaris NFS client.	Switch the NFS protocol to v3 using the Delphix engine CLI, manually terminate the stuck mount processes on the Solaris host and re-attempt the failed Delphix operation.

3.3.14 Version 10.0.0.0

Key	Summary	Workaround
DLPX-56944	In Solaris 5.11 Environments where Delphix OS user shell is set to csh, host parameters may be improperly parsed leading to Environment refresh and other job failures.	Contact Delphix Support for corrective actions.
DLPX-56978	Certain usernames are not available for use as they are reserved system words (e.g. "root", "postgres", "delphix")	None
DLPX-56979	LDAP server test fails if authentication is set to DIGEST-MD5, but setup still works correctly	None
DLPX-57142	The Job dashboard does not display the user that invoked each Job	None
DLPX-57412	CLI parameters must be used exactly as described in documentation, including spelling and capitalization, or they are ignored	None
DLPX-57673	Support bundle generation can be time consuming if the engine has a large number of core files to process	None

Key	Summary	Workaround
DLPX-57823	Changing the masking engine used by a VDB's masking job can lead to an internal error	None
DLPX-58185	Changing a Custom Policy while it is running on an object can cause the Policy execution to fail	Disable/enable after fixing the problem
DLPX-58226	Completed Cleanup Job, After Upgrade, Still Shows as Running	None
DLPX-59473	It is not possible to set a password policy that prevents an Administrator from re-using a previous password, though this can be set for other users	None
DLPX-66155	Restarting a Continuous Data Engine during a network throughput test is not recommended as it may lead to a system hang.	Create an explicit rule in ADFS that transforms the emailAddress attribute into a nameid. The rule type must be "Transform an incoming claim". The incoming claim type must be "Email address" and the outgoing claim type "Name ID". The nameid format must be "Email address".
DLPX-66860	For SSO/SAML, ADFS requires explicit rule to transform emailAddress attribute into nameid.	Rather than using a Windows domain user for authentication switch the dSource to use a database user. It seems SQL Server may be able to handle the massive number of connections that Continuous Data is establishing for each dSource better than Windows LSASS.EXE.
DLPX-77849	Excessive number of connections to SQL Server Instance observed by Delphix Environment User. This causes infra issues and leads to LSASS.exe to crash and host to reboot	Rather than using a Windows domain user for authentication switch the dSource to use a database user. It seems SQL Server may be able to handle the massive number of connections that Delphix is establishing for each dSource better than Windows LSASS.EXE.

Key	Summary	Workaround
DLPX-77986	On a SAP ASE host with multiple SAP ASE instances, If access to the first instance picked up for discovery fails due to invalid credentials, the discovery job will exit immediately preventing discovery of the remaining instances.	Fix any credential failures
DLPX-78589	During the upgrade, when Delphix was trying to run the SAP ASE "UNMOUNT" command while trying to quiesce VDBs, UNMOUNT command hung as this command doesn't run under a timeout and due to this upgrade job stalled.	When the ASE UNMOUNT command is hung, the ASE instance must be restarted.
DLPX-80193	Provide proper error message in case provisioning fails due to database is in read-only mode.	Change source database to read-write, create a backup and sync to the dSource and provision VDB.
DLPX-80920	A failed Delphix engine upgrade can cause plugin (for plugins using docker runtime) operation to fail with grpc_status 14.	1. Verify the upgrade and then apply the upgrade. 2. If the issue is hit after a failed upgrade, Restarting the management stack will resolve the issue.
DLPX-81300	Windows Environment Add/Refresh operations may fail if the iSCSI Initiator Name is not a valid IQN	Set initiator name to a valid name, eg: iqn.1991-05.com.microsoft:10-43-47-42.qa-ad.delphix.com. The “Default” button can also be selected in windows iscsi configuration to reset to the default (valid) name.
DLPX-81478	If transaction log is taken using `standby_access` option for SAP ASE database, the validated sync worker will not be able to restore that log and will fail with Delphix Fatal Exception as Delphix currently doesnt support this option.	Change the backup script creating transaction logs Ingest FULL backups

Key	Summary	Workaround
DLPX-81559	After upgrading an Oracle VDB from 12c to 19c, the VDB refresh fails with "Failed to mount database instance" due to ORA-01130 or ORA-00201 errors.	Update the VDB parameters (either directly or via a VDB Config Template). For more information, please see KBA6234[78]
DLPX-83643	If you upgrade the engine or disable and then enable a virtual PDB, the displayed NFS version in the GUI or API might be incorrect. This happens if the virtual PDB is inside a virtual CDB and was previously mounted with NFSv3, but the host now supports NFSv4 mounts. However, the issue will automatically resolve itself during the next refresh operation.	Perform a vPDB refresh operation to update the NFS protocol version to 4.
DLPX-84075	SQL Server VDB Refresh operations may fail if the PowerShell command [IO.Path]::GetTempFileName() returns no value"	Go to the below mentioned temp directory, and delete all the files from it: C:\Window\ServiceProfiles\NetworkService\appdata\local\temp
DLPX-84598	sysadmin user cannot view actions initiated by different sysadmin user	Perform an export with the same unique name as of the VDB and then manually change the physical database's unique name.
DLPX-85493	MSSQL: Linking and AttachSource operations will fail with unnecessary permissions of source user on the staging host and staging database. This issue can be seen from 6.0.17.0 onwards.	Adding source user to the staging instance with read permission on master database and Giving write permission on connector directory path {Connector_Installation}/sourceValidation

78 https://support.delphix.com/Continuous_Data_Engine_(formerly_Virtualization_Engine)/Oracle/How_to_Fix_ORA-01130_or_ORA-00201_Errors_When_Provisioning_an_Oracle_VDB_(KBA6234)

3.3.15 Version 9.0.0.0

Key	Summary	Workaround
DLPX-77849	Excessive number of connections to SQL Server Instance observed by Delphix Environment User. This causes infra issues and leads to LSASS.exe to crash and host to reboot	Rather than using a Windows domain user for authentication switch the dSource to use a database user. It seems SQL Server may be able to handle the massive number of connections that Delphix is establishing for each dSource better than Windows LSASS.EXE.
DLPX-84977	For RAC environments, export of an Oracle VDB to a database with a new unique name is not supported.	Perform the export with the same unique name as of the VDB and then manually change the physical database's unique name.
DLPX-84423	Export operation does not retain the read-only mode in the newly converted physical database after an in-place conversion of a read-only Oracle VDB or vPDB	None
DLPX-84598	sysadmin user cannot view actions initiated by different sysadmin user	None
DLPX-84655	A false warning message that listener registration was not successful is posted when enabling a VDB or a vCDB, users can ignore this message.	Ignore the warning message.
DLPX-84679	Unable to add Staging push PDB if Staging Environment has more than one repository.	From the Add dSource Wizard dSource Configuration screen for Staging Push. • Before selecting PDB as the Database Type, select CDB. • Select the correct Staging Environment and repository. • Select PDB for the Database Type. • Now CDB will be shown under Container Database • Fill in the remaining PDB details

Key	Summary	Workaround
DLPX-85076	Instance init file is not copied to $ORACLE_BASE_CONFIG/dbs during staging push dSource linking	Specify the pfile parameter in the startup command as `pfile='/<MOUNT_BASE>/<DATABASE_UNIQUE_NAME>/script/<DATABASE_SID>/init<DATABASE_SID>.ora'`

3.3.16 Version 8.0.0.0

Key	Summary	Workaround
DLPX-77849	Excessive number of connections to SQL Server Instance observed by Delphix Environment User. This causes infra issues and leads to LSASS.exe to crash and host to reboot	Rather than using a Windows domain user for authentication switch the dSource to use a database user. It seems SQL Server may be able to handle the massive number of connections that Delphix is establishing for each dSource better than Windows LSASS.EXE.
DLPX-84679	Unable to add Staging push PDB if Staging Environment has more than one repository.	From the Add dSource Wizard dSource Configuration screen for Staging Push. • Before selecting PDB as the Database Type, select CDB. • Select the correct Staging Environment and repository. • Select PDB for the Database Type. • Now CDB will be shown under Container Database • Fill in the remaining PDB details
DLPX-85076	Instance init file is not copied to $ORACLE_BASE_CONFIG/dbs during staging push dSource linking	Specify the pfile parameter to startup command as `pfile parameter=/<MOUNT_BASE>/<DATABASE_UNIQUE_NAME>/script/<DATABASE_SID>/init<DATABASE_SID>.ora`

3.3.17 Version 7.0.0.0

Upgrades from versions < 6.0.17.0 to any version between 6.0.17.0 and 7.0.0.0 on a replication target engine may fail due to the management services being down, which will require a support call. This applies to “Delay the Reboot” or “Apply Now” upgrades.

Key	Summary	Workaround
DLPX-77849	Excessive number of connections to SQL Server Instance observed by Delphix Environment User. This causes infra issues and leads to LSASS.exe to crash and host to reboot	Rather than using a Windows domain user for authentication switch the dSource to use a database user. It seems SQL Server may be able to handle the massive number of connections that Delphix is establishing for each dSource better than Windows LSASS.EXE.
DLPX-84255	In a Single Engine Continuous Vault product, adding a new Sybase dSource to a locked group may result in the background environment monitoring process to stop working.	No workaround.
DLPX-84495	If upgrading from versions < 6.0.17.0 to any version between 6.0.17.0 and 7.0.0.0, a support call is needed.	Call support.
DLPX-84679	Unable to add Staging push PDB if Staging Environment has more than one repository	From the Add dSource Wizard dSource Configuration screen for Staging Push. • Before selecting PDB as the Database Type, select CDB. • Select the correct Staging Environment and repository. • Select PDB for the Database Type. • Now CDB will be shown under Container Database • Fill in the remaining PDB details

Key	Summary	Workaround
DLPX-85076	Instance init file is not copied to $ORACLE_BASE_CONFIG/dbs during staging push dSource linking	Specify the pfile parameter to startup command as `pfile parameter=/<MOUNT_BASE>/<DATABASE_UNIQUE_NAME>/script/<DATABASE_SID>/init<DATABASE_SID>.ora`

3.3.18 Version 6.0.17.0

Key	Summary	Workaround
DLPX-77849	Excessive number of connections to SQL Server Instance observed by Delphix Environment User. This causes infra issues and leads to LSASS.exe to crash and host to reboot	Rather than using a Windows domain user for authentication switch the dSource to use a database user. It seems SQL Server may be able to handle the massive number of connections that Delphix is establishing for each dSource better than Windows LSASS.EXE.
DLPX-66860	For SSO/SAML, ADFS requires explicit rule to transform emailAddress attribute into nameid.	Create an explicit rule in ADFS that transforms the {{emailAddress}} attribute into a {{nameid}}. The rule type must be "Transform an incoming claim". The incoming claim type must be "Email address" and the outgoing claim type "Name ID". The nameid format must be "Email address".
DLPX-83622	In 6.0.15.0 and 6.0.16.0, Fluentd gems from plugins were installed in the base Fluentd. These left over gems can cause Fluentd to fail post-upgrade.	No workaround.
DLPX-83643	After engine upgrade or after a disable and enable of a vPDB, the NFS version in GUI/API may be incorrect (if it's a vPDB in a vCDB, it was mounted with NFSv3 and now the host supports NFSv4 mounts). This condition corrects itself on the next refresh operation.	Perform a vPDB refresh operation to update the NFS protocol version to 4.

Key	Summary	Workaround
DLPX-83575	The port in the connection string for a vPDB in a Linked CDB may be shown incorrectly when it is registered to a non-default listener.	Use the port in the connection string of the CDB to connect.

3.3.19 Version 6.0.16.0

Key	Summary	Workaround
DLPX-77849	Excessive number of connections to SQL Server Instance observed by Delphix Environment User. This causes infra issues and leads to LSASS.exe to crash and host to reboot	Rather than using a Windows domain user for authentication switch the dSource to use a database user. It seems SQL Server may be able to handle the massive number of connections that Delphix is establishing for each dSource better than Windows LSASS.EXE.
DLPX-82448	Python2 has been deprecated and is being removed from the platform. During this removal, there are still trace artifacts of legacy Python versions being found on the engine.	No workaround.
DLPX-81559	After upgrading an Oracle VDB from 12c to 19c, the VDB refresh fails with "Failed to mount database instance" due to ORA-01130 or ORA-00201 errors.	Update the VDB parameters (either directly or via a VDB Config Template). For more information, please see KBA6234[79]
DLPX-82169	Provisioning a TDE-enabled vPDB to a CDB with a different patch level than the source CDB will fail.	Patch the dSource to match the target and provision again. If this is not an option, run datapatch on the vPDB, update the timeflow in engine metadata to a CONFIGURED state, restart the vPDB, and run SnapSync. This second option requires an engineering escalation.

79 https://support.delphix.com/Continuous_Data_Engine_(formerly_Virtualization_Engine)/Oracle/How_to_Fix_ORA-01130_or_ORA-00201_Errors_When_Provisioning_an_Oracle_VDB_(KBA6234)

3.3.20 Version 6.0.15.0

Key	Summary	Workaround
DLPX-77849	Excessive number of connections to SQL Server Instance observed by Delphix Environment User. This causes infra issues and leads to LSASS.exe to crash and host to reboot	Rather than using a Windows domain user for authentication switch the dSource to use a database user. It seems SQL Server may be able to handle the massive number of connections that Delphix is establishing for each dSource better than Windows LSASS.EXE.
DLPX-81610	If a tablespace is encrypted before an incremental snapshot is taken, provisioning from that snapshot can lead to the tablespace containing partially unencrypted data, which will not be accessible.	Take a snapshot with parameter "Force Full Backup" and provision a new vPDB/VDB with this snapshot.

3.3.21 Version 6.0.14.0

Key	Summary	Workaround
DLPX-77849	Excessive number of connections to SQL Server Instance observed by Delphix Environment User. This causes infra issues and leads to LSASS.exe to crash and host to reboot	Rather than using a Windows domain user for authentication switch the dSource to use a database user. It seems SQL Server may be able to handle the massive number of connections that Delphix is establishing for each dSource better than Windows LSASS.EXE.
DLPX-80489	TDE-enabled vPDB provisions failing with LOCAL_AUTOLOGIN configuration	Use a regular autologin wallet in the target CDB.
DLPX-80822	vPDB provision from a fully encrypted shared undo parent to local undo CDB results in a vPDB with new UNDO TS which is not encrypted	Encrypt the local undo space manually after provision, or with the Configure Clone hook point.

Key	Summary	Workaround
DLPX-81125	MSSQL Export fails when performed on a target with a Microsoft SQL Server instance running with the network service user	If you update the instance owner of the SQL server, you should refresh the environment to reflect the new user in the engine metadata.
DLPX-81128	Masking: Unable to create/edit a profile set from GUI	Profile set can be add/edited using API

3.3.22 Version 6.0.13.0

Key	Summary	Workaround
DLPX-38908	LogSync should automatically resolve faults for transient issues.	Manually resolve the faults generated when Oracle LogSync encounters transient issues.
DLPX-57078	Job cancel requests during provisioning are not processed until the end of the step.	Find and kill the Oracle `pmon` process for the instance that is being provisioned.
DLPX-64386	LogSync thread hangs when trying to remove temporary RMAN command file from source host toolkit directory.	Remove the `rm` alias for the Delphix OS user.
DLPX-76382	Force disable should succeed despite environmental problems.	No workaround.
DLPX-77849	Excessive number of connections to SQL Server Instance observed by Delphix Environment User. This causes infra issues and leads to LSASS.exe to crash and host to reboot	Rather than using a Windows domain user for authentication switch the dSource to use a database user. It seems SQL Server may be able to handle the massive number of connections that Delphix is establishing for each dSource better than Windows LSASS.EXE.
DLPX-78412	SCM/Talaria failure reason should be communicated in fault.	No workaround.
DLPX-78986	Prevent DSP connections for disabled cluster nodes.	No workaround.

Key	Summary	Workaround
DLPX-79212	While restoring full backups of file stream type database on open staging DB and taking snapshots, snapshots are consuming full space instead of referring to each snapshot.	No workaround.
DLPX-79355	V2P export throws an error when suspended and resumed at the Opening Database step.	Cancel the V2P export job and start the V2P export operation all over again.
DLPX-79502	SnapSync fails if more than 1000 tempfiles exist in the whole CDB.	No workaround.
DLPX-79596	DB files are getting deleted for staging DB if staging push dSource is forced disabled after performing a restore outside Delphix and then enabled again.	No workaround.
DLPX-79833	You can no longer select performance metric resolution for fluentd configurations.	No workaround.
DLPX-80385	Fluentd does not support deferred upgrades when upgrading to 6.0.13.0, and beyond, when starting from a prior release and using the Historic Splunk Insight solution.	Reboot after upgrade.

3.3.23 Version 6.0.12.0

Key	Summary	Workaround
DLPX-67604	Manually recovering a database after V2P from a snapshot of dSource fails with an error.	No workaround.
DLPX-69775	Updating Oracle credentials with an empty string throws an error when Simplified Connection Management is enabled.	Use the Delphix CLI to unset the user.

Key	Summary	Workaround
DLPX-75209	Network configuration is lost when changing EC2 instance type.	On Nitro-based instance types, use the virtual machine's virtual serial console and login to the sysadmin CLI, then add a DHCP address to the network interface.
DLPX-75878	The JDBC connection string for an Oracle vPDB does not get updated after a listener port change.	A second environment refresh will update the connection string for the vPDB.
DLPX-77231	When source discontinuity on the dSource is followed by resync on the livesource, one or more livesource workers may fail to start. This prevents livesource status from getting updated and the first snapshot from being taken after a resync.	No workaround.
DLPX-77849	Excessive number of connections to SQL Server Instance observed by Delphix Environment User. This causes infra issues and leads to LSASS.exe to crash and host to reboot	Rather than using a Windows domain user for authentication switch the dSource to use a database user. It seems SQL Server may be able to handle the massive number of connections that Delphix is establishing for each dSource better than Windows LSASS.EXE.
DLPX-78689	Oracle vPDB snapshot job fails after doing reset logs on a linked CDB.	Disable the vPDB and re-enable it to clear out the snapshot job errors.
DLPX-78700	Oracle vPDB source enable jobs for Oracle 21c are taking more than 15 minutes to complete. This issue is only seen during the first Oracle vPDB enable operation on Oracle 21c.	No workaround.

3.3.24 Version 6.0.11.0

Key	Summary	Workaround
DLPX-44544	A SnapSync of an Oracle standby dSource in Real-Time Apply mode sometimes calculates the snapshot's timestamp incorrectly. This can cause ORA-01194 or ORA-01152 errors when provisioning to a timestamp after the snapshot.	To provision from a snapshot of an Oracle standby dSource in Real-Time Apply mode, provision by SCN instead of timestamp.
DLPX-57971	The latest snapshot of a LiveSource may take a long time to show the SCN/ timestamp range on its card in the GUI.	No workaround.
DLPX-72123	Detaching or deleting an Oracle dSource may fail on RAC environments due to the failure of deletion of RMAN backups on RAC and the operation needs to be retried with the force option.	If the detach or delete operation on an Oracle dSource fails, retry the command using the force option.
DLPX-75209	Network configuration is lost when changing EC2 instance type.	On Nitro-based instance types, use the virtual machine's virtual serial console and login to the sysadmin CLI, then add a DHCP address to the network interface.
DLPX-77664	Oracle SnapSync fails with "RMAN-06183: datafile or datafile copy (file="" number="">>) larger="" than="" maxsetsize"="" if="" a="" datafile="" resized="" in="" the="" middle="" of="">)>	One of the following three workarounds can be applied: 1. Re-issue SnapSync. 2. Reduce the frequency of datafile resizes, or 3. Ensure datafile resize operations are not being performed while the SnapSync operation is in progress.
DLPX-77849	Excessive number of connections to SQL Server Instance observed by Delphix Environment User. This causes infra issues and leads to LSASS.exe to crash and host to reboot	Rather than using a Windows domain user for authentication switch the dSource to use a database user. It seems SQL Server may be able to handle the massive number of connections that Delphix is establishing for each dSource better than Windows LSASS.EXE.

Key	Summary	Workaround
DLPX-77869	Unable to drop and recreate descending order (or any other functional) index as part of Oracle Connector masking/ reidentification/tokenization jobs.	Oracle interprets descending order as functional indexes. There is no workaround. This is a known limitation of Drop Indexes for Oracle connectors that will be resolved in a future release.
DLPX-78015	V2P export with absolute data files is failing with an internal error.	No workaround.
DLPX-78263	A SnapSync of an Oracle standby dSource in Real-Time Apply mode sometimes fails with "exception.oracle.snl.linkedsource.current_scn.invalid" if the rate of change in the database is low.	Force a log switch on all primary instances/ nodes and then try another SnapSync.

3.3.25 Version 6.0.10.0

Key	Summary	Workaround
DLPX-77467	Loading the setup app dashboard (as sysadmin) renders a server error popup with instruction to contact Delphix Support. This message can be ignored. However, it is known that this error impairs the ability to configure web proxy, PhoneHome, and SMTP servers via the GUI.	These settings can still be configured via the CLI. This issue will be fixed in the next version release.
DLPX-64082	Oracle provisioning scripts have hard-coded timeouts.	Retry the provisioning operation to see if it succeeds. Otherwise, contact Delphix Support.
DLPX-72369	RAC migration for VDB with a deleted parent may fail with error "Cannot update RAC instances if virtual source 'xxx' has a deleted parent."	No workaround.
DLPX-74896	Race condition during refresh may result in incorrect engine metadata entry for parent snapshot.	No workaround.

Key	Summary	Workaround
DLPX-75148	DSP throughput tests do not work when from-version is < 6.0.6.0 and target-version is >= 6.0.6.0.	No workaround, however, the user will need to upgrade the source version.
DLPX-75215	Switching AWS instance types can leave the Delphix engine with no network configuration.	On Nitro-based instance types, use the virtual machine's virtual serial console and login to the sysadmin CLI, then add a DHCP address to the network interface.
DLPX-75995	On Windows 2019 Server, with KB4598230 cumulative update, adding or refreshing an environment fails when PowerShell transcription is enabled.	The user has to turn off the transcription if the staging is on Windows 2019 Server with recent updates.
DLPX-77231	When source discontinuity on the dSource is followed by resync on the Oracle LiveSource, one or more LiveSources workers may fail to start. This prevents the LiveSource status from getting updated and the first snapshot from being taken after a resync.	No workaround.
DLPX-77849	Excessive number of connections to SQL Server Instance observed by Delphix Environment User. This causes infra issues and leads to LSASS.exe to crash and host to reboot	Rather than using a Windows domain user for authentication switch the dSource to use a database user. It seems SQL Server may be able to handle the massive number of connections that Delphix is establishing for each dSource better than Windows LSASS.EXE.

3.3.26 Version 6.0.9.0

Key	Summary	Workaround
DLPX-72186	CDB logfile retention works incorrectly if a PDB has multiple timeflows pointing to the same CDB timeflow.	No workaround.

Key	Summary	Workaround
DLPX-74749	Oracle ENVIRONMENT_REFRESH_AND_DISCOVER job may fail with "The object OraclePDBConfig does not exist on the system".	Manually run an environment refresh after the ENVIRONMENT_REFRESH_AND_DISCOVER job fails.
DLPX-75517	Provisioning an Oracle vPDB to a vCDB fails with "`ORA-00959: tablespace 'TEMP' does not exist`" if there is no temporary tablespace named `TEMP` within the production `PDB$SEED` database.	One of the following two workarounds can be applied: 1. Re-attempt the vPDB provision to a linked CDB. The provision should succeed. 2. Manually create (or rename) the `TEMP` tablespace in the production `PDB$SEED` database, take a new snapshot of the vPDB, and provision from that snapshot to the vCDB.
DLPX-75737	Retention saves unnecessary logs if the bookmark falls exactly on a snapshot end SCN or snapshot end timestamp.	No workaround.
DLPX-75995	On Windows 2019 Server, with KB4598230 cumulative update, adding or refreshing an environment fails when PowerShell transcription is enabled.	The user has to turn off the transcription if the staging is on Windows 2019 Server with recent updates.
DLPX-76388	Entering key pairs directly into hook environment variables (a new feature in 6.0.9.0 (see page 1065)) results in an internal error.	Use password variables instead, or use a password vault. See Passing Credentials Securely to Hook Operations[80]

80 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/6262834/Passing+credentials+securely+to+hook+operations

3.3.27 Version 6.0.8.0

Key	Summary	Workaround

DLPX-64307	Environment refresh should ignore cluster discovery for VDBs	The following workaround should resolve the issue without Delphix support intervention.
		For RAC databases
		1. Add the instances to the clusterware configuration, ensuring you add all instances configured via Delphix. For example:
		`srvctl add instance -db <Database Name> -instance <Instance Name> -node <Node Name>`
		2. Refresh the environment. At this point, Delphix will add the instances back to the Delphix configuration, and you should be able to perform actions such as stop, start, disable, enable, etc.
		From here, we recommend removing the clusterware configuration.
		3. Stop the VDB via the Delphix GUI or CLI.
		4. Remove the instances from clusterware
		`srvctl remove instance -db <Database Name> -instance <Instance Name>`
		5. Remove the database from clusterware
		`srvctl remove database -db <Database Name>`
		6. Refresh the environment.
		7. Start the VDB via the Delphix GUI or CLI.
		For RACOne databases
		1. Stop the instance.

Key	Summary	Workaround

		`sqlplus "/as sysdba" shutdown immediate`
		2. Ensure the RACOne configuration has the "instance prefix" configuration set.
		`[oracle@mwtestx1 ~]$ srvctl config database -d <vdb name> | grep "Instance name prefix"`
		`Instance name prefix: VDB`
		3. If this is not set, then set it using
		`srvctl modify database -db <vdb name> -instance <instance prefix>`
		4. Set the pfile location to allow clusterware to start the instance
		`srvctl modify database -db <vdb name> -spfile <path to pfile>`
		The pfile is located in
		`<Delphix Mount base>/<VDB Name>/datafile/spfile.ora`
		5. Start the instance using clusterware
		`srvctl start database -d vdb`
		When querying the status of the database, it should now show a running instance. As an example:

Key	Summary	Workaround

		`[oracle@mwtestx1 ~]$  srvctl status database -d VDB Instance VDB_1 is running on node mwtestx1 Online relocation: INACTIVE`
		6. Refresh the environment. At this point, Delphix will add the instances back to the Delphix configuration, and you should be able to perform actions such as stop, start, disable, enable, etc.
		From here, we recommend removing the clusterware configuration.
		7. Stop the VDB via the Delphix GUI or CLI.
		8. Remove the database from clusterware
		`srvctl remove database -db <Database Name>`
		9. Remove the database from clusterware
		`srvctl remove database -db <Database Name>`
		10. Refresh the environment.
		11. Start the VDB via the Delphix GUI or CLI.

DLPX-72186	CDB logfile retention works incorrectly if a PDB has multiple timeflows pointing to the same CDB timeflow	No workaround.
DLPX-73409	Duplicate listener entry gets generated in engine metadata if Oracle listener is manually started with non-uppercase name	Follow these steps: 1. Restart the listener without a name parameter so it displays in uppercase. 2. Update VDBs to use uppercase LISTENER. 3. Refresh the environment. lowercase listener will be removed and LISTENER updated appropriately.

Key	Summary	Workaround
DLPX-74749	Oracle ENVIRONMENT_REFRESH_AND_DISCOVER job may fail with "The object OraclePDBConfig does not exist on the system"	Manually run an environment refresh after the ENVIRONMENT_REFRESH_AND_DISCOVER job fails.
DLPX-74860	Provisioning a 2nd generation VDB from a dSource with imported RO transportable tablespaces fails with ORA-19654	No workaround.
DLPX-74882	Masking's SFTP client no longer compatible with SolarWinds and Goanyware SFTP servers	No workaround.
DLPX-75517	Provisioning an Oracle vPDB to a vCDB fails with " `ORA-00959: tablespace 'TEMP' does not exist` " if there is no temporary tablespace named `TEMP` within the production `PDB$SEED` database	One of the following two workarounds can be applied: 1. Re-attempt the vPDB provision to a linked CDB. The provision should succeed. 2. Manually create (or rename) the `TEMP` tablespace in the production `PDB$SEED` database, take a new snapshot of the vPDB and provision from that snapshot to the vCDB.

3.3.28 Version 6.0.7.0

Key	Summary	Workaround
DLPX-74457	Cluster discovery for Oracle RAC partially fails on Solaris 10	Switch the default login shell for the Delphix OS user from `/bin/sh` to `/bin/bash`.
DLPX-74749	Oracle ENVIRONMENT_REFRESH_AND_DISCOVER job may fail with "The object OraclePDBConfig does not exist on the system"	Manually run an environment refresh from the Delphix UI if the ENVIRONMENT_REFRESH_AND_DISCOVER job fails with the mentioned error.

Key	Summary	Workaround
DLPX-76718	Time required to display the point-in-time pop-over on timelines for Self-Service Templates with replicated objects can degrade linearly with each replication.	No workaround but reducing replication frequency can reduce the impact.

3.3.29 Version 6.0.6.0

Key	Summary	Workaround
DLPX-73224	When provisioning from a non-multitenant source to a virtual pluggable database (vPDB), the post-plug hook script (dx-post-plug-hook.sh) should exit with the vPDB either closed or open unrestricted. If the vPDB is left open with restricted access subsequent Snapshots of the vPDB will fail with "oracle.ucp.UniversalConnectionPoolException" and " java.sql.SQLException: ORA-01035 ORACLE only available to users with RESTRICTED SESSION privilege" .	To prevent the issue: Ensure that the vPDB is either closed/mounted or open unrestricted when exiting from dx-post-plug-hook.sh. After it has happened: close and reopen the vPDB.
DLPX-72749	Provisioning a TDE-enabled vPDB with system encrypted tablespaces fails with the error "ORA-28374: typed master key not found".	No workaround, contact Delphix Support (TBD)
DLPX-72655	Provisioning an Oracle TDE-enabled vPDB fails intermittently if the dSource is encrypted after linking	Enabling TDE on an existing non-encrypted dSource is not supported. Detach, rename the Delphix dSource name and re-attach it as a new TDE-enabled dSource before re-attempting the provisioning operation.

Key	Summary	Workaround
DLPX-73224	When provisioning from a non-multitenant source to a virtual pluggable database (vPDB), the post-plug hook script (dx-post-plug-hook.sh) should exit with the vPDB either closed or open unrestricted. If the vPDB is left open with restricted access subsequent Snapshots of the vPDB will fail with "oracle.ucp.UniversalConnectionPoolException" and " java.sql.SQLException: ORA-01035 ORACLE only available to users with RESTRICTED SESSION privilege"	To prevent the issue: Ensure that the vPDB is either closed/mounted or open unrestricted when exiting from dx-post-plug-hook.sh. After it has happened: close and reopen the vPDB.
DLPX-73357	Rewinding an Oracle TDE-enabled vPDB to the snapshot before vPDB migration may fail with `exception.oracle.tde.export.keys.failed` if all of the steps of the keystore merge procedure are not followed correctly	Verify that the CDB is restarted after merging the keys of the old target CDB into the new target CDB and re-attempt the rewind operation. If the rewind still fails, provision a new vPDB from the required snapshot instead of the rewind operation.
DLPX-72181	The "Restore Self-Service data container from the bookmark of sibling data container" operation may fail in an Oracle TDE environment with `exception.oracle.tde.export.keys.failed` if the keystore merge procedure is not followed correctly. In this case, the warning "Refresh operation on the TDE-configured container <target container> from the Timeflow of another container <sibling container> requires merging of the TDE keystores." will be displayed in the Delphix UI events log	Verify that the procedure for merging the sibling keystores is followed correctly and re-attempt the self-service operation.
DLPX-73742	Provisioning an Oracle TDE-enabled vPDB may fail with the error "ORA-28367: wallet does not exist" if the TDE wallet for the target linked CDB is stored on ASM storage	Storing the TDE wallet on ASM storage is currently unsupported. Modify `sqlnet.ora` to point to the keystore location outside of the ASM diskgroup and re-attempt the provisioning operation.

Key	Summary	Workaround
DLPX-73788	Provisioning or enabling an Oracle TDE-enabled vPDB fails when $ORACLE_BASE is used in sqlnet.ora	Any environment variable referenced in sqlnet.ora must always be set in the environment for the Delphix OS user. Ensure that the environment variable $ORACLE_BASE is set in the shell initialization file for the Delphix OS user and re-attempt the operation.
DLPX-73789	If WALLET_ROOT initialization parameter is configured on a TDE-enabled dSource PDB, provisioning may fail since the auxiliary CDB instance uses dSource keystore location.	If the provision fails, there are 2 workarounds: 1. Provide permissions to the Delphix OS user to create files in the location identified by WALLET_ROOT directory of the source keystore on the target host. 2. Modify the dSource database to not use WALLET_ROOT to identify the TDE keystore.

3.3.30 Version 6.0.2.0

Key	Summary	Workaround
DLPX-69638	Masking job created on engine 6.0.1.1 or prior is failing after the upgrade to version 6.0.2.0 or later	Masking jobs created in 6.0.1.x using a Hana JDBC driver will need to be updated to grant the following permission {"java.io[81].FilePermission" "/", "read"} in 6.0.2.0. All drivers created in and after 6.0.2.0 will be granted this permission by default.

81 http://java.io/

3.3.31 Version 6.0.0.0

Key	Summary	Workaround
DLPX-60397	If a mapping algorithm is included in multiple jobs, only one job should be run at a time. If multiple jobs are run at the same time, then the mapping algorithm might contain multiple mappings to the same value or the jobs might deadlock.	Only run one job at a time.
DLPX-60947	Self-Service template with replica VDB is not updated with new Timeflow on incremental replication update	The latest replica VDB data can still be accessed by doing a Self-Service container Refresh, rather than a point-in-time restore from the template.
DLPX-61079	Certificate import validation may incorrectly reject a root CA certificate	Support must manually import the certificate into the truststore.
DLPX-61405	Masking operation should wait for zfs delete queue to drain	Replication may send more data than expected if masking involves dropping large DBF files.
DLPX-64493	V5 API /roles endpoint missing certain items	View and set these privileges through the GUI
DLPX-66155	Failed DSP engine test leads to multiple blocked client.jar processes on target hosts.	Restarting a Delphix Engine during a network throughput test is not recommended as it may lead to a system hang.
DLPX-66860	ADFS does not like NameIDPolicy sent by SSO app	Create an explicit rule in ADFS that transforms the {{emailAddress}} attribute into a {{nameid}}. The rule type must be "Transform an incoming claim". The incoming claim type must be "Email address" and the outgoing claim type "Name ID". The nameid format must be "Email address".

Key	Summary	Workaround
DLPX-66973	Date format is changed after importing the environment	Either (a) use the GUI import feature and then review the imported date formats for correctness or (b) use EngineSync to export/import jobs, which will not alter the date format.

3.4 API changes

This section covers the following topics:

- API changes in Delphix 23.0.0.0 (see page 354)
- API changes in Delphix 22.0.0.0 (see page 355)
- API changes in Delphix 21.0.0.0 (see page 356)
- API changes in Delphix 20.0.0.0 (see page 357)
- API changes in Delphix 19.0.0.0 (see page 358)
- API changes in Delphix 18.0.0.0 (see page 359)
- API Changes in Delphix 17.0.0.0 (see page 359)
- API Changes in Delphix 16.0.0.0 (see page 360)
- API Changes in Delphix 15.0.0.0 (see page 361)
- API Changes in Delphix 14.0.0.0 (see page 362)
- API Changes in Delphix 13.0.0.0 (see page 363)
- API Changes in Delphix 12.0.0.0 (see page 364)
- API changes in Delphix 11.0.0.0 (see page 367)
- API changes in Delphix 10.0.0.0 (see page 370)
- API changes in Delphix 9.0.0.0 (see page 372)
- API changes in Delphix 8.0.0.0 (see page 374)
- API changes in Delphix 7.0.0.0 (see page 377)
- API changes in Delphix 6.0.17.0 (see page 380)
- API changes in Delphix 6.0.16.0 (see page 383)
- API changes in Delphix 6.0.15.0 (see page 385)
- API changes in Delphix 6.0.14.0 (see page 388)
- API changes in Delphix 6.0.13.0 (see page 391)
- API changes in Delphix 6.0.12.0 (see page 396)
- API changes in Delphix 6.0.11.0 (see page 400)
- API changes in Delphix 6.0.10.0 (see page 405)
- API changes in Delphix 6.0.9.0 (see page 408)
- API changes in Delphix 6.0.8.0 (see page 411)
- API changes in Delphix 6.0.7.0 (see page 416)
- API changes in Delphix 6.0.6.0 (see page 419)
- API changes in Delphix 6.0.5.0 (see page 420)
- API changes in Delphix 6.0.4.0 (see page 423)
- API changes in Delphix 6.0.3.0 (see page 425)

3.4.1 API changes in Delphix 23.0.0.0

In Delphix 23.0, the new API version is 1.11.34. This section describes all API changes since API version 1.11.33 that was released with Delphix 22.0. All URL paths are relative to `/resources/json/delphix`.

3.4.1.1 What's new?

API Object	Path	Type	Name	Description
`ASEHostEnvironmentParameters`	NA (value type)	Property	`enableTls`	True if you want to discover the SAP ASE instances configured with TLS/SSL.
`ASEHostEnvironmentParameters`	NA (value type)	Property	`skipServerCertificateValidation`	Setting it to true will skip the server certificate validation during the SSL handshake. Only set it if you do not want to add the required certificate into the Delphix Engine truststore but still want to use the TLS/SSL-enabled database connection. This is not recommended for a production environment as it is less secure.
`ASEInstance`	NA (value type)	Property	`tlsEnabled`	True if the SAP ASE instance is TLS/SSL enabled on the given port.
`GcpObjectStoreTest`	NA (value type)	API Type	`GcpObjectStoreTest`	A Google Cloud object store connectivity test object.
`GcpObjectStore`	NA (value type)	API Type	`GcpObjectStore`	A Google Cloud object store.

3.4.1.2 What's changed?

There are no changes in this release.

3.4.2 API changes in Delphix 22.0.0.0

In Delphix 22.0, the new API version is 1.11.33. This section describes all API changes since API version 1.11.32, which was released with Delphix 21.0. All URL paths are relative to `/resources/json/delphix`.

3.4.2.1 What's changed?

API Object	Path	Type	Name	Change
`Namespace`	`namespace`	Property	`locked`	`default` is set to `false`
`StorageTestParameters`	NA (value type)	Property	`testRegion`	`maximum` is set to `1099511627776`
`NfsEncryptionConfig`	`/service/nfs/encryption`	API Type	`NfsEncryptionConfig`	The end point `/service/nfs/encryption` is moved to `/service/security/env_encryption` and the object `NfsEncryptionConfig` is replaced with `EnvEncryptionConfig`

3.4.2.2 What's new?

API Object	Path	Type	Name	Description

`CapacityBreakdown`	NA (value type)	Property	`currentTimeflowUnvirtualizedSpace`	Unvirtualized space used by the current (active) TimeFlow. This is approximately equal to the space a VDB would take upon a virtual-to-physical (V2P) operation.
`EnvEncryptionConfig`	`security/env_encryption`	API Type	`EnvEncryptionConfig`	Configuration for the environment encryption services like NFS and iSCSI.

3.4.3 API changes in Delphix 21.0.0.0

In Delphix 21.0, the new API version is 1.11.32. This section describes all API changes since API version 1.11.31 that was released with Delphix 20.0. All URL paths are relative to `/resources/json/delphix`.

3.4.3.1 What's new?

There are no new APIs in this release.

3.4.3.2 What's changed?

API Object	Path	Type	Name	Change
`OracleDBConfig`	NA (value type)	Property	`tdeKeystoreConfigType`	Property type changed from `string` to `[ "string", "null" ]` Property update changed from `readonly` to `optional`
`S3ObjectStore`	NA (value type)	Property	`region`	Property `create` changed from `required` to `optional`

3.4.4 API changes in Delphix 20.0.0.0

In Delphix 20.0, the new API version is 1.11.31. This section describes all API changes since API version 1.11.30, which was released with Delphix 19.0. All URL paths are relative to `/resources/json/delphix`.

3.4.4.1 What's New?

There are no new APIs in this release.

3.4.4.2 What's Changed?

API Object	Path	Type	Name	Change
`Host`	NA (value type)	Property	`nfsEncryptionPort`	Renamed from `nfsEncryptionPort` to `encryptionPort.`
`MSSqlStagingPushSyncStrategy`	NA (value type)	Property	`dbState`	Property `dbState` added - User provided db state that will be used to create staging push db.
`OracleStagingSourceParameters`	NA (value type)	Property	`mountBase`	Property `mountBase` modified to include new attribute `format`
`SourceEnvironment`	NA (value type)	Property	`nfsEncryptionEnabled`	Renamed from `nfsEncryptionEnabled` to `encryptionEnabled`
`UnixRuntimeMountInformation`	NA (value type)	Property	`nfsEncryptionEnabled`	Renamed from `nfsEncryptionEnabled` to `encryptionEnabled`

3.4.5 API changes in Delphix 19.0.0.0

In Delphix 19.0, the new API version is 1.11.30. This section describes all API changes since API version 1.11.29, which was released with Delphix 18.0. All URL paths are relative to `/resources/json/delphix`.

3.4.5.1 What's new?

There are no new APIs in this release.

3.4.5.2 What's changed?

API Object	Path	Type	Name	Description
`OracleDBConfig`	NA	Property	`tdeKeystoreConfigType`	`tdeKeystoreConfigType` has been added
`OracleHostParameters`	NA	Property	`tdeExternalKeyManagerCredential`	`tdeExternalKeyManagerCredential` has been added.
`OracleHostParameters`	NA	Property	`tdeOkvHomePath`	`tdeOkvHomePath` has been added.
`OracleLinkFromStaging`	NA	Property	`tdeKeystoreConfigType`	`tdeKeystoreConfigType` has been added.
`OracleVirtualCdbSource`	NA	Property	`redoLogSizeInMB`	`redoLogSizeInMB` has been removed.

API Object	Path	Type	Name	Description
`OracleVirtualCdbSource`	NA	Property	`redoLogGroups`	`redoLogGroups` has been removed.
`OracleVirtualPdbSource`	NA	Property	`parentTdeKeystorePath`	`null` has been added to types list along with string.

3.4.6 API changes in Delphix 18.0.0.0

In Delphix 18.0, the new API version is 1.11.29. This section describes all API changes since API version 1.11.28, which was released with Delphix 17.0. All URL paths are relative to `/resources/json/delphix`.

3.4.6.1 What's New?

There are no new APIs in this release.

3.4.6.2 What's Changed?

API Object	Path	Type	Name	Description
`EventFilter`	NA	Property	`eventTypes`	`pattern` has been updated in `items`.
`FluentdConfig`	`service/fluentd/configuration`	API Type	`FluentdConfig`	`rootOperations` has been removed.

3.4.7 API Changes in Delphix 17.0.0.0

In Delphix 17.0, the new API version is 1.11.28. This section describes all API changes since API version 1.11.27, which was released with Delphix 16.0. All URL paths are relative to `/resources/json/delphix`.

3.4.7.1 What's New?

API Object	Path	Type	Name	Description
`NfsEncryptionConfig`	/resources/json/delphix/service/nfs/encryption	API Type	`NfsEncryptionConfig`	Configuration for the nfs encryption of this application. Contains property “ `cipherSuite` ” to allow change of ciphersuite used in NFS Encryption.
`SourceEnvironment`	/resources/json/delphix/environment	Property	`enableAllSources`	Enable all sources of an environment.
`SourceEnvironment`	/resources/json/delphix/environment	Property	`disableAllSources`	Disable all sources of an environment.

3.4.8 API Changes in Delphix 16.0.0.0

In Delphix 16.0, the new API version is 1.11.27. This section describes all API changes since API version 1.11.26, which was released with Delphix 15.0. All URL paths are relative to `/resources/json/delphix`.

3.4.8.1 What's Changed?

API Object	Path	Type	Name	Change
`PublicSystemInfo`	/about	Property	`userManagementEnabled`	Property userManagementEnabled has been removed.

`PublicSystemInfo`	/about	Property	`centralManagementProductName`	Property centralManagementProductName has been removed.
`CloudEnableParameters`	NA (value type)	API Type	`CloudEnableParameters`	CloudEnableParameters has been removed.
`CloudStatus`	/service/cloud	API Type	`CloudStatus`	CloudStatus has been removed.
`DelphixDataServicesComponentInfo`	NA (value type)	API Type	`DelphixDataServicesComponentInfo`	DelphixDataServicesComponentInfo has been removed.
`OracleNodeListener`	NA	nameParent	`OracleNodeListener`	nameParent has been added.
`UserManagement`	NA (value type)	API Type	`UserManagement`	UserManagement has been removed.

3.4.9 API Changes in Delphix 15.0.0.0

In Delphix 15.0, the new API version is 1.11.26. This section describes all API changes since API version 1.11.25, which was released with Delphix 14.0. All URL paths are relative to `/resources/json/delphix`.

3.4.9.1 What's Changed?

API Object	Path	Type	Name	Change
`FluentdPlugin`	/service/fluentd/plugins	Property	`downloadFluentdLog`	Property downloadFluentdLog has been added.
`OracleVirtualSource`	NA (value type)	Property	`invokeDatapatch`	Property invokeDatapatch has been added.

3.4.9.2 What's New?

API Object	Path	Type	Name	Description
`OracleExportDBTimeflowPointTransferStrategy`	NA (value type)	API Type	`OracleExportDBTimeflowPointTransferStrategy`	Create a non-MT physical database from a TimeFlow point.
`OracleExportPDBTimeflowPointTransferStrategy`	NA (value type)	API Type	`OracleExportPDBTimeflowPointTransferStrategy`	Create a physical pluggable database from a TimeFlow point.

3.4.10 API Changes in Delphix 14.0.0.0

In Delphix 14.0, the new API version is 1.11.25. This section describes all API changes since API version 1.11.24, released with Delphix 13.0. All URL paths are relative to `/resources/json/delphix`.

3.4.10.1 What's Changed?

API Object	Path	Type	Name	Change
`FluentdPlugin`	/service/fluentd/plugins	Property	`gems`	Property gems have been removed.
`Host`	/host	Property	`nfsAddressList`	Property nfsAddressList has been updated for property format from host to hostOrCIDRAddress.

3.4.10.2 What's New?

API Object	Path	Type	Name	Description
`Host`	/host	API Type	`nfsEncryptionPort`	Property nfsEncryptionPort have been added.
`MSSqlAdditionalDatabaseParameters`	NA (value type)	API Type	`MSSqlAdditionalDatabaseParameters`	Collection of all the parameters that are not inherited during provisioning of a vdb.
`MSSqlSnapshot`	NA (value type)	API Type	`additionalDataBaseParameters`	Collection of all the parameters that are not inherited during provisioning of a vdb.
`OciObjectStoreTest`	NA (value type)	API Type	`OciObjectStoreTest`	An Oracle Cloud object store connectivity test object.
`OciObjectStore`	NA (value type)	API Type	`OciObjectStore`	An Oracle Cloud object store.
`SourceEnvironment`	/environment	API Type	`nfsEncryptionEnabled`	Flag indicating whether nfs is encrypted or not.
`UnixRuntimeMountInformation`	NA (value type)	API Type	`nfsEncryptionEnabled`	The flag for NFS Encryption.

3.4.11 API Changes in Delphix 13.0.0.0

In Delphix 13.0, the new API version is 1.11.24. This section describes all API changes since API version 1.11.23, which was released with Delphix 12.0. All URL paths are relative to /resources/json/delphix.

3.4.11.1 What's Changed?

API Object	Path	Type	Name	Change
`OracleRefreshParameters`	NA (value type)	Property	`force`	Property force has been added.
`OracleRollbackParameters`	NA (value type)	Property	`force`	Property force has been added.
`TimeflowSnapshot`	/snapshot	Property	`operations`	Property parameters has been removed from property operations

3.4.11.2 What's New?

API Object	Path	Type	Name	Description
`ContainerStorageInfo`	NA (value type)	API Type	`ContainerStorageInfo`	Container Storage Information.
`Container`	/database	API Type	`containerStorageInfo`	Container Storage Information.

3.4.12 API Changes in Delphix 12.0.0.0

In Delphix 12.0, the new API version is 1.11.23. This section describes all API changes since API version 1.11.22, which was released with Delphix 11.0. All URL paths are relative to `/resources/json/delphix`.

3.4.12.1 What's Changed?

API Object	Path	Type	Name	Change
`KeyPairCredential`	NA (value type)	Property	`publicKey`	Property create has been updated from required to optional
`OracleBaseExternalLinkData`	NA (value type)	API Type	`sourcingPolicy`	Property sourcingPolicy has been added
`OracleExportDBInPlaceTransferStrategy`	NA (value type)	API Type	`OracleExportDBInPlaceTransferStrategy`	Property operationsPostV2P has been added
`OracleExportPDBInPlaceTransferStrategy`	NA (value type)	API Type	`OracleExportPDBInPlaceTransferStrategy`	Property operationsPostV2P has been added
`OracleDatabaseContainer`	NA (value type)	Property	`sourcingPolicy`	Ref has been updated from delphix-oracle-sourcing-policy.json to delphix-oracle-base-sourcing-policy.json
`OracleSourcingPolicy`	NA (value type)	Property	`OracleSourcingPolicy`	Ref has been updated from delphix-sourcing-policy.json to delphix-oracle-base-sourcing-policy.json
`OracleVirtualSource`	NA (value type)	API Type	`OracleVirtualSource`	Property allowRefreshRewindPostV2P has been added.
`ValidateNTPParameters`	NA (value type)	Property	`address`	Format has been updated from hostname to host.

3.4.12.2 What's New?

`OracleBaseSourcingPolicy`	NA (value type)	API Type	`OracleBaseSourcingPolicy`	Oracle Database policies for managing SnapSync and LogSync across sources for a Oracle container.
`OracleStagingSourcingPolicy`	NA (value type)	API Type	`OracleStagingSourcingPolicy`	Database policies for managing LogSync for Oracle Staging push container.
`SuperuserSessionLogDeleteRecord`	NA (value type)	API Type	`SuperuserSessionLogDeleteRecord`	Represents a deletion of a Delphix superuser session log file.
`SuperuserSessionLogDownloadRecord`	NA (value type)	API Type	`SuperuserSessionLogDownloadRecord`	Represents a download of a Delphix superuser session log file.
`SuperuserSession`	/superuser/session	API Type	`SuperuserSession`	Audit logs for superuser sessions.
`OracleExportOperationsPostV2P`	NA (value type)	API Type	`OracleExportOperationsPostV2P`	Describes operations allowed on virtual source post V2P.
`OracleExportTransferStrategy`	NA (value type)	API Type	`OracleExportTransferStrategy`	The transfer strategy for exporting a database whether in-place or TimeFlow point based.
`SystemInfo`	system	API Type	`SystemInfo`	Property maxNativeMemoryGb has been added

3.4.13 API changes in Delphix 11.0.0.0

In Delphix 11.0, the new API version is 1.11.22. This section describes all API changes since API version 1.11.21, which was released with Delphix 10.0. All URL paths are relative to `/resources/json/delphix`.

3.4.13.1 What's changed?

API object	Path	Type	Name	Change
`AzureAuthentication`	NA (value type)	API Type	`AzureAuthentication`	Properties have been added tenantId and clientId.
`CpuUtilDatapoint`	NA (value type)	API Type	`CpuUtilDatapoint`	Removed property dtrace.
`AzureSecretsAuthentication`	NA (value type)	API Type	`AzureSecretsAuthentication`	Removed properties tenantId and clientId.
`AzureSecretsAuthentication`	NA (value type)	Property	`clientSecret`	Description has been updated for property clientSecret
`CipherSuite`	/service/tls/cipherSuite	Property	`name`	enum property values have been updated.
`Container`	/database	Property	`delete, operations, rollback, sync`	PgSQLDatabaseContainer and MySQLDatabaseContainer have been removed from defaultType.
`DeleteParameters`	NA (value type)	API Type	`DeleteParameters`	The description has been updated.
`DeletionDependency`	NA (value type)	Property	`size`	The description has been updated for property size.

API object	Path	Type	Name	Change
`KeyPairCredential`	NA (value type)	Property	`privateKey`	The format has been updated from password to pemKey.
`KeyPairCredential`	NA (value type)	Property	`publicKey`	Property format has been updated from password to hostKey and property create has been updated from required to optional.
`NamedKeyPairCredential`	NA (value type)	Property	`privateKey`	The property format has been updated from password to pemKey.
`NamedKeyPairCredential`	NA (value type)	Property	`publicKey`	The property format has been updated from password to hostKey.
`OracleExportTimeflowFilesystemLayout`	NA (value type)	API Type	`OracleExportTransferStrategy`	Property rmanFileSectionSizeInGb has been added.
`PemClientCertificate`	NA (value type)	Property	`privateKey`	The property format has been updated from password to pemKey.
`RefreshParameters`	NA (value type)	API Type	`RefreshParameters`	The description has been updated.
`RollbackParameters`	NA (value type)	API Type	`RollbackParameters`	The description has been updated.
`SourceConfig`	/ sourceconfig	Property	`operations`	PgSQLDBClusterConfig has been removed from defaultType.
`SourceDisableParameters`	NA (value type)	API Type	`SourceDisableParameters`	The description has been updated

API object	Path	Type	Name	Change
`SourceEnableParameters`	NA (value type)	API Type	`SourceEnableParameters`	The description has been updated
`SourceIngestionData`	NA (value type)	Property	`containerType`	MYSQL_DB_CONTAINER and PGSQL_DB_CONTAINER have been removed from enum.
`SourceStartParameters`	NA (value type)	API Type	`SourceStartParameters`	The description has been updated
`SourceStopParameters`	NA (value type)	API Type	`SourceStopParameters`	The description has been updated
`SourceTypeAggregateIngestedSize`	NA (value type)	Property	`containerType`	MYSQL_DB_CONTAINER and PGSQL_DB_CONTAINER have been removed from enum.
`Source`	/source	Property	`operations, disable, start, stop`	PgSQLLinkedSource, PgSQLStagingSource, PgSQLVirtualSource, MySQLLinkedSource, MySQLStagingSource, MySQLVirtualSource
`User`	/user	Property	`publicKey`	Property redact-in-logs have been removed.

3.4.13.2 What's new?

API object	Path	Type	Name	Description
`OracleExportTimeflowFilesystemLayout`	NA (value type)	API Type	`OracleExportTransferStrategy`	Property rmanFileSectionSizeInGb has been added.

API object	Path	Type	Name	Description
`OracleStagingSource`	NA (value type)	API Type	`OracleStagingSource`	Property datafileMountPath and archiveMountPath has been added
`TimeConfig`	/service/time	API Type	`TimeConfig`	Property rootOperations has been added.
`ValidateNTPParameters`	NA (value type)	API Type	`ValidateNTPParameters`	Validate an NTP server by querying it for the time.
`AzureCertificateAuthentication`	NA (value type)	API Type	`AzureCertificateAuthentication`	Client certificate for authenticating to an Azure vault.

3.4.14 API changes in Delphix 10.0.0.0

In Delphix 10.0.0.0, the new API version is 1.11.21. This section describes all API changes since API version 1.11.20, which was released with Delphix 9.0. All URL paths are relative to

`/resources/json/delphix.`

What's changed?

API object	Path	Type	Name	Change
`NfsConfig`	/service/nfs	Property	`mountVersion`	Property mountVersion (enum) has been updated to include NFSv4.
`OracleVirtualCdbSource`	NA (value type)	API Type	`OracleVirtualCdbSource`	Removed property nodeListeners.

API object	Path	Type	Name	Change
`PasswordVaultTestParameters`	NA (value type)	API Type	`PasswordVaultTestParameters`	Removed properties host and port.
`SystemInfo`	/System	Property	`maxHeapSizeGb`	Maximum heap size of the management application.

3.4.14.1 What's new?

API object	Path	Type	Name	Description
`AzureAuthentication`	NA (value type)	API Type	`AzureAuthentication`	Parameters for authenticating to an Azure vault.
`AzureSecretsAuthentication`	NA (value type)	API Type	`AzureSecretsAuthentication`	Secret credential for authenticating to an Azure vault.
`AzureVaultCredential`	NA (value type)	API Type	`AzureVaultCredential`	The Azure vault based security credential.
`AzureVaultTestParameters`	NA (value type)	API Type	`AzureVaultTestParameters`	Azure password vault test configuration.
`AzureVault`	NA (value type)	API Type	`AzureVault`	Azure password vault configuration.

API object	Path	Type	Name	Description
`HostedVaultTestParameters`	NA (value type)	API Type	`HostedVaultTestParameters`	Hosted password vault test configuration.

3.4.15 API changes in Delphix 9.0.0.0

In Delphix 9.0, the new API version is 1.11.20. This section describes all API changes since API version 1.11.19, that was released with Delphix 8.0.0.0. All URL paths are relative to `/resources/json/delphix.`

3.4.15.1 What's changed?

API object	Path	Type	Name	Change
`MSSqlVirtualSource`	NA (value type)	Property	`configParams`	Can now be updated via CLI.
`CyberArkPasswordVault`	NA (value type)	API Type	`CyberArkPasswordVault`	Returns a list of password vault objects on the system. Limits CLI visibility for create, update and delete of password vault objects to system users only.
`NamespaceFailoverParameters`	NA (value type)	API Type	`NamespaceFailoverParameters`	Removed property smartFailover.
`PasswordVault`	/service/passwordVault	API Type	`PasswordVault`	Removed property host and port.

API object	Path	Type	Name	Change
`TimeZone`	/timezone	Property	`id`	Property id (enum) has been updated to include America/Rosario,Europe/Belfast.

3.4.15.2 What's new?

API object	Path	Type	Name	Description
`MSSqlVirtualSource`	N/A (value type)	Property	`configTemplate`	Optional database template to use for provisioning, refresh and enable. If set, configParams will be ignored on the provision or refresh.
`HostedPasswordVault`	N/A (value type)	PropertyAPI Type	`HostedPasswordVault`	Password vaults with host and port.
`UnixHost`	N/A (value type)	Property	`toolkitPath`	Added property format with value unixabsolutepath.
`OracleDBExportParameters`	N/A (value type)	API Type	`OracleDBExportParameters`	Parameters to use as input to export Oracle non-MT databases.
`OracleEnhancedExportParameters`	N/A (value type)	API Type	`OracleEnhancedExportParameters`	The enhanced parameters to use as input to export Oracle databases.
`OraclePDBExportParameters`	N/A (value type)	API Type	`OraclePDBExportParameters`	Parameters to use as input to export Oracle PDB databases.

API object	Path	Type	Name	Description
`OracleExportPDBInPlaceTransferStrategy`	N/A (value type)	API Type	`OracleExportPDBInPlaceTransferStrategy`	Convert a virtual PDB to a physical PDB in-place.
`OracleExportDBInPlaceTransferStrategy`	N/A (value type)	API Type	`OracleExportDBInPlaceTransferStrategy`	Convert a non-MT virtual DB to a physical DB in-place.
`OracleExportASMStorageStrategy`	N/A (value type)	API Type	`OracleExportASMStorageStrategy`	Storage strategy for exporting database files to ASM.
`OracleASMLayout`	N/A (value type)	API Type	`OracleASMLayout`	ASM diskgroups for datafiles, archive logs/redo logs.

3.4.16 API changes in Delphix 8.0.0.0

In Delphix 8.0.0.0, the new API version is 1.11.19. This section describes all API changes since APIversion 1.11.18, which was released with Delphix 7.0.0.0. All URL paths are relative to `/resources/json/delphix.`

3.4.16.1 What's changed?

API object	Path	Type	Name	Change
`FluentdConfig`	/service//fluentd/configuration	Property	`performanceMetricsResolution`	Can now set property via CLI.

API object	Path	Type	Name	Change
`NamespaceFailoverParameters`	N/A (value type)	Property	`getFailbackCapability`	Return object has changed from string to object.
`RepavePrepareParameters`	N/A (value type)	API Type	`RepavePreparaParameters`	Removed properties: • quiesceSources • enableSourcesOnFailure • ignoreMaskingJobsInProgress Added property: • unquiesceSourcesOnFailure
`SNMPV2Config`	/service/snmp/v2	Property	`severity`	Added enum value AUDIT.
`Source`	/source	Property	`update`	Restrict CLI visibility to domain users only.
`Source`	/source	Property	`operations`	Restrict CLI visibility to domain users only.
`Source`	/source	Property	`disable`	Restrict CLI visibility to domain users only.
`Source`	/source	Property	`stop`	Restrict CLI visibility to domain users only.
`Source`	/source	Property	`lock`	Restrict CLI visibility to domain users only.
`SsoSuccessfulLoginRecord`	N/A (value type)	API Type	`SsoSuccessfulLoginRecord`	Redact email from logs.

API object	Path	Type	Name	Change
`User`	/user	Property	`isDefault`	Can no longer set property via CLI.

3.4.16.2 What's new?

API object	Path	Type	Name	Description
`CloudEndpoint`	N/A (value type)	API Type	`CloudEndpoint`	A mapping of a recommend region and endpoint for use with Cloud Engines.
`FailbackCapability`	N/A (value type)	API Type	`FailbackCapability`	A replica namespace's fallback capability.
`OracleLinkedSourceOperations`	N/A (value type)	API Type	`OracleLinkedSourceOperations`	Describes operations which are performed on linked sources at various times.
`SystemInfo`	/system	Property	`rootOperations`	Added getRecommendedCloudStorageEndpoints to get list of recommended endpoints and regions for Cloud Engine setup.
`OracleBaseLinkData`	N/A (value type)	Property	`operations`	Added preLogSync property and new optional fields.
`OracleLinkedSource`	N/A (value type)	Property	`operations`	Added preLogSync property and new optional fields.

3.4.17 API changes in Delphix 7.0.0.0

In Delphix 7.0.0.0, the new API version is 1.11.18. This section describes all API changes since APIversion 1.11.17, which was released with Delphix 6.0.17.0. All URL paths are relative to `/resources/json/delphix.`

3.4.17.1 What's changed?

API object	Path	Type	Name	Change
`Domain`	N/A (value type)	API Type	`lock`	Restrict CLI visibility to domain users only.
`MSSqlExportParameters`	N/A (value type)	Property	`prodSyncConfigParam`	Property removed.
`NettyVersionInfo`	/netty	Property	`NettyVersionInfo`	API object removed.
`OracleExportAsmParameters`	N/A (value type)	Property	`OracleExportAsmParameters`	API object removed.
`SystemPackage`	/system/package	Property	`SystemPackage`	API object removed.

3.4.17.2 What's new?

API Object	Path	Type	Name	Description
`Domain`	N/A (value type)	Property	`cliVisibility`	Restrict CLI visibility to system or domain users.

API Object	Path	Type	Name	Description
`FluentdAttribute`	N/A (value type)	API Type	`FluentdAttribute`	Fluentd attribute.
`FluentdConfig`	N/A (value type)	API Type	`FluentdConfig`	Fluentd configuration information.
`FluentdPlugin`	/service/fluentd/plugins	API Type	`FluentdPlugin`	Upload and manage fluentd plugins.
`FluentdRegularAttribute`	N/A (value type)	API Type	`FluentdRegularAttribute`	Fluentd attribute with a plain value.
`FluentdSecretAttribute`	N/A (value type)	API Type	`FluentdSecretAttribute`	Fluentd attribute with a secret value.
`MSSqlSnapshot`	N/A (value type)	Property	`emptySnapshot`	Readonly - True if the staging push dSource snapshot is empty.
`OracleStagingSource`	N/A (value type)	Property	`customEnvVars`	Custom environment variables for Oracle databases.
`OracleBaseStagingLinkData`	N/A (value type)	API Type	`OracleBaseStagingLinkData`	Represents common parameters to link an Oracle database using a staging database.
`OracleStagingPushSyncParameters`	N/A (value type)	API Type	`OracleStagingPushSyncParameters`	The parameters to use as input to sync a staging Oracle database.
`OracleStagingPushSyncStrategy`	N/A (value type)	API Type	`OracleStagingPushSyncStrategy`	Oracle specific parameters for staging push sync strategy.

API Object	Path	Type	Name	Description
`OracleSyncFromStagingParameters`	N/A (value type)	API Type	`OracleSyncFromStagingParameters`	The parameters to use as input to sync from a Staging Oracle database.
`OracleLinkFromStaging`	N/A (value type)	API Type	`OracleLinkFromStaging`	Represents parameters to link a non-pluggable Oracle database using a staging database.
`OraclePDBLinkFromStaging`	N/A (value type)	API Type	`OraclePDBLinkFromStaging`	Represents parameters to link a pluggable Oracle database using a staging database.
`OracleSourceLessSyncStrategy`	N/A (value type)	API Type	`OracleSourceLessSyncStrategy`	Base type for Oracle source less sync strategy and associated parameters.
`S3ObjectStoreRepaveApplyParameters`	N/A (value type)	API Type	`S3ObjectStoreRepaveApplyParameters`	An Amazon Simple Storage Service (Amazon S3) object store.
`BlobObjectStoreRepaveApplyParameters`	N/A (value type)	API Type	`BlobObjectStoreRepaveApplyParameters`	An Azure blob object store.
`BlockStorageRepaveApplyParameters`	N/A (value type)	API Type	`BlockStorageRepaveApplyParameters`	Block Storage.
`RepaveApplyParameters`	N/A (value type)	API Type	`RepaveApplyParameters`	The parameters to use as input to repave apply.
`RepavePrepareParameters`	N/A (value type)	API Type	`RepavePrepareParameters`	The parameters to use as input to repave prepare.

3.4.18 API changes in Delphix 6.0.17.0

In Delphix 6.0.17.0, the new API version is 1.11.17. This section describes all API changes since API version 1.11.16, which was released with Delphix 6.0.16.0. All URL paths are relative to `/resources/json/delphix`.

3.4.18.1 What's changed?

API object	Path	Type	Name	Change
`AppDataExportParameters`	N/A (value type)	API Type	`AppDataExportParameters`	Object now subclasses GenericExportParameters
`ASEAttachData`	N/A (value type)	Property	`stagingPreScript`	Property removed.
`ASEAttachData`	N/A (value type)	Property	`stagingPostScript`	Property removed.
`ASELinkData`	N/A (value type)	Property	`stagingPreScript`	Property removed.
`ASELinkData`	N/A (value type)	Property	`stagingPostScript`	Property removed.
`ASEStagingSource`	N/A (value type)	Property	`preScript`	Property removed.
`ASEStagingSource`	N/A (value type)	Property	`postScript`	Property removed.
`DbExportParameters`	N/A (value type)	API Type	`DbExportParameters`	Object now subclasses GenericExportParameters

API object	Path	Type	Name	Change
`ExportParameters`	N/A (value type)	Property	`timeflowPointParameters`	Property removed and added to GenericExportParameters
`ExportParameters`	N/A (value type)	Property	`sourceConfig`	Property removed and added to GenericExportParameters
`OracleHostParameters`	N/A (value type)	Property	`tdeKeystoresRootPath`	Restrict format to valid unix path
`OracleVirtualPdbSource`	N/A (value type)	Property	`targetVcdbTdeKeystorePath`	Remove maxLength constraint. Add minLength constraint of 1 character.
`RefreshParameters`	N/A (value type)	API Type	`RefreshParameters`	Object now subclasses ReprovisionParameters
`RollbackParameters`	N/A (value type)	API Type	`RollbackParameters`	Object now subclasses ReprovisionParameters
`TimeflowPointSemantic`	N/A (value type)	Property	`location`	Add OLDEST_SNAPSHOT to acceptable values

3.4.18.2 What's new?

API object	Path	Type	Name	Description
`Container`	/database	Property	`cdbContainer`	Restrict listing to only include the PDB datasets belonging to the specified CDB dataset reference. This option is mutually exclusive with all the other options.
`Container`	/database	Object Operations	`deprovision`	Deprovisions a container.
`Container`	/database	Object Operations	`reprovision`	Reprovisions a container. This should only be called after a deprovision using the ReprovisionParameters returned from that operation.
`EventsConfig`	/service/events	API Type	`EventsConfig`	Event Message Configuration.
`GenericExportParameters`	N/A (value type)	API Type	`GenericExportParameters`	The parameters to use as input to export requests.
`JobRetentionConfig`	/job/retention	API Type	`JobRetentionConfig`	Configuration for job retention.
`OracleStagingSource`	N/A (value type)	Property	`allowAutoStagingRestartOnHostReboot`	Indicates whether Delphix should automatically restart this staging source when target host reboot is detected.

API object	Path	Type	Name	Description
`ReprovisionParameters`	N/A (value type)	API Type	`ReprovisionParameters`	The input parameters to refresh and rollback requests.
`SourceIngestionData`	N/A (value type)	Property	`containerTypeLabel`	Human readable description of containerType.
`SystemInfo`	/system	Property	`cloudRegion`	The region of the current system if hosted on an applicable cloud service provider.
`SystemInfo`	/system	Property	`smtpConfigured`	Whether SMTP has been configured."

3.4.19 API changes in Delphix 6.0.16.0

In Delphix 6.0.16.0, the new API version is 1.11.16. This section describes all API changes since API version 1.11.15, which was released with Delphix 6.0.15.0. All URL paths are relative to `/resources/json/delphix`.

3.4.19.1 What's changed?

API object	Path	Type	Name	Change
`NetworkRoute`	/network/route	Property	`gateway`	Parameter has been made optional.
`OracleBaseSourceRuntime`	N/A (value type)	Property	`databaseMode`	UNMOUNTED mode added.

3.4.19.2 What's new?

API object	Path	Type	Name	Description
`FluentdConfig`	N/A (value type)	Property	`performanceMetricsResolution`	Performance metrics resolution.
`HostNfsChecksParameters`	N/A (value type)	API Type	`HostNfsChecksParameters`	Mechanism to verify user can mount/ unmount on a remote host.
`OracleStagingSourceParameters`	N/A (value type)	Property	`physicalStandby`	Whether this staging database will be configured as a physical standby.
`OracleStagingSource`	N/A (value type)	Property	`physicalStandby`	Whether this staging database will be configured as a physical standby.
`PasswordResetConfig`	N/A (value type)	API Type	`PasswordResetConfig`	Password Reset Configuration
`PasswordResetRequestParameters`	N/A (value type)	API Type	`PasswordResetRequestParameters`	Parameters in a password reset request.
`PasswordResetValidationParameters`	N/A (value type)	API Type	`PasswordResetValidationParameters`	Self-service password Reset validation parameters.

API object	Path	Type	Name	Description
`PasswordResetValidationResult`	N/A (value type)	API Type	`PasswordResetValidationResult`	Self-service password Reset validation result.
`PluginManifest`	N/A (value type)	Property	`hasLinkedSourceSize`	Indicates whether linked.source_size() operation has been implemented.
`PluginManifest`	N/A (value type)	Property	`hasVirtualSourceSize`	Indicates whether virtual.source_size() operation has been implemented.
`SourceEnvironment`	/environment	Property	`nfsChecks`	Tests that the environment user can run mount and unmount successfully on the host.

3.4.20 API changes in Delphix 6.0.15.0

In Delphix 6.0.15.0, the new API version is 1.11.15. This section describes all API changes since API version 1.11.14, which was released with Delphix 6.0.14.0. All URL paths are relative to `/resources/json/delphix`.

3.4.20.1 What's changed?

API object	Path	Type	Name	Change
`DatabaseTemplate`	N/A (value type)	Property	`sourceType`	MySQLVirtualSource and PgSQLVirtualSource removed from acceptable values. OracleLinkedSource added to acceptable values.

API object	Path	Type	Name	Change
`S3ObjectStoreTestResult`	N/A (value type)	API Type	`timestamp`	Object removed and replaced by ObjectStoreTestResult
`ObjectStore`	/storage/objectStorage	Property	`root`	Change root path to /resources/json/delphix/storage/objectStorage
`S3ObjectStore`	N/A (value type)	Standard operations	`read` `update`	Properties removed and added to ObjectStore

`S3ObjectStore`	N/A (value type)	Root operations	`testConnection`
			`cacheHitsReport`
			`clearCacheHits`
				Properties removed and added to ObjectStore

`TimeZone`	N/A (value type)	Property	`sendSocketBuffer`	Add Pacific/Kanton timezone

3.4.20.2 What's new?

API object	Path	Type	Name	Description
`BlobObjectStoreAccess`	N/A (value type)	API Type	`BlobObjectStoreAccess`	Blob object store access

API object	Path	Type	Name	Description
`BlobObjectStoreAccessKey`	N/A (value type)	API Type	`BlobObjectStoreAccessKey`	Blob object store access key
`BlobObjectStoreAccessManagedIdentities`	N/A (value type)	API Type	`BlobObjectStoreAccessManagedIdentities`	Blob object store access through Managed Identities
`ObjectStoreCacheHitsReport`	N/A (value type)	API Type	`ObjectStoreCacheHitsReport`	A cache hits report for an object store
`ObjectStoreTestResult`	N/A (value type)	API Type	`ObjectStoreTestResult`	An object store connectivity test result
`ObjectStoreTest`	N/A (value type)	API Type	`ObjectStoreTest`	An object store connectivity test object
`ObjectStore`	/storage/objectStorage	Standard operations	`read` `update`	Retrieve the specified ObjectStore object Update the specified ObjectStore object

API object	Path	Type	Name	Description

`ObjectStore`	/storage/objectStorage	Root operations	`testConnection`
			`cacheHitsReport`
			`clearCacheHits`
				Test connectivity to an object store
				Get a ZettaCache hits-by-size report
				Clear the accumulated ZettaCache hits-by-size data

3.4.21 API changes in Delphix 6.0.14.0

In Delphix 6.0.14.0, the new API version is 1.11.14. This section describes all API changes since API version 1.11.13, which was released with Delphix 6.0.13.0. All URL paths are relative to `/resources/json/delphix`.

3.4.21.1 What's changed?

API object	Path	Type	Name	Change
`CompatibilityCriteria`	N/A (value type)	Property	`processor`	Constraints removed
`JobEvent`	N/A (value type)	Property	`timestamp`	Description changed to “Time the event last occurred.”
`MSSqlExportParameters`	N/A (value type)	Property	`filesystemLayout`	Description changed to note this filesystem configuration is specific to anexported MSSQL database.

API object	Path	Type	Name	Change
`NetworkDSPTestParameters`	N/A (value type)	Property	`queueDepth`	Changed from 32 to 64
`NetworkDSPTestParameters`	N/A (value type)	Property	`blockSize`	Changed from 65536 to 1048576
`NetworkDSPTestParameters`	N/A (value type)	Property	`sendSocketBuffer`	Changed from 262144 to 1048576
`NetworkDSPTestParameters`	N/A (value type)	Property	`receiveSocketBuffer`	Changed from 262144 to 1048576
`OracleBaseDBConfig`	N/A (value type)	Property	`user`	Field changed to be nullable
`OracleBaseDBConfig`	N/A (value type)	Property	`credential`	Field changed to be nullable
`OracleVirtualPdbSource`	N/A (value type)	Property	`tdeKeyIdentifier`	minLength changed from 1 to 34
`Schedule`	N/A (value type)	Property	`cutoffTime`	Added to description: “A value of 0 indicates no cutoff”

3.4.21.2 What's new?

API object	Path	Type	Name	Description
`AppDataLinkedSource`	N/A (value type)	Property	`locked`	Whether the source is protected from deletion and other data-losing actions.

API object	Path	Type	Name	Description
`ASELinkedSource`	N/A (value type)	Property	`locked`	Whether the source is protected from deletion and other data-losing actions.
`ASEStagingSource`	N/A (value type)	Property	`locked`	Whether the source is protected from deletion and other data-losing actions.
`JobEvent`	N/A (value type)	Property	`startTimestamp`	Time the event first occurred.
`MSSqlFileMappingParameters`	N/A (value type)	API Type	`MSSqlFileMappingParameters`	The parameters to use as input to provide File Mapping for MSSQL databases.
`MSSqlLinkedSource`	N/A (value type)	Property	`locked`	Whether the source is protected from deletion and other data-losing actions.
`MSSqlStagingSource`	N/A (value type)	Property	`locked`	Whether the source is protected from deletion and other data-losing actions.
`MSSqlTimeflowFilesystemLayout`	N/A (value type)	API Type	`MSSqlTimeflowFilesystemLayout`	A filesystem layout that matches the filesystem of a Delphix MSSQL TimeFlow.
`OracleLinkedSource`	N/A (value type)	Property	`locked`	Whether the source is protected from deletion and other data-losing actions.

API object	Path	Type	Name	Description
`OracleStagingSource`	N/A (value type)	Property	`locked`	Whether the source is protected from deletion and other data-losing actions.
`Source`	/source	Root operation	`lock`	Protects source from deletion and other data-losing actions. Cannot be undone.

3.4.22 API changes in Delphix 6.0.13.0

In Delphix 6.0.13.0, the new API version is 1.11.13. This section describes all API changes since API version 1.11.12, which was released with Delphix 6.0.12.0. All URL paths are relative to `/resources/json/delphix`.

3.4.22.1 What's changed?

API object	Path	Type	Name	Change
`MSSqlNoBackupSyncParameters`	API Type	N/A (value type)	`MSSqlNoBackupSyncParameters`	STAGING_PUSH_INGESTION feature flag removed
`MSSqlSourcelessSyncStrategy`	API Type	N/A (value type)	`MSSqlSourcelessSyncStrategy`	STAGING_PUSH_INGESTION feature flag removed
`MSSqlStagingPushSyncStrategy`		N/A (value type)	`MSSqlStagingPushSyncStrategy`	STAGING_PUSH_INGESTION feature flag removed
`NetworkInterface`	API Type	/network/interface	`NetworkInterface`	Removed Domain User access on CLI

API object	Path	Type	Name	Change
`NetworkRoute`	API Type	/network/route	`NetworkRoute`	Removed Domain User access on CLI
`NetworkRoute`	Property	/network/route	`protocol`	Changed from "readonly" to "optional" when creating.
`OAuth2Config`	API Type	/service/oauth2	`OAuth2Config`	Added Domain User access from CLI. Provide System user only access for updates and root operations.
`ObjectStore`	API Type	N/A (value type)	`ObjectStore`	DELPHIX_DATA_BANK feature flag removed. Added update operation.
`OracleDBConfig`	Properties	N/A (value type)	`databaseName` `uniqueName`	Replace property pattern with format - oracleDatabaseName oracleDbUniqueName
`OracleInstance`	Property	N/A (value type)	`instanceName`	Replace property pattern with format - oracleInstanceName
`OraclePDBConfig`	Property	N/A (value type)	`databaseName`	Replace property pattern with format - oraclePDBName
`S3ObjectStoreAccessInstanceProfile`	API Type	N/A (value type)	`S3ObjectStoreAccessInstanceProfile`	DELPHIX_DATA_BANK feature flag removed.

API object	Path	Type	Name	Change
`S3ObjectStoreAccessKey`	API Type	N/A (value type)	`S3ObjectStoreAccessKey`	DELPHIX_DATA_BANK feature flag removed.
`S3ObjectStoreAccess`	API Type	N/A (value type)	`S3ObjectStoreAccess`	DELPHIX_DATA_BANK feature flag removed.
`S3ObjectStoreTestResult`	API Type	N/A (value type)	`S3ObjectStoreTestResult`	DELPHIX_DATA_BANK feature flag removed.
`S3ObjectStoreTest`	API Type	N/A (value type)	`S3ObjectStoreTest`	DELPHIX_DATA_BANK feature flag removed.
`S3ObjectStore`	API Type	/storage/objectStorage	`S3ObjectStore`	DELPHIX_DATA_BANK feature flag removed. "Update" operation added for all the properties.

3.4.22.2 What's new?

API object	Path	Type	Name	Description
`FluentdAttributeDefinition`	N/A (value type)	API Type	`FluentdAttributeDefinition`	Fluentd attribute definition.
`FluentdAttribute`	N/A (value type)	API Type	`FluentdAttribute`	Fluentd attribute.

API object	Path	Type	Name	Description
`FluentdConfig`	/service/fluentd/configuration	API Type	`FluentdConfig`	Fluentd configuration information.
`FluentdPlugin`	/service/fluentd/plugins	API Type	`FluentdPlugin`	Upload and manage fluentd plugins.
`FluentdRegularAttribute`	N/A (value type)	API Type	`FluentdRegularAttribute`	Fluentd attribute with a plain value.
`FluentdSecretAttribute`	N/A (value type)	API Type	`FluentdSecretAttribute`	Fluentd attribute with a secret value.
`MaskingServiceConfig`	/maskingjob/servicecon fig	Property	`maxJobFetchCount`	Maximum number of jobs to fetch from masking service. Defaults to 500.
`NettyVersionInfo`	/netty	API Type	`NettyVersionInfo`	View netty version and switch between legacy and latest netty version.
`ObjectStore`	N/A (value type)	Property	`configured`	States whether an object store has been configured.
`OracleBaseStagingLinkData`	N/A (value type)	API Type	`OracleBaseStagingLinkData`	Represents common parameters to link an Oracle database using a staging database.
`OracleLinkFromStaging`	N/A (value type)	API Type	`OracleLinkFromStaging`	Represents parameters to link a non-pluggable Oracle database using a staging database.

API object	Path	Type	Name	Description
`OraclePDBLinkFromStaging`	N/A (value type)	API Type	`OraclePDBLinkFromStaging`	Represents parameters to link a pluggable Oracle database using a staging database.
`OracleSourceLessSyncStrategy`	N/A (value type)	API Type	`OracleSourceLessSyncStrategy`	Base type for Oracle source less sync strategy and associated parameters.
`OracleStagingPushSyncParameters`	N/A (value type)	API Type	`OracleStagingPushSyncParameters`	The parameters to use as input to sync a staging Oracle database.
`OracleStagingPushSyncStrategy`	N/A (value type)	API Type	`OracleStagingPushSyncStrategy`	Oracle specific parameters for staging push sync strategy.
`OracleStagingSourceParameters`	N/A (value type)	Property	`instanceName`	The name (sid) of the instance.
`S3ObjectStore`	/storage/objectStorage	Root operations	`cacheHitsReportclearCacheHits`	Get print out of the ZettacCache hit-by-size histogram. Clear the current ZettacCache hit-by-size histogram.
`S3ObjectStoreTestResult`	N/A (value type)	Property	`errorMessage`	Error message from connectivity test.

3.4.23 API changes in Delphix 6.0.12.0

In Delphix 6.0.12.0, the new API version is 1.11.12. This section describes all API changes since API version 1.11.11, which was released with Delphix 6.0.11.0. All URL paths are relative to `/resources/json/delphix`.

3.4.23.1 What's changed?

API object	Path	Type	Name	Change
`AbstractToolkit`	/toolkit	property	`language`	Removed.
`CipherSuite`	/service/tls/cipherSuite	property	`name`	Removed values for enum - "TLS_RSA_WITH_AES_256_CBC_SHA", "TLS_RSA_WITH_AES_128_CBC_SHA", "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA", "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA", "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA", "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA", "TLS_DHE_RSA_WITH_AES_256_CBC_SHA", "TLS_DHE_RSA_WITH_AES_128_CBC_SHA", "TLS_DHE_DSS_WITH_AES_256_CBC_SHA", "TLS_DHE_DSS_WITH_AES_128_CBC_SHA".
`HttpConnectorConfig`	/service/httpConnector	property	`tlsVersions`	Removed values for enum - "SSLv3", "TLSv1", "TLSv1_1"
`OAuth2Config`	/service/oauth2	property	`userMatchingFieldType`	Updated description.

API object	Path	Type	Name	Change

`OracleLinkedSource`	N/A (value type)	property	`config`
			`backupLevelEnabled`
			`rmanChannels`
			`filesPerSet`
			`checkLogical`
			`encryptedLinkingEnabled`
			`compressedLinkingEnabled`
			`bandwidthLimit`
			`numberOfConnections`
				Removed.

`OracleSource`	N/A (value type)	property	`config`	Removed.
`StorageDevice`	/storage/device	property	`list` `operations`	Removed CliDisplay parameters - "bootDevice", "allocating". Removed operations - "remove", "removeVerify".

API object	Path	Type	Name	Change
`UnixHost`	N/A (value type)	property	`oracleJdbcKeystorePassword`	Removed.
`User`	/user	property	`apiUser`	Renamed to allowPasswordAuthentication (see in What's New?)

3.4.23.2 What's new?

API object	Path	Type	Name	Description
`ConfiguredStorageDevice`	N/A (value type)	property	`allocating`	True if the device is being used for allocations. Allocations will be disabled automatically if the device is going to be removed. The allocating property will also be false for devices that are not configured.
`MSSqlNoBackupSyncParameters`	N/A (value type)	API Type	`MSSqlNoBackupSyncParameters`	The parameters to use as input to sync MSSQL databases without a backup.
`MSSqlSourcelessSyncStrategy`	N/A (value type)	API Type	`MSSqlSourcelessSyncStrategy`	MSSQL specific parameters for sourceless sync strategy.
`MSSqlStagingPushSyncStrategy`	N/A (value type)	API Type	`MSSqlStagingPushSyncStrategy`	MSSQL specific parameters for staging push sync strategy.

API object	Path	Type	Name	Description
`OracleHostParameters`	N/A (value type)	API Type	`OracleHostParameters`	Oracle specific host parameters.
`OracleLinkedSource`	N/A (value type)	property	`syncStrategy` `overrideMapsTo`	Parameters used to sync the container. These are persisted and used for every sync operation. Override 'mapsTo' property from source.js for the query parameters defined in 'list' section.
`OracleManagedSource`	N/A (value type)	property	`config`	Reference to the configuration for the source.
`Plugin`	N/A (value type)	property	`language`	Implementation language for workflows in this plugin.
`StorageDeviceRemovalStatus`	/storage/remove	property	`rootOperations`	Add "start" and "verify" root operations to • remove storage devices • verify storage devices can be removed.
`StorageDeviceRemoveParameters`	N/A (value type)	API Type	`StorageDeviceRemoveParameters`	The parameters to use as input when removing devices.
`UnixHost`	N/A (value type)	property	`oracleHostParameters`	The Oracle specific parameters associated with the host.

API object	Path	Type	Name	Description
`User`	/user	property	`allowPasswordAuthentication`	If the Delphix Engine has been configured for SAML SSO, only users with this property set can authenticate with a username and password for API or CLI access.

3.4.24 API changes in Delphix 6.0.11.0

In Delphix 6.0.11.0, the new API version is 1.11.11. This section describes all API changes since API version 1.11.10, which was released with Delphix 6.0.10.0. All URL paths are relative to `/resources/json/delphix`.

3.4.24.1 What's changed?

API object	Path	Type	Name	Change
`AdvancedSettingsInfo`	/system/advancedSettings	API Type	`AdvancedSettingsInfo`	Hidden behind feature flag SYSTEMTUNABLEINTERFACE
`DelphixManagedBackupIngestionStrategy`	N/A (value type)	API Type	`DelphixManagedBackupIngestionStrategy`	Removed.
`ExternalBackupIngestionStrategy`	N/A (value type)	API Type	`ExternalBackupIngestionStrategy`	Removed.
`IngestionStrategy`	N/A (value type)	API Type	`IngestionStrategy`	Removed.

API object	Path	Type	Name	Change
`MSSqlDBConfig`	N/A (value type)	property	`mssqlUser`	Removed.

`MSSqlLinkedSource`	N/A (value type)	property	`config`
			`sharedBackupLocations`
			`ingestionStrategy`
			`mssqlNetbackupConfig`
			`mssqlCommvaultConfig`
				Removed.

`MSSqlSourceSyncStrategy`	N/A (value type)	property	`config`	Optional while creating.
`NoBackupIngestionStrategy`	N/A (value type)	API Type	`NoBackupIngestionStrategy`	Removed.
`OracleLinkFormExternal`	N/A (value type)	property	`config`	Extends reference changed from OracleBaseLinkData to OracleBaseExternalLinkData. Config property removed.
`OraclePDBLinkFromExternal`	N/A (value type)	property	`config`	Extends reference changed from OracleBaseLinkData to OracleBaseExternalLinkData. Configproperty removed.

API object	Path	Type	Name	Change
`OracleTimeflow`	N/A (value type)	property	`tdeUUID`	No Longer behind feature flag ORACLEMTTDE.

`OracleVirtualPdbSource`	N/A (value type)	property	`parentTdeKeystorePath`
			`parentTdeKeystorePassword`
			`tdeExportedKeyFileSecret`
			`tdeUUID`
				No Longer behind feature flag ORACLEMTTDE.

`Source`	N/A (value type)	property	`config`	nameParent changed from config to container. Configproperty removed from properties and list operation.

`SsoFailedLoginRecord`	N/A (value type)	property	`userAgent`
			`originIp`
			`reason`
				Extends reference changed from TypedObject to FailedLoginRecord.
				All properties removed.

`SsoSuccessfulLoginRecord`	N/A (value type)	property	`userAgent` `originIp`	Extends reference changed from TypedObject to LoginRecord. Properties removed.

3.4.24.2 What's new?

API object	Path	Type	Name	Description
`AppDataCompatibilityCriteria`	N/A (value type)	API Type	`AppDataCompatibilityCriteria`	The compatibility criteria to use for filtering the list of available AppData repositories.
`AppDataSource`	N/A (value type)	property	`config`	Reference to the configuration for the source.
`ASESource`	N/A (value type)	property	`config`	Reference to the configuration for the source.
`FailedLoginRecord`	N/A (value type)	API Type	`FailedLoginRecord`	Represents an successful login using an external identity provider.
`LoginRecord`	N/A (value type)	API Type	`LoginRecord`	Represents a failed SSO login.
`MSSqlDelphixManagedSyncStrategy`	N/A (value type)	property	`config`	Reference to the configuration for the source.
`MSSqlExternalManagedSourceSyncStrategy`	N/A (value type)	property	`config`	Reference to the configuration for the source.
`MSSqlLinkedSource`	N/A (value type)	property	`syncStrategy` `overrideMapsTo`	Sync strategy for this source. Override mapsTo property for the query parameter defined in list at the root type.

API object	Path	Type	Name	Description
`MSSqlStagingSource`	N/A (value type)	property	`config`	Reference to the configuration for the source.
`MSSqlUser`	N/A (value type)	property	`config`	Reference to the configuration for the source.
`MySQLSource`	N/A (value type)	property	`config`	Reference to the configuration for the source.
`OAuth2Config`	/service/oauth2	API Type	`OAuth2Config`	OAuth2 Configuration.
`OAuth2FailedLoginRecord`	N/A (value type)	API Type	`OAuth2FailedLoginRecord`	Represents a failed OAuth2 login.
`OAuth2SuccessfulLoginRecord`	N/A (value type)	API Type	`OAuth2SuccessfulLoginRecord`	Represents a successful OAuth2 login.
`OracleBaseExternalLinkData`	N/A (value type)	API Type	`OracleBaseExternalLinkData`	Represents common parameters to link all externally managed Oracle databases.
`OracleSourceBasedSyncStrategy`	N/A (value type)	API Type	`OracleSourceBasedSyncStrategy`	Base type for Oracle source based sync strategy and associated parameters.
`OracleSource`	N/A (value type)	property	`config`	Reference to the configuration for the source.

API object	Path	Type	Name	Description
`OracleSyncStrategy`	N/A (value type)	API Type	`OracleSyncStrategy`	Base type for Oracle sync strategy and associated parameters.
`OracleVirtualPdbSource`	N/A (value type)	property	`tdeKeyIdentifier`	ID of the key created by Delphix, as recorded in encryption_keys.key_id.
`OracleVirtualSource`	N/A (value type)	property	`newDBID`	Indicates whether Delphix will generate a new DBID during VDB provision or refresh.
`OsAdminStatus`	/osadmin/engineStatus	API Type	`OsAdminStatus`	Information for the current state of the Delphix Engine.
`PgSQLSource`	N/A (value type)	property	`config`	Reference to the configuration for the source.
`PluginManifest`	N/A (value type)	property	`hasVirtualCleanup`	Indicates whether virtual.cleanup() operation has been implemented.
`ReplicationSourceState`	N/A (value type)	property	`lastKnownTargetVersion`	The Delphix version of the target engine during the last replication using the spec.
`ReplicationTargetState`	N/A (value type)	property	`lastKnownSourceVersion`	The Delphix version of the source engine during the last replication of the namespace.

3.4.25 API changes in Delphix 6.0.10.0

In Delphix 6.0.10.0, the new API version is 1.11.10. This section describes all API changes since API version 1.11.9, which was released with Delphix 6.0.9.0. All URL paths are relative to `/resources/json/delphix`.

3.4.25.1 What's changed?

API object	Path	Type	Name	Change
`CredentialsEnvVars`	N/A (value type)	API Type	`CredentialsEnvVars`	Extends reference changed from TypedObject to IdentifiableArrayElement.
`CredentialsEnvVars`	N/A (value type)	property	`baseVarName` `credentials`	Optional for an update. Optional for update.
`IscsiTarget`	/storage/iscsi/target	property	`state`	Added enum value ERROR.
`KeyPairCredential`	N/A (value type)	property	`privateKey` `publicKey`	Required for create. Optional for update.
`NetworkRoute`	/network/route	property	`list`	Add enum value protocol in "cliDisplay".
`Operation`	N/A (value type)	API Type	`Operation`	Extends reference changed from TypedObject to IdentifiableArrayElement.
`OracleDatabaseContainer`	N/A (value type)	property	`racMaxInstanceLag`	FeatureFlag 'ORACLERACMAXINSTANCELAG' is removed. default value changed to 3.
`OracleExportParameters`	N/A (value type)	property	`filesystemLayout`	Extends reference changed from TimeFlowFilesystemLayout to OracleExportTimeflowFilesystemLayout

API object	Path	Type	Name	Change
`SystemInitializationParameters`	N/A (value type)	property	`devices`	Optional while creating.
`TimeZone`	/timezone	property	`id`	Removed values for enum - "ACT", "AET", "AGT", "ART", "AST", "SST", "US/Pacific-New", "VST".

3.4.25.2 What's new?

API object	Path	Type	Name	Description
`BlobObjectStoreAccess`	N/A (value type)	API Type	`BlobObjectStoreAccess`	Blob object store access
`BlobObjectStoreAccessKey`	N/A (value type)	API Type	`BlobObjectStoreAccessKey`	Blob object store access key
`BlobObjectStoreAccessManagedIdentities`	N/A (value type)	API Type	`BlobObjectStoreAccessManagedIdentities`	Blob object store access through Managed Identities
`ObjectStoreCacheHitsReport`	N/A (value type)	API Type	`ObjectStoreCacheHitsReport`	A cache hits report for an object store
`ObjectStoreTestResult`	N/A (value type)	API Type	`ObjectStoreTestResult`	An object store connectivity test result

API object	Path	Type	Name	Description
`ObjectStoreTest`	N/A (value type)	API Type	`ObjectStoreTest`	An object store connectivity test object
`ObjectStore`	/storage/objectStorage	Standard operations	`read` `update`	Retrieve the specified ObjectStore object Update the specified ObjectStore object

`ObjectStore`	/storage/objectStorage	Root operations	`testConnection`
			`cacheHitsReport`
			`clearCacheHits`
				Test connectivity to an object store
				Get a ZettaCache hits-by-size report
				Clear the accumulated ZettaCache hits-by-size data

3.4.26 API changes in Delphix 6.0.9.0

In Delphix 6.0.9.0, the new API version is 1.11.9. This section describes all the API changes since API version 1.11.8, which was released with Delphix 6.0.8.0. All URL paths are relative to `/resources/json/delphix`.

3.4.26.1 What changed

API object	Path	Type	Name	change
`AppDataProvisionParameters`	N/A (value type)	property	`sourceConfig`	Type changed from AppDataSourceConfig to AppDataDirectSourceConfig.

API object	Path	Type	Name	change
`BaseSupportBundleParameters`	N/A (value type)	property	`bundleType`	Added enum value DOCKER_LOG.
`CipherSuite`	/service/tls/cipherSuite	property	`name`	Added enum values TLS_AES_128_CCM_8_SHA256, TLS_AES_128_CCM_SHA256, TLS_AES_128_GCM_SHA256, TLS_AES_256_GCM_SHA384, and TLS_CHACHA20_POLY1305_SHA256.
`HttpConnectorConfig`	/service/httpConnector	property	`tlsVersions`	Added enum value TLSv1_3.
`Namespace`	/namespace	property	`locked`	No longer behind a feature flag.
`OracleCluster`	/environment	property	`name` `version`	No longer read-only Deleted.
`ReplicationSpec`	N/A (value type)	property	`lockedProfile`	No longer behind a feature flag.
`WindowsCluster`	/environment	property	`name`	No longer read-only.

3.4.26.2 What is new

API object	Path	Type	Name	Description
`AlertProfile`	/alert/profile	property	`user`	User to which the alert profile is assigned. Defaults to the logged-in user.
`CredentialsEnvVars`	N/A (value type)	API type	`CredentialsEnvVars`	Credentials to be placed in environment variables for an operation.
`IscsiTargetDiscoverParameters`	N/A (value type)	property	`chapUsername` `chapPassword` `chapUsernameIn` `chapPasswordIn`	CHAP username to be used for iSCSI Discovery. CHAP password to be used for iSCSI Discovery. Target/Mutual CHAP username to be used for iSCSI Discovery (bidirectional authentication). Target/Mutual CHAP password to be used for iSCSI Discovery (bidirectional authentication).
`IscsiTarget`	/storage/iscsi/target	property	`chapUsername` `chapPassword` `chapUsernameIn` `chapPasswordIn`	CHAP username to be used for iSCSI Target authentication. CHAP password to be used for iSCSI Target authentication. Target/Mutual CHAP username (bidirectional authentication). Target/Mutual CHAP password (bidirectional authentication).
`SourceOperation`	N/A (value type)	property	`credentialsEnvVarsList`	List of environment variables that will contain credentials for this operation.

API object	Path	Type	Name	Description
`SystemVersion`	/system/version	property	`hotfixVersion`	The hotfix version.

3.4.27 API changes in Delphix 6.0.8.0

In Delphix 6.0.8.0, the new API version is 1.11.8. This section describes all API changes since API version 1.11.7, which was released with Delphix 6.0.7.0. All URL paths are relative to `/resources/json/delphix`.

3.4.27.1 What changed

API object	Path	Type	Name	Change
`AttachData`	N/A (value type)	property	`config`	Deleted.
`Container`	/database	operation	`undo`	HTTP method changed from GET to POST.
`LinkData`	N/A (value type)	property	`config`	Deleted.

API object	Path	Type	Name	Change
`MSSqlAttachData`	N/A (value type)	property	`backupLocationCredentials` `backupLocationUser/source` `config` `externalFilePath` `ingestionStrategy` `mssqlCommvaultConfig` `mssqlNetbackupConfig` `mssqlUser` `sharedBackupLocations`	Deleted. (See the new syncStrategy property in the table below.)

API object	Path	Type	Name	Change
`MSSqlLinkData`	N/A (value type)	property	`backupLocationCredentials` `backupLocationUser` `config` `externalFilePath` `ingestionStrategy` `mssqlCommvaultConfig` `mssqlNetbackupConfig` `mssqlUser` `sharedBackupLocations`	Deleted. (See the new syncStrategy property in the table below.)
`MSSqlLinkedSource`	/source	property	`backupLocationCredentials` `backupLocationUser` `externalFilePath`	Deleted.
`NamespaceFailoverParameters`	N/A (value type)	property	`smartFailover`	Default changed to true.
`OracleSyncFromExternalParameters`	N/A (value type)	property	`filesForFullBackup`	No longer behind a feature flag.

API object	Path	Type	Name	Change
`SourceIngestionData`	N/A (value type)	property	`containerType`	Removed obsolete value VMWARE_CONTAINER from enum.
`SourceRepository`	/repository	property	`linkingEnabled`	Deleted.
`SourceTypeAggregateIngestedSize`	N/A (value type)	property	`containerType`	Removed obsolete value VMWARE_CONTAINER from enum.
`Source`	/source	property	`status`	Removed value DETACHED from enum.
`VMware*`		API types	`VMware*`	Deleted all types whose names start with VMware (obsolete).

3.4.27.2 What is new

API object	Path	Type	Name	Description
`AdvancedSettingsInfo`	/system/advancedSettings	API type	`AdvancedSettingsInfo`	Set advanced virtualization, OS, network, and service tunables.
`ASESnapshot`	/snapshot	property	`dbEncryptionKey`	Name of database encryption key present for this snapshot.
`IscsiInitiator`	/storage/iscsi/initiator	API type	`IscsiInitiator`	Endpoint for iSCSI initiator information.
`MSSqlAttachData`	N/A (value type)	property	`syncStrategy`	Configuration that determines what sync strategy the source will use.

API object	Path	Type	Name	Description
`MSSqlDelphixManagedSyncStrategy`	N/A (value type)	API type	`MSSqlDelphixManagedSyncStrategy`	MSSQL specific parameters for delphix managed source based sync strategy.
`MSSqlExternalManagedSourceSyncStrategy`	N/A (value type)	API type	`MSSqlExternalManagedSourceSyncStrategy`	MSSQL specific parameters for externally managed source based sync strategy.
`MSSqlLinkData`	N/A (value type)	property	`syncStrategy`	Configuration that determines what sync strategy the source will use for linking.
`MSSqlSourceSyncStrategy`	N/A (value type)	API type	`MSSqlSourceSyncStrategy`	MSSQL specific parameters for source based sync strategy.
`MSSqlSyncStrategy`	N/A (value type)	API type	`MSSqlSyncStrategy`	Base type for mssql specific parameters for sync strategy.
`Namespace`	/ namespace	property	`locked`	Indicates the namespace is locked. (Behind the DATA_VAULT feature flag.)
`OracleDatabaseContainer`	/container	property	`racMaxInstanceLag`	Maximum number of log sequences that a node can lag behind on RAC before considering instance offline. (Behind the ORACLERACMAXINSTANCELAG feature flag.)

API object	Path	Type	Name	Description
`ReplicationSpec`	N/A (value type)	property	`lockedProfile`	Indicates the replication profile is locked. (Behind the DATA_VAULT feature flag.)
`SyncStrategy`	N/A (value type)	API type	`SyncStrategy`	Base type for specific parameters for sync strategy.
`Tunable`	N/A (value type)	API type	`Tunable`	The name of the tunable and its value.
`TunableIdentifier`	N/A (value type)	API type	`TunableIdentifier`	The subsystem and name for a tunable.

3.4.28 API changes in Delphix 6.0.7.0

In Delphix 6.0.7.0, the new API version is 1.11.7. This section describes all API changes since API version 1.11.6, which was released with Delphix 6.0.6.0. All URL paths are relative to `/resources/json/delphix`.

3.4.28.1 What's changed

API object	Path	Type	Name	Change
`PgSQLDBClusterConfigConnectivity`	N/A (value type)	API type	`PgSQLDBClusterConfigConnectivity`	Deleted.
`OracleBaseDBConfig`	/sourceconfig	property	`nonSysCredentials`	Changed type from PasswordCredential to Credential.

API object	Path	Type	Name	Change
`OracleLinkFromExternal`	N/A (value type)	property	`nonSysCredentials`	Changed type from PasswordCredential to Credential.
`ReplicationSecureList`	N/A (value type)	API type	`ReplicationSecureList`	No longer behind feature flag MDD.
`SNMPV3Manager`	/service/snmp/v3/manager	property	`username`	Introduced a minimum length of 1.
`Source`	/source	property	`status`	Added DETACHED to the list of possible values.

3.4.28.2 What 's new

API object	Path	Type	Name	Description
`AbstractToolkit`	/toolkit	root operation	`schemaDefinitions`	Get the platform's JSON schema definitions that plugin schemas can reference.

API object	Path	Type	Name	Description
`ASESyncParameters`	N/A (value type)	property	`dropAndRecreateDevices`	If this parameter is set to true, it will drop the older devices and create new devices during manual sync operations instead of trying to remap the devices. This might increasethe space utilization on Delphix Engine.
`NamedKeyPairCredential`	N/A (value type)	API type	NamedKeyPairCredential	Username and key-pair credential.
`NamedPasswordCredential`	N/A (value type)	API type	NamedPasswordCredential	Pair of username and password credential.
`OracleSyncFromExternalParameters`	N/A (value type)	property	filesForFullBackup	List of datafiles to take a full backup of. This would be useful in situations where certain datafiles could not be backed up during previous SnapSync due to corruption or because they went offline.
`SsoConfig`	/service/sso	property	`entityId`	Audience Restriction (SP entity ID, Partner's Entity ID) of this engine as an SSO service provider.

API object	Path	Type	Name	Description
`SsoConfig`	/service/sso	property	`responseSkewTime`	Maximum time difference allowed between a SAML response and the engine's current time, in seconds. If not set, it defaults to 86,400 seconds (one day).
`SsoConfig`	/service/sso	property	`maxAuthenticationAge`	How far in the past to accept authentications to the identity provider, in seconds. If not set, it defaults to 120 seconds.
`SupportBundleConfiguration`	/service/support/bundle	property	`maxActions`	Maximum number of actions not referenced anywhere else to include from the metadata store. The most recent such actions are included.
`SupportBundleConfiguration`	/service/support/bundle	operation	`read`	Retrieve the specified SupportBundleConfiguration object.
`SupportBundleConfiguration`	/service/support/bundle	operation	`update`	Update the specified SupportBundleConfiguration object.

3.4.29 API changes in Delphix 6.0.6.0

In Delphix 6.0.6.0, the new API version is 1.11.6. This section describes all API changes since API version 1.11.5, which was released with Delphix 6.0.5.0. All URL paths are relative to `/resources/json/delphix`.

3.4.29.1 What's changed

API object	Path	Type	Name	Change
`OracleMultitenantProvisionParameters`	N/A (value object)	object property	`timeflowPointParameters`	The source timeflow point can now reference an Oracle non-multitenant database. In which case the non-multitenant database will be provisioned from the timeflow point to a virtual pluggable database.

3.4.30 API changes in Delphix 6.0.5.0

In Delphix 6.0.5.0, the new API version is 1.11.5. This section describes all API changes since API version 1.11.4, which was released with Delphix 6.0.4.0. All URL paths are relative to `/resources/json/delphix`.

3.4.30.1 What's changed

API object	Path	Type	Name	Change
`CloudStatus`	/service/cloud	property	`delphixDataServicesComponentStatus`	Made read-only for update.
`CloudStatus`	/service/cloud	property	`delphixDataServicesComponentInfo`	Made read-only for update.
`CloudStatus`	/service/cloud	root operation	`enable`	Added optional payload CloudEnableParameters.

API object	Path	Type	Name	Change
`OracleDBConfig`	/sourceconfig	property	`tdeKeystorePassword`	Impose a maximum length of 256.
`OracleInstance`	N/A (value object)	property	`instanceNumber`	Change type from number to integer.
`OracleSnapshot`	/snapshot	property	`redoLogSizeInBytes`	Change type from number to integer.
`OracleStartParameters`	N/A (value object)	property	`instances`	Change type from number to integer.
`OracleStopParameters`	N/A (value object)	property	`instances`	Change type from number to integer.
`TimeZone`	/timezone	property	`id`	Added enum values America/Nuuk and Asia/Qostanay.

3.4.30.2 What's new

API object	Path	Type	Name	Description
`CloudEnableParameters`	N/A (value object)	value type	`CloudEnableParameters`	Parameters to the Cloud Enable operation
`CloudStatus`	/service/cloud	property	`proxyMode`	Whether an HTTP proxy must be used to connect to Central Management.

API object	Path	Type	Name	Description
`CloudStatus`	/service/cloud	property	`proxyConfiguration`	Proxy configuration for communication with Delphix Central Management. This property is ignored unless the 'proxyMode' property is set to CLOUD_SPECIFIC_SETTING.
`CloudStatus`	/service/cloud	operation	`update`	Update the specified CloudStatus object. Payload is a CloudStatus.
`HostConfiguration`	N/A (value object)	property	`powerShellVersion`	The PowerShell version installed on the windows target host.
`OracleSTConvertedToPDBAttachData`	N/A (value object)	value type	`OracleSTConvertedToPDBAttachData`	Sub-type of OraclePDBAttachData.
`OracleTimeflow`	/timeflow	property	`tdeUUID`	Unique identifier for TimeFlow-specific TDE objects that reside outside of Delphix storage.
`OracleVirtualPdbSource`	/source	property	`parentTdeKeystorePassword`	The password of the keystore specified in parentTdeKeystorePath.
`OracleVirtualPdbSource`	/source	property	`tdeExportedKeyFileSecret`	Secret to be used while exporting and importing vPDB encryption keys if Transparent Data Encryption is enabled on the vPDB.

API object	Path	Type	Name	Description
`OracleVirtualPdbSource`	/source	property	`tdeUUID`	Unique identifier for PDB-specific TDE objects that reside outside of Delphix storage.
`StaticHostAddress`	/service/host/address	new API	`StaticHostAddress`	Static mapping of hostname to IP address.
`WindowsHost`	/host	property	`connectorVersion`	The Windows Connector version that is installed on the provided host.
`WindowsHost`	/host	property	`connectorDotNetFrameworkVersion`	The .NET Framework version used for Windows Connector Service.

3.4.31 API changes in Delphix 6.0.4.0

In Delphix 6.0.4.0, the new API version is 1.11.4. This section describes all API changes since API version 1.11.3, which was released with Delphix 6.0.3.0. All URL paths are relative to `/resources/json/delphix`.

3.4.31.1 What's changed

API object	Path	Type	Name	Change
`OracleDBConfigConnectivity`	N/A (value object)	object property	`password`	Name changed to credentials Type changed to Credential
`OracleBaseAttachData`	N/A (value object)	object property	`oracleFallbackCredentials`	Type changed to Credential

API object	Path	Type	Name	Change
`OracleBaseDBConfig`	/sourceconfig	object property	`credentials`	Type changed to Credential
`OracleBaseLinkData`	N/A (value object)	object property	`oracleFallbackCredentials`	Type changed to Credential
`OracleDatabaseContainer`	/database	object property	`tdeProvisioningEnabled`	Deleted
`OracleManagedSource`	/source	object property	`mountBase`	Format is now unixrestrictedpath
`OracleVirtualPdbSource`	/source	object property	`parentTdeKeystorePath`	Format is now unixrestrictedpath
`UnixRuntimeMountInformation`	N/A (value object)	object property	`nfsVersionReason`	Added enum value INCOMPLETE_V4_CONFIG
`X500DistinguishedNameComposite`	N/A (value object)	object property	`dname`	Minimum length is now 1

3.4.31.2 What's new

API object	Path	Type	Name	Description
`X509Certificate`	/service/certificate	new API	`X509Certificate`	X509 Certificate.

3.4.32 API changes in Delphix 6.0.3.0

In Delphix 6.0.3.0, the new API version is 1.11.3. This section describes all API changes since API version 1.11.2, which was released with Delphix 6.0.2.0. All URL paths are relative to `/resources/json/delphix`.

3.4.32.1 What's changed

API object	Path	Type	Name	Change
ApplyVersionParameters	N/A (value object)	object property	verify	Default changed from true to false.
OracleBaseAttachData	N/A (value object)	object property	dbUser dbCredentials	Name changed to oracleFallbackUser. The database user. Optional if bequeath connections are enabled (to be used in case of bequeath connection failures). Name changed to oracleFallbackCredentials. The password for the database user. Optional if bequeath connections are enabled (to be used in case of bequeath connection failures).
OracleBaseLinkData	N/A (value object)	object property	dbUser dbCredentials	Name changed to oracleFallbackUser. The database user. Optional if bequeath connections are enabled (to be used in case of bequeath connection failures). Name changed to oracleFallbackCredentials. The password for the database user. Optional if bequeath connections are enabled (to be used in case of bequeath connection failures).
Plugin	/toolkit	object property	status	Deleted

3.4.32.2 What's new

API object	Path	Type	Name	Description
AbstractToolkit	/toolkit	object property	status	The status of the toolkit. ACTIVE indicates toolkit is actively referenced and in use. INACTIVE means toolkit needs to go through a manual upgrade operation before it can be used.
Certificate	/service/tls/caCertificate /service/tls/endEntityCertificate	object property	isCertificateAuthority	Whether this certificate is a Certificate Authority (CA).
ConfiguredStorageDevice	/storage/device	object property	fragmentation	Percent fragmentation for this device.
CyberArkVaultCredential	N/A (value object)	subclass of Credential	CyberArkVaultCredential	CyberArk vault based security credential.
DNSConfig	/service/dns	object property	source	The source of the DNS configuration (STATIC or DHCP).
HashiCorpVaultCredential	N/A (value object)	subclass of Credential	HashiCorpVaultCredential	HashiCorp vault based security credential.
LicenseInfo	/license	object	LicenseInfo	Retrieve all external licenses.
NetworkDSPTestParameters	N/A (value object)	object property	xportScheduler	The transport scheduler to use.

API object	Path	Type	Name	Description
OracleInstall	/repository	object property	oracleBaseConfig oracleBaseHome	The Oracle Base Config directory. The Oracle Base Home directory.
OracleVirtualPdbSource	/source	object property	parentTdeKeystorePath	Path to a copy of the parent's Oracle transparent data encryption keystore on the target host. Required to provision from snapshots containing encrypted database files.
PasswordVault	/service/passwordVault	new API	PasswordVault	Password vault configuration.
Plugin	/toolkit	object property	luaName minimumLuaVersion	The name of the LUA toolkit that this plugin can upgrade. The minimum version (in major.minor format) of a LUA toolkit that this plugin can upgrade.
RunDefaultPowerShellOnSourceOperation	N/A (value object)	subclass of SourceOperation	RunDefaultPowerShellOnSourceOperation	A user-specifiable operation that runs a PowerShell command (using default version) on the target host.

API object	Path	Type	Name	Description
SnapshotCapacityData	/capacity/snapshot	object property list parameter	namespace snapshotTimezone snapshotLatestChange namespace	Reference to the namespace to which this snapshot belongs. Time zone of the source database at the time the snapshot was taken. The timestamp of the latestChangePoint of the associated snapshot. The namespace to list snapshot data for. If null, will limit returned snapshots to the default namespace.
StatisticSlice	/analytics	getData parameter	count	The number of data points to return. Engine metadata points will be combined in order to to meet this requirement. When count is specified at least two of the other getData parameters must be specified.
SystemInfo	/system	object property	poolFragmentation	Percent fragmentation for the domain0 pool.

3.5 Support matrices

This section covers the following topics:

- Kerberos support matrix (see page 429)
- Data source certifications (see page 434)
- Select connectors matrix (see page 438)

You can find all other dataset support matrices under the **Support and Requirements** sections of that particular datasets.

3.5.1 Kerberos support matrix

> Delphix Support Policies specifically list Major and Minor release coverage. If a minor release is listed as covered, then all patch releases under that minor release are certified.

Key:

Color	Supported?
Y	Yes
N	No
NA	Not Applicable

3.5.1.1 Oracle - Red Hat Enterprise Linux (RHEL)

Supported OS Version	Oracle 10g	Oracle 11gR1	Oracle 11gR2	Oracle 12cR1	Oracle 12cR2	Oracle 18c
	Supported DBMS Version					
RHEL 5.5	Y	Y	Y	NA	NA	NA
RHEL 5.6	Y	Y	Y	Y	NA	NA
RHEL 5.7	Y	Y	Y	Y	NA	NA
RHEL 5.8	Y	Y	Y	Y	NA	NA
RHEL 5.9	Y	Y	Y	Y	NA	NA
RHEL 5.10	Y	Y	Y	Y	NA	NA

RHEL 5.11	Y	Y	Y	Y	NA	NA
RHEL 6.0	NA	NA	Y	Y	NA	NA
RHEL 6.1	NA	NA	Y	Y	NA	NA
RHEL 6.2	NA	NA	Y	Y	NA	NA
RHEL 6.3	NA	NA	Y	Y	NA	NA
RHEL 6.4	NA	NA	Y	Y	Y	Y
RHEL 6.5	NA	NA	Y	Y	Y	Y
RHEL 6.6	NA	NA	Y	Y	Y	Y
RHEL 6.7	NA	NA	Y	Y	Y	Y
RHEL 6.8	NA	NA	Y	Y	Y	Y
RHEL 6.9	NA	NA	Y	Y	Y	Y
RHEL 6.10	NA	NA	Y	Y	Y	Y
RHEL 7.0	NA	NA	Y	Y	Y	Y
RHEL 7.1	NA	NA	Y	Y	Y	Y
RHEL 7.2	NA	NA	Y	Y	Y	Y
RHEL 7.3	NA	NA	Y	Y	Y	Y
RHEL 7.4	NA	NA	Y	Y	Y	Y
RHEL 7.5	NA	NA	Y	Y	Y	Y

3.5.1.2 SAP ASE - Red Hat Enterprise Linux (RHEL)

Supported OS Version	Supported DBMS Version		
	ASE 15.5	ASE 15.7	ASE 16
RHEL 5.5	Y	Y	**NA**
RHEL 5.6	Y	Y	**NA**
RHEL 5.7	Y	Y	NA
RHEL 5.8	Y	Y	NA
RHEL 5.9	Y	Y	NA
RHEL 5.10	Y	Y	NA
RHEL 5.11	Y	Y	NA
RHEL 6.0	N	N	NA
RHEL 6.1	N	N	N
RHEL 6.2	Y	Y	N
RHEL 6.3	Y	Y	N
RHEL 6.4	Y	Y	N
RHEL 6.5	Y	Y	Y
RHEL 6.6	Y	Y	Y
RHEL 6.7	Y	Y	Y
RHEL 6.8	Y	Y	Y

Supported OS Version	Supported DBMS Version		
RHEL 6.9	Y	Y	Y
RHEL 6.10	Y	Y	Y
RHEL 7.0	NA	Y	Y
RHEL 7.1	NA	Y	Y
RHEL 7.2	NA	Y	Y
RHEL 7.3	NA	Y	Y
RHEL 7.4	NA	Y	Y
RHEL 7.5	NA	Y	Y

3.5.1.3 DB2 support matrix

- ESE: Enterprise Server Edition
- AESE: Advanced Enterprise Server Edition
- For the supported DB2 versions, Delphix supports the corresponding DB2 Developer edition where the vendor, IBM, supports it
- 64-bit OS support only

3.5.1.4 Red Hat Enterprise Linux (RHEL)

Supported OS Version	Supported DBMS Version			
	AESE 10.5	AESE 10.5	ESE 11.1	AESE 11.1
RHEL 6.0	N	N	N	N

RHEL 6.1	N	N	N	N
RHEL 6.2	N	N	N	N
RHEL 6.3	N	N	N	N
RHEL 6.4	Y	Y	N	N
RHEL 6.5	Y	Y	N	N
RHEL 6.6	Y	Y	N	N
RHEL 6.7	Y	Y	Y	Y
RHEL 6.8	Y	Y	Y	Y
RHEL 6.9	Y	Y	Y	Y
RHEL 7.0	Y	Y	Y	Y
RHEL 7.1	Y	Y	Y	Y
RHEL 7.2	Y	Y	Y	Y
RHEL 7.3	Y	Y	Y	Y
RHEL 7.4	Y	Y	Y	Y

DB2 11.1 Developer Edition has been certified on RHEL 6.7.

3.5.1.5 DB2 on Advanced Interactive eXecutive (AIX)

Supported OS Version	Supported DBMS Version

	AESE 10.5	AESE 10.5	ESE 11.1	AESE 11.1
AIX 6.1	N	N	NA	NA
AIX 7.1	Y	Y	Y	Y
AIX 7.2	Y	Y	Y	Y

3.5.2 Data source certifications

3.5.2.1 Legal notice

IMPORTANT: The policies outlined in this document are subject to change. The policies and timelines are provided as rough estimates when certifications of interoperability/connectivity with third-party products may be completed and are subject to whether such interoperability/connectivity is part of the Delphix product roadmap and support policy and should not be regarded as a commitment.

Speak with Delphix Product Management if you need more specific information or are unsure of these policies.

3.5.2.2 Goals

The goal of this certification cadence is to maintain a cadence for certifications of new software releases.

The objectives are:

1. To set reasonable expectations for our organization and customers regarding how and when we will address certifications
2. To define an intended timeframe for certification of new data sources versions and operating systems versions to maintain an up-to-date support matrix

3.5.2.2.1 Certification expectations:

Please note that certification only covers previously supported functionality of the Delphix Engine. Any new features and functionality introduced in the database or operating system (DB/OS) will be tracked through the regular release planning and roadmap process and considered as features/functionality/fixes.

Specifically, the following paths exist with varying delivery timeline commitments, in the following priority order:

1. Security Response Policy[82]: In case security issues are discovered in a certain DB and/or OS version, we will follow our security response policy, which supersedes the goals described in this document.

82 https://support.delphix.com/Support_Policies/KBA1504_Delphix_Support_Policies

2. Certifications: As described in the document.
3. Roadmap Features: No policies outlined. This follows the regular roadmap planning process.

3.5.2.3 Data sources

This documentation currently applies only to the following data source integrations:

- Oracle
- SQL Server
- SAP ASE

3.5.2.4 Cadence for data source database version changes

Release type	Certification details	Goal
A new version of a database with all currently supported operating systems	Certification with all supported operating systems	8 months from new version GA date

3.5.2.5 Cadence for operating system version changes

Release type	Certification details	Goal
A new major operating system version with all currently supported databases	Certification with all supported versions of the database	6 months from new version GA date
A new update* operating system version with all currently supported databases	Certification with all supported versions of the database	4 months from new version GA date

*Terminology for an 'update' version varies across operating systems. See the Appendix for more information.

3.5.2.6 Appendix

3.5.2.6.1 Product lifecycle of Oracle related DB/OS vendors:

Software	Product lifecycle	Notes
Oracle Database	Link[83]	12.1 is Oracle Enterprise Edition (EE)[84]
Red Hat Enterprise Linux	Link[85]	
Solaris (x86 and SPARC)	Link[86]	
SuSE Linux (SLES)	Link[87]	
IBM AIX	Link[88]	
HP-UX	Link[89]	

3.5.2.6.2 Product lifecycle of SQL server related DB/OS vendors:

Software	Product Lifecycle
Microsoft SQL Server	Link[90]
Microsoft Windows	Link[91]

83 http://www.oracle.com/us/support/library/lifetime-support-technology-069183.pdf
84 https://docs.oracle.com/database/121/DBLIC/editions.htm#DBLIC116
85 https://access.redhat.com/support/policy/updates/errata
86 http://www.oracle.com/us/support/library/lifetime-support-hardware-301321.pdf
87 https://www.suse.com/lifecycle/
88 http://www-01.ibm.com/support/docview.wss?uid=isg3T1012517
89 http://h20195.www2.hpe.com/V2/GetDocument.aspx?docname=4AA4-7673ENW&cc=us&lc=en
90 https://support.microsoft.com/en-us/lifecycle
91 https://support.microsoft.com/en-us/lifecycle

3.5.2.6.3 Product lifecycle of SAP ASE related DB/OS vendors:

Software	Product Lifecycle
SAP ASE	Link[92]
Red Hat Enterprise Linux	Link[93]
Solaris (x86 and SPARC)	Link[94]
SuSE Linux (SLES)	Link[95]
IBM AIX	Link[96]

3.5.2.6.4 Footnote 1: Update version terminology

Operating System releases typically include updated versions, the terminology for which varies depending on the vendor. Below are the various operating systems which we support, as well as the terminology for updates to use with the document above.

Vendor	Update version terminology	Definition
Red Hat Enterprise Linux	Update version	Installed on top of major releases to provide larger-scale point-release updates via Red Hat Network, after initial installation via traditional methods (PXE, DVD, etc.).
Solaris (x86 and SPARC)	Update release	Update release number. The update release number for this Oracle Solaris release. The update value is 0 for the first customer shipment of an Oracle Solaris release, 1 for the first update of that release, 2 for the second update of that release, and so forth.

92 https://wiki.scn.sap.com/wiki/display/SYBASE/Targeted+ASE+16.0+Release+Schedule+and+CR+list+Information
93 https://access.redhat.com/support/policy/updates/errata
94 http://www.oracle.com/us/support/library/lifetime-support-hardware-301321.pdf
95 https://www.suse.com/lifecycle/
96 http://www-01.ibm.com/support/docview.wss?uid=isg3T1012517

Vendor	Update version terminology	Definition
SuSE Linux (SLES)	Service Pack (SP) release	Service Packs are minor updates and enhancements, usually with security fixes, that are applied to the major version of the OS.
IBM AIX	Service Pack (SP) release	A Technology Level (TL) gives new features, while an SP contains fixes for problems that are critical and can't wait until the next TL. Service Packs are limited to minimal corrections that don't change the way things work or add any new functionality.
HP-UX	Patch release	Patch sets are groups of patches that should be installed to support common HPE products and sub-systems.
Windows	New Release (R) version or Service Pack (SP) release	A service pack (SP) is a collection of updates and fixes, called patches, for an operating system or a software program. Many of these patches are often released before a larger service pack, but the service pack allows for an easy, single installation.

3.5.3 Select connectors matrix

Color	Supported?
Y	Yes
N	No
NA	Not Applicable

3.5.3.1 Couchbase

3.5.3.1.1 Data source version and OS version

	RHEL 6.x	RHEL 7.0	RHEL 8.0	Windows
Couchbase 6.0.x	Y	Y	Y	N
Couchbase 6.5.x	Y	Y	Y	N
Couchbase 6.6.x	Y	Y	Y	N
Couchbase 7.1.x	Y	Y	Y	N

3.5.3.1.2 Database/Plugin support matrix

	Plugin version 1.2.1	Plugin version 1.2.2	Plugin version 1.3.0
Couchbase 6.0.x	Y	Y	Y
Couchbase 6.5.x	Y	Y	Y
Couchbase 6.6.x	Y	Y	Y
Couchbase 7.1.x	N	Y	Y

3.5.3.1.3 Couchbase Edition Compatibility Matrix

	Community Edition XDCR only	Enterprise Edition
Couchbase 6.0.x	Y	Y
Couchbase 6.5.x	Y	Y

	Community Edition XDCR only	Enterprise Edition
Couchbase 6.6.x	Y	Y
Couchbase 7.1.x	Y	Y

3.5.3.1.4 Size in TB / sizing

- No known restrictions.
- Memory, Network, IOPS metrics from Couchbase GUI

3.5.3.1.5 Frequency of refreshes

No specific limitations

3.5.3.1.6 Source backup policy

- Supports backups using Couchbase Backup Manager (cbbkpmgr)
- Supports Couchbase XDCR technology

3.5.3.1.7 Use case (PIT Provisioning, Dev/Test)

- Does not support Point-in-Time Provisioning.
- Not appropriate for high throughput performance requirement.

3.5.3.1.8 Delphix option (SDD, API Calls, GUI Icons, GUI PIT Prov)

- V2P is not supported.

3.5.3.1.9 Timestamp on snapshot metadata

This plugin ingests Backups at any time, so the time flow may not go in linear order. Time of Backup is not readily available.

3.5.3.2 Architecture options

- Only 1 VDB per host is supported (couchbase restriction of 1 instance per Node)
- Dedicated 1 Staging host is needed for each dSource created using XDCR technology

3.5.3.2.1 Authentication type (LDAP, Kerberos,)

- Kerberos is not supported.
- LDAP not tested.

3.5.3.2.2 Ingestion type (Backup, Dump, Replication, Snapshot)

- Couchbase Backup Manager (cbbkpmgr)
- XDCR

3.5.3.2.3 Link to documentation

https://delphix.github.io/couchbase-plugin

3.5.3.3 Mongo

3.5.3.3.1 Data Source version and OS version

Supported Mongo Version	mongoDB 4.2.x	mongoDB 4.4.x	mongoDB 5.0.x	mongoDB 6.0.x
RHEL 6.x	Y	Y	Y	Y
RHEL 7.x	Y	Y	Y	Y
RHEL 8.x	Y	Y	Y	Y
CentOS 6.x	Y	Y	Y	Y
CentOS 7.x	Y	Y	Y	Y
CentOS 8.x	Y	Y	Y	Y
Windows	N	N	N	N

3.5.3.3.2 Engine Compatibility Matrix

MongoDB Version	Mongopy 1.0.0	Mongopy 1.0.1	Mongopy 1.0.2	Mongopy 1.1.0	Mongopy 1.2.0	Mongopy 1.2.1	Mongopy 1.3.0
6.0.7.x-6.0.11.x	Y	Y	N	N	N	N	N
6.0.12.x-6.0.15.x	Y	Y	Y	Y	N	N	N
7.x-18.x	N	N	N	N	Y	Y	Y

3.5.3.3.3 Mongo/Plugin Version Compatibility Matrix

MongoDB Version	Mongopy 1.0.0	Mongopy 1.0.1	Mongopy 1.0.2	Mongopy 1.1.0	Mongopy 1.2.0	Mongopy 1.2.1	Mongopy 1.3.0
4.x-5.x	Y	Y	Y	Y	Y	Y	Y
6.x	N	N	N	N	Y	Y	Y

3.5.3.3.4 Size in TB / sizing

- Not tested with large datasets.
- Memory, Network, IOPS metrics from Mongo OPS Manager

3.5.3.3.5 Frequency of refreshes

No specific limitations

3.5.3.3.6 Source backup policy

- Sharded Mongo always ingest full. Non-Sharded mongo ingestion may be full or incremental based on the mechanism used for ingestion.
- Sharded : Full
- Non-Sharded (Using Backups) - Full
- Non-sharded (Using replicaset member) - incremental

- Mongodump - Full or incremental based on type of source and type of ingestion

3.5.3.3.7 Use case (PIT Provisioning, Dev/Test)

- Does not support Point-in-Time Provisioning.
- Not appropriate for high throughput performance requirement.

3.5.3.3.8 Delphix option (SDD, API Calls, GUI Icons, GUI PIT Prov)

- V2P is not supported.

3.5.3.3.9 Timestamp on snapshot metadata

This plugin ingests Backups at any time, so the time flow may not go in linear order. Time of Backup is not readily available.

3.5.3.4 Architecture options

3.5.3.4.1 Sharding

Supported

	Unsharded VDB	Sharded VDB
Unsharded Source >= 3.6	Y	N
Sharded Source >= 4.2	N	Y

Example: Sharded sources are supported, but require backup vendor MongoOPS Manager to create backup.

3.5.3.4.2 Authentication type (LDAP, Kerberos,)

- LDAP and SCRAM supported
- Kerberos not supported

3.5.3.4.3 Ingestion type (Backup, Dump, Replication, Snapshot)

- MongoOPS Manager Backup Files
- Mongodump
- Replicaset

3.5.3.4.4 Encryption (At Rest, TLS/SSL, Encrypted backups)

Supported

3.5.3.4.5 Link to documentation

https://delphix.github.io/mongo-plugin/

3.5.3.5 Microsoft SQL backup ingestion

3.5.3.5.1 Data source version and OS version

	Windows 2012	Windows 2012 R2	Windows 2016	Windows 2019	Windows 2022
SQLServer 2012	Y	Y	Y	N	N
SQLServer 2014	Y	Y	Y	N	N
SQLServer 2016	Y	Y	Y	Y	N
SQLServer 2017	Y	Y	Y	Y	Y
SQLServer 2019	N	N	Y	Y	Y
SQLServer 2022	N	N	Y	Y	Y

3.5.3.5.2 Size in TB / sizing

No sizing limitation

3.5.3.5.3 Frequency of refreshes

No limitation

3.5.3.5.4 Source backup policy

dSource sync is managed by customers and can apply backup on staging whenever required.

3.5.3.5.5 Use case (PIT provisioning, Dev/Test)

Does not support Point-in-Time Provisioning.

3.5.3.5.6 Delphix option (SDD, API Calls, GUI Icons, GUI PIT Prov)

V2P is not supported

3.5.3.5.7 Timestamp on snapshot metadata (Affects other products too (HANA, but not Oracle/ SQLServer)

This plugin ingests Backups at any time, so the time flow may not go in linear order.

3.5.3.6 Architecture options

3.5.3.6.1 Failover clusters and availability groups

- Does not support Windows Failover Clusters or Availability Groups for Windows Staging and Target environments.
- Authentication Type (LDAP, Kerberos,)

Ex: LDAP and Kerberos are not supported at this time, but please contact Ranzo Taylor if you have a customer who requires this.

3.5.3.6.2 Ingestion type (Backup, Dump, Replication, Snapshot)

Full backups, Differential backups, Log Shipping, Transaction replication

3.5.3.6.3 Backup vendor (Commvault, Native, Replication)

All backup vendors are supported

3.5.3.6.4 Encryption (At Rest, TLS/SSL, Encrypted Backups)

- Encryption at Rest should work, but has not been tested.

- Encrypted Backups should work, and has not been tested.

3.5.3.6.5 Link to documentation

https://delphix.github.io/mssql_plugin_doc

3.5.3.7 Oracle backup ingestion

3.5.3.7.1 Data source version and OS version

	RHEL 6.x	RHEL 7.0	Windows
Oracle 9i/10g/11r1	N	N	N
Oracle >= 11gR2	Y	Y	N

3.5.3.7.2 Size in TB / sizing

No Restrictions.

3.5.3.7.3 Frequency of refreshes

No specific limitations

3.5.3.7.4 Source backup policy

No limitations for disk.

For Tape Backup - Autobackup should be enabled.

3.5.3.7.5 Use case (PIT Provisioning, Dev/Test)

- Disk - Archivelog need to be made available on target host for Point-in-Time Provisioning.
- Tape - Tape library installed on target and authorized to access tape from target host.
- Support using timestamp.

3.5.3.7.6 Delphix option (SDD, API Calls, GUI Icons, GUI PIT Prov)

- V2P is not supported.

3.5.3.7.7 Timestamp on snapshot metadata

This plugin ingests Backups at any time, so the time flow may not go in linear order. Time of Backup is not readily available.

3.5.3.8 Architecture Options

Full	Y	Y	Y	Y	Y	NA
Partial	Y	Y	Y	Y	Y	NA
PDB	Y	Y	Y	Y	Y	N
Partial PDB	Y	Y	Y	Y	Y	N
Use Existing CDB	NA	NA	NA	N	NA	N
Seed	NA	NA	NA	NA	N	NA

3.5.3.8.1 RAC

Manual Addition, Selection of nodes and Checks needed to create successful RAC VDBs.

3.5.3.8.2 Ingestion type (Backup, Dump, Replication, Snapshot)

- RMAN database backup

3.5.3.8.3 Backup vendor (Oracle RMAN, ZDLRA, DDBOOST, NETBACKUP, Generic Tape Drivers)

- Oracle RMAN
- ZDLRA
- DDBOOST
- NETBACKUP
- Other Tape Vendors not listed above not tested but most probably will work

3.5.3.8.4 Link to documentation

https://delphix.github.io/obi-plugin/

3.5.3.9 SAPIQ

3.5.3.9.1 Data source version and OS version

Supported OS Version	SAPIQ 16.0	SAPIQ 16.1
RHEL 7.0	Y	Y
RHEL 7.4	Y	Y
RHEL 7.9	Y	Y
Windows x.x	N	N

3.5.3.9.2 Size in TB / sizing

Tested at the customer with 600-800GB ingestion size.

3.5.3.9.3 Frequency of refreshes

No limitation

3.5.3.9.4 Source backup policy

Customers will manage backup located on the staging host required to create dSource.

3.5.3.9.5 Use case (PIT Provisioning, Dev/Test)

Does not support Point-in-Time Provisioning.

3.5.3.9.6 Delphix option (SDD, API Calls, GUI Icons, GUI PIT Prov)

- V2P is not supported.
- Password Vault is not supported.
- SDD is not supported.

3.5.3.9.7 Timestamp on snapshot metadata

This plugin ingests backups at any time, so the time flow may not go in linear order. Time of Backup is not readily available.

3.5.3.10 Architecture options

Supports SAPIQ Simplex Architecture for VDBs.

Architecture	SAPIQ 16.0	SAPIQ 16.1
Simplex	Y	Y
Multiplex	N	N

- IQ Plugin converts multiplex architecture to simplex during VDB provision.
- Ingest SAPIQ source databases which are backed up using SAPIQ built-in backup mechanism.
- Data Ingestion to Delphix Engine will always start with a full DB backup.
- As best practices, it is recommended to keep a similar SAPIQ version in all environments (source, staging and target).

3.5.3.10.1 Authentication type (LDAP, Kerberos)KE

- Kerberos not supported.
- LDAP not tested.

3.5.3.10.2 Ingestion type (Backup, Dump, Replication, Snapshot)

Full backups for Simplex Architecture Tested.

3.5.3.10.3 Backup Vendor

Tested with manual SAPIQ backups.

3.5.3.10.4 Encryption (At Rest, TLS/SSL, Encrypted Backups)

- Encryption at Rest has not been tested.
- Encrypted Backups have not been tested.

3.5.3.10.5 Link to documentation

https://delphix.github.io/SAPIQ/

3.5.3.11 Cockroach

3.5.3.11.1 Data source version and OS version

Supported OS Version	CRDB v21.1.x	CRDB v21.2.x	CRDB v22.1.x	CRDB v22.2.x
RHEL 7.x	Y	Y	Y	Y
RHEL 8.x	Y	Y	Y	Y
Windows	N	N	N	N

3.5.3.11.2 Architecture options

Architecture	Disk	Seed	AWS	GCP	Azure
Full	Y	Y	Y	N	N
Incremental	Y	Y	Y	N	N

3.5.3.11.3 Engine compatibility matrix

Engine Version	CRDB v1.0.0	CRDB v1.1.0	CRDB v1.1.1	CRDB v1.1.2
6.0.12.0 - 6.0.17.x	Y	Y	Y	Y
7.0.0.0 - 13.0.0.0	Y	Y	Y	Y

3.5.3.11.4 PostgreSQL JDBC driver matrix

PostgreSQL Driver Version	Supported
42.3.2	Y
42.3.6	Y
42.4.0	Y

3.5.3.11.5 JDBC masking support matrix

Masking Mode	Feature	Availability
In-Place Masking	Multi-Tenant	Available
	Streams/Threads	Available
	Batch Update	Available
	Drop Indexes	Unavailable
	Disable Constraint	Unavailable
	Disable Trigger	Unavailable NA for CockroachDB[97]
	Identity Column Support	Available
On-The-Fly Masking		Unavailable
Profiling	Multi-Tenant	Available
	Streams	Available

97 https://www.cockroachlabs.com/docs/stable/sql-feature-support.html

- Drop Indexes and Disable Constraint are not supported natively while masking CockroachDB. Use pre/post scripts if you want to use these features.
- CockroachDB supports rowid natively for tables with no primary key defined. It will automatically create a primary key over a hidden, INT-typed column named rowid.
- To create a successful JDBC connection to CockroachDB cluster running in secure mode, please enable the password authentication without TLS using flag `--accept-sql-without-tls` with cockroach start command.
- To mask the CockroachDB data source where CDC (change data capture) feature is enabled, disable all the running change-feed jobs using the below commandCockroachDB supports rowid natively for tables with no primary key defined. It will automatically create a primary key over a hidden, INT-typed column named rowid.

- ```
  SHOW CHANGEFEED JOBS; PAUSE JOBS (WITH x AS (SHOW CHANGEFEED JOBS) SELECT *
  FROM x WHERE status = ('running'));
  ```
  Resume CDC job once CockroachDB masking is completed.

- ```
  RESUME JOBS (WITH x AS (SHOW CHANGEFEED JOBS) SELECT * FROM x WHERE status =
  ('paused'));
  ```

3.5.3.11.6 Ingestion type

- AWS S3 Backup Ingestion
- Seed Ingestion

3.5.3.11.7 Limitations

- CockroachDB Ingestion from Azure, GCP, and Disk is not supported.
- CockroachDB Point-in-time restore is not supported.
- Delphix V2P is not supported.
- Delphix Password Vault is not supported.

3.5.3.11.8 Link to documentation

https://delphix.github.io/crdb/

3.5.3.12 Cassandra

3.5.3.12.1 Delphix engine compatibility

Engine Version	Cassandra v1.0.0	Cassandra v1.1.0	Cassandra v1.1.1	Cassandra v1.2.0
6.0.16.0 - 22.x	Y	Y	Y	Y

3.5.3.12.2 Supported Database / OS

Supported DBMS Version	DataStax Cassandra v5.1.16	DataStax Cassandra v6.8.9 - v6.8.33	Apache Cassandra v4.0.7 - v4.1.1
RHEL 7.4	Y	Y	Y
RHEL 7.9	Y	Y	Y
RHEL 8.0	Y	Y	Y
Windows	N	N	N

3.5.3.12.3 Supported Ingestion

Cassandra Distribution	Staging Push	Disk	AWS
DataStax Enterprise	Y	N	N
Open-Source Apache	Y	N	N

3.5.3.12.4 Link to documentation

https://delphix.github.io/cassandra/

3.6 Upgrade matrix

This section lists all the Delphix Engine versions that a user can upgrade to the required or the latest version.

- **Upgrades from versions older than 5.3.0**: Self-service upgrades to version 6.0.x.x are not supported for Delphix Engines running version 5.3.0 or older.
- **Upgrades from versions 5.3.6.0 to 5.3.9.0**: These versions can be upgraded to version 6.0.6.1. Following this, further upgrades to newer versions are possible.
- **Delphix upgrade policy for version 6.0.10.0 and Later**: Starting with version 6.0.10.0, direct upgrades are only supported for versions released within the last 12 months.
- **Direct upgrades to version 6.0.10.1**:
  - Versions 6.0.4.1 and above can directly upgrade to version 6.0.10.1.
  - Versions 6.0.0.0 to 6.0.4.0 can be upgraded to 6.0.7.0 first, and then to 6.0.10.1.

Version	Date Released	Can Upgrade To	VDB Downtime Required
> 5.3.5.0	Aug 8th, 2019	Contact Support	Yes
5.3.6.0	Oct 10th, 2019	6.0.0.0 - 6.0.6.1	Yes
5.3.7.0	Dec 9th, 2019	6.0.1.0 - 6.0.6.1	Yes
5.3.7.1	Jan 13th, 2020	6.0.1.0 - 6.0.6.1	Yes
5.3.8.0	Feb 10th, 2020	6.0.1.0 - 6.0.6.1	Yes
5.3.8.1	Feb 26th, 2020	6.0.1.0 - 6.0.6.1	Yes
5.3.9.0	Apr 14th, 2020	6.0.2.0 - 6.0.6.1	Yes
6.0.0.*	Jan 22nd, 2020	6.0.0.* - 6.0.7.*	Optional
6.0.1.*	Mar 19th, 2020	6.0.1.* - 6.0.7.*	Optional
6.0.2.*	May 7th, 2020	6.0.2.* - 6.0.7.*	Optional

Version	Date Released	Can Upgrade To	VDB Downtime Required
6.0.3.*	Jul 31st, 2020	6.0.3.* - 6.0.7.*	Optional
6.0.4.*	Sep 10th, 2020	6.0.4.* - 6.0.7.*	Optional
6.0.5.*	Nov 5th, 2020	6.0.5.* - 6.0.11.*	Optional
6.0.6.*	Jan 21st, 2021	6.0.6.* - 6.0.12.*	Optional
6.0.7.*	Mar 15th, 2021	6.0.7.* - 6.0.13.*	Optional
6.0.8.*	May 13th, 2021	6.0.8.* - 6.0.14.*	Optional
6.0.9.*	Jul 13th, 2021	6.0.9.* - 6.0.15.*	Optional
6.0.10.*	Sep 9th, 2021	6.0.10.* - 6.0.16.*	Optional
6.0.11.*	Nov 11th, 2021	6.0.11.* - 6.0.17.*	Optional
6.0.12.*	Jan 14th, 2022	6.0.12.* - 7.0.0.*	Optional
6.0.13.*	Mar 10th, 2022	6.0.13.* - 9.0.0.*	Optional
6.0.14.*	May 20th, 2022	6.0.14.* - 11.0.0.*	Optional
6.0.15.*	Jul 7th, 2022	6.0.15.* - 13.0.0.*	Optional
6.0.16.*	Sep 8th, 2022	6.0.16.* - 15.0.0.*	Optional
6.0.17.*	Nov 15th, 2022	6.0.17.* - 17.0.0.*	Optional
7.0.0.*	Jan 12th, 2023	7.0.0.* - 19.0.0.*	Optional
8.0.0.*	Feb 13th, 2023	8.0.0.* - 20.0.0.*	Optional
9.0.0.*	Mar 13th, 2023	9.0.0.* - 21.0.0.*	Optional

Version	Date Released	Can Upgrade To	VDB Downtime Required
10.0.0.*	Apr 13th, 2023	10.0.0.* - 22.0.0.*	Optional
11.0.0.*	May 24th, 2023	11.0.0.* - 23.0.0.*	Optional
12.0.0.*	Jun 21st, 2023	12.0.0.* - 23.0.0.*	Optional
13.0.0.*	Jul 19th, 2023	13.0.0.* - 23.0.0.*	Optional
14.0.0.*	Aug 23rd, 2023	14.0.0* - 23.0.0.*	Optional
15.0.0.*	Sep 20th, 2023	15.0.0.* - 23.0.0.*	Optional
16.0.0.*	Oct 18th, 2023	16.0.0.* - 23.0.0.*	Optional
17.0.0.*	Nov 21st, 2023	17.0.0.* - 23.0.0.*	Optional
18.0.0.*	Dec 20th, 2023	18.0.0.* - 23.0.0.*	Optional
19.0.0.*	Jan 24th, 2024	19.0.0.* - 23.0.0.*	Optional
20.0.0.*	Feb 21st, 2024	20.0.0.* - 23.0.0.*	Optional
21.0.0.*	Mar 20th, 2024	21.0.0.* - 23.0.0.*	Optional
22.0.0.*	Apr 17th, 2024	22.0.0.* - 23.0.0.*	N/A
23.0.0.*	May 22nd, 2024	23.0.0.*	

The * in the above table represents patch releases. When upgrading from X.Y.Z.x to X.Y.Z.x, the target version must be greater than the source version. Please refer to the How to Upgrade (see page 925) article for steps, or check with Delphix Support to perform the upgrade.

1. VDB Downtime is caused by a reboot of the Delphix Engine when DelphixOS is modified by an upgrade.
2. VDB Downtime may be optional for an upgrade when a release contains DelphixOS changes that are also optional.

3.7 Tested browser and operating systems

This topic describes the Web browsers and operating systems that have been tested for use with the Delphix Management application.

The following table lists the supported browsers and operating systems.

OS	Browsers
Windows 11	Microsoft Edge, Firefox, Chrome
Windows 10	Microsoft Edge, Firefox, Chrome
Windows 8.1	Microsoft Edge, Firefox, Chrome
Windows 7	Microsoft Edge, Firefox, Chrome
Mac OS X	Microsoft Edge, Firefox, Chrome

Delphix supports the latest version of Microsoft Edge, Firefox, and Chrome as well as the immediate prior version.

3.8 Deprecated and end-of-life features

3.8.1 Release 23.0.0.0

3.8.1.1 Deprecated features

Support for the following OS versions have been deprecated in this release:

- **RHEL 4.x and 5.x**
- **AIX 5.x and 6.x**

3.8.2 Release 21.0.0.0

3.8.2.1 Deprecated features

- **Delphix Self-Service**
  Has been deprecated and will reach its end of life in September 2025. Visit the community blog[98] for more information and guidance on transitioning to Data Control Tower's improved developer experience.

3.8.3 Release 17.0.0.0

3.8.3.1 End-of-life feature

Support for the following OS versions has reached EOL in this release:

- **ESXi 6.5 and 6.7**

3.8.4 Release 11.0.0.0

3.8.4.1 End-of-life feature

Support for the following OS version has reached EOL in this release:

- **ESXi 6.0**

3.8.5 Release 10.0.0.0

3.8.5.1 Deprecated features

Support for the following OS versions have been deprecated in this release:

- **ESXi 6.5 and 6.7**
  VMware's end of technical guidance for ESXi 6.5 and 6.7 is on 11/15/2023. To align our support, Delphix will end support of ESXi 6.5 and 6.7 in version 17.0.0.0 (November 2023 release).

98 https://community.delphix.com/blogs/nicholas-mathison/2024/02/08/delphix-end-of-support-for-delphix-self-service

3.8.6 Release 9.0.0.0

3.8.6.1 Deprecated features

- **Oracle** `move-to-asm.sh`
  The `move-to-asm.sh` scripted procedure has been deprecated in favor of the new feature **export an Oracle virtual DB or virtual PDB to a physical Oracle ASM or Exadata database**. For more details on this new feature, refer to the Export an Oracle VDB or vPDB to a Physical ASM or Exadata Database (see page 1628) section. The `move-to-asm.sh` scripted procedure is still supported in 9.0.0.0, but is discouraged by Delphix and will be eventually removed in future versions of the Delphix Continuous Data Engine.

3.8.7 Release 6.0.17.0

3.8.7.1 End-of-life features

Support for the following OS version has reached EOL in this release:

- **Oracle 11.1**
  Details of the Oracle database EOL can be found in the Oracle Lifetime Support Policy[99] PDF.

3.8.7.2 Deprecated features

Support for the following OS versions have been deprecated in this release:

- **ESXi 6.0**

3.8.8 Release 6.0.12.0

3.8.8.1 End-of-life features

- **Internet Explorer 11 support**
  Internet Explorer 11 is no longer supported by Delphix. The users are requested to refer to the list of supported browsers (see page 457).

Support for the following OS versions has reached EOL in this release:

- **EBS 12.1**
  Starting 6.0.12.0 and beyond, EBS 12.1 is no longer supported by Delphix. The users are requested to upgrade (see page 1686) to EBS 12.2 and start using this plugin.

---

99 https://www.oracle.com/us/assets/lifetime-support-technology-069183.pdf

- **SAP HANA 1.0**
  Starting 6.0.12.0 and beyond, SAP HANA 1.0 is no longer supported by Delphix. The users are requested to upgrade (see page 1959) to HANA 2.0 and start using this plugin.

3.8.9 Release 6.0.11.0

3.8.9.1 Deprecated features

Support for the following OS versions have been deprecated in this release:

- **Oracle 11.1**
  Details of the Oracle database EOL can be found in the Oracle Lifetime Support Policy[100] PDF.
- **TLS 1.0 and 1.1**
  These versions of TLS are known to be vulnerable, enterprise use is heavily discouraged.

3.8.9.2 End-of-life features

Support for the following OS versions has reached EOL in this release:

- **Oracle 10g**
- **TLS 1.0 and 1.1**
  These versions of TLS are known to be vulnerable, enterprise use is heavily discouraged.

3.8.10 Release 6.0.10.0

3.8.10.1 End-of-life features

- **EBS12.1**
  EBS 12.1 remains deprecated with a planned EOL (and removal from the product) in December 2021. This is in line with Oracle's planned EOL for EBS 12.1 later this year. The users are encouraged to upgrade[101] to Delphix's EBS 12.2.
- **Masking ruleset edit options**
  The Table Suffix, Add Column, Join Table, and List options were deprecated in the 6.0.3.0 release. These options have reached the EOL in the 6.0.10.0 release and have been completely removed from the product.
- **Internet Explorer 11 support**
  Internet Explorer 11 is planned for EOL in December 2021. This is in line with Microsoft's planned EOL for Internet Explorer 11 later this year.

100 https://www.oracle.com/us/assets/lifetime-support-technology-069183.pdf
101 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/5832988/EBS+plugin+installation

3.8.11 Release 6.0.9.0

3.8.11.1 End-of-life features

- Delphix Reporting (Formerly, *Mission Control*) is EOL starting 6.0.9.0 (July 1st, 2021). The users interested in continuing their reporting workflows are recommended to discuss Data Control Tower with their account teams.
- Also, EBS 12.1 remains deprecated with a planned EOL (and removal from the product) in December 2021. This is in line with Oracle's planned EOL for EBS 12.1 later this year. The users are encouraged to upgrade[102] to Delphix's EBS 12.2.

3.8.12 Release 6.0.8.0

3.8.12.1 End-of-life features

Support for the following OS versions has reached EOL in this release:

- SAP ASE 15.0.3
- Windows Server 2008 R2
- SQL Server 2008
- SQL Server 2008 R2

3.8.13 Release 6.0.7.0

3.8.13.1 End-of-life features

- ESX 5.5 support will be end-of-life in 6.0.7
- Masking Connectors: Db2 LUW and zOS v9, Db2 LUW and zOS v10, SQL Server 2005, 2008, 2008 R2

3.8.14 Release 6.0.5.0

The following list of OS and DB versions are now past their respective support windows. Delphix will deprecate these versions to stay in line with Microsoft, Oracle, and SAP lifecycle policies:

- Windows 2008 R2: Microsoft End of Extended Support (EoES) January 4th, 2020
- SQL Server 2008/2008 R2: Microsoft End of Extended Support (EoES) July 9th, 2019
- Oracle 10: Oracle End of Extended Support (EoES) July 2013
- SAP ASE 15.0.3: SAP End-of-Life (EoL) March 31st, 2015

102 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/5832988/EBS+plugin+installation

3.8.15 Release 6.0.4.0

3.8.15.1 End-of-life features

- **Masking**:
  - Job Scheduler - As of this release, Delphix has removed the Job Scheduler feature. The introduction of Masking's REST API several releases ago allowed customers to schedule job executions using their preferred job scheduler. As a result, the integrated scheduler is seldom used.

3.8.15.2 Deprecated features

- **Masking**:
  - FTP, SFTP, and mount upload for XML and Cobol formats - FTP/SFTP/Mount-based format import were the original modes for XML and Cobol files, since then, Delphix has added the ability to upload a format file, which is far simpler to set up. After the introduction of "upload", we've seen a dramatic shift away from the legacy import modes in favor of the simplicity of "upload".
  - Row Type Feature - Originally geared for limiting masking to subsets of rows within a column, this feature was seldomly used. Its functionality, if desired, can still be replicated via the Custom SQL feature.
  - Redundant Settings for 'Edit Table' under Rule Sets - Table Suffix, Add Column, Join Table, and List - These settings are redundant and can be replicated with the Custom SQL setting.
  - 'HAVING' clause from Masking API - Deprecating due to low use. This feature, if desired, can be replicated with Custom SQL.

3.8.16 Release 6.0.2.0

3.8.16.1 End-of-life features

- AWS i.3 (EOL) January 22, 2020
- Microsoft Azure GSx (EOL) May 7, 2020

3.8.17 Release 6.0.0.0

3.8.17.1 End-of-life features

- **Removed Virtualization Features**:
  A few features will be removed with the 6.0 release. If these features are in use, the upgrade will not proceed.
  - Cross Platform Provisioning (XPP)
  - AIX 6.1 Technology Levels 0-6 and AIX 7.1 Technology Levels 0-2 for Environments
  - Windows Server 2008 for Environments

> Windows Server 2008 will not have an upgrade blocker.

- **Removed Masking Features**:
  Please note that Excel files can still be masked by first converting them to one of Delphix's supported file types (CSV, etc). Also, XML CLOBs can be masked by extracting their values into a table (example - using extractValue in Oracle).
  - Native XML CLOB masking: After upgrade, columns masked as XML CLOBs will have the NULL SL algorithm assigned.
  - DB2 9.1, 9.5, and other 9.x versions of LUW & Z/OS
  - "Create target" job option: After upgrade jobs using "create target" will be removed.
  - "Bulk data" job option: After upgrade, jobs using "bulk data" will be turned into non-bulk data jobs.
  - Native Microsoft Excel Masking: After upgrade, MS Excel connectors, rulesets and jobs will be removed.

3.9 Licenses and notices

The Delphix Dynamic Data Platform includes licensed, third-party products from the following companies. These products are copyrighted and all rights are reserved by the respective companies:

- Highcharts, © Highsoft

The Delphix Masking engine includes licensed, third-party products from the following companies. These products are copyrighted and all rights are reserved by the respective companies:

- Kendo UI, © Telerik

Starting with 6.0.3.0, the license info is available via a CLI/API on the engine when logged in as a system administrator.

```
engine> cd license
engine license> getLicense
engine license getLicense *> commit
```

Access to the source code of such third party open source components may be permitted or required in certain instances under the applicable open source licenses by sending an email to mailto:open-source@delphix.com

3.10 Data source integration (plugin) release notes

This section covers the following topics:

- CockroachDB release notes (see page 464)
- Couchbase release notes (see page 466)
- IBM Db2 release notes (see page 470)
- MongoDB release notes (see page 486)
- MySQL release notes (see page 491)
- Oracle E-Business Suite (EBS) 12.2 release notes (see page 495)
- PostgreSQL release notes (see page 507)
- SAP HANA release notes (see page 521)
- Yugabyte release notes (see page 537)

3.10.1 CockroachDB release notes

New features (CockroachDB) (see page 464)

3.10.1.1 New features (CockroachDB)

3.10.1.1.1 Release 1.2.0

This release supports the following feature/features:

- **Skip Localities Check (optionally)**
  With this plugin release, we have introduced a checkbox in the dSource configuration option named "Skip Localities Check" allowing users to bypass the locality check of a cluster before initiating a restore operation.

3.10.1.1.2 Release 1.1.2

- CockroachDB: PoC for auth-implicit feature to restore from AWS S3.

> If you are using implicit authentication feature, ensure that AWS credentials are configured on staging host using `aws configure` command.

3.10.1.1.3 Release 1.1.1

- Record ZFS Storage Calculation Command (CockroachDB).
- CockroachDB Plugin Security Fixes and Hardening.

3.10.1.1.4 Release 1.1.0

- Cockroach (CRDB) Select Connector Virtualization Plugin.
- Enabled the NFS restore functionality.

3.10.1.1.5 Release 1.0.0

- First Version of CockroachDB Plugin.
- CE-27[103] Cockroach (CRDB) Select Connector Virtualization Plugin
- Code Structure referenced from Select Connector vSDK framework.
- Python version 3 compatibility.
- Improved Code Diagnosability / Readability.

3.10.1.2 Fixed issues (CockroachDB)

3.10.1.2.1 Release 1.2.0

Bug number	Description
CE-689	ERROR: cannot restore view "mvw_schedules" without restoring referenced table (or "skip_missing_views" option)
CE-690	ERROR: detected a mismatch in regions between the restore cluster and the backup cluster, missing regions detected: us-east-1, us-east-2, us-west-2

103 https://delphix.atlassian.net/browse/CE-27

Bug number	Description
CE-691	CRDB Plugin: dsource creation is failing with error: The network session with "[]" was interrupted
CE-719	Support appdata parameters redaction for engine metadata support bundle

3.10.1.3 Known Issues (CockroachDB)

3.10.1.3.1 Release 1.2.0

There are no known issues in this release.

3.10.2 Couchbase release notes

This section covers the following topic:

- New features (Couchbase) (see page 466)
- Fixed issues (Couchbase) (see page 467)

To view the Couchbase support matrix, visit the Couchbase[104] page.

3.10.2.1 New features (Couchbase)

3.10.2.1.1 1.3.1

This connector release consists of only one bug fix.

3.10.2.1.2 1.3.0

- Bucket size control has been implemented.
- Ingestion from multiple full backups is now supported.
- Bucket creation can be retried if failed.
- Record ZFS Storage Calculation command has been implemented.
- Corrected a Python error when raising an exception with traceback.

3.10.2.1.3 1.2.2

- Couchbase version 7.1.3 support.

104 https://cd.delphix.com/docs/latest/couchbase-support-matrices

- REST API implementation for the following:
  - cluster-init
  - xdcr-setup
  - setting-cluster

3.10.2.1.4 1.2.1

- Security fixes and improvements
- An additional requirement to upgrade is the installation of the `expect` utility on staging and target hosts.

3.10.2.2 Fixed issues (Couchbase)

3.10.2.2.1 Release 1.3.1

Key	Summary
CE-633	Couchbase Plugin: Dsource sizing not working due to filesystem permissions

3.10.2.2.2 Release 1.3.0

Key	Summary
CE-588	Add DE support matrix in github docs
CE-582	Github documentation update
CE-575	Fix Python error when raising an Exception with traceback.
CE-574	Upgrade Logic
CE-572	Retry bucket creation if failed
CE-554	Implement precommit checks in couchbase repo
CE-541	Ingestion from multiple full backups

Key	Summary
CE-540	bucket size control implementation

3.10.2.2.3 Release 1.2.2

Key	Summary
CE-550	Changes in cluster-init and server-add for couchbase 7.1.1
CE-549	Sanity testing with Couchbase 7.x Community Edition (7.1.1)
CE-548	Sanity testing with Couchbase version 6.6.1
CE-547	CodeQL fixes
CE-546	plugin version upgrade
CE-545	code hardening
CE-544	fix XDCR delete configuration
CE-543	scope and collection pre-creation for xdcr dsource type
CE-542	VDB creation fix for couchbase version 7.1.3
CE-525	server-add command fix
CE-523	bucket-flush command fix
CE-521	cluster init fails when couchbase instance is recreated on a machine
CE-520	status check of replication fails in xdcr dsource type
CE-519	snapshot fails with backup restore failure
CE-518	index creation inside scope support

Key	Summary
CE-511	REST API implementation for setting-cluster
CE-510	REST API implementation for xdcr-setup
CE-509	REST API implementation for cluster-init
CE-508	github documentation support matrix update
CE-507	Code changes to increase buffer size in bucket-list command
CE-506	Code changes to handle prompt changes in cluster-init
CE-477	Couchbase Version 7.1.3 Support

3.10.2.2.4 Release 1.2.1

Key	Summary
CE-341	Fix codeQL
CE-340	Code changes to mask password in "ps" using expect utility
CE-339	Addition of security tests in codeQL
CE-338	Master updation with latest code base as of 25/05/2023
CE-337	Creation and updation of 'develop' branch with latest codebase
CE-332	Disable python2.7 unit tests on couchbase repository
CE-331	Security vulnerability assessment

3.10.3 IBM Db2 release notes

This section covers the following topics:

- New features (IBM Db2) (see page 470)
- Fixed issues (IBM Db2) (see page 477)
- Known issues (IBM Db2) (see page 482)

To view the IBM Db2 support matrix, see IBM Db2 matrix[105].

3.10.3.1 New features (IBM Db2)

3.10.3.1.1 Release 4.7.2

- **Skip data path verification (Optionally)**
  You can now skip the data path verification process during VDB Snapsync if required. The `skipDataPathVerification` value is mandatory, if you are upgrading datasets from connector versions <=3.1.1 and do not wish to perform VDB re-provision or refresh.

3.10.3.1.2 Release 4.7.1

This plugin release consists of only bug fixes.

3.10.3.1.3 Release 4.7.0

- **SSL/TLS support for HADR ingestion**
  The Db2 plugin version 4.7.0 introduces SSL/TLS support for HADR ingestion, enhancing data security during replication by encrypting data in transit between primary and standby databases.
- **Dynamic UI for Db2 virtualization**
  With the introduction of various dynamic UI features, you can now experience a streamlined and user-friendly interface that adapts based on your selected choices, simplifying the linking process.

3.10.3.1.4 Release 4.6.1

This plugin release consists of only bug fixes.

3.10.3.1.5 Release 4.6.0

This plugin release has the following set of new features and bug fixes.

105 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/114534163

- **Staging push support for DPF databases**
  The Db2 Plugin 4.6.0 version introduces Staging push support for DPF databases. This allows you to ingest databases using a backup agent solution of your choice.
- **VDB Provision with user-defined mount base**
  This plugin version introduces support for user-defined mount base for VDBs, which allows you to provide a mount base of your choice during the provisioning of a VDB. **Note:** DIR_CACHE for the instance needs to be set to NO when users want to update the VDB dataset with the new mount base.
- **Optionally skip archive log validation**
  With this feature, you can skip the archive log validation process during Snapsync if you choose to do so.
- **Load copy files support for Staging push**
  The load copy files are now supported by the DB2 staging push automation script. This allows you to provide the load copy files along with the archive logs to successfully roll forward the database.

3.10.3.1.6 Release 4.5.1

This plugin release consists of only bug fixes.

3.10.3.1.7 Release 4.5.0

This plugin release has the following set of new features and bug fixes.

- **Support for multiple backup paths for dSource ingestion**
  The Db2 Plugin 4.5.0 version introduces the support for multiple backup paths instead of one for dSource ingestion. This allows you to provide single backup path for the dSource ingestion and add more backup paths with a dynamic UI to accommodate for the extra backup directories where the backup was split into.

3.10.3.1.8 Release 4.4.2

This plugin release consists of only bug fixes.

3.10.3.1.9 Release 4.4.1

This plugin release consists of only bug fixes.

3.10.3.1.10 Release 4.4.0

This plugin release has the following set of new features and bug fixes.

- **DB2 pull performance optimization**
  The Db2 Plugin 4.4.0 version introduces the Pull Performance Optimization feature, which attempts to optimize the dSource workflows by allowing users to provide the degree of parallelisation to run the database restore operation. This feature allows users to ingest large databases in a faster way,

provided that there are sufficient resources for Db2 plugin to implement this parallelisation in the operation.

- **Source sizing implementation on the DSIL DB2 plugin**
  The source sizing feature aligns the data source experience with Delphix database standards that will improve your ingestion reporting experience on the per TB pricing model. This feature enables you to calculate the dataset size based on the total allocated space on the mount point provided by the Delphix Engine for the dataset. For more details about this feature, see Source Sizing Implementation for Datasets[106].

3.10.3.1.11 Release 4.3.2

This plugin release consists of only bug fixes.

3.10.3.1.12 Release 4.3.1

This plugin release consists of only bug fixes.

3.10.3.1.13 Release 4.3.0

The DB2 plugin 4.3.0 introduces Staging push support for HADR dSources. You can now create and maintain dSources configured with HADR using the Staging push option.

3.10.3.1.14 Release 4.2.1

This plugin release consists of only bug fixes.

3.10.3.1.15 Release 4.2.0

This plugin release has the following set of new features and bug fixes.

- **Dsource snapshot with user-defined timestamp**
  The Db2 Plugin version 4.2.0 introduces the dsource snapshot with the user-defined timestamp feature. This feature enables the user to specify a timestamp during dSource snapshot so that staging will roll forward to the requested timestamp before performing the dSource snapshot. For more information, see Linking a Db2 dSource[107].

3.10.3.1.16 Release 4.1.3

This plugin release consists of only bug fixes.

106 https://cd.delphix.com/docs/latest/source-sizing-implementation-for-ibm-db2-dsource
107 https://cd.delphix.com/docs/latest/linking-ibm-db2-data-sources

3.10.3.1.17 Release 4.1.2

This plugin release consists of only bug fixes.

3.10.3.1.18 Release 4.1.1

This plugin release consists of only bug fixes.

3.10.3.1.19 Release 4.1.0

This plugin release has the following set of new features and bug fixes.

- **User-specified mount path For DB2 dSource**
  Users can specify the mount path of their choice on the target host to host the dSource dataset.
- **Staging push automation**
  A set of scripts is being provided that can be used to automate the steps related to the restore and rollforward operations on the dSource.

3.10.3.1.20 Release 4.0.1

The Db2 Plugin version 4.0.1 includes the below improvements.

- Added support for 11.1.4.6 fix pack.

3.10.3.1.21 Release 4.0.0

The Db2 Plugin version 4.0.0 includes the below improvements.

- Plugin generated log file will now be placed at `<toolkit path>/<Delphix_COMMON>/DB2_18f4ff11-b758-4bf2-9a37-719a22f5a4b8/logs/<instance name>`.
- More user-friendly log statements will be printed.
- The logging.properties and advanceConf.config configuration files are now merged into the db2_plugin.conf file.
- Dependency on redirect restore script to get the data path information is now removed.

3.10.3.1.22 Release 3.1.1

The Db2 Plugin version 3.1.1 introduces the support for full online backups, which are taken using the "exclude logs" syntax. This support is only applicable to the dSources created using the Backup and Logs ingestion mechanism on a single partition.

In Db2, users have the flexibility to exclude logs while taking the full online backups. The Db2 plugin will read the backup header from the user-provided backup file. The backup header is used to figure out whether the backup was taken using the "exclude logs" syntax or not. In case the plugin detects that it was taken using the "exclude logs" option, then the required changes will be made in the restore syntax.

3.10.3.1.23 Release 3.1.0

The Db2 Plugin version 3.1.0 introduces the **Staging Push** feature. This feature will allow users to control the restore of the staging database and they can also control the state of the restored staging database. With the **Staging Push** feature, the restore and rollforward operations will be managed by the user from the CLI. During dSource creation (or RESYNC) the plugin will create an ***empty mount point*** and will provide a restore and rollforward template that will help the user to create the redirect restore script. Once the dSource will be ready then the user can perform restore and rollforward operations from CLI. The restore must be as per the plugin provided template. After the restore, the staging database should remain in rollforward pending state, and prior to the next snapshot, the user can do a rollforward to a specific Point-in-Time (by providing a timestamp to rollforward command) against the available archive logs.

3.10.3.1.24 Release 3.0.2

This release addresses a limitation in Db2 wherein the VDB can accept the connections when refresh, disable and delete Delphix operations are in process which eventually can corrupt the VDBs. This plugin version will reduce such occurrences by introducing maintenance window in these operations during which the VDB will not accept external connections.

3.10.3.1.25 Release 3.0.1

The Db2 Plugin version 3.0.1 fixes the bug wherein the jobs on UI hung intermittently due to a known IBM Db2 bug.

3.10.3.1.26 Release 3.0.0

The Db2 Plugin version 3.0.0 introduces support to Db2 Database Partitioning Feature (DPF). This will help the customer to ingest the data to a staging dsource from a Db2 DPF enabled source.

- **Database partitioning feature**
  Database Partitioning Feature (DPF) lets you partition your database across multiple servers. Since you can add new machines and spread your database across them, this allows users to scale their database. This means more CPUs, more memory, and more disks from each of the additional machines are available for your database. DPF can be used to manage large databases for a variety of use cases including data warehousing, data mining, online analytical processing (OLAP), or online transaction processing (OLTP) workloads.
  DPF enables the user to divide a database into database partitions, a database partition is a part of a database that consists of its own data, indexes, configuration files, and transaction logs. Each database partition can be configured on the different physical server having its own set of computing resources, including CPU and storage. When a query is processed, the request is divided so each database partition processes the rows that it is responsible for. DPF can maintain consistent query performance as the table grows by providing the capability to add more processing power in the form of additional database partitions. This capability is often referred to as providing linear scalability using Db2s shared-nothing architecture.

DPF is an approach to sizing and configuring an entire database system. Please follow the recommended practices for optimal performance, reliability, and capacity growth. Please refer to IBM documentation of DPF for more details in IBM knowledge center[108].

- **Pipelining logic for implementing parallel restores**
  The plugin employs a parallel pipeline methodology so that the restore operation of non-catalog partitions can be performed in parallel in the Database Partitioning Feature (DPF). The number of parallel restores is determined by the value of “restorePipelineLimit” (default value is 10) in <Toolkit directory>/advanceConfFlag.conf. The parameter “restorePipelineLimit” is configurable by end-users. The plugin performs parallel restore for all non-catalog partitions. For e.g. If the total number of non-catalog partitions is 15, and the "restorePipelineLimit" parameter is set to 10, the first set of 10 restores will happen in parallel. The plugin will track the restore of each partition present in the pipe. Whenever a restore of a partition completes, it will move out from the pipe and a new partition will enter into that pipe. Thus, the plugin ensures that the pipe will always have the user-configured number of partitions being restored (default=10).
- **Data synchronization**
  When “Customer Supplied Archive Logs” feature is used along with Db2 DPF feature, users will have to place the archive logs inside a folder with a name as NODE<Partition number> where <Partition number> is a four-digit number. For example, if the source environment has two partitions then the user-provided log path will have folder names NODE0000 and NODE0001. Both the folders will have respective archive logs. Snapshot operation will be used to apply the archive logs.
- **Advance configuration file**
  The advanced config file is created at <Toolkit directory>/advanceConfFlag.conf during discovery or environment refresh operation if the file does not pre-exist already. This file will have the following configurable parameters:
  - **Common group (*notCommonGroupFlag)***
    - This parameter sets the permission of "mnts", "logs" and "code" directories at <Toolkit dir>/DB2 location.
    - Set *notCommonGroupFlag to false,* if the primary group of the primary environment user (user which is used to do discovery) is shared with the instance users.
    - Set *notCommonGroupFlag to true,* if you don't want to share any group between the primary environment user and the instance users.
    - Once the necessary changes are made, refresh the environment in order to make the changes applicable.

  By default, the *notCommonGroupFlag* parameter is not specified in the file. This means that the plugin implicitly assumes its value as false. The valid values for this parameter are true or false.

  - **Allow source database on same instance (*allowSourceDBonSameInst*)**
    By default, the DB2 Plugin restricts the provisioning of a VDB on a DB2 instance which contains a database with the same name as the VDB's source database
    For example:
    - When provisioning a VDB from a VDB then source VDB is treated as the source database.

108 https://www.ibm.com/developerworks/data/library/techarticle/dm-0608mcinerney/index.html

- When provisioning a VDB from a dSource then a staging database is treated as the source database.

Set *allowSourceDBonSameInst* to true, if the user wants to provision a VDB on an instance which contains a database with the same name as the VDB's source database. The valid values for this parameter are true or false.

- **Restore pipeline limit (*restorePipelineLimit*)**
  Db2 DPF Plugin performs restoration of all non-catalog partitions in parallel.
  The parameter *restorePipelineLimit* allows the user to configure the number of partitions that should be restored in parallel. When the new advanceConfFlag.conf file is created during environment refresh or discovery operation, the default value of *restorePipelineLimit* is set to 10. If the advanceConfFlag.conf file is already present, the user will have to set the value of the `restorePipelineLimit` parameter to the desired value. The valid value for this field is any positive integer.

A default config file created during discovery or environment refresh operation: “

```
# If the user is not sharing a common group between primary environment user
and instance user.

# Then the user needs to set parameter notCommonGroupFlag as true.

# notCommonGroupFlag=true

# During provision operation check if instance contains a database name which
is identical to the source DB name used for provisioning operation.

# By default plugin will never allow creating a VDB on the instance where
source database (or other database which is identical in name with source DB)
already exists.

# If the user still wants to create a VDB on the instance which contains a
database name which is identical to the source DB name used for provisioning
operation,

# then the user needs to set parameter allowSourceDBonSameInst as true.

# allowSourceDBonSameInst=true

# Limit for parallel restores

restorePipelineLimit=10
```

3.10.3.2 Fixed issues (IBM Db2)

3.10.3.2.1 Release 4.7.2

Key	Summary
DB2-1606	Custom mount base fails with 'attribute error' during VDB enable.
DB2-2349	Update Db Storage paths if old path structure, <mount_path>/<db_path> is found

3.10.3.2.2 Release 4.7.1

Key	Summary
DB2-2305	Custom mount base fails with 'attribute error' during VDB enable.

3.10.3.2.3 Release 4.7.0

There are no fixed issues in this release.

3.10.3.2.4 Release 4.6.1

Key	Summary
DB2-2190	Instead of failing the process, the housekeeping job should issue a warning.

3.10.3.2.5 Release 4.6.0

Key	Summary
DB2-2068	"VDB Name" requested to be entered twice during provisioning.
DB2-2117	Update LOGARCHCOMPR1 and LOARCHCOMPR2 to off during dSource creation

Key	Summary
DB2-2155	db2greg should run from the latest installation of db2 on staging/target for environment discovery
DB2-2156	All DB2 dSources with Backup+logs should only rollforward to PIT
DB2-2178	MRT query fails with SQL1262N on staging hosts that are ahead of UTC timezone
DB2-2179	If DB2LOADREC registry value already set raise error.

3.10.3.2.6 Release 4.5.1

Key	Summary
DB2-2112	DB2 pre-snapshot phase runs rollforward to EOL on RF pit configuration

3.10.3.2.7 Release 4.5.0

There are no fixed issues in this release.

3.10.3.2.8 Release 4.4.2

Ticket number	Summary
DB2-2004	Some DB2 VDB deletes/refresh when canceled leave database running on instance in strange state

3.10.3.2.9 Release 4.4.1

Ticket number	Summary
DB2-2044	Db2 plugin shouldn’t implement Source sizing for VDB datasets
DB2-2045	VDB provisioning fails with SQl5153N due to LOGSECOND value

Ticket number	Summary
DB2-2053	Provisioning VDB failed during db2relocatedb due to multiple storage group paths

3.10.3.2.10 Release 4.4.0

Ticket number	Summary
DB2-2033	Resync fails on staging hosts with fuser command
DB2-2041	HADR linking is failing for low privilege user environment

3.10.3.2.11 Release 4.3.2

Ticket number	Summary
DB2-1682	Db2 vSDK plugin should support all product editions
DB2-1971	Environment addition and other workflows fail when .profile is configured with banners
DB2-1992	Restore on dpf environment continues even when one or more nodes' restore fails in dSource creation
DB2-1997	Db2 staging push doesn't test for missing tablespaces/datapaths while running db2dart

3.10.3.2.12 Release 4.3.1

Ticket number	Summary
DB2-1973	Make sure the data paths are configured under "db_mount/DB_NAME" in staging push

3.10.3.2.13 Release 4.3.0

Ticket number	Summary
DB2-1952	`remove_event_bin_files` may fail due to regular expression mismatch for DIAGPATH value

3.10.3.2.14 Release 4.2.1

There are no fixed issues in this release.

3.10.3.2.15 Release 4.2.0

Ticket number	Summary
DB2-1901	VDB provision from dsource with rollforward to PIT is running `rollforward to end of logs and stop` instead of `rollforward stop`
DB2-1911	DB2 rollforward command failing with syntax error for `dpf` environments
DB2-1913	The rollforward command should work with timezone only when timestamp is provided.
DB2-1920	VDB provision fails with `Failed to execute rollforward command on database BDN9NAJZ` in case of dsource with backup and log ingestion.

3.10.3.2.16 Release 4.1.3

Ticket number	Summary
DB2-1812	CLI operations during snapsync may fail if commands return error messages in a different locale
DB2-1856	VDB provision fails with an issue during db2relocatedb

Ticket number	Summary
DB2-1875	VDB behavior when we provision them using the first snapshot vs the latter ones
DB2-1921	Fix permissions for “/toolkit_dir/DB2” along with mnts/ and logs
DB2-1904	Staging push vSDK: snapsync operation on dSource fails with TypeError: unhashable type: ‘SnapshotDefinitionDataPaths’

3.10.3.2.17 Release 4.1.2

Ticket number	Summary
DB2-1826	Moved Staging Push redirect restore template out of the plugin and added contextual values to plugin logs
DB2-1796	Unable to get the complete content of redirect restore script in case the script size is more than 1MB
DB2-1824	DB2 VDB provisioning fails during relocateDB

3.10.3.2.18 Release 4.1.1

There are no fixed issues in this release.

3.10.3.2.19 Release 4.1.0

There are no fixed issues in this release.

3.10.3.2.20 Release 4.0.1

Key	Summary
DB2-1684	[Staging Push]Enable operation on dSource should generate the Restore and Rollforward template file if it is missing

3.10.3.2.21 Release 4.0.0

There are no fixed issues in this release.

3.10.3.2.22 Release 3.1.1

There are no fixed issues in this release.

3.10.3.2.23 Release 3.1.0

There are no fixed issues in this release.

3.10.3.2.24 Release 3.0.2

There are no fixed issues in this release.

3.10.3.2.25 Release 3.0.1

There are no fixed issues in this release.

3.10.3.2.26 Release 3.0.0

There are no fixed issues in this release.

3.10.3.3 Known issues (IBM Db2)

3.10.3.3.1 Release 4.7.2

Key	Summary	Workaround
DB2-2231	During upgrade from 6.0.x to 7.0, 8.0, ….21.0 , Delphix engine destroys the plugin's scratch path where all the plugin logs, configuration files are present. During the environment refresh, these paths and files are regenerated. However, any edits done in db2_plugin.conf will not be persisted	Before upgrading the engine, the user will have to make a copy of the db2_plugin.conf file to keep the changes intact.

3.10.3.3.2 Release 4.7.1

Key	Summary	Workaround
DB2-2056	Upgrade from lua version to VSDK has issues with VDB Snapsync	Re-provisioning the database with the dSource's snapshot fixes the issue.
DB2-2124	If the storage paths are subdirectories of another storage path, the instance name at target and true prod cannot be the same.	Create instance name on target different than true production

3.10.3.3.3 Release 4.7.0

Key	Summary	Workaround
DB2-2056	Upgrade from lua version to VSDK has issues with VDB Snapsync	Re-provisioning the database with the dSource's snapshot fixes the issue.
DB2-2124	If the storage paths are subdirectories of another storage path, the instance name at target and true prod cannot be the same.	Create instance name on target different than true production

3.10.3.3.4 Release 4.6.1

Key	Summary	Workaround
DB2-2056	Upgrade from lua version to VSDK has issues with VDB Snapsync	Re-provisioning the database with the dSource's snapshot fixes the issue.
DB2-2124	If the storage paths are subdirectories of another storage path, the instance name at target and true prod cannot be the same.	Create instance name on target different than true production

3.10.3.3.5 Release 4.6.0

Key	Summary	Workaround
DB2-2056	Upgrade from lua version to VSDK has issues with VDB Snapsync	Re-provisioning the database with the dSource's snapshot fixes the issue.
DB2-2124	If the storage paths are subdirectories of another storage path, the instance name at target and true prod cannot be the same.	Create instance name on target different than true production

3.10.3.3.6 Release 4.5.1

Key	Summary	Workaround
DB2-2107	db2ckbkp fails for Backup paths with spaces	Remove blank spaces from backup paths
DB2-2124	The instance name at target and true prod cannot be the same if the storage paths are sub directories of another storage path.	Create instance name on target different than true production

3.10.3.3.7 Release 4.5.0

Key	Summary	Workaround
DB2-2107	db2ckbkp fails for Backup paths with spaces	Remove blank spaces from backup paths

3.10.3.3.8 Release 4.4.2

There are no known issues in this release.

3.10.3.3.9 Release 4.4.1

There are no known issues in this release.

3.10.3.3.10 Release 4.4.0

There are no known issues in this release.

3.10.3.3.11 Release 4.3.2

There are no known issues in this release.

3.10.3.3.12 Release 4.3.1

There are no known issues in this release.

3.10.3.3.13 Release 4.3.0

There are no known issues in this release.

3.10.3.3.14 Release 4.2.1

There are no known issues in this release.

3.10.3.3.15 Release 4.2.0

There are no known issues in this release.

3.10.3.3.16 Release 4.1.3

There are no known issues in this release.

3.10.3.3.17 Release 4.1.2

There are no known issues in this release.

3.10.3.3.18 Release 4.1.1

There are no known issues in this release.

3.10.3.3.19 Release 4.1.0

There are no known issues in this release.

3.10.3.3.20 Release 4.0.1

There are no known issues in this release.

3.10.3.3.21 Release 4.0.0

There are no known issues in this release.

3.10.3.3.22 Release 3.1.1

There are no known issues in this release.

3.10.3.3.23 Release 3.1.0

There are no known issues in this release.

3.10.3.3.24 Release 3.0.2

There are no known issues in this release.

3.10.3.3.25 Release 3.0.1

There are no known issues in this release.

3.10.3.3.26 Release 3.0.0

Key	Summary	Workaround / Comments
DB2-1318	Jobs on UI might hang intermittently due to a known IBM Db2 bug	This issue will be only observed when multiple datasets jobs, within the same Db2 instance, will get executed at the same time.

3.10.4 MongoDB release notes

This section covers the following topics:

- New features (MongoDB) (see page 487)
- Fixed issues (MongoDB) (see page 488)
- Known issues (MongoDB) (see page 491)

To view the MongoDB support matrix, visit the MongoDB support matrix[109] page.

109 https://cd.delphix.com/docs/latest/mongodb-support-matrix

3.10.4.1 New features (MongoDB)

3.10.4.1.1 1.4.0

- Supported `requireTLS` tlsMode for the `extendedcluster` dsource type.
- Improved debugging capabilities by un-redacting any generic information in engine metadata.

3.10.4.1.2 1.3.0

- Conducted comprehensive UI refactoring to improve user experience, streamline navigation, and enhance overall interface efficiency.
- Improved MongoDB operations by introducing support for parallel execution of `mongodump` and `mongorestore`, significantly speeding up backup and restoration processes.

3.10.4.1.3 1.2.1

- Implemented security enhancements and hardening measures for the MongoDB plugin, ensuring robust protection against vulnerabilities and unauthorized access.
- Updated the documentation to include a support matrix for Atlas MongoDB dSource types, providing clear guidance on compatibility and configuration options.

3.10.4.1.4 1.2.0

- Introduced Source Sizing for the MongoDB plugin, enhancing the capacity planning and resource allocation process for more efficient database management.
- Addressed security concerns by concealing the MongoDB instance password in the staging host's `ps` output, preventing unauthorized access and improving system security.
- The MongoDB plugin now automatically captures and reports errors directly from MongoDB logs, to help streamline troubleshooting.
- MongoDB 6 and Cluster-to-Cluster Sync is now supported.
- MongoDB 6 on RHEL 8 is now supported.

3.10.4.1.5 1.1.0

- A refined version of the CRDB logger has been implemented to enhance data security, which includes generic masking of sensitive information (such as database passwords) preventing them from being displayed in plugin logs in clear text.
- SDD support has been added.

3.10.4.1.6 1.0.2

- Users can now specify the location of `mongodump` and `mongorestore`.
- Added support to accept a `.tar` extension of a MongoDB Ops Manager backup file in addition to `.tar.gz`.
- MongoDB 5.0.9 is now supported.

3.10.4.1.7 1.0.0

- First release version of the MongoDB connector

3.10.4.2 Fixed issues (MongoDB)

3.10.4.2.1 1.4.0

Bug number	Resolution
CE-702	Unredact schema.json parameters except sensitive information.
CE-703	Port requireTLS mode support for extendedcluster from HF 1.3.2.1

3.10.4.2.2 1.3.2.1

Bug number	Resolution
CE-685	Support tlsMode requireTLS for a staging database created using extendedcluster dsource type

3.10.4.2.3 1.3.2

Bug number	Resolution
CE-62	Fixed MongoDB Plugin: Onlinedump dSource Pre-Snapshot operation failure issue.

Bug number	Resolution
CE-449	Fixed the issue of linking MongoDB dSource dialogue allowing a Windows path in the Mount Path that results in fatal internal error.
CE-450	Fixed unable to read UI dialogue instructions for MongoDB Config Server Backup File.
CE-456	Fixed issue to display the mongorestore output in verbose mode
CE-457	Fixed the issue of MongoDB creating empty snapshots when there are no backups to restore.
CE-458	Fixed an intermittent issue of Offlinemongodb dSource runs SnapSync on policy schedule and creates snapshot despite no new backups.
CE-502	Fixed the issue of linking a MongoDB dSource failure due to user already exists.
CE-551	Fixed MongoDB VDB refresh sporadically failure due to Address already in use.
CE-559	Fixed the MongoDB dSource dialogue linking confusion.
CE-611	Validate if mongosh client exists in the PATH over the deprecated "mongo" client.
CE-638	Fixed the issue of VirtualSource object has no attribute mongo_shell_path.
CE-639	Previously shutdown mongod process fails to start throwing SocketException Address already in use.
CE-643	Fixed the mongodump and mongorestore defects at SSL/TLS enablement for address already in use in provisioning.
CE-644	MongoServerError: not primary and secondaryOk=false - consider using db.getMongo().setReadPref() or readPreference in the connection string
CE-645	VirtualSourceDefinition object has no attribute d_source_type

3.10.4.2.4 1.2.0

Bug number	Resolution
CE-142	Fixed an issue where dSource enable failed with “IllegalObjectAccess” after replication failover.
CE-189	Fixed an issue when a MongoDB instance is terminated with `kill` that would sometimes fail and leave child processes orphaned.
CE-240	Fixed an issue where `sh.addShard()` fails if the staging database password contains special characters.
CE-256	`sudo` rule examples in MongoDB staging and target requirements have been corrected.
CE-79	A description of the expected format for backup files has been added.
CE-186	Fixed an intermittent MongoDB issue where it errors out with “Address already in use” if temporary socket files for available ports exist on the staging environment.

3.10.4.2.5 1.1.0

Bug number	Resolution
DXE-986	Removed an additional “Source sharded” prompt if the type of data source (dSource_Type) has already been selected as `shardedsource`.
DXE-987	Removed the "Convert each shard to replica set" option from the dSource linking process, as it applies only to VDBs.

3.10.4.2.6 1.0.2

Bug number	Resolution
DXE-985	Fixed an issue where the incorrect date format in backup metadata files produce a linking error.
DXE-964	Fixed an issue where enabling authentication with `User Auth Mode` as `ldap` breaks dSource/VDB creation.

3.10.4.2.7 1.0.1

- Sharded dSource creation workflow has been fixed.

3.10.4.3 Known issues (MongoDB)

The following known issues take effect in all MongoDB plugin versions.

- The MongoDB plugin does not recognize any environment variables in hooks.
- dSources cannot be upgraded via GUI. Only the CLI interface is supported.

3.10.5 MySQL release notes

This section covers the following topics:

- New features (MySQL) (see page 491)
- Fixed issues (MySQL) (see page 493)

To view the MySQL support matrix, see MySQL matrix (see page 1282).

3.10.5.1 New features (MySQL)

3.10.5.1.1 Release 4.5.0

This plugin release supports the following feature/features:

- **Dynamic UI:** This release introduces dynamic UI changes that will make only those fields visible that are required for the selected ingestion mechanism (dSource Type) staging push or replication. As part of this feature, the following changes have been made:
    - The **Database List** and **Logsync** fields are now removed from the linking screen.

- **Provide different Replication User** is added on the linking screen. This field is used in scenarios where you explicitly want to provide different replication user credentials.

3.10.5.1.2 Release 4.4.0

This plugin release consists of only bug fixes.

3.10.5.1.3 Release 4.3.0

This release renames the 'Manual Backup Ingestion' mechanism to 'Staging Push'. You can notice this change when you create a dsource. For more information, refer to Linking using Staging Push mode[110].

3.10.5.1.4 Release 4.2.0

This connector version introduces data source sizing which enables users to calculate the size of ingested MySQL databases. For more information, refer to Source sizing implementation for MySQL dSources[111].

3.10.5.1.5 Release 4.1.0

This connector version blocks the creation, rollback, and refresh actions of VDBs on target environments when the installed MySQL version is not the same as the dSource's installed MySQL version. This is a protective feature that prevents the accidental corruption of MySQL VDBs.

3.10.5.1.6 Release 4.0.0

This connector version supports MySQL v8 across all certified editions.

3.10.5.1.7 Release 3.0.0

This connector version has the following new features:

- The MySQL connector version 3.0.0 has been enhanced for improved security and stability. We have added fixes for the following:
  - Dataset creation workflows can detect if the provided port number is already in use by an existing MySQL instance and notify the user.
  - Corrected the reported dSource and VDB statuses.
  - Fixed the **Snapshot with Parameters** functionality of the Manual Backup Ingestion to work smoothly and without errors.
  - Improved UI descriptions, UI and connector validations, and error messages.
- Added support for Percona Server for MySQL version 5.7 using Manual Backup Ingestion and Binary Log Replication ingestion models.

110 https://cd.delphix.com/docs/latest/staging-push
111 https://cd.delphix.com/docs/latest/mysql-source-sizing-implementation-for-dsource

- Integrated support for MySQL and Percona Server for MySQL version 5.7 in a consolidated release stream. If you have installed divergent or custom branches of the MySQL connector you should look to migrate to this production stream. Future features and bugs will be resolved here.
- Implemented various security fixes.

3.10.5.1.8 Release 2.1.0

The MySQL connector 2.1.0 release consists of security improvements.

3.10.5.1.9 Release 2.0.26

This connector version has the following new features:

- Selecting Databases: Ability to select specific databases in the Source MySQL server when creating a dSource.
- Delphix generated config file: If a `my.cnf` file is not provided while creating a dSource, Delphix now creates one.
- New Exit Return Codes: MySQL connector 2.0.26 introduces more granular error exit codes.

> Delphix began supporting MySQL connector version from v2.0.26.

3.10.5.2 Fixed issues (MySQL)

3.10.5.2.1 Release 4.5.0

Ticket number	Summary
MYSQL-242	Plugin Operation fails while writing logs if any previous logs exits in the toolkit directory.

3.10.5.2.2 Release 4.4.0

Ticket number	Summary
MYSQL-230	Added missing my.cnf update logic within the Rollback operation.

Ticket number	Summary
MYSQL-225	A quick fix to prevent code break.
MYSQL-214	Verified the CLI syntax, and conveyed it to the support team. Validated major CLI operations for the MySQL Document revamp as well.

3.10.5.2.3 Release 4.3.0

Ticket number	Summary
MYSQL-210	ESCL-4684 - Retry logic for Refresh
MYSQL-207	Runbash failing while shutting down(kill) the MySQL instance

3.10.5.2.4 Release 4.2.0

There are no fixed issues in this release.

3.10.5.2.5 Release 4.1.0

Ticket number	Summary
MYSQL-118	The MySQL Instance name is changed to Source Config name to reduce confusion during configuration and standardize across our connectors.

3.10.5.2.6 Release 4.0.0

Ticket number	Summary
MYSQL-138	The length of the MySQL Instance name has now been changed to support up to 255 characters.
MYSQL-81	Changed the command to change the root password which facilitates seamless working of the linking process.

Ticket number	Summary
MYSQL-72	Changed the commands to take backups as they were deprecated.

3.10.5.2.7 Release 3.0.0

There are no fixed issues in this release.

3.10.5.2.8 Release 2.1.0

There are no fixed issues in this release.

3.10.5.2.9 Release 2.0.26

There are no fixed issues in this release.

3.10.6 Oracle E-Business Suite (EBS) 12.2 release notes

This section covers the following topics:

- New features (EBS) (see page 495)
- Fixed issues (EBS) (see page 499)
- Known issues (EBS) (see page 505)

To view the EBS support matrix, see Oracle E-Business Suite (EBS) matrix (see page 1657).

3.10.6.1 New features (EBS)

3.10.6.1.1 Release 4.6.0

The Oracle EBS plugin version 4.6.0 introduces the below enhancements and bug fixes:

- Enhances plugin logging in the following AppsTier workflows:
  - Enable Operation
  - Disable Operation
  - Start Operation
  - Stop Operation
  - Snapshot Operations – pre-snapshot & post-snapshot
  - Environment Status Check Workflow
- Revises Oracle EBS connector documentation for better clarity and usability.

3.10.6.1.2 Release 4.5.0

The Oracle EBS plugin version 4.5.0 introduces the below enhancements and bug fixes:

- With this release, the plugin now enables the linking and virtualization of DBTECHSTACK on AIX operating systems. This update extends support for AIX DBTECHSTACK and database virtualization.
- In this release, an enhancement has been introduced to extend the timeout for the `adcfgclone.pl dbTechStack` job. Previously set at 1800 seconds, the new timeout has been increased to 5400 seconds (90 minutes). This extended timeout is now applicable across all operating systems.
- In the 19c Database version, the `adcdblnctl.sh` script is now the designated tool for executing the listener, regardless of the operating system type or version.
- The DBTECHSTACK plugin now manages the configuration of the `ROOTPRE_EXECUTED=Y ENV` environment variable.
- The CDB environment file is now sourced before triggering the `txkGenCDBTnsAdmin.pl` utility/script. This occurs after the sourcing of `txkSetCfgCDB.env`.

3.10.6.1.3 Release 4.4.0

The Oracle EBS plugin version 4.4.0 introduces below enhancements and bug fixes:

- **Updated Hooks content in hooks utility to support Solaris**
  In this release, we have addressed issues related to the execution of hooks commands in Solaris environments. Certain commands were previously failing to execute as intended. To ensure uniform functionality across both Linux and Solaris machines, modifications have been implemented in the hooks utility. You can now expect consistent performance across platforms.
- **Support for load balancer implementation in EBS setup**
  This release adds support for load balancer implementation in EBS setup, where you can now set `profile_option_value` for `apps_framework_agent` to load balancer hostname.
- **Updated sudoers entries**
  This release adds new sudoers entries to documentation. Please update sudoers entries if you are using privilege elevation. For more information, refer to this[112] documentation.
- **Other general enhancements and bug fixes**

3.10.6.1.4 Release 4.3.0

This plugin release has the following set of new features.

- **EBS Intelligent AppsTier linking**
  This will remove the manual intervention while you are performing cutover from RUN FS(fs1) to PATCH FS(fs2) and now new PATCH FS(fs1), new RUN FS(fs2) and you don't have to make any changes in dSource configuration so linking will work seamlessly.

---

112 https://cd.delphix.com/docs/19.0.0.0/preparing-target-ebs-r12-2-environments-for-provis#id-(16.0.0.0)PreparingtargetEBSR12.2environmentsforprovisioning-SudoersentryforLinux

With 4.3.0 or later, we do not need to specify the "relative paths" of files to exclude in the Paths to Exclude list. Delphix Engine will automatically detect the RUN FS.

- **EBS HealthChecker utility**
  The HealthChecker will perform various pre-requisite checks reporting to ensure OS libraries, utilities, OS user permissions, oracle central inventory, OS resources (CPU, RAM, swap space, storage) etc. are as per Oracle vendor EBS sizing guidelines. if anything is missing, it will be reported and must be fixed before Delphix Virtualization workflows are triggered by end-user
  The HealthChecker ensures Delphix's linking, provision, and refreshes of EBS workflows run smoothly.
- **Performance improvement**
  The latest 4.3.0 plugin version has an improved performance, due to the removal of explicit status checks of WLS Managed servers during the virtual AppsTier provision and refresh phase.
- **Improvement in the re-run of failed VDB provision or refresh workflows**
  With this improvement, you don't have to delete a failed VDB. Instead, you can disable the VDB with the force option, clean up the target server after receiving the 'stale file handle' and stale processes problems, and then re-try the provision or refresh workflow.

3.10.6.1.5 Release 4.2.2

This plugin release consists of only bug fixes.

3.10.6.1.6 Release 4.2.1

This plugin release consists of only bug fixes.

3.10.6.1.7 Release 4.2.0

This plugin release has the following set of new features.

- **EBS database port customization**
  This feature supports the provisioning of VDBs in EBS dbTechStack & AppsTier plugins to run on a customized port. Using this feature you can provision database binaries(ORACLE_HOME) cloning and listener process to run on a custom DB port and Application tier to connect to that port. The EBS Database Port Customization provide flexibility to choose the desired port for running the Delphix-created VDBs & listeners.

> This feature is supported only in Oracle 19c Database.

3.10.6.1.8 Release 4.1.0

The Oracle EBS plugin version 4.1.0 introduces the hooks utility feature and also addresses the handling of an additional prompt for ISG Datasource password during AppsTier provisioning operation for `adcfgclone` execution.

- **Hooks utility feature**
  This feature provides flexibility to end-users where they can provide a command with configure/pre-snapshot as an option in configuring and pre-snapshot hooks during DB provisioning. This is a huge improvement in terms of user experience in comparison to the conventional approach where end-users had to add entire content (by copying it from the user documentation) into the various hooks. The hooks utility also manages the logs for hooks execution; it'll be helpful in scenarios where end users want to track the hook execution. We have not removed or restricted the existing approach of providing hooks content from user documentation, the hooks utility can be used as an alternative to the existing approach.

3.10.6.1.9 Release 4.0.2

This plugin release consists of only bug fixes.

3.10.6.1.10 Release 3.0.3

This plugin release consists of only bug fixes.

3.10.6.1.11 Release 3.0.2

This is a patch release that aims to provide the following updates to EBS 12.2 plugin.

- Updated Hooks for 19c MT low and high privileged users.
- Fixed an issue where 19c DBTechStack provision fails during `adcfgclone` with java exception.

3.10.6.1.12 Release 3.0.1

This is a patch release that aims to provide the following updates to EBS 12.2 plugin.

- Added support for RHEL 8.3
- Updated Hooks for 19c MT low and high privileged users.

3.10.6.1.13 Release 3.0.0

The Oracle EBS plugin version 3.0.0 aims to support Oracle 19c MT architecture. Moreover, a new field has been added on the provisioning UI. This will provide the flexibility to the end-user to provide the maximum time allowed to virtualize the application tier.

3.10.6.1.14 Release 2.0.1

The Oracle EBS plugin version 2.0.1 aims to successfully stop application services while provisioning or refresh operation. Hence, creating successful snapshots.

3.10.6.1.15 Release 2.0.0

The Oracle EBS plugin version 2.0.0 aims to implement Sudo Privileges feature for both EBS 12.2 Database Tier and Application Tier.

3.10.6.1.16 Release 1.8.4

The Oracle EBS plugin version 1.8.4 aims to implement Weblogic password complexity validation as per the Oracle standards.

3.10.6.1.17 Release 1.8.3

The Oracle EBS plugin version 1.8.3 introduces a new feature while refreshing EBS appsTier where a Pre Refresh hook will help in aborting concurrent manager services and reduce the time taken to gracefully stop the EBS application.. The password required to execute the same will be pulled from a text file present on the target appsTier host which will be created automatically during appsTier provisioning.

3.10.6.1.18 Release 1.8.2

Oracle EBS plugin version 1.8.2 introduces a new feature where the user will not need to specify the source apps schema password in EBS DB hooks provision wizard. This password can now be pulled from a text file present on the target DB host which will be created automatically during DBTechStack provisioning.

3.10.6.2 Fixed issues (EBS)

3.10.6.2.1 Release 4.6.0

Ticket number	Summary
EBS-1510	AppsTier provision/refresh workflow used to halt at APPLPTMP prompt because expect block generated by adcfgclone.pl utility doesn't match with older versions as the keyword "**value**" after APPLPTMP gives us problem in AD/TXK codelevel 13 or later releases. Prompt change for APPLPTMP input during adcfgclone.pl[113] utility run. adcfgclone.pl with AD/TXK codelevel 12 or lower generates prompt like -

113 http://adcfgclone.pl

	`Choose a value which will be set as APPLPTMP value on the target node [1]:`
	adcfgclone.pl with AD/TXK codelevel 13 or later generates prompt like -
	`Choose a value which will be set as APPLPTMP on the target node [1]:`
	With 4.6.0 release, It smoothly handles the situation for AD/TXK codelevel 15 or lower versions as well.

EBS-1490	A regression bug from previous plugin version 4.5.0, update_adpreclone_file function is failing to comment check_db_connection(); line in adpreclone.pl on DBTechStack side
EBS-1483	Correcting check_value variable's value in 12c hooks

3.10.6.2.2 Release 4.5.0

Ticket number	Summary
EBS-1469	Correction in functions update_adpreclone_file() and replace_string_in_file(). A regression bug from 4.4.0 where adpreclone.pl version is latest and we comment check_db_connection() function call in adpreclone.pl script while taking dbTechStack snapshot and uncommenting the changes after taking the VDB dataset snapshot.
EBS-1461	Removing dependency of lsof utility in solaris to get the running processes information.

3.10.6.2.3 Release 4.3.0

Ticket number	Summary
EBS-1319	In DBTECHSTACK provision, Plugin should not run adpreclone.pl dbTechStack command
EBS-1312	WLS password is saved in updatedspassword file
EBS-1255	Unable to refresh a failed provision

EBS-874	Detach oracle home and delete apps password file during unconfigure/delete operation and cleanup before provision

3.10.6.2.4 Release 4.2.2

Ticket number	Summary
EBS-1292, HF-1225	Plugin was using wrong Autoconfig script location if `s_jdbc_connect_descriptor_generation` was not true in CONTEXT_FILE
EBS-1280	Strong password check for WLS password as per EBS requirement
EBS-1169	Source apps password file removed from all remote hosts to improve security.
EBS-1285	Implementation of AppData Bundle Redaction parameters in EBS plugin
EBS-1256	Expect utility availability is checked on additional application nodes for smooth multi-node provisioning.

3.10.6.2.5 Release 4.2.1

Ticket number	Summary
EBS-1265	Hooks utility fails to get the short hostname if target server is SunOS SOLARIS
EBS-1278	Handling detach ORACLE HOME when inventory.xml has multiple HOMES
EBS-1259	If someone is upgrading from Lua(2.0.x) to vSDK version(4.x.x) and using oracle 11g, the end user will not be able to take snapshot in Delphix Engine
EBS-1275	Fix lsof issue in Solaris and Linux

3.10.6.2.6 Release 4.2.0

There are no fixed issues in this release.

3.10.6.2.7 Release 4.1.0

There are no fixed issues in this release.

3.10.6.2.8 Release 4.0.2

There are no fixed issues in this release.

3.10.6.2.9 Release 3.0.3

Key	Summary
EBS-1027	EBS 12.2 Plugin will validate user-provided ORACLE_HOME during DbTier provisioning
EBS-1028	Enhancements identified in dbTechStack code
EBS-1038	Fix shellchecks warnings/errors in hooks and sudoers entry for target dbtier
EBS-1066	Make the check stricter for listener process in DBTechStack status module and fix shellchecks

3.10.6.2.10 Release 3.0.2

Key	Summary
EBS-890	19C DBTechStack provision fails during adcfgclone with java exception

3.10.6.2.11 Release 3.0.1

There are no fixed issues in this release.

3.10.6.2.12 Release 3.0.0

Key	Summary
EBS-767	EBS appsTier linking failed while identifying Delta patch level information from low-privilege OS user
EBS-779	Continued timeouts - toolkit.ebs122-app-1-8-2.errorTimeout {admanagedsrvctl}
EBS-787	Toolkit should perform prior dependency/pre-requisite check in target host before starting to provision dBtechStack and Appstier
EBS-788	Error not being shown to the customer
EBS-798	GUI shows vfiles ORACLE_HOME as stopped however it is not
EBS-778	Expect timeout set to 8 hours which is too short
EBS-798	GUI shows vfiles ORACLE_HOME as stopped however it is not
EBS-816	EBS 12.2 disable Appstier vdb failed with Delphix cannot stop all processes under /u01/oracle/VIS on host

3.10.6.2.13 Release 2.0.1

Key	Summary
EBS-783	Provisioning VDB from Appstier dSource fails with adpreclone connection with Oracle WebLogic Administration Server

3.10.6.2.14 Release 2.0.0

Key	Summary
EBS-757	EBS plugin should check for data accessibility during status check.

3.10.6.2.15 Release 1.8.4

Key	Summary
EBS-712	WLS Password Complexity Issue - Merge ESCL_2983 into EBS 12.2 Plugin.
EBS-713	Multi-line support for SQL query O/P - merge ESCL-2988 into EBS 12.2 Plugin.

3.10.6.2.16 Release 1.8.3

Key	Summary
EBS-663	Abort EBS ConcMgr services before stopping EBS appsTier services during refresh.

3.10.6.2.17 Release 1.8.2

Key	Summary
EBS_566	Display port pool number in the error toolkit.ebs122-app-1-7-0.portPoolNotFreeError.
EBS-597	Delphix Hooks Password-Storing Apps password to file for EBS appsTier

3.10.6.3 Known issues (EBS)

3.10.6.3.1 Release 4.6.0

Key	Summary	Workaround
EBS-1509	With privilege elevation user, if `perl rpm/fileset` is corrupted at the OS level then perl binary that comes with EBS AppsTier or DBTechStack cannot be used to invoke `adpreclone.pl` or `adcfgclone.pl` utility.	You must install `perl rpm/fileset` at the OS level and the `perl` utility must be usable by Oracle OS users.

3.10.6.3.2 Release 4.5.0

There are no known issues in this release.

3.10.6.3.3 Release 4.4.0

There are no known issues in this release.

3.10.6.3.4 Release 4.3.0

There are no known issues in this release.

3.10.6.3.5 Release 4.2.2

There are no known issues in this release.

3.10.6.3.6 Release 4.2.1

There are no known issues in this release.

3.10.6.3.7 Release 4.2.0

There are no known issues in this release.

3.10.6.3.8 Release 4.1.0

There are no known issues in this release.

3.10.6.3.9 Release 4.0.2

There are no known issues in this release.

3.10.6.3.10 Release 3.0.3

There are no known issues in this release.

3.10.6.3.11 Release 3.0.2

There are no known issues in this release.

3.10.6.3.12 Release 3.0.1

Key	Summary
EBS-836	"ebs_<PDB_SID>" listener services are getting disappeared on VDB stop/start. As a workaround, a new Third Configure Clone Hook has been added.

3.10.6.3.13 Release 3.0.0

Key	Summary
EBS-836	"ebs_<PDB_SID>" listener services are getting disappeared on VDB stop/start. As a workaround, a new configure clone hook (see page 1701) has been added.

3.10.6.3.14 Release 2.0.1

There are no known issues in this release.

3.10.6.3.15 Release 2.0.0

There are no known issues in this release.

3.10.6.3.16 Release 1.8.4

There are no known issues in this release.

3.10.6.3.17 Release 1.8.3

Key	Summary
EBS-692	EBS link operation fails when source EBS environment is linked with BYOOJ
DLPX-69617	OperationRunnerImpl incorrectly assumes toolkit JDK always exists

3.10.6.3.18 Release 1.8.2

Key	Summary
EBS-692	EBS link operation fails when source EBS environment is linked with BYOOJ
DLPX-69617	OperationRunnerImpl incorrectly assumes toolkit JDK always exists

3.10.7 PostgreSQL release notes

This section covers the following topics:

- New features (PostgreSQL) (see page 507)
- Fixed issues (PostgreSQL) (see page 512)
- Known issues (PostgreSQL) (see page 519)

To view the PostgreSQL support matrix, see PostgreSQL matrix (see page 1769).

3.10.7.1 New features (PostgreSQL)

3.10.7.1.1 Release 4.2.3

This plugin release consists of only bug fixes.

3.10.7.1.2 Release 4.2.2

This plugin release consists of only bug fixes.

> This plugin release includes a bug fix that may not function optimally in certain customer environments. Consequently, access to download this version is restricted in this release. If you need to download this specific version, please reach out to our dedicated Customer Support team for assistance.

3.10.7.1.3 Release 4.2.1

This plugin release consists of only bug fixes.

3.10.7.1.4 Release 4.2.0

This plugin release has the following set of new features.

- **Auto Roll Forward Optional**
  With PostgreSQL Plugin 4.2.0 and onwards, the plugin will apply the latest wal logs copied to the wal log path only when the snapshot button clicks from UI. For more information, refer to Creating a dSource with external backups using WAL Logs (see page 1802).
- **Ubuntu 18.04 with Postgres 15 certification**
  This release certifies Ubuntu 18.04 with Postgres 15.

3.10.7.1.5 Release 4.1.1

This plugin release consists of only bug fixes.

3.10.7.1.6 Release 4.1.0

This plugin release has the following set of new features.

- **Source sizing for dSource with ZFS**
  With PostgreSQL Plugin 4.1.0 and onwards, dSource database size will be calculated using `du` command on dSource mount path. For more information, see Source sizing implementation for dSource (see page 1824).
- **Azure Database for PostgreSQL - Flexible Server Certification**
  Added support for the PaaS-to-IaaS ingestion model of Azure Flexible Server 11, 12, 13, and 14 versions.

3.10.7.1.7 Release 4.0.0

This plugin release has the following set of new features.

- **Provisioning a PostgreSQL VDB PiT with external logs**
  With the latest PostgreSQL Plugin 4.0.0 and onwards, you can provision a VDB to a point in time by providing the WAL logs on the target host. For detailed information, see Provisioning a PostgreSQL VDB PiT with External Logs (see page 1831).
- **pg_dumpall support for single database ingestion**
  PostgreSQL plugin 4.0.0, supports `pg_dumpall` which has been added with Single Database Ingestion to get the global objects like roles and tablespaces.

3.10.7.1.8 Release 3.2.0

This plugin release has the following set of new features.

- **Enhancement to the external backup mechanism**
  With the 3.2.0 release, while using External Backup Mechanism it is now **not** mandatory for you to provide the backups as ZIP files. This enhancement thus removes the mandatory requirement of `pg_basebackup` to be encapsulated in `PostgreSQL_T<YYYYMMDD>.zip format`. You can now store the backups (as tars) on the staging server directly, without providing the backups as ZIP files (reducing the time spent on creating ZIP files) using the following additional ways:
  - Within folders (named `PostgreSQL_T<YYYYMMDDhhmmss>`) created inside the provided backup path.
  - Within the provided backup path - `PostgreSQL Backup File Path`

  Switching to the new mechanism is easy, just store the full backup within the same backup path and run a resync operation.
  The plugin still supports the usage of zip files.
- **Introducing a LOG_LEVEL parameter in the PostgreSQL plugin configuration**
  PostgreSQL plugin supports remote side logging, with all plugin logs available within the scratch path on the remote host (`<tookit_path>/Delphix_COMMON_*/plugin/postgres-vsdk_*`). You can tweak the log parameters related to log rotation within the plugin configuration file called `dlpx_postgres_config.ini`. The 3.2.0 version, introduces a new parameter called **LOG_LEVEL**, with the capability of handling the amount of logs being generated on the remote host. Accepted values are **DEBUG, INFO, WARNING, and ERROR**, with **INFO** being the default value.

3.10.7.1.9 Release 3.1.0

This plugin release has the following set of new features.

- **Source sizing implementation on PostgreSQL plugin**
  This feature enables you to check the size of PostgreSQL cluster or database being virtualized from the Delphix Engine. The calculation of the dataset size is based on the ingestion method used. For staging pull methods, if the whole cluster is ingested then the size of the cluster will be calculated or

else the size of the ingested single database will be calculated. Source sizing cannot be implemented when staging push method is used.

> This feature requires the plugin to run SQL queries on the datasets. If the plugin cannot login to the dataset for any reason, then source sizing cannot be implemented. For more details about this feature, see Source Sizing Implementation for dSource (see page 1824).

If you are using privilege elevation and upgrading to PostgreSQL 3.1.0 version, then you will have to change the sudoers entry. For more details see, Prerequisites for Privilege Elevation using DLPX_DB_EXEC Script (see page 1780).

3.10.7.1.10 Release 3.0.0

This plugin release has the following set of new features.

- **Staging push**
  The PostgreSQL Plugin 3.0.0 release introduces a new data ingestion mechanism i.e. Staging Push that enables you to push data into the Delphix-provided mount point on your own. It is an alternative Delphix dSource data ingestion approach that enables you to push the data in Delphix, which is different from the conventional pull model where Delphix pulls the data.
  Delphix relinquishes the responsibility to prepare the data (ie: recover to a specific state) on the staging push. It is different from conventional staging where Delphix controls the state of recovery of the staging database.
  There are two mechanisms to create dSource on the staging server.
  - Pull architecture: The Postgres plugin pulls the data into the Delphix-provided mount location.
  - Staging Push architecture: You push the data into the Delphix-provided mount point on your own.

  For more details, see Staging Push Implementation (see page 1821).

3.10.7.1.11 Release 2.1.1

This plugin release consists of only bug fixes.

3.10.7.1.12 Release 2.1.0

This plugin release has the following set of new features.

- **Privilege elevation**
  The Privilege Elevation feature allows you to use a low privileged OS user to create a dSource or VDB. To use this feature, you must provide a username in the **Privileged OS account (optional)** field on the dSource creation page. The plugin will then use the `dlpx_db_exec` script to perform the privilege

elevation. The script has to be updated via the engine API before the creation/discovery of an environment.

3.10.7.1.13 Release 2.0.0

The PostgreSQL Plugin version 2.0.0 includes the following improvements and features:

- Plugin generated log file will now be placed at <toolkit path>/Delphix_COMMON_<long id>/plugin/postgres-vsdk_12c9a11f-fab6-4792-a5da-49d726c004df/<os user> instead of <toolkit path>/postgres/logs.
- Improved logging
- The plugin logs will be part of the support bundle.
- The plugin logs will be available on the target host as well as on the Delphix Engine host.
- Introduction of a new generic config ini file dlpx_postgres_config.ini which will be present at the scratch path. The previous log configure file dlpx_pg_logrotate.conf will no longer be available and the contents of this file will be moved to the generic config ini file-*dlpx_postgres_config.ini*.
- The PostgreSQL plugin will now have one more ingestion mechanism - Single Database Ingestion that allows you to virtualize a single database.

3.10.7.1.14 Release 1.5.1

This release includes the Streaming Replication Monitoring feature that monitors the streaming replication by reading the Postgres database logs every minute.

3.10.7.1.15 Release 1.5.0

This plugin release consists of only bug fixes.

3.10.7.1.16 Release 1.4.4

This plugin release consists of only bug fixes.

3.10.7.1.17 Release 1.4.3

This plugin release consists of only bug fixes.

3.10.7.1.18 Release 1.4.2

This plugin release consists of only bug fixes.

3.10.7.1.19 Release 1.4.1

This plugin release consists of only bug fixes.

3.10.7.1.20 Release 1.4.0

PostgreSQL Plugin version 1.4.0 contains the following certifications :

- Operating System - RHEL/CentOS 7.8 version Support with PostgreSQL 12.x.

3.10.7.2 Fixed issues (PostgreSQL)

3.10.7.2.1 Release 4.2.3

Key	Summary
POST-442	PostgreSQL: SnapSync with resync says it will block share but does not
POST-963	Poor error when dSource fails to enable due to a failure during DB_SYNC after adding new dSource
POST-971	PostgreSQL plugin fails to provision on hosts without locale en_US.UTF-8 installed
POST-1003	min_wal_size should be copied from source
POST-1094	Tablespace handling isn't working correctly for VDBs provisioned from single db ingestion dSource snapshot
POST-1082	Revert POST-1027 change for graceful restore
POST-1100	Restrict upgrade from LUA to 4.2.3 version of vSDK directly

3.10.7.2.2 Release 4.2.2

Key	Summary
POST-1000	Staging push error "Not able to find database files inside it" confuses users
POST-1025	The PG dSource cluster should not be stopped to run a SnapSync operation in any case

Key	Summary
POST-1040	pg_dumpall command should use a single database parameter

3.10.7.2.3 Release 4.2.1

Key	Summary
POST-1027	PostgreSQL SnapSync fails to restore certain tables but succeeds without warning users. See the Known issues section if you are experiencing errors with the SnapSync process.
POST-1013	PostgreSQL VDB is not closing open connections during refresh / rollback
POST-1034	Source sizing is failing in plugin with internal error

3.10.7.2.4 Release 4.2.0

There are no fixed issues in this release.

3.10.7.2.5 Release 4.1.1

Key	Summary
POST-690	Incorrect error: The mount path has been changed from the UI
POST-936	def locate_files & def __find_repo_using_env_variable are not able to suppress the warning "Permission denied"

3.10.7.2.6 Release 4.1.0

Key	Summary
POST-720	Postgres staging and target database should use the same local as the source database
POST-85	The Plugin post-snapshot workflow breaking for non-english locales

3.10.7.2.7 Release 4.0.0

There are no fixed issues in this release.

3.10.7.2.8 Release 3.2.0

Ticket number	Summary
POST-293	Compressed backup and backup timestamp
POST-470	Pg_vSDK >> Plugin should add a parameter to control debug log ON/OFF in `dlpx_pg_logrotate.conf`
POST-497	Ingestion from backup should be more efficient
POST-699	Removing source sizing for VDB

3.10.7.2.9 Release 3.1.0

Ticket number	Summary
POST-596	New mechanism of creating files to remove wildcard entry for `echo` in sudoers file.
POST-611	Ubuntu Certification >> Linking is failing when pg_hba.conf and pg_ident.conf files are not at data directory

Ticket number	Summary
POST-642	dSource Single DB Ingestion >> User should not be allowed to change database name parameter, unless it's a RESYNC operation
POST-668	Staging Push >> Failure to provision a VDB using privilege elevation from Staging Push snapshot with provisioning allowed

3.10.7.2.10 Release 3.0.0

Ticket number	Summary
POST-537	`postgresql.auto.conf` and `recovery.signal` config files needs to be handled in VDB creation
POST-546	`logging_collector` should be enabled for the dSource
POST-563	SP >> Snapshot operation is failing if the port value is quoted with the strings in config file
POST-565	PG 2.1.1 >> Variable value is missing in error
POST-566	Time difference issue between Delphix Engine and Staging Environment Host
POST-571	Environment variables between low-privileged user & high-privileged user are not translated

3.10.7.2.11 Release 2.1.1

Ticket number	Summary
POST-555	Logging of large content is failing at staging/target side
POST-562	Echo command in .bash_profile of the user blocks several Plugin operations

3.10.7.2.12 Release 2.1.0

Ticket number	Summary
POST-494	Using dlpx_db_exec for privilege elevation
POST-511	`max_connections` parameter on staging should be set according to the source
POST-518	`netstat` utility is not available in SuSE 15

3.10.7.2.13 Release 2.0.0

PostgreSQL version 2.0.0 contains the following bug fixes and enhancements:

Ticket number	Summary
POST-300	pg_basebackup failure not captured
POST-303	Default value for PostgreSQL Mount Path is wrong
POST-306	Delphix GUI allows PostgreSQL VDB port to be changed while the VDB is running
POST-422	Snapshots created for dSource despite failure to start staging database
POST-428	Instance failure reason not captured
POST-431	Unable to provision PostgreSQL VDB with parameters that use the strftime specification formatting
POST-441	Postgres dsource creation with existing backup fails due to missing wal file

3.10.7.2.14 Release 1.5.1

Ticket number	Summary
POST-418	Replication monitoring - Lua plugin

3.10.7.2.15 Release 1.5.0

Ticket number	Summary
POST-202	If the port is not present in postgresql.conf on the source, it's not provided in staging and VDBs
POST-283	Security enhancement - Plugin overrides listen_addresses parameter in posgresql.conf to "localhost" on staging & target server
POST-292	dSource configuration parameters don't accept empty values
POST-295	The staging server is created using a copy of the production parameters file
POST-297	Postgres Plugin Should check if the toolkit directory is accessible by the OS user
POST-298	primary_conninfo' parameter is not getting commented out in vdb flow
POST-302	Allow users to set empty values for a parameter in a configuration file
POST-312	delphix_postgres_debug.log is too large to manage
POST-316	Propagation of Hotfix 1112 code changes to plugin 1.5.0 release

3.10.7.2.16 Release 1.4.4

Key	Summary
POST-291	Error Handling for Authentication Failure in `pg_basebackup` command

3.10.7.2.17 Release 1.4.3

Key	Summary
POST-284	The plugin doesn't check if the staging Postgres database is in a stable state or not

3.10.7.2.18 Release 1.4.2

Key	Summary
POST-274	`data_directory` parameter in the postgresql.conf file is causing failure in dSource and VDB flow
POST-275	Linking Fails if the parameter which needs to be configured is commented through multiple `#`
POST-276	Linking Fails if the mandatory parameter required to setup replication is commented in config file

3.10.7.2.19 Release 1.4.1

Key	Summary
POST-269	Postgres DB_SYNC fails - ESCL-3179
POST-270	Plugin should handle postgresql.auto.conf entries during dsource creation

3.10.7.2.20 Release 1.4.0

There are no fixed issues in this release.

3.10.7.3 Known issues (PostgreSQL)

3.10.7.3.1 Release 4.2.3

There are no known issues in this release.

3.10.7.3.2 Release 4.2.2

- POST-1027 - The PostgreSQL SnapSync resolution which was intended to ingest missing tables, introduced additional failures during the SnapSync process. The SnapSync process will fail without a clear resolution if an extension is missing. The issue has been resolved in v4.2.3.

3.10.7.3.3 Release 4.2.1

- POST-1027 - The PostgreSQL SnapSync resolution which was intended to ingest missing tables, introduced additional failures during the SnapSync process. The SnapSync process will fail without a clear resolution if an extension is missing. The issue has been resolved in v4.2.3.

3.10.7.3.4 Release 4.2.0

There are no known issues in this release.

3.10.7.3.5 Release 4.1.1

There are no known issues in this release.

3.10.7.3.6 Release 4.1.0

There are no known issues in this release.

3.10.7.3.7 Release 4.0.0

There are no known issues in this release.

3.10.7.3.8 Release 3.2.0

There are no known issues in this release.

3.10.7.3.9 Release 3.1.0

There are no known issues in this release.

3.10.7.3.10 Release 3.0.0

There are no known issues in this release.

3.10.7.3.11 Release 2.1.1

There are no known issues in this release.

3.10.7.3.12 Release 2.1.0

There are no known issues in this release.

3.10.7.3.13 Release 2.0.0

There are no known issues in this release.

3.10.7.3.14 Release 1.5.1

There are no known issues in this release.

3.10.7.3.15 Release 1.5.0

There are no known issues in this release.

3.10.7.3.16 Release 1.4.4

There are no known issues in this release.

3.10.7.3.17 Release 1.4.3

There are no known issues in this release.

3.10.7.3.18 Release 1.4.2

There are no known issues in this release.

3.10.7.3.19 Release 1.4.1

There are no known issues in this release.

3.10.7.3.20 Release 1.4.0

There are no known issues in this release.

3.10.8 SAP HANA release notes

This section covers the following topics:

- New features (SAP HANA) (see page 521)
- Fixed issues (SAP HANA) (see page 523)
- Known issues (SAP HANA) (see page 526)

To view the SAP HANA support matrix, see SAP HANA matrix (see page 1982).

3.10.8.1 New features (SAP HANA)

3.10.8.1.1 Release 6.3.3

This plugin release has no new features. This release fixes the logic in the plugin to calculate the port numbers during VDB creation for HANA installations having instance numbers other than 00. For more information, refer to the Fixed Issues (see page 523) section.

3.10.8.1.2 Release 6.3.2

This plugin release consists of only bug fixes.

3.10.8.1.3 Release 6.3.1

This plugin release consists of only bug fixes.

3.10.8.1.4 Release 6.3.0

This plugin release has the following set of new features.

- **Source sizing implementation on HANA plugin**
  The SAP HANA 2.0 Plugin version 6.3.0 introduces the source sizing feature, that aligns the data source experience with Delphix database standards which will improve your ingestion reporting experience on the per TB pricing model. This feature enables you to calculate the dataset size based on the total allocated space on the mount point provided by the Delphix Engine for the dataset. Source sizing is not implemented for VDB.

3.10.8.1.5 Release 6.2.1

The SAP HANA 2.0 Plugin version 6.2.1 validates the SAP HANA's Staging Push solution with third party backup vendors.

3.10.8.1.6 Release 6.2.0

The SAP HANA 2.0 Plugin version 6.2.0 introduces the **Staging Push** feature. This feature provides a new data ingestion mechanism that helps users to push data into the Delphix provided mount point on their own.

3.10.8.1.7 Release 6.1.1

This patch release contains an enhancement that allows the `create_backup_metadata.sh` script to be executed in a non-interactive mode as well.

3.10.8.1.8 Release 6.1.0

This plugin release has the following set of new features.

- **SAP HANA port control**
  With this release, a solution has been introduced that keeps the port numbers consistent throughout the VDB life cycle so that the connections made to the VDBs are not disrupted during their life cycle.

3.10.8.1.9 Release 6.0.1

This plugin release consists of only bug fixes.

3.10.8.1.10 Release 6.0.0

This is a conversion from SAP HANA direct plugin to vSDK-based, staged plugin. This plugin will support only customer-provided/External backups ingestion mechanism.

3.10.8.1.11 Release 4.3.2

This plugin release consists of only bug fixes.

3.10.8.1.12 Release 4.3.1

This plugin release consists of only bug fixes.

3.10.8.1.13 Release 4.3.0

Below features are being provided:

1. **Non-ingestion of backups into Delphix**: This means that database backups will not be transferred into the Delphix Engine during any of the operation. This will be driven on the basis of a checkbox that can be selected by the customer during dSource creation.
2. Support provided for log backups along with the data backups (only when non-ingestion has been chosen during dSource creation).
3. Ability to use Delphix provided storage for all virtualized tenant services. This is valid even for the user-created services outside the plugin.
4. Multiple HANA installations on the same VM will also be supported.
5. Support for Self-Service (Jetstream) has been certified with this release.
6. Certification of plugin support with HANA Standard and Enterprise Edition 2.0 SP05 on RHEL 8.1.

3.10.8.1.14 Release 4.2.0

With this release, we are including the support for basepath_databackup parameter. Backups created via plugin will now be created on the PATH specified by this parameter. This parameter can be defined at the system and at the tenant level. If this parameter has not been specified then default location will be used for creating the backups.

3.10.8.1.15 Release 4.1.2

This plugin release consists of only bug fixes.

3.10.8.1.16 Release 4.1.1

This plugin release consists of only bug fixes.

3.10.8.2 Fixed issues (SAP HANA)

3.10.8.2.1 Release 6.3.3

Delphix ticket number	Summary
CTCHANA-1961	Port numbers determination during VDB creation is failing for the hana instance other than 00

3.10.8.2.2 Release 6.3.2

Delphix ticket number	Summary
CTCHANA-1936	Handling services not having volumes information in m_volumes for VDB creation.

3.10.8.2.3 Release 6.3.1

Delphix ticket number	Summary
CTCHANA-1914	Refining the plugin discovery logic to search for HANA instances.

3.10.8.2.4 Release 6.3.0

Delphix ticket number	summary
CTCHANA-1906	Changing the plugin's environment discovery logic to refer "HDB --version" instead of "landscapeDescription.xml"
CTCHANA-1900	Removal of tenant parameters from UI and plugin
CTCHANA-1891	Incorrect execution time for VDB

3.10.8.2.5 Release 6.2.1

There are no fixed issues in this release.

3.10.8.2.6 Release 6.2.0

There are no fixed issues in this release.

3.10.8.2.7 Release 6.1.1

Delphix ticket number	Summary
CTCHANA-1661	v6.0.0.0 create_backup_metadata.sh needs ability to run non-interactively
CTCHANA-1667	HANA Recovery failure on staging host yields difficult to decipher error

3.10.8.2.8 Release 6.1.0

Delphix ticket number	Summary
CTCHANA-821	HANA SQL Port value should remain consistent for the life of VDB.

3.10.8.2.9 Release 6.0.1

There are no fixed issues in this release.

3.10.8.2.10 Release 6.0.0

There are no fixed issues in this release.

3.10.8.2.11 Release 4.3.2

Delphix ticket number	Summary
CTCHANA-1666	Backporting the password change through hooks issue to the master branch
CTCHANA-1660	remove dependency on /etc/SuSE-release for suse o/s platforms

3.10.8.2.12 Release 4.3.1

There are no fixed issues in this release.

3.10.8.2.13 Release 4.3.0

There are no fixed issues in this release.

3.10.8.2.14 Release 4.2.0

There are no fixed issues in this release.

3.10.8.2.15 Release 4.1.2

There are no fixed issues in this release.

3.10.8.2.16 Release 4.1.1

There are no fixed issues in this release.

3.10.8.3 Known issues (SAP HANA)

3.10.8.3.1 Release 6.3.2

Delphix ticket number	Summary	Workaround/Comments
DLPX-75383	Refresh/Disable not working after failed delete operation	Force delete the VDB. Manually remove the mount point
DLPX-75385	Undo Refresh operation after a failed Refresh job leads to 502 Bad gateway and management stack restart	None
DLPX-67911	Failed enable operation leaving a mount point and next delete operation creating a stale file handle	Manually remove the mount point
CTCHANA-1599	Mount Point change not detected by the plugin may lead to multiple mounts	Mount point not in use can be removed manually
CTCHANA-1598	Plugin loses control in case of user cancel operation.	Manual cleanup can be done in case the plugin leaves any tenant database/files on the VM

Delphix ticket number	Summary	Workaround/Comments
CTCHANA-1586	Refresh after a Force disable on a VDB fails with 'mount not empty' error	Manually remove the mount before proceeding with the Refresh
CTCHANA-1565	Stop/Start operation on a VDB post failed post_refresh hook fails with inappropriate error	None
CTCHANA-1514	dSource tenant database can be accessed and written onto	None
CTCHANA-1610	'test_force_disable_enable_vdb' fails across multiple images	None

3.10.8.3.2 Release 6.3.1

Delphix ticket number	Summary	Workaround/Comments
DLPX-75383	Refresh/Disable not working after failed delete operation	Force delete the VDB. Manually remove the mount point
DLPX-75385	Undo Refresh operation after a failed Refresh job leads to 502 Bad gateway and management stack restart	None
DLPX-67911	Failed enable operation leaving a mount point and next delete operation creating a stale file handle	Manually remove the mount point
CTCHANA-1599	Mount Point change not detected by the plugin may lead to multiple mounts	Mount point not in use can be removed manually
CTCHANA-1598	Plugin loses control in case of user cancel operation.	Manual cleanup can be done in case the plugin leaves any tenant database/files on the VM
CTCHANA-1586	Refresh after a Force disable on a VDB fails with 'mount not empty' error	Manually remove the mount before proceeding with the Refresh

Delphix ticket number	Summary	Workaround/Comments
CTCHANA-1565	Stop/Start operation on a VDB post failed post_refresh hook fails with inappropriate error	None
CTCHANA-1514	dSource tenant database can be accessed and written onto	None
CTCHANA-1610	'test_force_disable_enable_vdb' fails across multiple images	None

3.10.8.3.3 Release 6.3.0

Delphix ticket number	Summary	Workaround/Comments
DLPX-75383	Refresh/Disable not working after failed delete operation	Force delete the VDB. Manually remove the mount point
DLPX-75385	Undo Refresh operation after a failed Refresh job leads to 502 Bad gateway and management stack restart	None
DLPX-67911	Failed enable operation leaving a mount point and next delete operation creating a stale file handle	Manually remove the mount point
CTCHANA-1599	Mount Point change not detected by the plugin may lead to multiple mounts	Mount point not in use can be removed manually
CTCHANA-1598	Plugin loses control in case of user cancel operation.	Manual cleanup can be done in case the plugin leaves any tenant database/files on the VM
CTCHANA-1586	Refresh after a Force disable on a VDB fails with 'mount not empty' error	Manually remove the mount before proceeding with the Refresh
CTCHANA-1565	Stop/Start operation on a VDB post failed post_refresh hook fails with inappropriate error	None

Delphix ticket number	Summary	Workaround/Comments
CTCHANA-1514	dSource tenant database can be accessed and written onto	None
CTCHANA-1610	'test_force_disable_enable_vdb' fails across multiple images	None

3.10.8.3.4 Release 6.2.1

Delphix ticket number	Summary	Workaround/Comments
DLPX-75383	Refresh/Disable not working after failed delete operation	Force delete the VDB. Manually remove the mount point
DLPX-75385	Undo Refresh operation after a failed Refresh job leads to 502 Bad gateway and management stack restart	None
DLPX-67911	Failed enable operation leaving a mount point and next delete operation creating a stale file handle	Manually remove the mount point
CTCHANA-1599	Mount Point change not detected by the plugin may lead to multiple mounts	Mount point not in use can be removed manually
CTCHANA-1598	Plugin loses control in case of user cancel operation.	Manual cleanup can be done in case the plugin leaves any tenant database/files on the VM
CTCHANA-1586	Refresh after a Force disable on a VDB fails with 'mount not empty' error	Manually remove the mount before proceeding with the Refresh
CTCHANA-1565	Stop/Start operation on a VDB post failed post_refresh hook fails with inappropriate error	None
CTCHANA-1514	dSource tenant database can be accessed and written onto	None

Delphix ticket number	Summary	Workaround/Comments
CTCHANA-1610	'test_force_disable_enable_vdb' fails across multiple images	None

3.10.8.3.5 Release 6.2.0

Delphix ticket number	Summary	Workaround/Comments
DLPX-75383	Refresh/Disable not working after failed delete operation	Force delete the VDB. Manually remove the mount point
DLPX-75385	Undo Refresh operation after a failed Refresh job leads to 502 Bad gateway and management stack restart	None
DLPX-67911	Failed enable operation leaving a mount point and next delete operation creating a stale file handle	Manually remove the mount point
CTCHANA-1599	Mount Point change not detected by the plugin may lead to multiple mounts	Mount point not in use can be removed manually
CTCHANA-1598	Plugin loses control in case of user cancel operation.	Manual cleanup can be done in case the plugin leaves any tenant database/files on the VM
CTCHANA-1586	Refresh after a Force disable on a VDB fails with 'mount not empty' error	Manually remove the mount before proceeding with the Refresh
CTCHANA-1565	Stop/Start operation on a VDB post failed post_refresh hook fails with inappropriate error	None
CTCHANA-1514	dSource tenant database can be accessed and written onto	None
CTCHANA-1610	'test_force_disable_enable_vdb' fails across multiple images	None

3.10.8.3.6 Release 6.1.1

Delphix ticket number	Summary	Workaround/Comments
DLPX-75383	Refresh/Disable not working after failed delete operation	Force delete the VDB. Manually remove the mount point
DLPX-75385	Undo Refresh operation after a failed Refresh job leads to 502 Bad gateway and management stack restart	None
DLPX-67911	Failed enable operation leaving a mount point and next delete operation creating a stale file handle	Manually remove the mount point
CTCHANA-1599	Mount Point change not detected by the plugin may lead to multiple mounts	Mount point not in use can be removed manually
CTCHANA-1598	Plugin loses control in case of user cancel operation.	Manual cleanup can be done in case the plugin leaves any tenant database/files on the VM
CTCHANA-1586	Refresh after a Force disable on a VDB fails with 'mount not empty' error	Manually remove the mount before proceeding with the Refresh
CTCHANA-1565	Stop/Start operation on a VDB post failed post_refresh hook fails with inappropriate error	None
CTCHANA-1514	dSource tenant database can be accessed and written onto	None
CTCHANA-1610	'test_force_disable_enable_vdb' fails across multiple images	None

3.10.8.3.7 Release 6.1.0

Delphix ticket number	Summary	Workaround/Comments
DLPX-75383	Refresh/Disable not working after failed delete operation	Force delete the VDB. Manually remove the mount point
DLPX-75385	Undo Refresh operation after a failed Refresh job leads to 502 Bad gateway and management stack restart	None
DLPX-67911	Failed enable operation leaving a mount point and next delete operation creating a stale file handle	Manually remove the mount point
CTCHANA-1599	Mount Point change not detected by the plugin may lead to multiple mounts	Mount point not in use can be removed manually
CTCHANA-1598	Plugin loses control in case of user cancel operation.	Manual cleanup can be done in case the plugin leaves any tenant database/files on the VM
CTCHANA-1586	Refresh after a Force disable on a VDB fails with 'mount not empty' error	Manually remove the mount before proceeding with the Refresh
CTCHANA-1565	Stop/Start operation on a VDB post failed post_refresh hook fails with inappropriate error	None
CTCHANA-1514	dSource tenant database can be accessed and written onto	None
CTCHANA-1610	'test_force_disable_enable_vdb' fails across multiple images	None

3.10.8.3.8 Release 6.0.0

Delphix ticket number	Summary	Workaround/Comments
DLPX-75383	Refresh/Disable not working after failed delete operation	Force delete the VDB. Manually remove the mount point
DLPX-75385	Undo Refresh operation after a failed Refresh job leads to 502 Bad gateway and management stack restart	None
DLPX-67911	Failed enable operation leaving a mount point and next delete operation creating a stale file handle	Manually remove the mount point
CTCHANA-1599	Mount Point change not detected by the plugin may lead to multiple mounts	Mount point not in use can be removed manually
CTCHANA-1598	Plugin loses control in case of user cancel operation.	Manual cleanup can be done in case the plugin leaves any tenant database/files on the VM
CTCHANA-1586	Refresh after a Force disable on a VDB fails with 'mount not empty' error	Manually remove the mount before proceeding with the Refresh
CTCHANA-1565	Stop/Start operation on a VDB post failed post_refresh hook fails with inappropriate error	None
CTCHANA-1514	Dsource tenant database can be accessed and written onto	None
CTCHANA-821	HANA SQL Port value should remain consistent for the life of VDB.	None
CTCHANA-1610	'test_force_disable_enable_vdb' fails across multiple images	None

3.10.8.3.9 Release 6.0.0

Delphix ticket number	Summary	Workaround/Comments
DLPX-75383	Refresh/Disable not working after failed delete operation	Force delete the VDB. Manually remove the mount point
DLPX-75385	Undo Refresh operation after a failed Refresh job leads to 502 Bad gateway and management stack restart	None
DLPX-67911	Failed enable operation leaving a mount point and next delete operation creating a stale file handle	Manually remove the mount point
CTCHANA-1599	Mount Point change not detected by the plugin may lead to multiple mounts	Mount point not in use can be removed manually
CTCHANA-1598	Plugin loses control in case of user cancel operation.	Manual cleanup can be done in case the plugin leaves any tenant database/files on the VM
CTCHANA-1586	Refresh after a Force disable on a VDB fails with 'mount not empty' error	Manually remove the mount before proceeding with the Refresh
CTCHANA-1565	Stop/Start operation on a VDB post failed post_refresh hook fails with inappropriate error	None
CTCHANA-1514	Dsource tenant database can be accessed and written onto	None
CTCHANA-821	HANA SQL Port value should remain consistent for the life of VDB.	None

3.10.8.3.10 Release 4.3.2

Delphix ticket number	Summary	Workaround/Comments
HANA-821	HANA SQL Port value should remain consistent for the life of VDB.	None
DLPX-44359	Job cancellation may fail.	None
CTCHANA-930	Upgrade from 3.4.5 to 3.5.0/3.5.1/4.0.0/4.1.1/4.1.2/4.2.0 needs a manual deletion of the existing tenant databases for disable/enable to work.	Drop the tenant database manually before enabling the VDB.

3.10.8.3.11 Release 4.3.1

Delphix Ticket Number	Summary	Workaround/Comments
HANA-821	HANA SQL Port value should remain consistent for the life of VDB.	None.
DLPX-44359	Job cancellation may fail.	None.
CTCHANA-930	Upgrade from 3.4.5 to 3.5.0/3.5.1/4.0.0/4.1.1/4.1.2/4.2.0 needs a manual deletion of the existing tenant databases for disable/enable to work.	Drop the tenant database manually before enabling the VDB.

3.10.8.3.12 Release 4.3.0

Delphix ticket number	Summary	Workaround/Comments
HANA-821	HANA SQL Port value should remain consistent for the life of VDB.	None.

Delphix ticket number	Summary	Workaround/Comments
DLPX-44359	Job cancellation may fail.	None.
CTCHANA-930	Upgrade from 3.4.5 to 3.5.0/3.5.1/4.0.0/4.1.1/4.1.2/4.2.0 needs a manual deletion of the existing tenant databases for disable/enable to work.	Drop the tenant database manually before enabling the VDB.

3.10.8.3.13 Release 4.2.0

Delphix ticket number	Summary	Workaround/Comments
HANA-821	HANA SQL Port value should remain consistent for the life of VDB.	None.
DLPX-44359	Job cancellation may fail.	None.
CTCHANA-930	Upgrade from 3.4.5 to 3.5.0/3.5.1/4.0.0/4.1.1/4.1.2/4.2.0 needs a manual deletion of the existing tenant databases for disable/enable to work.	Drop the tenant database manually before enabling the VDB.

3.10.8.3.14 Release 4.1.2

Delphix ticket number	Summary	Workaround/Comments
HANA-821	HANA SQL Port value should remain consistent for the life of VDB.	None.
DLPX-44359	Job cancellation may fail.	None.
CTCHANA-930	Upgrade from 3.4.5 to 3.5.0/3.5.1/4.0.0/4.1.1/4.1.2 needs a manual deletion of the existing tenant databases for disable/enable to work.	Drop the tenant database manually before enabling the VDB.

3.10.8.3.15 Release 4.1.1

Delphix ticket number	Summary	Workaround/Comments
HANA-821	HANA SQL Port value should remain consistent for the life of VDB.	None.
DLPX-44359	Job cancellation may fail.	None.
CTCHANA-930	Upgrade from 3.4.5 to 3.5.0/3.5.1/4.0.0/4.1.1 needs a manual deletion of the existing tenant databases for disable/enable to work.	Drop the tenant database manually before enabling the VDB.

3.10.9 Yugabyte release notes

- New features (Yugabyte) (see page 537)

3.10.9.1 New features (Yugabyte)

To deploy and configure the Yugabyte plugin, refer to Overview (see page 2266).

3.10.9.1.1 Release 1.1.0

3.10.9.1.1.1 Features

- Backup ingestion from AWS S3 location.

3.10.9.1.2 Release 1.0.0

3.10.9.1.2.1 Features

- Backup ingestion from NFS location.

> Future releases may add support for additional OS platforms and Delphix Continuous Data Engine versions.

4 Overview

4.1 What is Delphix?

Delphix is the organization behind the Delphix DevOps Data Platform, which combines enterprise-wide data coverage with data compliance to enable modern CI/CD workflows, accelerate the journey to the cloud, transform user experiences, and increase the adoption of disruptive AI technologies. The DevOps Data Platform is composed of various products, such as Delphix Continuous Data, Continuous Compliance, Data Control Tower, Delphix Compliance Services, and various integrations.

4.2 What is Delphix Continuous Data?

Delphix Continuous Data is a data management solution that provides the ability to securely copy and share data. The solution will ingest from a data source, create data copies, and flexibly manage them based on the organization's governance model. Delphix Continuous Data's copies are full, read-write capable database instances. This enables all data consumers, such as application, artificial intelligence, or machine learning teams, to create and manage previously expensive databases at a fraction of the total time, effort, and cost. These capabilities are often summarized as virtualization.

4.3 Deployment

Delphix Continuous Data is distributed through a virtual software appliance that can be installed on a wide variety of hypervisor platforms, such as VMware, Azure, Google Cloud Platform, and Amazon Web Services. Data copies are then served from a shared storage footprint called Delphix Storage. All of the day-to-day virtualization configuration and capabilities can be performed by logging in through the Administrator, Management, or Self-Service UI, and extended further through APIs or CLI tooling. The installed appliance is frequently referred to as the engine.

4.4 Environments

In addition to the engine itself, Delphix Continuous Data requires the availability of additional environments (or hosts) to manage the ingestion of a data source and provisioning of its copies. There are three primary environment types: Source, Target, and Staging.

- The **Source** environment is the originating data source location.
- The **Target** environment is where the data copies will be provisioned.
- The **Staging** environment is an optional environment where the sourced data can be pre-processed to minimize any performance concerns.

Frequently, Delphix Continuous Data will connect directly to the Staging environment, which allows administrators to maintain a controlled connection to the Source environment and update the Staging environment as needed.

Deciding which environments are connected to Delphix Continuous Data and how many are required depends on the chosen data source ingestion architecture. Learn more on the General architecture (see page 1014) page.

4.5 Data Source

Delphix Continuous Data ingests the data's files and logs to maintain an internal, condensed, and authoritative copy of the linked data source. This copy, along with all incremental updates, is referred to as the dSource. The dSource is kept in sync based on the configured ingestion method and user-defined policy.

Once created, Delphix Continuous Data maintains a Timeflow of the dSource which is a record of changes to the data. These changes are recorded in increments called Snapshots. From any snapshot within that Timeflow, a virtual database (VDB) can be instantly created or updated. Learn more about data sources in the dSource management (see page 1040) section.

4.6 Virtual Databases

Virtual Databases (VDBs) are the virtual copies of the data. VDBs are initially provisioned (aka created) from selecting a dSource's snapshot. Once provisioned, a VDB is an independent, read-write database, and changes made to the VDB by users or applications are written to new, compressed blocks in Delphix storage. Then, as the VDB is updated, their state is recorded within the Timeflow via Snapshots like dSources. At any time, VDBs can be provisioned from other VDBs and the data within a VDB can be refreshed from its parent VDB or dSource. Learn more about virtual databases in the Virtual Databases management (see page 1046) section.

4.7 Product information

4.7.1 Overview

Delphix Engines are delivered and maintained as a closed virtual software appliance. Like all virtual appliances, these engines are a tightly integrated combination of a special-purpose operating system and business logic; The **Continuous Data Engine** can be configured for **data virtualization**, while the **Continuous Compliance Engine** can be configured for **data masking**.

The product is delivered as a closed appliance, due to dependencies between software components within the virtual appliance that require end-to-end testing. Delphix does not provide administrative access to the operating system for any reason, including installing software, making customizations, or performing security scans. More details about the administrative model are provided in later sections of this page.

4.7.2 Packaging

Delphix software is delivered and maintained as a virtual software appliance. The mode of product delivery is dependent on the hosting hypervisor platform described in the table below:

Hypervisor Platform	Delivery Vehicle	Notes
VMware ESXi	OVA image	
Amazon AWS EC2	AMI image	Manual deployment to EC2
Amazon AWS EC2	Guest Machine Appliance	Amazon Marketplace
Microsoft Azure	Guest Machine Appliance	Azure Marketplace
Google Cloud Platform	Guest Machine Appliance	
Hyper V	Hyper image	

4.7.3 Updates, upgrades, and versions

Regardless of the initial packaging used to deploy Continuous Data or Continuous Compliance engines, updates are supplied as a single upgrade image of the new release, and the same image can be used for any prior release from which an upgrade is supported. This upgrade image delivers a completely new appliance, including both the operating system and business logic components.

The upgrade process retains prior configuration and user data to ensure that nothing is lost during the upgrade process, in addition to retaining a copy of the previous version of the components for automatic recovery in case an upgrade fails. Installed versions older than the prior version are automatically deleted during the upgrade.

There is no component patching of Continuous Data or Continuous Compliance engines; fixes are delivered in new versions of the software as a new software appliance. There is no management of patches involved and each engine version is a consistent, tested virtual appliance.

4.7.4 Administration

Administration of Continuous Data or Continuous Compliance engines happens through product interfaces only. These interfaces provide the proper configuration and testing of user infrastructure components, such as network addresses, storage, Domain Name Service (DNS) servers, authentication servers (LDAP), etc. The interfaces also control the business logic and control of the overall platform, including how data is used and provisioned by the system.

Although the special-purpose operating system may be accessed by Delphix Support and Engineering personnel for the purpose of diagnostics and problem remediation, there are no user-accessible interfaces at the operating system level. Users are not provided access to the underlying operating system nor can any custom software be installed on the appliance.

4.7.5 Customization

The product has several endpoints allowing customization for improved integration with user environments, local business workflow requirements, and alternative data sources.

Interface	Functional Area	Description
Data Plugins	Continuous Data	Delivered by Delphix Services or Integration Partners. These plugins allow for supporting additional data types including both structured and unstructured data.
Privilege Elevation Profiles	Continuous Data	Delivered by Delphix Services or Integration Partners. These customizations allow for the use of privilege mechanisms other than sudo on Linux and Unix target environments. Sudo is the product default.
Hook Scripts	Continuous Data	These user-managed scripts allow for custom business logic to be applied to Oracle and SQL Server data sources and virtual databases. The scripts are not integrated into the appliance but are referenced and invoked by the product during data operations.
Custom Algorithms	Continuous Compliance	Delivered by Delphix Services or Integration Partners. Custom algorithms provide specialized data transformations to secure or anonymize sensitive data.

In addition to these endpoints, Delphix provides a robust set of application programming interfaces (APIs) that enable business automation and fully integrated data operations into client workflows.

4.8 Delphix glossary

This glossary is your guide to exploring Delphix terms and definitions.

4.8.1 Products

Term	Definition
Delphix Continuous Data	Delphix Continuous Data is a Delphix product to deliver data on-demand to application developers and testers. Running as a virtual appliance, it is sometimes referred to as a Data Engine.
Delphix Continuous Data with Elastic Data	Elastic Data is a storage feature of Delphix Continuous Data that allows optimal data management to minimize costs through the use of block and elastic storage.
Delphix Continuous Compliance	Delphix Continuous Compliance is a Delphix product for discovering sensitive data and replacing it with realistic but fictitious data. Running as a virtual appliance, it is sometimes referred to as a Compliance Engine.
Data Control Tower	Data Control Tower (DCT) is a Delphix product that provides a data mesh to unify data governance, automation, and compliance across all applications and cloud platforms.

4.8.2 Interfaces

Term	Definition
Delphix Setup	Delphix Setup is the Delphix Continuous Data's user interface for system administrators to configure their engine settings, such as storage, support bundle, authentication, and network configurations.
Delphix Management	Delphix Management is Delphix Continuous Data's user interface for product administrators to manage their virtualized or masked datasets.
Delphix Self-Service	Delphix Self-Service is the Delphix Continuous Data user interface designed specifically for project teams, application developers, and testers to manage their virtualized datasets.
Command Line Interface (CLI)	Command Line Interface (CLI) is the engine's terminal interface which allows users to perform various administrative commands. This is not to be confused with the `dxtoolkit`.

Term	Definition
Delphix Download Portal	Delphix Download Portal[114]is the location where users download Delphix's products.
Delphix Support Portal	Delphix Support Portal[115]is the location where users receive support for Delphix's products.

4.8.3 Core concepts

Term	Definition
Virtualization	Virtualization describes the capability of producing a functioning database or filesystem copy that is lightweight and ephemeral.
Masking	Masking describes the capability of iterating through a dataset to identify all sensitive fields and replace them with desensitized values to eliminate risk in lower environments.
Data Source	Data Source is a database or unstructured files located in a user's environment. It generally describes ingestion sources and is typically located in a Source or Staging Environment.
Dataset	Dataset is an instance of any collection of data, such as VDB, dSource, vFiles, data source, or database. The dataset may or may not be managed by Delphix Continuous Data.
Transaction Log(s)	Transaction Log(s) are a general term for a database's record of all events or changes. Typically used for PITR and various ingestion mechanisms. Databases that support this mechanism might refer to these files differently, perhaps calling them something like “redo logs”, but the purpose is the same.
Point-in-Time or Point-in-Time Recovery	Point-in-Time is the recovery or roll forward of a virtual database up to a user specified point in time. This is made possible through the availability of Transaction Logs ingested via LogSync or another mechanism. This is synonymous with Point-in-Time Recovery (PITR).

114 https://download.delphix.com/
115 https://support.delphix.com/

4.8.4 Continuous Data

Term	Definition
Environment	Environment is a collection of hardware and/or software required to run a source or virtualized dataset. (For example, a Linux system running PostgreSQL.) This may either be a standalone instance or a cluster of instances. See Source, Staging, and Target Environments definitions for different environment types.
dSource	dSource is the copy of a source database's persistent data layer that Delphix Continuous Data uses to create and update virtual databases (VDBs). Based on the ingestion model and data source type, the dSource could be exposed through a mount point and interacts with a Staging Database Instance, Database, or Files. Consult the data source connector documentation for specific details.
Virtual Database (VDB)	Virtual Database (VDB) is a full read-and-write copy of the source data that is provisioned from either a dSource or another VDB. A VDB is provisioned and managed by Delphix Continuous Data.
Timeflow	Timeflow describes the timeline of data of a virtual database or dSource.
Snapshot	Snapshot represents the state of a dataset at a specific moment in time. They are used to create or refresh the same or another timeline.
Hooks	Hooks are mechanisms that allow the execution of custom operations at specific points in various processes like linking, provisioning, and managing virtual datasets.
Replication	Replication is the process of copying objects from one engine (the source) to another (the target).
Unstructured Files (vFiles)	Unstructured files (vFiles) are data stored in a filesystem that is not typically accessed by a Database Management System (DBMS) or similar software.

Term	Definition
Delphix Connector	Delphix Connector is a service that runs on the Windows Staging and Target Environments to enable communication with Delphix Continuous Data. The Delphix Connector should not be confused with Data Source Connectors.
Delphix Toolkit	Delphix Toolkit is a set of applications that are installed on the Source, Staging, and Target Environments which enables communication with Delphix Continuous Data. This should not be confused with `dxtoolkit`.
Delphix Storage	Delphix Storage is an underlying component in Delphix Continuous Data's architecture. Delphix Continuous Data uses this storage for ingesting the source data and storing the ZFS snapshots of the ingested data.
HostChecker	HostChecker is a standalone application that validates environments and hosts are configured correctly. HostChecker is available on the Delphix Download Portal.
Object(s)	Object(s) is a general term to refer to a Delphix Continuous Data entity, such as VDB, dSource, Snapshot, or Environment.

4.8.5 Architecture

Term	Definition
Host	Host is a single server within the environment collection. An environment is considered to be one or more hosts. For example, a RAC environment contains multiple hosts.
Staging Architecture	Staging Architecture is an ingestion strategy where a copy of the source data is recreated on an alternate environment where persistent storage is provided by Delphix Continuous Data.
Source Environment	Source Environment is the "production" or "golden" environment that contains the ideal database the user would like to virtualize in Delphix Continuous Data.

Term	Definition
Staging Environment	Staging Environment hosts the replica, copy, or backup of the database which Delphix Continuous Data will link and ingest.
Target Environment	Target Environment is the configured infrastructure, with available database binaries, in which virtual database copies will be hosted.
Privilege Elevation	Privilege Elevation is the optional functionality to provide a low-privileged environment user with sufficient permissions to perform the required ingestion and provisioning actions. This solution is not required, but suggested for production implementation of Delphix Continuous Data.
Installation	Installation is the location on source, staging, or target hosts that maintain the dataset's binaries and data. Often referred to as Repository or Dataset Home. Installations that are part of a Source and Staging Host will also be composed of one or more Databases. These Databases enable Delphix Continuous Data to identify the location from which a specific source dataset can be ingested. Often referred to as a Source Config.
Ingestion	Ingestion is the intake process by which Delphix configures the linking connection, obtains a copy of the backing data for the dSource, persists the data to Delphix Storage and takes a snapshot.
Direct Ingestion	Direct Ingestion is a form of intake into Delphix Storage where the persistent data of a dSource is provided directly to Delphix without any need for staging (example: from Oracle RMAN)
Staged Ingestion	Staged Ingestion is a form of intake into Delphix where the persistent data of a dSource must be recreated on an alternate staging host which has mounted Delphix Storage
Staging Pull	Staging Pull is an ingestion method where Delphix controls and executes the necessary operations to prepare the data content in the Staging Environment.
Staging Push	Staging Push is an ingestion method where the user (not Delphix) controls and executes the operations to prepare the data content in the Staging Environment.

4.8.6 Datasets

Term	Definition
Link	Link is the process of establishing a relationship between a data source and Delphix Continuous Data. After linking a data source, Delphix Continuous Data can import data periodically and manage it as it evolves over time. Also called Attach.
Unlink	Unlink is the process of disconnecting Delphix Continuous Data from the data source. Also called Detach.
Discovery	Discovery is a process that allows the connector to gather information about installation details in an environment. The database's discovery typically occurs as part of the initial environment setup process.
SnapSync	SnapSync is the standard process for importing data from a source into Delphix Continuous Data. An initial SnapSync is performed to create a dSource. Incremental SnapSyncs are performed to provide additional points in time to the dSource. Also, referred to as Sync.
LogSync	LogSync is an advanced process imports more granular source change data through Transaction Logs. This mechanism, often called Point-in-Time Recovery, allows for finer control of VDB point-in-time provision, refresh, or rollbacks.
Validated Sync	Validated Sync is a supportive process that executes cleanup and preparation activities on the Staging environment's database.
Resync	Resync is a Delphix operation by which the existing contents of the dSource are overwritten with a new full copy (ie a full sync) of the source data
Provision	Provision is an operation in which a new child dataset is created from a specific snapshot of the parent dataset (i.e. A VDB from a dSource, or a VDB from a VDB).
Refresh	Refresh is an operation where the virtual database is re-provisioned from its dSource's snapshot or a point in time. All current contents of the dataset are forfeited and the contents are fully restored to the state which has been captured in the chosen snapshot.

Term	Definition
Rewind	Rewind is an operation in which the contents of a specific virtual database are reset to the last point in its Timeflow. All current contents of the dataset are forfeited and the contents are fully restored to the state which has been captured in the chosen snapshot.
Enable	Enable is an operation in which a specific virtual database is deployed to a chosen Target environment. Specifically, the persisted virtual database is made available on the environment host via the appropriate network storage operation(s), and then visible and usable on that host. If there is an associated DBMS using that virtual database, the DBMS instance is also started.
Disable	Disable is an operation in which a virtual database is removed from the Target environment on which it is currently deployed. If there is an associated DBMS using the virtual database, that DBMS instance is also stopped.
Start	Start is an operation in which the associated DBMS is started. The virtual database must be in an ENABLED state on the Target environment for this operation to succeed
Stop	Stop is an operation in which an associated DBMS is stopped, but the virtual database is left in an ENABLED state on the Target environment.
Migrate	Migrate is an operation in which a specific virtual database is dissociated from one Target environment and moved to another Target environment on the same Delphix Continuous Data. A complete virtual database migration typically involves DISABLE, MIGRATE, and ENABLE operations. See REPLICATION to migrate datasets from one Delphix Continuous Data to another.
Virtual To Physical (V2P)	Virtual To Physical (V2P) is an operation to take a virtual database and provision it to a third-party database outside of Delphix Continuous Data. This is often used for disaster recovery purposes or as part of a performance test.

4.8.7 Data Source Connectors

Term	Definitions
Connector	Connector refers to the Delphix Continuous Data's data source connection mechanism. A connector enables Delphix Continuous Data functionality with a specific data source system or DBMS. See other connector types for specific details.
Standard Connector	Standard Connector are connectors that are built and supported by Delphix. They are included for free with a Delphix Continuous Data License Agreement.
Select Connector	Select Connector are connectors that are built and supported by Delphix but require a separate License Agreement.
Premium Connector	Premium Connector are connectors that are built and supported by a third party. They often require a separate License Agreement.
Plugin	Plugin is the software delivery framework for many Connectors. The user must upload the plugin into Delphix Continuous Data to install the associated Connector.

4.8.8 Transparent Data Encryption (TDE) definitions

Term	Definition
Artifact directory	Directory on the target system (not on Delphix Storage) which stores keys needed to support Delphix workflows on TDE-enabled vPDBs. It is located under the toolkit directory.
Auxiliary container database (CDB)	Provisioning an Oracle vPDB requires running recovery to bring the snapshotted datafiles into a consistent state. This needs to be done in the context of a container database, which is created on the target system. After the recovery is complete, the vPDB is unplugged and plugged into the target container, and the auxiliary container is deleted.

Term	Definition
Exported KeyFile	File located on the target Oracle host which contains keys that have been exported from the KeyStore. It is encrypted with a secret that is specified when it is exported. The exported KeyFile itself cannot be used as a KeyStore, but its contents can be imported into a new KeyStore.
Key rotation	Process for changing the master encryption key in the KeyStore via ADMINISTER KEY MANAGEMENT SET KEY. This does not remove the original key, rather it adds a new key to the wallet and future data will be encrypted with the new key.
KeyStore/wallet	File found on the Oracle host which stores the keys used to encrypt and decrypt the internal table keys in a database. Every KeyStore has a password that is set when it is first created and must be supplied for operations on it.
Parent KeyStore	KeyStore with the keys used to encrypt the dSource PDB files.
Target KeyStore	KeyStore for the target CDB into which the TDE-encrypted vPDB is plugged.

4.8.9 Users roles

User role	Role privileges	Group privileges
Reader	Access statistics on the dSource, VDB, or Snapshot such as usage, history, and space consumption	Access statistics on all dSources, VDBs, or Snapshots in the group such as usage, history, and space consumption
Provisioner	• Access statistics on the dSource, VDB, or Snapshot such as usage, history, and space consumption • Provision VDBs from owned dSources and VDBs	• Access statistics on all dSources, VDBs, or Snapshots in the group such as usage, history, and space consumption • Provision VDBs from all dSources and VDBs in the group

User role	Role privileges	Group privileges
Owner	• Provision VDBs from owned dSources and VDBs • Perform Virtual to Physical (V2P) from owned dSources • Access the same statistics as a Reader • Refresh or Rollback VDBs • Take Snapshots of dSources and VDBs	• Provision VDBs from all dSources and VDBs in the group • Refresh or Rollback all VDBs in the group • Snapshot all dSources and VDBs in the group • Perform Virtual to Physical (V2P) from owned dSources • Apply custom policies to dSources and VDBs • Create template policies for the group • Assign Owner privileges for dSources and VDBs • Access the same statistics as a Provisioner, Data Operator, or Reader
Data operator	• Access statistics on the dSource, VDB, or snapshot such as usage, history, and space consumption • Refresh or rollback VDBs	• Access statistics on all dSources, VDBs, or snapshots in the group such as usage, history, and space consumption • Refresh or rollback all VDBs in the group

4.8.10 User privileges

User type	Role privileges	Group privileges
sysadmin	Can perform typical system administration duties such as: modifying NTP, SNMP, SMTP settings; managing storage; downloading support logs for the Delphix Continuous Data Engine, and performing upgrades and patches. The sysadmin user launches the initial Delphix Setup configuration application and has access to the Command Line Interface (CLI).	Has privileges for storage, upgrades, network, etc.

4.8.11 Types of notification

Type	Notification
Event	Completion of some action in the Delphix Continuous Data Engine. Examples include user-initiated tasks such as snapshots or VDB provisioning, policy-based tasks, and background monitoring and maintenance tasks.
Alert	Caused by a single event on a Delphix Continuous Data Engine. Also known as a System Event, and viewable through the System Event Viewer. Examples include warnings on source/target environment settings, recoverable errors, or incorrect connection settings. Alert Levels: Informational, Warning, Critical
Fault	A persistent event on a Delphix Continuous Data Engine that remains until the issue is resolved. The fault may be marked resolved automatically or require that it be resolved manually. Selecting to Ignore a fault will also ignore future faults of that exact type against the same object. System faults describe states and configurations that may negatively impact the functionality of the Delphix Continuous Data Engine and which can only be resolved through active user intervention. Examples: Delphix Continuous Data Engine storage failure, communication failures between the Delphix Continuous Data Engine and a source or target environment/host Fault Levels: Warning, Critical

4.8.12 Delphix Self-Service terms

Term	Explanation
Administrator	Has full access to all report data and can configure and administer Delphix Self-Service. Additionally, can use the Delphix Continuous Data Engine to: • add/delete reports • add/delete users • change tunable settings • add/delete tags • create and assign data templates and containers

Term	Explanation
Bookmark	A logical reference to a point in time on a branch. You can use it as a point from which to fork new branches. It can also be the target of policies – for example, you can arrange to keep this bookmark for two years. Bookmarks are a way to mark and name a particular moment of data on a timeline. You can restore the active branch's timeline to the moment of data marked with a bookmark. You can also share bookmarks with other Delphix Self-Service users, which allows them to restore their own active branches to the moment of data in your container. The data represented by a bookmark is protected and will not be deleted until the bookmark is deleted.
Branches	Branches are task-specific groupings you can create within a data container. A branch is used to track a logical task and contains a timeline of the historical data for that task. As you work within your data container, you can create more branches overtime to run or complete separate tasks. Branches represent a logical sequence of activity, separate from the underlying data lineage. This is the main concept introduced in the core engine and forms the basis of many higher-level primitives. Branches: • Can have only one timeline active at any time • Can be user-visible (e.g. exported to a user target) or implementation (e.g. just a staging source to run a series of transformations)
Branch group/target group	A collection of multiple Branches that are treated as a single entity. The system can determine compatibility automatically, or a template can be used to create more complex orchestration.
Branch timeline	A dynamic point-in-time interface for user actions within the Branch. Common activities include re-setting Data Sources to run a test, refreshing the Data Container with the most current source data, and bookmarking data to share or track interesting moments of time along the branch timeline.
Data container	Consists of one or more Data Sources, such as databases, application binaries, or other application data. Allows users to: • Undo any changes to their application data in seconds or minutes • Have immediate access to any version of their data over the course of their project • Share their data with other people on their team, without needing to relinquish control of their own container • Refresh their data from production data without waiting for an overworked DBA

Term	Explanation
Data template	Created by the Engine Administrator, data templates consist of the data sources users need in order to manage their data playground and their testing and/or development environments. Data templates serve as the parent for a set of data containers that the administrator assigns to Delphix Self-Service users. Additionally, data templates enforce the boundaries for how data is shared. Data can only be shared directly with other users whose containers were created from the same parent data template.
Data User	Delphix Self-Service data users have access to production data provided in a data container. The data container provides these users with a playground in which to work with data using the Self-Service Toolbar.

4.9 Product icon reference

This topic illustrates the icons that appear on dSources and virtual databases (VDBs) in the Delphix Engine graphic user interface and describes the meaning of each, along with tips for clearing those that represent errors.

Icon	Description
	Environment
	Host
	Cluster
	dSource
	VDB
	Masked VDB
	vFiles
	Live Source
	Universal control for adding, creating users and objects

Icon	Description
	Provision a Virtual Database (VDB)
	Refresh a VDB
	Take a snapshot
	Rewind or roll back a VDB to a point in time
	Copy Virtual Database to a physical database (V2P)
	Open Log Sync to create a dataset from a point in time
	Filter control to narrow the results in a list view
	Toggle to expand the current state
	Toggle to collapse current state
	Timeflow view selector
	There is a warning fault associated with the dataset
	There is a critical fault associated with the dataset

5 Deployment

This section covers the necessary requirements to deploy the Delphix Engine across multiple different environments. Specific technical requirements for both on-premise and cloud deployments can be found under each platform topic (VMware, AWS, Azure, Google Cloud, etc.). This section covers the following topics:

- Accessing the Continuous Data Engine (see page 556)
- Standard deployment architecture (see page 559)
- Checklist of information required for installation and configuration (see page 560)
- Network connectivity requirements (see page 566)
- Installation and initial system configurations (see page 571)
- Validating host deployment with host Checker (see page 603)
- Deployment for VMware (see page 604)
- Deployment for KVM (see page 615)
- Deployment for Hyper-V (see page 619)
- Deployment for AWS EC2 (see page 625)
- Deployment for Microsoft Azure (see page 637)
- Deployment for Google cloud platform (see page 646)
- Deployment for OCI (see page 651)
- Deployment for IBM cloud (see page 661)
- Hotfix information (see page 670)

5.1 Accessing the Continuous Data Engine

5.1.1 Default users

After installation, the Delphix Continuous Data Engine will have two users; **admin** and **sysadmin**. Email address is a mandatory input for both the accounts. The user can create multiple sysadmin, admin users and other users with more restricted privileges. For more information, refer to the Managing System Administrators[116] section.

The below table lists the privileges of both admin and sysadmin users.

User	Initial password	Abilities and duties

116 https://cd.delphix.com/docs/latest/managing-system-administrators

sysadmin	sysadmin	• System administration duties such as:
		  • Modifying NTP, SNMP, SMTP settings
		  • Managing storage
		  • Downloading support logs for Delphix Continuous Data Engine
		  • Performing upgrades and patches
		• Launching the initial **Server Setup** configuration application
		• Accessing Command Line Interface (CLI)

admin	delphix	• Managing data objects, all collectively referred to as the Delphix Continuous Data Engine:
		  • dSources
		  • Virtual databases (VDBs)
		  • Users
		  • Groups
		  • Related policies and resources
		• Managing the Delphix Continuous Data Engine using either the browser-based **Engine Admin** application or the Command Line Interface (CLI).

5.1.2 Access methods

The three main interfaces to the Delphix Continuous Data Engine includes:

- Web browser (http or https)
- Command Line Interface (CLI) through SSH
- Delphix Web APIs

5.1.2.1 Accessing the Delphix Continuous Data Engine using a web browser

To access Delphix Continuous Data Engine administrator and the Delphix Management application, use **http://<engine IP or hostname>** or **https://<engine IP or hostname>**.

> In order to use your hostname, your Domain Name System (DNS) administrator must add the hostname and IP Address to your DNS system.

5.1.2.1.1 Delphix Management application

By default, the GUI login screen of the Delphix Continuous Data Engine will log you into the Delphix Management application, where the activities such as linking source databases and VDB provisioning takes place. This Delphix Management application contains a link which directs the user to the Delphix Server Setup application.

Once the user has logged into the Delphix Server Setup application, the user can switch to the Delphix Management application login by clicking **Management** in the upper-right corner of the Delphix Server Setup screen.

5.1.2.1.2 The Delphix Server Setup Graphical User Interface (GUI)

For accessing the Delphix Continuous Data Engine to perform system administrator actions such as adding storage, changing system-level parameters, etc., the user can switch the login screen to the Delphix Server Setup login by clicking **Server setup** on the default login screen.

If the user is logged into the Delphix Management application, the user can switch to the Delphix Server Setup login screen by clicking the username in the upper-right corner and selecting **Setup.**

5.1.2.2 Accessing the Delphix Continuous Data Engine using administrative Command Line Interface (CLI)

Delphix administrative CLI can be used in various scenarios such as:

- For users who prefer using CLI or automation over a graphical user interface.
- For users with limited bandwidth connections.

The user can access the Delphix Administrative CLI through a Secure Socket Shell (SSH) session. Similar to GUI, there are separate Administrative CLIs for the Delphix Continuous Data Engine administrator (admin) and Delphix System administrator (sysadmin) roles. The appropriate Administrative CLI will be automatically selected based on the role granted to the user.

Using your SSH client of choice, access the Delphix Continuous Data Engine in a method similar to the following:

```
ssh <user>@<engine IP or hostname>
```

Where <user> is a predefined user on the Delphix Continuous Data Engine, and <engine IP> is the IP address of a Delphix Continuous Data Engine.

```
ssh delphix_admin@dlpxengine1
```

All Delphix Continuous Data Engine functionality is available via the Administrative CLI. For more information on using the Delphix Administrative CLI, refer to the Command Line Interface Guide[117] section.

117 https://cd.delphix.com/docs/latest/command-line-interface-guide

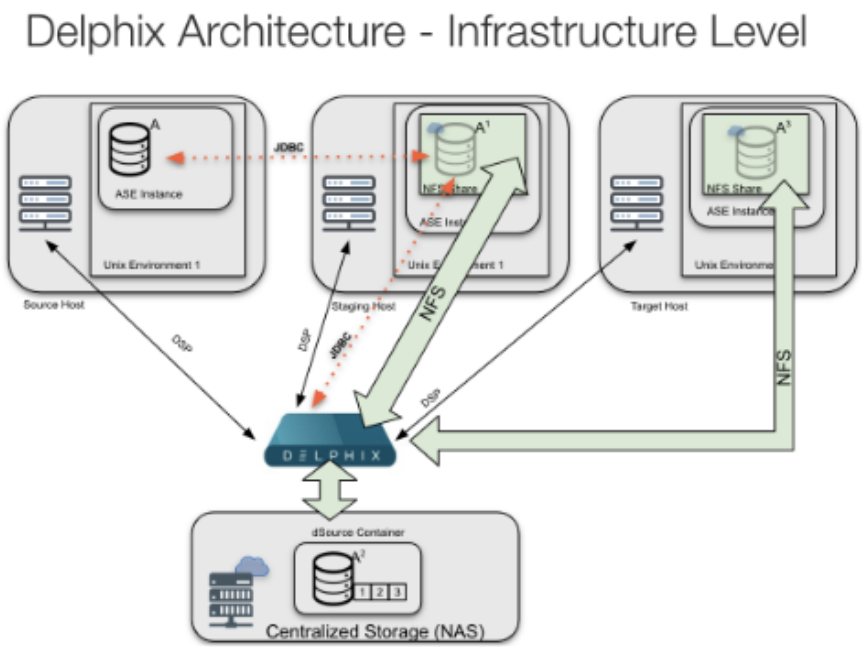
5.1.2.3 Accessing the Delphix Continuous Data Engine using the Delphix web API

Delphix offers a set of RESTful web service APIs with which a user can administer the Delphix Continuous Data Engine. Web API calls allow the user to create powerful, complex automation, often in coordination with other technologies like Jenkins, Puppet, Chef, and Ansible. All Delphix Continuous Data Engine functionality is available via the Web API. For more information refer to Delphix Web Service API Guide[118] section.

5.2 Standard deployment architecture

5.2.1 Overview

At a high level, Continuous Data functions to efficiently ingest data off of a source host by creating a compressed copy of that data, called a dSource, on the Delphix Engine. From there, the Delphix Engine can easily create multiple virtual databases (VDBs) on a target host with a marginal increase in storage. From an architecture perspective, the diagram below details how an engine interacts with an organizations infrastructure.



Architectural diagram of a common Delphix deployment (configurations vary depending on RDBMS platforms).

On an infrastructure level, Continuous Data relies on a Source, Staging, and Target Hosts, in addition to the Delphix Engine needing specific compute requirements (outlined in the platform-specific requirements section). Storage Requirements, like compute, vary by host type and are outlined in subsequent sections. Staging and Target host storage is provided from the Delphix Engine, which is mounted over the network similar to any Target host (NFS/iSCSI).

118 https://cd.delphix.com/docs/latest/web-services-api-guide

5.3 Checklist of information required for installation and configuration

5.3.1 Overview

This article describes the information required for the initial installation and configuration of the Delphix Engine.

The **InstallWorkbooks** articles list all the values that need to be captured, as well as sample values. Choose the workbook for the corresponding hypervisor. In addition to required values, the InstallWorkbooks also lists useful data to document as part of the installation.

- InstallWorkbook_Azure.xls [119]

[119] https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149500217?preview=%2F149500217%2F149500238%2FInstallWorkbook_Azure.xls

- InstallWorkbook_VMware.x… [120]
- InstallWorkbook_AWS.xls [121]

5.3.2 Hypervisor specific options

As a virtual appliance, Delphix supports a number of different hypervisors and configurations. The choice of hypervisor and network configuration will determine what configurations are needed for the first step in the installation of Delphix.

Attribute	Sample data	Used with the following hypervisor options			
		VMware with Static IP	VMware with DHCP	AWS	Azure

120 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149500217?preview=%2F149500217%2F149500232%2FInstallWorkbook_VMware.xls

121 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149500217?preview=%2F149500217%2F149500235%2FInstallWorkbook_AWS.xls

Attribute	Sample data	Used with the following hypervisor options			
Name	mydelphix1	*	*	*	*
Gateway IP	192.168.0.1	*	*	*	*
Root Volume Size	127GB	*	*	*	*
ESX Hostname	thx1138	*	*		
Domain	mydomain.com	*		*	*
DNS server IP(s), comma-separated	10.80.1.1, 10.80.1.2	*		*	*
Static IP and Subnet Mask in CIDR Notation	172.16.180.130/24	*			
Number of vSCSI Controllers	4	*			
VMDK/RDMs and Sizes	SAN 03- 3 TB, SAN04-3 TB, SAN05-3 TB & SAN06-3 TB	*			
Virtual Private Cloud (VPC) Name	vpc-673822a23			*	
Subnet	subnet-748391e26			*	
Auto-assign Public IP	disable			*	
EC2 instance type	r3.8xlarge			*	
Security Groups	sg-78sdg99879, sg-8798s77009			*	

Attribute	Sample data	Used with the following hypervisor options			
Root Volume Type	Magnetic			*	
Number of EBS volumes	4			*	
Size of EBS volumes	500G			*	
Type of EBS volumes	Provisioned IOPS SSD			*	
IOPS set on each volume	5,000			*	
Encryption on EBS volumes	TRUE			*	
Virtual Network (vnet)					*
Subnet					*
Network Interface w/ Static Public IP					*
VM Size	Example: D8s_v3				*
Network Security Group					*
OS Disk	127 GB - Premium SSD				*
Data Disks	Premium SSDs				*

5.3.3 Optional information required for initial configuration

Class	Attribute	Sample data
Time	NTP Servers (Highly Recommended)	172.16.180.10, 172.16.180.11
	Timezone	US/Pacific
Serviceability (Highly Recommended)		
	Enable Phone Home? (Highly Recommended)	Yes
	Web Proxy Server IP Address	192.168.1.200
	Web Proxy Server Port	8080
	Web Proxy Server Username (If Required)	user
	Web Proxy Server Password (If Required)	***
	Enable SMTP Server (Highly Recommended)	Yes
	SMTP Server IP Address	192.168.22.22
	SMTP Server Port Number	25
	SMTP Server From Email Address	delphix@hostname.mydomain.com[122]
	SMTP Server User (Optional)	user

122 mailto:delphix@hostname.mydomain.com

Class	Attribute	Sample data
	SMTP Server Password (Optional)	***
	SMTP Use TLS Authentication? (Optional, Default=No)	No
	SMTP Send Timeout (Optional, Default=60)	60
Authentication (Highly Recommended)		
	LDAP Server IP Address	192.168.7.7
	LDAP Server Port	389 (636 for SSL)
	Protect LDAP traffic with SSL/ TLS (true | false)	false
	LDAP SSL Authentication (SIMPLE | DIGEST_MD5)	SIMPLE
SNMP Integration		
	SNMP Server IP Address	192.168.77.156
	SNMP Server Port	162
	SNMP Community	mycommunity
	SNMP Use INFORM Instead of TRAP	false
	SNMP Severity Level (INFORMATIONAL, WARNING, CRITICAL)	CRITICAL
Splunk integration (Highly recommended)		

Class	Attribute	Sample data
	Splunk Server IP address	192.168.8.8
	Splunk Server HEC Port Number	8088
	Splunk HEC Token	12345678-1234-1234-1234-1234567890AB
	Index Name for Events	delphix_events
	Index Name for Metrics	delphix_metrics
ADMIN		
	Email Address	u12345@mydomain.com[123]
SYSADMIN		
	Email Address	u12345@mydomain.com[124]

5.4 Network connectivity requirements

5.4.1 Overview

This topic covers the general network and connectivity requirements for the Delphix Engine, including connection requirements, port allocation, and firewall and Intrusion Detection System (IDS) considerations. For platform-specific network and connectivity requirements, see relevant topics under the **Requirements** article for each platform.

123 mailto:u12345@mydomain.com
124 mailto:u12345@mydomain.com

5.4.2 General outbound from the Delphix engine port allocation

Protocol	Port Numbers	Use
TCP	25	Connection to a local SMTP server for sending email
TCP/UDP	53	Connections to local DNS servers
UDP	123	Connection to an NTP server
UDP	162	Sending SNMP TRAP messages to an SNMP Manager
TCP	443	HTTPS connections from the Delphix Engine to the Delphix Support upload server
TCP/UDP	636	Secure connections to an LDAP server
TCP	8415	Connections to a Delphix replication target (see Configuring Replication (see page 2381))
TCP	50001	Connections to source and target environments for network performance tests

5.4.3 General inbound to the Delphix Engine port allocation

Protocol	Port Number	Use
TCP	22	SSH connections to the Delphix Engine

Protocol	Port Number	Use
TCP	80	HTTP connections to the Delphix GUI
UDP	161	Messages from an SNMP Manager to the Delphix Engine
TCP	443	HTTPS connections to the Delphix Management Application
TCP	8415	Delphix Session Protocol connections from all DSP-based network services including Replication, SnapSync for Oracle, V2P, and the Delphix Connector.
TCP	50001	Connections from source and target environments for network performance tests via the Delphix CLI
TCP	54046	Connections from Source and Target Environments to the Engine When NFS Encryption is enabled

5.4.4 Firewalls and intrusion detection systems (IDS)

Production databases on source environments (for dSources) are often separated from the non-production environment by firewalls. Firewalls can add milliseconds to the latency between servers. Accordingly, for best performance, there should be no firewalls between the Delphix Engine and the virtual database (VDB) target environments. If the Delphix Engine is separated from a source environment by a firewall, the firewall must be configured to permit network connections between the Delphix Engine and the source environments for the application protocols (ports) listed above.

Intrusion detection systems (IDSs) should also be made permissive to the Delphix Engine deployment. IDSs should be made aware of the anticipated high volumes of data transfer between dSources and the Delphix Engine.

5.4.5 Setting up network access to the Delphix engine

5.4.5.1 Overview

This article outlines the procedure for setting up network access to the Delphix Engine. Follow the initial installation instructions in Installing the Delphix engine (see page 571) before addressing this procedure.

5.4.5.2 Procedure

1. Power on the Delphix Engine and open the Console.
2. Wait for the Delphix Management Service and Delphix Boot Service to come online. This might take up to 10 minutes during the first boot. Wait for the large orange box to turn green.
3. Press any key to access the sysadmin console.
4. Enter `sysadmin` for the username and the password.
5. A description of available network settings and instructions for editing will be shown.

```
Delphix Engine Network Setup

To access the system setup through the browser, the system must first be
configured for networking in your
environment. From here, you can configure the primary interface, DNS, hostname,
and default route. When DHCP is
configured, all other properties are derived from DHCP settings.

To see the current settings, run "get." To change a property, run "set
<property>=<value>." To commit your changes,
run "commit." To exit this setup and return to the standard CLI, run "discard."

    defaultRoute    IP address of the gateway for the default route -- for
 example, "1.2.3.4."

    dhcp            Boolean value indicating whether DHCP should be used for
 the primary interface. Setting this value
                    to "true" will cause all other properties (address,
hostname, and DNS) to be derived from the DHCP
                    response

    dnsDomain       DNS Domain -- for example, "delphix.com"

    dnsServers      DNS server(s) as a list of IP addresses -- for example,
"1.2.3.4,5.6.7.8."
```

```
    hostname        Canonical system hostname, used in alert and other logs --
for example, "myserver"

    primaryAddress  Static address for the primary interface in CIDR notation
-- for example, "1.2.3.4/22"

Current settings:

    defaultRoute: 192.168.1.1
    dhcp: false
    dnsDomain: example.com
    dnsServers: 192.168.1.1
    hostname: Delphix
    primaryAddress: 192.168.1.100/24
```

6. Configure the `hostname`. If using DHCP, this step can be skipped.

```
delphix network setup update *> set hostname=<hostname>
```

Use the same hostname entered during the server installation.

7. Configure DNS. If using DHCP, this step can be skipped.

```
delphix network setup update *> set dnsDomain=<domain>
delphix network setup update *> set dnsServers=<server1-ip>[,<server2-ip>,...]
```

8. Configure either a static or DHCP address:

DHCP configuration

```
delphix network setup update *> set dhcp=true
```

Static configuration

```
delphix network setup update *> set dhcp=false
delphix network setup update *> set primaryAddress=<address>/<prefix-len>
```

> The static IP address must be specified in CIDR notation (for example, `192.168.1.2/24`).

9. Configure a default gateway. If using DHCP, this step can be skipped.

```
delphix network setup update *> set defaultRoute=<gateway-ip>
```

10. Commit the changes. Use the get command prior to committing to verify the desired configuration.

```
delphix network setup update *> commit
Successfully committed network settings. Further setup can be done through the
browser at:

        http://<address>

Type 'exit' to disconnect, or any other commands to continue using the CLI.
```

11. Check that the Delphix Engine can now be accessed through a Web browser by navigating to the displayed IP address, or hostname if using DNS.

12. Exit setup.

1.
```
delphix> exit
```

5.5 Installation and initial system configurations

This section covers the following topics:

- Initial setup (see page 571)
- Customizing the Delphix engine system settings (see page 599)
- Installing an OVA or AMI (see page 602)

5.5.1 Initial setup

5.5.1.1 Overview

Delphix runs as a virtual appliance deployed in various types of platforms, as outlined in these **Deployment** articles. When first logging into an instance, the setup wizard will help with initial configurations for network, storage, authentication, and more. This article describes each step in the setup process and will outline and the different options available.

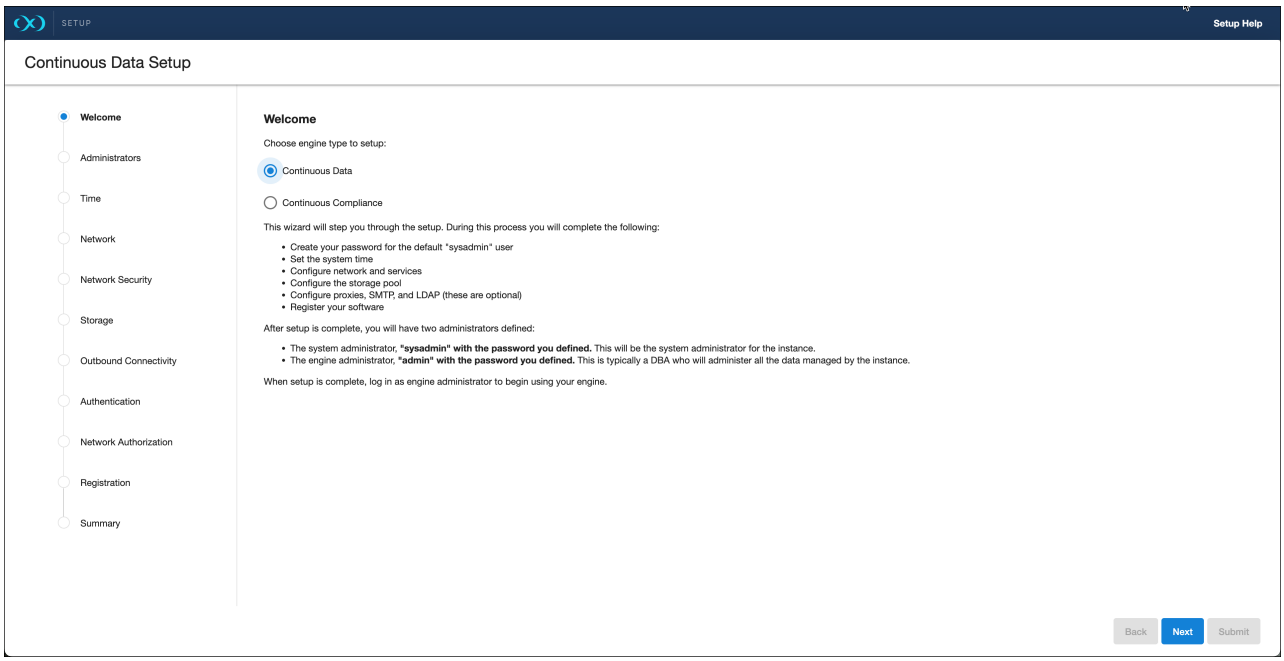
When first connecting to the Delphix Engine via any supported browser, enter the default sysadmin login; **username**: sysadmin, **password**: sysadmin. On the first login, there will be a prompt to change the default password for security purposes.

> A login failure issue could occur if the Delphix Engine clock is not in sync with the IdP (identity provider) clock. To resolve the issue, either use the NTP (network time) clock or set up the skew time property in the SSO (Single Sign-On) configuration.

> Anti-virus programs can impact both performance and operation. Delphix recommends anti-virus scanning exclude folders where Delphix files are maintained, in addition to the normal exclusions put in place for MSSQL operation. This is largely applicable to Windows machines.

5.5.1.2 Welcome

The Welcome section asks users to select the engine type being setup, whether Continuous Data or Continuous Compliance. This article explains the Continuous Data setup, the Continuous Compliance version can be found at First time setup[125].



5.5.1.3 Administrators

Each **Continuous Data** engine has two default accounts:

- **System Administrator**: `sysadmin` with a user-defined password. This will be the system administrator for the instance.
- **Engine Administrator**: `admin` with a user-defined password. This is typically a DBA who will administer all the data managed by the instance.

Provide an email address and password for both users in the Administrator section.

Each **Continuous Compliance** engine has this default account:

- **Masking Administrator**: `admin` with a user-defined password. This will be the user responsible for setting up other users and handling administrative actions.

125 https://masking.delphix.com/docs/latest/first-time-setup

5.5.1.4 Time

The Delphix engine leverages its time setting to determine policies and actions that take place within the application. Manually set the time or choose from an NTP server, an explanation of these options are shown below.

Option	Notes
Set NTP Server (recommended)	After selecting this option, select an NTP server from the list, or select **Add NTP Server** to manually enter one or more server(s). When configuring a Delphix Engine on VMware, be sure to configure the NTP client on the host to use the same servers that are entered here.
Manually Select Time and Date	Select the **Use browser time and date** option to set the system time, or select the date and time by using the calendar and clock icons. If this option is selected, the date and time will persist as the local time, even if time zones are changed.

5.5.1.5 Network

The initial network configuration will be pre-populated based on the deployment platform used for Delphix. For VMware deployments, Delphix defaults to the VMXNET3 network adapter.

Select **Settings** for each Network interface to manage the following options:

Option	Notes
DHCP or Static network addressing	For Static addressing, enter an IP Address and Subnet Mask. The static IP address must be specified in CIDR notation (for example, 192.168.1.2/24).
Jumbo Frames	This setting is highly recommended. VMXNET3 supports Ethernet jumbo frames, which can be used to maximize throughput and minimize CPU utilization.
Routing	A default gateway will be specified in this section.
DNS Services	Enter a DNS Domain Name and DNS Server to be used for this engine.

5.5.1.6 Network security

Delphix installs certificates signed by the engine's Certificate Authority. Users have the ability to manage their own certificates for HTTPS and DSP (Delphix Session Protocol) connections to and from the Delphix Engine. You can add or modify certificates and certificate signing requests (CSRs) via the **...** option.

When you update the Certificate Authority certificate, your HTTPS and DSP certificates will be automatically updated.

For more information please refer to Certificate management[126] in the Security section of this documentation.

5.5.1.7 Storage

5.5.1.7.1 Storage for engines backed by block devices

The Delphix Continuous Data Engine automatically discovers and displays storage devices. For each device, confirm that **Usage Assignment** is set to Data.

You can associate additional storage devices with the Continuous Data Engine after initial configuration, as there are two options for storage disk usage assignment:

1. **Enabled**: Once you set the storage unit assignment for a disk and save the configuration, it cannot be changed again.
2. **Unassigned**: These are disks being held for later use.

Configure at least four disks for the storage of user data. This makes the Continuous Data Engine storage manager function more efficient, since duplicated metadata can be distributed across multiple disks.

5.5.1.7.2 Storage for Elastic Data Engines backed by object storage

- To configure object storage, select from:
  - Azure Blob storage[127]
  - AWS object storage[128]
  - GCP object storage (see page 588)
  - OCI object storage[129]
  - Other S3 compatible object storage[130]
- If utilizing on-premise object storage solutions, opt for Other S3 compatible object storage[131].

126 https://cd.delphix.com/docs/latest/certificate-management
127 https://cd.delphix.com/docs/latest/azure-object-storage-setup
128 https://cd.delphix.com/docs/latest/aws-object-storage-setup
129 https://cd.delphix.com/docs/latest/oci-object-storage-setup
130 https://cd.delphix.com/docs/latest/other-s3-compatible-object-storage-setup
131 https://cd.delphix.com/docs/latest/other-s3-compatible-object-storage-setup

5.5.1.8 Outbound connectivity

5.5.1.8.1 Web proxy

If a Web Proxy Server is necessary for your environment, select **Configure web proxy**, then enter the hostname and credentials for that server.

5.5.1.8.2 Phone home (service)

The support and phone home bundles contain metadata from the Delphix Engine, but do not include the customer data that has been ingested into the Continuous Data Engine. Redaction of known PII data (i.e., names and email addresses for Delphix users) is done on-engine, before and after bundles are uploaded to Delphix, to ensure the latest redaction rules are applied to each bundle without requiring the engine to be continually upgraded. There may be some limited environment data in the bundle (i.e., IP addresses and database names) that are needed for debugging purposes. Support bundles are automatically deleted within 30 days after the support case is closed, or 30 days after upload, whichever comes later.

5.5.1.8.3 Enabling/disabling phone home

Enabling this service sends information to Delphix periodically over HTTPS (SSL). This data is securely managed by the internal team for product analysis and improvements. This feature requires a connection to the internet and will use the Web Proxy Server configuration.

Perform the following steps to enable/disable phone home.

1. Login to the Delphix Continuous Data Engine setup using the sysadmin credentials.
2. From the Outbound Connectivity widget, click **Modify**.
3. To enable phone home, select the checkbox before the **Enable phone home service** option. If enabled, this service will automatically send a stream of anonymous, non-personal metadata describing user interaction with the product's user interface.
4. To disable, deselect the checkbox before the **Enable phone home service** option.
5. Click **Save** to save for the settings.

5.5.1.8.4 User-click analytics

User-click analytics is a lightweight method to capture how users interact with Delphix product UIs, allowing Delphix to collect browser-based, user-click data. Delphix does not collect, transmit, or store any personally identifiable information (PII) such as email addresses, IP addresses, usernames, etc.

5.5.1.8.5 SMTP server

Select **Use an SMTP Server** and enter the server name or IP address to enable email notifications for events and alerts. When a critical fault occurs with the Delphix Engine, it will automatically send an email alert to the

admin user. Make sure to configure the SMTP server so that alert emails can be sent to this user. See **System Faults** for more information.

5.5.1.9 Authentication

On the Authentication page, configure authentication protocols such as LDAP and SAML/SSO. See User and authentication management[132] for further details.

5.5.1.9.1 LDAP

To avoid configuration issues, consult with the lightweight directory access protocol (LDAP) administrator before attempting to set up LDAP authentication of users for the Delphix engine. When configuring LDAP, provide an LDAP Server. Two authentication methods are currently supported: **SIMPLE** and **DIGEST_MD5**.[133]

Select to **Protect LDAP traffic with SSL/TLS** if desired. This option requires an import of the server certificate. If LDAP has been set up as an authentication service for the Delphix Engine, add new users with LDAP as their authentication mechanism. For more information, see the User groups[134] article.

5.5.1.9.2 SAML/SSO

To enable SAML/SSO, there are two properties to set:

1. **Audience Restriction**: The audience restriction must be set to the entity id configured in the Delphix Server via the Delphix Setup. Its default value is **https://Delphix Server ID**, where **Delphix Server ID** is a 36-character hexadecimal string of the form xxxxxxxx-xxxx-xxxx-xxxxxxxxxxxx.
    a. See Determining the Delphix server ID and host name[135] for more on the Delphix Server ID. If the Continuous Data Engine does not exist or is unreachable, enter a temporary value (such as `delphix-sp-id`), which must later be replaced by the actual Delphix Server ID.
2. **IdP Metadata**: an XML document that must be exported from the application created in the IdP. Paste its contents into the provided field.

5.5.1.10 Kerberos

The Kerberos page allows for Kerberos authentication to communicate between hosts connected with Continuous Data. Enabling this option will allow Kerberos key-based authentication when adding new environments to Continuous Data.

1. **Realm**: the domain over which a Kerberos authentication server has the authority to authenticate a user, host, or service.
2. **Principal**: a unique identity to which Kerberos can assign tickets.

---

132 https://cd.delphix.com/docs/latest/user-and-authentication-management
133 https://delphixdocs.atlassian.net/wiki/spaces/CC/pages/9962775/Introduction+to+Delphix+Masking
134 https://cd.delphix.com/docs/latest/users-and-groups
135 https://cd.delphix.com/docs/latest/determining-the-delphix-server-id-and-host-name

3. **Keytab**: a file containing pairs of Kerberos principals and encrypted keys (which are derived from the Kerberos password).

5.5.1.11 Registration

As described in Registration management[136], registration allows Delphix Support to access the engine, properly diagnose, and identify any issues during support cases. If the Continuous Data Engine has access to the internet, auto-register the engine with Delphix Support credentials in the **Online Registration** section.

If external connectivity is not immediately available, perform the manual registration.

1. Copy the Delphix Engine registration code displayed.
2. Transfer the Delphix Engine's registration code to a workstation with access to the external network Internet. For example, e-mail the registration code to an externally accessible email account.
3. On a machine with access to the internet, use a browser to navigate to the **Delphix Registration Portal** at http://register.delphix.com[137].
4. Log in with Delphix support credentials.
5. Paste the Registration Code.
6. Click **Register**.[138]

> The Continuous Data Engine will work without registration, but it is recommended to register each engine as part of the setup. Failing to register the Continuous Data Engine will impact its supportability.

5.5.1.12 Summary

The final Summary section will enable a review of the configurations for each page in the setup tutorial. Confirm that everything looks correct, and click submit to complete the setup.

5.5.1.12.1 After Setup

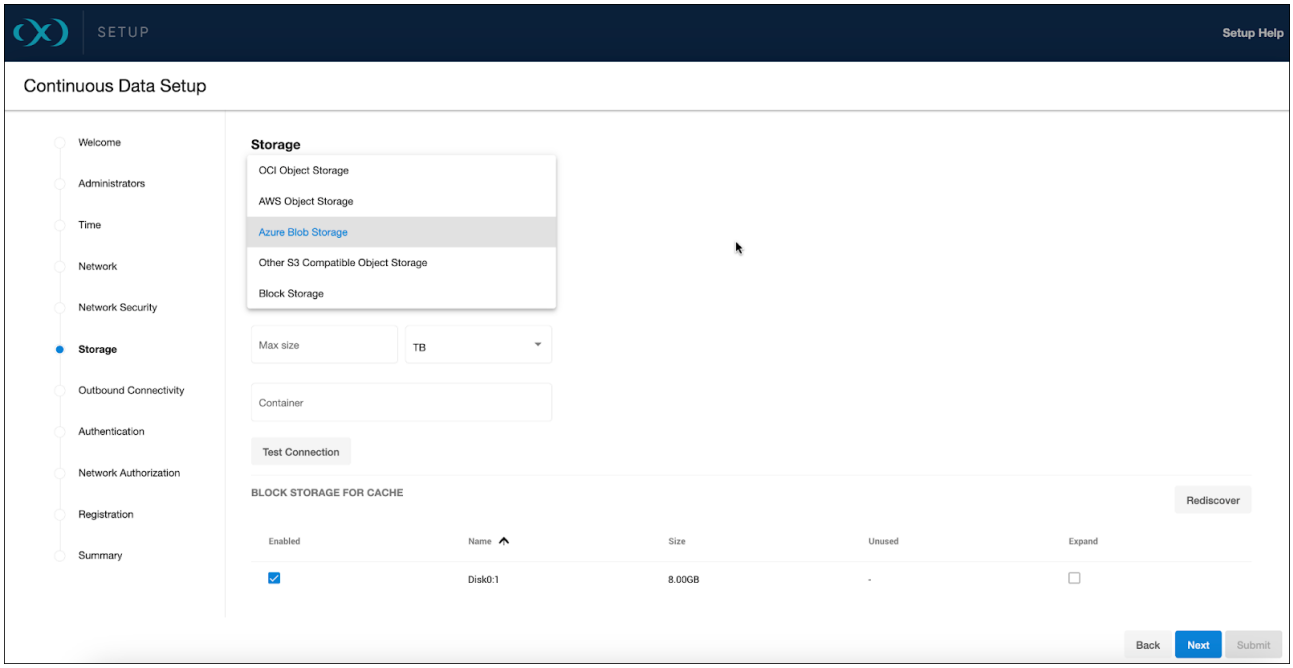
- After the configuration is complete, the Continuous Data Engine will restart and launch the browser-based Delphix Management application.
- After the Delphix Management application launches, the admin can log in using the initial default username **admin** and the initial default password **Delphix**. On the first login, there will be a prompt to change the initial password.

136 https://cd.delphix.com/docs/latest/registration-management
137 https://register.delphix.com/
138 https://delphixdocs.atlassian.net/wiki/spaces/CC/pages/9962775/Introduction+to+Delphix+Masking

5.5.1.13 Azure object storage setup

From the **Storage** setup page, select **Azure Blob Storage** from the **Storage Type for Data** dropdown menu. For specific block storage requirements, see the **Storage configuration** section of Deployment for Microsoft Azure[139].

> Delphix is compatible solely with the **Online access** tier (hot tier[140] of Standard GPv2) for Azure-Blob[141].

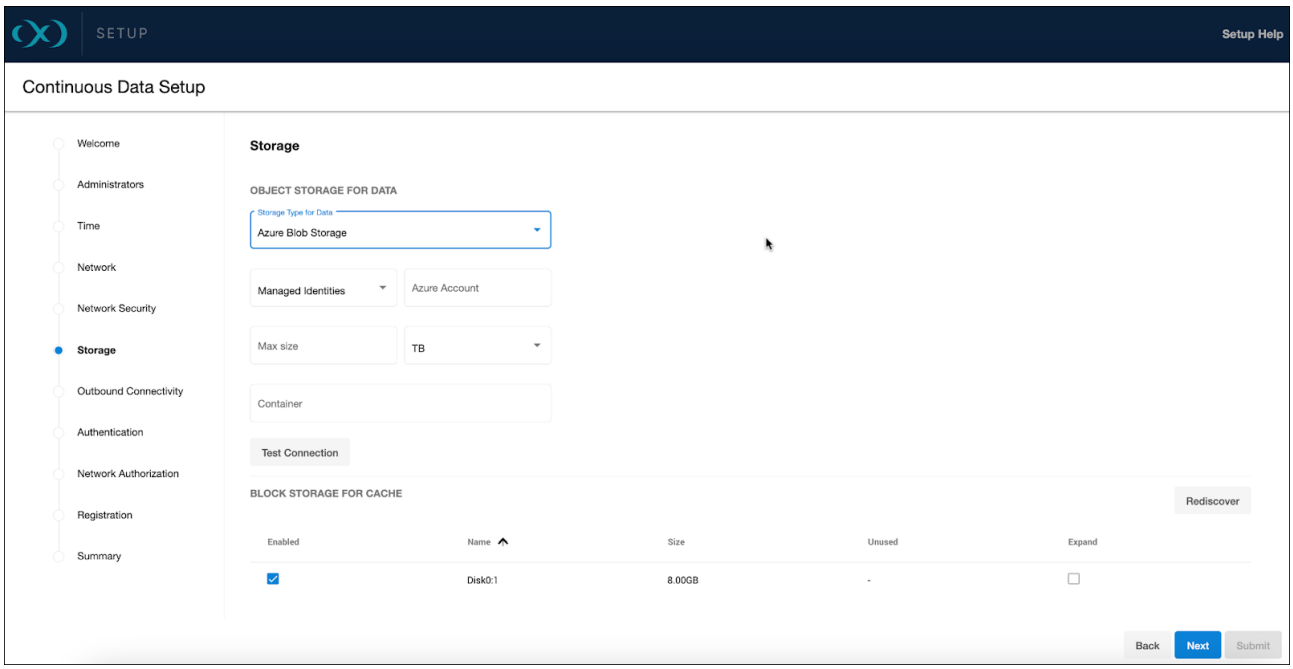


5.5.1.13.1 Access method

You can choose between Managed Identities[142], including System-assigned and User-assigned types, or the Access Key[143] method. Make sure you do not have versioning on your blob/bucket, otherwise deleted space will not be reclaimed since it will be held by the versioning logic of the cloud vendor.

139 https://cd.delphix.com/docs/latest/deployment-for-microsoft-azure
140 https://learn.microsoft.com/en-us/azure/storage/blobs/access-tiers-overview
141 https://learn.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction#storage-accounts
142 https://learn.microsoft.com/en-us/azure/active-directory/managed-identities-azure-resources/overview
143 https://learn.microsoft.com/en-us/azure/storage/common/storage-account-keys-manage?toc=%2Fazure%2Fstorage%2Fblobs%2Ftoc.json&bc=%2Fazure%2Fstorage%2Fblobs%2Fbreadcrumb%2Ftoc.json&tabs=azure-portal

> Managed Identities are recommended to minimize security risks and maintenance related to access keys.



5.5.1.13.2 Storage capacity

Enter the desired data storage capacity. Azure Blob's billing is based on actual space used, not provisioned. Therefore, if you allocate 10TB but only use 1TB, charges will apply only to the used space. The specified number also serves as a quota to limit the blob storage from expanding beyond an unforeseen point. This quota can be modified later through the sysadmin login or Setup UI, but not reduced below the current usage by the Delphix Engine.

5.5.1.13.3 Container setup

Input your storage container's name and test the connection to confirm VM accessibility to the container.

5.5.1.13.4 Block storage for cache

Block devices are utilized as cache, reducing latencies for frequently read data and acting as temporary storage for synchronous writes before the writes are persisted to object storage. Size the cache based on your knowledge of the frequently accessed data, adding an extra 10% for overhead, or start with a cache size at 50% of all dSources to be added to the engine.

> For detailed block storage requirements, see the **General storage** section of Deployment for Microsoft Azure[144].

To support the engine's throughput, set up the disks accordingly. For example, Ultra disks are recommended for their customizable high IOPS and throughput, even at smaller sizes.

- Ultra disks can be configured to have high IOPS and throughput at relatively small sizes. At 300 IOPS per GiB ratio, a 256 GB volume can have 76,800 IOPS and 4000 MB/s throughput.
- Premium SSD disks can have high performance, though they need to be much larger. For example, a P80 32TiB volume has 20K IOPS and 900 MB/s. The new Premium SSD v2 disks have better IOPS per GiB ratio though they are not available in all regions. SSD v2 is currently untested due to this unavailability.

5.5.1.13.5 Storage account and permission setup

5.5.1.13.5.1 Storage container

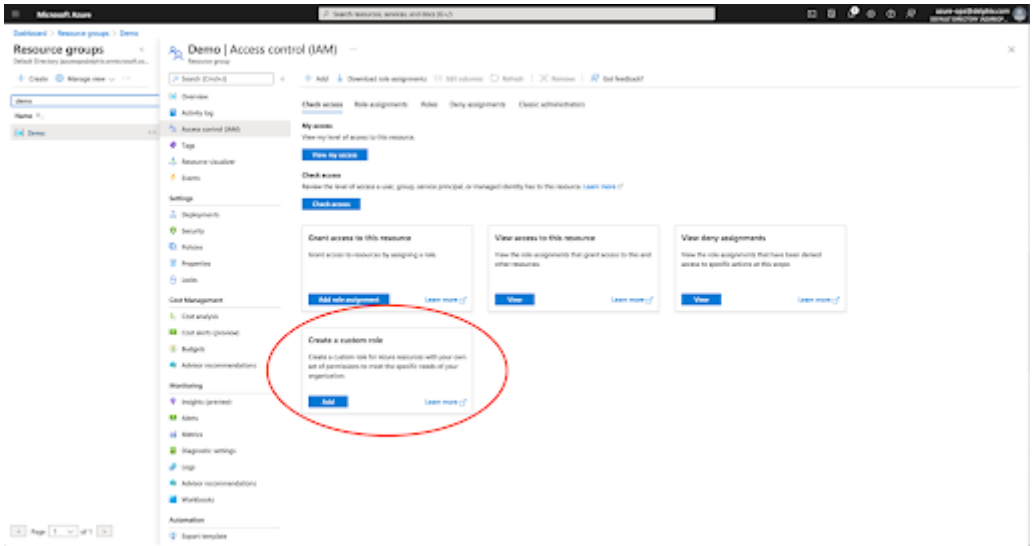
Create a dedicated Storage Account for your Delphix Engine by navigating to Storage Accounts and clicking **Create**. Remember that Azure imposes ingress and egress limitations per storage account, which are detailed in their documentation: https://docs.microsoft.com/en-us/azure/storage/common/scalability-targets-standard-account

Next, create a Storage Container within the Storage Account to be used for your Delphix Engine by navigating to the chosen **Storage Account**, selecting **Containers** under **Data Storage**, and clicking **+ Container**. Note the names of the Storage Container, Storage Account, and Resource Group.

5.5.1.13.5.2 Managed role

Forge a custom role to grant access to the storage account, which will be assigned to your virtual machine post-creation. Navigate to the Resource Group containing your Storage Container and select **Access Control (IAM)**.

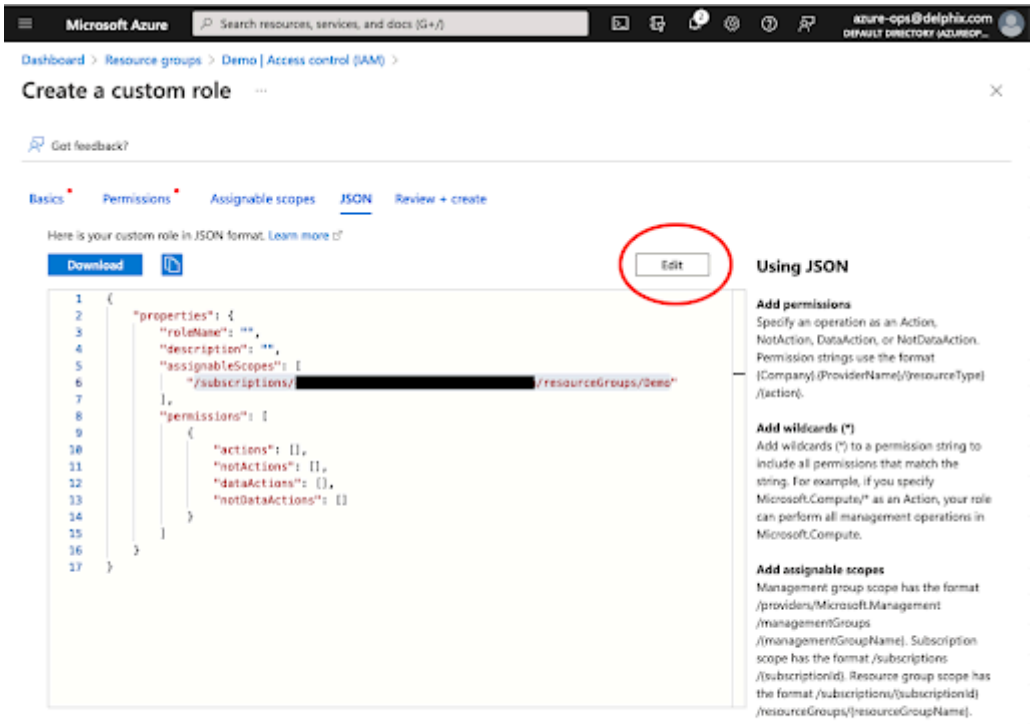
144 https://cd.delphix.com/docs/latest/deployment-for-microsoft-azure#Storage-configuration



Then **Create a custom role** with a minimum scope of the storage account that your Delphix Engine will be using:

Permission Type	Permission
Action	Microsoft.Storage/storageAccounts/blobServices/containers/read
DataAction	Microsoft.Storage/storageAccounts/blobServices/containers/blobs/write
DataAction	Microsoft.Storage/storageAccounts/blobServices/containers/blobs/read
DataAction	Microsoft.Storage/storageAccounts/blobServices/containers/blobs/move/action
DataAction	Microsoft.Storage/storageAccounts/blobServices/containers/blobs/add/action
DataAction	Microsoft.Storage/storageAccounts/blobServices/containers/blobs/delete

For the JSON view, navigate to the role's JSON view, click **Edit**, and replace the bolded sections with your specific values. Save the changes afterwards.



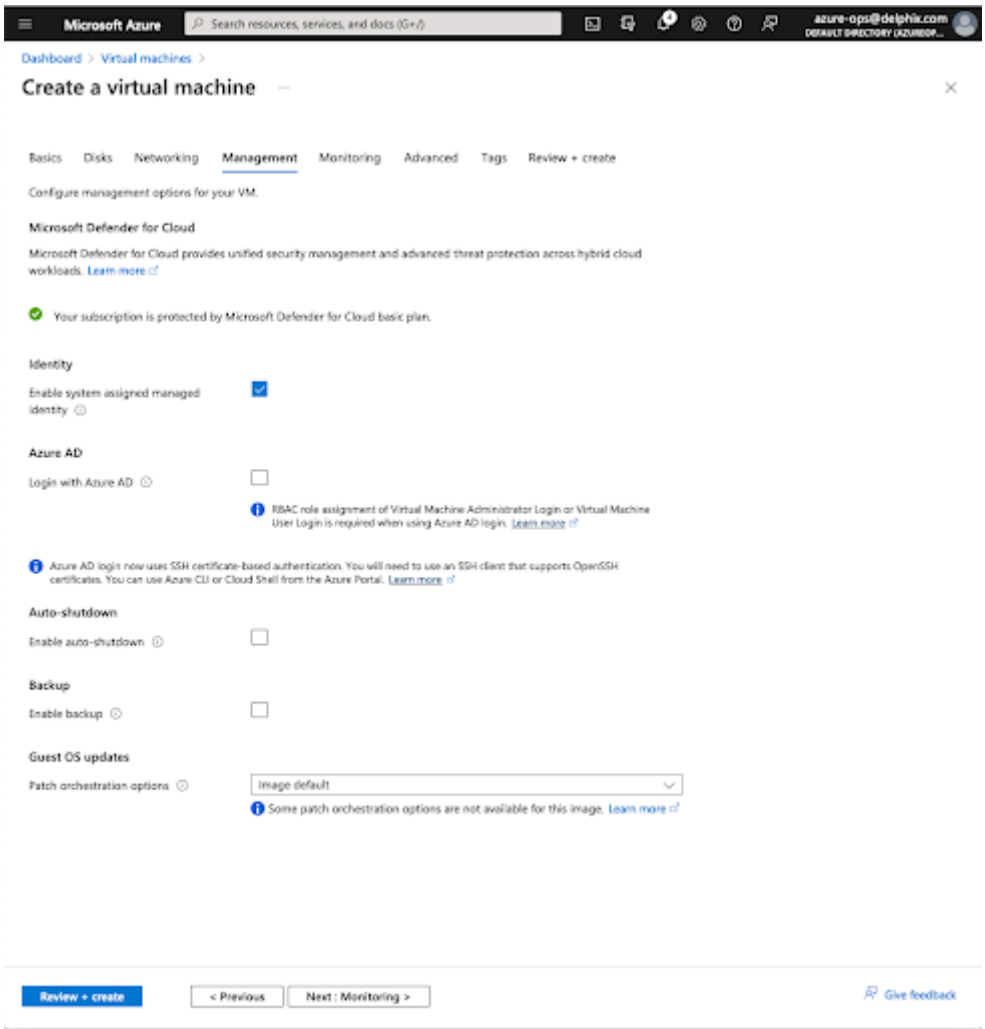
Use the example below, but replace the following placeholders with your values: `Your_Role_Name`, `Your_Subscription_ID`, `Your_Resource_Group_Name`, `Your_Storage_Account_Name`.

```
{
  "properties": {
    "roleName": "<Your_Role_Name>",
    "description": "Delphix object storage Azure role permissions",
    "assignableScopes": [
      "/subscriptions/<Your_Subscription_ID>/resourceGroups/
<Your_Resource_Group_Name>/providers/Microsoft.Storage/storageAccounts/
<Your_Storage_Account_Name>"
    ],
    "permissions": [
      {
        "actions": [
          "Microsoft.Storage/storageAccounts/blobServices/containers/read"
        ],
        "notActions": [],
        "dataActions": [
          "Microsoft.Storage/storageAccounts/blobServices/containers/blobs/write",
          "Microsoft.Storage/storageAccounts/blobServices/containers/blobs/read",
          "Microsoft.Storage/storageAccounts/blobServices/containers/blobs/move/
action",
```

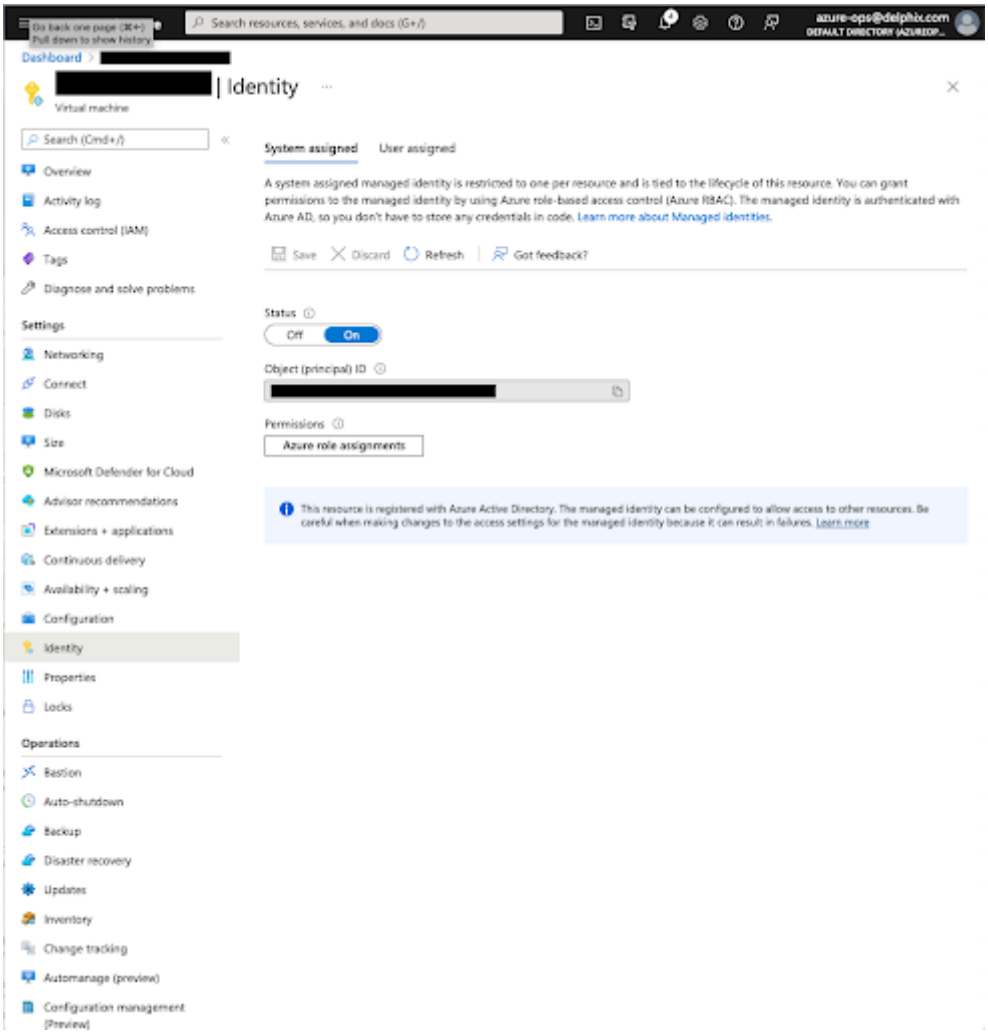
```
          "Microsoft.Storage/storageAccounts/blobServices/containers/blobs/add/
action",
          "Microsoft.Storage/storageAccounts/blobServices/containers/blobs/delete"
        ],
        "notDataActions": []
      }
    ]
  }
}
```

5.5.1.13.6 Virtual machine

Ensure the system-assigned Managed Identity is enabled when creating your virtual machine.



Navigate to the virtual machine and click on **Identity** and then **Azure role assignments**. After creation, assign the newly created role to the virtual machine for the necessary permissions.



Click **Add role assignments**. For the scope, select **Storage**. Select your subscription. For **Resource**, select the Storage Account you will be using. For the **Role**, select the new role you created above.

5.5.1.13.7 Testing permissions

To ensure that the role permissions are functioning properly, assign the role to a generic Ubuntu VM with the Azure CLI installed. Follow the Azure CLI installation[145] guide.

Then, execute the following commands to test access to your storage container:

```
delphix@demo-vm:~$ az login --identity
[
  {
    "environmentName": "AzureCloud",
    "homeTenantId": "<Tenant>",
    "id": "<ID>",
    "isDefault": true,
```

145 https://learn.microsoft.com/en-us/cli/azure/install-azure-cli

```
    "managedByTenants": [],
    "name": "Pay-As-You-Go",
    "state": "Enabled",
    "tenantId": "<Tenant>",
    "user": {
    "assignedIdentityInfo": "MSI",
    "name": "systemAssignedIdentity",
    "type": "servicePrincipal"
    }
  }
]

# List the storage containers
delphix@demo-vm:~$ az storage container list --account-name
<Your_Storage_Account_Name> --output table --auth-mode login
Name                                                        Lease Status    Last
Modified
-----------------------------------------------------------  --------------
-------------------------
<Your_Storage_Container_Name>                                                2022-04-2
8T18:57:01+00:00

# List the contents of the new empty container
delphix@demo-vm:~$ - <Your_Storage_Account_Name> --container-name
<Your_Storage_Container_Name> --output table --auth-mode login

# List the contents of the storage container again
delphix@demo-vm:~$ az storage blob list --account-name <Your_Storage_Account_Name> --
container-name <Your_Storage_Container_Name> --output table --auth-mode login

Name  Blob Type Blob Tier Length Content Type Last Modified          Snapshot
--------  -----------  -----------  --------  --------------
-------------------------  ----------
test.txt  BlockBlob             15    text/plain  2022-04-28T19:36:09+00:00

# Download the storage blob
delphix@demo-vm:~$ az storage blob download  --account-name
<Your_Storage_Account_Name> --container-name <Your_Storage_Container_Name> --output
table --auth-mode login --name test.txt --file downloaded-test.txt
Finished[#############################################################]  100.0000%

Name  Blob Type Blob Tier Length Content Type Last Modified          Snapshot
--------  -----------  -----------  --------  --------------
-------------------------  ----------
test.txt  BlockBlob             15    text/plain  2022-04-28T19:36:09+00:00
delphix@demo-vm:~$ cat downloaded-test.txt

This is a test
# Delete the storage blob
delphix@demo-vm:~$ az storage blob delete  --account-name <Your_Storage_Account_Name>
--container-name <Your_Storage_Container_Name> --output table --auth-mode login --
name test.txt
```

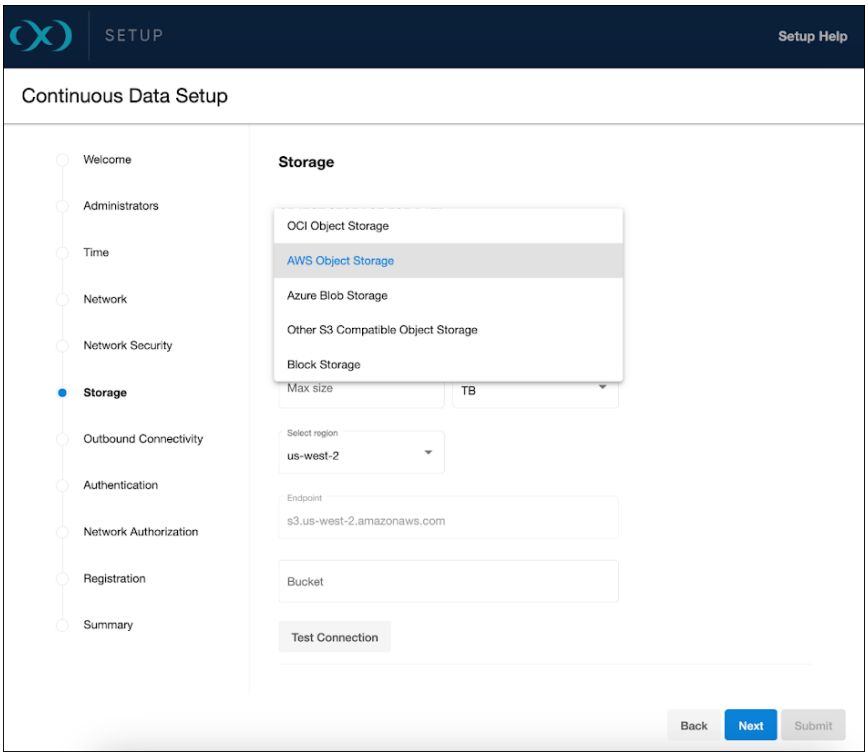
```
delphix@demo-vm:~$azstoragebloblist--account-name<Your_Storage_Account_Name>--
container-name<Your_Storage_Container_Name>--outputtable--auth-modelogi
```

5.5.1.13.8 Key based access

If Managed Identities are not used, static Storage Account access keys can be an alternative. Azure's comprehensive guidelines on managing Storage Account access keys, including best practices for key security and rotation, can be found in the Storage account keys manage[146] article.

5.5.1.14 AWS object storage setup

From the **Storage** setup page, select **AWS Object Storage** from the **Storage Type for Data** dropdown menu.

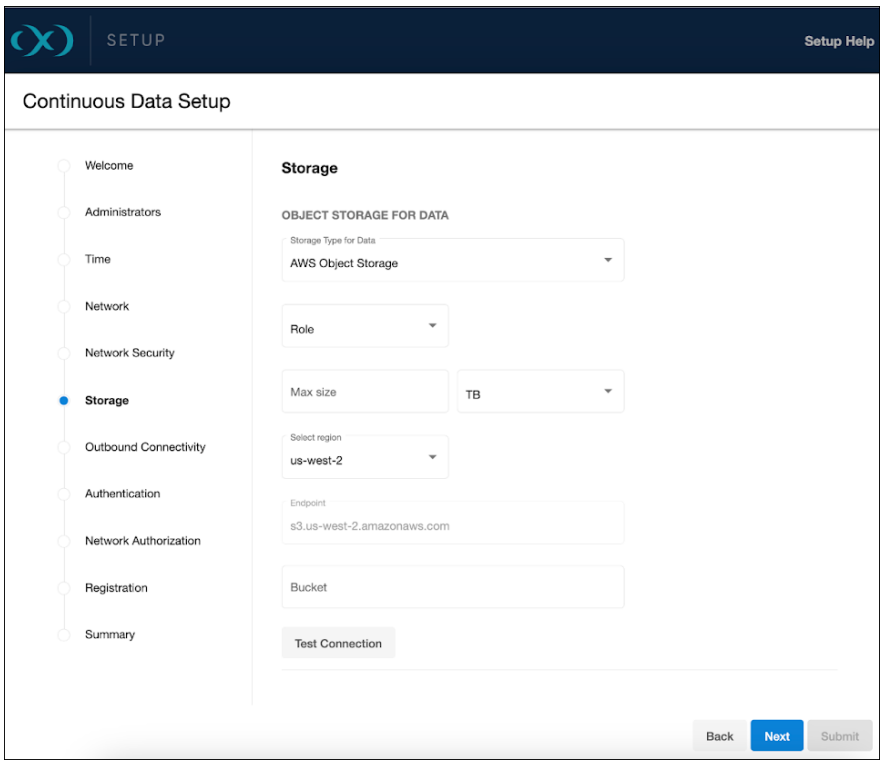


5.5.1.14.1 Access method

Choose the preferred method for the engine to access the storage bucket: Role (Instance profile[147]) or Access Key[148]. Role-based access is advocated to mitigate security risks and minimize the maintenance associated with access keys.

146 https://docs.microsoft.com/en-us/azure/storage/common/storage-account-keys-manage
147 https://repost.aws/knowledge-center/ec2-instance-access-s3-bucket
148 https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html

> Role-based access is recommended to minimize security risks and maintenance related to access keys.



5.5.1.14.2 Select region

From the **Select region** dropdown menu, choose from the available AWS regions. If your desired region is not listed, select **Other**, which will prompt you to fill out the **Endpoint**[149] field.

5.5.1.14.3 Storage capacity

Input the desired data storage capacity. AWS S3's pricing is based on the actual storage used. For instance, if you provision 10TB but only 1TB is utilized, you will be charged for the 1TB only. Adjusting this number post-setup is straightforward via the sysadmin login or Setup UI, but it cannot be reduced below the current usage by the Delphix engine. The specified number also serves as a quota to limit the object storage from expanding beyond an unforeseen point.

Enter the name of your bucket in the **Bucket** field and test the connection to verify the VM's access to the bucket.

149 https://docs.aws.amazon.com/general/latest/gr/rande.html

5.5.1.14.4 Block storage for cache

Block devices are utilized as cache, reducing latencies for frequently read data and acting as temporary storage for synchronous writes before the writes are persisted to object storage. The cache should be sized to match the frequently accessed data plus an additional 10% for bookkeeping. Alternatively, begin with a cache sized at 50% of all dSources to be added to the engine.

> For specific block storage requirements, refer to the **EBS configuration** section in the Deployment for AWS EC2[150] page.

When setting up the disks, ensure they support the throughput of the engine. For example:

- gp3 disks are recommended as they offer good performance at a lower cost. At 500 IOPS per 1GiB ratio, a 32 GiB volume can be configured to have the gp3 volume 16K IOPS limit and 1000 MB/s throughput. For reference, r5n.8xlarge instance has a 30K IOPS and 850 MB/s throughput limits so two gp3 devices would be sufficient for the instance.
- io2 disks have lower latency at a higher cost. However, the lower latency is not beneficial once the instance IOPS or throughput limit is reached.

5.5.1.14.5 Elastic data engines on AWS: Permissions

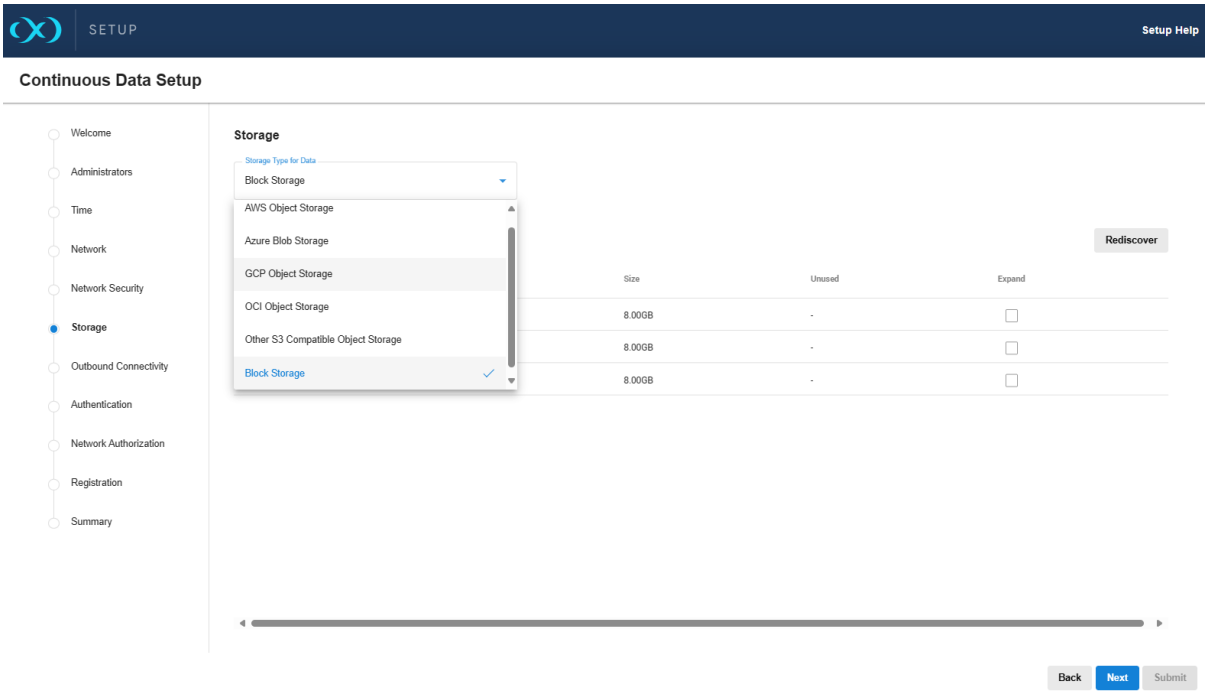
At a minimum, the following permissions are required for the object storage bucket:

```
AWS: [
  "s3:PutObject",
  "s3:GetObject",
  "s3:ListBucket",
  "s3:DeleteObject"
]
```

5.5.1.15 GCP object storage setup

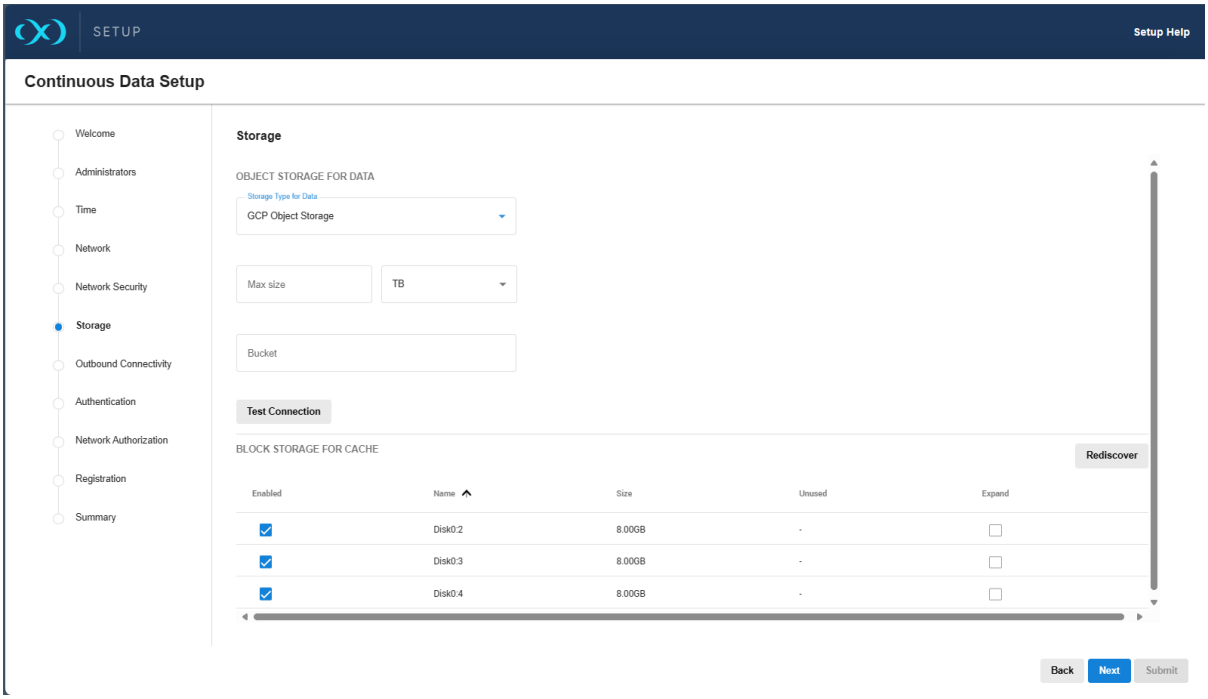
From the **Storage** setup page, select **GCP Object Storage** from the **Storage Type for Data** dropdown menu.

150 https://cd.delphix.com/docs/latest/deployment-for-aws-ec2#EBS-configuration



5.5.1.15.1 Access method

In GCP, the ideal way of accessing object storage is from an instance using a service account. A service account in GCP is similar to an IAM role in AWS, promoting secure and low-maintenance access to the object storage.



5.5.1.15.2 Storage capacity

When specifying storage capacity, note that GCP charges are based on the actual storage used. Thus, if you provision for 10TB but only utilize 1TB, billing will reflect the 1TB of actual usage. Adjusting this capacity is manageable through the sysadmin login or Setup UI, but cannot be decreased below what the Delphix Continuous Data Engine is currently utilizing. The specified number also serves as a quota to limit the object storage from expanding beyond an unforeseen point.

5.5.1.15.3 Bucket

Enter the name of your bucket in the **Bucket** field and test the connection to verify the VM's access to the bucket.

5.5.1.15.4 Block storage for cache

Block devices such as cache are used to reduce latencies for frequently read data and as temporary storage for synchronous writes before the writes are sent to object storage. If you already know the size of the frequently accessed data, then size the cache equal to (size of frequently accessed data + Extra 10% for bookkeeping purposes), If not, start with sizing the cache to 50% of the size of all dSources that will be added to the engine.

> For the detailed block storage requirements, refer to the **General storage configuration** section in the Deployment for GCP (see page 650) documentation.

Set up the disks such that they can support the throughput of the engine. **SSD Persistent Disks** are recommended.

5.5.1.15.5 Elastic data engines on GCP: Permissions

5.5.1.15.5.1 Setup

Create a service account and custom role and assign that role to a virtual machine.

- Enable private Google Access on the subnet used by the VM if it is not in the public subnet
- Create a virtual machine
- Create a bucket
- Create a compute service account

5.5.1.15.5.2 Permissions

Create a service account and provide access scoped to storage APIs or all cloud APIs. The account creating the virtual machine should have `serviceAccountUser` permissions on the service account so that while creating the instance this account could be assigned.

5.5.1.15.5.3 scopes = ["cloud-platform"]

The service account should have permissions to the following roles:

- legacyBucketWriter
- legacyObjectOwner

Here is an example of applying permissions from this service account on a bucket.

☐	dcoa-prod-dose@dlpx-dcoa-prod.iam.gserviceaccount.com	dcoa-prod-dose	Storage Legacy Bucket Writer Storage Legacy Object Owner	✎

You can also create and use a custom role for providing more restrictive permissions.

5.5.1.15.5.4 Bucket Protection and Lifecycle

Any protection policy (e.g. Object versioning, retention, event-based hold, etc…) or added Lifecycle rules are not supported. If present, the connectivity test to the bucket will fail or will have an adverse effect on the engine's data storage stability.

5.5.1.15.5.5 Testing Bucket Permissions

You can test the connection from the Delphix engine to the bucket via the Setup UI, but there is no way in the Setup UI to test the connection without starting from the beginning of setup for the engine. To test only the engine connection to a bucket or multiple buckets, use the testConnection API in the Delphix Continuous Data Engine CLI and look for the result status after executing the commit operation. If the connection test is successful, the result status will be true. Here is an example of how to use the Delphix Continuous Data Engine CLI to test the engine connection to the object storage bucket, which results in a successful connection. The engine hostname in the example is `test-gcp`:

```
$ ssh sysadmin@test-gcp
test-gcp> storage objectStorage testConnection

test-gcp objectStorage testConnection *> set type=GcpObjectStoreTest

test-gcp storage objectStorage testConnection *>

test-gcp storage objectStorage testConnection *> ls

Properties

    type: GcpObjectStoreTest (*)

    bucket: (required)

test-gcp storage objectStorage testConnection *> set bucket=test-gcp-1

test-gcp storage objectStorage testConnection *> commit

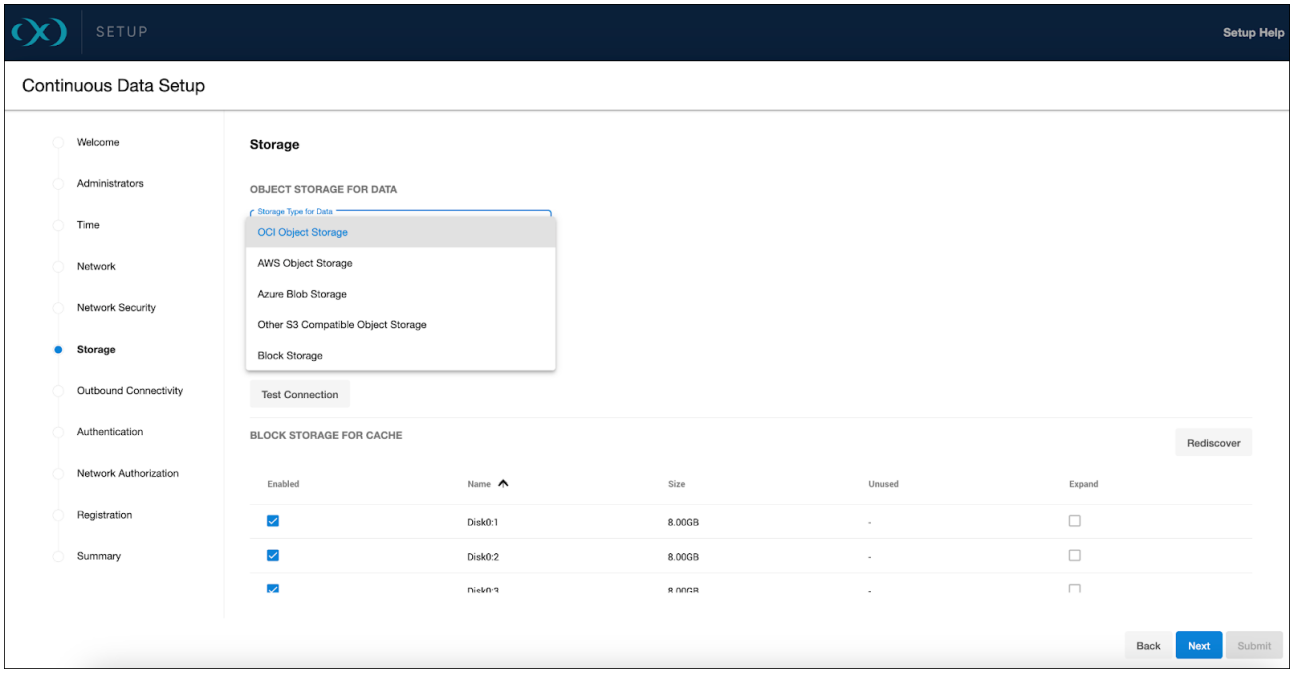
    type: ObjectStoreTestResult
```

```
    errorMessage: (unset)

    result: true
```

5.5.1.16 OCI object storage setup

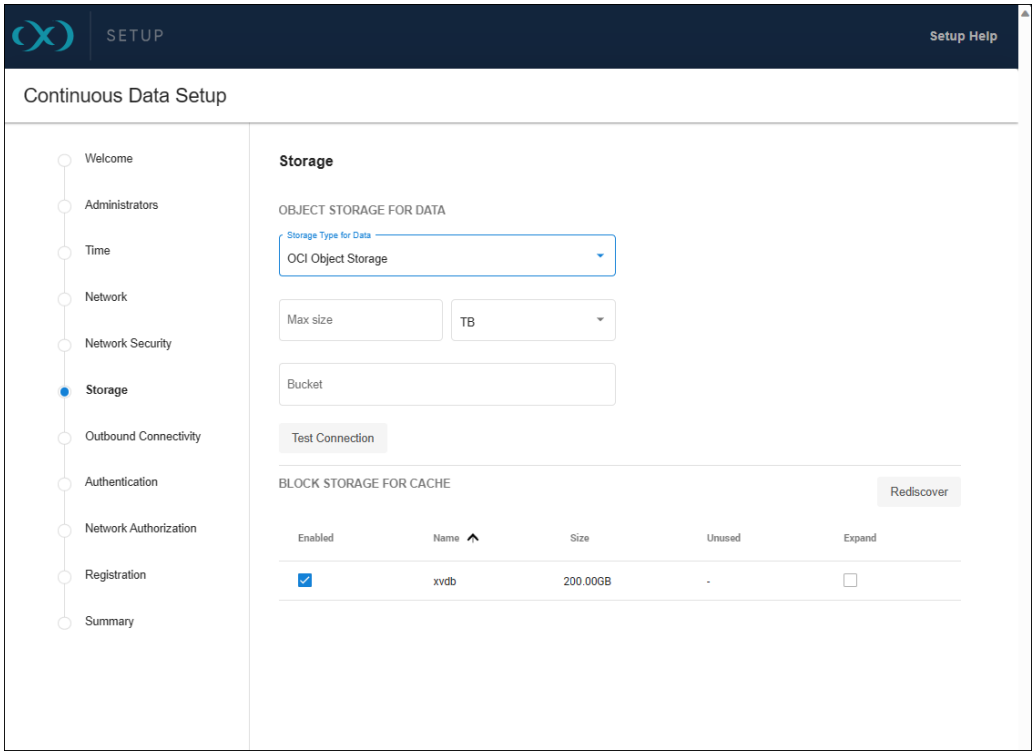
From the **Storage** setup page, select **OCI Object Storage** from the **Storage Type for Data** dropdown menu.



5.5.1.16.1 Access method

For accessing the object storage, OCI supports Role (Instance Principals[151]) as the sole method. This approach parallels the role-based access of AWS IAM, promoting secure and low-maintenance access to the object storage.

151 https://docs.oracle.com/en-us/iaas/Content/Identity/Tasks/callingservicesfrominstances.htm



5.5.1.16.2 Storage capacity

When specifying storage capacity, note that OCI charges based on the actual storage used. Thus, if you provision for 10TB but only utilize 1TB, billing will reflect the 1TB of actual usage. Adjusting this capacity is manageable through the sysadmin login or Setup UI, but cannot be decreased below what the Delphix engine is currently utilizing. The specified number also serves as a quota to limit the object storage from expanding beyond an unforeseen point.

Ensure that the bucket name is correctly entered in the **Bucket** field and test the connection to verify the VM's access to the bucket.

5.5.1.16.3 Block storage for cache

Block devices are utilized as cache, reducing latencies for frequently read data and acting as temporary storage for synchronous writes before the writes are persisted to object storage. The cache size should be equivalent to the size of the frequently accessed data, plus an additional 10% for overhead. If the data size is unknown, initiate with a cache sized at 50% of the dSources to be added to the engine.

> For detailed block storage requirements, refer to the **General storage configuration** section in the Deployment for OCI[152] documentation.

152 https://cd.delphix.com/docs/latest/deployment-for-oci#id-(16.0.0.0)DeploymentforOCI-Generalstorageconfiguration

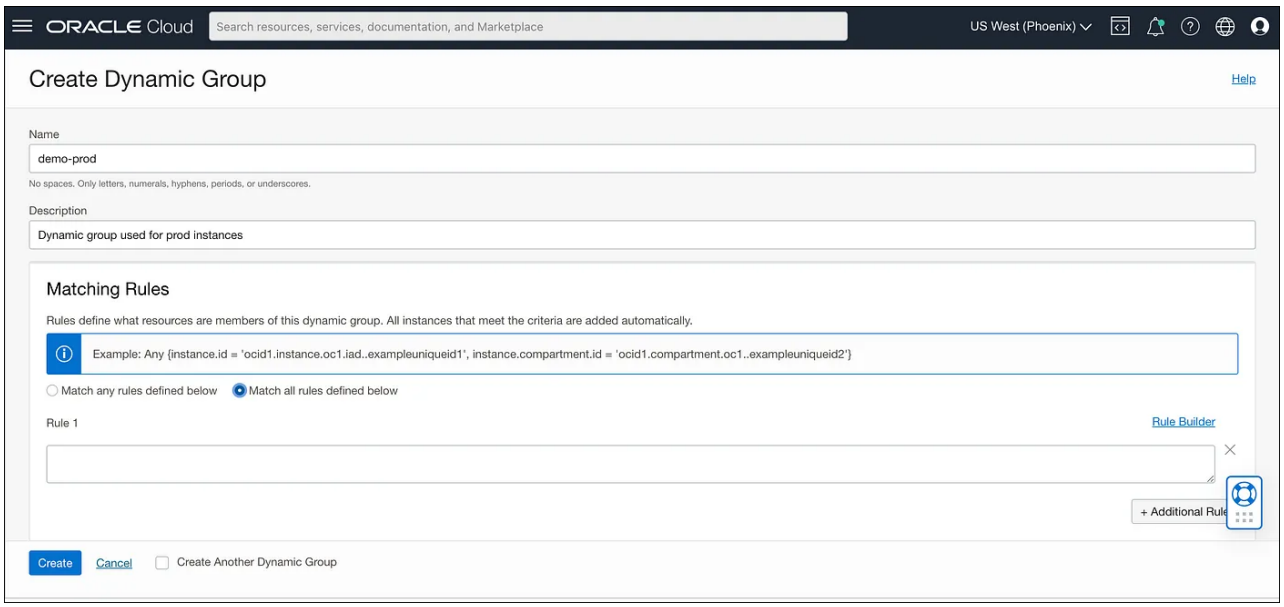
It is important to configure the disks to support the engine's throughput, such as using Ultra High Performance or Higher Performance volumes, which are supported. The maximum IOPS and throughput will be proportional to the volume size, requiring a minimum size of 200GB for Higher Performance volumes.

5.5.1.16.4 Permissions and setup

5.5.1.16.4.1 Setup

The OCI IAM allows for access to various services from an OCI instance without API keys or customer secrets. Instance principals provide a secure method for service access, similar to AWS IAM Roles.

To create a dynamic group in the OCI Console, navigate to **Identity** → **Dynamic Groups** → **Create Dynamic Group**. Specify the **Name** and **Description** the group and choose if you want all the rules or just one to match for the dynamic group.



5.5.1.16.4.2 Creating rules

Dynamic groups require rules to identify which instances belong to them. For instance, rules can be based on the tag value of an instance or the presence of instances in a specific compartment. Here are examples of rules:

- `tag.InstanceType.value='prod'`: Check the tag value on the instance and match it with some defined value.
- `instance.compartment.id = 'ocid1.compartment.oc1..aaaaaaxx'`: All the instances present in a specific compartment.

5.5.1.16.4.3 Policy

Post group creation, it is necessary to define a policy specifying the access level for the service. A policy instructs the OCI IAM to provide access to the dynamic group. To understand OCI Policies in detail, visit How Policies Work (without Identity Domains)[153]. Policies can be created via the OCI Console by navigating to **Identities** → **Policies** → **Create Policy**, where you can specify the **Name**, **Description**, **Compartment**, and **Policy Statement**.

The policy should allow the dynamic group to manage and inspect buckets and objects within a specified compartment. Here is an example of a policy statement:

```
Allow dynamic-group <dynamic group name> to manage buckets in compartment id
<compartment id> where target.bucket.name=<regex for bucket name>
Allow dynamic-group <dynamic group name> to inspect buckets in compartment id
<compartment id>
Allow dynamic-group <dynamic group name> to manage objects in compartment id
<compartment id> where target.bucket.name=<regex for bucket name>
```

Once the setup is complete, instances launched in the specified compartment with the defined tag will have access to manage object storage objects within that compartment.

5.5.1.16.5 Testing permissions

You can demonstrate the use of instance principals with the OCI CLI, which does not require managing security credentials. The OCI CLI can be installed on various operating systems following the Quickstart[154] guide. With the CLI, you can make calls using the `--auth instance_principal` parameter. Here are some example commands:

```
# Get details of a bucket
oci os bucket get -bn prodbucket --auth instance_principal

# Upload an object to a bucket
oci os object put -bn prodbucket --auth instance_principal --file object.txt
```

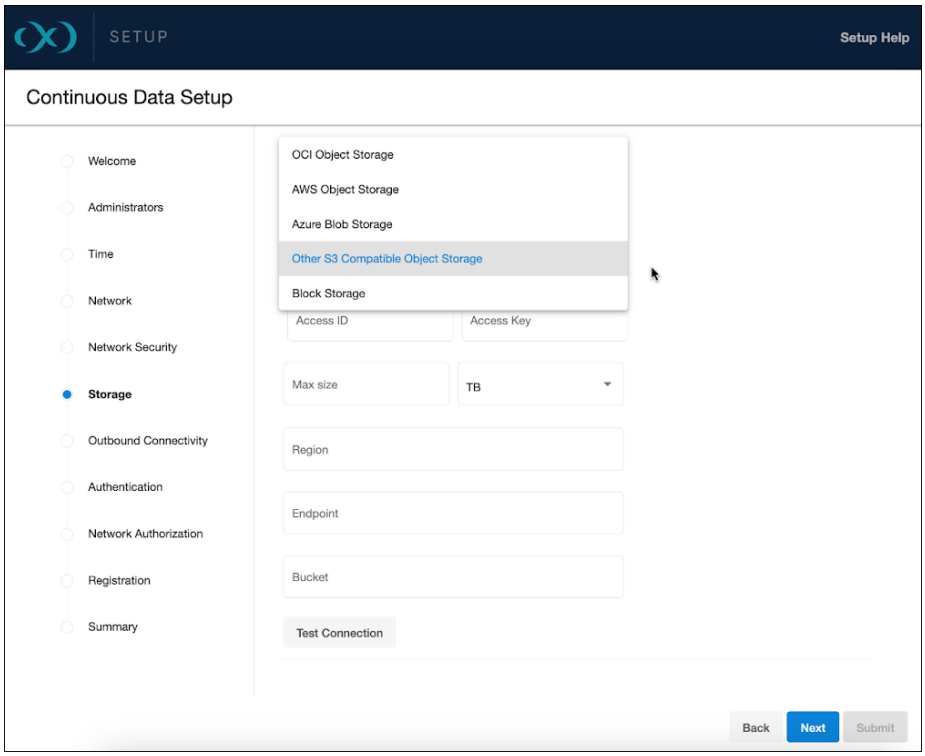
Setting `OCI_CLI_AUTH=instance_principal` as an environment variable is also an option.

> Instance principals are compatible with OCI CLI and SDKs such as Python and Java but do not work with the Amazon S3 Compatibility API, which requires customer secret keys.

153 https://docs.oracle.com/en-us/iaas/Content/Identity/Concepts/policies.htm
154 https://docs.oracle.com/en-us/iaas/Content/API/SDKDocs/cliinstall.htm

5.5.1.17 Other S3 compatible object storage setup

From the **Storage** setup page, select **Other S3 Compatible Object Storage** (for instance, on-premise object storage) from the **Storage Type for Data** dropdown menu.



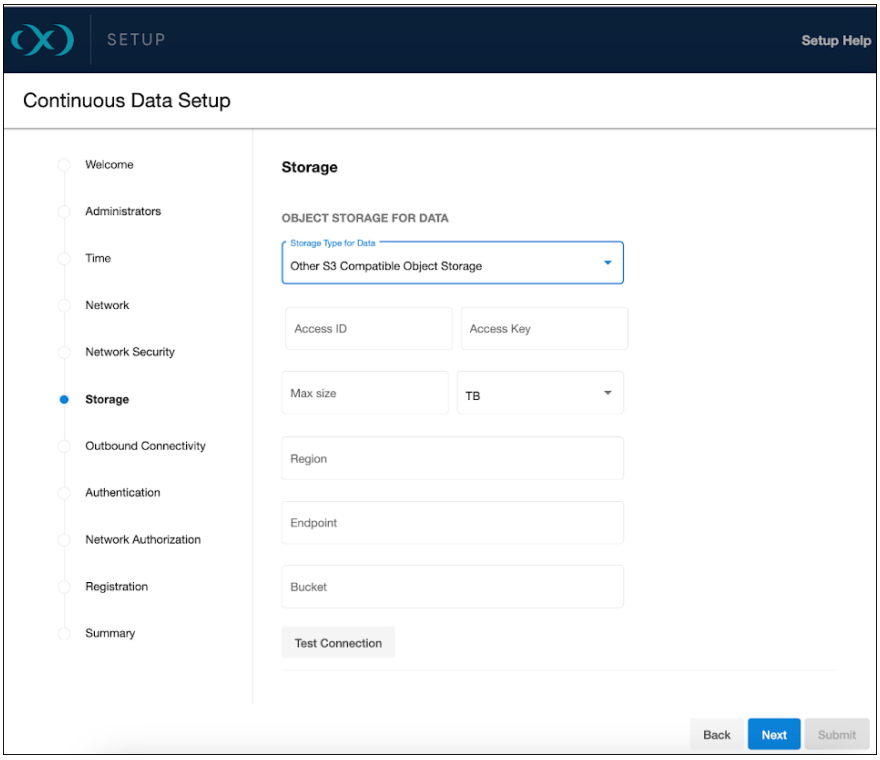
5.5.1.17.1 Access method

For on-premise object storage we currently support storage vendors that conform to the following

- S3 REST API compatibility
- strong read-after-write consistency
- supports S3 key id and secret access key authentication
- perpetual key support

Should your on-premise object storage require a security certificate for secure connections between the Delphix engine and the object storage, this certificate must be installed on the Delphix engine before configuring the storage. For more information, refer to the Certificate management[155] page.

155 https://cd.delphix.com/docs/latest/certificate-management



5.5.1.17.2 Authentication details

The following details are required for accessing the on-premise object storage:

- Access ID
- Access Key
- Region (optional, depending on whether a region is required for your setup)
- Endpoint
- Bucket

Enter these details carefully and ensure that you test the connection to confirm that the VM can access the bucket.

5.5.1.17.3 Storage capacity

Determine the amount of data you wish to store on the engine. This size should be akin to the total storage on a traditional engine. With Elastic Data Engines, modifying this number post-setup is straightforward, accessed via the sysadmin login or Setup UI, but it cannot be decreased below the engine's current usage. The specified number also serves as a quota to limit the object storage from expanding beyond a certain point.

5.5.1.17.4 Block storage for cache

Block devices are utilized as cache, reducing latencies for frequently read data and acting as temporary storage for synchronous writes before the writes are persisted to object storage. When sizing the cache, if the size of the frequently accessed data is known, set the cache size to equal that amount plus an extra 10%

for overhead. If the size is uncertain, start with a cache that is 50% of the total size of all dSources to be added to the engine.

> For detailed block storage requirements, see the **General storage** section of Deployment for VMware[156].

Ensure that the disks are set up to handle the engine's throughput needs.

5.5.1.18 Updating object storage endpoints

In 18.0.0.0, parameters like endpoint, region, bucket/container, and access credentials can be updated. This ability is only available via the CLI, with a graphical interface soon to come.

The following example CLI session shows how the `accessCredentials` can be updated (from using keys to using managed identities) on a cloud engine:

```
az-test> storage objectStorage
az-test storage objectStorage> get
    type: BlobObjectStore
    accessCredentials:
        type: BlobObjectStoreAccessKey
        azureAccount: doseprod
        azureKey: ********
    cacheDevices: lun0,lun1,lun2
    configured: true
    container: az-test
    endpoint: -
    size: 1TB
az-test storage objectStorage> update
az-test storage objectStorage update *> edit accessCredentials
az-test storage objectStorage update accessCredentials *> set
type=BlobObjectStoreAccessManagedIdentities
az-test storage objectStorage update accessCredentials *> get
    type: BlobObjectStoreAccessManagedIdentities (*)
    azureAccount: doseprod
az-test storage objectStorage update accessCredentials *> commit
az-test storage objectStorage> get
    type: BlobObjectStore
    accessCredentials:
        type: BlobObjectStoreAccessManagedIdentities
        azureAccount: doseprod
    cacheDevices: lun0,lun1,lun2
    configured: true
    container: az-test
```

156 https://cd.delphix.com/docs/latest/deployment-for-vmware#SCSI-controller

```
    endpoint: -
    size: 1TB
az-test storage objectStorage>
```

5.5.2 Customizing the Delphix engine system settings

5.5.2.1 Overview

This article describes how to customize the initial system setup requirements for memory, number of CPUs, storage disks, and network configuration. The OVA file used to install the Delphix Engine is configured for the minimum system requirements. These can be customized to match the capabilities of a specific system.

5.5.2.2 Procedure

1. Shut down the guest operating system and power off the Delphix Engine.
2. Under **Getting Started**, select **Edit Virtual Machine Settings**.
3. The system settings can now be customized.

Setting	Options
Memory Size	Set to 64GB or larger based on sizing analysis. In the **Resource Allocation** panel, ensure that **Reserve all guest memory** is checked. To adjust the resource allocation of a VM without rebooting it, Delphix supports "hot-plugging" CPUs and memory to ZFS. Please note that this is only available on ESX. For more information, refer to the Enabling hot-adding section on this page.
Number of CPUs	Allocate 8 vCPUs or more based on Delphix licensing. vCPUs should be fully reserved to ensure that the Delphix Engine does not compete for CPU cycles on an overcommitted host.
Supported Controllers	Delphix supports Paravirtual SCSI (PVSCSI) controllers or LSI Parallel controllers. A mix of different types of SCSI controllers is not supported within the engine. It is recommended to use PVSCSI controllers by default. Although PVSCSI supports high throughput with minimal processing cost, the performance improvements on Delphix Engine may vary from case to case.

Setting	Options
Disks for Data Storage	Add virtual disks to provide storage for user data such as dSources and VDBs. The underlying storage must be redundant. Add a minimum of 150GB per storage disk. All virtual disks should be the same size and have the same performance characteristics. If using VMFS, use thick provisioned, lazy zeroed disks. To alleviate IO bottlenecks at the virtual controller layer, spread the virtual disks across all four virtual SCSI controllers.
Data Storage Multipathing Policy	For EMC storage, the multipathing policy should always be set to round-robin (default for 5.x). Additionally, change the IO Operation Limit from the default of 1000 to 1. This should be strongly considered for other storage platforms as well. See VMware KB article EMC VMAX and DMX symmetrix storage array recommendations for optimalperformance on VMware ESXi/ESX[157].
Network	The network configuration is set to have a VMXNET3 network adapter. VMXNET3 is a tuned network interface that is included with the VMtools provided in the OVA file. It will be assigned to VM Network. **JUMBO Frames** VMXNET3 supports Ethernet jumbo frames, this can be used to maximize throughput and minimize CPU utilization. **Adding Link Aggregation via VMware NIC Teaming** To increase throughput or for failover, add multiple physical NICs to the vSwitch that is connected to the Delphix Engine. To increase throughput, NIC Teaming must use the Route Based on IP Hash protocol for load balancing. **Dedicate Physical NICs to the Delphix Engine** For best performance, assign the Delphix Engine to network adapters that are used exclusively by Delphix.

157 https://kb.vmware.com/s/article/2072070

5.5.2.3 Enabling hot-adding

To manually enable hot-adding in the VCenter console, use the following steps.

1. Select the Delphix Engine.
2. Right-click on the engine and select **Edit settings**.
3. Edit settings:
   a. To enable CPU hot-add, navigate to the CPU section and select the **CPU Hot Plug** checkbox.
   b. To enable memory hot-add, navigate to the memory section and select the **Memory Hot Plug** checkbox.
4. Modify the appropriate setting to the new value; note that only expansion of CPU and memory is supported, and attempts to reduce the values available to the Delphix Engine will fail.
5. Click **Ok** to confirm the changes.

Newly-added memory is not automatically reflected in system information APIs nor in the UI. Restarting the management service is needed to refresh this information.

For customers migrating from versions before 6.0, an extra step is required to enable CPU/memory hot-plug.

1. After migrating, turn off the VM.
2. Open **VM Settings**.
3. Select **VM Options**.
4. Under **General**, update the guest OS to Ubuntu (64-bit).
5. Under **Advanced**, select **Edit Configuration**.
6. Set **disk.EnableUUID** to TRUE.
7. Follow normal hot-plug enable steps.

> For customers that are upgrading their existing engines from versions below 6.0.6.0, two vSphere properties need to be enabled manually after the upgrade from vSphere in order to use this feature. The properties are **cputHotAddEnabled** and **memoryHotAddEnabled**.
> When upgrading to 6.0 from 5.3, the guest OS and version in vSphere also needs to be updated to Ubuntu, in addition to the above two properties. Without this, it has been noticed that hot-adding memory is not allowed from vSphere, even if memoryHotAddEnabled is enabled.

5.5.2.4 Post-requisites

- After making any changes to the system settings, power on the Delphix Engine again and proceed with the initial system configuration as described in Setting up network access to the Delphix engine (see page 569).

5.5.3 Installing an OVA or AMI

5.5.3.1 Overview

This article outlines how to install the Delphix Engine using an OVA or AMI file.

5.5.3.2 Procedure to install an OVA

Use the Delphix-supplied OVA file to install the Delphix Engine. The OVA file is configured with many of the minimum system requirements. The underlying storage for the install is assumed to be redundant SAN storage.

1. Download the OVA file from https://download.delphix.com[158]. A support login is needed from the Delphix representative or welcome letter.
    a. Navigate to the Delphix Product Releases/<Current Version>/Appliance Images page.
2. Login using the vSphere client to the vSphere server (or vCenter Server) where the Delphix Engine will be installed.
3. In the vSphere Client, click **File**.
4. Select **Deploy OVA Template**.
5. Browse to the OVA file.
6. Click **Next**.
7. Select a **hostname** for the Delphix Engine. This hostname will also be used in configuring the Delphix Engine network.
8. Select the **data center** where the Delphix Engine will be located.
9. Select the **cluster** and the **ESX host**.
10. Select one (1) **data store** for the **Delphix OS**. This datastore can be **thin-provisioned** and must have enough free space to accommodate the 127GB comprising the Delphix operating system.
11. Select four (4) or more **data stores** for Database Storage for the Delphix Engine. The Delphix Engine will stripe all of the Database Storage across these VMDKs, so for optimal I/O performance, each VMDK must be equal in size and be configured **Thick Provisioned - Eager Zeroed**. Additionally, these VMDKs should be distributed as evenly as possible across all four SCSI I/O controllers.
12. Select the **virtual network** to be used. If using multiple physical NICs for link aggregation, vSphere NIC teaming must be used. Do not add multiple virtual NICs to the Delphix Engine itself. The Delphix Engine should use a single virtual network. For more information, see Optimal Network Architecture for the Delphix Engine (see page 719).
13. Click **Finish**. The installation will begin and the Delphix Engine will be created in the location specified.

158 https://download.delphix.com/

14. Once the installation has completed, power on the Delphix Engine and proceed with the initial system configuration as described in Setting Up Network Access to the Delphix Engine (see page 569).

5.5.3.3 Procedure to install an AMI

Use the Delphix-supplied AMI file to install the Delphix Engine.

1. On the Delphix download site, click the AMI to be shared and accept the Delphix License agreement. Alternatively, follow a link given by the Delphix solutions architect.
2. On the **Amazon Web Services Account Details** form presented:
   a. Enter the **AWS Account Identifier**, which can be found here: https://console.aws.amazon.com/billing/home?#/account[159]. To use the **Gov Cloud AWS Region**, be sure to enter the ID for the AWS Account which has Gov Cloud enabled.
   b. Select which **AWS Region** the AMI should be shared in. To have the AMI shared in a different region, contact the Delphix account representative for the proper arrangements.
3. Click **Share.** The Delphix Engine will appear in the list of AMIs in AWS momentarily.

5.6 Validating host deployment with host Checker

5.6.1 Overview

Delphix has developed a hostchecker script that contains standardized checks for source and target hosts - these checks generally fall into three buckets:

- OS and Host permissions/access
- Network Port Checks
- DB-specific functionality

OS and Host permissions/access and network port checks can (and should) be performed prior to Delphix installation to ensure a smooth deployment.

Each DB should have a specifically associated hostchecker, there is detailed documentation on the DB-specific hostchecker page.

5.6.2 Procedure

1. Download the appropriate **HostChecker tarball** for the engine from https://download.delphix.com/. Tarballs follow the naming convention "hostchecker_<OS>_<processor>.tar". For example, if validating a linux x86 host, download the hostchecker_linux_x86.tar tarball.
2. Create a working directory and extract the **HostChecker files** from the **HostChecker tarball**

159 https://console.aws.amazon.com/billing/home#/account

```
mkdir dlpx-host-checker
cd dlpx-host-checker/
tar -xf hostchecker_linux_x86. tar
```

3. Run the `sh` script contained within:

```
sh hostchecker.sh
```

This will extract the JDK included in the tarball (if necessary) and invoke the HostChecker.

```
ora10205@bbdhcp: /home/ora10205/hostchecker- > sh hostchecker.sh
Extracting the JDK from the tarball jdk-6u45-linux-i586. tar .gz.
```

Do not run the HostChecker as root; this will cause misleading or incorrect results from many of the checks.

4. Select which **checks** to run. Note that it is possible to run checks without spawning the interface. Enter `--help` to get a list of arguments that can pass to the HostChecker.
5. As the checks are made, enter the requested **arguments**.
6. Read the output of the check. The general format is that severity increases farther down the output. First comes informational output, then warnings, then errors. If you see a message that starts with `Internal Error`, forward it to Delphix Support immediately. This represents a potential bug in the HostChecker, and not necessarily a problem with your environment.
7. Error or warning messages will explain any possible problems and how to address them. Resolve the issues that the HostChecker describes. Do not be surprised or undo your work if more errors appear the next time you run HostChecker, because the error you just fixed may have been masking other problems.
8. Repeat steps 3 - 7 until all the checks return no errors or warnings.

5.7 Deployment for VMware

5.7.1 Overview

This article outlines the requirements for deploying the Delphix Engine on VMware (including supported versions and instance configurations), as well as recommended configuration parameters for optimal performance.

The Delphix Engine is an intensive platform, both from a network and a storage perspective. If the Delphix Engine competes with other virtual machines on the same host for resources it will result in increased latency for all operations. As such, it is crucial that the ESXi host is not over-subscribed, as this eliminates the possibility of a lack of resources for the Delphix Engine. This includes allowing a percentage of CPU resources for the hypervisor itself as it can de-schedule an entire VM if the hypervisor is needed for managing IO or compute resources.

> Delphix disk storage capabilities are based on the backend storage provided. Performance, redundancy, and stability characteristics are determined by the hypervisor or Cloud provisioned storage.

5.7.2 Supported ESX versions

Requirements	Notes
• VMware Cloud • VMware ESXi 8.0, 8.0 U1 • VMware ESXi 7.0, 7.0 U1, 7.0 U2, 7.0 U3 • VMware ESXi 8.0, 8.0 U1, 8.0 U2	• If a minor release version is listed as supported, then all patch releases applicable to that minor release are supported. • Public cloud deployments such as VMware cloud on AWS[160] and Azure VMware solution[161] are supported. • While ESXi 7.0 U3 is supported, this is only for patch releases 3c and newer, due to critical issues in the earlier patch versions[162].

5.7.3 Virtual machine hardware versions

The Delphix Engine VM is distributed as an OVA for VMware, and is configured with the hardware version corresponding to the lowest ESXi version that release is qualified for. As the Engine is upgraded, the ESXi versions supported may change, but the hardware version may remain the same based on the original deployment.

Users who wish to upgrade the VM hardware version for compatibility, enhanced feature support, or other reasons may feel free to do so, with consideration for any compatibility concerns that may arise in environments where multiple ESXi versions are present, as an upgraded hardware version can affect other VMware operations (vMotion, etc). VMware support and documentation should be consulted before committing any hardware version upgrade for a guest VM, but Delphix does not maintain any version requirements in this regard.

160 https://www.vmware.com/products/vmc-on-aws.html
161 https://azure.microsoft.com/en-us/products/azure-vmware
162 https://kb.vmware.com/s/article/86398

5.7.4 Virtual CPUs

Requirements	Notes
8 vCPUs	• CPU resource shortfalls can occur both on an over-committed host as well as competition for host resources during high IO utilization. • CPU reservations are **strongly recommended** for the Delphix VM so that Delphix is guaranteed the full complement of vCPUs even when resources are overcommitted. • It is suggested to use a single core per socket unless there are specific requirements for other VMs on the same ESXi host.
Never allocate all available physical CPUs to virtual machines	• CPU for the ESXi Server to perform hypervisor activities must be set aside before assigning vCPUs to Delphix and other VMs. • We recommend that a minimum of 8-10% of the CPUs available are reserved for hypervisor operation. (e.g. 12 vCPUs on a 128 vCore system).

5.7.5 Memory

Requirements	Notes
128 GB vRAM (recommended) 64GB vRAM (minimum)	• The Delphix Engine uses its memory to cache database blocks. More memory will provide better read performance. • Memory reservations are **required** for the Delphix VM. The performance of the Delphix Engine will be significantly impacted by the over-commitment of memory resources in the ESX Server. • Reservations ensure that the Delphix Engine will not be forced to swap pages during times of memory pressure on the host. A swapped page will require orders of magnitude more time to be brought back to physical memory from the ESXi swap device.

Requirements	Notes
Never allocate all the available physical memory to virtual machines.	• The default ESX minimum free memory requirement is 6% of the total RAM. When free memory falls below 6%, ESX starts swapping out the Delphix guest OS. Delphix recommends leaving about 8-10% free to avoid swapping. • For example, when running on an ESX Host with 512GB of physical memory, no more than 470GB (92%) should be allocated to the Delphix VM (and all other VMs on that host).
Memory for the ESX Server to perform hypervisor activities must be set aside before assigning memory to Delphix and other VMs.	Failure to ensure sufficient memory for the host can result in a hard memory state for all VMs on the host which will result in a block for memory allocations.

5.7.6 Network

Requirements	Notes
The ova is pre-configured to use one virtual ethernet adapter of type VMXNET 3.	• Jumbo frames are highly recommended to reduce CPU utilization, decrease latency and increase network throughput. (typically 10-20% throughput improvement) • If additional virtual network adapters are desired, they should also be of type VMXNET 3.
A 10GbE NIC in the ESX Server is recommended.	• For VMs having only gigabit networks, it is possible to aggregate several physical 1GbE NICs together to increase network bandwidth (but not necessarily to reduce latency). Refer to the VMware Knowledge Base article NIC Teaming in ESXi and ESX[163]. However, it is not recommended to aggregate NICs in the Delphix Engine VM.

163 https://delphixdocs.atlassian.net/wiki/pages/resumedraft.action?draftId=4293295

Requirements	Notes
If the network load in the ESX Server hosting the Delphix engine VM is high, dedicate one or more physical NICs to the Delphix Engine.	• Adding NICs only works if VMs are discovered using different interfaces. The NFS/iSCSI mounts will only use the network associated with the discovery. • See General Network and Connectivity Requirements (see page 717) for information about specific port configurations, and Network Performance Configuration Options (see page 719) for information about network performance tuning

To bootstrap a networking configuration to reach the Delphix Engine, after deploying it into your environment for the first time, you can use one of the following options:

- DHCP
- Cloud-init for public clouds
- You can login to the serial console to configure networking via the CLI
- OVF guest customizations (see page 599) to pass in network configuration to the VM before it has a network connection. For the customization steps:
    - Type the name of the profile and click **Next**
    - The domain will be ignored by Delphix. Click **Next**
    - The time zone setting will be ignored by Delphix, Click **Next**
    - Delphix doesn't allow scripts, click **Next**
    - Manually select custom settings, select a **NIC** and click **Edit**
        - Provide **Netmask** and **Gateway**
        - Select one of the options for IP
        - Confirm changes and click **Next**
    - Add **DNS**
    - Confirm creation of profile, click **Finish**
    - You can use this template to clone a VM

5.7.7 SCSI controller

Requirements	Notes
PVSCSI (default)/ LSI Logic Parallel	When adding virtual disks make sure that they are evenly distributing the load across the maximum of 4 virtual SCSI controllers. Spreading the disks across available SCSI controllers evenly will ensure optimal IO performance from the disks. For example, a VM with 4 SCSI controllers and 8 virtual disks should distribute the disks across the controllers as follows: disk0 = SCSI(0:0) - System Disk on Controller 0 Port 0 (ignore for purposes of load balancing) disk1 = SCSI(0:1) - Data Disk on Controller 0 Port 1 disk2 = SCSI(1:1) - Data Disk on Controller 1 Port 1 disk3 = SCSI(2:1) - Data Disk on Controller 2 Port 1 disk4 = SCSI(3:1) - Data Disk on Controller 3 Port 1 disk1 = SCSI(0:2) - Data Disk on Controller 0 Port 2 disk2 = SCSI(1:2) - Data Disk on Controller 1 Port 2 disk3 = SCSI(2:2) - Data Disk on Controller 2 Port 2 disk4 = SCSI(3:2) - Data Disk on Controller 3 Port 2 For load purposes, we generally focus on the Delphix storage (data disks) and ignore the controller placement of the system disk. IDE and SATA controllers are not supported on VMware platforms

5.7.8 General storage

VMware offers options for disk storage, which include "Independent - Persistent" and "Independent - Non-persistent". The non-persistent setting is catastrophic to the Delphix platform when the VM reboots, thus, **Independent - Persistent** is required for installation.

Requirements	Notes
Storage used for Delphix must be provisioned from storage that provides data protection.	For example, using RAID levels with data protection features, or equivalent technology. The Delphix engine product does not protect against data loss originating at the hypervisor or SAN layers. For more information refer to, Optimal Storage Configuration Parameters for the Delphix Engine (see page 896).

5.7.9 Delphix storage options

There are three types of data that Delphix stores on disk, which is:

1. **Delphix VM Configuration Storage:** stores data related to the configuration of the Delphix VM. VM Configuration Storage includes the VMware ESX configuration data as well as log files.
2. **Delphix Engine System Disk Storage:** stores data related to the Delphix Engine system data, such as the Delphix .ova settings.
3. **Database Storage:** stores data used by Delphix objects such as dSources and virtual databases (VDBs).

5.7.9.1 Delphix VM configuration storage

The Delphix VM configuration should be stored in a VMFS volume (often called a "datastore").

Requirements	Notes
The VMFS volume should have enough available space to hold all ESX configuration and log files associated with the Delphix Engine.	If a memory reservation is not enabled for the Delphix Engine (in violation of memory requirements stated above), then space for a paging area equal to the Delphix Engine's VM memory must be added to the VMFS volume containing the Delphix VM configuration data.

5.7.9.2 Delphix engine system disk storage

The VMFS volume must be located on shared storage in order to use vMotion and HA features.

Requirements	Notes
The Delphix Engine system disk should be stored in a VMDK.	• The VMDK for the Delphix Engine System Disk Storage is often created in the same VMFS volume as the Delphix VM definition. In that case, the datastore must have sufficient space to hold the Delphix VM Configuration, the VDMK for the system disk, and a paging area if a memory reservation was not enabled for the Delphix Engine.
The Delphix .ova file is configured for a 127GB system drive.	• The VMFS volume where the .ova is deployed should, therefore, have at least 127GB of free space prior to deploying the .ova.

5.7.9.3 Database storage

5.7.9.4 Option 1: Block Storage for database storage

Shared storage is required in order to use vMotion and HA features. In addition to making sure the latest VMware patches have been applied, check with the organizations hardware vendor for updates specific to the hardware configurations. VMDKs (Virtual Machine Disks) or RDMs (Raw Device Mappings) operating in virtual compatibility mode can be used for database storage.

Requirements	Notes
A minimum of 4 VMDKs or RDMs should be allocated for database storage.	• Allocating a minimum of 4 VMDKs or RDMs for database storage enables the Delphix File System (DxFS) to make sure that its file systems are always consistent on disk without additional serialization. This also enables the Delphix Engine to achieve higher I/O rates by queueing more I/O operations to its storage.

Requirements	Notes

If using VMDKs:
• Each VMDK should be the only VMDK in its VMFS volume
• The VMFS volumes should be assigned to dedicated physical LUNs on redundant storage.
• The VMDKs should be created with the Thick Provision Lazy Zeroed option.
	• Provisioning VMDKs from isolated VMFS volumes on dedicated physical LUNs:
	  • Reduces contention for the underlying physical LUNs
	  • Eliminates contention for locks on the VMFS volumes from other VMs and/or the ESX Server Console
	  • Enables higher availability of the Delphix VM by allowing vSphere to vMotion the VM to a different ESX host in the event of a failure of the Delphix ESX host
	  • Initialize is intended to offset any 'first-write' penalty for thin provisioned storage
	  • It is not required to wait for the initialize storage device actions to complete before starting new tasks on the engine. For example, dSource data ingestion, snapSync policies, vdb creation, etc… are not blocked by the disk initialization.

The quantity and size of VDMKs or RDMs assigned must be identical across all 4 controllers	• If the underlying storage array allocates physical LUNs by carving them from RAID groups, the LUNs should be allocated from different RAID groups. This eliminates contention for the underlying disks in the RAID groups as the Delphix engine distributes IO across its storage devices.
The physical LUNs used for VMFS volumes and RDMs should be of the same type in terms of performance characteristics such as latency, RPMs, and RAID level.	• The total number of disk drives that comprise the set of physical LUNs should be capable of providing the desired aggregate I/O throughput (MB/sec) and IOPS (Input/Output Operations per Second) for all virtual databases that will be hosted by the Delphix Engine.
The physical LUNs used for VMFS volumes can be thin-provisioned in the storage array.	• If the storage array allocates physical LUNs from storage pools comprising dozens of disk drives, the LUNs should be distributed evenly across the available pools.

Requirements	Notes
For best performance, the LUNs used for RDMs should not be thin-provisioned in the storage array but should be thick-provisioned with a size equal to the amount of storage that will be initially allocated to the Delphix Engine. The RDM can be expanded in the future when more storage is needed.	• Using thin-provisioned LUNs in the storage array for VMFS volumes can be useful if adding storage to the Delphix engine in the future is anticipated. In this case, the LUNs should be thin-provisioned with a size larger than the amount of storage that will be initially allocated to the Delphix Engine. When appropriate, to add more storage to the Delphix engine, use vSphere to expand the size of the VMDKs. Be sure to specify that the additional storage is also thick-provisioned and eager-zeroed.

In addition to making sure the latest VMware patches have been applied, check with the organization's hardware vendor for updates specific to the hardware configurations.

5.7.9.5 Option 2: Elastic Data where Object storage is used for database storage and block storage is used for cache.

We support Elastic Data with on prem object storage for the database storage. Traditional disks as cache are used to reduce latencies for frequently read data and as temporary storage for synchronous writes before the writes are sent to object storage.

- For on prem object storage we currently support storage vendors that confirm to the following
  - s3 REST API compatibility
  - strong read-after-write consistency
  - supports s3 key id and secret access key authentication
  - perpetual key support
- On-premise object storage may require a security certificate to create secure connections between the Delphix engine and object storage. If a certificate is required, then it must be installed on the Delphix engine prior to configuring the storage. Refer to Certificate management (see page 965) for more information.
- Refer to Initial setup (see page 571) for additional details for setting up the Elastic Data Engine.

5.7.10 Additional VMware configuration notes

- Running Delphix inside of vSphere is supported.
- Using vMotion on a Delphix VM is supported.
- Device passthrough is not supported.

5.7.11 Procedure to install an OVA

Use the Delphix-supplied OVA file to install the Delphix Engine. The OVA file is configured with many of the minimum system requirements. The underlying storage for the install is assumed to be redundant SAN storage.

1. Download the OVA file from https://download.delphix.com[164]. You will need a support login from your sales team or a welcome letter.
    a. Navigate to the Delphix Product Releases/<Current Version>/Appliance Images page.
2. Login using the vSphere client to the vSphere server (or vCenter Server) where you want to install the Delphix Engine.
3. In the vSphere Client, click **File**.
4. Select **Deploy OVA Template**.
5. Browse to the OVA file.
6. Click **Next**.
7. Select a **hostname** for the Delphix Engine.
   This hostname will also be used in configuring the Delphix Engine network.
8. Select the **data center** where the Delphix Engine will be located.
9. Select the **cluster** and the **ESX host**.
10. Select one (1) **data store** for the **Delphix OS**. This datastore can be **thin-provisioned** and must have enough free space to accommodate the 127GB comprising the Delphix operating system.
11.
    a. For traditional block storage engines
        i. Select four (4) or more **data stores** for Database Storage for the Delphix Engine. The Delphix Engine will stripe all of the Database Storage across these VMDKs, so for optimal I/O performance, each VMDK must be equal in size and be configured **Thick Provisioned - Eager Zeroed**. Additionally, these VMDKs should be distributed as evenly as possible across all four SCSI I/O controllers.
    b. For Elastic data
        i. The VMDK will be used as cache to reduce latencies for frequently read data and as temporary storage for synchronous writes before the writes are sent to object storage. For optimal I/O performance, each VMDK must be equal in size and be configured **Thick Provisioned - Eager Zeroed.** Make sure the disks satisfy the I/O needs of the engine. Eg: At 500 IOPS per 1GiB ratio, a 32 GiB volume can be configured to have the 16K IOPS limit. Two devices would be sufficient for the instance that requires 30K IOPS. You can always add disks later if the cache is insufficient. You can only reduce the cache by removing disks, so if you think you are over provisioning the cache, increase the number of disks used and reduce the size of each disk so that removal is possible at a later point in time.
12. Select the **virtual network** you want to use.
    If using multiple physical NICs for link aggregation, you must use vSphere NIC teaming. Do not add multiple virtual NICs to the Delphix Engine itself. The Delphix Engine should use a single virtual network. For more information, see Optimal network architecture for the Delphix engine (see page 719).
13. Click **Finish**.
    The installation will begin and the Delphix Engine will be created in the location you specified.

164 https://download.delphix.com/

14. Once the installation has completed, power on the Delphix Engine and proceed with the initial system configuration as described in Setting up network access to the Delphix engine (see page 569)

5.8 Deployment for KVM

5.8.1 Overview

This article outlines the requirements for deploying the Delphix Engine via Linux KVM, as well as recommended configuration parameters.

Contact a Delphix representative to request this capability. Delphix will assist in assuring that all KVM requirements are met to successfully run a Delphix Engine with the most appropriate configurations for the use case.

The Delphix Engine is intensive from both a network and storage perspective. If the Delphix Engine competes with other virtual machines on the same host for resources it will result in increased latency for all operations. As such, it is crucial that your KVM host is not over-subscribed, as this eliminates the possibility of a lack of resources for the Delphix Engine. This includes allowing a percentage of CPU resources for the hypervisor itself as it can de-schedule an entire VM if the hypervisor is needed for managing I/O or compute resources.

The KVM ecosystem includes many versions/variations. Delphix is announcing general support for KVM, but will forgo any formal certification for specific versions and rather will focus on declared support for specific Linux Kernels.

Delphix disk storage capabilities are based on the backend storage provided. Performance, redundancy, and stability characteristics are determined by the hypervisor or Cloud provisioned storage.

5.8.2 Pre-requisites

1. The KVM provider must explicitly state that the Ubuntu 18.04/20.04 kernel is supported by their variation of KVM.

Most, if not all, KVM hypervisor providers should have public facing documentation which include support matrices for guest OS compatibility. If the KVM hypervisor provider does not explicitly state support for the Ubuntu kernel versions used in the supported Linux Delphix versions (either 18.04 currently, or 20.04 in the future), Delphix cannot support the KVM provider.

As an example, RedHat Linux (a KVM hypervisor provider), provides a publicly accessible guest operating system support matrix[165]. As of December 2021, this matrix does not include support for any Ubuntu guest OS, and therefore Delphix would also not support deployment on RedHat Linux KVM. Oracle Linux (a different KVM hypervisor provider) also provides a publicly accessible guest operating system support matrix[166], which declares support for Ubuntu 16.04, 18.04, and 20.04. In this case, since the hypervisor provider declares support for the Delphix Ubuntu kernel(s), Delphix will support deployment on Oracle Linux KVM.

2. A check of the Delphix appliance on the variation/version of KVM.

It is **required** that a Delphix representative works with the organization to check the Delphix appliance (and/or use case) on the specified KVM variation/version. This check will ensure that the software is compatible.

5.8.3 Virtual CPUs

Requirements	Notes
8 vCPUs	• CPU resource shortfalls can occur on an over-committed host as well as competition for host resources during high IO utilization. • CPU reservations are strongly recommended for the Delphix VM so that Delphix is guaranteed the full complement of CPUs even when resources are overcommitted. • It is suggested to use a single core per socket unless there are specific requirements for other VMs on the same KVM host.
Never allocate all available physical CPUs to virtual machines.	• A CPU for the KVM Server to perform hypervisor activities must be set aside before assigning vCPUs to Delphix and other VMs. • We recommend that a minimum of 8-10% of the CPUs available are reserved for hypervisor operation. (e.g. 12 vCPUs on a 128 vCore system).

165 https://access.redhat.com/articles/973163
166 https://docs.oracle.com/en/operating-systems/oracle-linux/kvm-user/kvm-AboutOracleLinuxKVM.html#kvm-linux-guest

5.8.4 Memory

Requirements	Notes
128 GB vRAM (recommended) 64GB vRAM (minimum)	• The Delphix Engine uses its memory to cache database blocks. More memory will provide better read performance. • Memory reservations are required for the Delphix VM. The performance of the Delphix Engine will be significantly impacted by the over-commitment of memory resources in the KVM Server. • Reservations ensure that the Delphix Engine will not be forced to swap pages during times of memory pressure on the host. A swapped page will require orders of magnitude more time to be brought back to physical memory from the KVM swap device.
Memory for the KVM Server to perform hypervisor activities must be set aside before assigning memory to Delphix and other VMs.	• Failure to ensure sufficient memory for the host can result in a hard memory state for all VMs on the host which will result in a block for memory allocations.

5.8.5 Network

Requirements	Notes
Virtual ethernet adapter requirements.	• Jumbo frames are highly recommended to reduce CPU utilization, decrease latency, and increase network throughput (typically 10-20% throughput improvement). • A 10GbE NIC/network is recommended for performance dSource and VDB operations.
If the network load in the KVM Server hosting the Delphix engine VM is high, dedicate one or more physical NICs to the Delphix Engine.	• Adding NICs only works if VMs are discovered using different interfaces. The NFS/iSCSI mounts will only use the network associated with the discovery. • See General Network and Connectivity Requirements (see page 717) for information about specific port configurations, and Network Performance Configuration Options (see page 719) for information about network performance tuning.

5.8.6 SCSI controller

Notes
When adding virtual disks, make sure that they are evenly distributing the load across the maximum of 4 virtual SCSI controllers. Spreading the disks across available SCSI controllers evenly will ensure optimal IO performance from the disks. For example, a VM with 4 SCSI controllers and 8 virtual disks should distribute the disks across the controllers as follows:
disk0 = SCSI(0:0) - System Disk on Controller 0 Port 0 (ignore for purposes of load balancing)
disk1 = SCSI(0:1) - Data Disk on Controller 0 Port 1
disk2 = SCSI(1:1) - Data Disk on Controller 1 Port 1
disk3 = SCSI(2:1) - Data Disk on Controller 2 Port 1
disk4 = SCSI(3:1) - Data Disk on Controller 3 Port 1
disk5 = SCSI(0:2) - Data Disk on Controller 0 Port 2
disk6 = SCSI(1:2) - Data Disk on Controller 1 Port 2
disk7 = SCSI(2:2) - Data Disk on Controller 2 Port 2
disk8 = SCSI(3:2) - Data Disk on Controller 3 Port 2
For load purposes, we generally focus on the Delphix storage (data disks) and ignore the controller placement of the system disk. IDE and SATA controllers are not supported on KVM platforms

5.8.7 General storage

Delphix recommends using a minimum of four disks to run your Delphix Engine. One disk is used for the Delphix File System (DxFS) to ensure that its file systems are always consistent on disk without additional

serialization. The other three or more equally sized disks will be used for data storage. This also enables the Delphix Engine to achieve higher I/O rates by queueing more I/O operations to its storage.

Requirements	Notes
Storage used for Delphix must be provisioned from storage that provides data protection.	For example, using RAID levels with data protection features, or equivalent technology. The Delphix Engine does not protect against data loss originating at the hypervisor or SAN layers. When adding additional disks to the block storage pool, storage device initialization will not occur since it is not required on this hypervisor. For more information refer to, Optimal storage configuration parameters for the Delphix engine (see page 896).

5.9 Deployment for Hyper-V

5.9.1 Overview

This article outlines the requirements for deploying the Delphix Engine on Hyper-V (including supported versions and instance configurations), as well as recommended configuration parameters for optimal performance.

Contact a Delphix representative to request this capability. Delphix will assist in assuring that all Hyper-V requirements are met to successfully run a Delphix Engine with the most appropriate configuration for the use case.

The Delphix Engine is intensive both from a network and a storage perspective. If the Delphix Engine competes with other virtual machines on the same host for resources it will result in increased latency for all operations. As such, it is crucial that your Hyper-V host is not over-subscribed, as this eliminates the possibility of a lack of resources for the Delphix Engine. This includes allowing a percentage of CPU resources for the hypervisor itself as it can de-schedule an entire VM if the hypervisor is needed for managing I/O or compute resources.

> Delphix disk storage capabilities are based on the backend storage provided. Performance, redundancy, and stability characteristics are determined by the hypervisor or Cloud provisioned storage.

5.9.2 Supported versions

- Hyper-V Version: 10.0 and later
- Virtual Machine: Gen 1 only is supported

5.9.3 Virtual CPUs

Requirements	Notes
8 vCPUs	• CPU resource shortfalls can occur on an 0ver-committed host as well as competition for host resources during high IO utilization. • CPU reservations are strongly recommended for the Delphix VM so that Delphix is guaranteed the full complement of CPUs even when resources are overcommitted. • It is suggested to use a single core per socket unless there are specific requirements for other VMs on the same Hyper-V host.
Never allocate all available physical CPUs to virtual machines.	• CPU for the Hyper-V Server to perform hypervisor activities must be set aside before assigning vCPUs to Delphix and other VMs • We recommend that a minimum of 8-10% of the CPUs available are reserved for hypervisor operation. (e.g. 12 vCPUs on a 128 vCore system).

5.9.4 Memory

Requirements	Notes
128 GB vRAM (recommended) 64GB vRAM (minimum)	• The Delphix Engine uses its memory to cache database blocks. More memory will provide better read performance. • Memory reservations are required for the Delphix VM. The performance of the Delphix Engine will be significantly impacted by the over-commitment of memory resources in the Hyper-V Server. • Reservations ensure that the Delphix Engine will not be forced to swap pages during times of memory pressure on the host. A swapped page will require orders of magnitude more time to be brought back to physical memory from the Hyper-V swap device.

Requirements	Notes
Memory for the Hyper-V Server to perform hypervisor activities must be set aside before assigning memory to Delphix and other VMs.	Failure to ensure sufficient memory for the host can result in a hard memory state for all VMs on the host which will result in a block for memory allocations.

5.9.5 Network

Requirements	Notes
Virtual ethernet adapter requirements.	• SR-IOV is recommended for all virtual ethernet adapters that will be used for Delphix data I/O. • Jumbo frames are recommended. • A 10GbE NIC in the Hyper-V Server is recommended.
If the network load in the Hyper-V Server hosting the Delphix engine VM is high, dedicate one or more physical NICs to the Delphix Engine.	• Adding NICs only works if VMs are discovered using different interfaces. The NFS/iSCSI mounts will only use the network associated with the discovery. • See General network and connectivity requirements (see page 717) for information about specific port configurations, and Network performance configuration options (see page 719) for information about network performance tuning.

5.9.6 SCSI controller

Requirements	Notes

IDE Controller for the boot drive
SCSI Controller for database storage
	Per Hyper-V Storage I/O Performance Tuning Guidelines[167]; For optimal performance, it is recommended that you attach multiple disks to a single virtual SCSI controller and create additional controllers only as they are required to scale the number of disks connected to the virtual machine.
	For example, a VM with 3 virtual disks should distribute the disks across the single SCSI controller as follows:
	• IDE Controller 1
	  • Boot Drive
	• SCSI Controller
	  • disk1
	  • disk2
	  • disk3
	For load purposes, we generally focus on the Delphix storage (data disks) and ignore the controller placement of the system disk.

5.9.7 General storage

Delphix recommends using a minimum of four disks to run your Delphix Engine. One disk is used for the Delphix File System (DxFS) to ensure that its file systems are always consistent on disk without additional serialization. The other three or more equally sized disks will be used for data storage. This also enables the Delphix Engine to achieve higher I/O rates by queueing more I/O operations to its storage.

167 https://learn.microsoft.com/en-us/windows-server/administration/performance-tuning/role/hyper-v-server/storage-io-performance

Requirements	Notes
Storage used for Delphix must be provisioned from storage that provides data protection.	For example, using RAID levels with data protection features, or equivalent technology. The Delphix Engine does not protect against data loss originating at the hypervisor or SAN layers. When adding additional disks to the block storage pool, storage device initialization will not occur since it is not required on this hypervisor. For more information refer to, Optimal storage configuration parameters for the Delphix engine (see page 896).

5.9.8 Delphix storage options

There are two types of data that Delphix stores on disk which must be stored on NTFS volumes:

1. Delphix Engine System Disk Storage: This is the system boot drive.
2. Database Storage: stores data used by Delphix objects such as dSources and virtual databases (VDBs).

5.9.8.1 Delphix engine system disk storage

Requirements	Notes
The Delphix Engine disks must be stored on NTFS volume(s).	The volume for the Delphix Engine System Disk Storage is often created on the same volume as the Delphix VM definition. In that case, the volume must have sufficient space to hold the Delphix VM Configuration, the virtual disk for the system disk, and a paging area if a memory reservation was not enabled for the Delphix Engine.
The Delphix .vhdx file is configured for a 128GB system drive.	The volume where the .vhdx is deployed should, therefore, have at least 127GB of free space prior to deploying the .vhdx.

5.9.8.2 Database storage

In addition to making sure the latest Hyper-V patches have been applied, check with your hardware vendor for updates specific to your hardware configuration. VHDXs (virtual machine disks).

Requirements	Notes
A minimum of 3 VHDXs should be allocated for database storage.	Allocating a minimum of 3 VHDXs for database storage enables the Delphix File System (DxFS) to make sure that its file systems are always consistent on disk without additional serialization. This also enables the Delphix Engine to achieve higher I/O rates by queueing more I/O operations to its storage.
If using VHDXs: • Each VHDX should be the only VHDX on its NTFS volume • The VHDX volumes should be assigned to dedicated physical LUNs on redundant storage.	Provisioning VHDXs from isolated volumes on dedicated physical LUNs: • Reduces contention for the underlying physical LUNs • Eliminates contention for locks on the volumes from other VMs and/or the Hyper-V Server Console
	If the underlying storage array allocates physical LUNs by carving them from RAID groups, the LUNs should be allocated from different RAID groups. This eliminates contention for the underlying disks in the RAID groups as the Delphix engine distributes IO across its storage devices.
The physical LUNs used for NTFS volumes should be of the same type in terms of performance characteristics such as latency, RPMs, and RAID levels.	The total number of disk drives that comprise the set of physical LUNs should be capable of providing the desired aggregate I/O throughput (MB/sec) and IOPs (Input/Output Operations per Second) for all virtual databases that will be hosted by the Delphix Engine
Disk types supported by the physical LUNs used for NTFS volumes are: Fixed-size and Dynamically expanding.	If the storage array allocates physical LUNs from storage pools comprising dozens of disk drives, the LUNs should be distributed evenly across the available pools.

5.9.9 Procedure for deploying with Hyper-V

5.9.9.1 Overview

This article outlines the procedure for deploying the Delphix Engine in Hyper-V.

5.9.9.2 Creating a Delphix engine VM

1. Download the image from https://download.delphix.com[168] and copy it to a VM directory.
2. Start the Hyper-V Manager and specify **Name** and **Location**, then select **Next**.
3. Select **Generation 1**, configure memory and then select **Next**.
   Memory: 64 GB (minimum), 128 GB (recommended)
4. Set up Networking by selecting **vNIC**, then select **Next**.
5. Attach the downloaded image as a boot disk. Create a unique boot disk for each image. Please note that boot disks cannot be shared.
   a. Use an existing virtual hard disk.
   b. Browse to the location of VM.
   c. Select the Image.
6. Select **Finish**, the VM will appear in the inventory.

5.9.9.3 Customize the VM by selecting **settings**

1. Delphix recommends having the IDE as the first device to boot from (under BIOS setting).
2. Adjust the number of CPUs (min 8).
3. Add a minimum of four equal-sized **Hard Drives** for the database storage. Use VHDX formatted disks. Recommend Fixed Size disk type. Please note that Differencing Disk Types are not supported.
4. Connect to the console and start the VM.
5. Once the installation has been completed, power on the Delphix Engine and proceed with the initial system configuration as described in Setting up network access to the Delphix engine (see page 569).

5.10 Deployment for AWS EC2

5.10.1 Overview

This article outlines the virtual machine requirements, including memory and data storage, for deploying the Delphix Engine on Amazon EC2 (Elastic Cloud Compute). Once the requirements listed on this page are reviewed, refer to the next articles in the AWS EC2 Deployment topic:

- Prerequisites to Deploying in AWS (see page 629)
- Procedure for Deploying in AWS (see page 631)

168 https://download.delphix.com/

> Delphix disk storage capabilities remain seamlessly operable. There are no additional limitations to your storage, unless otherwise imposed by the [respective] Hypervisor or Cloud environment.

5.10.2 Instance types

The following is a list of instance types that are supported to deploy Delphix on AWS EC2. Delphix periodically certifies new instance types, which will be added to the list here.

Requirements	Notes
Memory Optimized instance families: The minimum requirements are listed below: • 8 vCPUs minimum • 64 GB RAM minimum • 10 Gbps min (25 Gbps recommended) • Enhanced Networking • Processors: Intel or AMD (No ARM/ Graviton) • Storage: SSD or NVMe (No HDD) Recommended instance families: • R4 • R5n • R5b • R6in • R6idn	• Larger instance types provide more CPU, which can prevent resource shortfalls under high I/O throughput conditions. • Larger instances also provide more memory, which Continuous Data uses to cache database blocks. More memory will provide better read performance. • Delphix Elastic Data Engines (engines backed by object storage) will need more bandwidth than traditional engines (backed by EBS) because the s3 bucket used to store data is accessed over a network pipe that is shared with other engines activity. Information on AWS instances[169]

5.10.3 Network configuration

It is recommended to enable Enhanced Networking to maximize performance. For more information view Enhanced Networking on Linux (see page 1343). Note that the enhanced networking driver is included with Delphix software, and no steps need to be taken on the Delphix Engine to enable the feature other than rebooting.

169 https://aws.amazon.com/ec2/instance-explorer/?ec2-instances-cards.sort-by=item.additionalFields.category-order&ec2-instances-cards.sort-order=asc&awsf.ec2-instances-filter-category=*all&awsf.ec2-instances-filter-processors=*all&awsf.ec2-instances-filter-accelerators=*all&awsf.ec2-instances-filter-capabilities=*all

Requirements	Notes
Virtual Private Cloud	The Delphix Engine and all of the source and target environments must be deployed in a VPC network to ensure that private IP addresses are static and do not change when restarting instances. When adding environments to the Delphix Engine, use the host's VPC (static private) IP addresses.
Static Public IP	The EC2 Delphix instance must be launched with a static IP address; however, the default behavior for VPC instances is to launch with a dynamic public IP address – which can change whenever the instance restarts. If using a public IP address for the Delphix Engine, static IP addresses can only be achieved by using assigned AWS Elastic IP Addresses[170]
Security Group Configuration	The default security group will only open port 22 for SSH access. The security group must be modified to allow access to all of the networking ports used by the Delphix Engine and the various source and target engines. See Network Performance Configuration Options (see page 719) for information about network performance tuning. See General Network and Connectivity Requirements (see page 717) for information about specific port configurations.

5.10.4 EBS configuration

Deploying Delphix on AWS EC2 requires EBS volumes. Since EBS volumes are connected to EC2 instances via the network, other network activity on the instance can affect throughput to EBS volumes. Delphix recommends EBS General Purpose SSD (GP2 and GP3) or Provisioned IOPS SSD volumes to provide consistent and predictable storage performance. For more information on EBS volumes, see external documentation href="https://docs.aws.amazon.com/AWSEC2/19.0.0.0/UserGuide/ebs-volume-types.html ">EBS Volume Types.

Recommendations	Notes
EBS Provisioned IOPS SSD Volumes (Highly Recommended)	Delphix does not support the use of instance store volumes. Use EBS volumes with provisioned IOPs in order to provide consistent and predictable performance. The number of provisioned IOPs depends on the estimated IO workload on the Delphix Engine. Provisioned IOPs volumes must be configured with a volume size to provisioned IOPs per the EBS Volume Types[171] guidelines.

170 https://docs.aws.amazon.com/AWSEC2/19.0.0.0/UserGuide/elastic-ip-addresses-eip.html
171 https://docs.aws.amazon.com/AWSEC2/19.0.0.0/UserGuide/ebs-volume-types.html

Recommendations	Notes
EBS General Purpose SSD Volumes	Delphix supports GP2 and GP3 General Purpose SSDs. You can now migrate your Amazon EBS volumes from GP2 to GP3. For more details on migration, see this Amazon documentation[172] on EBS Volumes Migration. Delphix Elastic Data (Engines backed by s3 storage) need gp3 or higher.

5.10.5 General storage configuration

Requirements	Notes
General Storage Configuration	For Delphix Continuous Cloud Engines backed by s3 object storage, read the section ***Delphix Elastic Data Engines (Engines backed by object storage)*** in Initial Setup (see page 571). For Regular engines (backed by EBS): • Allocate initial storage equal to the size of the physical source database storage. For high redo rates and/or high DB change rates, allocate an additional 10-20 %. • All data storage volumes must be EBS volumes. Delphix recommends using a minimum of four equally sized storage disks to run the Delphix Engine. This allows the Delphix File System (DxFS) to assure its file systems are always consistent on disk without additional serialization. This also enables the Delphix Engine to achieve higher I/O rates by queueing more I/O operations to its storage. When adding additional disks to the block storage pool, storage device initialization will not occur since it is not required on this cloud.

5.10.6 Additional AWS configuration notes

- Using storage other than EBS is not supported.
- Limits on the number of volumes are dictated by the EBS instance type, and is generally advised that over 40 can be expected to cause issue on Linux VMs. More information can be found in the AWS Volume Limits[173] and AWS Volume Constraints[174] articles. The maximum device limit imposed by AWS can be handled by the Delphix Engine.

172 https://aws.amazon.com/blogs/storage/migrate-your-amazon-ebs-volumes-from-gp2-to-gp3-and-save-up-to-20-on-costs/

173 https://docs.aws.amazon.com/AWSEC2/19.0.0.0/UserGuide/volume_limits.html

174 https://docs.aws.amazon.com/AWSEC2/19.0.0.0/UserGuide/volume_constraints.html

- Cold HDD (sc1) volumes (not supported due to poor performance)
- Using fast storage for EBS volumes is supported and recommended, including (in order of decreasing speed):
    - Provisioned IOPs SSD volumes (recommended)
    - General Purpose SSD volumes (supported)
    - Throughput Optimized HDD (supported, not recommended due to performance)
- Use of EBS volumes encrypted at creation is supported (during initial deployment from AMI, as well as storage devices added post-deployment), but can have negative performance consequences. Conversion of existing EBS volumes is possible in AWS but is not supported in Delphix at this time.
- Cloud engines are only supported with IMDSV1 at this point in time.

5.10.7 Prerequisites to deploying in AWS

5.10.7.1 Overview

This article outlines the prerequisites for deployment of the Delphix Engine on AWS. The setup user should have experience launching and configuring instances in the Amazon Web Services environment. Review and complete the tasks in the next section before deployment.

5.10.7.2 Prerequisites

1. Review Checklist of information required for installation and configuration (see page 560).
2. Make sure that the Amazon account being used to deploy the Delphix Engine has an appropriate level of enablement to subscribe to the Delphix Engine for AWS subscription.
3. Determine which virtual private cloud (VPC) is being used when launching the virtualization instance. To maximize performance, deploy the Delphix Engine instance in the same VPC/subnet in which the virtual databases (VDBs) will be created.
    a. Provisioning a VDB requires a compute instance running the same database engine as the source. Please note, however, that the target instance only needs storage to accommodate the OS, database platform binaries, etc., because Delphix delivers all of the data files.
4. Make sure that the necessary ports are open.
    a. Using the Delphix Engine for AWS will require connections to source and target database servers. Such connections require various ports to be open, enabling communications. For a detailed list of the network and port requirements, click the link that corresponds with the relevant database platform:
        i. Network and connectivity requirements for oracle environments (see page 1399)
        ii. Network access requirements for SQL server environments (see page 2089)
        iii. Network and connectivity requirements for SAP ASE base environments (see page 1892)
        iv. Network and connectivity requirements for Db2 environments (see page 1020)
5. Update Security Group settings to accommodate the necessary connections.

   a. Select the same Security Group that the current (or future) non-production EC2 compute nodes utilize.
   b. Modify the Security Group to allow access to all of the networking ports used by the Delphix Engine and the various source and target platforms. See links above for information about specific port configurations.

6. Allocate storage.
   a. To properly size the initial storage capacity and determine the number and size of EBS Provisioned IOPs Volumes required, download and utilize the Delphix-dynamic-data-platform-storage-calculator[175].
      It is helpful to first create a list of the data sources intended for making dSources. A data source is typically a production database linked to the Virtualization Engine, enabling to create virtual, full, read-write copies of the source within minutes. The list should include the database name, platform (for example, Oracle or SQL Server), current size (in GB), the estimated number of virtual copies, and retention period (in days) of snapshots (backup copies).
   b. All data storage volumes must be EBS volumes. Delphix recommends using a minimum total of four disks to run the Delphix Engine. One disk is used for the boot device. The other four equally sized disks will be used for data storage. This also enables the Delphix Engine to achieve higher I/O rates by queueing more I/O operations to its storage.
   c. Provisioned IOPs EBS volumes are highly recommended.

7. During the Manual Deployment option, use the guidelines outlined in Virtual machine requirements for AWS EC2 platform (see page 625).

5.10.7.3 Geographic distribution in regions and availability zones

The latency will be directly related to not just the Availability Zone configuration, but more specifically the geographies of those zones. The latencies can vary from tens to hundreds of milliseconds if the zones are geographically diverse (US West to US East would certainly be expected to perform better than US West to Europe). AWS advertises that all AZs in a given region are interconnected with high-bandwidth and low-latency networking per this AWS article[176].

Applications that are performance-sensitive will benefit from colocating the target servers and Engine in the same region if possible. Another possibility is building a failover strategy for those highly sensitive servers, enabling them to failover to another AZ in the instance where an Engine needs to be failed over. The architecture selected and geographies will also naturally be dependent on your redundancy requirements (your organization may require geographically diverse failover options beyond the ~60 miles advertised within a region).

175 https://documentation.delphix.com/continuous-data-11-0-0-0/docs/overview-of-data-virtualization
176 https://aws.amazon.com/about-aws/global-infrastructure/regions_az/#Availability_Zones

5.10.8 Procedure for deploying in AWS

5.10.8.1 Overview

This article outlines the procedure for deploying the Delphix Engine on AWS, using the Delphix Marketplace Image or Amazon Machine Image (AMI).

5.10.8.2 Procedure to install an AMI

Use the Delphix-supplied AMI file to install the Delphix Engine.

1. On the Delphix download site, select the AMI to be shared and accept the Delphix License agreement. Alternatively, use a link provided by the organization's Delphix solutions architect.
2. On the **Amazon Web Services Account Details** form presented:
   a. Enter the AWS Account Identifier. To use the **GovCloud AWS Region**, be sure to enter the ID for the AWS Account which has GovCloud enabled.
   b. Select which **AWS Region** that the AMI will be shared in. For an AMI shared in a different region, contact the organization's Delphix account representative to make the proper arrangements.
3. Click **Share.** The Delphix Engine will appear in the list of AMIs in AWS momentarily.

5.10.8.3 Subscribe to the Delphix virtualization engine marketplace image

1. Login to the AWS Console at https://aws.amazon.com/console.
2. Navigate to the AWS Marketplace.
3. In the **Search** field, enter **Delphix**.
4. Select the **Delphix DevOps Data Platform for AWS**. Select the appropriate offer based on the data you will ingest into the Delphix platform.
5. Review the information on the initial Marketplace page.
6. Click **Continue to Subscribe**.
7. Review the software subscription information and select the **Manual Launch** tab.
8. Select **Accept Software Terms**. Your subscription will take a few minutes to become enabled.
9. From the **Subscription** page, click **Return to Product Page**. There you will see **Launch with EC2 Console** enabled in all of the available regions.

5.10.8.4 Launching the Delphix engine

1. In the desired region, select **Launch with EC2 Console**. Launching will redirect to the standard EC2 instance launch process.
2. Select the size of the virtual machine to deploy. For supported instance types and capabilities, see Virtual machine requirements for AWS platform (see page 625).
3. Click **Next: Configure Instance Details**.
4. Configure instance details as follows:
   a. Set the **number of instances** to **1**.
   b. **Purchasing option** – Leave **Request Spot Instances** unchecked.
   c. **Network** – Select the VPC into which the instance will be deployed.
   d. **Subnet** – Select the Subnet into which the instance will be deployed.
   e. **Auto-assign Public IP** – Select **Disable**.
   f. **Placement group** – Optional. The default is **No placement group**.
   g. **IAM Role** – None
   h. **Shutdown behavior** – Stop
   i. **Enable termination protection** – Yes (select checkbox). This ensures that the Delphix Engine is not accidentally terminated.
   j. **Monitoring** – Optional
   k. **EBS-optimized instance** – Yes (select checkbox)
   l. **Tenancy** – Shared – Run a shared hardware instance
   m. **Network Interfaces**– Eth0
      Note : This option will not appear if a default subnet is chosen. In this case, the first option below will apply.
      i. Auto-assign (default if using DHCP)
      ii. Enter static IP in Primary IP field (recommended for consistency)
5. Click **Next: Add Storage**.
6. By default, the Delphix Virtualization instance includes a 150GB OS volume. Ensure that this volume is set as a **General Purpose SSD (GP2)** type.
7. For traditional Block storage based engines

a. Using the Delphix Virtualization E... [177], enter information from the predetermined inventory into the worksheet. This will automatically calculate the number, size, and IOPS required for the EBS volumes.

Delphix Virtualization Engine - Storage Calculator for AWS Marketplace					
DB Name	**DB Type**	**DB Source Size (GB)**	**Est. # of Virtual DBs**	**Retention for Snapshots (Days)**	**Delphix Storage (GB)**
oradb1	Oracle	2500	5	14	3000
sqldb1	MS-SQL	1500	5	14	1800
sqldb2	MS-SQL	150	5	14	180
					0
					0
Totals		4150			4980
# of EBS Volumes					4
Size per Volume					1250
IOPs per Volume					1000

In the above example, there are three (3) source databases listed with their current sizes. Each is estimated to have 5 virtual copies as well as a 14-day retention period for snapshots/backups. The calculator indicates to add four (4) EBS-provisioned IOPS volumes, each being 1250GB with 1000 IOPS.

Note

This calculator is only for estimating the anticipated storage size and IOPS needs. A manual adjustment may be required prior to or after launch depending on storage and performance needs.

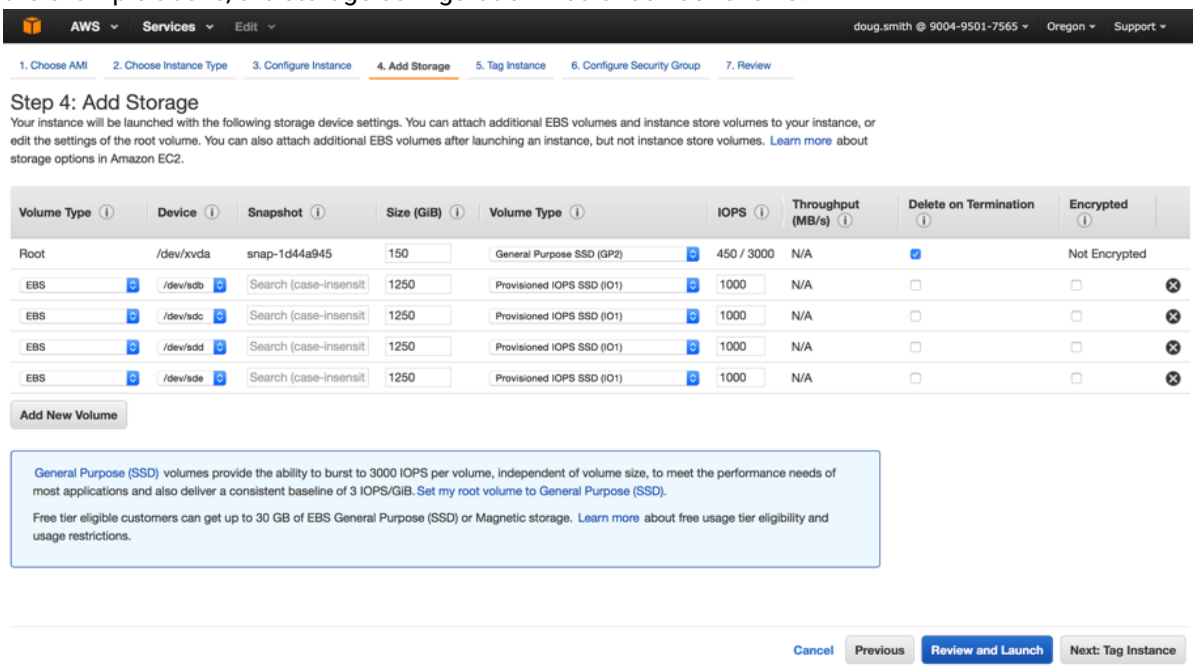
b. For Elastic Data Engines (Refer to Initial setup (see page 571) for additional details) s3 bucket acts as the storage and EBS disks as cache are used to reduce latencies for frequently read data and as temporary storage for synchronous writes before the writes are sent to object storage. Make sure the disks satisfy the I/O needs of the engine. You can always add disks later if the cache is insufficient. You can only reduce the cache by removing disks, so if you think you are over provisioning the cache, increase the number of disks used and reduce the size of each disk so that removal is possible at a later point in time. When selecting disks for Elastic Data Engines:

177 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149495932?preview=%2F149495932%2F149495951%2FDelphix+Virtualization+Engine+Storage+Calculator+for+AWS.xlsx

i. gp3 disks are recommended as they offer good performance at a lower cost. At 500 IOPS per 1GiB ratio, a 32 GiB volume can be configured to have the gp3 volume 16K IOPS limit and 1000 MB/s throughput. For reference, r5n.8xlarge instance has a 30K IOPS and 850 MB/s throughput limits so two gp3 devices would be sufficient for the instance.
ii. io2 disks have lower latency at a higher cost. However, the lower latency is not beneficial once the instance IOPS or throughput limit is reached.

8. Click **Add New Volume**. The new volume will be created using the specifications in the estimation provided. The below values are sample values set for a Volume Type. See EBS configuration (see page 625) for related information.
   a. **Volume Type** – EBS
   b. **Device** – Use default provided
   c. **Size (GiB)** – 1250
   d. **Volume Type** – Provisioned IOPS SSD (IO1) Delphix deploys as an EC2 instance on AWS using EBS volumes for storage. AWS provides the following options for Elastic Block Storage (EBS) storage volumes, namely
      i. Standard
      ii. General Purpose SSD (GP2)
      iii. General Purpose SSD (GP3)
      iv. Provisioned IOPS SSD (IO1)
   e. **IOPS** – 1000
   f. **Delete on Termination** – Optional. If this option is not set and the Delphix Engine is terminated, the EBS volumes will persist, and they will need to be removed manually.
   g. **Encrypted** – Optional

9. Follow the above procedure for adding as many EBS volumes as specified in the calculator. Based on the example above, the storage configuration would look as follows:



10. Verify the storage configuration matches the output from the calculator.
11. Click **Next: Tag Instance**.
12. To provide a name for the Delphix Virtualization Instance, enter it in the **Value** field corresponding to the **Key Name**.
13. Click **Next: Configure Security Group**.
14. Assign a security group. Do one of the following:
    a. Create a new security group.
    b. If an existing security group meets Delphix requirements, it can be selected. For more details on required and recommended ports for general Delphix usage, see General network and connectivity requirements (see page 717).
15. Click **Review and Launch**.
16. Verify the Delphix Virtualization Instance details and click **Launch**.
17. Proceed without a key pair.
18. Select the **I acknowledge** checkbox.
19. Click **Launch Instances**.
20. Click **View Instances**.
21. To complete the initial Delphix Virtualization Engine configuration, wait for the instance to be up and running. The status is available on the **Instances** page. Note that it can take up to 15 minutes for the first launch of Delphix, and even if the status of the server is “running” it may not yet be ready.

5.10.8.5 Configuring the Delphix engine

1. Connect to the running Delphix instance with a web browser. Use the IP address or DNS name noted in the Instance Description.
2. Upon successful connection, the browser will automatically redirect to the **Delphix Setup Page**.
3. Refer to the standard product deployment instructions (see page 571) to complete the Delphix deployment.

5.10.8.6 Logging in for the first time

On the first time login to the Delphix Engine, follow these steps:

1. Enter the default Administration User: **sysadmin**.
2. Enter **sysadmin** for the password (when installing a new engine via AWS AMI, the initial sysadmin password is the AWS Instance ID).
   Note
   Find the <*Instance ID*> in the **Instances** section of the AWS EC2 Management Console.
3. Once redirected, enter the following as prompted:
    a. **Email address** for the admin account.

b. A new **password**. This password will be used for the Engine Administrator with the username "admin".
c. Confirm the new password.
Note:
From now on, this is the password used to log in to the Delphix Engine Administrator interface to manage Datasets.

4. Click **Continue**.

5.10.8.7 Next steps

Congratulations! The Delphix Virtualization Engine should be successfully deployed in AWS.

Use Delphix documentation to learn how to:

- configure a database source
- configure target environments
- create virtual databases (VDBs)

5.10.9 AWS RDS Custom for Oracle and SQL Server

5.10.9.1 Overview

The Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It allows organizations to automate time-consuming administration tasks such as hardware provisioning, database setup, patching, and backups. It frees focus that can be utilized on applications, eliminating extra overhead. Amazon RDS is available for Oracle and SQL Server, amongst other database services.

AWS has announced a new version of their RDS databases, thus, Delphix is adding AWS RDS Custom support for Oracle. AWS RDS Custom is an RDS service that allows Delphix to access the operating system and database level functions required to provision vDBs into the AWS RDS PaaS (Product as a Service) database service.

With AWS RDS Custom, organizations using Delphix can obtain the full benefits of PaaS and Delphix automation for their target environments.

- Users ingest data from production in RDS Oracle/SQL Server, mask the data, and provision secure copies back to RDS. All Delphix automation tools work and also manual operations via the UI.
- Delphix data compression and block sharing capabilities will allow organizations to potentially reduce their AWS RDS storage consumption.

> Support for AWS RDS Custom is granted only on supported database and operating system versions.

Contact a Delphix representative to request this capability.

5.11 Deployment for Microsoft Azure

5.11.1 Overview

This article outlines the virtual machine requirements, including memory and data storage, for deploying the Delphix Engine on the Azure Public Cloud and Government Cloud. Once the requirements listed on this page are reviewed, refer to the next articles on Azure deployment:

- Prerequisites to deploying in Microsoft Azure (see page 640)
- Procedure for deploying in Microsoft Azure (see page 642)

> Delphix disk storage capabilities remain seamlessly operable. There are no additional limitations to your storage, unless otherwise imposed by the [respective] Hypervisor or Cloud environment.

5.11.2 Instance types

Requirements	Notes
Instance families: The minimum requirements are listed below: • 8vCPUs minimum • 64GB RAM minimum • Network Bandwidth: 10Gbps min (25Gbps recommended) • Azure Accelerated Networking • Processors: Intel or AMD (No ARM) • Storage • SSD or NVMe (No HDD) Recommended instance families: • Dsv2-series • Dsv3-series • Esv3-series • Esv4-series • Esv5-series	• Larger instance types provide more CPU, which can prevent resource shortfalls under high I/O throughput conditions. • Larger instances also provide more memory, which the Delphix Engine uses to cache database blocks. More memory will provide better read performance. • Delphix Elastic Data Engines (engines backed by blob storage) will need more bandwidth than traditional engines (backed by block storage) because the blob bucket used to store data is accessed over a network pipe that is shared with other engines activity. • Information on Azure instances[178]

5.11.3 Network configuration

The use of Azure Accelerated Networking is supported for Delphix Engines and is recommended to maximize network performance. Accelerated Networking provides more bandwidth with more consistent and lower network latencies.

178 https://learn.microsoft.com/en-us/azure/virtual-machines/sizes

Requirements	Notes
Azure virtual network (VNet)	The Delphix Engine must have <= 1 ms of latency to the Target and Staging environments. Longer latencies are permitted between the Delphix Engine and Source environments, but may impact Snapsync timings. VNET peering may be used, as long as latency requirements are met. For very low latency requirements, consider Proximity Placement Groups for Delphix Engine, Target, and Staging.
Network security group (NSG)	The security group must be modified that allows access to all networking ports used by the Delphix Engine and the various source/target platforms. See General network and connectivity requirements (see page 717) for information about specific port configurations. See Network performance configuration options (see page 719) for information about network performance tuning.

5.11.4 Storage configuration

Requirements	Notes
General Storage Configuration	• Allocate initial storage equal to the size of the physical source databases. For high redo rates and/or high DB change rates, allocate an additional 10-20% storage. • Add storage when storage capacity approaches 30% free. • Delphix recommends using a minimum of four equally sized storage disks to run the Delphix Engine. This allows the Delphix File System (DxFS) to assure its file systems are always consistent on disk without additional serialization. This also enables the Delphix Engine to achieve higher I/O rates by queueing more I/O operations to its storage. • Maximize Delphix Engine RAM for a larger system cache to service reads. • When adding additional disks to the block storage pool, storage device initialization will not occur since it is not required on this cloud. • See, Delphix storage migration (see page 768).

Requirements	Notes
Azure Premium Storage	• Premium storage utilizes solid-state drives (SSDs). Ultra disk is also supported. • Devices up to 32TB are supported. • A maximum of 256TB is supported. • Cache setting: For Data disks, ReadWrite. For Elastic Data cache disks, NONE. • I/O requests of up to 256 kilobytes (KB) are counted as a single I/O operation (IOP) for provisioned IOPS volumes. • IOPS vary based on storage size with a maximum of 20,000 IOPS. • For Delphix Continuous Cloud Engines backed by object storage, read the section ***Delphix Elastic Data Engines (Engines backed by object storage)*** in Initial Setup (see page 571).

5.11.5 Extensions

Azure VM Extensions are not currently supported.

5.11.6 Prerequisites to deploying in Azure

5.11.6.1 Overview

This article outlines the prerequisites for deployment of the Delphix Engine on Azure. The setup user should have experience launching and configuring instances in the Microsoft Azure environment. Review and complete the tasks in the next section before deployment.

5.11.6.2 Prerequisites

1. Review Checklist of information required for installation and configuration (see page 560).
2. Determine which virtual private cloud (VPC) is being used when launching the Virtualization instance. To maximize performance, deploy the Delphix Engine instance in the same VPC/subnet where the virtual databases (VDBs) will be created.

> Provisioning a VDB requires a compute instance running the same database engine as the source. Please note, however, that the target instance only needs storage to accommodate the OS, database platform binaries, etc., because Delphix delivers all of the data files.

3. Make sure that the necessary ports are open.

> Using the Delphix Engine for Azure will require connections to source and target database servers. Such connections require various ports to be open, enabling communications. For a detailed list of the network and port requirements, click the link that corresponds with the relevant database platform:
>
> 1. Network and Connectivity Requirements for Oracle Environments (see page 1399)
> 2. Network Access Requirements for SQL Server (see page 2089)

4. Update Security Group settings to accommodate the necessary connections.

   a. Select the same **Network Security Group** that the current (or future) non-production Azure virtual machines utilize.

   b. Modify the Network Security Group to allow access to all of the networking ports used by the Delphix Engine and the various source and target engines. See links above for information about specific port configurations.

5. Allocate storage.

> It is helpful to first create a list of the data sources intended for making dSources. A data source is typically a production database linked to the Virtualization Engine, enabling to create virtual, full, read-write copies of the source within minutes. The list should include the database name, platform (for example, Oracle or SQL Server), current size (in GB), the estimated number of virtual copies, and retention period (in days) of snapshots (backup copies).

   a. All data storage disks must be comprised of Azure Premium Storage devices. Delphix recommends using a minimum total of five disks to run the Delphix Engine. One disk is used for the boot device. The other four equally sized disks will be used for data storage. This also enables the Delphix Engine to achieve higher I/O rates by queueing more I/O operations to its storage.

   b. Azure premium storage devices have different levels of guaranteed IOPs that vary from 120 IOPs to 20,000 IOPs. Please refer to Microsoft's documentation[179] to ensure that the selected devices will meet the requirements for your deployment. UltraHD disk storage is supported for performance-sensitive customers.

   c. It is highly recommended to use Azure-managed disks.

   d. Use the guidelines outlined in Deployment for Microsoft Azure (see page 637)

179 https://docs.microsoft.com/en-us/azure/storage/storage-premium-storage#scalability-and-performance-targets

5.11.7 Procedure for deploying in Azure

5.11.7.1 Overview

This article outlines the procedure for deploying the Delphix Engine in Microsoft Azure.

5.11.7.2 Deploying the Delphix engine

If you are deploying a cloud engine on Azure Blob please read the Delphix Elastic Data Engines Initial setup (see page 571) section before deploying the cloud engine.

1. Navigate to the Azure Marketplace at https://[180]azuremarketplace.microsoft.com[181]
2. In the **Search** field, enter **Delphix**.
3. Select the **Delphix Virtualization for Azure (3TB)** for an hourly subscription or **Delphix DataOps Platform for Azure** for customers with existing licenses.
4. Review the information on the initial Marketplace page.
5. Click **GET IT NOW**.
6. Click **Create**.
7. Follow the creation wizard to deploy the Delphix Engine.
8. Provide the basic information for the Delphix Engine.
   a. Select a name for the virtual machine.
   b. For the OS disk type, select **Premium SSD**.
   c. Enter a **username** and **password**.
      Info : This information is never used but is part of the built-in wizard used by the marketplace. Any information provided in these fields is discarded.
   d. Select your **Subscription**, **Resource group**, and **Location**.
   e. Click **OK**.
9. Select the **Size** of the virtual machine you want to deploy. For supported instance types and capabilities, See Virtual machine requirements for Azure platform (see page 637)

   Each instance type will determine the limits for:
   - Network bandwidth
   - Maximum data disk
   - Total IOPS
10. Click **Select**.
11. Configure the **Settings** for this virtual machine:
    a. Use managed disks.

---

180 http://https/portal.azure.com
181 http://azuremarketplace.microsoft.com

   b. Select or create a **Virtual Network** and **Subnet**.
   c. Optional: select a **Public IP address** (this will make an engine accessible over the internet)
   d. Select or create a **Network Security Group**.
   e. **Extensions** and **High Availability sets** are not currently supported. Please ensure that no extensions or availability sets have been configured.
   f. Enable **Boot diagnostics**.

12. Click **OK**.
13. Review the configuration in the **Summary** screen and click **OK**.
14. Review the offer details in the **Buy** screen and click **Purchase** to deploy the Delphix Engine.

5.11.7.3 Configuring the Delphix engine virtual machine

1. From the **Azure Portal** sidebar, select **Virtual Machines**. The running virtual machines will be displayed.
2. Click the virtual machine's **name** to open its detailed information.
3. Click **Disks** to open the detailed panel.
4. Click **Add data disk**.
5. Create a new device:
   a. In the **Create disk** drop-down menu, select **Create managed disk**. This will open the managed disk wizard.
   b. In the wizard,
      i. For block engine
         1. ensure that the **Account type** indicates **Premium (SSD)** and that the **Source Type** shows **None (empty disk)**.
      ii. For Elastic Data engine
         1. Ultra disks are recommended as they can be configured to have high IOPS and throughput at relatively small sizes. At 300 IOPS per GiB ratio, a 256 GB volume can have 76,800 IOPS and 4000 MB/s throughput.
         2. Premium SSD disks can have high performance though they need to be much larger, e.g. P80 32TiB volume has 20K IOPS and 900 MB/s. The new Premium SSD v2 disks have better IOPS per GiB ratio though they’re not available in all regions
         3. You can always add disks later if the cache is insufficient. You can only reduce the cache by removing disks, so if you think you are over provisioning the cache, increase the number of disks used and reduce the size of each disk so that removal is possible at a later point in time.
   c. Select the size of the device to create. Note that the device size will determine the IOPS limit for that device.
   d. Click **Create**.
6. Once the device is created, change the **Host Caching** parameter for that device to **None**.
7. Repeat the above steps to create additional disks.

8. Once adding all the disks is complete, click **Save**.
9. Optional: modify the virtual machine's **Network Security Group**.
    a. From the side panel of the virtual machine's detail screen, click **Network interfaces**.
    b. Click the name of the network interfaces to modify.
    c. From the **Network interface** details window, click the **Network security group** specified in the **Overview** pane. In the example below, the user would click **delphix-engine-nsg**:
    d. From the **Network security group** details pane, modify the **Inbound security rules** and/or the **Outbound security rules** by selecting from the side panel and clicking **Add**.
    e. Add the appropriate rule and click **OK**. The screenshot below shows adding the inbound rule for the Delphix Session Protocol.
    f. Repeat the above steps for all the rules that should be added.

5.11.7.4 Configuring the Delphix engine

1. Connect to the running Delphix instance with a web browser. Use the Public IP address or DNS name noted in the **Network Interface** Panel. Upon successful connection, the browser will automatically redirect to the **Delphix Setup Page**.
2. Refer to the standard product deployment instructions (see page 571) to complete the Delphix deployment.

5.11.7.5 Next steps

Congratulations! The Delphix Virtualization Engine should be successfully deployed in Azure.

Use Delphix documentation to learn how to:

- configure a database source
- configure the target environments
- create virtual databases (VDBs)

5.11.8 Enabling Azure accelerated networking

5.11.8.1 Overview

This article describes how to enable Azure Accelerated Networking. If a VM was created without Accelerated Networking, enabling this feature on an existing VM is possible. The VM must meet the following prerequisites:

- VM must be a supported size for Accelerated Networking
- VM must be a supported Azure Gallery image (and kernel version for Linux)
- All individual VMs or VMs in an availability set must be stopped/deallocated before enabling Accelerated Networking on any NIC

For more information, please see the Azure documentation[182]

5.11.8.2 Individual VMs and VMs in an availability set

Stop/deallocate the VM or, if an Availability Set, all the VMs in the Set:

- If the VM was created individually (without an availability set), only the individual VM needs to stop/deallocate to enable Accelerated Networking.
- If the VM was created with an availability set, all VMs contained in the availability set will need to stop/deallocate before enabling Accelerated Networking on any Network Interface Card (NIC).

Azure CLI

```
az vm deallocate \
    --resource-group myResourceGroup \
    --name myVM
```

Once stopped, enable Accelerated Networking on the NIC of the VM:

Azure CLI

```
az network nic update \
    --name myNic \
    --resource-group myResourceGroup \
    --accelerated-networking true
```

Restart the VM (or all VMs if using an availability set) and confirm that Accelerated Networking is enabled:

Azure CLI

```
az vm start --resource-group myResourceGroup \
    --name myVM
```

After the restart, the Mellanox VF (Virtual Function) device will be exposed to the VM.

182 https://docs.microsoft.com/en-us/azure/virtual-network/create-vm-accelerated-networking-cli

5.12 Deployment for Google cloud platform

5.12.1 Overview

This article covers the virtual machine requirements, including memory and data storage, for the deployment of the Delphix Engine on Google Cloud Platform (GCP). Once the requirements listed on this page are reviewed, refer to the next articles on GCP deployment:

- Prerequisites to deploying in GCP (see page 648)
- Procedure for deploying in GCP (see page 650)

> Delphix disk storage capabilities remain seamlessly operable. There are no additional limitations to your storage, unless otherwise imposed by the [respective] Hypervisor or Cloud environment.

5.12.2 Machine types

The following is a list of instance types that are supported to deploy Delphix on GCP. Delphix periodically certifies new instance types, which will be added to the list here.

Recommendations	Notes
General-purpose machine families. The minimum requirements are listed below: • 8vCPUs minimum • 64GB RAM minimum • 10Gbps min (25Gbps recommended) • Processors: Intel or AMD (No ARM) • Storage • SSD or NVMe (No HDD) Recommended instance families: • N2 standard • N2 high-mem	• The Delphix Engine most closely resembles a storage appliance and performs best when provisioned using a storage-optimized instance type. • Larger instance types provide more CPU, which can prevent resource shortfalls under high I/O throughput conditions. • Larger instances also provide more memory, which the Delphix Engine uses to cache database blocks. More memory will provide better read performance. Information on GCP general-purpose machines[183]

5.12.3 Network configuration

Requirements	Notes
Virtual private cloud	The Delphix Engine and all of the source and target environments must be deployed in a VPC network to ensure that private IP addresses are static and do not change when restarting instances. When adding environments to the Delphix Engine, use the host's VPC (static private) IP addresses.
Static public IP	The GCP Delphix instance must be launched with a static IP address; however, the default behavior for VPC instances is to launch with a dynamic public IP address – which can change whenever the instance restarts. If using a public IP address for the Delphix Engine, static IP addresses can be assigned using the Google cloud documentation[184]

183 https://cloud.google.com/compute/docs/general-purpose-machines
184 https://cloud.google.com/compute/docs/ip-addresses/reserve-static-external-ip-address

Requirements	Notes
Security group configuration	The default security group will only open port 22 for SSH access. The security group must be modified to allow access to all of the networking ports used by the Delphix Engine and the various source and target engines. See Network performance configuration options (see page 719) for information about network performance tuning. See General network and connectivity requirements (see page 717) for information about specific port configurations.

5.12.4 Storage configuration

Requirements	Notes
General storage configuration	Allocate initial storage equal to the size of the physical source database storage. For high redo rates and/or high DB change rates, allocate an additional 10-20 %. Add storage when storage capacity approaches 30% free. Delphix recommends using a minimum of four equally sized storage disks to run the Delphix Engine. This allows the Delphix File System (DxFS) to assure its file systems are always consistent on disk without additional serialization. This also enables the Delphix Engine to achieve higher I/O rates by queueing more I/O operations to its storage. When adding additional disks to the block storage pool, storage device initialization will not occur since it is not required on this cloud. For Delphix Continuous Cloud Engines backed by object storage, read the section ***Delphix Elastic Data Engines (Engines backed by object storage)*** in Initial Setup (see page 571). Block storage for caching requires **SSD Persistent Disk** or higher performance volumes for caching.

5.12.5 Prerequisites to deploying in GCP

5.12.5.1 Overview

This article outlines the prerequisites for deployment of the Delphix Engine on GCP. The setup user should have experience launching and configuring instances in the Google Cloud Platform environment. Review and complete the tasks in the next section before deployment.

5.12.5.2 Prerequisites

1. Review the Checklist of information required for installation and configuration (see page 560)
2. Make sure that the Google account being used to deploy the Delphix Engine has an appropriate level of enablement for the **Delphix virtualization for GCP** subscription.
3. Determine which virtual private cloud (VPC) is being used when launching the virtualization instance. To maximize performance, deploy the Delphix Engine instance in the same VPC/subnet in which the virtual databases (VDBs) will be created.
   Note:

   Provisioning a VDB requires a compute instance running the same database engine as the source. Please note, however, that the target instance only needs storage to accommodate the OS, database platform binaries, etc., because Delphix delivers all of the data files.
4. Make sure that the necessary ports are open.
   Note:

   Using the Delphix Engine for GCP will require connections to source and target database servers. Such connections require various ports to be open, enabling communications. For a detailed list of the network and port requirements, click the link that corresponds with the relevant database platform.
   a. Network and Connectivity Requirements for Oracle Environments (see page 1399)
   b. Network and Connectivity Requirements for SQL Server Environments (see page 2089)
   c. Network and Connectivity Requirements for SAP ASE Base Environments (see page 1884)
   d. Network and Connectivity Requirements for Db2 Environments (see page 1020)
5. Update Security Group settings to accommodate the necessary connections.
   a. Select the same Security Group that the current (or future) non-production EC2 compute nodes utilize.
   b. Modify the Security Group to allow access to all of the networking ports used by the Delphix Engine and the various source and target platforms. See links above for information about specific port configurations.
6. Allocate storage.
   a. To properly size the initial storage capacity and determine the number and size of Provisioned IOPs Volumes required, first create a list of the data sources intended for making dSources. A data source is typically a production database linked to the Virtualization Engine, enabling to create virtual, full, read-write copies of the source within minutes. The list should include the database name, platform (for example, Oracle or SQL Server), current size (in GB), the estimated number of virtual copies, and retention period (in days) of snapshots (backup copies).
   b. Delphix recommends using a minimum total of four disks to run the Delphix Engine. One disk is used for the boot device. The other four equally sized disks will be used for data storage. This also enables the Delphix Engine to achieve higher I/O rates by queueing more I/O operations to its storage.

5.12.5.3 Additional GCP configuration notes

- Delphix supports both Zonal and Regional SSD persistent disks.

5.12.6 Procedure for deploying in GCP

5.12.6.1 Overview

This article outlines the procedure for deploying the Delphix Engine using a tar.gz file in Google Cloud Console or in Google Cloud Marketplace.

5.12.6.2 Deployment in Google cloud console

1. Download the latest **Delphix Platform for GCP** tar.gz file from https://download.delphix.com/.
2. Create a **GCP storage bucket** and upload the tar.gz file from the first step.
3. Once the upload is complete, navigate to **Compute Engine > Images** and select **Create Image.**
4. In the **Create Image** dialog, provide the following information.
    a. Image Name
    b. Source: Cloud Storage File
    c. Select the bucket and newly uploaded Delphix File.
    d. Location: Multi-Regional or Regional. (To deploy Delphix in multiple regions - select **Multi-Region**)
    e. Family: Optional
    f. Description: Optional
    g. Labels: Optional
    h. Encryption: Select based on the organization's policy
    i. Click **Create**. (Creating the image could take some time)
5. Once the image creation is complete, select the newly created image and create an instance from it.
6. Configure the instance creation screen with the following items.
    a. Identify and select the region and zone to run Virtual Databases.
    b. Choose a supported machine type, commensurate with the expected workload.
    c. Change/modify the **Boot Disk** to **SSD Persistent Disk**. (Defaults to **Standard Persistent Disk**)
    d. Add **Data Disks** (minimum of four recommended) with a total size at least equal to the total source DB sizes. Be sure to utilize SSD Persistent Disks.
    e. Update Networking, commensurate with the Network and Subnet(s) on which the target non-production instances reside. Note, a “target” is an instance running the DB platform identical to the source DBs.
    f. Once all of the necessary instance information has been configured, select **Create**.
    g. Wait for the instance to be created/available and connect to the newly deployed Engine using the assigned IP/hostname via the support web browser.

5.12.6.3 Deployment in Google cloud marketplace

1. Login to the **Google Cloud Marketplace**.
2. In the **Search** field, enter **Delphix**.
3. Select the **Delphix Data Virtualization for GCP (3TB)**.
4. Review the information on the initial Marketplace page.
5. Click **Launch**.
6. Review the deployment configuration and software subscription information.
7. Accept Google Cloud's and Delphix Corp's Terms of Service.
8. Click **Deploy**, which will start the deployment of the instance.
9. After the deployment, add equally-sized data disks to the instance.

5.12.6.4 Configuring the Delphix engine

1. Connect to the running Delphix instance with a web browser. Use the IP address or DNS name noted in the Instance Description. Upon successful connection, the browser will automatically redirect to the **Delphix Setup Page**.
2. Refer to the standard product deployment instructions (see page 571) to complete the Delphix deployment.

5.12.6.5 Next Steps

Congratulations! The Delphix Virtualization Engine should be successfully deployed in Google Cloud Platform.

Use Delphix documentation to learn how to:

- configure a database source
- configure the target environments
- configure virtual databases (VDBs)

5.13 Deployment for OCI

5.13.1 Overview

This article outlines the virtual machine requirements, including memory and data storage, for deploying the Delphix Engine on Oracle Cloud Infrastructure (OCI). Once the requirements listed on this page are reviewed, refer to the next articles on OCI deployment:

- Prerequisites to deploying in OCI (see page 655)

- Procedure for deploying in OCI (see page 656)

> Delphix disk storage capabilities remain seamlessly operable. There are no additional limitations to your storage, unless otherwise imposed by the [respective] Hypervisor or Cloud environment.

5.13.2 Compute image types

Delphix distributes product images for OCI using the QCOW2 image type. Compute Images must be imported into OCI using the Paravirtualized launch mode; currently, images using the Emulated launch mode are not supported.

5.13.3 Supported shapes

The following is a list of shapes that are supported to deploy Delphix on OCI.

Requirements	Notes
Compute shapes The minimum requirements are listed below: • 8vCPUs minimum • 64GB RAM minimum • Network Bandwidth: 10 Gbps min (25 Gbps recommended) • Processors: Intel or AMD (No ARM) • Storage • SSD or NVMe (No HDD) Recommended compute shapes: • VM.Standard2 • VM.Standard3.Flex • VM.Standard.E[3,4,5].Flex	• The Delphix Engine most closely resembles a storage appliance and performs best when provisioned using a storage-optimized shape. • Larger shapes provide more CPU, which can prevent resource shortfalls under high I/O throughput conditions. • Larger shapes also provide more memory, which the Delphix Engine uses to cache database blocks. More memory will provide better read performance. • For more information on OCI shapes, refer to OCI shapes[185].

185 https://docs.oracle.com/en-us/iaas/Content/Compute/References/computeshapes.htm#Compute_Shapes

5.13.4 Network configuration

Requirements	Notes
Virtual cloud network (VCN)	• The Delphix Engine and all of the source and target environments must be deployed in a VCN to ensure that private IP addresses are static and do not change when restarting instances. • By default, OCI subnets are considered public. When defining a subnet, it is recommended to set it as private. Unless required by the environment, the VCN should not include a Public Subnet. • When adding environments to the Delphix Engine, use the host's VCN (static private) IP addresses.
Static private IP	• The Delphix instance should be launched with a static private IP address. For security reasons, it is encouraged to avoid configuring the engine with a Public IP address; however, it could be passable to use a dynamic Public IP address in addition to a static Private IP address if the environment requires such access.
Security rules configuration	• OCI allows two firewall features: **Network Security Groups** (NSGs) and **Security Lists**. Oracle recommends the use of NSGs over Security Lists because NSGs let the VCN subnet architecture be separate from the application security requirements. More information can be found in this Oracle documentation[186] • A VCN will use a Security List to define default rules. By default, the security list will only open port 22 for SSH access. The Security List must be modified, or new NSGs created, to allow access to all of the networking ports used by the Delphix Engine and the various source and target engines. • This dual implementation of firewall or security rules may be different from other clouds, please see OCI documentation for best practices. • See Network performance configuration options (see page 719) for information about network performance tuning. • See General network and connectivity requirements (see page 717) for information about specific port configurations.

186 https://docs.cloud.oracle.com/en-us/iaas/Content/Network/Concepts/securityrules.htm#Security_Rules

5.13.5 General storage configuration

Requirements	Notes
General storage configuration	• Allocate initial storage equal to the size of the physical source database storage. For high redo rates and/or high DB change rates, allocate an additional 10-20 %. • Add storage when storage capacity approaches 30% free. • Delphix recommends using a minimum of four equally sized storage disks to run the Delphix Engine. This allows the Delphix File System (DxFS) to assure its file systems are always consistent on disk without additional serialization. This also enables the Delphix Engine to achieve higher I/O rates by queueing more I/O operations to its storage. • Must use Block Volume for data storage. • Block Volumes must be attached using Paravirtualized mode. • When adding additional disks to the block storage pool, storage device initialization will not occur since it is not required on this cloud.
OCI storage configuration	• Currently supported Shapes only support Block Volumes; File Storage is not supported. • Paravirtualized block devices are required; currently, iSCSI devices are not supported. • Elastic Performance Configuration Options (aka Volume Performance Policy): use Higher Performance. • For Delphix Continuous Cloud Engines backed by object storage, read the section ***Delphix Elastic Data Engines (Engines backed by object storage)*** in Initial Setup (see page 571). • Block storage for caching requires a minimum of 200G Higher Performance volumes for caching.

5.13.6 Additional OCI configuration notes

- When running low on storage space, Delphix recommends adding additional equivalently sized block storage volumes, or devices, instead of resizing existing volumes.
- If existing storage volumes must be expanded, this must be done using the “online” resizing strategy specified in OCI documentation; “offline” storage resizing is not supported and may lead to unexpected downtime. If an existing storage volume is expanded, use the Setup or sysadmin interface to expand each storage “device” or volume. The additional storage, as a result of a resize,

will not be available for use until the storage devices are explicitly instructed to make use of the additional space.
- If expanding storage volumes, it is recommended that all volumes are expanded to the same size. When storage volumes or devices are the same size, the Delphix product is able to balance I/O distribution among the disks for optimal performance.
- Hot removal of storage volumes is not supported.

5.13.7 Prerequisites to deploying OCI

5.13.7.1 Overview

This article outlines the prerequisites for deployment of the Delphix Engine on OCI. The setup user should have experience launching and configuring instances in the Oracle Cloud Infrastructure environment. Review and complete the tasks in the next section before deployment.

5.13.7.2 Prerequisites

1. Contact a Delphix representative to request this capability. Delphix will assist in assuring that all Oracle Cloud requirements are met to successfully run a Delphix Engine with the most appropriate configuration for the use case.
2. Review the Checklist of information required for installation and configuration (see page 560)
3. Review Oracle's Creating an instance documentation[187]
4. Make sure that the Oracle account being used to deploy the Delphix Engine has an appropriate level of permissions to upload files to a Bucket in Object Storage, and use that object to produce a Custom Image.
5. Determine which virtual cloud network (VCN) is being used when launching the virtualization instance. To maximize performance, deploy the Delphix Engine instance in the same VCN/subnet in which the virtual databases (VDBs) will be created.
   Note:

   Provisioning a VDB requires a compute instance running the same database engine as the source. Please note, however, that the target instance only needs storage to accommodate the OS, database platform binaries, etc., because Delphix delivers all of the data files.
6. Make sure that the necessary ports are open.

   Using the Delphix Engine for OCI will require connections to source and target database servers. Such connections require various ports to be open, enabling communications. For a detailed list of the network and port requirements, click the link that corresponds with the relevant database platform.
   - Network and Connectivity Requirements for Oracle Environments (see page 1399)
   - Network and Connectivity Requirements for SQL Server Environments (see page 2089)

187 https://docs.oracle.com/en-us/iaas/Content/Compute/Tasks/launchinginstance.htm

- Network and Connectivity Requirements for SAP ASE Environments (see page 1892)
- Network and Connectivity Requirements for Db2 Environments (see page 1020)

7. Update Security Group (or Security List) settings to accommodate the necessary connections.
   a. Select the same Security Group that your current (or future) non-production compute nodes utilize.
   b. Modify the Security Group to allow access to all of the networking ports used by the Delphix Engine and the various source and target platforms. See Oracle Network Security Groups[188]
8. Allocate storage.
   a. To properly size the initial storage capacity and determine the number and size of Provisioned IOPs Volumes required, first create a list of the data sources intended for making dSources. A data source is typically a production database linked to the Virtualization Engine, enabling to create virtual, full, read-write copies of the source within minutes. The list should include the database name, platform (for example, Oracle or SQL Server), current size (in GB), the estimated number of virtual copies, and retention period (in days) of snapshots (backup copies).
   b. Delphix only supports using Block Volume devices for data storage. These block devices must be attached to the Delphix Engine using the Paravirtualized attach mode (iSCSI is not currently supported). Block Volumes must be provisioned in the same Availability Domain as the Delphix Engine, otherwise, they will not be discoverable. Block Volumes should be configured to use the "Higher performance[189]" performance option. All Block Volumes attached to a single Delphix Engine should be of the same size. A minimum of four (4) Block Volumes should be attached to a Delphix Engine.
9. During the Manual Deployment option, use the guidelines outlined in Virtual machine requirements for OCI (see page 651)

5.13.8 Procedure for deploying in OCI

5.13.8.1 Overview

This article outlines the procedure for deploying the Delphix Engine in Oracle Cloud Infrastructure.

5.13.8.2 Download and verify the Delphix engine image

1. Contact an account manager to request access to the OCI variant of the Delphix product.
2. Follow the link given by the Delphix solutions architect. Download the **Delphix_x.x.x.x_….Standard_OCI.qcow**2 file and the **SHA256SUMS** file.

188 https://docs.cloud.oracle.com/en-us/iaas/Content/Network/Concepts/networksecuritygroups.htm?Highlight=security

189 https://docs.cloud.oracle.com/en-us/iaas/Content/Block/Concepts/blockvolumeperformance.htm

3. Once both files have finished downloading and are in the same directory, run the following command to verify the download: `$ grep -i OCI.qcow2 ./SHA256SUMS | sed -E 's,Appliance_Images/(Controlled_Availability/)?,,g' | sha256sum --check`

5.13.8.3 Upload the Delphix engine image as an object

1. Authenticate with OCI and navigate to the Infrastructure console[190]
2. Use the navigation menu to reach the Object storage buckets, core infrastructure, page[191] (Menu > Object Storage > Object Storage).
3. Set the **List Scope Compartment**. This will depend on the organization's strategy for managing OCI resources.
4. Create a storage bucket[192] or select an existing bucket.
5. Click the blue **Upload** button.
6. In the **Upload Objects** modal window, specify an optional prefix and choose the OCI specific QCOW2 file that was previously downloaded.
7. Click the blue **Upload** button.

5.13.8.4 Creating a custom compute image from an object

1. Authenticate with OCI and navigate to the Infrastructure console[193]
2. Use the navigation menu to reach the Compute custom images, core infrastructure, page[194] (Menu > Compute > Custom Images).
3. Set the **List Scope Compartment**.
4. Click the blue **Image Import** button.
5. In the **Import Image** modal window, select a suitable compartment in the **Create In Compartment** field that conforms to the organization's strategy on managing OCI resources.
6. In the **Name** field enter a unique name to identify the Custom Compute Image (using the same name as the image object from the previous step is recommended). Upload the Delphix Engine Image as an Object.
7. For **Operating System** select **Linux**.
8. Next, identify an object by specifying its Compartment, Bucket, and Object Name. Or, specify an Object Storage URL.

190 https://console.us-phoenix-1.oraclecloud.com/
191 https://console.us-phoenix-1.oraclecloud.com/object-storage/buckets
192 https://docs.cloud.oracle.com/en-us/iaas/Content/GSG/Tasks/addingbuckets.htm#Putting_Data_into_Object_Storage
193 https://console.us-phoenix-1.oraclecloud.com/
194 https://console.us-phoenix-1.oraclecloud.com/compute/images

Note:
The Object Details will identify this value as **URL Path (URI)**.

9. For **Image Type** select **QCOW2**.
10. For **Launch Mode** select **Paravirtualized Mode**.
11. For organizations that have a tagging policy for cloud-based resources, expand the **Tagging Options** section, and define tags.
12. Click the blue **Import Image** button.

5.13.8.5 Launching the Delphix engine

1. Authenticate with OCI and navigate to the Infrastructure Console[195]
2. Use the navigation menu to reach the Compute instances, core infrastructure, page[196] (Menu > Compute > Instances).
3. Set the **List Scope Compartment**.
4. Click the blue **Create Instance** button.
5. In the **Create Compute Instance** window pane, specify a unique name for the VM.
6. For the **Create In Compartment** field, select a suitable compartment that conforms to the organization's strategy on managing OCI resources.
7. In the **Image or operating system** section, click the **Change Image** button. Switch to the **Custom Images** tab. Find the Delphix image that corresponds to the instance being deployed. Click the blue **Select Image**button.
   Note:
   If the Delphix Custom Image is not visible, look for the **Change Compartment** option near the top of the current window pane.
8. Each Availability Domain has its own quota, it is ok to use AD-1, AD-2, or AD-3 - but, be sure to make note of which Availability Domain is being used.
   Note:
   Compute Instances and attached Storage will need to be in the same Availability Domain.
9. In the **Shape** section, click the **Change Shape** button. For **Instance type**, specify **Virtual Machine**. Select an OCI Shape that is supported by Delphix.
10. Continue on to the **Configure networking** section.
    a. If the network is not specified correctly, it is likely that firewall issues will occur. In this case, please consult the organization's IT or DevOps team. If the organization is using Network Security Groups (NSGs), mark the **Use Network Security Groups to Control Traffic** checkbox after consulting the appropriate teams. Last, select the **Do Not Assign a Public IP Address** radio button.
11. The **Boot Volume** section can be skipped.

---

195 https://console.us-phoenix-1.oraclecloud.com/
196 https://console.us-phoenix-1.oraclecloud.com/compute/instances

12. In the **Add SSH Keys**, select the **No SSH Keys** radio option. The Delphix product is a closed appliance and manages users independently.
13. In general, all of the Advanced Options can be skipped. For organizations that have a tagging policy for cloud-based resources, expand into the Advanced Management section, and look for the **Tagging** sub-section to define tags.
14. Click the blue **Create** button - wait about 2-5 minutes for the Delphix Engine instance to boot.

5.13.8.6 Create block storage volumes

1. Authenticate with OCI and navigate to the Infrastructure console[197]
2. Use the navigation menu to reach the Block volumes, core infrastructure, page[198] (Menu > Block Storage > Block Volumes).
3. Set the **List Scope Compartment**.
4. Click the blue **Create Block Volume** button.
5. In the **Create Block Volume** modal window, specify a unique name for this Block Volume. It can be helpful if this name is descriptive or identifies the VM it is intended to be attached to and ends in a sequence number.
6. For the **Availability Domain,** this value MUST be the same Availability Domain used for the Delphix Engine instance, otherwise, this volume will not be available for use.
7. In the **Volume Size and Performance** section, select the **Custom** option. Set the size of the volume to be sufficiently large (with room for growth) to support the databases that will be virtualized, or masked, by this Delphix Engine. Set the **Default Volume Performance** to the **Higher Performance** setting.
8. A **Backup Policy** is not required and can be left blank or **No Backup Policy Selected**. However, depending on the organization's needs, consider selecting a Backup Policy.
9. For **Encryption,** the default option of **Encrypt Using Oracle-Managed Keys** is permissible. Optionally, for independent encryption keys, use the **Encrypt Using Customer-Managed Keys** option.
10. For organizations that have a tagging policy for cloud-based resources, expand the **Tagging Options** section and define tags.
11. Uncheck the checkbox that says **View Detail Page After This Block Volume is Created**. This will prevent navigating away from the Block Volumes page because multiple Block Volumes will need to be created at the same time.
12. Click the blue **Create Block Volume** button.
13. A Delphix Engine requires a minimum of four equally sized Block Volumes for data storage. Repeat Steps 4-12 as many times as necessary.

197 https://console.us-phoenix-1.oraclecloud.com/
198 https://console.us-phoenix-1.oraclecloud.com/block-storage/volumes

5.13.8.7 Attach block storage volumes

(For an Elastic Data engine, the block storage will be used for caching. See additional details in the Delphix Elastic Data Engines section in Initial setup (see page 571))

1. Authenticate with OCI and navigate to the Infrastructure console[199]
2. Use the navigation menu to reach the Block volumes, core infrastructure, page[200] (Menu > Block Storage > Block Volumes).
3. Set the **List Scope Compartment**.
4. From the list of pre-existing Block Volumes, identify the resources being attached to the Delphix Engine and wait until the volume state becomes Available.
5. Select one of the **Block Volumes** to enter the **Block Volume Details** page.
6. On the left-hand side, locate the **Resources** menu and select **Attached Instances**.
7. If the Block Volume has not been previously attached to another VM, selecting the blue **Attach to Instance** button will be available.
8. In the **Attach to Instance** modal window, specify the **Attachment Type** as **Paravirtualized**. Currently, iSCSI is not supported.
9. For **Access Type** use the **READ/WRITE** option.
10. Next, identify a Delphix Engine by selecting an instance or by specifying an instance OCID. If the Delphix Engine instance is not shown in the **Select an Instance** drop-down menu, the **Change Compartment** option could be needed. Block Volumes can only be attached to VM instances that were created in the same Availability Domain - if these values do not match, either re-provision Block Volumes or the Delphix Engine in the correct Availability Domain.
11. Click the blue **Attach** button.
12. Repeat Steps 4-11 until all associated Block Volume resources have been attached to the Delphix Engine instance.

5.13.8.8 Configuring the Delphix engine

1. Connect to the running Delphix Engine instance with a web browser. Use the IP address or FQDN noted in the Instance Description. Upon successful connection, the browser will display a login prompt to enter the Delphix Setup Page.
2. Refer to the standard product deployment instructions (see page 571) to complete the Delphix deployment.

199 https://console.us-phoenix-1.oraclecloud.com/
200 https://console.us-phoenix-1.oraclecloud.com/block-storage/volumes

5.13.8.9 Next steps

Congratulations! The Delphix Virtualization Engine should be successfully deployed in OCI.

Use Delphix documentation to learn how to:

- configure a database source
- configure target environments
- create virtual databases (VDBs)

5.14 Deployment for IBM cloud

5.14.1 Overview

This article covers the virtual machine requirements, including memory and data storage, for the deployment of the Delphix Engine on IBM Cloud. Once the requirements listed on this page are reviewed, refer to the next articles on GCP deployment:

- Prerequisites for deploying in IBM Cloud (see page 664)
- Procedure for deploying in IBM Cloud (see page 665)

> Delphix disk storage capabilities remain seamlessly operable. There are no additional limitations to your storage, unless otherwise imposed by the [respective] Hypervisor or Cloud environment.

5.14.2 Supported profiles

The following is a list of profiles that are supported to deploy Delphix on IBM Cloud.

Requirements	Notes
Memory instance families. The minimum requirements are listed below: • 8vCPUs minimum • 64GB RAM minimum • 10Gbps min (25Gbps recommended) • Processors: Intel or AMD (No ARM) • Storage • SSD or NVMe (No HDD) Recommended instance families: • mx2	• The Delphix Engine most closely resembles a storage appliance and performs best when provisioned using a storage-optimized profile. • Larger profiles provide more CPU, which can prevent resource shortfalls under high I/O throughput conditions. • Larger profiles also provide more memory, which the Delphix Engine uses to cache database blocks. More memory will provide better read performance. Information on IBM Cloud instances[201]

5.14.3 Network configuration

Requirements	Notes
Virtual server instances	• Deploy the Delphix Engine and all of the source and target environments in the same VPC network. • When adding environments to the Delphix Engine, use the host's VPC IP addresses.
Security configuration	• The default security group will only open port 22 for SSH access. The security group must be modified to allow access to all of the networking ports used by the Delphix Engine and the various source and target engines. • See Network performance configuration options (see page 719) for information about network performance tuning. • See General network and connectivity requirements (see page 719) for information about specific port configurations. • Reference: IBM cloud security and compliance[202] documentation

Port Requirements

Using the Delphix Engine for OCI will require connections to source and target database servers. Such connections require various ports to be open, enabling communications. For a detailed list of the network and port requirements, click the link that corresponds with the relevant database platform.

201 https://cloud.ibm.com/docs/vpc?topic=vpc-profiles&interface=ui
202 https://cloud.ibm.com/docs/security-compliance?topic=security-compliance-getting-started

- Network and connectivity requirements for Oracle environments (see page 1399)
- Network and connectivity requirements for SQL Server environments (see page 1399)
- Network and connectivity requirements for SAP ASE environments (see page 1892)

5.14.4 General storage configuration

Requirements	Notes
General storage configuration	• Allocate initial storage equal to the size of the physical source database storage. For high redo rates and/or high DB change rates, allocate an additional 10-20 %. • Add storage when storage capacity approaches 30% free. • Delphix recommends using a minimum of four equally sized storage disks to run the Delphix Engine. This allows the Delphix File System (DxFS) to assure its file systems are always consistent on disk without additional serialization. This also enables the Delphix Engine to achieve higher I/O rates by queueing more I/O operations to its storage. • When adding additional disks to the block storage pool, storage device initialization will not occur since it is not required on this cloud. • Reference: IBM block storage documentation[203]

5.14.5 Additional IBM configuration notes

- Expandable volume is a beta feature that is available for evaluation and testing purposes. This feature is available in the US South, US East, London, and France regions. Contact an IBM Sales representative if interested in getting access at https://cloud.ibm.com/docs/vpc?topic=vpc-expanding-block-storage-volumes .
- After performing an “online” resize/expansion of a storage volume using IBM Cloud tools, then use the Delphix sysadmin interface to “Expand” the storage device; otherwise, the newly allocated storage space, from the resize/expansion, will not be used.
- Resize/expansion of a storage volume using IBM Cloud is not supported while the Delphix engine is in a stopped state.
- Removing a storage volume should be done while the machine is running. First use the Delphix sysadmin CLI interface to “Unconfigure” the storage device, then remove it from IBM Cloud.

203 https://cloud.ibm.com/docs/vpc?topic=vpc-block-storage-profiles#tiers-beta

5.14.6 Prerequisites for deploying in IBM Cloud

5.14.6.1 Overview

This article outlines the prerequisites for deployment of the Delphix Engine on IBM Cloud. The setup user should have experience launching and configuring instances in the IBM Cloud environment. Review and complete the tasks in the next section before deployment.

5.14.6.2 Prerequisites

1. A Delphix software license is required. New users should contact Delphix[204] to get started.
2. Review the Checklist of information required for installation and configuration (see page 560)
3. Review IBM's cloud documentation[205] for IBM Cloud-specific information.
4. Determine which virtual private cloud (VPC) is being used when launching the virtualization instance. To maximize performance, deploy the Delphix Engine instance in the same VCN/subnet in which the virtual databases (VDBs) will be created. **Note:** Provisioning a VDB requires a compute instance running the same database engine as the source. Please note, however, that the target instance only needs storage to accommodate the OS, database platform binaries, etc., because Delphix delivers all of the data files.
5. Make sure that the necessary ports are open. **Note:** Using the Delphix Engine for GCP will require connections to source and target database servers. Such connections require various ports to be open, enabling communications. For a detailed list of the network and port requirements, click the link that corresponds with the relevant database platform.
   - Network and connectivity requirements for Oracle environments (see page 1399)
   - Network access requirements for SQL server (see page 2089)
   - Network and connectivity requirements for SAP ASE environments (see page 1892)
   - Network and connectivity requirements for Db2 environments (see page 1020)
6. Update Security Group settings to accommodate the necessary connections.
7. Select the same Network Security Group that your current (or future) non-production Azure virtual machines utilize.
8. Modify the Security Group to allow access to all of the networking ports used by the Delphix Engine and the various source and target platforms. See the links above for information about specific port configurations.
9. Allocate storage.

---

204 https://www.delphix.com/company/contact
205 https://cloud.ibm.com/docs

> It is helpful to first create a list of the data sources from which you intend to make dSources. A data source is typically a production database that you link to the Delphix Engine, enabling you to create virtual, full, read-write copies of the source within minutes. The list should include the database's name, platform (for example, Oracle or SQL Server), current size (in GB), the estimated number of virtual copies, and retention period (in days) of snapshots (backup copies).

a. All data storage disks must be comprised of Premium Storage devices.

b. Delphix recommends using a minimum total of four disks to run the Delphix Engine. One disk is used for the boot device. The other four equally sized disks will be used for data storage. This also enables the Delphix Engine to achieve higher I/O rates by queueing more I/O operations to its storage.

c. Premium storage devices have different levels of guaranteed IOPs that vary from 120 IOPs to 7500 IOPs. Please refer to IBM Cloud documentation to ensure that the selected devices will meet the requirements for your deployment.

d. Use the guidelines outlined in Procedure for Deploying in IBM Cloud.

5.14.7 Procedure for deploying in the IBM Cloud

5.14.7.1 Overview

This article outlines the procedure for deploying the Delphix Engine in IBM Cloud. There are two methods for deploying a Delphix Engine in the IBM Cloud using the Software Catalog[206] or Manually Uploading the Delphix Image

5.14.7.2 Deploying from the IBM software catalog

1. Navigate to the IBM Software Catalog[207] and search for Delphix.
2. Select the Delphix Data Virtualization Tile or Masking Tile for the Masking product.
3. Scroll down to the Deployment Values section and input specifics for the environment.

Required parameters	Description
hostname	The name of the VSI used to deploy Delphix.
profile	Compute profile to be used for deploying Delphix (see recommended profiles).

206 https://cloud.ibm.com/catalog#software
207 https://cloud.ibm.com/catalog#software

Required parameters	Description
ssh_key	Public SSH key is to be used when provisioning the VSI.
subnet_id	The id of the subnet where the VSI will be provisioned.
volumecount	Number of block storage volumes
volumeprofile	Block storage profile to use (recommended is >= 10 IOPS/GB
volumesize	Block storage volume size.
vpcname	Name of the VPC where the VSI is provisioned.
zone	VPC zone to provision the environment.

5.14.7.3 Downloading the Delphix image

Contact an account manager to request access to the IBM variant of the Delphix product.

1. Follow the link given by the Delphix solutions architect. Download the **Delphix_Verson…._Standard_IBM.qcow2** file and the **SHA256SUMS** file.
2. Once both files have finished downloading and are in the same directory, run the following command to verify the download:

```
$ grep -i IBM.qcow2 ./SHA256SUMS | sed -E 's,Appliance_Images/(Controlled_Availability/)?,,g' | sha256sum --check
```

5.14.7.4 Uploading the Delphix engine image as an object

1. Authenticate with the IBM Cloud and navigate to the Dashboard[208]

208 https://cloud.ibm.com/

2. Use the navigation menu to reach the Resource List page[209]. The Resource List page can be navigated from the Dashboard by clicking on **Storage** within the Resource Summary pane.
3. Expand Storage from the menu and select the appropriate resource group. Depending on the organization's strategy for managing IBM resources, creating a resource group[210] may be required.
4. Create a storage bucket[211] or select an existing bucket.
5. Click the blue **Upload** button and select **Files**.
6. When the pop-up menu appears, select the **Transfer Type**. Aspera High-Speed Transfer is required for large files, the plugin will need to be installed. It will automatically navigate through the steps to install the plugin.
7. In the **Upload Files** (objects) window, click on the **Select Files** (objects) button and choose the **IBM-specific QCOW2** file that was previously downloaded.
8. Click the **Upload** button.

5.14.7.5 Creating a custom image

1. Authenticate with IBM Cloud and navigate to the Dashboard[212]
2. Use the navigation menu to reach the Custom images page[213] for VPC within the VPC infrastructure (IBM Cloud pull-down menu > VPC Infrastructure > Custom images).
3. Click the blue **Create** button.
4. In the **Import Custom Image** page, specify a unique name for the image.
5. From the **Resource Group** drop-down, select the organization's resource group.
6. Optional: In the **Tags** section, provide appropriate tags[214] to organize resources.
7. Select the appropriate **Region**.
8. Select the **Cloud Object Storage** bucket containing the uploaded image by navigating to **Instances** > **Location** > **Bucket** from the drop-down menus. The downloaded **QCOW2** image should appear in the pane below the three drop-down menus.
9. Within the **Operating System** section, click on the **Ubuntu Linux** tile and select **ubuntu-18-04-amd64** from the drop-down menu.
10. Once all the parameters are entered, in the right pane click on the blue button to import custom image.

---

209 https://cloud.ibm.com/resources
210 https://cloud.ibm.com/docs/account?topic=account-rgs#create_rgs
211 https://cloud.ibm.com/docs/cloud-object-storage/getting-started.html#gs-create-buckets
212 https://cloud.ibm.com/
213 https://cloud.ibm.com/vpc-ext/compute/images
214 https://cloud.ibm.com/docs/account?topic=account-tag

5.14.7.6 Launching the Delphix engine

1. Authenticate with IBM Cloud and navigate to the Dashboard[215]
2. Use the navigation menu to reach the Virtual server Instances page[216] within the VPC Infrastructure (IBM Cloud pull-down menu > VPC Infrastructure > Virtual Server Instances). Note: To maximize performance, deploy the Delphix Engine instance in the same VPC/subnet in which you will create your virtual databases (VDBs).
3. Click the blue **Create** button.
4. In the **New Virtual Server** for VPC page, specify a unique name for the VM.
5. From the **Virtual Private Cloud** drop-down, select the organization's VPC.
6. From the **Resource Group** drop-down, select the organization's resource group.
7. Optional: In the **Tags** section, provide appropriate tags[217] to organize resources.
8. Select the **Location** of the IBM Cloud resources.
9. In the **Operating System** section, click on the **Select Custom Image** link within the Custom Image block.
10. In the pop menu, select the IBM-specific image previously uploaded.
11. Within the **Profile** section, click on **View** all profiles. Select one of the supported instance types and click **Save**.
12. The **User Data** section can be skipped.
13. The Boot Volume section can be skipped since it would already have the default values.
14. Creating block storage volumes can be done at a later time and will be discussed in the next section.
15. Continue on to the **Network Interfaces** section. If a subnet is already configured in the zone and VPC, then this section will already have a default network interface. Otherwise, create a subnet with the appropriate security groups. If the network is not specified correctly, it is likely that firewall issues will occur. In this case, please consult the organization's IT or DevOps team. If the organization is using Network Security Groups (NSGs), mark the **Use Network Security Groups to Control Traffic** checkbox after consulting the appropriate teams. Last, select the **Do Not Assign a Public IP Address** radio button.
16. Click the **Create virtual server instance** button on the right panel. This will take a couple of minutes.

5.14.7.7 Creating block storage volumes

1. Authenticate with IBM Cloud and navigate to the Dashboard[218]

---

215 https://cloud.ibm.com/
216 https://cloud.ibm.com/vpc-ext/compute/vs
217 https://cloud.ibm.com/docs/account?topic=account-tag
218 https://cloud.ibm.com/

2. Use the navigation menu to reach the Block Storage Volumes[219] within VPC Infrastructure (IBM Cloud pull-down menu > VPC Infrastructure > Block Storage Volumes).
3. Click the blue **Create** button.
4. In the **Block Storage Volume for VPC** modal window, specify a unique name for this Block Volume. It can be helpful if this name is descriptive or identifies the VM it is intended to be attached to and ends in a sequence number.
5. From the **Resource Group** drop-down, select the organization's resource group.
6. Optional: In the **Tags** section, provide appropriate tags[220] to organize resources.
7. Select the **Location** of the IBM Cloud resources.
8. Enter the required **IOPS**. The recommended supported IOPS is 10/GB.
9. Enter the storage size in GB. Set the size of the volume to be sufficiently large, with room for growth, to support the databases that will be virtualized, or masked, by this Delphix Engine.
10. Keep the **Encryption** settings at default, e.g. Provider Managed.
11. Click the blue **Create Volume** button.
12. Delphix recommends using a minimum total of four disks to run the Delphix Engine. One disk is used for the boot device. The other four equally sized disks will be used for data storage. This also enables the Delphix Engine to achieve higher I/O rates by queueing more I/O operations to its storage. Repeat Steps 3-11 as many times as necessary.

5.14.7.8 Attaching block storage volumes

1. Authenticate with IBM Cloud and navigate to the Dashboard[221]
2. Use the navigation menu to reach the Block storage volumes[222] within VPC Infrastructure (IBM Cloud pull-down menu > VPC Infrastructure > Block Storage Volumes).
3. From the list of pre-existing **Block Volumes**, identify the volumes attaching to a Delphix Engine and wait until the volume state becomes Available.
4. Note that the volumes being attached have **Attachment Type** set as a hyphen.
5. Select the right side of the volume row menu, then select **Attach to Instance**.
6. In the **Attach Virtual Server Instance** modal window, select the virtual server instance (Delphix Engine) from the drop-down menu.
7. Click on the blue **Attach Volume** button.
8. Repeat Steps 3-7 until all associated **Block Volume** resources have been attached to the Delphix Engine instance.

---

219 https://cloud.ibm.com/vpc-ext/storage/storageVolumes
220 https://cloud.ibm.com/docs/account?topic=account-tag
221 https://cloud.ibm.com/
222 https://cloud.ibm.com/vpc-ext/storage/storageVolumes

5.14.7.9 Configuring the Delphix engine

1. Connect to the running Delphix Engine instance with a web browser. Use the IP address or DNS name noted in the Instance Description. Upon successful connection, the browser will display a login prompt to enter the **Delphix Setup Page**.
2. Refer to the standard product deployment instructions (see page 2363) to complete the Delphix deployment.

5.14.7.10 Next steps

Congratulations! The Delphix Virtualization Engine should be successfully deployed in IBM Cloud.

Use Delphix documentation to learn how to:

- configure the database source (see page 1046)
- configure target environments (see page 2338)
- create virtual databases (VDBs) (see page 1047)

5.15 Hotfix information

5.15.1 Overview

If there are any hotfixes installed for the Delphix Engine, the list can be accessed via the system administrator CLI with the following commands.

```
engine> cd system
engine system> ls
....
hotfixes: [HF-111]
....
```

6 Configuration

6.1 Configuration

Once you have deployed Delphix in the infrastructure of your choice, you will need to manage the settings of each Delphix Engine. The configuration section covers everything you need to do and know about Delphix settings and system administration. Each Delphix engine has its own settings, user profiles, policies, and many other configuration parameters. The majority of these are managed per engine, but increasingly will be managed via the Central Managed service. There are also many administrative functions you may need to manage, such as storage and capacity utilization, system monitoring integrations, and Support bundle upload. You may also want to change your engine's configuration, such as adding new storage disks, authentication mechanisms, or network properties.

6.1.1 Registration management

Each Delphix Engine can be registered via our internal system, typically for association with the Delphix Support team. Learn how to ensure up-to-date registration of your Delphix Engines.

6.1.2 User and authentication management

There are various user types and configurations to consider when using Delphix. Learn how to set up users, and manage authentication mechanisms such as Single Sign-on and Kerberos.

6.1.3 Network and DNS management

You may want to manage and configure certain network services, such as DNS, for Delphix. Here we specify general network and connectivity requirements and detail how to test network performance.

6.1.4 Capacity and resource management

Delphix will be responsible for managing many different data sources and data types. Learn storage and quota best practices, as well as the different options to optimally manage capacity.

6.1.5 Monitoring and log management

Using both Delphix and its associated datasets will generate many types of logs. Here, we explain the types of logs that Delphix creates as well as the monitoring tool integrations, such as SNMP and Splunk.

6.1.6 Performance analytics management

Delphix offers various performance analytics tools to help users monitor throughput, latency, and other key metrics. Learn how to leverage these tools and architect your Delphix deployment for optimal performance.

6.1.7 Starting, stopping, and restarting your engine

Occasionally, you may need to reboot or stop your Delphix engine, here, you will find steps to securely and safely restart your engines.

6.1.8 Usage data management

The Delphix User-click Analytics feature is a lightweight method to capture how users interact with Delphix product user interfaces. User-click Analytics may also be disabled via the UI.

6.2 Registration management

Registering your Delphix Engines is a prerequisite for serviceability from the Delphix Support team. Registration allows Delphix Support to access the engine and properly diagnose and identify any issues during support cases. It is important to manage your registration regularly in order to ensure the security of Support access, as described in the section below.

Registering a Delphix Engine with support is different from registering it with Data Control Tower, *formerly* Central Management. For more information refer to Data Control Tower[223]

> The registration code contains an encrypted key that only Delphix can decrypt, which is unique for each engine. Delphix uses this key to generate one-time authentication codes that authorized Support personnel can use to log into the engine during support sessions.

6.2.1 Retrieving the Delphix engine registration code

To enable Delphix Support, you may perform registration either during the initial setup of the Delphix Engine. If you want to register later, you may retrieve the registration code from the system setup portal.

1. You can retrieve the Delphix Engine Registration Code during the Initial Setup or later through the Delphix Setup application after logging in as a system administrator.
2. In the Registration panel, click View.
3. The Registration Code is displayed in the bottom half of the Registration window.

223 https://delphixdocs.atlassian.net/wiki/spaces/Offerings/pages/19989470/Data+Control+Tower

4. If your local workstation is connected to the external Internet, you can auto-register the Delphix Engine:
    a. Enter your Delphix Support Username and Support Password.
    b. Click Register.
5. If external connectivity is not immediately available, you must register manually.
    a. Copy the Delphix Engine registration code
    b. Log in with your support credentials at https://register.delphix.com[224]
    c. Paste the Registration Code and click Register.

Following registration, you will receive an email confirming the registration of your Delphix Engine

6.2.2 Regenerating the Delphix engine registration code

Delphix recommends that you regenerate the registration code every six months to rotate the secret key in order to maximize the Support Security of the Delphix Engine.

Procedure

1. Log into the CLI (command-line interface) of the Delphix Engine with the sysadmin credentials.
2. Type /registration/regenerate and hit enter.
3. Type commit and hit enter. After a few seconds, the new code will be displayed.
4. Re-register the engine with this new code.

Failing to re-register the Delphix Engine after regenerating the registration code may prevent Support personnel from accessing the engine. In such a case, a support session cannot begin until the engine has been re-registered with the new registration code.

6.3 User and authentication management

There are two main topics in this section:

- Managing users and groups who will access and manage Delphix.
- Configuring various authentication mechanisms to access both Delphix and connected environments and datasets.

This section covers the following topics:

- Users and groups (see page 674)
- Authentication mechanisms (see page 689)

224 https://register.delphix.com/

6.3.1 Users and groups

6.3.1.1 User types and user management

There are three user types in the Delphix user model: the system administrator, the Delphix user, and the Self-service user.

> Usernames must start with a letter and contain only alphanumeric characters, hyphens, underscores, and/or periods.

6.3.1.1.1 System administrators

System administrator users are responsible for managing the Delphix Engine itself, but not the objects (Environments, dSources, VDBs) within the server. For example, a system administrator is responsible for setting the time on the Delphix Engine and its network address, restarting it, creating new system administrator users (but not Delphix users), and other similar tasks.

A user called sysadmin is the default system, administrator user. While this user can be suspended, it may not be deleted. When the Delphix Management application first launches, this user can log in using the username sysadmin and password sysadmin.

To create or modify system administrators, first, log in to Delphix Setup and navigate to the Users section of the homepage. Here, you can:

- **Add new** system administrators with the plus sign
- **Change** system administrator passwords with the pencil icon
- **Delete** system administrators with the trashcan icon
- **Suspend** system administrators with the pause button
- **Reinstate** system administrators with the play button

6.3.1.1.2 Delphix users

Delphix users are responsible for managing the environments and datasets within Delphix, such as dSources, virtual databases (VDBs), users, groups, and related policies and resources.

A Delphix user can be marked as a Engine Administrators. Engine Administratorss have three special privileges:

- They can manage other Delphix users
- They implicitly have Owner privileges for all Delphix objects
- They can create new groups and new environments

The default Delphix user provided with a Delphix Engine is a Engine Administrators and is called admin. Like the sysadmin user, the admin cannot be deleted. When the Delphix Management application launches,

the admin user can log in using the password specified during the initial setup when Delphix was first launched.

Only these two users require password-based authentication. Also, other users may use other mechanisms such as LDAP or Kerberos, as described in Configuring and managing kerberos (see page 699) and Configuring and using LDAP with the Delphix Engine (see page 692).

6.3.1.1.3 Self-service users

Delphix Self-Service has two types of users: the admin user and the data user.

Admin users have full access to all report data and can configure Delphix Self-Service, additionally, they can:

- Use the Delphix Engine to add/delete users
- Change tunable settings
- Add/delete tags
- Create and assign data templates and containers

Data users have access to production data provided in a data container. The data container provides these users with a playground in which to work with data using the Self-Service Toolbar.

For more information on Self-service users, visit our Self-service documentation (see page 2420).

6.3.1.2 User privileges for Delphix objects

The user roles on Delphix objects consist of four types, which the Engine Admin user assigns: Provisioner, Owner, Data Operator, and Reader. These privileges apply both to objects, such as dSources and Virtual Databases (VDBs), and to groups, which are containers that hold those objects.

The Engine Administrators user can assign privileges to groups, dSources, and VDBs. Privileges are inherited, meaning that privileges assigned to a group are effective for the dSources and VDBs contained in that group.

If a user does not have a privilege in relation to an object or group, then he or she has no visibility into that object or group.

Roles and Privileges for Delphix Objects

Role	Object privileges	Group privileges
Owner	• Can provision VDBs from owned dSources and VDBs • Can perform Virtual to Physical (V2P) from owned dSources and VDBs • Can access statistics on the dSource, VDB, or snapshot such as usage, history, and space consumption • Can refresh or rollback VDBs • Can snapshot dSources and VDBs • Can start, stop, or re-start VDBs • Can drop VDBs	• Can provision VDBs from all dSources and VDBs in the group • Can refresh or rollback all VDBs in the group • Can snapshot all dSources and VDBs in the group • Can perform Virtual to Physical (V2P) from owned dSources and VDBs • Can view Templates for policies. • Can not create, edit, or delete a policy template from the policy page. • Can assign Owner privileges for dSources and VDBs • Can access the same statistics as a Provisioner, Data Operator, or Reader • Can start, stop, or re-start VDBs
Provisioner	• Can access statistics on the dSource, VDB, or snapshot such as usage, history, and space consumption • Can provision VDBs from dSources and VDBs	• Can access statistics on all dSources, VDBs, or snapshots in the group such as usage, history, and space consumption • Can provision VDBs from all dSources and VDBs in the group
Data Operator	• Can access statistics on the dSource, VDB, or snapshot such as usage, history, and space consumption • Can refresh or rollback VDBs • Can snapshot dSources and VDBs	• Can access statistics on all dSources, VDBs, or snapshots in the group such as usage, history, and space consumption • Can refresh or rollback all VDBs in the group • Can snapshot all dSources and VDBs in the group
Reader	• Can access statistics on the dSource, VDB, or snapshot such as usage, history, and space consumption	• Can access statistics on all dSources, VDBs, or snapshots in the group such as usage, history, and space consumption

Role	Object privileges	Group privileges

Self-Service Only	• In the Delphix Self Service UI this user can:
	  • Refresh
	  • Restore
	  • Bookmark
	  • Reset
	  • Branch
	  • Stop/Activate
	  • Share	

6.3.1.3 Managing groups

Creating groups helps you manage policies and privileges over objects within that group. When privileges are created for users at the group level, those privileges apply to all objects of that type within the group. When new objects are created or added to the group, the policies and privileges you have created at the group level will be applied to them.

- Assigning group and object ownership (see page 678)
- Adding and deleting groups (see page 678)

6.3.1.4 Authentication mechanisms

Delphix supports a variety of authentication mechanisms to connect to several different interfaces and systems. For example, you can connect via the UI using the default users described above, or you can connect to the CLI using an API token.

There are three categories of authentication related to Delphix: the Delphix UI, the Delphix CLI/API, and external systems such as Kerberos access to connected source and target hosts. Below are detailed pages related to each of these three sections:

- UI authentication:
  - Data Control Tower, formerly Central Management
  - Username and password
  - LDAP: Directory-based authentication to Delphix engines rather than the default local access
  - Single Sign-on: Integration and support for identity providers to authenticate users on a per engine basis using SAML2-SSO.
- CLI authentication:
  - Username and password
  - Auto-authentication via SSH keys: to automatically sign in to the Delphix CLI without requiring user-input credentials
- API authentication

    - Username and password
    - API Tokens (for Delphix Engines registered with Data control tower[225])
    - OAuth2 JSON Web Tokens
- External systems:
    - Username and password
    - SSH keys
    - Kerberos: Authentication for environments and data sources using Kerberos

> **Keberos support**
>
> Kerberos support is for access to connected environments, rather than the Delphix engine itself. This is an advanced topic and will require a solid understanding of Delphix concepts and architecture.

6.3.1.5 Assigning group and object ownership

This topic describes how to assign group and object ownership to users in the Delphix Domain.

1. Log into the **Delphix Management** application.
2. Select **Manage > Users**.
3. For an existing user, select a user then select the edit icon.
4. Click **Next**.
5. In the **Privileges** tab assign **Owner** or **Provisioner** rights for groups or objects within groups. You do not have to assign a specific owner or auditor right for each object.
6. Click Next when finished.
7. For new users, refer to Users and Groups (see page 674). When you click **Submit**, the User Profile manager will reload, and then you can follow steps 4 - 6 to assign privileges.

6.3.1.6 Adding and deleting groups

This topic describes how to add and delete groups within the Delphix Domain.

225 https://delphixdocs.atlassian.net/wiki/spaces/DCT

User terminology for Delphix admin has been changed to engine admin.
The default domain user created on Delphix Engines is now **admin** instead of delphix_admin. When engines created before 5.3.1 are upgraded to 5.3.1 or later they will retain their old username 'delphix_admin'. To avoid complications Delphix recommends creating users with an admin role and then Disabling delphix_admin.

6.3.1.6.1 Adding a group

1. Log into the **Delphix Management** application.
2. From the **Manage** menu, select **Datasets**.
3. Select the **plus** icon and then select **Add Dataset Group**.
4. Enter a **Group Name** and an optional description.
5. Click **Add**.

At least one group must exist
At least one group must always exist on the Delphix Engine in order to link a dSource. If you delete the last group, you will need to create a new group in order to create a dSource.

6.3.1.6.2 Deleting a group

1. Log into the **Delphix Management** application as a user with **Engine Admin** privileges or group **OWNER** privileges for the target group.
2. From the **Datasets** panel, select the target group.
3. Click the **Trash Can** icon.
4. Click **Delete**.

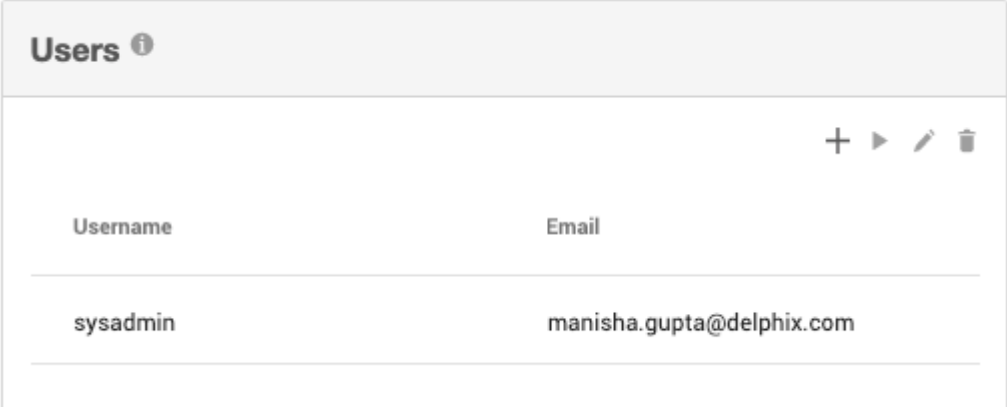
Deleting groups containing objects

A group cannot be deleted if it contains VDBs or dSources. All databases within a group must be deleted prior to deleting the group.

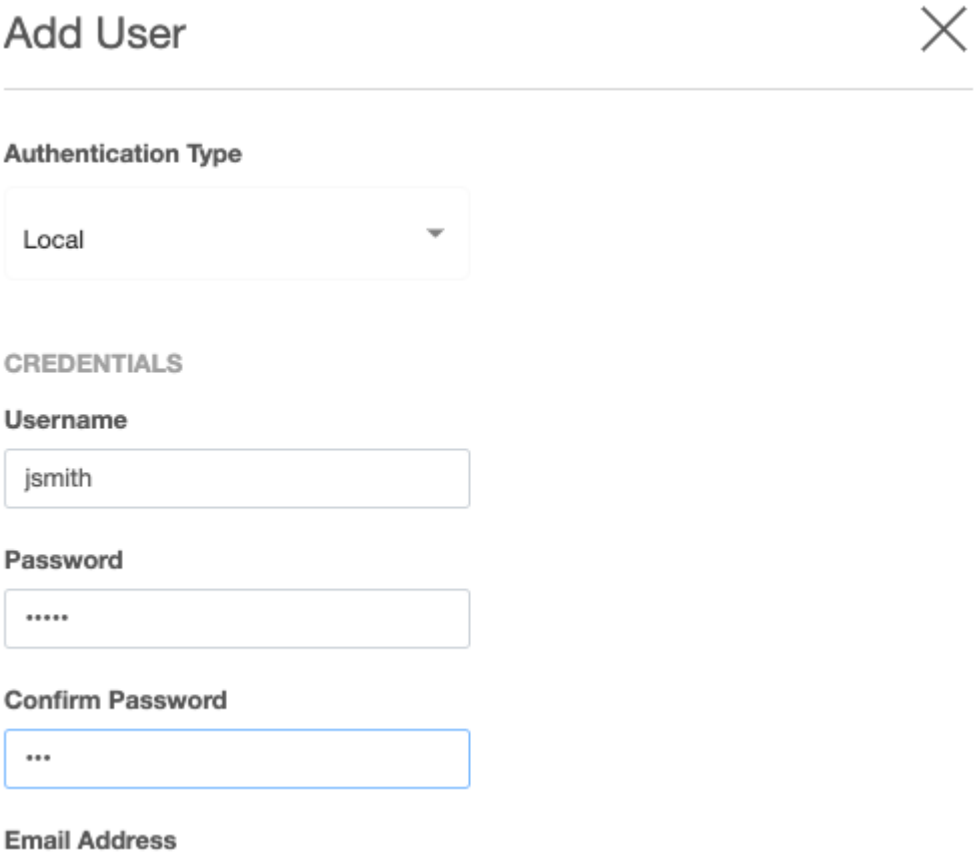
6.3.1.7 Managing system administrators

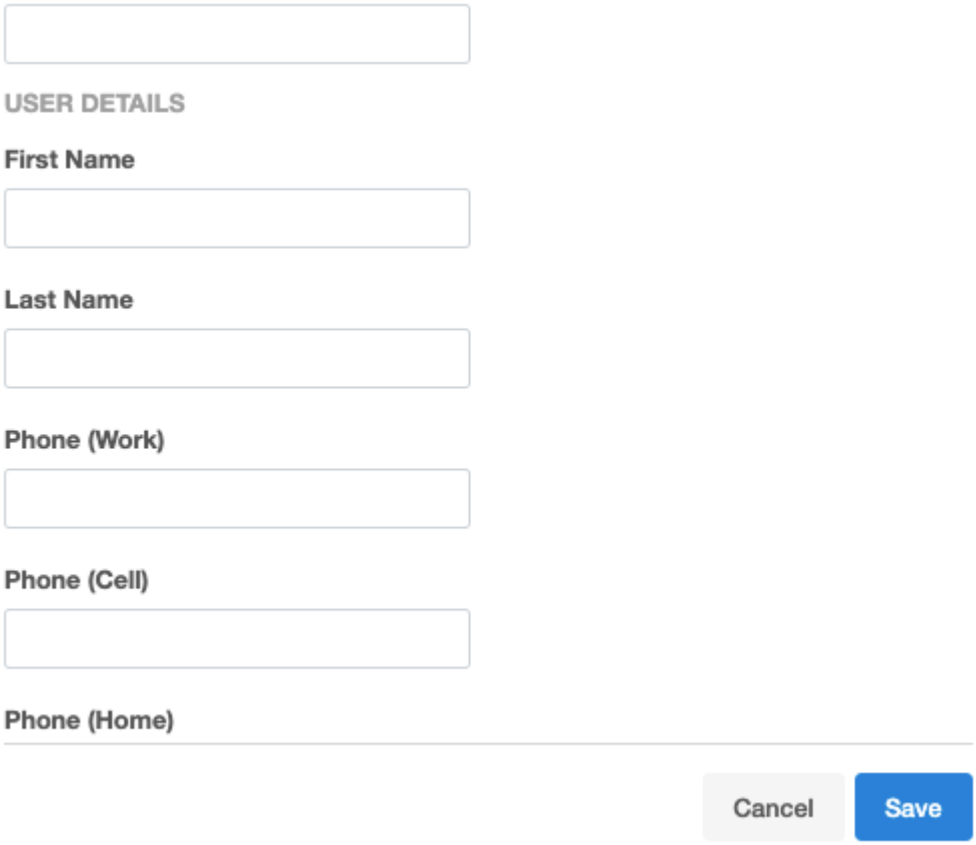
6.3.1.7.1 Adding a new system administrator

1. Launch the **Setup** application.
2. From the **Dashboard** select the **+** icon located in the **Users** card next to the filter field.



3. Enter the required information.

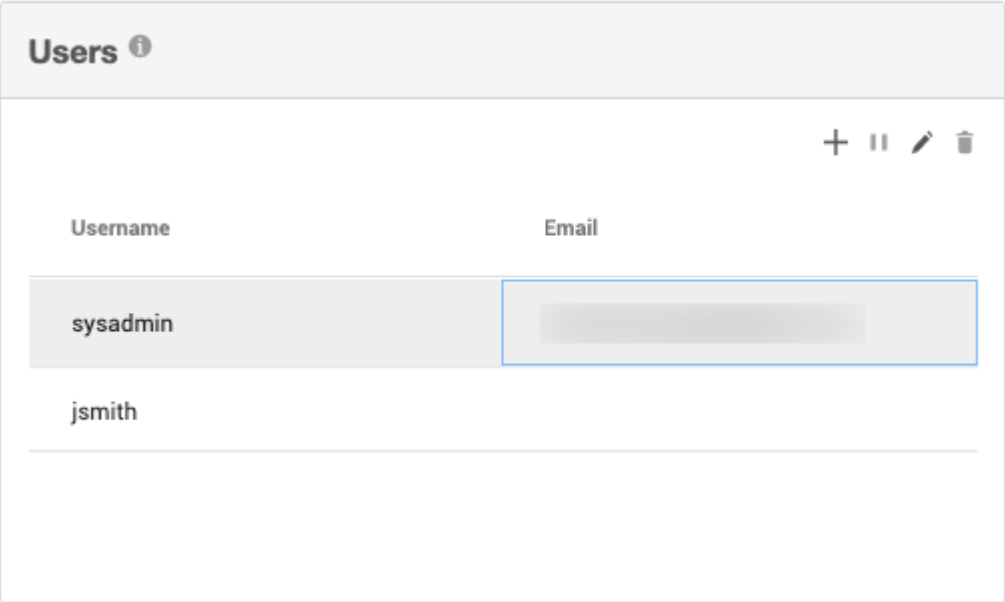




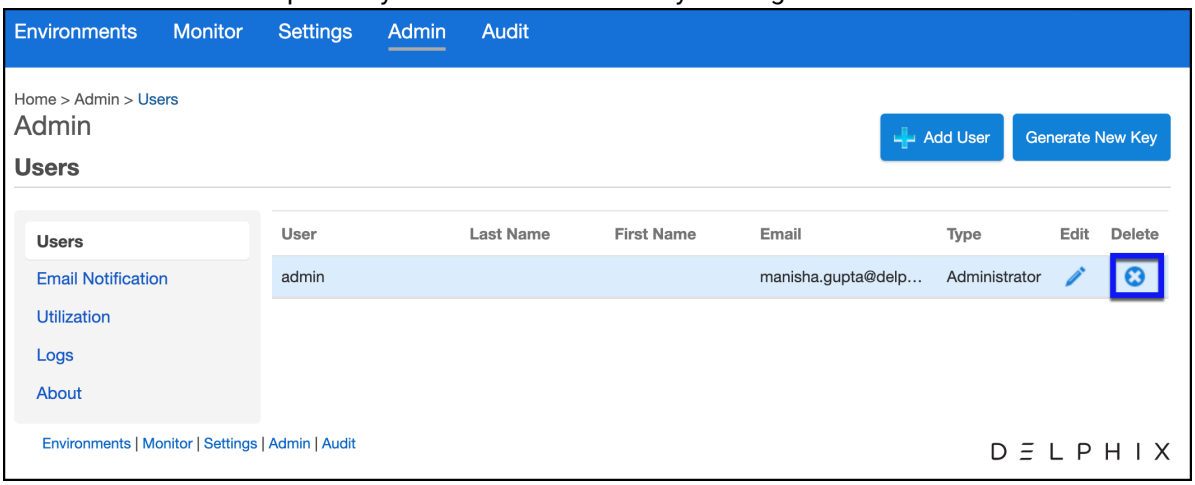
4. Click **Save**.

6.3.1.7.2 Deleting and suspending system administrators

1. Launch the **Delphix Setup** application and log in using the **sysadmin** (or another system administrator) credentials.
2. In the **User** panel, click the user you want to suspend or delete.
3. Suspend the user by clicking the **pause** icon
   ❚❚



4. Once the user has been paused you can delete the user by clicking the **trash can**.



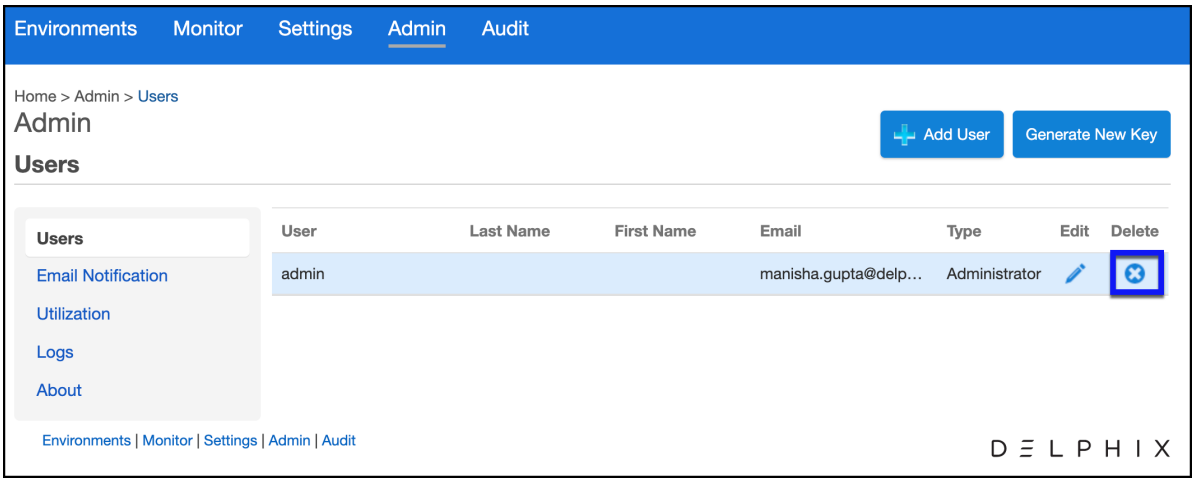
5. From the Delete User dialog select **Delete**.

Suspending the sysadmin user

> The sysadmin user is a required user for the Delphix Engine. This user cannot be deleted but can be suspended. Suspending the sysadmin user prevents that user from being able to log into **Delphix Setup** or to the console via **ssh**.

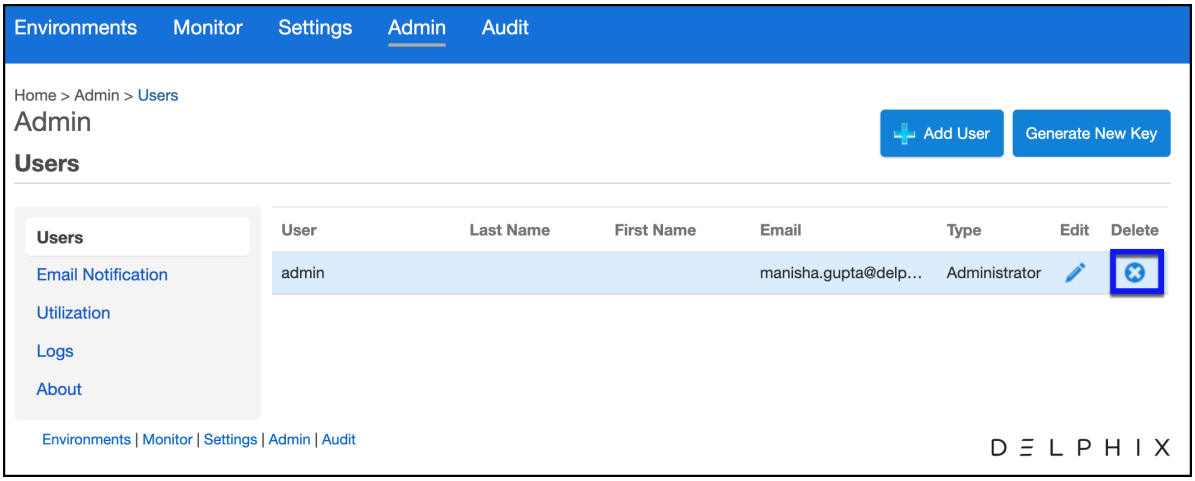
6.3.1.7.3 Reinstating system administrators

1. Launch the **Delphix Setup** application and login using system administrator credentials.
2. In the **User** panel, click on the name of the user you want to reinstate.
3. Reinstate the user by clicking the **enable user** icon

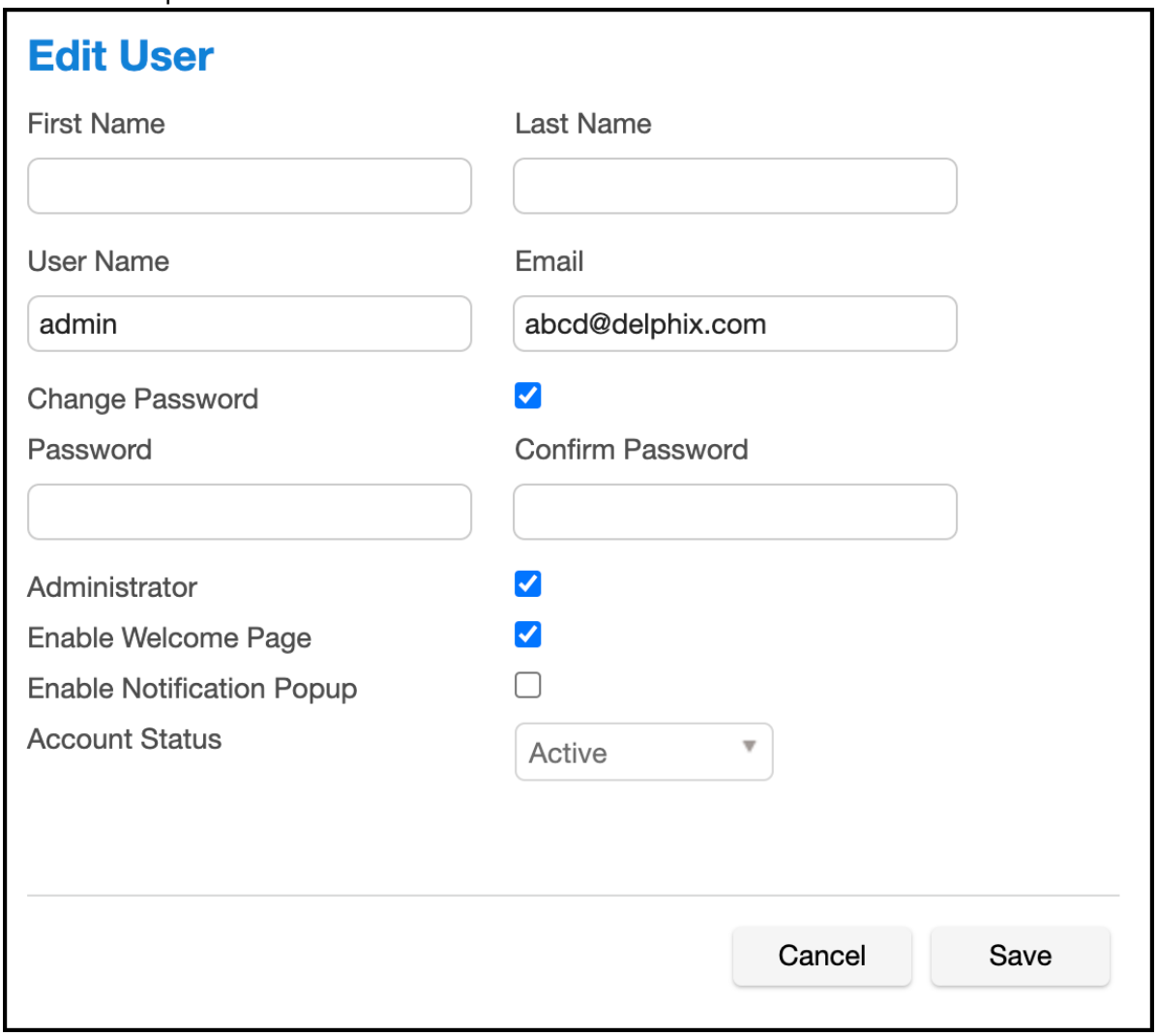


6.3.1.7.4 Changing system administrator passwords

1. Launch the **Delphix Setup** application and log in using **sysadmin** level credentials.
2. In the **User** panel, click the user whose password you want to change.



3. Select the **Edit** icon.
4. Enter the new password fields.



5. Click **Save**.

6.3.1.7.5 Adjust session timeout (command-line only)

Configuring session timeout for System Setup users is not available in the GUI. As such, the only method is via CLI using the commands below.

```
Delphix> user
Delphix user> select sysadmin
Delphix user 'sysadmin'> ls
Properties
    type: User
    name: sysadmin
    apiUser: true
    authenticationType: NATIVE
    emailAddress: user.name@domain.com
    enabled: true
    firstName: (unset)
    homePhoneNumber: (unset)
    isDefault: true
    lastName: (unset)
    locale: en-US
    mobilePhoneNumber: (unset)
    passwordUpdateRequest: NONE
    principal: sysadmin
    publicKey: (empty)
    reference: USER-1
    sessionTimeout: 30min
    userType: SYSTEM
    workPhoneNumber: (unset)

Operations
delete
update
disable
enable
updateCredential
Delphix user 'sysadmin'> update
Delphix user 'sysadmin' update *> set sessionTimeout=180
Delphix user 'sysadmin' update *> commit
```

6.3.1.8 Managing Delphix users

This section describes how to manage users. Here, you can learn how to:

- Add Users (see page 0)
- Edit, Delete and Suspend Users (see page 0)
- Manage Profile Information (see page 0)
- Delphix User Account Lockouts (see page 0)

6.3.1.8.1 Adding users

6.3.1.8.1.1 Prerequisites

If you intend to validate user logins using LDAP authentication, make sure a system administrator has configured LDAP.

6.3.1.8.1.2 Procedure

1. Launch the **Delphix Management** application.
2. Click **Manage**.
3. Select **Users**.
4. Click plus icon to **Add User**.
5. Enter the mandatory fields **Username**, **Email Address**, and **New Password** for the new user.
   a. Rules for creating a username:
      - Must be between 1 to 256 characters.
      - Can be just letters, just numbers, or just any of the following special characters (_, -, ., @) or a combination of all of these. For example, a username could be just "@".
      - Can start with any of the above-listed characters and is case-sensitive.

      Your password has no restrictions.
6. Select the **User Type.**
7. Click **Next.**
8. In the **Privileges** tab enter the privileges for the user.
9. Click **Next** and review the summary.
10. Click **Submit.**

Assigning owner and provisioner privileges
Assigning owner privileges at the group level conveys ownership privileges over all objects in that group. Click the **expand** icon next to each group name to see all objects in that group.
You can also assign ownership privileges only for specific objects in a group. You do not have to assign owner or auditor privileges for all Delphix objects, only those for which you want to grant the user-specific access.

6.3.1.8.2 Editing, deleting, and suspending users

> **The delphix_admin user**
> The user named **delphix_admin** cannot be deleted since this is a user created by the Delphix Engine. However, you can suspend it.

> When engines created before 5.3.1 are upgraded to 5.3.1 or later they will retain their old username 'delphix_admin'. To avoid complications Delphix recommends creating users with an admin role and then Disabling delphix_admin.

1. Launch the **Delphix Management** application.
2. Select **Manage > Users**.
3. Click the user's name to open the user's profile panel.
4. Click the **disable** icon to disable the user.
5. Click the **trash can** icon to delete the user.

> Deleting a user cannot be undone.

6.3.1.8.3 Managing individual profile information

1. After logging in, click your name in the menu bar.
2. Click **Profile**.
3. Edit profile information as necessary.
4. Select options for the event level that will trigger a notification email.
5. Select a time period for **Session Timeout**.
6. Click **Password** to edit your password.
7. Click **OK** when finished.
8. Click **Privileges** to see your privileges (Auditor or Owner) for Delphix objects.

6.3.1.8.4 Delphix user account lockouts

6.3.1.8.4.1 User account lockouts

This feature applies to all kinds of users – Delphix and LDAP. It also applies to usernames that do not correspond to any user in the system. A user who enters a wrong password three times in a row is "locked out" (i.e., unable to continue attempting to log in) for an initial period of 30 seconds. After three more bad login attempts, the user must wait 60 seconds, then 90 seconds, and so on.

6.3.1.8.4.2 Troubleshooting a user account lockout

The initial wait time for any future lockouts is reset to 30 seconds when the user successfully logs in or when an administrator resets the user's password. When an administrator resets a locked-out user's password, the user can immediately attempt to log in.

6.3.1.8.5 Self-service password reset

This feature allows users to independently reset their forgotten passwords to reduce the need for administrator intervention.

The password reset functionality is integrated into the login interface. Users will find a **Forgot Password?** link on the login page, which initiates the password reset process. This feature's availability can be controlled through a feature flag, and a configurable validity period for password reset tokens is implemented.

Additional notes

- Non-admin users can reset their passwords without needing administrator assistance.
- Users are required to validate their email identity for password resets, linking each user account with an email address.

6.3.1.8.5.1 Security considerations

- Self-service password reset should record source network addresses for complete audit records.
- Requests from unsecured clients (HTTP) are automatically blocked, ensuring password reset is only accessible through secure connections (HTTPS).
- Rate-limiting and monitoring are implemented to prevent abuse and to ensure secure handling of password reset requests.

6.3.1.8.5.2 User interface

- **Login page:** A *Forgot Password?* link directs to the password reset request page.
- **Password reset request page:** Users enter their username or associated email address. Upon submission, an automated response indicates that password reset instructions will be sent to the registered email if the provided information matches an account.
- **New password setting page:** Users will be prompted to enter and confirm a new password, adhering to the engine’s password policy requirements.

Additional notes

- **Password Reset Validity:** Administrators can configure the validity period for password reset tokens.
- **Password Reset Emails:** These emails include a unique token for resetting the password, an explanation of the token's validity period, and guidance for users who did not initiate the reset request.

6.3.1.9 Managing individual profile information

This topic describes how individual users can manage personal settings such as personal information, passwords, event notifications, and session timeouts. It also describes how users can view their privileges for Delphix objects.

6.3.1.9.1 Procedure

1. After logging in, click your name in the menu bar.
2. Click **Profile**.
3. Edit profile information as necessary.
4. Select options for the event level that will trigger a notification email.
5. Select a time period for **Session Timeout**.
6. Click **Password** to edit your password.
7. Click **OK** when finished.
8. Click **Privileges** to see your privileges (Auditor or Owner) for Delphix objects.

6.3.2 Authentication mechanisms

There are various authentication mechanisms that Delphix supports besides signing in via username and password to engines via the GUI, CLI or API. Below, you'll find documentation for the following capabilities:

- Auto-authentication via SSH keys: to automatically sign in to the Delphix CLI without requiring user-input credentials
- LDAP: Directory-based authentication to Delphix engines rather than the default local access.
- Single Sign-on: Integration and support for identity providers to authenticate users.
- Kerberos: Authentication for environments and data sources using Kerberos.
- OAuth2: Offers an alternative, password-less authentication method for API access to the Delphix Engine.

Kerberos support

Kerberos support is for access to connected environments, rather than the Delphix Engine itself. This is an advanced topic and will require a solid understanding of Delphix concepts and architecture.

6.3.2.1 How to setup auto-authentication

Generally, users need to enter a username and password when logging into the Delphix CLI. There are situations in which users may find entering a password cumbersome, or manual password entry may not be possible. These situations can be alleviated by setting up auto-authentication for the Delphix CLI.

There are two basic steps:

1. Generate a public and Private RSA key pair.
2. Register the public key with the specific Delphix Engine user.

There are two methods available:

- PuTTY
- OpenSSH with OpenSSL

If the examples provided do not work for you, you may need to consult your SSH documentation, we can only provide support for the Delphix Engine side of the connection. In both examples we grant password less login to the **sysadmin** user to host **Delphix5010**.

6.3.2.1.1 Using PuTTY

You will need both **putty.exe** and **puttygen.exe** for this.

Launch **puttygen.exe**

Set the **Type of key to generate** to *SSH-2 RSA* and the **Number of bites in generated key** to a suitable value such as *2048*. Click **Generate**

Once it has generated the key pair, leave the password fields blank and save the public and private keys to file.

Add the full contents of the public key to each Delphix Engine user you want to allow automatic login for.

```
Delphix5010> user Delphix5010 user> select sysadmin Delphix5010
user 'sysadmin'> update Delphix5010 user 'sysadmin' update> set
publicKey="ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEAjdQYr1WU6UPr6FZqyt3eKNJEkAe8IdKQ8hcuBWa3HvRVmUuv0L
ykm5AYQlIW0B33aWusr0o+2FVTzt3/6G1lLCf7wfhCShlJsYgwgmHeEGjixK5tacFCD8r+8dALaXlv
8uOlddK0A2LPXbCCCIRL7IyVEnlSbUFY8s+E/
2R3owy5XSbLJLE1el5m1lQPOyUuQddAh25ruWR+1HHSAWG3p+wofOh6l7czkEcq7fPjtAZvivX90e8
Ggt6JQ8bv6td7aJWObU2Y9YY0HLLHot7NQ4AT/
0tXSRKAG8sIdL7tY9hbHMNHRftCLzfn7mL+Qk8TjUYni3JGB4Vyi0bmkj6nHQ== rsa-
key-20160309" Delphix5010 user 'sysadmin' update> commit
```

1. In Putty, create a profile that uses the private key. In the PuTTY Connection settings set **SSH** > **Auth** >*Private key file for authentication* to the private key file you just generated
2. Next, still in the PuTTY Connection settings set **Data** > *Auto-login username* to `sysadmin@SYSTEM`.
3. Test the connection by setting the connection hostname as you normally would for PuTTY and click **Open**.

```
Connection > SSH . Auth
```

6.3.2.1.2 Using OpenSSH with open SSL

Generally, OpenSSH will already have default public and private keys that can be used, if not (or the default keys are password locked) you can create them this way. OpenSSL is required but OpenSSH will take care of the background OpenSSL stuff for you.

1. Create your RSA key pair to a bit length suitable for your security needs (2048 is commonly required for recent security audits)

```
$ ssh-keygen -b 2048 -t rsa -P ' ' -f /etc/ssh/ssh_host_rsa_key
```

Results in a matching public file called `/etc/ssh/ssh_host_rsa_key.pub` If you want to create different key pairs, just specify a different file path.

2. Add the full contents of the public key (`/etc/ssh/ssh_host_rsa_key.pub` in this example) to each Delphix Engine user you want to allow automatic login for.

```
Delphix5010> user Delphix5010 user> select sysadmin Delphix5010
user 'sysadmin'> update Delphix5010 user 'sysadmin' update> set
publicKey="ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEAjdQYr1WU6UPr6FZqyt3eKNJEkAe8IdKQ8hcuBWa3HvRVmUuv0L
ykm5AYQlIW0B33aWusr0o+2FVTzt3/6G1lLCf7wfhCShlJsYgwgmHeEGjixK5tacFCD8r+8dALaXlv
8uOlddK0A2LPXbCCCIRL7IyVEnlSbUFY8s+E/
2R3owy5XSbLJLE1el5m1lQPOyUuQddAh25ruWR+1HHSAWG3p+wofOh6l7czkEcq7fPjtAZvivX90e8
Ggt6JQ8bv6td7aJWObU2Y9YY0HLLHot7NQ4AT/
0tXSRKAG8sIdL7tY9hbHMNHRftCLzfn7mL+Qk8TjUYni3JGB4Vyi0bmkj6nHQ== rsa-
key-20160309" Delphix5010 user 'sysadmin' update> commit
```

3. Test no password login on the command line from your client.

```
$ ssh -i /etc/ssh/ssh_host_rsa_key sysadmin@Delphix5010
```

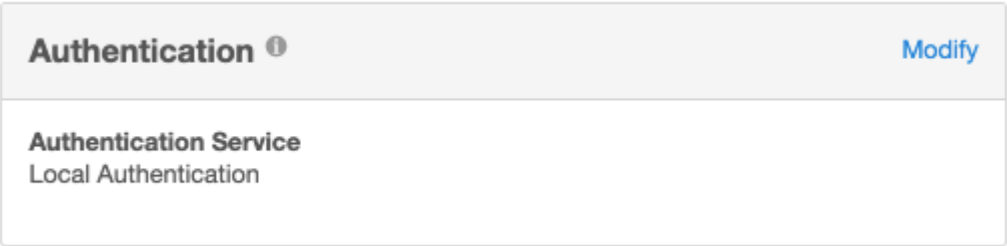
6.3.2.2 Configuring and using LDAP with the Delphix engine

Using LDAP with the Delphix Engine requires the following:

- configure the Delphix Engine to use LDAP
- add LDAP users in the Delphix Management application

6.3.2.2.1 Configuring LDAP on the Delphix engine

1. From the Delphix Setup application configure the LDAP server with the Delphix Engine by selecting Modify in the Authentication section.



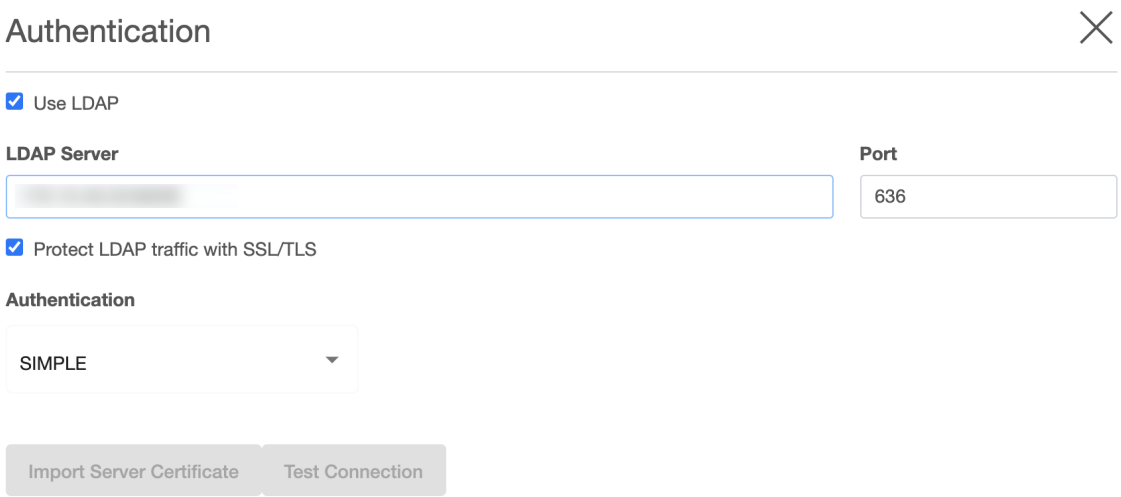
2. Enter the information about the LDAP Authentication Server. This must be an LDAP server that is configured for authentication. This information should come from the LDAP admin who runs the server. As a general rule only use simple auth. If using SSL/TLS typically use port 636 and import a certificate. If not using SSL/TLS, use port 389 and you will not need a certificate. If the remote LDAP server has disabled anonymous access and the user is trying to use SSL/TLS, the user will be unable

to import the certificate. If this occurs file a support case so that Delphix Support can help manually upload the certificate.

Note:

Import Server certificate option may import more than one certificate. It is recommended to import the CA certificate in the TrustStore and then click the Test Connection option to validate. For more information on adding a certificate in the TrustStore, refer to TrustStore Settings (see page 989).

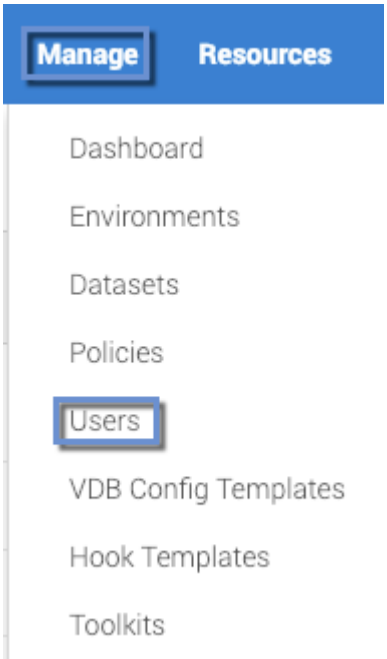
Test Connect will issue an anonymous login request to the LDAP server. If the LDAP server has disabled anonymous access the test will fail. Test the server by adding a valid LDAP user and try logging in.



3. After updating the information and clicking the **Save** button, the *Authentication Service* section should reflect the proper information.

6.3.2.2.2 Create a new LDAP user account

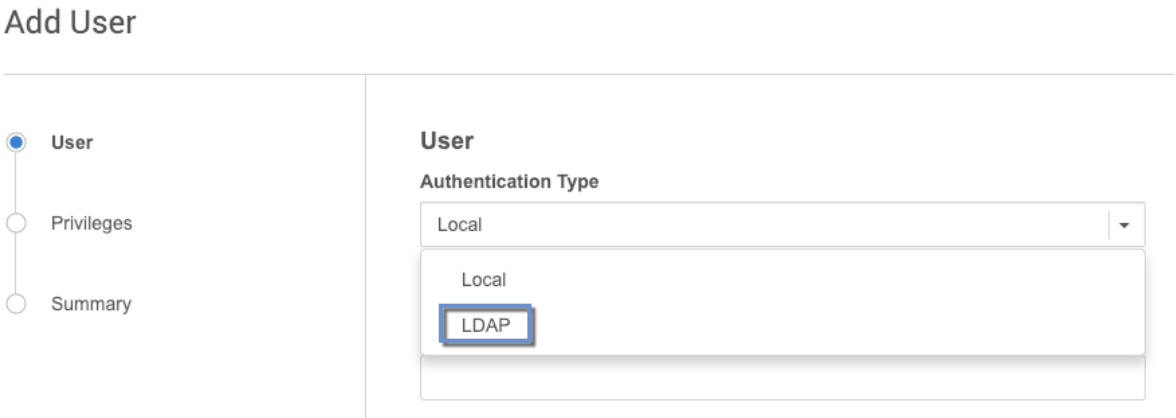
1. Login to the Delphix Management application and go to **Manage > Users** to add a new user.



2. In the **Users** screen, click the **Add User** icon

   +

   and choose **LDAP** as the *Authentication Type*.



3. Fill out the data fields and decide if the user will be a *Delphix Admin*. For more info on the *Delphix Admin* setting please see Managing System Administrators (see page 680). When adding the principal, it is mandatory to specify the entire DN of the user to be added.

Each entry in an LDAP tree has a unique identifier; its Distinguished Name (DN). This consists of its Relative Distinguished Name (RDN), constructed from some attribute(s) in the entry, follow the parent entry's DN. Think of the DN as the full file path and the RDN as its relative filename in its parent folder (e.g. if /foo/bar/myfile.txt is the DN then myfile.txt would be the RDN). Some users prefer to use the term fully qualified DN to emphasize that a proper DN should include all of the components.

6.3.2.2.2.1 Example of LDAP tree in which the base is:

dc=example,dc=com

and people are stored in a People subtree with RDN:

ou=people

and each individual is keyed by the cn (common name) attribute.

An example DN in this case would be: cn=Tony,ou=people,dc=example,dc=com

When adding an LDAP user you will be asked for the following information:

- Principal - which is the DN from above
- email address
- username - used to login into Delphix

A password is no longer required because it will authenticate against the password already stored in the LDAP entry, which is presumably known to the individual already. It is probably best if someone familiar with the LDAP tree and using it for authentication were involved at least initially to help understand how to describe the fully qualified DN for users.

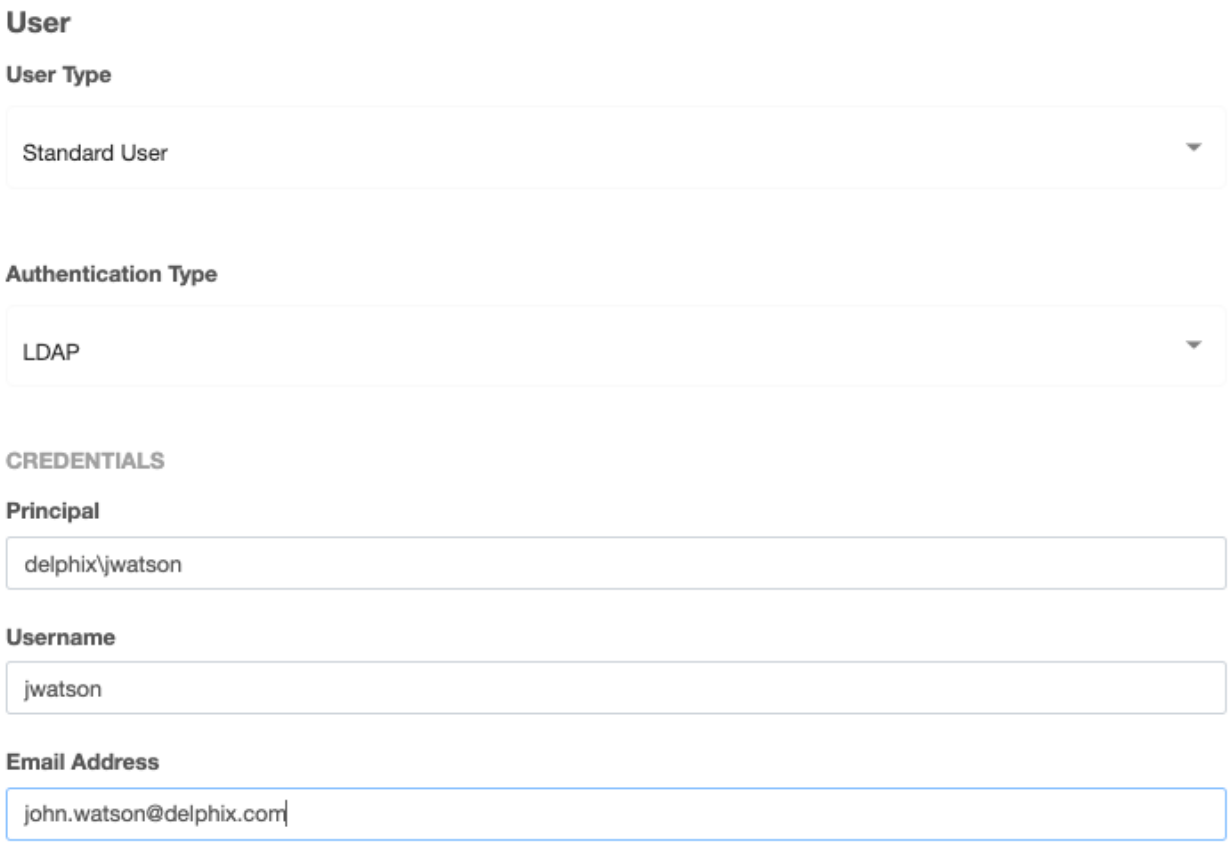
6.3.2.2.3 Using Microsoft AD as an LDAP server

Using Microsoft AD as a modified LDAP server is also possible. Microsoft AD allows some shortcuts in the specification of the DN when binding.

Examples:

- <domain>\<user logon name>
- <user logon name>@<domain>.com

As with generic LDAP, it is probably best if someone familiar with using the AD LDAP instance for authentication was involved.



When users log in, they will enter the username as chosen above, and the password that matches the principal entered above.

6.3.2.3 Configuring single sign-on

6.3.2.3.1 Overview

Continuous Data and Continuous Compliance Engines above version 5.3.3 support authentication via the SAML 2.0 standard (SP initiated and IdP initiated).

This page provides instructions on how to set up Single Sign-on (SSO) on the Data Engine. Single Log-out (SLO) is not supported. This means that logging out of a Data Engine will not terminate sessions on other Data Engines, nor will it terminate the IdP session.

6.3.2.3.2 Identity provider configuration

The steps to configure an Identity Provider (IdP) are specific to each IdP product (e.g. Okta, OneLogin, PingFederate). The terminology may vary, but one SAML 2.0 application (or SP connection) will need to be created for each Data Engine. The engine does not expose a metadata document. The following attributes must be entered:

- **ACS URL (Assertion Consumer Service URL)**: `http(s)://<delphix-engine>/sso/response`

    - Delphix strongly recommends that HTTPS is used instead of HTTP for all UI and API communication with the Data Engine. For HTTPS or the automatic HTTP to HTTPS redirect, use the `https://` scheme in the ACS URL, otherwise use the `http://` scheme. Refer to the CLI Cookbook: Changing HTTP and HTTPS web connections[226] page on how to set up HTTPS.
- **SAML Bindings**: Data Engines support the POST and redirect bindings.
- **Audience Restriction (SP entity ID, Partner's Entity ID)**: The audience restriction must be set to the entity id configured in the Delphix Server via the Delphix Setup (see below). The default value is `https://<Delphix Server ID>` where **<Delphix Server ID>** is a 36-character hexadecimal string of the form xxxxxxxx-xxxx-xxxx-xxxxxxxxxxxx. See the Determining the Delphix Server ID and Host Name[227] for more on the Delphix Server ID.
    - If the Data Engine does not exist or is unreachable, enter a temporary value (such as `delphix-sp-id`) to later be replaced by the actual Delphix Server ID.
- **Signature policy**: The Data Engine does not sign authentication requests; it requires that either the responses, assertions, or both are signed.
- **Name ID**: The SAML `NameID` attribute must be set to the email address of the user. See the *User management when SSO is enabled* section below for more information.

The SP initiated flow must be enabled in the IDP.

6.3.2.3.3 New engine configuration

Follow this procedure when installing a new Data Engine.

1. Connect to the Data Engine at `http://<DataEngine>/login/index.html#serverSetup`. The **Delphix Setup** application will launch when connecting to the server.
2. Enter the **sysadmin** login credentials; this account has a default username of **sysadmin** and password of **sysadmin**.
3. In the Authentication step of the Delphix Setup wizard, check the **use SAML/SSO** box and enter the required information:
    a. The entity id is a unique identifier of the Data Engine for SSO providers. The default value is **https://<Delphix Server ID>**, where <Delphix Server ID> is a 36-character hyphen-separated string in the form of 8 chars, 4 chars, 4 chars, 4 chars, and 12 chars (e.g. `00000000-0000-00aa-aa00-0000aa0000aa`). This is just an identifier; there is no resource at that URL. This value can be changed to any string, but note that some identify providers require this to have a URL format.
    b. The IdP metadata is an XML document that must be exported from the application created in the IdP (see the *Identity provider configuration* section above).
    c. Optional **Advanced** settings include the **response skew time**, which is the maximum time difference allowed between a SAML response and the engine's current time, in seconds. If not set, it defaults to 120 seconds. The **maximum age of IdP authentication** indicates how far in

226 https://cd.delphix.com/docs/19.0.0.0/cli-cookbook-changing-http-and-https-web-connectio
227 https://cd.delphix.com/docs/19.0.0.0/determining-the-delphix-server-id-and-host-name

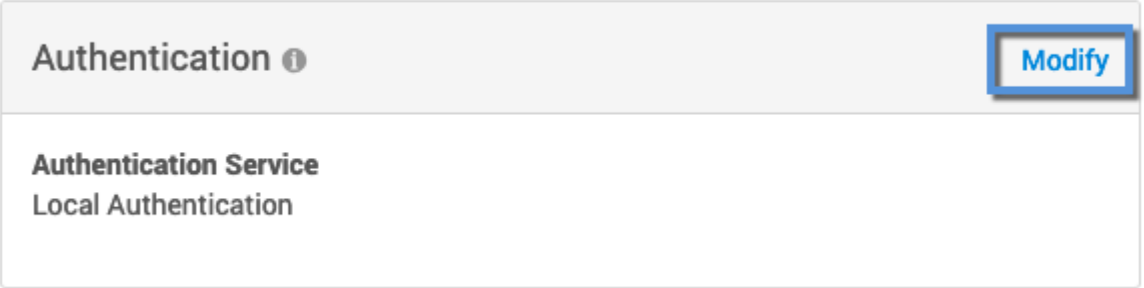
the past to accept authentications to the identity provider, in seconds. If not set, it defaults to 86,400 seconds (one day).

4. Complete the remaining setup steps as usual.

6.3.2.3.4 Existing engine configuration

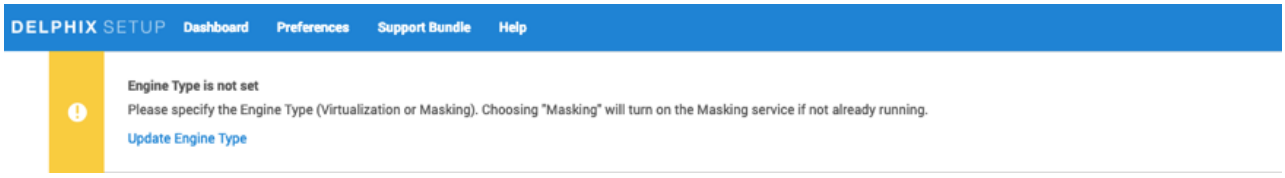
Follow this procedure to enable SSO on an already configured engine.

1. Connect to the Data Engine at `http://>>`. The **Delphix Setup** application will launch once connected to the server.
2. Enter the **sysadmin** login credentials.
3. In the **Authentication** tile, select **Modify**.



4. In the Authentication dialog, check the use SAML/SSO box, then enter the entity id (if not using the default), IdP metadata, and the optional advanced time settings (described in the *New engine configuration* section).

> **Engine type**
>
> If this is an upgraded engine, make sure the engine type is set from the banner of the ServerSetup application dashboard.



6.3.2.3.5 User management when SSO is enabled

Access to the Delphix Setup application is not affected by the use of SSO, it only affects access to the Delphix Management Application and Compliance Application. When SSO is enabled, authentication to the Delphix Management Application or Compliance Application UIs are performed via SAML/SSO instead of a combination of username and password. Non-administrators can no longer change their email address.

An administrator must create a Delphix user for each user to whom access via SSO must be granted. The Delphix user can be used to assign roles and permissions. The email address of the Delphix user is used to match users authenticated via SAML/SSO and must be set to the exact value defined in the IdP. This same value is used in the `NameID` attribute of the SAML response. Supported `NameID` formats are as follows.

```
urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified
urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress
```

If multiple Delphix users share an email address, all Delphix users will have access to the SAML/SSO session.

Once SSO is enabled, usernames and passwords (LDAP or locally stored) still need to be used for API access, and newly created user will have API access disabled by default.

6.3.2.3.6 API access

Neither the API access for use in scripts and integrations nor the Continuous Data CLI access requires SSO. Instead, username and password (optionally with LDAP integration) authentication must be used for API or CLI access. When SSO is enabled on a Data Engine, new users by default will have no API access and no password. Administrators can enable API access for any user through the User Management or Masking UIs. The user's password or LDAP credentials are used for API authentication.

Users created before enabling SSO will maintain their API access. Users with API access but no email address are useful for scripts or integration via the API – they cannot be used to log in via the UI. When SSO is enabled, only administrators can change or set email addresses.

A user with API access may also log in via SAML into an SSO-enabled engine through the UI when they have an email set.

6.3.2.3.7 Troubleshooting

If authentication via SSO fails with a message stating that the issue time is either too old or in the future, the error is due to the time on the Data Engine not being in sync with the time of the Identity Provider server. If the time on the Data Engine is not correct, correct the time settings manually or configure NTP. Alternatively, update the response skew time parameter (see the *Existing engine configuration* section).

If authentication via SSO fails with a message stating that the authentication to the Identity Provider is too old, re-authenticate with the identity provider or adjust the maximum age parameter (see the *Existing engine configuration* section).

6.3.2.4 Configuring and managing kerberos

Version 6.0.7.0 or later recommended for kerberos

Any Delphix Engine intending to leverage Kerberos credentials should be running version 6.0.7.0 or later. Versions 6.0.0.0-6.0.6.1 may encounter issues in authentication ticket renewal, causing Environment and Dataset job failures. More information can be found in this Delphix Knowledge base[228] article.

This section covers the following topics :

- Delphix kerberos implementation (see page 700)
- SSH implementation (see page 702)
- Kerberos requirements (see page 704)
- Configuring kerberos[229]

6.3.2.4.1 Delphix kerberos implementation

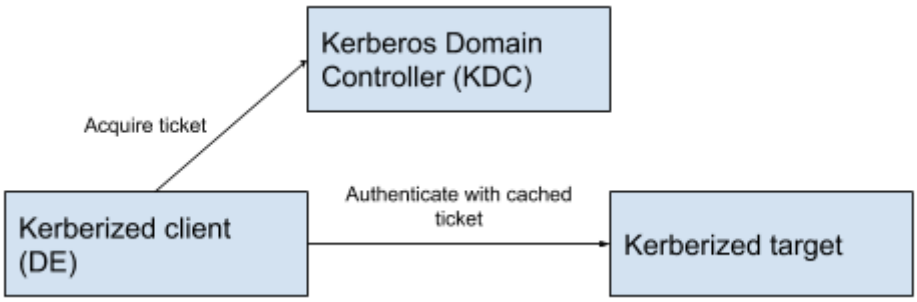
Version 6.0.7.0 or later recommended for Kerberos
Any Delphix Engine intending to leverage Kerberos credentials should be running version 6.0.7.0 or later. Versions 6.0.0.0-6.0.6.1 may encounter issues in authentication ticket renewal, causing Environment and Dataset job failures. More information can be found in this Delphix Knowledge Base[230] article

6.3.2.4.1.1 Shared infrastructure/ticket management

The Delphix Engine (DE) has a single Kerberos principal shared between all connections to the host (SSH, ASE JDBC, etc).

228 https://support.delphix.com/Delphix_Virtualization_Engine/Delphix_Admin/Environment_Authentication_Using_Kerberos_Credentials_Fails_on_Delphix_versions_6.0.0-6.0.6.1_(KBA7495)
229 https://delphixdocs.atlassian.net/wiki/pages/createpage.action?fromPageId=149523998&linkCreation=true&spaceKey=CD&title=Configuring+Kerberos
230 https://support.delphix.com/Delphix_Virtualization_Engine/Delphix_Admin/Environment_Authentication_Using_Kerberos_Credentials_Fails_on_Delphix_versions_6.0.0-6.0.6.1_(KBA7495)

6.3.2.4.1.2 Overview of the authentication process



1. The client acquires a ticket from the Kerberos Domain Controller (KDC) (e.g kinit <principal>) which it stores locally.
2. The client uses a ticket from KDC to authenticate with the target (e.g., ssh- or JDBC authentication using gssapi to pass the cached ticket acquired in step 1).

Kerberos master/replica KDCs

Kerberos supports a master/replica system with multiple KDCs running on different hosts. This is used for High Availability (HA) or to provide faster service via a local node in dispersed network environments. Delphix supports a list of KDCs for the Kerberos realm to which it has been joined.

Delphix infrastructure to support the authentication process

Kerberized environment user

Delphix has introduced a KerberosCredential type that indicates the global Kerberos principal to be used for authentication, rather than user-specific credentials.

Keytab based authentication

It is possible to use kinit with a keytab file instead of password-based authentication to acquire tickets. This is similar in principle to passwordless SSH authentication and allows Delphix to function in the customer's environment without storing any passwords on the Delphix Engine. It does, however, put us at the mercy of the customer's keytab expiration policy.

The Delphix Engine creates a background thread that periodically checks the expiration of the cached Kerberos credentials. If the credentials have expired, it calls kinit using the keytab that was provided.

Keytab file storage

Keytab file data is sent via a web service API as a Base64 encoded string. This is then decoded back to the binary file and persisted on local storage on the Delphix Engine with root user ownership permissions.

6.3.2.4.1.3 Default behavior

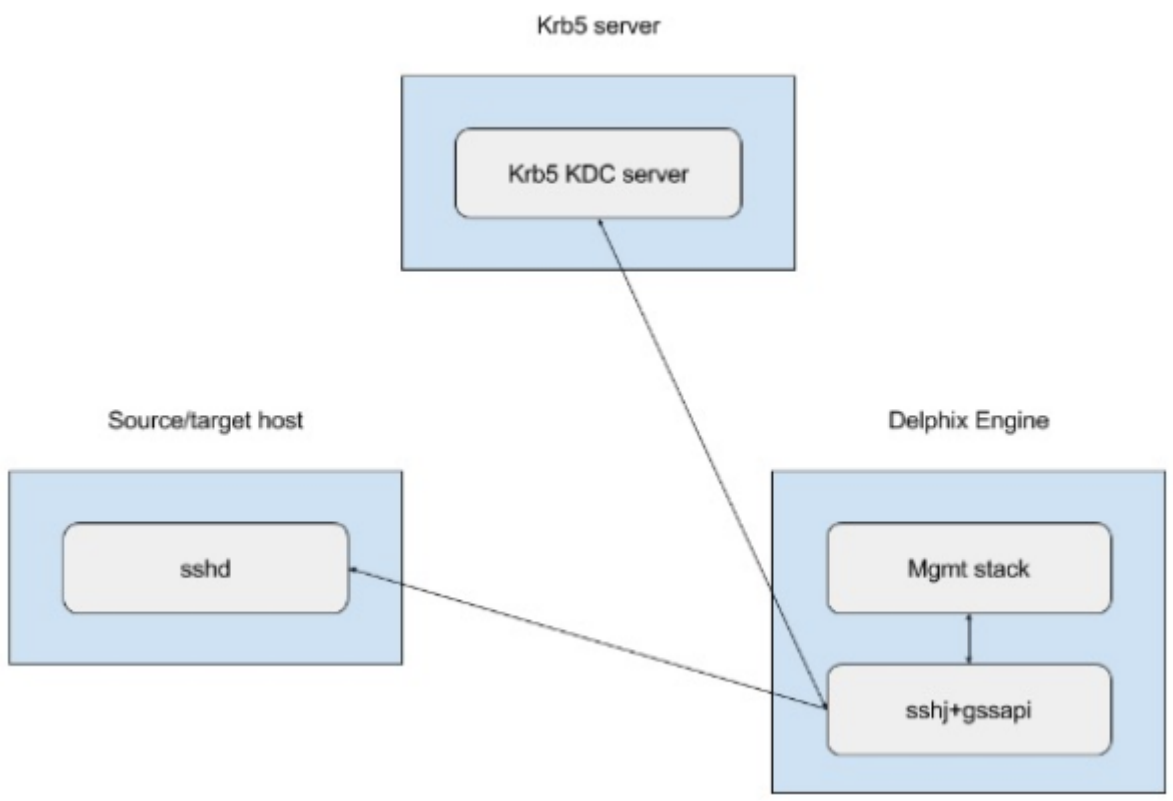
Default Kerberos ticket refresh configuration:

- Delphix checks if the TGT-cached Kerberos ticket should be refreshed every hour.
- The TGT-cached ticket for the global Delphix principal will be refreshed if it expires in less than two hours. The default values can be changed by Delphix Support.

6.3.2.4.2 SSH implementation

The management stack uses sshj+gssapi to pass already-generated Kerberos tickets to the Kerberized sshd on the source/target side if prompted to do so by the end-user passing a Kerberized environment user to existing wrapper functions.

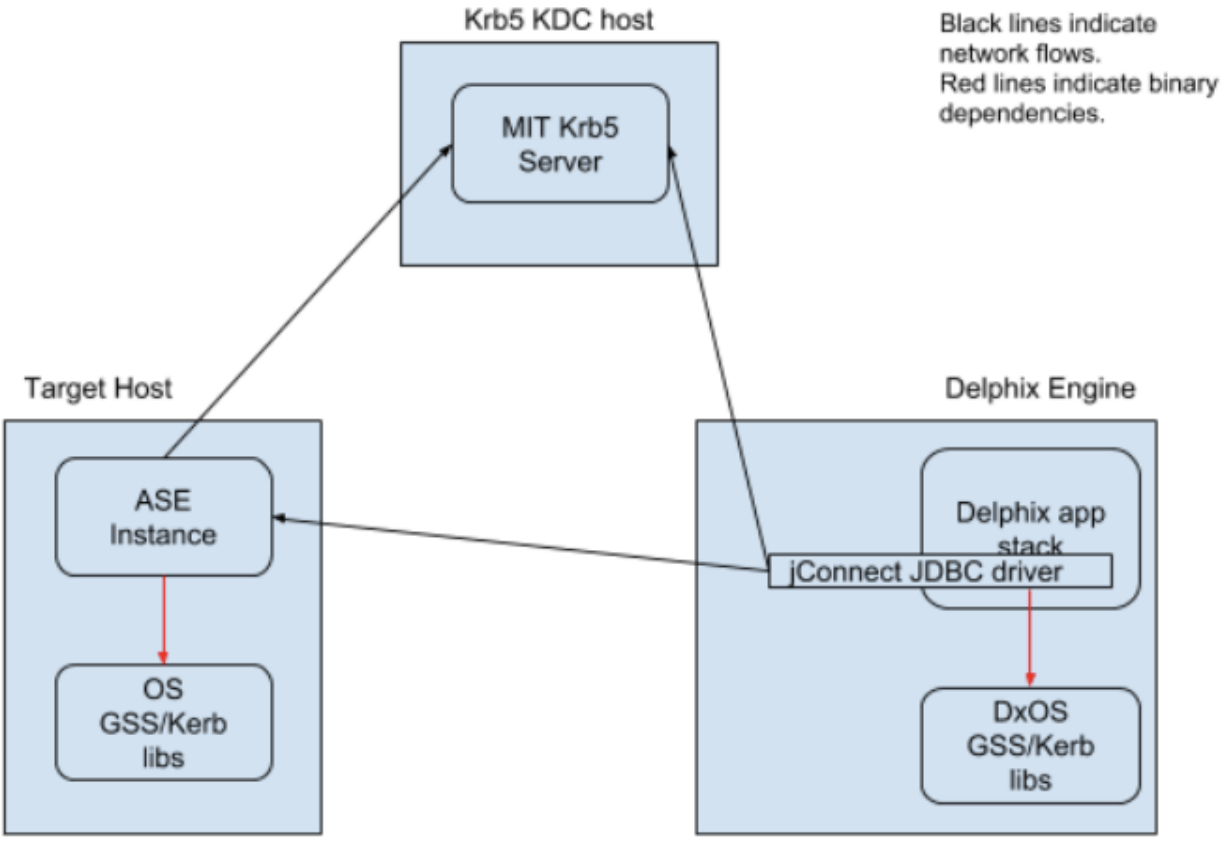
> The only thing changing from password-based or regular passwordless SSH authentication is the authentication step. Command execution remains unchanged.



6.3.2.4.2.1 SAP ASE, Oracle, and DB2 connections

Delphix connects to SAP ASE, Oracle, and DB2 instances using the two listed mechanisms below. This example configuration uses an SAP ASE instance.

- via isql process
- via the jConnect JDBC driver



When connecting via isql Delphix uses the "-V" parameter rather than specifying a username/password. The "-V" option uses the Kerberos principal in the current user's cached credentials file. Delphix relies on the end customer to configure this appropriately for their environment (for example, the cached credentials could be populated by a PAM module during login). Delphix also expects that the `KRB5CCNAME` is set appropriately or the credential cache is in the host default location.

When connecting via JDBC, Delphix uses additional connection options: `REQUEST_KERBEROS_SESSION=true&SERVICE_PRINCIPAL_NAME=`. By default, the instance Service Principal Name (SPN) is identical to the instance name for authentication. Delphix allows the instance SPN to be manually set on a per-repository basis to allow for non-default values. The jConnect JDBC driver connects using the cached credentials that were obtained as described in the Shared infrastructure/Ticket Management section.

For example, if the instance name is `ASE_INSTANCE_1` and has been configured to use `REALM.COM`, then the instance will attempt to authenticate with the KDC using `ASE_INSTANCE_1@REALM.COM`. However, this is configurable and can be specified either via an environment variable or a command-line option to the

data server process. If an environment variable is used to configure the SPN, the instance must be manually discovered via web service APIs or the Delphix CLI.

6.3.2.4.3 Kerberos requirements

6.3.2.4.3.1 Prerequisites

Basic requirements prior to the configuration:

- MIT Kerberos 1.4.4 KDC
- Kerberos `REALM` name
- Global Kerberos principal name (specified without trailing `@REALM` name)
- Global Kerberos principal `keytab` data encoded as a base 64 string
- KDC hostnames and port numbers (one or more in priority list order)

6.3.2.4.3.2 Environment requirements

> **Version 6.0.7.0 or later recommended for Kerberos**
> Any Delphix Engine intending to leverage Kerberos credentials should be running version 6.0.7.0 or later. Versions 6.0.0.0-6.0.6.1 may encounter issues in authentication ticket renewal, causing Environment and Dataset job failures. More information can be found in this Delphix Knowledge Base[231] article.

The following hosts and software versions are required:

- A source host with the following configuration:
  - A running SAP ASE, Oracle, or DB2 instance.
  - A database to link from and its corresponding full database dump.
  - The Delphix principal is able to access the SAP ASE instance and SSH onto the host as per our product documentation (see Requirements for SAP ASE (see page 1884)).
  - The credential cache for the Delphix principal is populated and kept current. The environment variable `KRB5CCNAME` is set to the location for a credential cache. Login to the Adaptive Server via "`isql_r64 -V -R>><>`" or "`isql_r -V -R>><>`" or otherwise make sure that "`isql`" points to either "`isql_r64 or isql_r`" so that "`"isql -V -R>><>` works.
- A staging host with the following configuration:
  - A running SAP ASE, Oracle, or DB2 instance with the same version as the source instance.

231 https://support.delphix.com/Delphix_Virtualization_Engine/Delphix_Admin/Environment_Authentication_Using_Kerberos_Credentials_Fails_on_Delphix_versions_6.0.0-6.0.6.1_(KBA7495)

    - The Delphix principal is able to access the SAP ASE instance and SSH onto the host as per our product documentation (see Requirements for SAP ASE (see page 1884)).
    - The credential cache for the Delphix principal is populated and kept current. The environment variable `KRB5CCNAME` is set to the location for a credential cache. Login to the Adaptive Server via " `isql_r64 -V -R>><>` " or " `isql_r -V -R>><>` " or otherwise make sure that " `isql` " points to either " `isql_r64 or isql_r` " so that " `"isql -V -R>><>` works.
- A target host to create a VDB on. Configuration details:
    - A running SAP ASE, Oracle, or DB2 instance with the same version as the source instance.
    - The Delphix principal is able to access the SAP ASE instance and SSH onto the host as per our product documentation (see Requirements for SAP ASE (see page 1884)).
    - The credential cache for the Delphix principal is populated and kept current. The environment variable `KRB5CCNAME` is set to the location for a credential cache. Login to the Adaptive Server via " `isql_r64 -V -R>><>` " or " `isql_r -V -R>><>` " or otherwise make sure that " `isql` " points to either " `isql_r64 or isql_r` " so that " `"isql -V -R>><>` works.

6.3.2.4.3.3 Supported databases and Kerberos configurations

For detailed Kerberos support please refer to Kerberos Support Matrix (see page 429)

Delphix Engine 5.3.2.0 is the first generally available customer release to support Kerberos on a subset of supported OS' with SAP ASE, Oracle, and DB2 databases.

> With the Delphix Engine 5.3.2.0 release, many of the previously unsupported GUI functions now function when Kerberos is enabled. See Configuring Kerberos via the UI (see page 705) below for details, including limitations.

6.3.2.4.4 Configuring Kerberos during engine setup

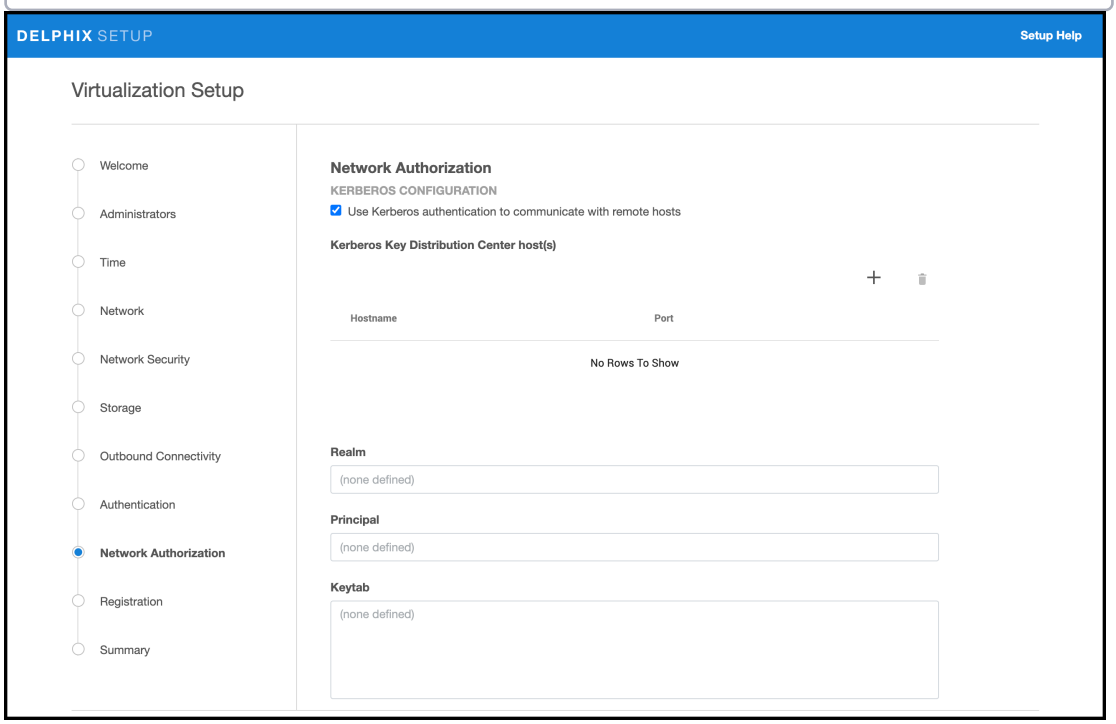
> **Version 6.0.7.0 or later recommended for Kerberos**
> Any Delphix Engine intending to leverage Kerberos credentials should be running version 6.0.7.0 or later. Versions 6.0.0.0-6.0.6.1 may encounter issues in authentication ticket renewal, causing Environment and Dataset job failures. More information can be found in this Delphix Knowledge Base[232] article.

232 https://support.delphix.com/Delphix_Virtualization_Engine/Delphix_Admin/Environment_Authentication_Using_Kerberos_Credentials_Fails_on_Delphix_versions_6.0.0-6.0.6.1_(KBA7495)

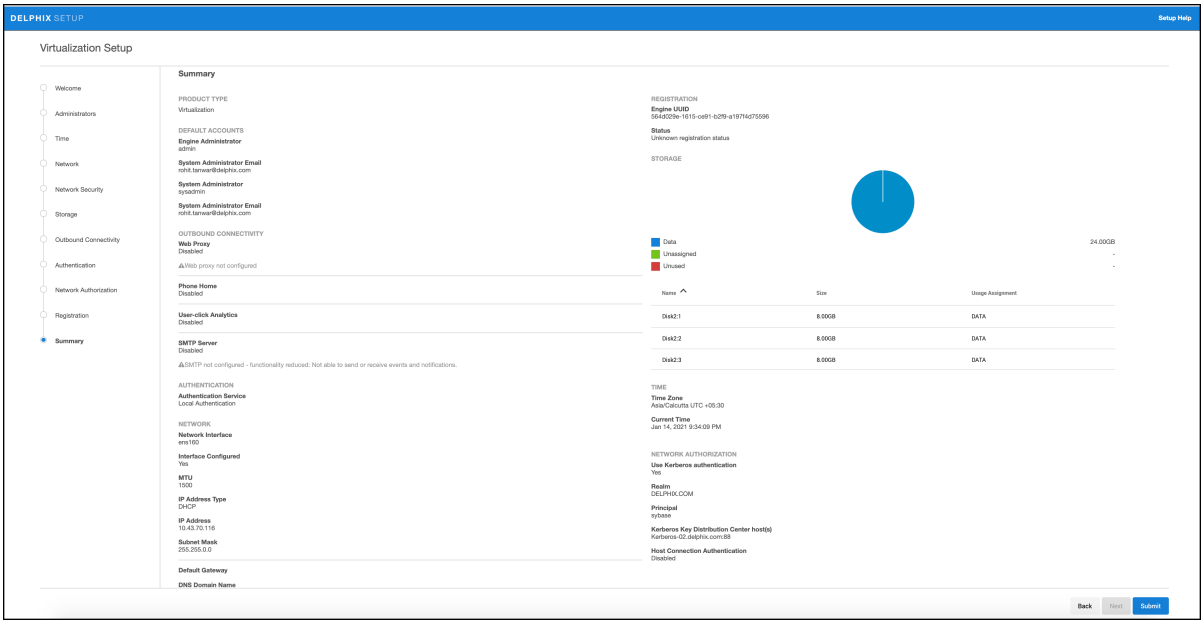
For using the kerberos authentication along with another authentication,

1. In the **Virtualization Setup** wizard, Click **Next** on the **Welcome** screen.
2. Complete the following tabs as you normally would.
3. In the **Network Authorization** tab, select the **Use Kerberos authentication to communicate with the remote hosts** checkbox.
4. Provide values in the following fields:
   - **Realm**
   - **Principal**
   - In the **Keytab** field, enter the key in a base64 encoded format.
     Note:
     You must provide the key must in a single line. For example, you can use the following command to view the base64 representation of the Keytab key.

```
echo "$(base64 --wrap=0 <keytab>)"
```



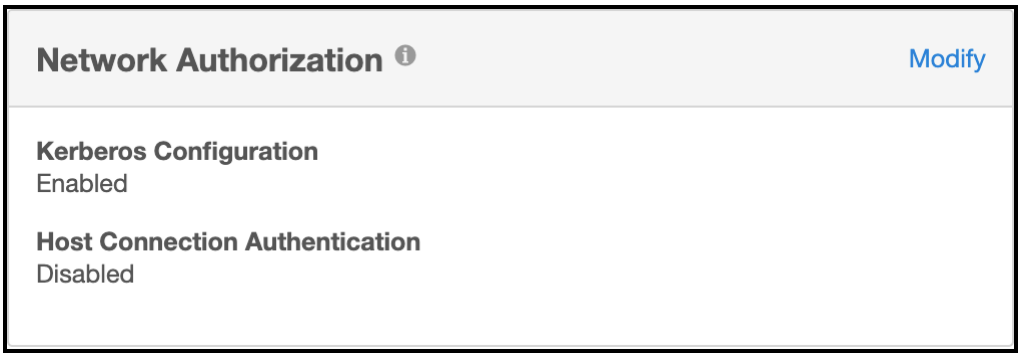
5. Then select **Next**.
6. Confirm your selections and select **Submit**.



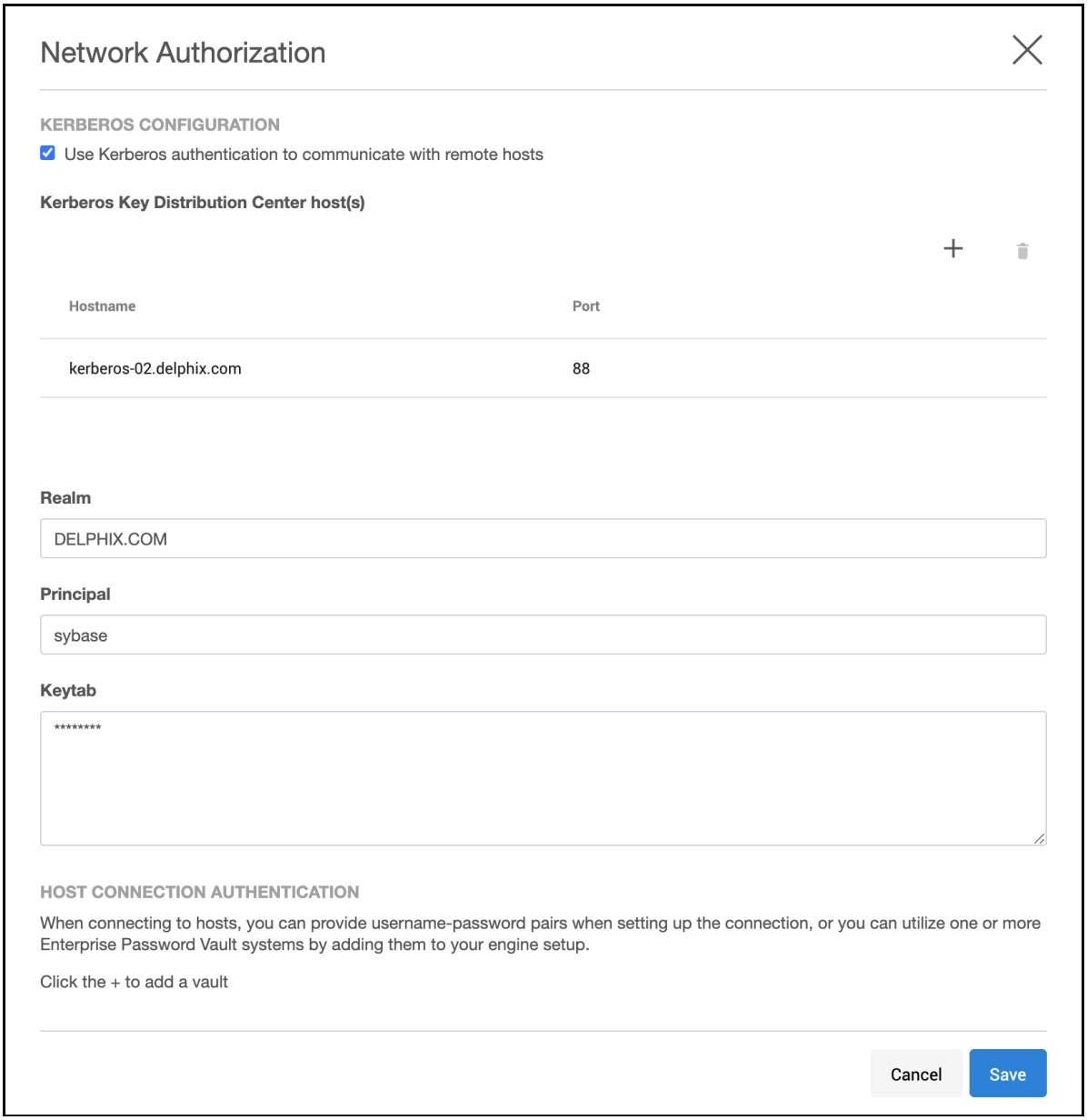
6.3.2.4.4.1 Editing Kerberos configuration settings

To edit your configuration settings after setup, login as a sysadmin user and complete the following steps:

1. On the Delphix SETUP Dashboard, from the **Network Authorization** panel, click **Modify**.



2. Select the check box under KERBEROS CONFIGURATION, select the hostname and edit the configuration.



3. Click **Save**.

6.3.2.4.5 Add a Kerberos environment

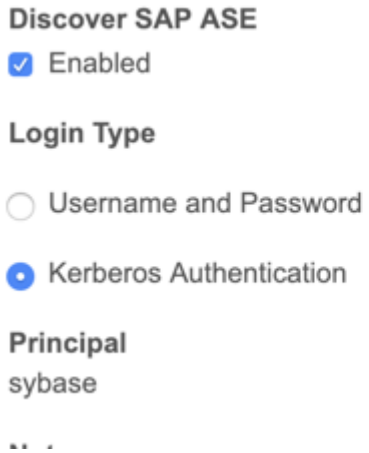
.

> **Version 6.0.7.0 or later recommended for Kerberos**

> Any Delphix Engine intending to leverage Kerberos credentials should be running version 6.0.7.0 or later. Versions 6.0.0.0-6.0.6.1 may encounter issues in authentication ticket renewal, causing Environment and Dataset job failures. More information can be found in this Delphix Knowledge Base[233] article

It is possible to create a Unix/Linux Standalone environment with Kerberos authentication.

1. Login to the **Delphix Management** application as an **admin**.
2. From the **Manage** menu, select **Environments**.
3. Then click on the plus icon to open a wizard to create a new environment.
4. Select your **Host OS** and **Server Type,** then select **Next.**
5. Under **Login Type**, select **Kerberos Authentication**.
6. If **Discover SAP ASE** is enabled, ASE DB Kerberos authentication will be available, select **Kerberos Authentication**.



7. Select **Submit**.

6.3.2.4.5.1 Changing the environment user

1. Login to the **Delphix Management** application as an **admin**.
2. From the **Manage** menu, select **Environments**.
3. Select an Environment, the Details tab allows you to see Environment information.
4. On the grid of Environment Users, you can see existing users. Click the **plus icon**, to add a new user.
5. It is now possible to create a user with Kerberos Authentication. There can only be one Kerberos user per environment. The Principal is taken to the Kerberos Configuration.

It is possible to set the Kerberos user as Primary.

233 https://support.delphix.com/Delphix_Virtualization_Engine/Delphix_Admin/Environment_Authentication_Using_Kerberos_Credentials_Fails_on_Delphix_versions_6.0.0-6.0.6.1_(KBA7495)

6.3.2.4.5.2 Changing SAP ASE DB user and DB password

1. Login to the **Delphix Management** application as an **admin**.
2. From the **Manage** menu, select **Environments**.
3. Select an Environment, the Details tab allows you to see Environment information.
4. Click the SAP ASE Information **pencil** icon.

5. Click the edit icon and edit ASE DB User and ASE DB Password.
6. Click the **checkmark** to save.

6.3.2.4.5.3 Updating environment notes

1. Login to the **Delphix Management** application as an **admin**.
2. From the **Manage** menu select **Environments**.
3. Select an Environment, the Details tab allows you to see Environment information.
4. Under **SAP ASE Information**, you can see **Notes**. To edit click the **pencil** icon.
5. Click the **checkmark** to save.

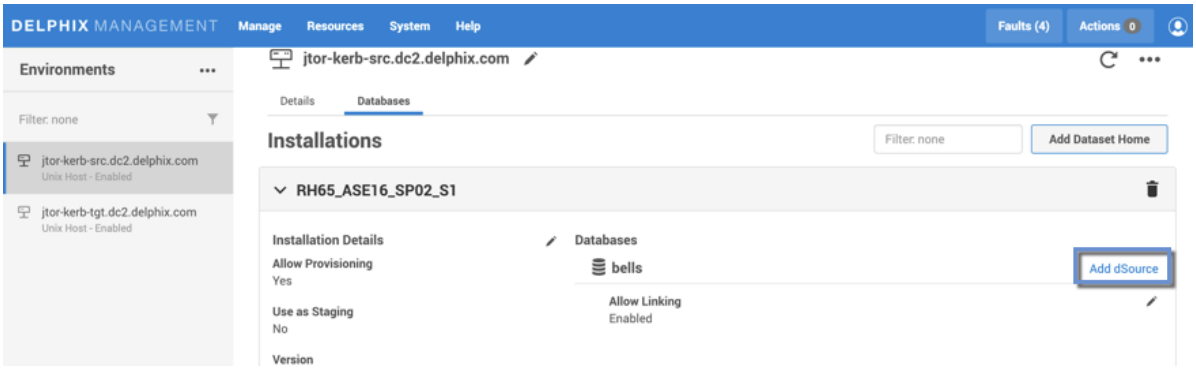
6.3.2.4.5.4 Changing host address, SSH port number, and toolkit path

1. Login to the **Delphix Management** application as an **admin**.
2. From the **Manage** menu select **Environments**.
3. Select an Environment, the Details tab allows you to see Environment information.
4. Click the **pencil** icon located next to **Attributes** to edit the Host Address or SSH Port number or Toolkit Path.
5. Select the **checkmark** to save.

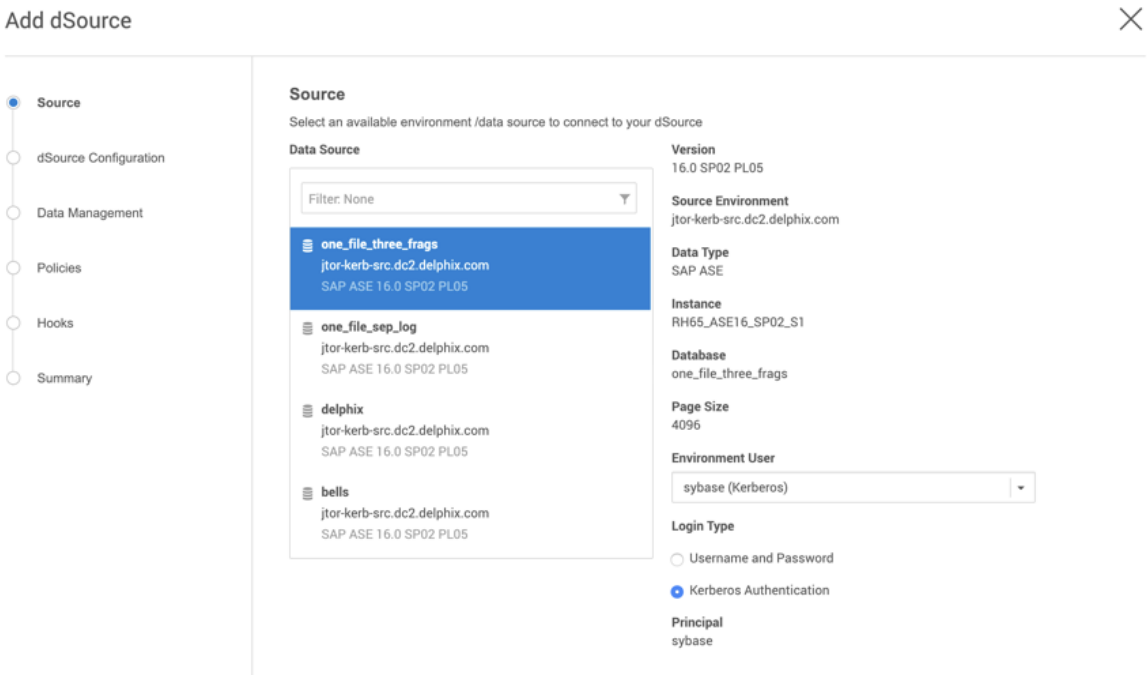
6.3.2.4.6 Adding and linking a dSource

6.3.2.4.6.1 Adding a dSource

1. Login to the **Delphix Management** application as an **admin**.
2. From the **Manage** menu select **Environments**.
3. Select an Environment and click the **Databases** tab. This tab provides details on all the available repositories.
4. Click on the **Add dSource** link.



5. Click on a **Data Source**, then select **Kerberos Authentication**.



6. Complete all remaining fields, and then click **Submit**.

6.3.2.4.6.2 Linking a dSource

1. Login to the **Delphix Management** application as an **admin**.
2. From the **Manage** menu select **Datasets**.
3. Select a **Dataset**.
4. From the Actions menu located on the top-right select **Link dSource**.

   All information in the Link dSource screen is automatically filled in with the data you previously selected.

6.3.2.5 Configuring OAuth2 authentication for API access

6.3.2.5.1 Overview

This article provides instructions on how to set up Open Authorization 2.0 (OAuth2) on the 6.0.11 Delphix Engine. OAuth2 offers an alternative, password-less authentication method for API access to the Delphix Engine.

Delphix Engine (Masking and Virtualization) version 6.0.11.0 supports authentication using JSON Web Tokens (JWTs) issued by a known authorization server or identity provider (IdP). It is necessary for JWTs to contain a claim that can be used to associate an authentication request with a user that exists in the Delphix Engine. This article describes how to configure the Delphix Engine to validate tokens and associate token claims with Delphix Engine users.

> The authentication feature described in this section differs from the API token authentication feature supported by Delphix Engines registered with Data Control Tower (*formerly* Central Management). For more information on API token authentication, refer to Data Control Tower[234]

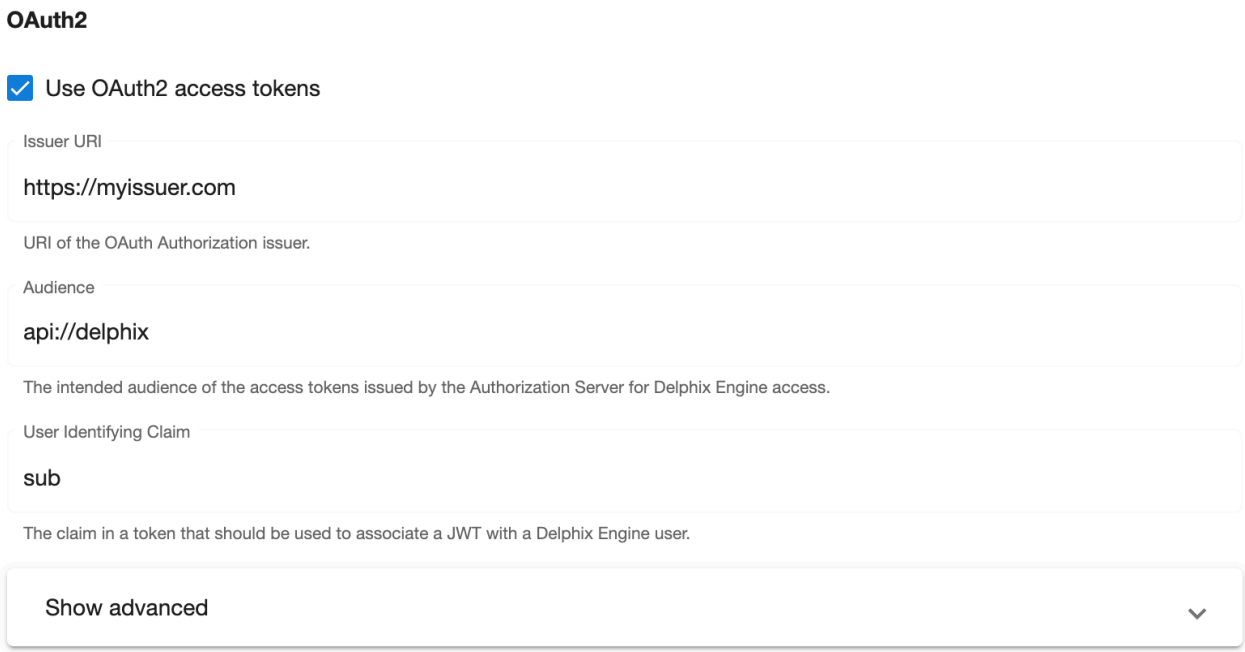
6.3.2.5.2 Configuration options

The following options for configuring OAuth2 for the Delphix Engine are available in the Delphix CLI (as the sysadmin user, under `service; oauth2`) as well as via the API endpoint `/resources/json/delphix/service/oauth2`.

234 http://dct.delphix.com/docs

Option	Description	Default (if applicable)
`audience`	Specifies the expected value of the audience claim (`aud`) of JWTs indicating that the tokens are intended for this particular Delphix engine.	`api://delphix`
`enabled`	Specifies whether the OAuth2 feature should be enabled.	`false`
`issuerURI`	(**Required**) Specifies the base location or identifier of the authorization server (IdP) which the Delphix Engine will use to validate incoming JWTs.	
`jwkSetURI`	Specifies the URI used to retrieve the JSON Web Key (JWK) set, if supported by the authorization server (IdP).	
`tokenSkewTime`	Specifies the maximum time difference (in seconds) allowed between the validity period of a JWT and the engine's current time.	`60`
`userIdClaim`	Specifies which claim in a JWT should be used by the Delphix Engine (in conjunction with the `userMatchingFieldType` setting) to associate a token with a user configured on the Delphix Engine. The default (`sub`) corresponds to the subject claim of the token.	`sub`
`userMatchingFieldType`	Specifies which property of a Delphix Engine user will be used to match with the claim (specified in `userIdClaim`) of a JWT. The Delphix Engine can be configured to match users based on a user's `name` , `emailAddress` , or `principal` properties.	`PRINCIPAL`

These options may also be set in the Delphix Setup application. Please refer to the following procedures for configuring OAuth2 parameters for new or existing Delphix Engines.



6.3.2.5.2.1 New engine configuration

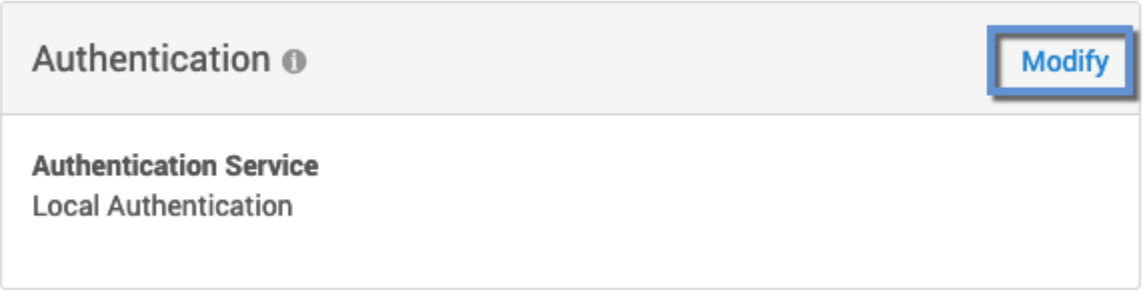
Follow this procedure to configure OAuth2 on a new Delphix Engine.

1. Connect to the Delphix Engine at `http://>>`. The **Delphix Setup** application will launch when connecting to the server.
2. Enter the **sysadmin** login credentials; this account has a default username of **sysadmin** and password of **sysadmin**.
3. In the Authentication step of the Delphix Setup wizard, check the **Use OAuth2 access tokens** box and configure the OAuth2 settings as desired. (Refer to the table above for more information on these settings.)
4. Complete the remaining setup steps as usual.

6.3.2.5.2.2 Existing engine configuration

Follow this procedure to configure OAuth2 on an existing Delphix Engine.

1. Connect to the Delphix Engine at `http://>>`. The **Delphix Setup** application will launch when connecting to the server.
2. Enter the **sysadmin** login credentials.
3. In the **Authentication** tile, select **Modify**.



4. In the Authentication dialog, check the **Use OAuth2 access tokens** box and configure the OAuth2 settings as desired. (Refer to the table above for more information on these settings.)

6.3.2.5.3 Example of API access using OAuth2 token

1. Obtain a JWT from the Authorization Server (IdP). (The details for this process will vary depending on the IdP vendor.) For the purposes of this example, the contents of the token are stored in the environment variable `t`.
2. Access the oauth2-login API endpoint of the Delphix Engine, providing the OAuth2 token. In this example the session information is stored in the file `cookies.txt` in the working directory. For Virtualization Engines the oauth2-login API endpoint is `/virtualization/api/oauth2-login`. For Masking Engines the oauth2-login API endpoint is `/masking/api/oauth2-login`.

   Virtualization API Endpoint Example

   ```
   h=<engine address>; curl -i -X POST $h/virtualization/api/oauth2-login -H
   "Authorization: Bearer $t" -b cookies.txt -c cookies.txt -H 'Content-Type:
   application/json' -d '{"type": "APISession", "version": {"type":
   "APIVersion","major": 1,"minor": 11,"micro": 11}}'
   ```

   Masking API Endpoint Example

   ```
   h=<engine address>; curl -i -X POST $h/masking/api/oauth2-login -H
   "Authorization: Bearer $t" -H 'Content-Type: application/json' -b cookies.txt
   -c cookies.txt
   ```

3. Refer to the saved cookies.txt file in subsequent `curl` invocations. The example below can be used to list the users configured on the Delphix Engine.

   Virtualization API Example

   ```
   curl -X GET -b cookies.txt -c cookies.txt -H 'Content-Type: application/json'
    $h/resources/json/delphix/user
   ```

Masking API Example

```
# Store authorization code returned by /masking/api/oauth2-login in $m
curl -i -X GET -b cookies.txt -c cookies.txt -H 'Content-Type: application/
json' -H "Authorization: $m" $h/masking/api/v5.1.11/users
```

6.3.2.5.4 CLI access using OAuth2 token

When OAuth2 is enabled, CLI logins will prompt the user to supply either an OAuth2 access token or a password to authenticate:

CLI Access Prompt with OAuth2 Enabled

```
$ ssh myengine -l admin
Enter access token or press enter to provide a password:
```

To authenticate using an OAuth2 token, paste its contents when this prompt is shown. As with API authentication, the OAuth2 token must be current (not expired) and must contain a claim that can be associated with a valid Delphix Engine user, based on the **userIdClaim** and **userMatchingFieldType** values set on the Delphix Engine.

To bypass OAuth2 authentication and use password authentication, press **Enter** when this prompt is shown and a conventional password prompt will be displayed.

6.3.2.5.5 User matching policy

If the `userIdClaim` component of a JWT matches more than one Delphix Engine user (for example, if `userMatchingFieldType` is set to `EMAIL_ADDRESS`, and the same email address is associated with multiple Delphix Engine users), the oldest user account (by time of creation) will be authenticated.

> Suggestion
> To ensure all users can be authenticated using OAuth2, make sure that the property specified in `userMatchingFieldType` is populated and unique for all Delphix Engine users.

6.3.2.5.6 HTTPS Proxy Configuration

If the OAuth2 identity provider cannot be directly reached by a Delphix Engine, a HTTPS proxy may be used. The sysadmin user can create or modify HTTPS proxy settings via the CLI/API endpoint `service; proxy`, or in the **Delphix Setup** application by selecting to modify the **Outbound Connectivity** tile, checking the **Configure web proxy** box, and entering the host, port, and (optionally) username and password of the proxy server.

6.4 Network and DNS management

You may want to manage and configure certain network services, such as DNS, for Delphix. Here we specify general network and connectivity requirements, as well as detail how you can test network performance. General Network and Connectivity Requirements.

This section covers the following topics:

- General network and connectivity requirements (see page 717)
- Network performance configuration options (see page 719)
- Determining the Delphix server ID and host name (see page 733)
- Configuring multiple DNS domain names in DNS search list (see page 736)
- How to change the IP address of the Delphix engine (see page 739)
- How to change the hostname of the Delphix engine (see page 741)
- How to change the DNS server of the Delphix engine (see page 743)
- Configuring a second network interface (see page 744)

6.4.1 General network and connectivity requirements

6.4.1.1 Overview

This topic covers the general network and connectivity requirements for the Delphix Engine, including connection requirements, port allocation, and firewall and Intrusion Detection System (IDS) considerations. For platform-specific network and connectivity requirements, see the relevant topics under the **Requirements** section for each platform.

6.4.1.2 General outbound from the Delphix engine port allocation

Protocol	Port numbers	Use
TCP	25	Connection to a local SMTP server for sending email
TCP/UDP	53	Connections to local DNS servers
UDP	123	Connection to an NTP server
UDP	162	Sending SNMP TRAP messages to an SNMP Manager
TCP	443	HTTPS connections from the Delphix Engine to the Delphix Support upload server

Protocol	Port numbers	Use
TCP/UDP	636	Secure connections to an LDAP server
TCP	8415	Connections to a Delphix replication target. See Configuring Replication (see page 2381)
TCP	50001	Connections to source and target environments for network performance tests.

6.4.1.3 General inbound to the Delphix engine port allocation

Protocol	Port number	Use
TCP	22	SSH connections to the Delphix Engine
TCP	80	HTTP connections to the Delphix GUI
UDP	161	Messages from an SNMP Manager to the Delphix Engine
TCP	443	HTTPS connections to the Delphix Management Application
TCP	8415	Delphix Session Protocol connections from all DSP-based network services including Replication, SnapSync for Oracle, V2P, and the Delphix Connector.
TCP	50001	Connections from source and target environments for network performance tests via the Delphix CLI.

6.4.1.4 Firewalls and intrusion detection systems (IDS)

Production databases on source environments (for dSources) are often separated from the non-production environment by firewalls. Firewalls can add milliseconds to the latency between servers. Accordingly, for best performance, there should be no firewalls between the Delphix Engine and the virtual database (VDB) target environments. If the Delphix Engine is separated from a source environment by a firewall, the firewall must be configured to permit network connections between the Delphix Engine and the source environments for the application protocols (ports) listed above.

Intrusion detection systems (IDSs) should also be made permissive to the Delphix Engine deployment. IDSs should be made aware of the anticipated high volumes of data transfer between dSources and the Delphix Engine.

6.4.2 Network performance configuration options

This section covers the following topics:

- Optimal network architecture for the Delphix engine (see page 719)
- Network operations using the Delphix session protocol (see page 722)
- Network performance test tool interface (see page 725)
- Working with dataset performance (see page 730)
- Network performance expectations and troubleshooting (see page 731)

6.4.2.1 Optimal network architecture for the Delphix engine

This topic describes basic network performance considerations for the Delphix Engine.

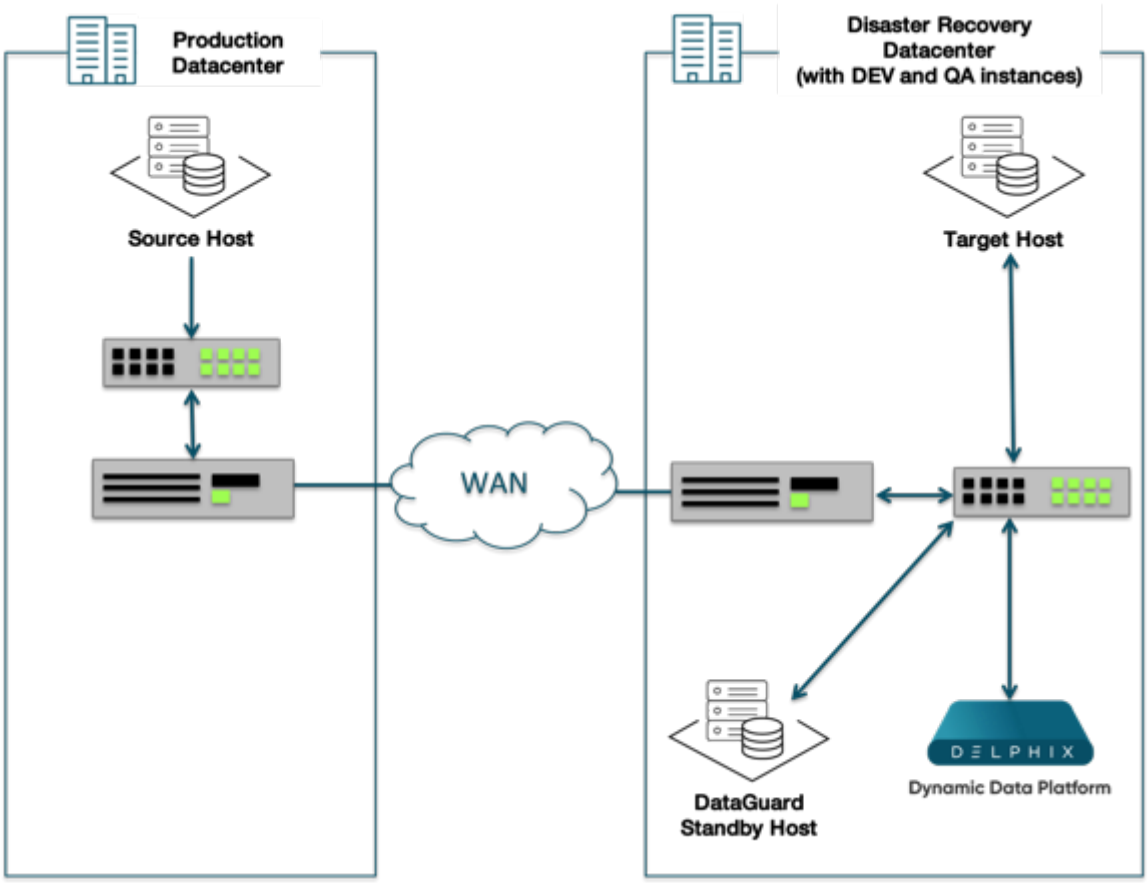
6.4.2.1.1 Network architecture and latency

All VDB I/O operations are serviced over the network. Delphix uses NFS as the primary transport for Oracle VDBs, and iSCSI for MS SQL VDBs. The network architecture, latency, and capacity between the Delphix Engine and the target environment are key network components for improving the performance of a Delphix deployment. The latency between the Delphix Engine and the source environment is not relevant for the best performance of VDBs.

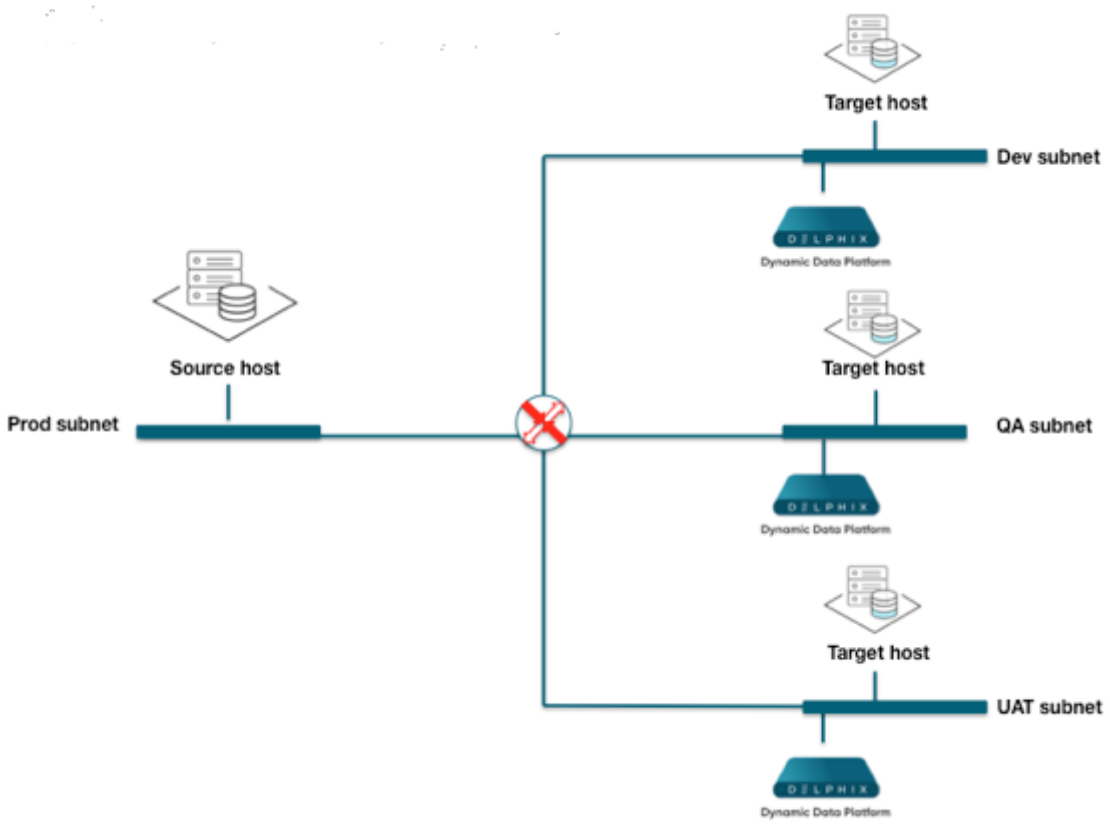
For optimal performance of VDBs, round-trip latency between the Delphix Engine and the target environment should be kept under 1 millisecond, and preferably in the range of 300 microseconds. If network latency exceeds 500 microseconds, the VDBs will not perform as well as a database connected to physical storage.

Latency can be introduced by having to route the network packets across multiple networks, or by the presence of routers, switches, and firewalls between the Delphix Engine and the target environment. Best practices to reduce network latency include:

- Keep the Delphix Engine on the same subnet as the target environment
- Reduce the number of hops between the Delphix Engine and the target environment
    - Reduce the number of switches in the network. Each switch can add 50 - 100 microseconds of latency to the network.
    - Reduce the number of routers in the network. Each router can add 500 - 1000 microseconds of latency in a network, and the round trip for an I/O operation could increase by as much as 1 - 2 milliseconds.
- There should be no firewalls between the Delphix Engine and the target environment.
- When linking the Delphix Engine to a source database across a WAN, consider the time needed for the initial link and load. It may be necessary to schedule the load operation as multiple steps across multiple days.



A common WAN deployment architecture



Deployment of the Delphix engine on Separate sub-nets

6.4.2.1.2 Network throughput and bandwidth

Network throughput measures the rate at which data can be sent continuously between two servers on a network. Network throughput is affected by network latency, but the dominant factor affecting throughput is the bandwidth of the network. As a point of comparison, consider the bandwidth available for three types of Ethernet networks:

Ethernet type	Network Bandwidth
100Mb Ethernet (100Base-T)	~=10MB/sec
Gigabit Ethernet (GbE)	~=100MB/sec
10 Gigabit Ethernet (10GbE)	~=1GB/sec

Low network throughput can impact the Delphix Engine in a number of ways:

- Increasing the amount of time it takes to perform a SnapSync operation, both for initial load and subsequent regular snapshots

- Managing LogSync operations in a high change environment
- Poor VDB performance when an application is performing large sequential I/O operations, such as sequential table scans for reporting or business intelligence, or RMAN backups of the VDB.

Delphix Engine throughput must exceed the sum of the peak I/O loads of all VDBs. Delphix incorporates an I/O-Collector toolkit to collect I/O data from each production source database and pre-production server.

Best practices to improve network throughput include:

- Use 10 Gigabit Ethernet (10GbE)
- Use a dedicated storage network

If you are concerned about your network throughput, you can test it with the built-in CLI tool for network testing.

6.4.2.2 Network operations using the Delphix session protocol

This topic describes how the Delphix Engine uses the Delphix Service Protocol (DSP) for network operations, and how this affects features such as replication, V2P, and SnapSync.

6.4.2.2.1 Overview

Delphix Session Protocol, or DSP, is a communication protocol that operates at the session and presentation layer in the Open Systems Interconnection (OSI) model.

Application Layer	application specific logic
Presentation Layer	data encoding, digest, compression, encryption
Session Layer	connection management, error recovery, security, remote operation
Transport Layer	end-to-end connection, message segmentation, sequencing, reliability, flow control
Network Layer	packet fragmentation, routing, logical addressing
Data Link Layer	physical addressing
Physical Layer	media, signal, binary transmission

DSP supports the request-reply pattern for communication between two networked peers. In contrast to the traditional remote procedure call (RPC) models, which focus exclusively on low-level details such as data encoding and wireframing, DSP implements a generic session layer that supports a number of advanced functionalities desired for network communication, including:

- Full-duplex remote operation execution and end-to-end cancellation support
- Advanced connectivity model with connection trunking and ordered delivery

- Fault resilience with automatic connection and session recovery, exactly-once semantics, and optional data digest
- High performance with concurrent execution, session flow control, optional data compression, and bandwidth throttling
- Built-in security support with pluggable SASL authentication mechanisms and optional TLS encryption
- Asynchronous model for session management and remote operation

Most of the features above are essential to the proper operation of a distributed application and yet non-trivial to implement. By offering them in the framework, we can significantly simplify the development of enterprise quality distributed applications.

DSP is officially registered with the Internet Assigned Numbers Authority[235] under the service name dlpx-sp and port number 8415.

Currently, DSP supports Java language binding and provides a java based service framework for distributed applications.

6.4.2.2.2 Key concepts

The foundation of DSP is built on top of a few key abstractions, namely, **exchange**, **task**, **nexus**, and **service**. For an overview of how DSP works and the features it provides, let's start with these abstractions.

An **exchange** refers to an application-defined protocol data unit which may be a request or a response. DSP supports the request-response pattern for communication. For each request sent, there is a corresponding response that describes the result of the execution. An application protocol is made up of a set of exchanges.

A **nexus** (a.k.a., session) refers to a logical conduit between the client and server application. In contrast, a transport connection (a.k.a., connection) refers to a "physical" link. A nexus has a separate naming scheme from the connection, which allows it to be uniquely and persistently identified independently of the physical infrastructure. A nexus has a different lifecycle than the connection. It is first established over a leading connection. After it comes into existence, new connections may be added and existing ones removed. It must have at least one connection to remain operational but may live on even after all connections are lost. Nexus lifecycle management actions, such as create, recover, and destroy, are always initiated by the client with the server remaining passive.

A nexus has dual channels, namely, the fore channel and the backchannel. The fore channel is used for requests initiated from the client to the server, and the backchannel from the server to the client. From a request execution perspective, the nexus is full-duplex and the channels are functionally identical, modulo the operational parameters that may be negotiated independently for each channel. A channel supports a number of features for request processing, such as ordered delivery, concurrent execution, remote cancellation, exactly-once semantics, and throughput throttling.

A **service** refers to a contract that consists of all exchanges (both the requests and the corresponding responses) defined in an application protocol. Given the full-duplex nature of request execution in DSP, part of the service is fulfilled by the server and the remaining by the client, where the client and server are from the nexus management perspective.

A **task** implements a workflow that typically involves multiple requests executed in either or both directions over the nexus. A task is a self-contained building block, available in the form of a sharable module including both the protocol exchanges and implementation, that can be easily integrated into other application

235 http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.txt

protocols. A library of tasks may significantly simplify distributed application development by making it more of an assembly experience.

The following is a diagram that illustrates the key abstractions and how they are related to each other.

6.4.2.2.3 Security

As a network protocol, DSP is designed with security in mind from the onset. It supports strong authentication as well as data encryption. It follows a session-based authentication model which requires each connection to authenticate before it is allowed to join the session. Authentication is performed using the Simple Authentication and Security Layer (SASL) framework, a standard-based pluggable security framework. The currently supported SASL mechanisms include DIGEST-MD5, PLAIN with TLS, CRAM, and ANONYMOUS. Optionally, TLS encryption may be negotiated between the client and the server for data privacy.

6.4.2.2.4 Performance

DSP offers a number of features to enable the support for high-performance network applications. For example, it allows multiple requests to be exchanged in both directions simultaneously, which provides effective pipelining of data transfer to minimize the impact of network latency while ensuring the total ordering at the same time. It supports trunking that can effectively aggregate the throughput across multiple connections, which is crucial for a long fat network (LFN) and 10GigE. It also provides optional compression support which boosts performance over a bandwidth-limited network. We have observed, through both internal benchmarking and in customer environments, DSP-based applications delivering multi GigE in an ideal environment and getting a performance boost of as much as x10 in bandwidth-limited settings.

6.4.2.2.5 Resiliency

DSP automatically recovers from transient connection loss without any application involvement. It may also detect random data corruption on the wire and automatically recovers from it. In both cases, outstanding requests are retried once the fault condition is resolved.

DSP offers control over a remotely executing request. Once a request is initiated, the application may cancel it at any time before completion. In the rare event of a session loss, a new session creation request will be held until the old session has been reinstated. It ensures that we never leave any unknown or unwanted activities on the remote side and provides better predictability and consistency guarantees over an otherwise unreliable network.

6.4.2.2.6 Diagnosability

Application exceptions encountered during remote execution of a request are communicated back to the initiator through DSP. A standard Java API is used to facilitate the handling of remote exceptions that is in many ways identical to local ones.

DSP provides detailed information and statistics at the session level. The information may be used to examine the state of the session as well as diagnose performance problems. It is currently exposed via an internal support tool called the JMX tool.

6.4.2.2.7 Supported applications

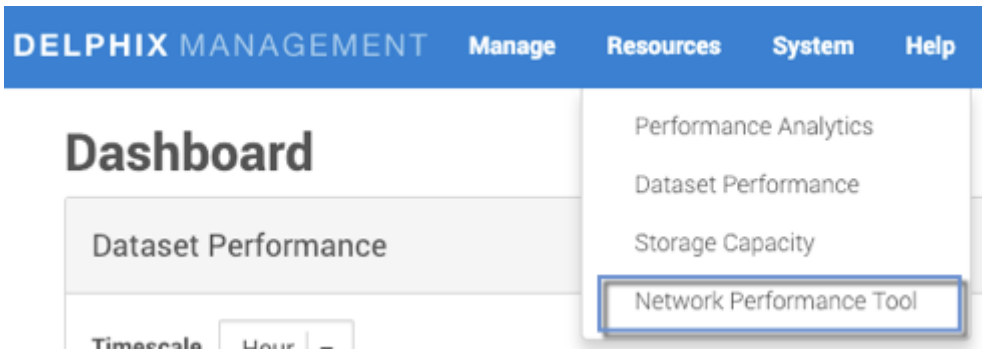
Replication is the first feature to take advantage of DSP. It has been rebuilt on top of DSP and shipping in the field since 3.1. In the latest release, a number of host-based applications, such as SnapSync, V2P, and Delphix connector, use DSP as well.

6.4.2.3 Network performance test tool interface

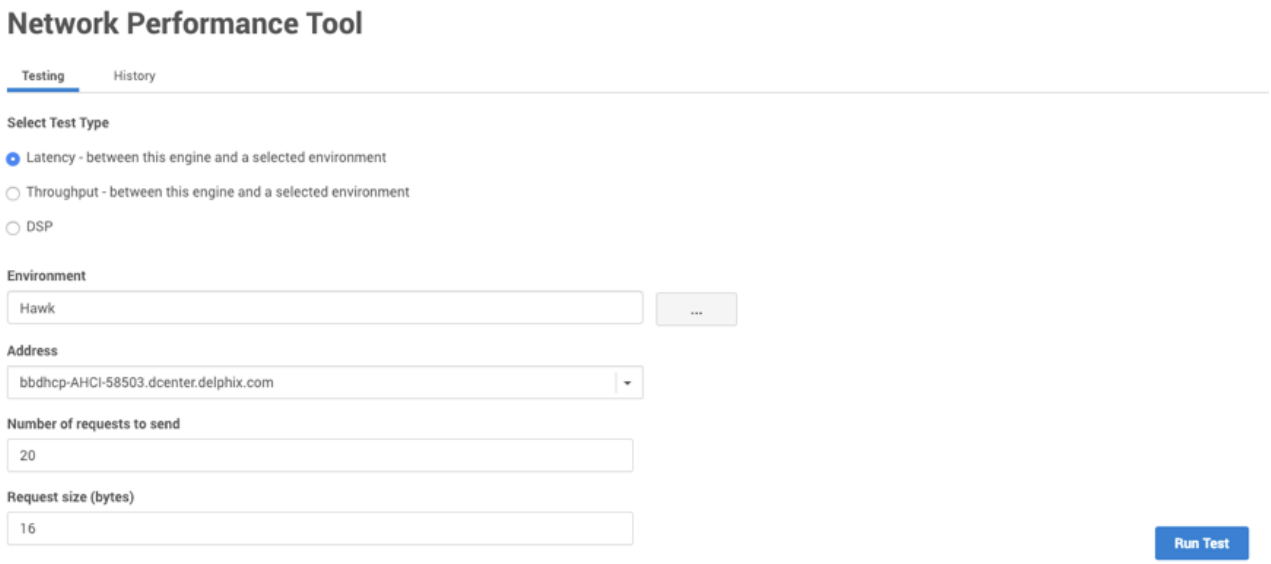
6.4.2.3.1 Accessing the network performance test tool

To access the Network Performance Test Tool:

1. In the top navigation bar, click **Resources**.
2. Select **Network Performance Tool**.



The **Network Performance Tool** page will appear. There are two tabs: **Testing** and **History**.



6.4.2.3.2 The testing tab

On the **Testing** tab, you can select one of three test types:

Test type	Parameters required
Latency	• Environment – If you selected an environment in the **Datasets** panel, this field will auto-populate. See below for instructions on changing the selected environment. • Number of requests to send • Request size (bytes)
Throughput	• Environment • Duration – in seconds • Number of connections • Direction – Select either **Transmit** or **Receive** • Block size (bytes)
Delphix session protocol	• Target Engine • Username • Password • Duration – in seconds • Block size (bytes) • Number of connections • Queue depth • Direction – Select either **Transmit** or **Receive** • Traffic Options

6.4.2.3.2.1 CLI

This command provides both remoteHost and remoteAddress:

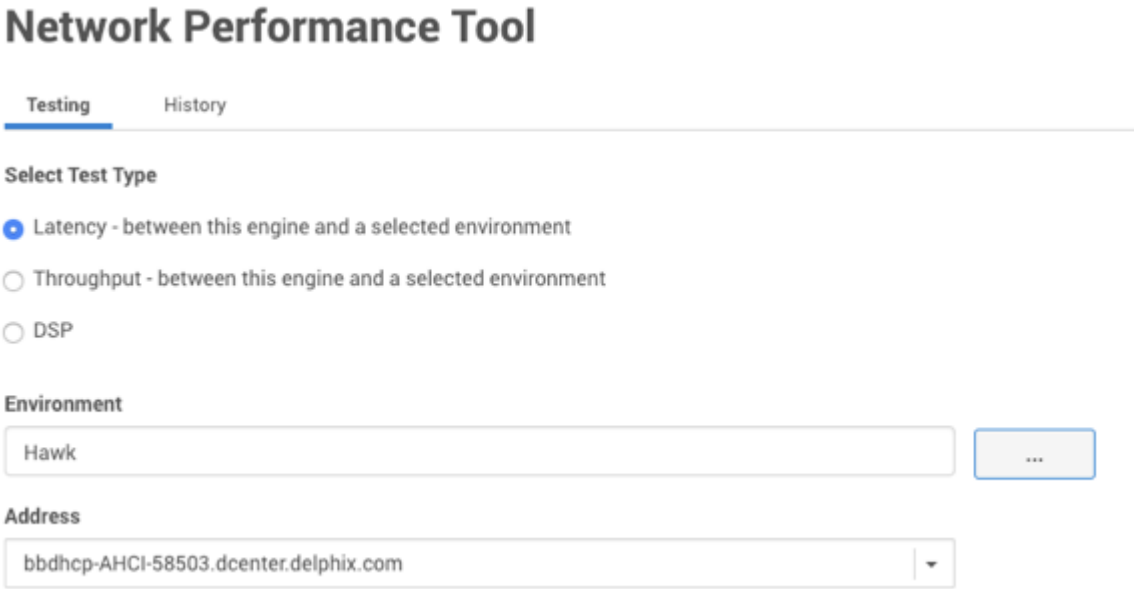
```
env.name network test latency create *> ls
Properties
    type: NetworkLatencyTestParameters
    remoteAddress: (unset)
    remoteHost: (unset)
    requestCount: 20
    requestSize: 8B
```

This is an important distinction, as the latency test has no requirement for an address to be connected to a known Environment, since it is a simple ping test. The Throughput tests are dependent, but Latency tests should use the correct values for remoteAddress and remoteHost.

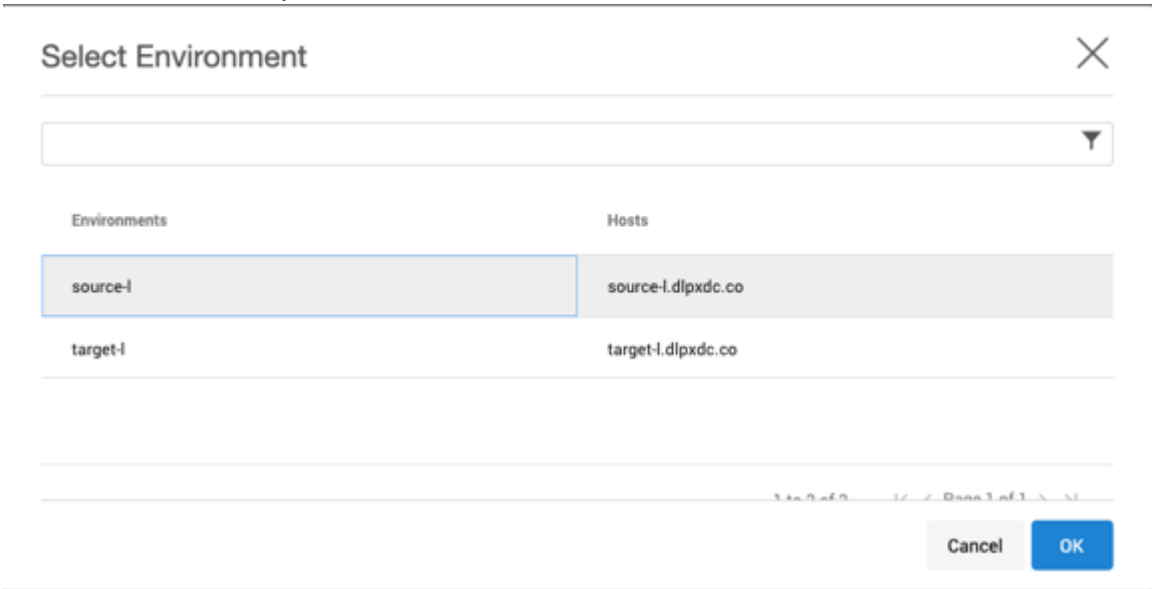
6.4.2.3.2.2 Selecting an environment

For either a Latency or Throughput test, you must select an environment. If you selected an environment in the **Datasets** panel, this field will auto-populate. If you need to select a different environment:

1. Next to the **Environment** field, click the button with three dots.



2. Select the environment you want to use in the test.



3. Click **OK**.

6.4.2.3.2.3 Running and canceling a network performance test

To run a test:

1. Enter all the required parameters.
2. Click **Run Test**.
3. To cancel the test click **Cancel**.

6.4.2.3.3 The history tab

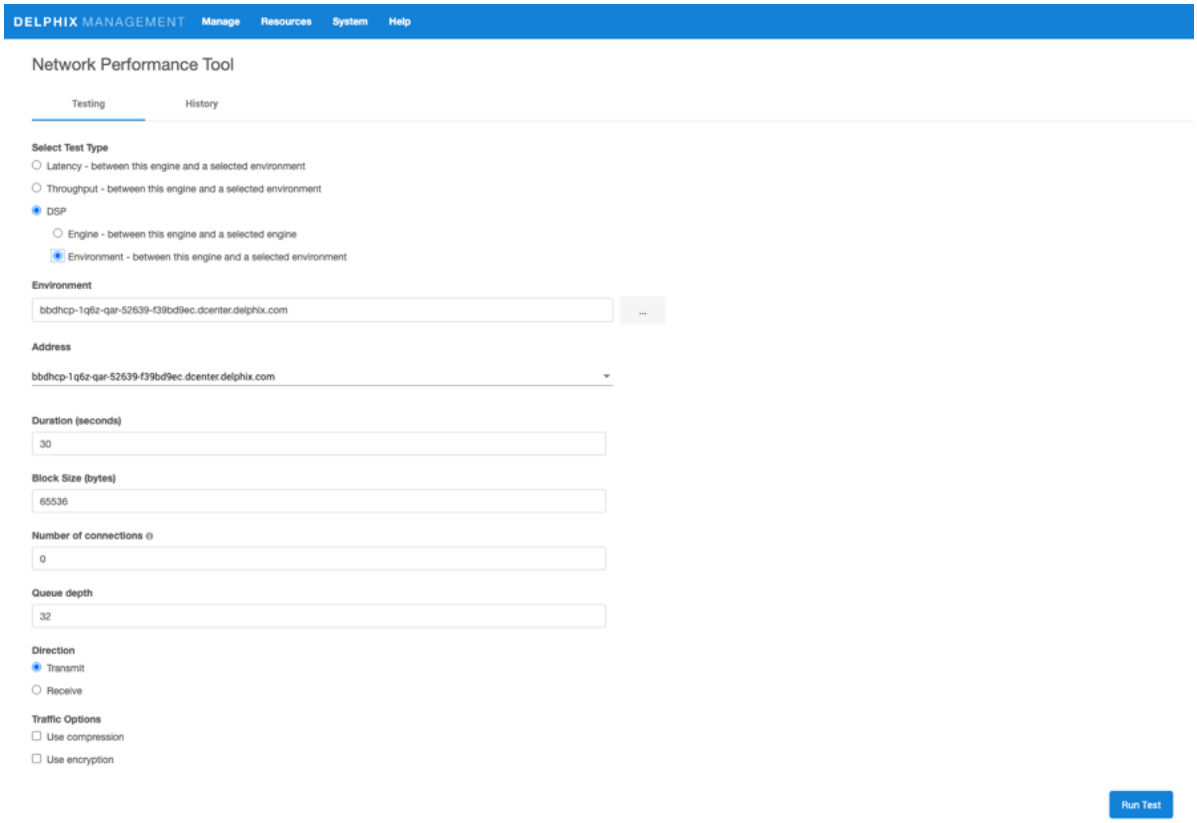
In the **History** tab, you can view the results of all previous tests you have run.

1. Select the radio button for the type of test for which you want to see the results:
    a. Latency, or
    b. Throughput, or
    c. Delphix Session Protocol
2. Optional: sort the tests by clicking one of the column headings.
3. Click the particular test for which you want to see the results.
   The details of that test will appear.
4. Click **OK** to return to the **History** tab.

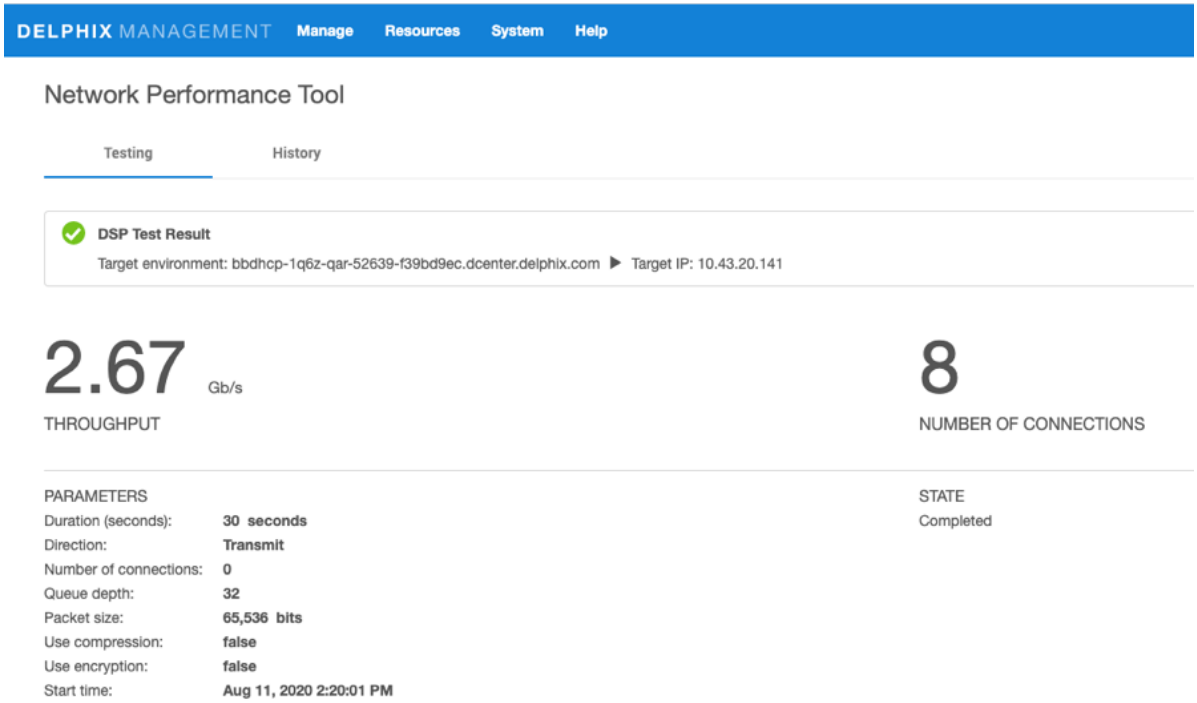
6.4.2.3.4 Delphix session protocol test from primary engine to replication engine

Delphix uses the DSP protocol to communicate between primary and replication engines.

1. Login to Delphix Management application using an Engine administrator account.
2. Click the **Resource** menu and select **Network Performance Tool.**
3. Select the **Delphix Session Protocol**option.
    a. Select Engine option to run the DSP test between a primary engine and a replication engine.
    b. Provide hostname or IP address of the replication engine.
    c. Provide admin user credentials for the replication engine. The user name must not include @DOMAIN or other qualifiers.
    d. Enter a **Duration** in seconds (30 default).
    e. Enter **Block Size** in bytes.
    f. Enter the **Number of Connections** (optional).
    g. Provide **Queue Depth**.
    h. Select Direction - **Transmit** or **Receive.**
    i. **Enter Traffic Options by selecting either Use compression or Use encryption.**
    j. Click the **Run Test** button.



4. View the test results.

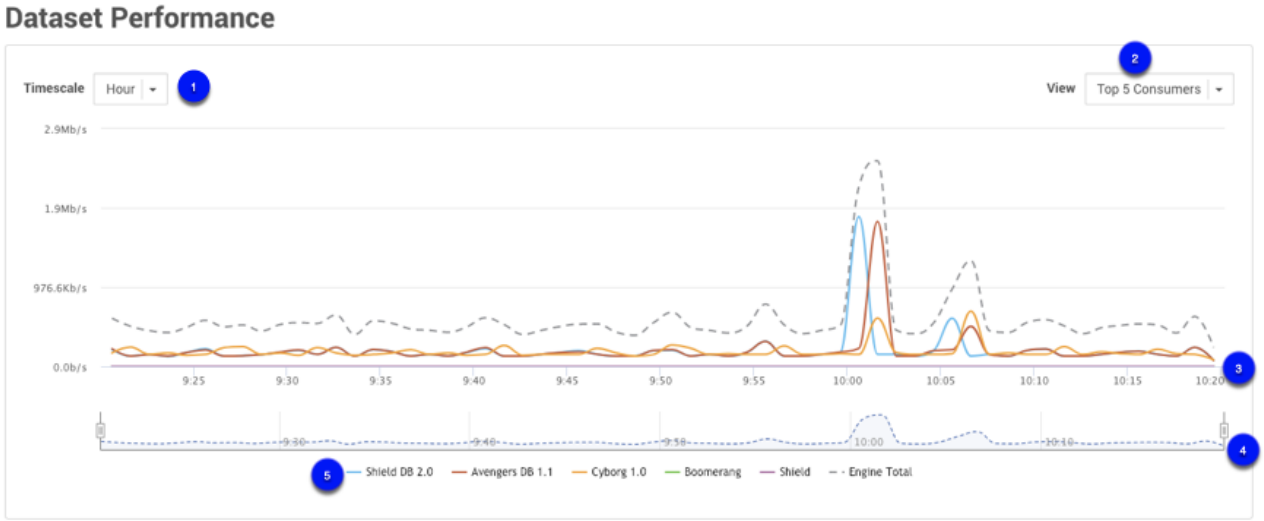


6.4.2.4 Working with dataset performance

6.4.2.4.1 Accessing the dataset performance graph

1. Log into the Delphix Management application.
2. In the **Resources** menu, select **Dataset Performance**.
3. Use the controls described below to view statistics and their related graphs.

6.4.2.4.2 General graph display and controls



Dataset	Type	Environment	Host	Current	Hour	6 Hr	Day	Percentile
Engine Total	-	-	-	573.00 Kb/s	573.00 Kb/s	638.00 Kb/s	874.00 Kb/s	-
Boomerang	dSource	Hawk	bbdhcp-AHCI-585...	0.00 b/s	0.00 b/s	0.00 b/s	0.00 b/s	20
Shield DB 2.0	VDB	Starfire	bbdhcp-tgt-AHCI-...	76.00 Kb/s	195.00 Kb/s	196.00 Kb/s	323.00 Kb/s	70
Cyborg 1.0	VDB	Hawk	bbdhcp-AHCI-585...	83.00 Kb/s	183.00 Kb/s	212.00 Kb/s	214.00 Kb/s	50
Avengers DB 1.1	VDB	Starfire	bbdhcp-tgt-AHCI-...	72.00 Kb/s	195.00 Kb/s	230.00 Kb/s	337.00 Kb/s	90
Shield	dSource	Hawk	bbdhcp-AHCI-585...	0.00 b/s	0.00 b/s	0.00 b/s	0.00 b/s	20

1 **Timescale:** Select from hour, 6 hours, day

2 **View:** View data by top 5 consumers or by the 75% percentile

3 **Dateline:** Displays timestamps of data points in the graph.

4 **Timeline selector:** Specifies the start and end time for the currently displayed data. The range displayed is controlled by selecting the slider. Drag the slider to view statistics for the specific time.

5 **Graph legend:** If more than one set of information is presented on the graph, Graph Legend displays a description and color for each set.

6 **Dataset table:** Displays the information for each Dataset in a table format. Selecting a Dataset link takes you to the Dataset Status page.

7 **Export**: Exports the displayed table data.

6.4.2.5 Network performance expectations and troubleshooting

6.4.2.5.1 Overview

Once you have run your network performance tests (see page 2740) to each source and target environment, you should confirm that they meet expectations. Corporate networks commonly leverage 10Gbps "line speeds" between servers. Although networks are shared environments, switching infrastructure is typically used to isolate traffic between different hosts, with the goal of allowing each host to reach its potential. While many environments perform within the 70-90% range of line speed, in well-maintained environments we can see 90%+ line speed.

In some circumstances, you can exceed typical expectations by implementing more best practice recommendations – for example, putting two VMs on the same hypervisor (e.g. ESX) host, in the same chassis or blade enclosure, or when teamed (bonded) NIC cards are used for extra bandwidth. Furthermore, when Jumbo Frames are implemented correctly, they provide a substantial improvement.

If your network is not meeting expectations, you will almost always need help from someone whose role is focused on networking to help arrive at a root cause. You may also need to obtain temporary access to other VMs and/or physical hosts to isolate some issue(s). Additionally, you may need the help of Delphix Support or Professional Services to perform some tests from the Delphix Engine to systems that are not already connected environments within the Delphix product. (The CLI test will only work for environments connected to the Delphix Engine).

With assistance or not, you can do a number of things to narrow down the potential causes of poor performance between two systems by a process of elimination. Below are some high-level steps to consider. Keep in mind that network throughput will always represent the least performant component, so many of the steps below are intended to help isolate which component may be performing poorly.

6.4.2.5.2 Troubleshooting and information gathering questions in rough order of priority - record and share your answers

1. Have all source/target tuning settings been applied?
    a. Is AIX in scope? LSO / LRO can have a significant impact, as can Jumbo Frames (see below)
2. What is the link speed on the hosts in question? Is NIC teaming / bonding / LACP in use?
    a. Linux: ethtool <device>
    b. Solaris: dladm show-phys
    c. Windows: wmic NIC where "NetEnabled='true'" get "Name,Speed"
3. What are the test results with greater or fewer connections in parallel?
4. Can we test throughput to alternate servers? (See below)
5. What is the overall latency? What is the latency to each (OSI layer 3) hop? Is there one hop that consistency has a higher cost? (check the latency with ping and hops with traceroute)
6. How many devices (OSI layer 2) are in the path? Your network team will need to help you identify these devices.

   a. Note: Only Layer 3 devices will show up when reviewing a traceroute however each layer two devices can impact traffic and each needs to be configured when implementing jumbo frames
   b. Example devices in path to physical server: 1. virtual NIC -> 2. Virtual switch (ESX) -> 3. Chassis NIC -> 4. Rack switch -> 5. Core Switch -> 6. Rack Switch -> 7. Physical NIC
   c. Example devices in path to virtual server: 1. virtual NIC -> 2. Virtual switch (ESX) -> 3. Chassis NIC -> 4. Rack switch -> 5. Chassis NIC -> 6. Virtual Switch (ESX) -> 7. Virtual NIC
7. What is the average network utilization on each hop? Is there congestion on any hop? (Network team will need to review)
8. Is QoS / VirtualConnect / 802.1p enabled? At what threshold will it engage? (Network team will need to review)
9. Is there a firewall or any deep packet inspection in the route? (Network team will need to review)
10. Are Jumbo Frames enabled on any or all hops? E.g. Delphix Engine, Virtual NIC, Virtual switch, and all hops down to the destination. (Network team will need to review)
   a. We have seen Delphix installations often benefit 10-20% from Jumbo frames, but certain platforms (such as AIX) can benefit much more dramatically
   b. Note that JF enablement on two hosts without confirming all the network pieces are properly enabled will result in VERY poor performance
   c. Test Jumbo Frames via with “Do Not Fragment” flag from the remote host to the Delphix Engine.
   d. Note that typical MTU Jumbo Frame setting is 9000 bytes, although some vendors recommend a little above or below this.
   e. The test below is at 8000, but you can test larger from there. Our goal is primarily to ensure that a number substantially larger than 1500 and somewhat close to the 9000 "de facto" standard is working.
   f. Whenever two hosts connect, they perform a handshake called Path MTU negotiation, where they agree on the highest MTU they both support. This is how we avoid impact when communicating to hosts with differing MTUs.
   g. `Linux$ ping` -M `do` -s 8000 [Delphix_Engine_IP]
   h. `Windows> ping -f -l 8000 [Delphix_Engine_IP]`
   i. `Solaris v10-# traceroute -F [Delphix_Engine_IP] 8000 (` "Do Not Fragment" not supported by ping on Solaris until v11)
   j. `Solaris v11+# ping -s -D [Delphix_Engine_IP] 8972`
11. Depending on the results above, a dedicated network or VLAN may help. Consider if that is an option for you. (Your network team will need to review)

6.4.2.5.2.1 Testing throughput testing to alternate servers

This will help isolate where a problem may be.

1. Delphix to Server A – already known

2. Delphix to Server B – physical; helps us see if there is a problem with the original server NIC or physical network settings
3. Server B to Server C – physical; helps us see if there is a problem with the Delphix server NIC or physical network settings
4. Delphix to Server D – virtual; helps us see if there is a problem with the virtual network or Delphix settings
5. Server D to Server E – virtual; same host; helps us see if there is a problem with the virtual network on the host
6. Server D to Server F – virtual; different host; helps us see if there is a problem with the virtual network

6.4.2.5.3 Conclusion

If you need further help, please contact Delphix Support or Professional Services to assist in getting the best performance possible from your environment.

6.4.3 Determining the Delphix server ID and host name

On occasion, it may be necessary to locate the **Delphix Server ID** and **Hostname**.

The Delphix Engine ID and Delphix Server ID are synonymous. The GUI currently uses "Server," and that is the terminology that will be used in this document.

The **Delphix Server ID** uniquely identifies each Delphix Engine. It is a 36-character hexadecimal string of the form **xxxxxxxx-xxxx-xxxx-xxxxxxxxxxxx**. You can view the Delphix Server ID in the Server Setup application, the Delphix Management application, or by using the Command Line Interface (CLI) method.

The **Hostname** is a name you assign. It typically matches the IP (DNS) name assigned to the Delphix Engine. The hostname can only be viewed by using the System Setup application or the CLI method.

6.4.3.1 Server setup application method

Login to the Delphix Setup application with sysadmin-level credentials:

1. Access the Delphix Engine through the URL: `http://<Delphix Engine>/ServerSetup.html` where `<DelphixEngine>` is the DNS name or IP address of the Delphix Engine for which you wish to find the Delphix Server ID and hostname.
2. Enter a valid **Username**.
3. Enter a valid **Password**.
4. Click **Log In**.

On the **Dashboard** screen, there is a **System Summary** panel in the lower-left portion of the screen**.** The **Server ID** field displays the Delphix Server ID, in this example **564D39A8-5077-C9D0-9EFD-82E848EBDAB6.**

System Summary

Server ID	564D39A8-5077-C9D0-9EFD-82E848EBDAB6
Manufacturer	VMware
Model	Virtual_disk
Serial	6000c2907f0001dd8000c240feee5b8c
Processor	2 x 2.90GHz
Memory	7.3GB
Features	XPP, MDD
Default locale	en-US

The Delphix Engine Hostname is located on the same screen, in the **Network** panel. The **Delphix Engine Hostname** field displays the hostname information.

Network Modify

Network Interface
ens160

Interface Configured
Yes

Jumbo Frames Enabled
No

IP Address Type
DHCP

IP Address
10.43.3.98

Subnet Mask
255.255.0.0

Default Gateway
10.43.0.1

DNS Domain Name
delphix.com

DNS Servers
172.16.101.11, 172.16.105.2

Hostname
js532.dcenter

6.4.3.2 Delphix admin application

The Delphix Server Hostname is not available from this view but typically matches the IP (DNS) name assigned to the Delphix Engine.

Login to the Delphix Management application with delphix_admin level credentials:

1. Access the Delphix Engine through the URL: `http://<Delphix Engine>/Server.html` where `<DelphixEngine>` is the DNS name of the IP address of the Delphix Engine for which you wish to find the Delphix Server ID.
2. Enter a valid **Username**.
3. Enter a valid **Password**.
4. Click **Log In**.
5. Under **System Summary**, the **Server ID** field displays the Delphix Server ID.

6.4.3.3 CLI method

1. Use SSH to access your Delphix Engine: `ssh <userid>@<delphix_engine>` where < `userid>` is a user ID with either delphix_admin- or sysadmin-level credentials.
2. Enter a valid password if prompted for one.
3. Enter `system ls`.

   You will see an output similar to this example:

```
Properties
   type: SystemInfo
   apiVersion:
   type: APIVersion
      major: 1
      micro: 0
      minor: 5
   buildTimestamp: 2015-02-24T09:24:58.000Z
   buildTitle: Delphix Engine 4.2.0.1
   buildVersion:
      type: VersionInfo
      major: 4
      micro: 0
      minor: 2
      patch: 1
   configured: true
   currentLocale: en-US
   enabledFeatures: XPP
   hostname: delphix42.dcenter
   installationTime: 2015-02-24T19:53:32.000Z
```

```
    locales: en-US
    memorySize: 3.99GB
    platform: VMware with BIOS date 04/14/2014
    processors:
        0:
            type: CPUInfo
            cores: 1
            speed: 2.40GHz
        1:
            type: CPUInfo
            cores: 1
            speed: 2.40GHz
    productName: Delphix Engine
    productType: standard
    sshPublicKey: ssh-rsa
 AAAAB3NzaC1yc2EAAAABIwAAAQEAszNCFnfziuK8dBdv6DNB+LrhVP1wRWc/
vXVrxrDlgYQTrqvEx4BKgHDZ2hnbAmqq2xXHR5Ah6WDsEfo6u5B45JZc8qHpx8VZSZa053IdMK9LEg
oKPepmo7JV3kVY9oHK9PngLm9tFK+hN7AUHcGTt68IHq54GWYQNBtx0kgSR5HtkkFhVfX2amFsHIsq
1K96bgRkL0I5f3SjF4NnyElgBU9grGDajm9RXv+sz+Fn7h79AtFm0+W2Ymr5gQrdgh2vPyeFtG8G7r
xnQx3qiRBY6lNqepBhitXnMYSduGfW+fMJpV8TOOJ9ZLCfE7rjAgH7RxPybTfb4u70sm2krS8SgQ==
root@delphix
    storageTotal: 23.07GB
    storageUsed: 2.00GB
    uuid: 564d2f7c-b84f-8bd1-6f45-2060ac9b9a65
```

The **Delphix server ID** is shown as the `uuid` property at the bottom of the output, and the **Hostname** is displayed in the `hostname` property.

4. Enter `exit` to leave the command-line interface.

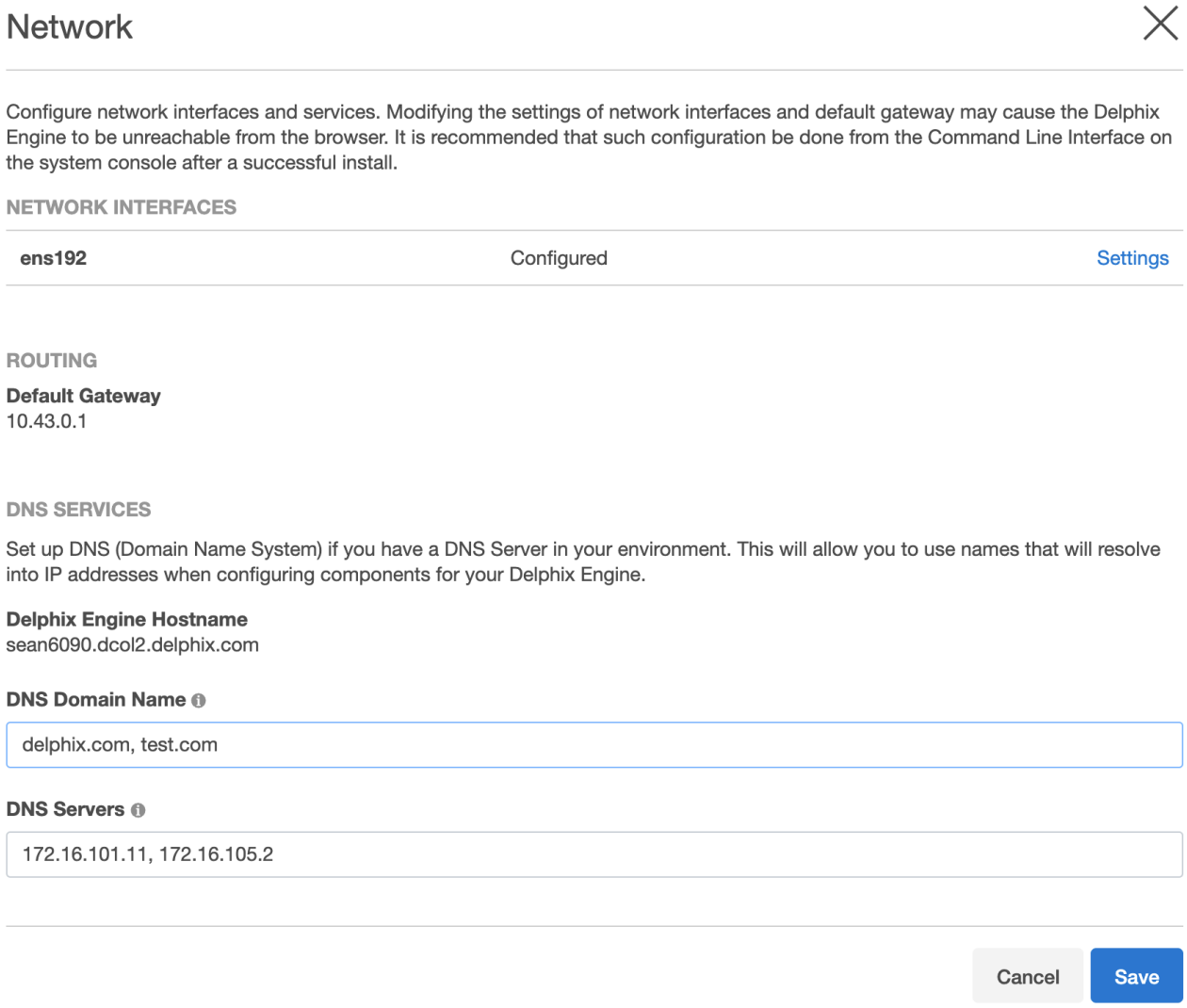
6.4.4 Configuring multiple DNS domain names in DNS search list

This topic describes the steps to configure multiple DNS domain names in the DNS search list.

6.4.4.1 Procedure

Perform the following steps to configure multiple DNS domain names in the DNS search list.

1. Launch the Delphix Engine Setup interface using the sysadmin credentials.
2. Navigate to the **Network** widget and click **Modify**.
3. Under **DNS SERVICES**, use the **DNS Domain Name** and **DNS Servers** boxes for adding multiple DNS domain name configurations.



Each domain name needs to be separated by a comma.

> If .local DNS domains are in use, then you must add these explicitly to the list of DNS domains configured in order for name resolution to be successful. Multiple .local subdomains can be added as desired (for example, dev.company.local), or "local" can simply be added to the DNS domain configuration to enable all .local domains to be successfully looked up in DNS. Multicast DNS is not currently supported by the Delphix Engine

In order to understand if there is more than one domain name in the search list, check for "DNS Suffix Search List" from the output of `ipconfig /all` in the Windows Server:

```
C:\Program Files\Delphix\DelphixConnector\connector>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : 10-43-13-231
   Primary Dns Suffix  . . . . . . . : ad.delphix.com
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : ad.delphix.com
                                       delphix.com
                                       dcenter.delphix.com
```

or check for "`search`" in `/etc/resolv.conf` on a Linux server:

```
[root@rhel62 ~]# cat /etc/resolv.conf search delphix.com nameserver 192.168.0.1
```

6.4.4.2 To update DNS using the CLI:

1. Log into the CLI as sysadmin and navigate to **service > dns**.

   ```
   ssh sysadmin@yourengine
   > service
   > dns
   ```

2. List the current DNS configuration and **update** to add new configurations.

   ```
   > ls
   > update
   > set domain=xxx.xxx, xxx.xxx
   ```

3. **Commit** the action and verify the new list.

   ```
   > commit
   > ls
   ```

   For example:

   ```
   delphix> /service dns
   delphix service dns> ls

   Properties
   ```

```
type: DNSConfig
domain: delphix.com
node: (unset)
servers: 192.168.0.1

Operations
update

delphix service dns> update
delphix service dns update *> set
domain=delphix.com,one.delphix.com,two.delphix.com
delphix service dns update *> commit
delphix service dns>
delphix service dns> ls

Properties
type: DNSConfig
domain: delphix.com,one.delphix.com,two.delphix.com
node: (unset)
servers: 192.168.0.1

Operations
update

delphix service dns>
```

6.4.5 How to change the IP address of the Delphix engine

6.4.5.1 Changing the IP address

1. Stop all running VDBs by clicking the **Stop** button on the VDB card.
2. Disable all dSources.
3. You can use either the command-line interface or the Delphix Setup application to change the IP address of the Delphix Engine.
   a. To use the command-line interface, follow the instructions described in Setting Up Network Access to the Delphix Engine (see page 569)
   b. To use the Delphix Setup application, go to **System > Server Setup** in the Delphix Management interface, or click **Server Setup** in the Delphix Engine login screen.
      i. In the **Network** panel, click **Modify**.
      ii. Under **DNS Services**, enter the new IP address.
      iii. Click **OK**.
4. Refresh all Environments by clicking the **Refresh** Symbol on the Environments screen.

5. Enable all dSources.
6. Start all VDBs by clicking the **Start** button on the VDB card.

6.4.5.2 Changing the IP address via CLI

1. Stop all running VDBs by clicking the **Stop** button on the VDB card.
2. Disable all dSources.
3. Log into the Delphix CLI using your *sysadmin* account. You can find instructions on how to do this in the Connecting to the CLI (see page 2522) article.

```
delphix> network
delphix network> interface
delphix network interface> ls
NAME
vmxnet3s0
delphix network interface> select vmxnet3s0
delphix network interface 'vmxnet3s0'> get
    type: NetworkInterface
    name: vmxnet3s0
    addresses:
        0:
            type: InterfaceAddress
            address: 10.1.2.3/24
            addressType: STATIC
            enableSSH: true
            state: OK
    dataNode: DATA_NODE-34
    device: vmxnet3s0
    macAddress: 0:c:29:32:96:a3
    mtu: 1500
    mtuRange: 60-9000
    reference: NETWORK_INTERFACE-vmxnet3s0-DATA_NODE-34
    state: OK
```

4. Run the update command and update the address to the new IP address for the Delphix Engine.

```
delphix network interface 'vmxnet3s0'> update
delphix network interface 'vmxnet3s0' update *> edit addresses.0
delphix network interface 'vmxnet3s0' update addresses.0 *> get
Properties
    type: InterfaceAddress
    address: 172.16.151.154/24
    addressType: STATIC
    enableSSH: true
```

```
delphix network interface 'vmxnet3s0' update addresses.0 *> set address=10.1.2.
4/24
delphix network interface 'vmxnet3s0' update addresses.0 *> get
    type: InterfaceAddress (*)
    address: 10.1.2.4/24 (*)
    addressType: STATIC (*)
    enableSSH: true (*)
```

5. Commit the operation.

```
delphix network interface 'vmxnet3s0' update addresses.0 *> commit
delphix network interface 'vmxnet3s0'> get
    type: NetworkInterface
    name: vmxnet3s0
    addresses:
        0:
            type: InterfaceAddress
            address: 10.1.2.4/24
            addressType: STATIC
            enableSSH: true
            state: OK
    dataNode: DATA_NODE-34
    device: vmxnet3s0
    macAddress: 0:c:29:32:96:a3
    mtu: 1500
    mtuRange: 60-9000
    reference: NETWORK_INTERFACE-vmxnet3s0-DATA_NODE-34
    state: OK
```

6. Re-enable the VDBs and dSources running from the engine.

6.4.6 How to change the hostname of the Delphix engine

6.4.6.1 Changing the hostname

Perform the following steps to change the hostname of a Delphix engine.

> Currently, it is only possible to change the hostname via the Command Line Interface (CLI).

1. Stop all running VDBs by clicking the **Stop** button on the VDB card.
2. Disable all dSources.

3. Log into the Delphix CLI using your ***sysadmin*** username and password. If you are using the same interface that you connected to the CLI on, it will interrupt the connection. Therefore, it is recommended to log in to the CLI on the console.

```
ssh sysadmin@yourdelphixengine
```

4. Run the following commands.

```
dlpx01> network
dlpx01 network> setup
dlpx01 network setup> ls
Properties
    defaultRoute: 192.168.0.1
    dhcp: false
    dnsDomain: plb.internal
    dnsServers: 192.168.0.111,198.142.152.164,198.142.152.165
    hostname: dlpx01
    primaryAddress: 192.168.0.109/24

Operations
update
dlpx01 network setup> update
dlpx01 network setup update *> set hostname=newhostname
dlpx01 network setup update *> commit
```

5. If DHCP is being used, the Delphix Engine will expect the Hostname to be provided by the DHCP server. As such, there will be no property 'hostname' to update. This process requires changing the IP addressing configuration from DHCP to static address configuration. To proceed you will need to take the following action (making sure that there is no conflict between your DHCP server and the changes you are implementing).

```
dlpx01 network setup> update
dlpx01 network setup update *> set dhcp=false
dlpx01 network setup update *> set hostname=newhostname
dlpx01 network setup update *> commit
newhostname network setup> update
newhostname network setup update *> set dhcp=true
newhostname network setup update *> commit
newhostname network setup> ls
Properties
    defaultRoute: 192.168.0.1
    dhcp: true
    dnsDomain: plb.internal
    dnsServers: 192.168.0.111,198.142.152.164,198.142.152.165
    hostname: newhostname
    primaryAddress: 192.168.0.109/24
```

6. Re-enable the VDBs and dSources running from the engine.

6.4.7 How to change the DNS server of the Delphix engine

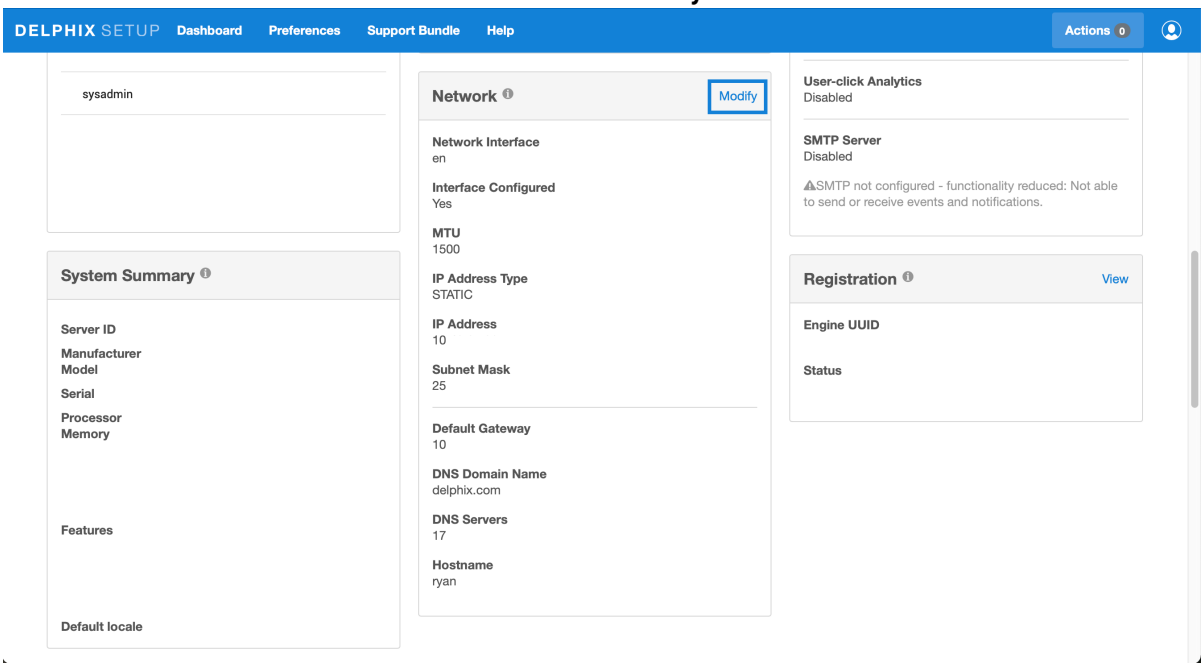
When DHCP is enabled on the Delphix Engine, the DNS server cannot be changed in the GUI. In this case, the DNS server can be changed in the command-line interface (CLI). If the IP address, net mask, or any other properties of the network interface itself are not changed, then VDBs do not need to shutdown. Login via Hypervisor Console for the Delphix Engine virtual machine is optional.

Use the commands below and enter the address of a server or a comma-separated list of addresses in replace of the example servers.

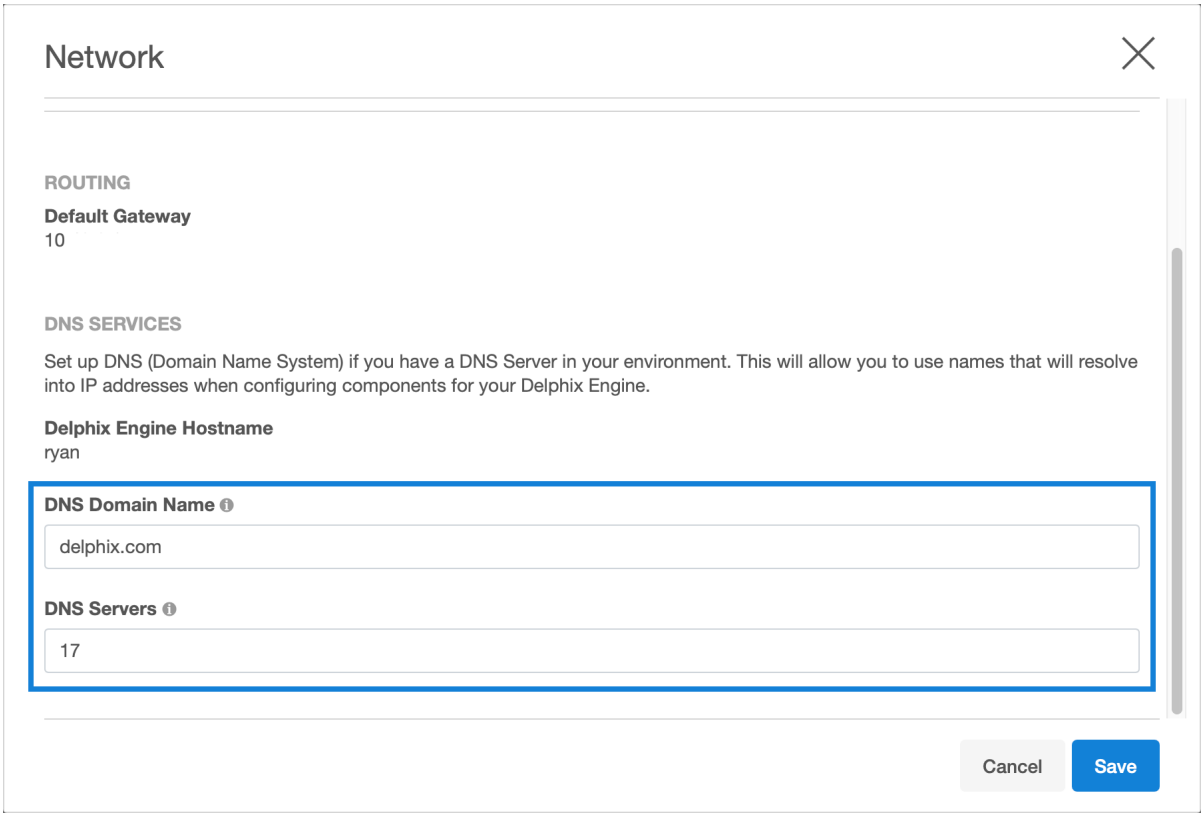
```
ssh -l sysadmin DelphixEngineName
Password:
DelphixEngineName> service dns
DelphixEngineName service dns> update
DelphixEngineName service dns update *> set servers=1.1.1.1,2.2.2.2
DelphixEngineName service dns update *> commit
DelphixEngineName service dns> exit
```

If DHCP is disabled and the address is static, the DNS server can be updated via the GUI.

1. Login to **Delphix Setup** with the appropriate credentials.
2. Locate the **Network** tile in the Dashboard and select **Modify**.



3. Scroll down in the **Network** window.
4. Use the **DNS Domain Name** and **DNS Servers** boxes for new configurations.



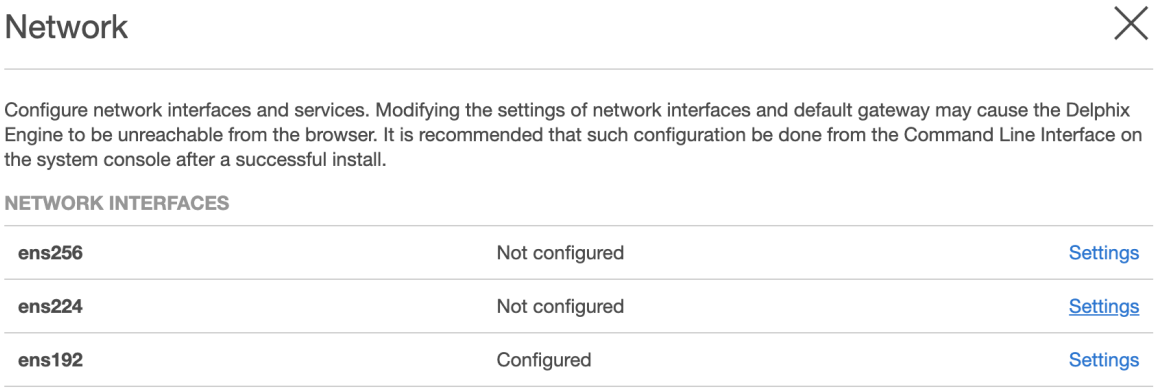
6.4.8 Configuring a second network interface

This topic describes how to configure a static IP address on a second network interface.

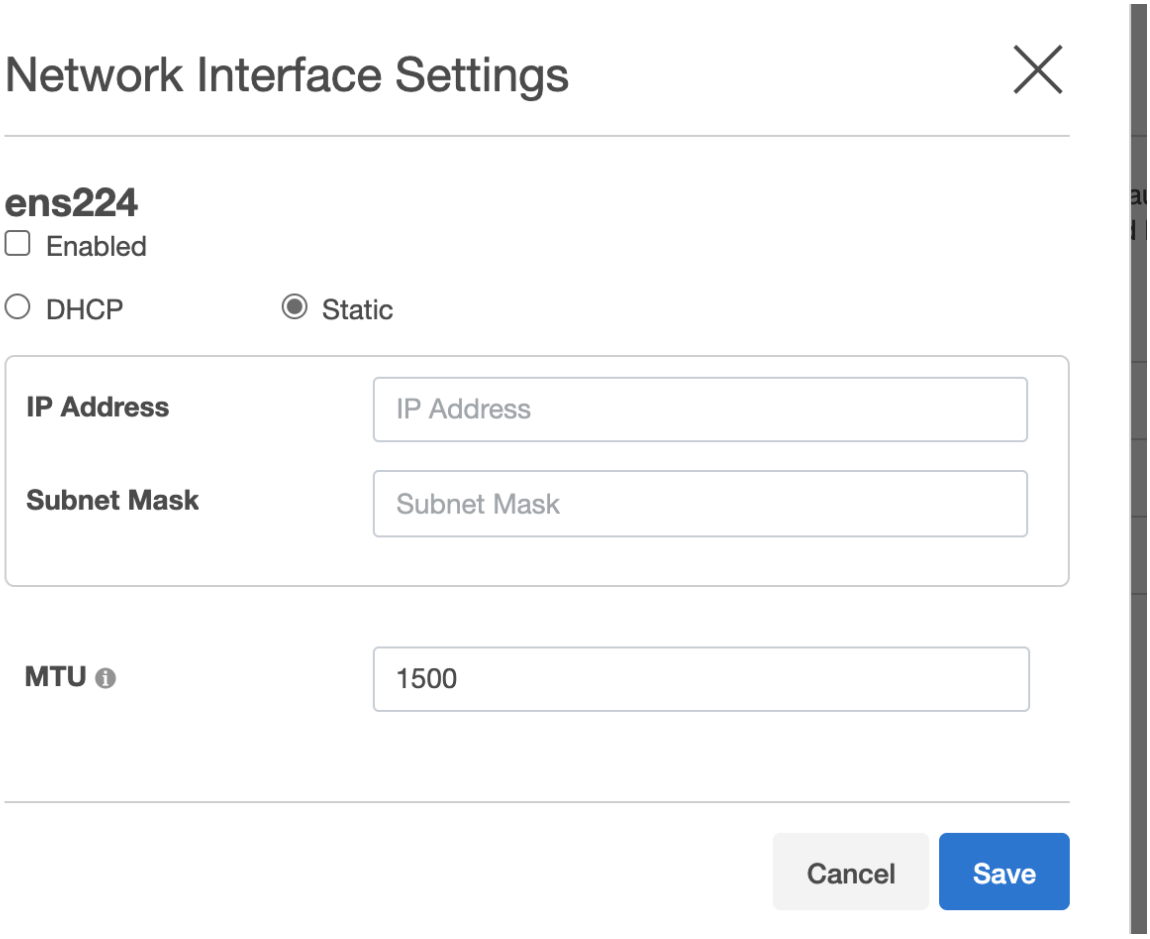
6.4.8.1 Procedure

Perform the following steps to configure a second network interface.

1. Launch the Delphix Engine Setup interface using the sysadmin credentials.
2. Navigate to the **Network** widget and click **Modify** to view the available network interfaces, and select the new interface to be configured.
3. Click the **Settings** button next to the network interface that you want to configure.



4. The **Network Interface Settings** screen appears. Select the checkbox before **Enabled** to enable the network.



5. Select one of the following: **DHCP** or **Static** andenter the **IP address** and **Subnet Mask** address in the respective fields.
6. **MTU**: Enter a value for the MTU field. This is the maximum size in bytes of a packet that can be transmitted on this interface.
7. Click **Save** to save the settings.

6.5 NFSv4 configuration

6.5.1 Overview

This article shows which OS versions support NFSv4. NFSv4 is enabled by default, but the target/staging hosts require additional configuration changes before it can be used.

Redhat/SLES: "NFSv4 Only - Enabling Recover Lost Locks" section under Linux/Redhat/CentOs (see page 907)

AIX: IBM AIX (see page 907) article.

OS version	NSFv4 support
RHEL 6.4 or later	Supported
SLES 11.4 or later	Supported
AIX 7.1, 7.2	Supported
Solaris 11	Supported

If these target host configurations are not set, the engine will choose NFSv3. The provided reason (e.g., "OS not supported") that is shown for choosing NFSv3 over NFSv4 can be observed under the Status tab of the selected VDB in the Delphix Engine GUI. Additional information is provided in this tab for datasets and current values.

6.5.2 Dataset status tab NFS reasons

NFS Reason	Notes
Default	NFSv4 is used by default for client mounts.
Old RedHat	NFSv4 is used with RedHat versions 6.4 and newer.
Unsupported OS	NFSv4 is not supported for HP-UX clients.
DNFS	Oracle Direct NFS (dNFS) is in use by the database. NFSv4 is supported by dNFS with Oracle versions 12c and newer. Additionally, due to Oracle bug 33596056, dNFS does not work with NFSv4 in Oracle versions 19.12-19.15 and 21.1-21.6.

NFS Reason	Notes
Tunable Override	The nfs.version tunable is set to force NFSv3 mounts.
Configuration Override	Appears with default operation, implies that a configuration parameter has changed.
No Recover Lost Locks	On Linux hosts, if the required recover_lost_locks setting is not enabled for the NFS client, then NFSv4 cannot be used.
Incomplete v4 Config	`On AIX hosts, if the nfsrgyd` daemon is not running or reverse DNS lookup is missing, then NFSv4 cannot be used.

6.5.3 NFSv4-Only mode

A Delphix Engine can be configured only to use NFSv4 (see page 746) and disable NFSv3 services.

Behavioral Differences Compared Automatic (default):

- If NFSv4 is not supported on the mount, then the Delphix Engine will fail to mount the VDBs and return a user exception with the reason "NFSv4 is not supported".
- If dNFS is enabled, the Delphix Engine will attempt to use NFSv4 in Oracle versions 19.12-19.15 and 21.1-21.6 despite Oracle bug 33596056. Additional details are in this KB[236].

> If NFSv3 is enabled, it will be turned off when the option is set:
> - If existing VDBs on a continuous data engine are using NFSv3, they will go down since the mounts will no longer serve data once the option is set.
> - If the Continuous Compliance Engine is using NFSv3 via remote mounts[237], they can no longer access data once the option is set.

This configuration is available via the System Administration CLI (see page 2555)

236 https://support.delphix.com/Continuous_Data_Engine_(formerly_Virtualization_Engine)/Oracle/Provision_fails_due_to_%22ORA-00600%3A_internal_error_code%2C_arguments%3A_%5Bksfdodm_libgetattr%5D%22_(KBA9251)

237 https://masking.delphix.com/docs/latest/managing-remote-mounts-for-vm-continuous-complianc

6.6 Capacity and resource management

Delphix will be responsible for managing many different data sources and data types. As such, it is critical to understand how to manage your capacity and storage resources within each Delphix engine. Learn storage and quota best practices, as well as the different options to optimally manage capacity.

This section covers the following topics:

- An overview of capacity and performance information (see page 748)
- Setting quotas (see page 760)
- Deleting objects to increase capacity (see page 763)
- Adding, expanding, and removing storage devices (see page 764)
- Delphix storage migration (see page 768)
- Managing source data (see page 781)
- An overview of held space (see page 782)

6.6.1 An overview of capacity and performance information

This topic describes the Delphix Engine performance reservoir and capacity threshold warnings and various ways to obtain information about capacity and resource usage for the Delphix Server.

6.6.1.1 The performance reservoir and capacity threshold warnings

In order to obtain the best performance and continued operations, the Delphix Engine requires a certain amount (minimum 512GB) of free space of the total quota for storage space. As storage capacity approaches this threshold, the following system faults occur:

- When **85%** of the total storage quota is reached or **1536GB** of free space is remaining (whichever is less), a **Warning** fault is triggered. You can resolve this fault by deleting objects in the Delphix Engine, adding storage, or changing policies, as described in the topics Adding, expanding, and removing storage devices (see page 764), Deleting objects to increase capacity (see page 763). Additionally, refreshing target databases will clear the space the engine uses to track changes over time for each DB.
  Note:
  When a **Warning** fault is first raised, it will not impact the functioning of the system. However, there may be an impact to Oracle LogSync and SQL Server Validated sync when a **Warning** fault is seen as storage usage decreases following a **Critical** storage usage fault.
- When **90%** of the total storage quota is reached or **1024GB** of free space is remaining (whichever is less), a **Critical** fault is triggered. You cannot ignore or resolve this fault. This fault will have a significant impact on the system's behavior such as:
  - All pending link, sync, refresh, and provisioning processes will be canceled, and no new operations can be initiated
  - Scheduled replication processes will first check for capacity on the target engine and hold data (replication currently in progress to the engine will not be halted)
  - Policy operations such as SnapSync, snapshot, and refresh are suspended for **all** databases

    - dSources stop pulling in new changes. LogSync is suspended for all **Oracle dSources**. Validated sync is disabled for **SQL Server** dSources.
    - No virtual database (VDB) snapshots can be taken.
- When **95%** of the total storage quota is reached or **512GB** of free space is remaining (whichever is less), a second **Critical** fault is triggered. This fault will have a significant impact on the system's behavior and certain dSources and VDBs will stop in order to maintain data integrity. For example, SQL Server VDBs will shut down.

As the free space of the system improves, the following functionalities can be automatically or manually resumed.

- When the system falls below **95%** of the total storage quota, you can manually start SQL Server VDBs that had stopped
- When the system falls below **90%** of the total storage quota:
    - SQL Server VDBs that had stopped will automatically start
    - New link, sync, refresh and provisioning operations are allowed
    - Policy operations such as SnapSync, Snapshot, and Refresh resume for all databases
- When the system falls below **85%** of the total storage quota
    - dSources start pulling in new changes from their corresponding data sources. LogSync is resumed for **Oracle dSources**. Validated sync is enabled for **SQL Server** dSources.

For more information, see Setting quotas (see page 760)

6.6.1.2 Ways to view capacity usage

You can access capacity and performance information for the Delphix Engine through several different means, including the **Dashboard** view, and the **Storage Capacity** screen.

6.6.1.2.1 The dashboard view

You can access the **Dashboard** view in the **Delphix Management** application by clicking **Dashboard** in the Manage menu. Note that the Dashboard view provides only summary information about capacity and performance. You must access the **Storage Capacity** and **Dataset Performance** screens in the **Resources** menu to manage storage space and database objects.

The Dashboard view provides more detailed information about the overall performance of the Delphix Engine:

- **Storage Capacity** - the amount of physical storage available and what is currently used
- **TimeFlow Ratio** - see above
- **VDB Ratio** - a measure of the amount of physical space that would be occupied by the database content against the amount of storage occupied by that same database content as VDBs.
- **Dataset Performance** - the amount of network bandwidth available and the amount that VDBs are currently utilizing, as well as information about specific VDB network usage

6.6.1.2.2 The storage capacity screen

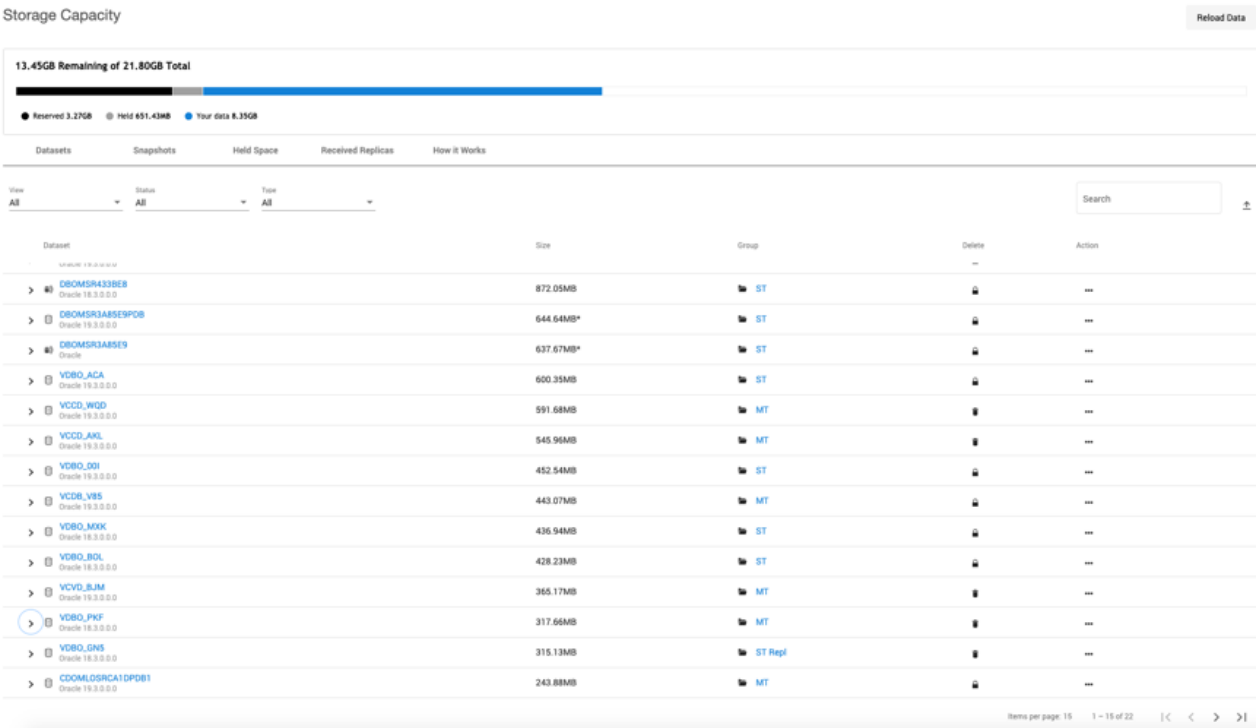
You can access the **Storage Capacity** screen through the **Resources** menu in the **Delphix Management** application. The **Storage Capacity** screen now has an improved format the interface makes it easier to identify which datasets are consuming the most space on your Delphix Engine.

Major additions in 6.0.6:

- Users can now see what's pinning a held space and batch delete the dependencies to free it up.

What's changed 6.0.6:

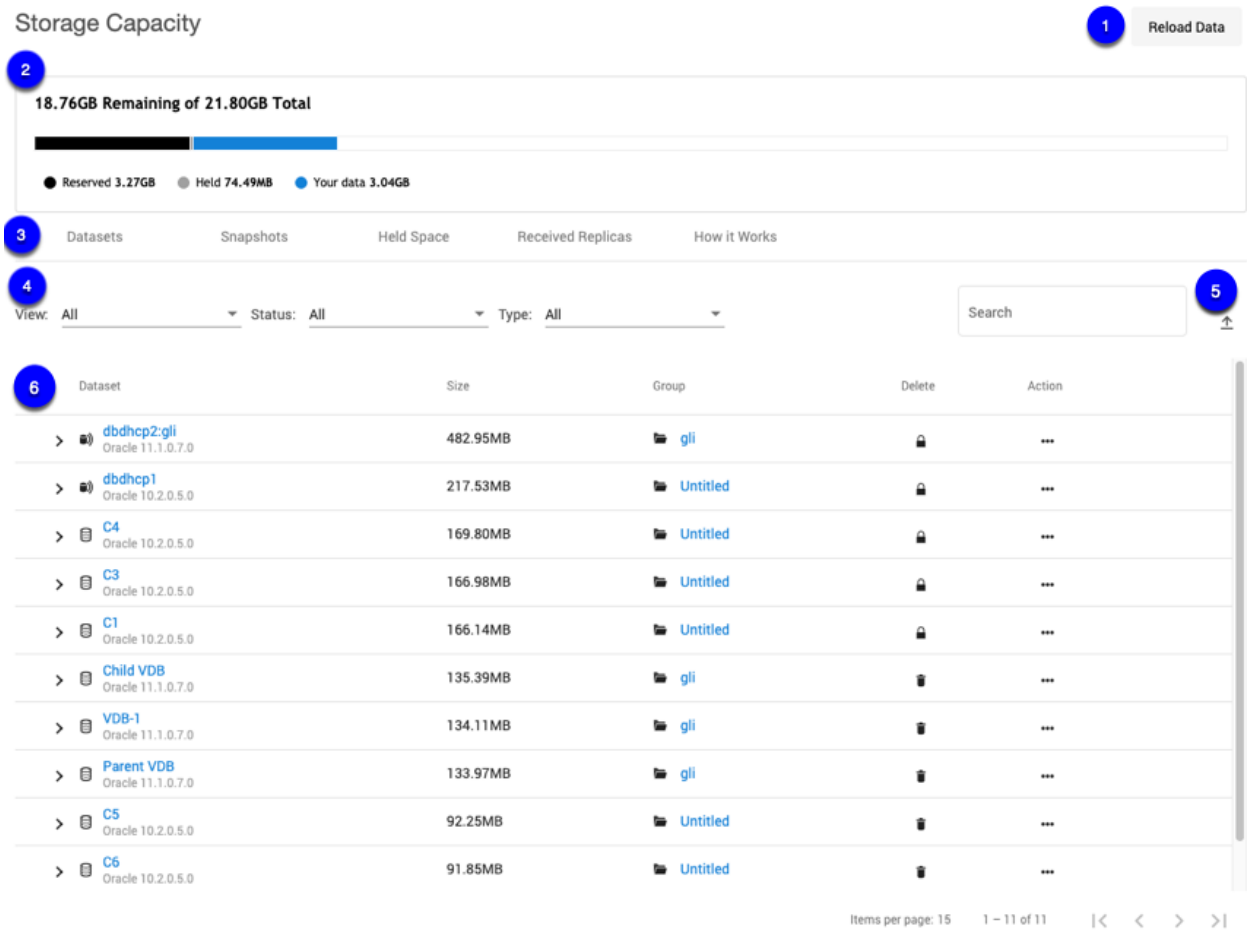
- Snapshot Delete dialog now shows a tree of "prerequisites" and "dependencies" that must have operations performed on them / be deleted in order to enable the deletion of the locked root snapshot.
- The Snapshot Delete dialog now automates the steps.
- Held Space shown in the table now represents deadbeat Timeflows rather than Containers



For more information please refer to Using and understanding the storage capacity screen (see page 750)

6.6.1.3 Using and understanding the storage capacity screen

The **Storage Capacity** screen can be reached via the **Capacity** item in the **Resources** menu. This screen shows how storage capacity is allocated for dSources, VDBs, and Snapshots, permitting storage space to be reclaimed through the deletion of objects and Snapshots.



1 Storage Capacity screen

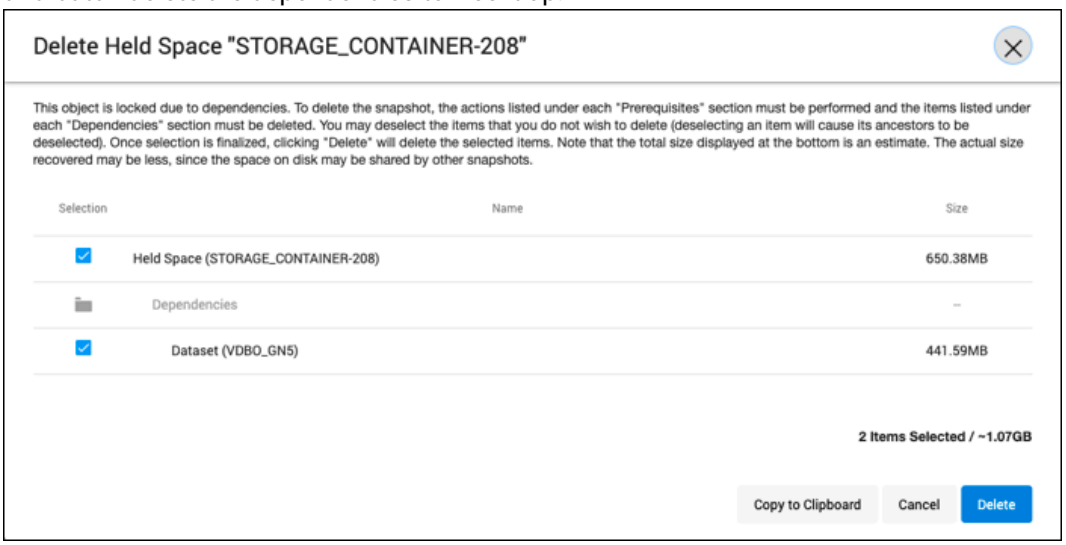
1. Select the **Reload Data** button to clear the capacity cache and refresh the page. This operation could take several minutes to clear the cache of a large Data Engine. During this operation, the Capacity page is not available.
2. The usage graph provides a visual representation of the distribution of storage consumption across the Data Engine.

- **Reserved** – The amount of reserved space is hard-coded to be 10% (maximum up to 1TB) of the system capacity and cannot be changed. The system has a 90% consumption cap.
- **Held** – Total held space size.
- **Your data** – User data (snapshot, log, etc.), which includes a shared space.

> Currently, this only includes dataset space. It does not include space of the engine metadata.

3. Click a tab to view one of the following tables:

- **Datasets** – This tab shows a list of datasets on the engine ordered by the size of the dataset. You can filter the list by whether the dataset is a VDB or dSource, locked or unlocked, and the type. The drop-down drawer of each dataset shows metadata about the corresponding dataset. If the dataset has descendants, the drop-down drawer will also show the list of **first-level** descendants. Use the Action menu (...) to delete or refresh a dataset. If a dataset has descendants, the delete column will show a lock icon that opens a dialog explaining why the dataset is locked. Replicated datasets are excluded.
- **Snapshots** – Rows are always sorted by descending size with a link to the dataset page for the Snapshot's parent dataset. Snapshots created from Replicas are excluded. The drop-down drawer for each snapshot shows the latest snapshot date of the dataset that this snapshot is from. If a snapshot has any descendants or cannot be deleted due to descendants, the delete column will show a locked icon indicating this snapshot is locked. Clicking on the lock icon will display the snapshot why-locked dialog. Freshly created Snapshots are not displayed, as this table is not updated automatically.
- **Held space** – Provides a list of Held Space on the engine. You can search the held spaces by their names (in this case their references). As shown in the screenshot, you can see what takes up space and batch delete the dependencies to free it up.



- **Received replicas** – Provides a list of namespaces sorted by descending size. This tab is updated automatically (except the size). The drop-down drawer lists groups inside of each replica, sorted by size.
- **How it works** – Provides general information about storage, retention policy, and manual deletion.

4. Filters available are table-specific and do not apply to the content inside a drawer.

- **View** – Dataset type (VDB, dSource, vFiles, etc.)
- **Status** – Locked/unlocked for deletion
- **Type** – Database type (Oracle, SQL Server, AppData, etc.)
- **Search** – Search by name (case insensitive)

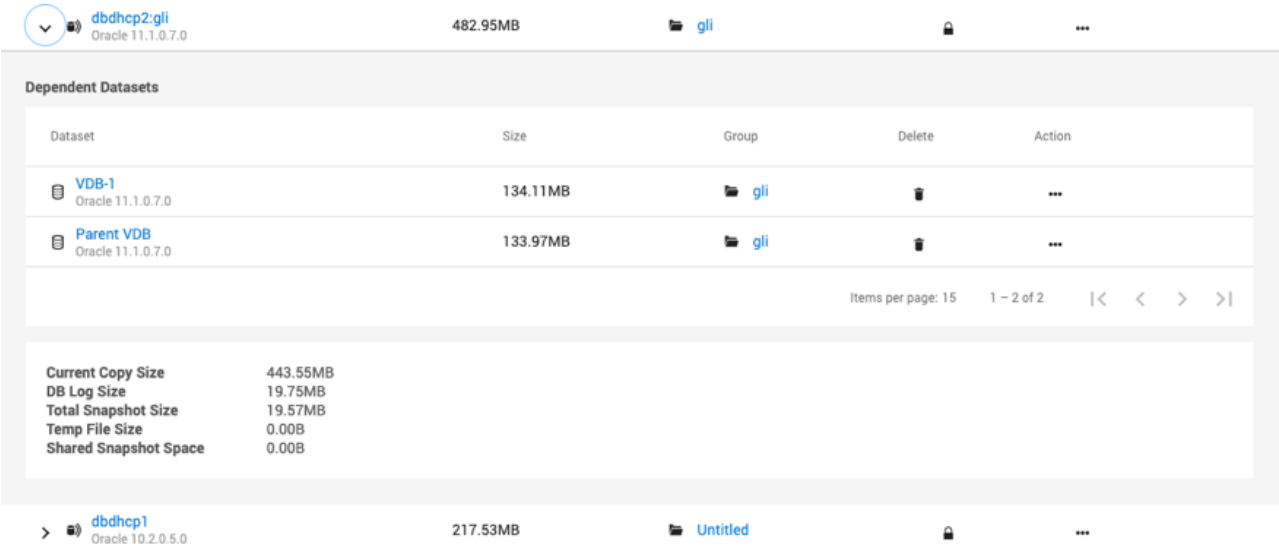
5. Select the export icon to export the information in the grid to a .csv file. Only the page you are viewing will be exported.

6. The **table** section displays a set of information about objects tracked by the Data Engine. The information displayed varies depending on the selected tab.

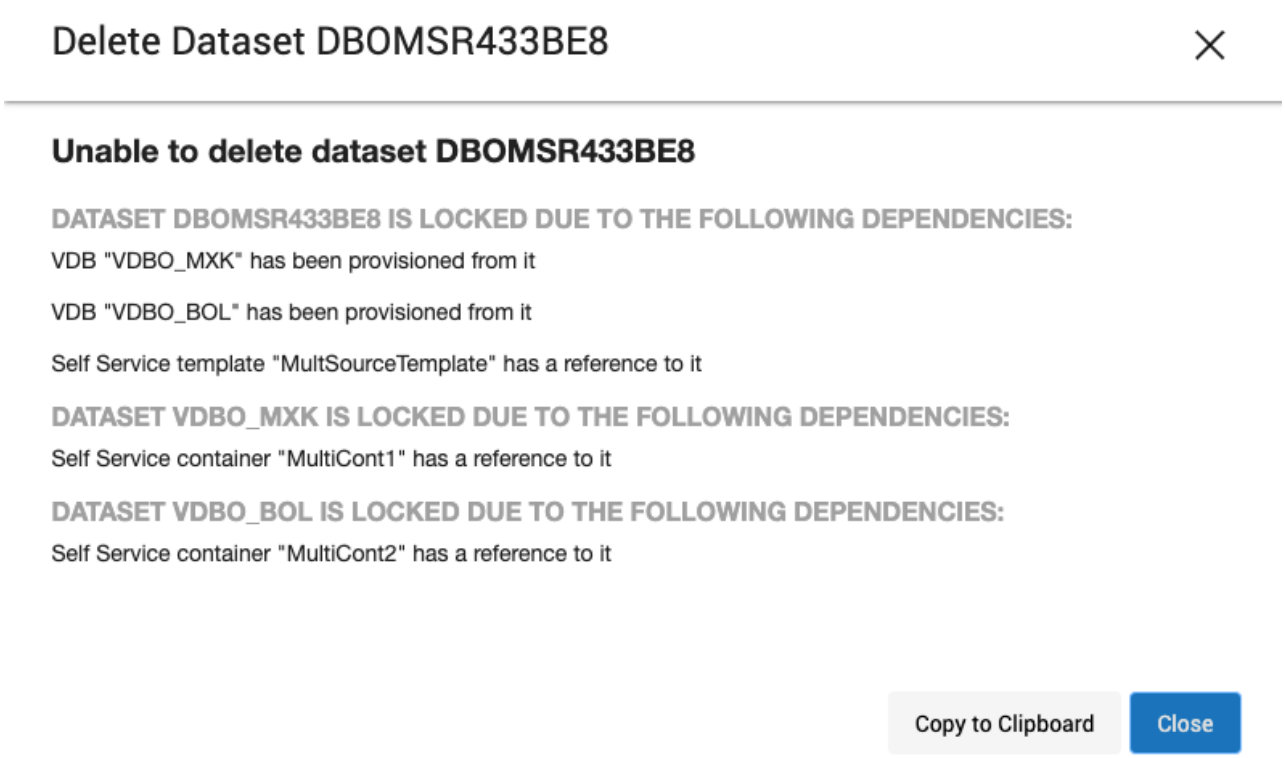
Datasets	
Dataset	Dataset type as an icon and name. Datasets created from replicas are excluded and deleted datasets have a strike-through name.
Size	Space actually used by the object.
Group	The name of the Dataset Group.
Delete	Either a trashcan or a lock icon will be shown. The trashcan opens a confirm/delete dialog, Oracle users can select to provide credentials. The lock icon explains why the selected object cannot be deleted.
Action	Refresh actions that can be taken against the dataset. • Refresh N/A • Refresh Dataset
Snapshots	
Timestamp	The timestamp reflecting when the snapshot was taken.
Size	Space actually used by the snapshot.
Dataset	The name of the dataset.
Group	The name of the Dataset Group.
Delete	Either a trashcan or a lock icon will be shown. The trashcan opens a confirm/delete dialog. The lock icon explains why the selected object cannot be deleted.
Held space	
Object	The name of the object.
Size	Space actually used by the object.

Delete	For held space, every row has a lock icon for opening the **why-locked** dialog.
Received replicas	
Namespace	Once replication is complete, the target engine will create a received replica, also known as a namespace.
Size	The amount of space used by the Namespace.

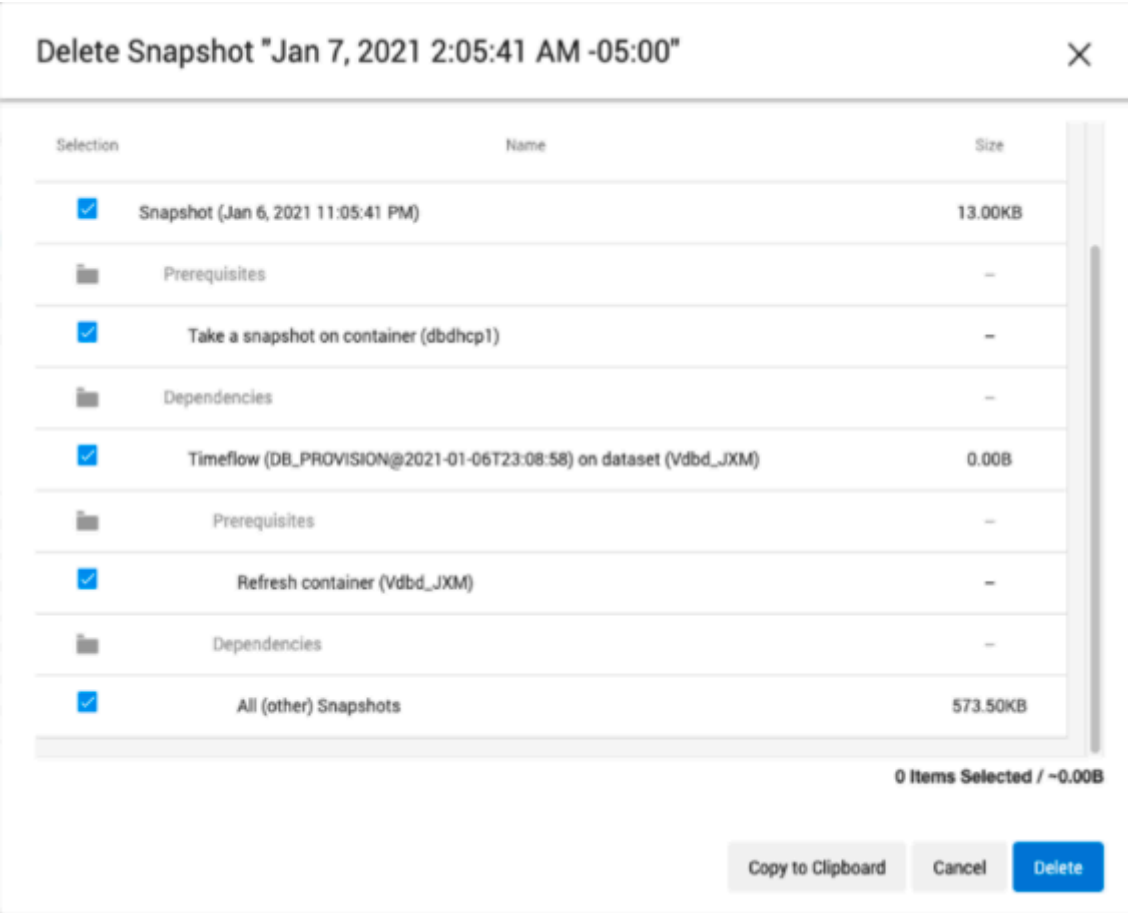
7. The dataset table or snapshot drawer provides information on dependent or descendant. The inner dependency table is direct children only and has the same behavior as the outer table. Filter or sorting from the outer table does not apply to the inner table.



The **Delete dataset dialog** – This dialog provides information on why a dataset cannot be deleted. It displays **all** descendants of a dataset, as well as any self-service objects on any of the descendant datasets. You can trace the chain of descendants from the current dataset all the way to leaf datasets. Click **Copy to Clipboard** to copy the list of descendants to the clipboard.



The **Delete Snapshot dialog** – The new snapshot dialog provides the complete steps on how to unlock the locked snapshot for deletion. It also allows users to select descendants and batch delete them. The dialog shows a tree of **prerequisites** and **dependencies** that must have an operation performed or be deleted in order to enable the deletion of the locked root snapshot. **Prerequisites** refer to operations that must be performed before the associated object can be deleted. Examples of prerequisite operations include: removing a keep-forever retention policy from a snapshot, deleting a Self-Service bookmark, or refreshing a dataset. **Dependencies** refer to objects that must be deleted before the parent object can be deleted. There is also a special **All (other) Snapshots** dependency item. This item represents all children snapshots of the parent Timeflow, which will also need to be removed, minus those with a dedicated row in the table.

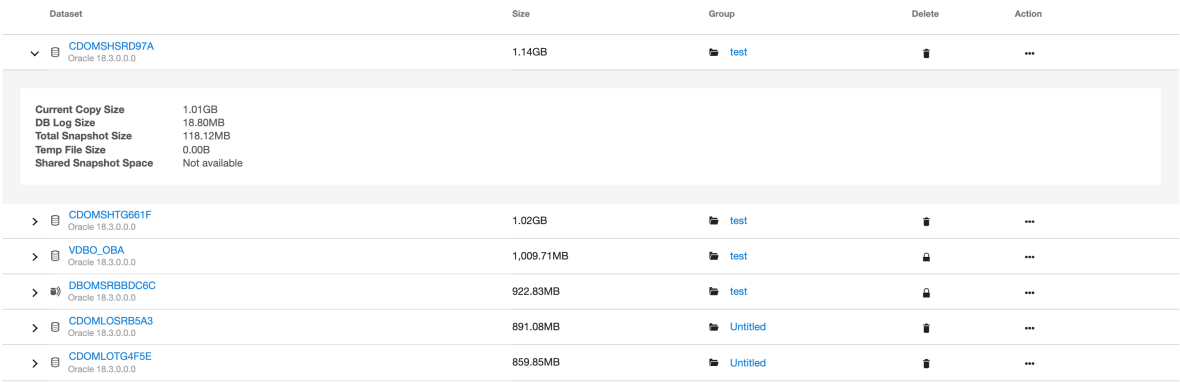


6.6.1.4 Understanding Delphix disk usage

Within the Delphix Timeflow, incremental restore points are called Snapshots. Delphix Snapshots use a shared block architecture to provide all the functionality of daily full backups using only a tiny percentage of the storage for full backups. Because shared block architectures are not common, a "common sense" understanding of how storage is associated with a Snapshot can be misleading.

- The size of a Snapshot is defined to be the size of changes that are unique to that Snapshot. In other words, changed blocks that are only associated with that one Snapshot.
- The latest Snapshot will always have 0 sizes initially, as there are no changes associated with it, nothing has changed since the Snapshot was taken, so there is no space used by unique changes.
- A block can be shared by multiple Snapshots if the block has not changed between the creation of those Snapshots. Any blocks that are shared amongst multiple Snapshots are accounted for in the shared Snapshot Space total.

The following screenshot provides an example of disk usage.



- **Current Copy Size:** The current copy size is the amount of space used on the Delphix Engine by the VDBs data but across all the Time flows.
- **DB Log Size:** The DB log size is the size for archive log files.
- **Temp File Size:** This field is currently not active (for future use).
- **Container Size:** The container size is the size of the whole dataset.
- **Total Snapshot Size:** The total snapshot size is the size of all snapshots combined, including anything shared.
- **Shared snapshot space:** The shared snapshot space is the space shared by snapshots.

6.6.1.5 Reviewing historical capacity from the CLI

You can retrieve historical capacity data through the Delphix Command Line Interface (CLI).

If your Delphix Engine is a new engine you would log in as Engine Admin and ssh admin@your engine. Engines created before 5.3.1 and upgraded to 5.3.1 or later will retain their old username 'delphix_admin'. To avoid complications Delphix recommends creating users with an admin role and then disabling delphix_admin.

6.6.1.5.1 Process

1. Log in as Delphix Admin (or an account with admin privileges)
   `ssh delphix_admin@yourengine`
2. Navigate to capacity system historical
   `delphix > capacity system historical`
3. Then you can list a start and end date for the utilization
   `delphix capacity system historical > list startDate=<time> endDate=<time>`

For example, looking at the system space utilization, we can show 8 hours starting Sep 1 with the following:

We can also set the output detail level to different granularity, which is based on **seconds**.

```
delphix> capacity
delphix capacity> system
delphix capacity system> historical
delphix capacity system historical> list startDate=2016-09-01T00:00:00.000Z
endDate=2016-09-01T08:00:00.000Z
TIMESTAMP SOURCE.ACTUALSPACE VIRTUAL.ACTUALSPACE
2016-09-01T00:25:11.359Z 5.05TB 9.58TB
2016-09-01T00:55:28.869Z 5.06TB 9.44TB
2016-09-01T01:25:42.940Z 5.06TB 9.46TB
2016-09-01T01:56:14.585Z 5.06TB 9.38TB
2016-09-01T02:26:50.893Z 5.06TB 9.38TB
2016-09-01T02:57:15.987Z 5.06TB 9.32TB
2016-09-01T03:27:35.381Z 5.06TB 9.34TB
2016-09-01T03:57:54.657Z 5.06TB 9.35TB
2016-09-01T04:28:12.099Z 5.06TB 9.35TB
2016-09-01T04:58:22.028Z 5.06TB 9.35TB
2016-09-01T05:28:33.680Z 5.06TB 9.34TB
2016-09-01T05:58:46.666Z 5.06TB 9.27TB
2016-09-01T06:28:57.181Z 5.06TB 9.21TB
2016-09-01T06:59:39.567Z 5.06TB 9.21TB
2016-09-01T07:30:03.527Z 5.06TB 9.20TB
2016-09-01T07:50:18.548Z 5.06TB 9.20TB
```

The following example will show the data for **each day** (86400 seconds) from Sep 1 to Oct 1:

```
delphix capacity system historical> list startDate=2016-09-01T00:00:00.000Z
endDate=2016-10-01T08:00:00.000Z resolution=86400
TIMESTAMP SOURCE.ACTUALSPACE VIRTUAL.ACTUALSPACE
2016-09-01T00:25:11.359Z 5.05TB 9.58TB
2016-09-02T00:34:16.844Z 5.03TB 8.45TB
2016-09-03T00:37:50.372Z 5.10TB 8.30TB
2016-09-04T00:32:22.877Z 4.68TB 8.79TB
2016-09-05T00:34:37.715Z 4.64TB 8.56TB
2016-09-06T00:37:27.480Z 4.65TB 8.44TB
2016-09-07T00:31:17.808Z 4.70TB 8.60TB
2016-09-08T00:33:34.219Z 4.74TB 8.70TB
2016-09-09T00:43:40.760Z 4.81TB 8.57TB
2016-09-10T00:48:27.222Z 5.14TB 8.95TB
2016-09-11T00:50:41.843Z 4.90TB 8.72TB
2016-09-12T00:50:33.215Z 4.92TB 8.82TB
2016-09-13T00:48:06.350Z 4.93TB 8.72TB
2016-09-14T00:42:36.904Z 5.34TB 8.79TB
2016-09-15T00:48:58.580Z 0B 0B
2016-09-16T00:43:12.565Z 5.64TB 9.23TB
2016-09-16T21:46:06.333Z 5.74TB 9.29TB
2016-09-18T19:41:29.692Z 5.74TB 9.29TB
2016-09-19T19:38:35.268Z 5.47TB 7.94TB
2016-09-20T19:39:07.809Z 5.57TB 8.50TB
2016-09-21T19:42:32.602Z 5.64TB 9.04TB
2016-09-22T19:37:53.507Z 5.82TB 8.77TB
2016-09-23T19:37:43.373Z 5.88TB 8.96TB
2016-09-24T19:29:40.625Z 5.78TB 9.03TB
2016-09-25T19:32:30.592Z 5.80TB 8.91TB
2016-09-26T19:26:43.971Z 5.81TB 8.86TB
2016-09-27T19:33:26.939Z 5.87TB 9.44TB
2016-09-28T19:33:01.483Z 5.94TB 9.44TB
2016-09-29T19:43:55.451Z 6.15TB 9.08TB
2016-09-30T19:37:38.462Z 6.19TB 9.72TB
2016-10-01T07:35:53.644Z 5.87TB 8.98TB
```

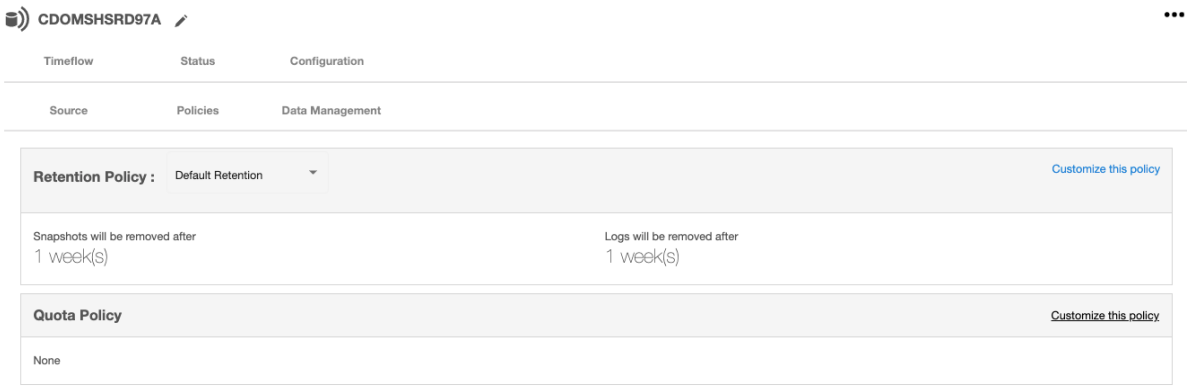
The following query will show the data for each week (604800 seconds), for a few months:

```
delphix capacity system historical> list startDate=2016-06-01T00:00:00.000Z
endDate=2016-10-01T08:00:00.000Z resolution=604800
TIMESTAMP SOURCE.ACTUALSPACE VIRTUAL.ACTUALSPACE
2016-06-01T00:16:19.391Z 3.36TB 5.96TB
2016-06-08T00:16:04.620Z 2.79TB 6.11TB
2016-06-15T00:16:35.471Z 2.95TB 4.84TB
2016-06-22T00:22:57.619Z 0B 0B
2016-06-29T00:28:48.350Z 3.38TB 6.53TB
2016-07-06T00:38:20.643Z 3.38TB 7.33TB
2016-07-13T00:36:38.876Z 3.30TB 8.05TB
2016-07-20T00:50:24.318Z 3.33TB 8.95TB
2016-07-27T00:49:06.488Z 3.36TB 9.28TB
2016-08-03T01:03:59.577Z 3.59TB 9.25TB
2016-08-10T01:12:21.949Z 3.81TB 8.80TB
2016-08-17T01:20:44.878Z 4.08TB 9.89TB
2016-08-24T01:21:01.080Z 4.23TB 9.00TB
2016-08-31T01:29:51.907Z 4.45TB 9.48TB
2016-09-07T01:31:56.383Z 4.70TB 8.67TB
2016-09-14T01:43:21.456Z 5.34TB 8.87TB
2016-09-21T01:42:34.318Z 5.58TB 8.84TB
2016-09-28T01:28:10.313Z 5.88TB 9.47TB
2016-10-01T07:35:53.644Z 5.87TB 8.98TB
```

6.6.2 Setting quotas

This topic describes how to set quotas for database objects.

1. Log into the **Delphix Management** application.
2. Select **Manage > Datasets**.
3. For setting a quota for a dataset, select the **Configuration** tab for the dataset. (For setting quota for a group, skip this step.)
4. Select the **Policies** tab and from the Quota tile and select **Customize this policy**.



5. In the **Quota Policy** window enter the amount of storage space you want to allocate for a quota.

6. Click **Save** to set the amount.

> Quotas and Low Space Errors
>
> Be very careful setting quotas. As a group or virtual database (VDB) approaches the quota level, snapshots may fail and logs may not be captured, causing LogSync to fail.

6.6.2.1 Quota thresholds[238]

The following is a table of the thresholds and a description of the actions taken at each threshold. This behavior is generic across all engines.

238 https://docs.delphix.com/display/VDR/Quota+Thresholds

Thresholds	Default ranges	Actions taken
Critical	Quota ≥ 95%	• Disable VDBs and dSources under quota (for a group quota, everything inside gets disabled). • Disallow jobs detailed in the DB actions table below. • Cancel all in-flight jobs listed in the DB actions table except for enabling. • Poll every minute.
Resume	95% > Quota ≥ 90%	• VDBs and dSources disabled by **critical** quotas will not be automatically re-enabled (DBs can still be manually re-enabled). • **Resume** fault is triggered if the quota has fallen from the **critical** threshold and if a DB was disabled from hitting the **critical** threshold. • **Warning** fault is triggered. • Poll every 3 minutes.
Warning	90% > Quota ≥ 80%	• **Warning** fault is triggered. • VDBs and dSources disabled from hitting **critical** quota will be re-enabled. • Poll every 3 minutes.
Safe	80% > Quota	• VDBs and dSources disabled from hitting **critical** quota will be re-enabled. • Poll every 5 minutes.

If a DB was disabled prior to reaching the **critical** quota, it will not be automatically re-enabled when falling to an acceptable range or when the quota is removed.

6.6.2.2 Disallowed database actions when in quota critical threshold

Here is a table of the actions that are not allowed when the quota is in the critical threshold.

DB actions table

Target is group	Target is not group
• Refresh target in the group. • Sync target in the group. • Link into the group. • Provision into the group. • Enable target in the group.	• Refresh target in the group. • Sync target in the group. • Enable target in the group.

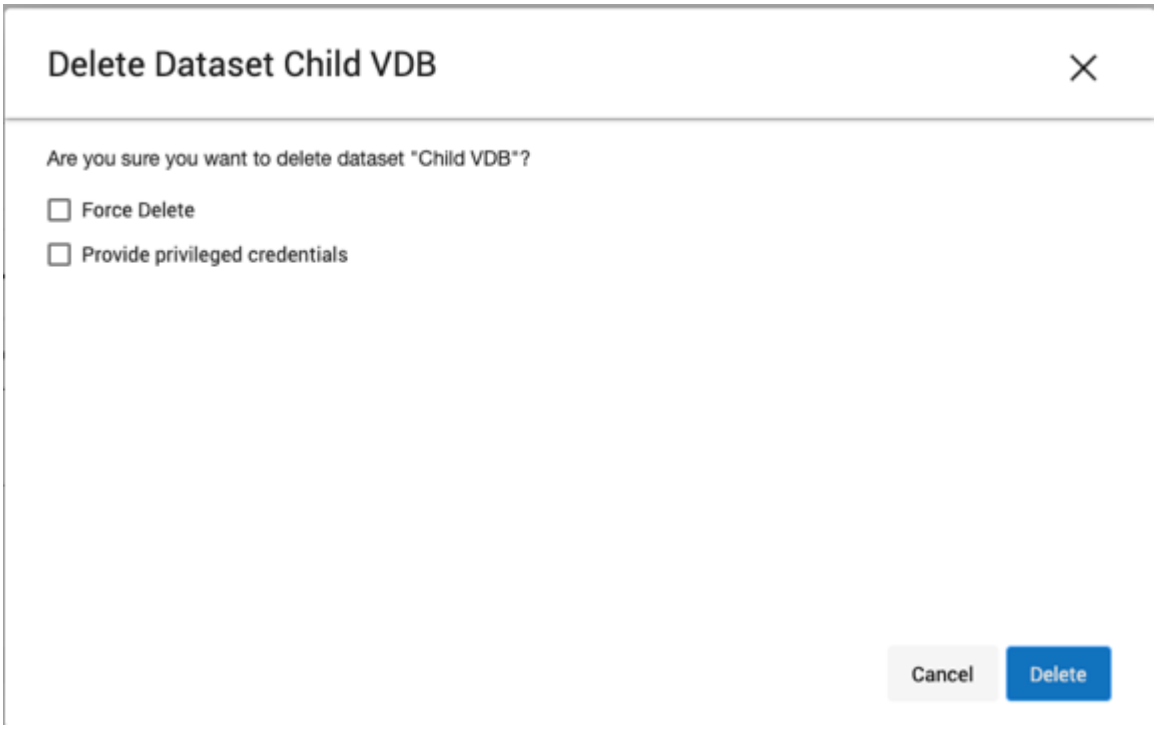
If a **critical** quota is placed on a target, dSources, and VDBs can still provision to groups that are quota below the **critical** threshold.

6.6.3 Deleting objects to increase capacity

This topic describes how to delete database objects to create additional capacity.

Deleting unused or outdated objects should be a regular part of Delphix Engine administration. This is especially important to prevent low space errors, which can cause the Delphix Engine to stop. The Delphix Engine holds a maximum of 400 objects.

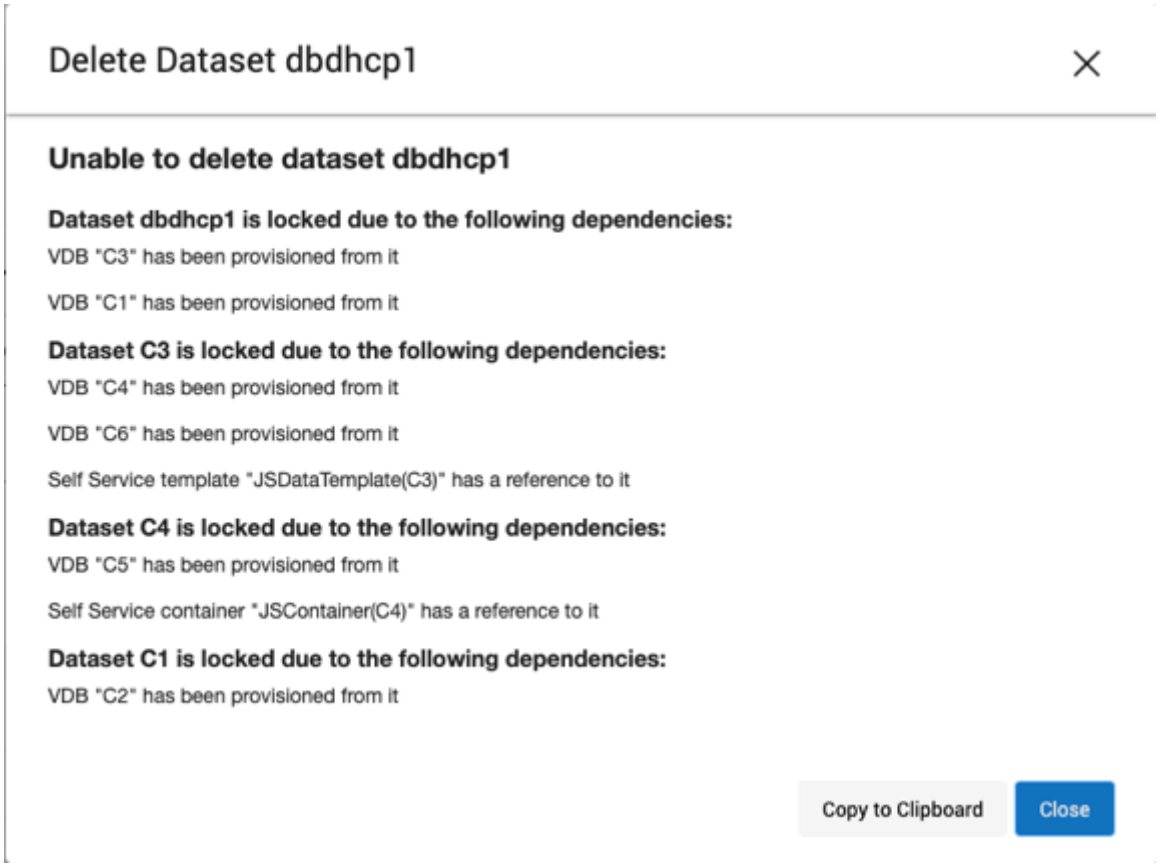
1. Log into the **Delphix Management** application.
2. Select **Resources > Storage Capacity**.
3. Next to the object you want to delete select the **Trashcan**.
4. In the **Delete** dialog, select **Force Delete**. Oracle users will have the option to provide additional credentials.



5. Click Delete.

Dependencies

If there are dependencies on the snapshot, you will not be able to delete the snapshot free space; the dependencies rely on the data associated with the snapshot. These items are displayed with a lock icon next to the name.



6.6.4 Adding, expanding, and removing storage devices

Multiple Device Removal in the Delphix Engine 6.0.12.0 and higher version introduces a breaking kernel module change that requires a reboot to load the new module. Therefore, a deferred reboot engine upgrade operation will be unable to remove devices until a reboot is performed.

6.6.4.1 Prerequisites

For expanding a storage device after initial configuration, first make sure to add capacity to it using the storage management tools available through the device's operating system. For example, capacity can be added in vSphere using the **Edit system settings**.

> **VMware Hypervisor (vSphere)**
>
> Rebooting the Delphix Engine to add/expand storage is not typically necessary when using vSphere.
>
> The guidelines for adding initial storage using all 4 virtual SCSI controllers should cause the Delphix Engine to see new storage without a reboot. However, if the new storage is leveraging a new virtual SCSI controller, the Delphix Engine will need a reboot to detect the new storage. See Deployment for VMware[239] for more information.

6.6.4.2 Adding or increasing storage and/or cache

If it is available you can add more storage devices to the Delphix Engine.

1. Launch the **Delphix Setup** application and log in using the **sysadmin** credentials.
2. In the **Storage** section of the **Server Setup Summary** screen, click **Modify**.
3. For engines backed by disks
   a. Under the **Block Storage** tab, the Delphix engine should automatically detect any new storage devices. If a newly added storage device does not appear in the **Storage** section of the **Server Setup Summary** screen, click **Rediscover**.
   b. Select the **Enable** check box before the device name to add the device to the storage pool and click **Save**.
4. For Delphix Elastic Data Engines
   a. You can modify the maximum amount of data that can be stored by the s3 storage under the **Object Storage** tab.
   b. The **Block Storage** tab lets you modify the number of EBS volumes used for cache. Please see step 3 above.
5. Click **Save**.

6.6.4.3 Expanding a storage/cache device

1. Launch the **Delphix Setup** application and log in using the **sysadmin** credentials.

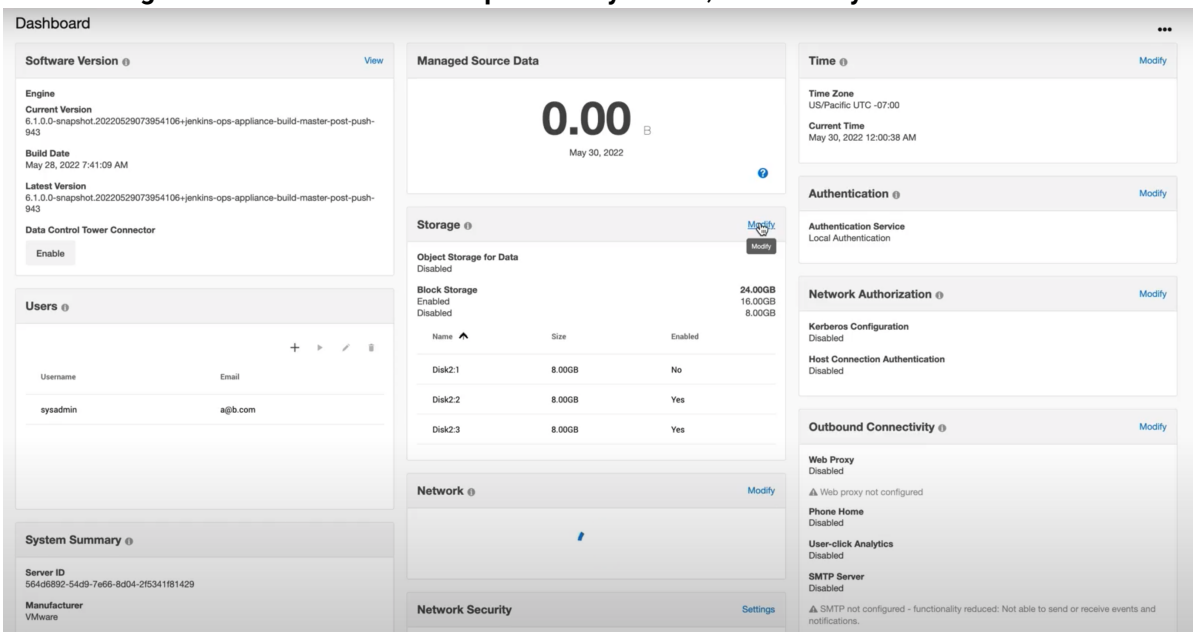
239 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/deployment-for-vmware

2. In the **Storage** section of the **Server Setup Summary** screen, click **Modify**.
3. Under the **Block Storage** tab, select **Expand** for each device that you want to expand. The **Expand** checkbox appears next to the name of devices that have added capacity (in other words, the underlying LUN has been expanded), and the **Unused** column indicates how much capacity is available for each device.
   Note:

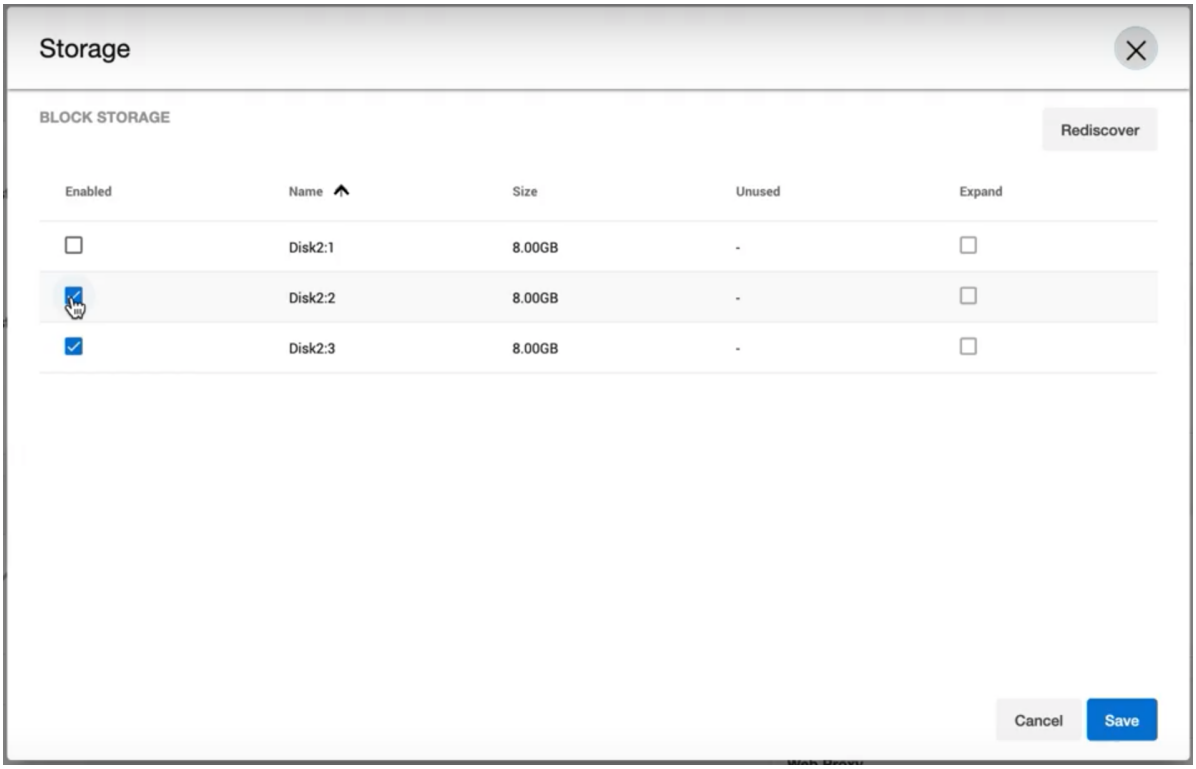
   It is imperative that the VMDKs created and assigned to the Delphix Engine do not exceed the capacity of the backing datastore. Any 'overcommitment' of storage resources can be expected to cause an unplanned outage, and may not be recoverable.
4. Click **Save**.

6.6.4.4 Removing device storage

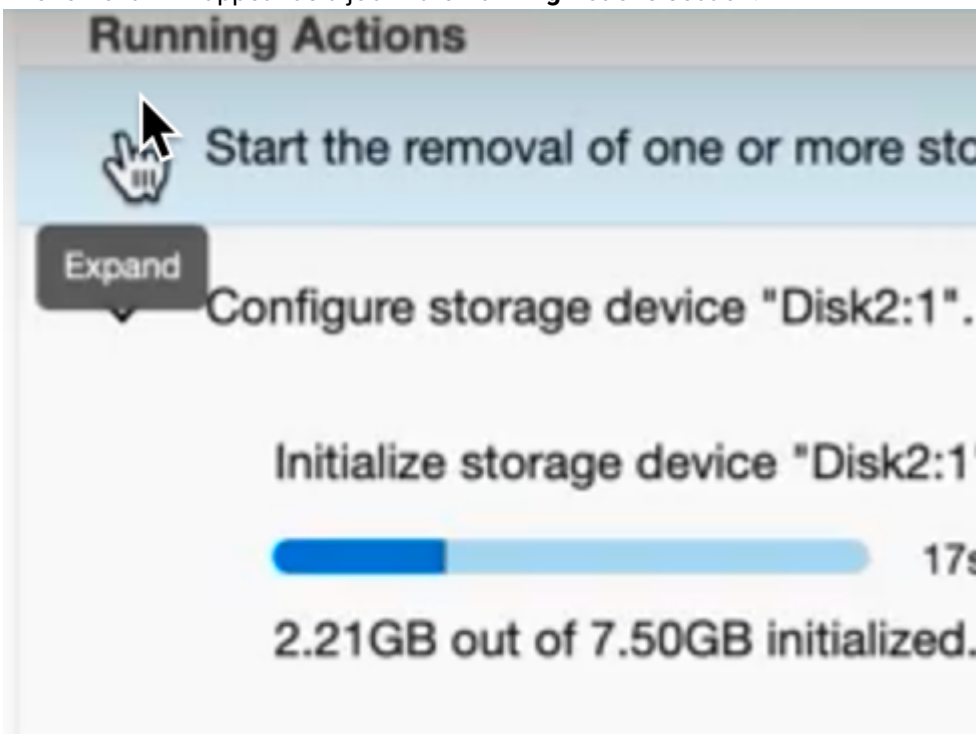
1. Launch the **Delphix Setup** application and log in using the **sysadmin** credentials.
2. In the **Storage** section of the **Server Setup Summary** screen, click **Modify**.



3. In the **Block Storage** tab, uncheck the **Enable box** for any storage device that will be removed.
   a. A minimum of 10GB of object storage is required, thus, a device or devices must be enabled to accommodate for at least 10GB. This will give an error otherwise.



4. Once the desired storage devices are disabled, click **Save**.
5. The removal will appear as a job in the **Running Actions** section.



6.6.5 Delphix storage migration

6.6.5.1 Getting started

Delphix storage migration allows you to remove storage devices from your Delphix Engine, provided there is sufficient unused space, thus allowing you to repurpose the removed storage device to a different server. You can also use this feature to migrate your Delphix Engine to different storage by adding the new storage devices and then removing the old storage devices.

Feature compatibility
This feature is only compatible with Delphix Engine Releases 5.0.4 and later. This feature is not supported on Delphix Elastic Data Engines (engines backed by blob/object storage).

Possible Migration Methods

Method	Pros	Cons
Delphix Storage Migration	• Good for migrating storage that was accidentally added to the engine or added to the engine improperly (wrong size). • May reduce fragmentation if new storage is added to replace old disks. • Good for migrating a small amount of storage (e.g.: < 10 TB).	• With Delphix versions prior to 5.3, this mapping table can consume 2-3GB of RAM for every 1TB of allocated data that is migrated, if the disk being removed has a high level of fragmentation. From version 5.3, DxFS will migrate the data in larger blocks, comprising both allocated and unallocated space. This allows for significantly fewer mapping entries, with memory usage typically reduced to 50-100MB per TB of allocated data that is migrated. • May increase fragmentation on remaining disks if no new disks are added. • Depending on the size of the disk and storage performance this method could be less performant than other methods. • Each device removed could take longer than the previous one as data is remapped across the remaining disks. • The maximum number of 20 devices can be removed in releases prior to Delphix version 5.1.
vMotion	• Fast	• If there is high fragmentation on the existing disks, this is copied to the new disks.

Method	Pros	Cons
Delphix Replication	• Data is completely rewritten from one Delphix Engine to another which significantly reduces fragmentation on the new Delphix Engine. • Replication can be configured to limit the impact on the network (compression and bandwidth). • Replication is resumable from network disruptions, on in the event of replication source or target stack/host restarts. It is currently NOT possible to manually suspend/resume a replication job.	• Depending on the number of objects to replicate as well as network and storage performance, this method could be considered slow. • This does require an outage to "migrate" the objects from the replication source Delphix Engine to the replication target Delphix Engine - outage time depends on several factors like the number of objects, incremental replication time, time to enable/disable objects, etc. • Only migrates storage objects like VDBs/dSources, and dependent environment information. Other items like users/policies/events/job history/config templates are NOT replicated.

6.6.5.2 Understanding Delphix storage migration

Delphix storage migration is a multi-step process:

1. Identify a storage device for removal. The device you choose will depend on your use case.
   a. To remove extra storage that is unused, you can select any device for removal. For best performance, select the device with the least allocated space; typically, this is the device that you added most recently. The allocated space is defined by the `usedSize` property of the storage device:

```
test-env@delphix 'Disk10:2'> ls
Properties
```

```
    type: ConfiguredStorageDevice
    name: Disk10:2
    bootDevice: false
    configured: true
    expandableSize: 0B
    model: Virtual disk
    reference:  STORAGE_DEVICE-6000c293733774b7bb0e4aea83513b36
    serial: 6000c293733774b7bb0e4aea83513b36
      size: 8GB
      usedSize: 7.56MB
    vendor: VMware
```

   b. To migrate the Delphix Engine to new storage, add storage devices backed by the new storage to the Delphix Engine. Then remove all the devices on the old storage.

2. Use the Delphix command-line interface (CLI) to initiate the removal of your selected device.
3. Data will be migrated from the selected storage device to the other configured storage devices. This process will take longer the more data there is to move; for very large disks, it could potentially take hours. You can cancel this step if necessary.
4. The status of the device changes from **configured** to **unconfigured** and an alert is generated to inform you that you can safely detach the storage device from the Delphix Engine. After this point, it is not possible to undo the removal, although it is possible to add the storage device back to the Delphix Engine.
5. Use the hypervisor to detach the storage device from the Delphix Engine. After this point, the Delphix Engine is no longer using the storage device, and you can safely re-use or destroy it.

6.6.5.3 Limitations of Delphix storage migration

After removal, the Delphix Engine uses memory to track the removed data. In the worst-case scenario, this could be as much as 2-3 GB of memory per TB of used storage. Note that this is used storage; the overhead of removing a 1TB device with only 500MB of data on it will be much lower than the overhead of removing a 10GB device with 5GB of data on it.

6.6.5.4 User interface

Delphix storage migration is currently available exclusively via the CLI. There are two entry points.

- `verifyStorageDeviceRemoval` - Verifies available pool space and available memory.
- `startStorageDeviceRemoval` - Kicks off job, detailed discussion of job concurrency/recovery.

6.6.5.5 Device removal for storage migration

Do not remove a configured storage device

> Do not remove a configured storage device or reduce its capacity. Removing or reducing a configured storage device will cause a fault with the Delphix Engine, and will require the assistance of Delphix Support for recovery.

1. Identify which device you want to remove.
   a. If you are using a VMware **RDM** disk, note the UUID of the device by looking at its name in the vSphere GUI. For more information, see the Getting the UUID of a RDM Disk from VMware, via the vSphere GUI[240] article.
   b. If you are using a VMware **virtual** disk, note the UUID of the device via the vSphere API. See the section of this VMware KB article[241] on how to get the UUID of your virtual disk.
   c. In EC2, note the attachment point – e.g., `/dev/sdf`.
   d. In KVM, note the UUID.
2. Login to the Delphix CLI as a **sysadmin** user.
3. Navigate to the storage/device directory with `cd storage/device`.
4. Select one or more devices to remove and make note of the name (e.g., select "Disk10:2" and "Disk10:3").

```
test-env@delphix storage/device> ls
Objects
NAME      CONFIGURED  SIZE  EXPANDABLESIZE
Disk10:2  true        8GB   0B
Disk10:0  true        24GB  0B
Disk10:1  true        8GB   0B
Disk10:3  true        8GB   0B
```

5. (For VMware only) Confirm the disk selection is correct by validating that the serial matches your UUID:

```
test-env@delphix storage device 'Disk10:2'> ls
Properties
   type: ConfiguredStorageDevice
   name: Disk10:2
   bootDevice: false
   configured: true
   expandableSize: 0B
   model: Virtual disk
   reference: STORAGE_DEVICE-6000c2909ccd9d3e4b5d62d733c5112f
   serial: 6000c2909ccd9d3e4b5d62d733c5112f
   size: 8GB
   usedSize: 8.02MB
   vendor: VMware
```

240 https://delphixdocs.atlassian.net/wiki/pages/resumedraft.action?draftId=6230802
241 http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1033370

6. Navigate to `storage/remove`. Run `verify` and set the devices to be removed, as shown below.

```
test-env@delphix storage remove > verify
test-env@delphix storage remove verify *> set devices="Disk10:2,Disk10:3"
test-env@delphix storage remove verify *> commit
   type: StorageDeviceRemovalVerifyResult
   newFreeBytes: 15.85GB
   newMappingMemory: 3.14KB
   oldFreeBytes: 23.79GB
   oldMappingMemory: 0B
```

7. Navigate to `storage/remove`. Run `start` and set the devices to be removed, as shown below.

```
test-env@delphix storage remove > start
test-env@delphix storage remove start *> set devices="Disk10:2,Disk10:3"
test-env@delphix storage remove start *> commit
    Dispatched job JOB-4
    STORAGE_DEVICES_START_REMOVAL job started.
    STORAGE_DEVICES_START_REMOVAL job completed successfully.
```

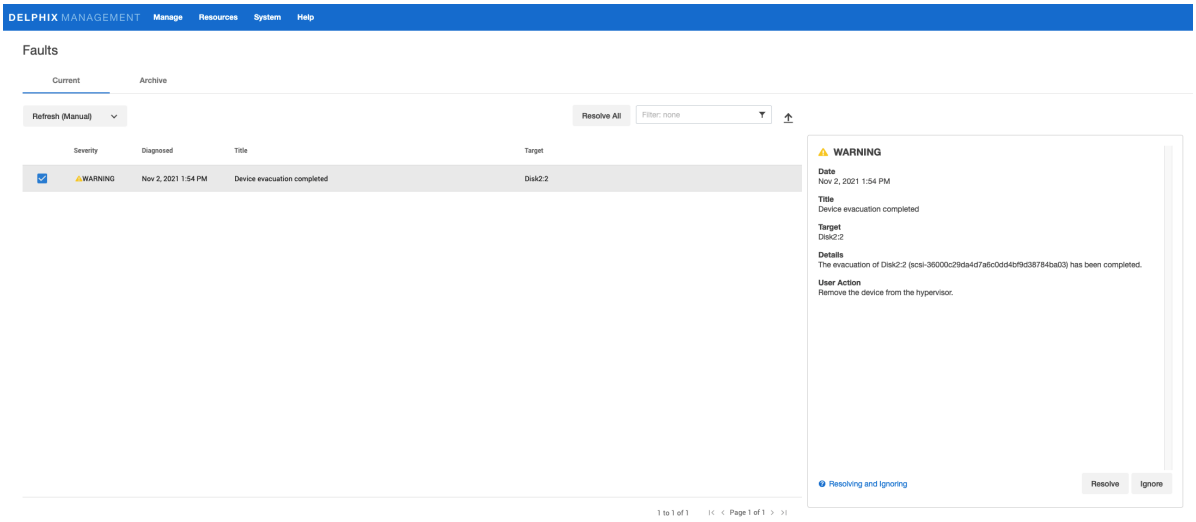
Note : This does not signify that the device migration has been completed. A `STORAGE_DEVICE_REMOVAL` job will start for each device to be removed, which handles the data migration from that disk.

8. Wait for device evacuation to complete. Alternatively, you can cancel the evacuation.

   Note:
   Do not detach the device from the Delphix Engine in your hypervisor until the data evacuation is completed.

   You can monitor the progress of the STORAGE_DEVICE_REMOVAL job in the Management GUI under **System >Jobs**.

9. Once the device evacuation has been completed, the job will finish and a fault will be generated. Detach the disks from your hypervisor and the fault will clear on its own. An example of the fault created is shown below.



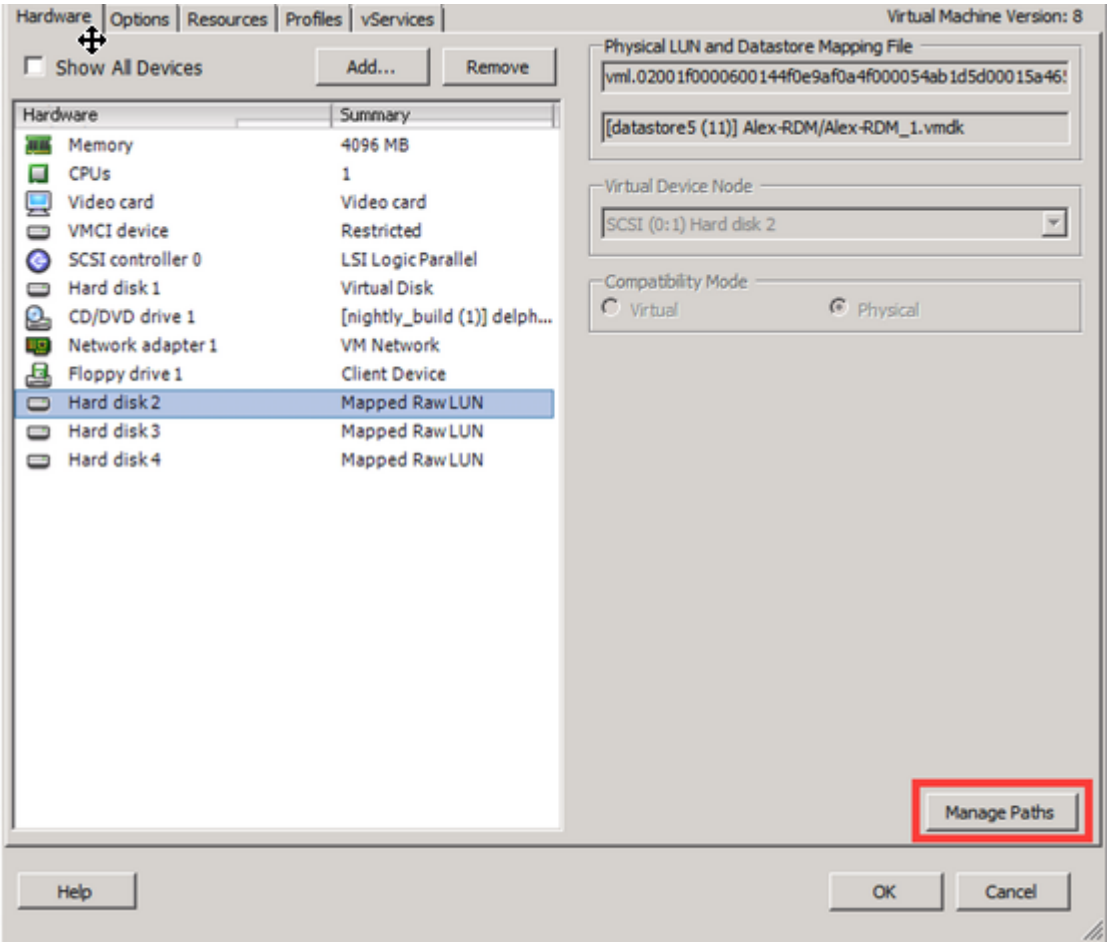
> **Using VMDKs**
>
> When using VMDKs, deleting the wrong VMDK could cause data loss. Therefore, it is highly advisable to detach the device, then verify that the Delphix Engine continues to operate correctly, and lastly delete the VMDK.

6.6.5.6 Getting the UUID of an RDM disk from VMware via the vSphere GUI.

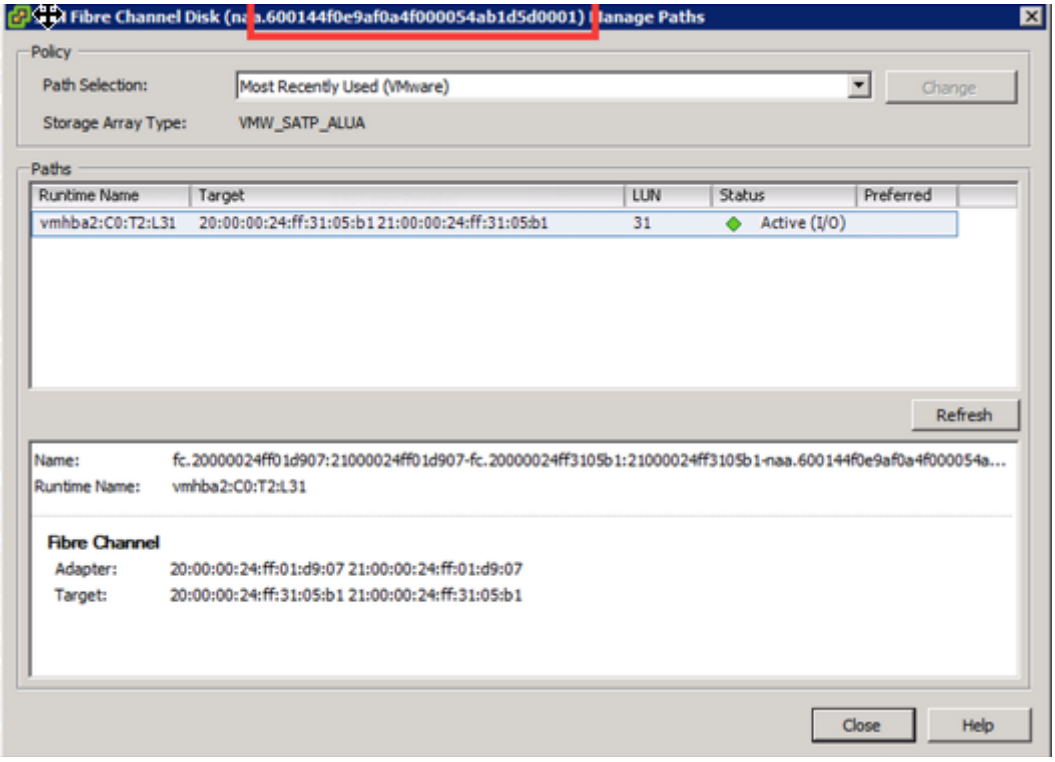
> In the event that the disk serial number displayed in Delphix does not match the UUID in VMware, the Delphix Engine must be powered off and back on in order to make VMware provide the correct values to the guest operating system (Delphix). This has been necessary when using **vmkfstools** with **setuuid**. When forcing the guest OS to re-read the SCSI sense data for the device, VMWare still provides the original values. Even after a simple reboot VMWare still provides the previous UUID values. It was not until the VM was explicitly powered off and back on did VMWare present the new UUID values to the guest. After which the UUIDs matched between the **vmkfstools getuuid** command and the CLI output.

In the ESX graphical user interface (GUI), select your **VM**.

1. Click **Edit settings**.
2. If not already displayed, select the **Hardware** tab.
3. Select the **device** you want to remove.
4. Click **Manage Paths**.



The UUID of the device appears in the title bar, as seen below.



6.6.5.7 Getting the UUID of a VMDK from VMware, via ssh to the ESX server

1. ssh onto the ESX server as the root user.
2. Navigate to the directory containing the .vmdk files for the Delphix VM.

```
Use the 'vmkfstools -J getuuid <.vmdk filename>' command to obtain the UUID,
for example:
```

```
/vmfs/volumes/25894daa-f7b2b044/delphix01-2356 # vmkfstools -J getuuid
delphix01-2356_1.vmdk
UUID is 60 00 C2 91 01 bc 8e 72-31 a4 cd b0 b3 f6 e5 74
```

6.6.5.8 Getting the UUID of a VMDK from VMware, via VMware PowerCLI

```
PS C:\> Connect-VIServer -Server durban -Protocol https -Username root -Password
root_password

Name                           Port  User
----                           ----  ----
durban                         443   root
```

```
PS C:\> Get-VM delphixVM | Get-HardDisk | select name,filename,@{name="UUID";expr={$_
.extensiondata.backing.uuid}}

Name                                        Filename
UUID
----                                        --------
----
Hard disk 1                                 [zfs_delphixVM] dlpx-5.1.9.0-432-61155cf.
.. 6000C294-a115-0327-e417-02560d86e944
Hard disk 2                                 [zfs_delphixVM] dlpx-5.1.9.0-432-61155cf.
.. 6000C299-38fe-5050-1eb2-1ee6db62b257
Hard disk 3                                 [zfs_delphixVM] dlpx-5.1.9.0-432-61155cf.
.. 6000C294-662d-c674-8957-03e0514b7006
Hard disk 4                                 [zfs_delphixVM] dlpx-5.1.9.0-432-61155cf.
.. 6000C29d-0719-1072-0f85-96da2efef4a3

PS C:\> Disconnect-VIServer

Confirm
Are you sure you want to perform this action?
Performing operation "Disconnect VIServer" on Target "User: root, Server: durban,
Port: 443".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "Y")
: Y
```

6.6.5.9 In-place block-to-object storage migration

6.6.5.9.1 Introduction

The Elastic Data product uses object storage to provide a cost-effective and flexible medium for storing an engine's data without compromising performance for most workloads. From version 15.0.0.0 and later, it is possible for a sysadmin to migrate an engine's storage pool residing in local block storage to object storage backends, in-place and with no downtime.

> If your kernel version is below 15.0, you cannot perform an in-place block-to-object storage migration. To enable this feature, you must first run a full upgrade to version 15.0 or higher.

6.6.5.9.2 Prerequisites

The migration process requires:

- An object storage endpoint with access credentials where the data from the block-based storage pool will be migrated to. The object storage backend needs to be supported by the Continuous Data Engine (e.g. AWS S3, Azure Blob, OCI Object Storage, and various on-prem solutions).
- One or more high performance block device(s), to be used as an object storage cache during and after the migration is finalized. Disks that are currently configured in your Continuous Data Engine cannot be used for this purpose.

6.6.5.9.3 Limitations

In-place migration is an irreversible operation. Once started, it cannot be stopped, nor paused, nor reversed.

Additionally, during the migration process, sysadmins are not able to add or remove block devices for data storage. Cache device removal is not allowed either while the migration is in progress. Cache device additions are allowed at any time (e.g. to improve performance during or after the migration).

6.6.5.9.4 Other methods of migration

An alternative to this feature would be replicating data to a new engine that was created with an object storage-backed pool from the start. This approach has some potential drawbacks:

- **This migration path is not viable for engines that already use replication.**
- **The process is complex and potentially error-prone**. The sysadmin would need to: Create a new VM, start the replication, notify their users, update infrastructure to use the new VM, and finally delete the old VM.
- **Infrastructure requirements temporarily double.** While transitioning, the sysadmin needs to use double the infrastructure that would normally be needed. This can be a tough requirement to meet when multiple product engines need to be migrated.
- **There is downtime involved.** If the write-load of the transitioning VMs is high, sysadmins may need to plan some downtime for finalizing replication and switching users over to the new VM.
- **The process can be time-consuming.** Multi-TB storage pools may take too long to migrate, as the network is involved in two places: replication and writes to the object storage. An in-place migration only involves the network when writing to the object storage.

6.6.5.9.5 Instructions

> The following example walks through a migration in AWS for a storage pool migrating from EBS Volumes to S3. Please adjust the steps for your cloud platform accordingly.

1. Add one or more high-performance block device(s) to your engine from the cloud console or hypervisor to be used as a cache. In this example, two gp3 EBS Volumes have been added to a demo VM.
2. Login to the Delphix CLI as a **sysadmin** user.

3. Navigate to the storage/device directory with `cd storage/device`. Type `ls` to list all devices currently detected by the VM and find the name of the block device you just added.

```
ip-10-110-196-27> cd storage/device
ip-10-110-196-27 storage device> ls
Objects
NAME  BOOTDEVICE  CONFIGURED  SIZE  EXPANDABLESIZE  FRAGMENTATION  ALLOCATING
xvda  true   true    70GB  0B   NA     true
xvdb  false  true   100GB  0B   54%    true
xvdc  false  true   100GB  0B   58%    true
xvdd  -      false   50GB   -     -                 -
xvde  -      false   50GB   -     -                 -
```

In the above sample, the devices just added stands out because they are the only non-configured ones; `xvdd` and `xvde`.

4. Navigate to the /storage/migrate directory with `cd /storage/migrate/`. Type `start` to start the migration API form and (optionally) `ls` to list all fields that are required for submitting the migration job.

```
ip-10-110-196-27 storage device> cd /storage/migrate/
ip-10-110-196-27 storage migrate> start
ip-10-110-196-27 storage migrate start *> ls
Properties
    type: BlobObjectStore
    accessCredentials: (unset)
    cacheDevices: (unset)
    configured: false
    container: (unset)
    endpoint: (unset)
    size: (unset)
```

5. Set the correct `type` of object storage (in this case `S3ObjectStore`) with the `set` command:

```
ip-10-110-196-27 storage migrate start *> set type=S3ObjectStore
```

6. Set the access credential type and info (this can vary by hypervisor/cloud provider).
   a. Using AWS instance profile authentication is simple and can be done with a single command:

   ```
   ip-10-110-196-27 storage migrate start *> set
   accessCredentials.type=S3ObjectStoreAccessInstanceProfile
   ```

   b. If access key credentials are preferred, then the access ID and its respective key should be specified:

```
ip-10-110-196-27 storage migrate start *> set
accessCredentials.type=S3ObjectStoreAccessKey
ip-10-110-196-27 storage migrate start *> set
accessCredentials.accessId=<redacted>
ip-10-110-196-27 storage migrate start *> set
accessCredentials.accessKey=<redacted>
```

7. Input the object storage's endpoint and related info (the fields may vary based on the hypervisor/cloud provider object storage that you want to connect to):

```
ip-10-110-196-27 storage migrate start *> set endpoint=https://s3-us-
west-2.amazonaws.com
ip-10-110-196-27 storage migrate start *> set region=us-west-2
ip-10-110-196-27 storage migrate start *> set bucket=delphix-data
ip-10-110-196-27 storage migrate start *> set size=500G
```

The `size` field is the storage quota that the sysadmin can use to limit the engine's usage of the object storage backend. The sysadmin can change that value post-migration as needed if their storage requirements change.

8. Finally, set the cache to the new devices that were added in step one (see name of devices in step three):

```
ip-10-110-196-27 storage migrate start *> set cacheDevices=xvdd,xvde
```

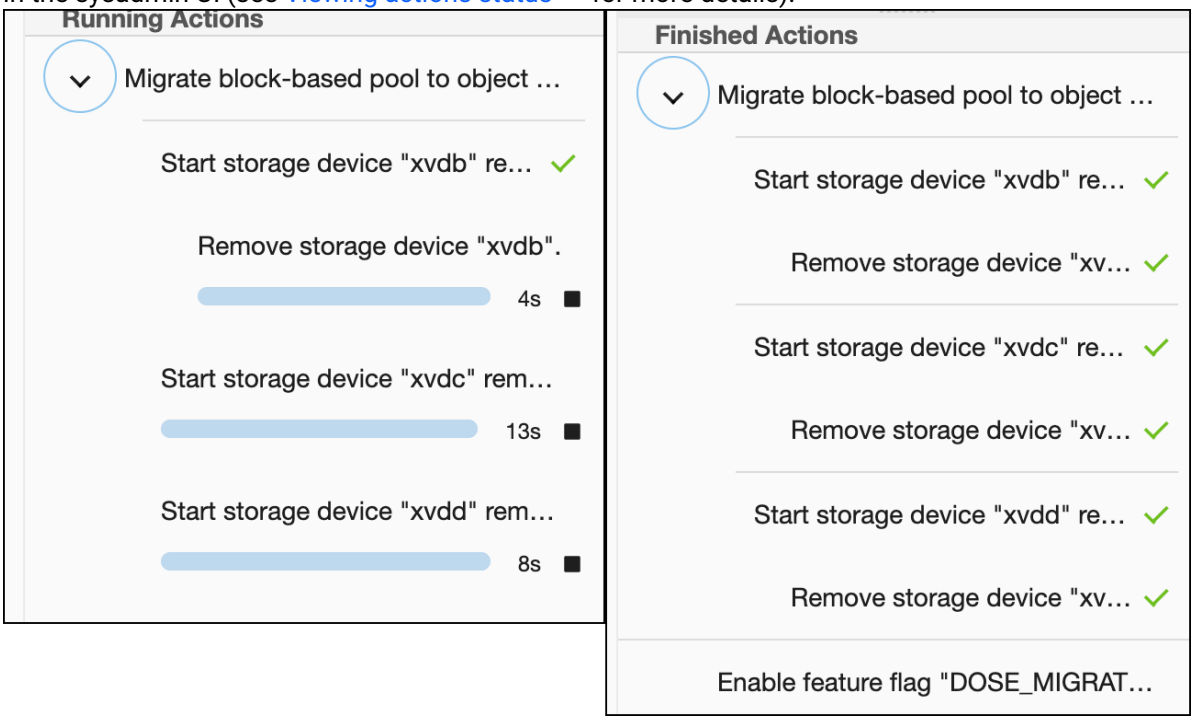
9. (Optional) Type `ls` once more to verify that all your info is entered correctly:

```
ip-10-110-196-27 storage migrate start *> ls
Properties
    type: S3ObjectStore (*)
    accessCredentials:
        type: S3ObjectStoreAccessKey (*)
        accessId: ******* (*)
        accessKey: ******** (*)
    bucket: delphix-data (*)
    cacheDevices: xvdd, xvde (*)
    configured: false
    endpoint: https://s3-us-west-2.amazonaws.com (*)
    region: us-west-2 (*)
    size: 50GB (*)
```

10. Type `commit` to submit the migration job (warning, this is an irreversible operation - see Limitations section). If there is something wrong with the credentials, the connectivity of the VM, or any of the above parameters that you typed above, the command will fail and give a helpful message. Otherwise, if everything is correct, you will see the message below:

```
ip-10-110-196-27 storage migrate start *> commit
    Dispatched job JOB-23
    OBJECT_STORE_MIGRATE job started for "unknown".
    OBJECT_STORE_MIGRATE job for "unknown" completed successfully.
```

The above means that the job submission was successful and it is running in the background. There are a couple of ways to track the job's progress. The easiest way is viewing the Actions Panel in the sysadmin UI (see Viewing actions status[242] for more details).



The other way is through the CLI by typing `cd /storage/migrate` and then running `ls`:

```
ip-10-110-228-202 storage migrate> ls
Properties
    type: ObjectStoreMigrationStatus
    mappingMemory: 0B
    startTime: 2023-08-25T23:48:03.077Z
    state: ACTIVE
    total: 20.13MB
```

242 https://cd.delphix.com/docs/latest/viewing-action-status

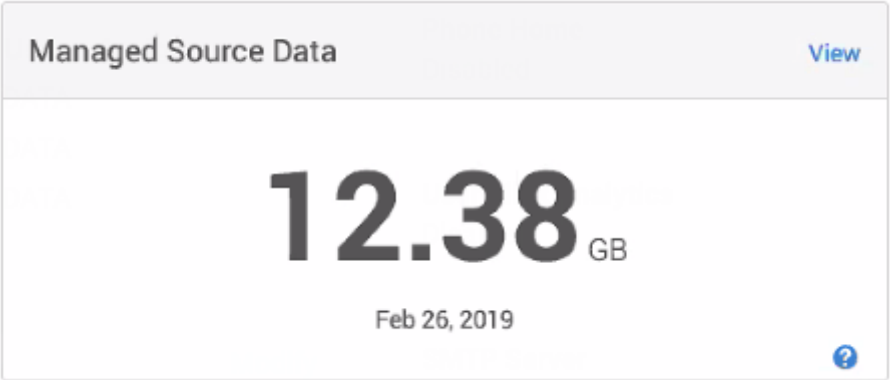
6.6.6 Managing source data

The **Managed source Data** dashboard tile provides an aggregate sum of the size of ingested source databases. This helps to understand the total amount of data per-engine that is managed by Delphix.

Managed Source Data refers to the size of the source data that is ingested and managed by the Delphix Engine. It is the physical, allocated size of the source database.

Today, the value displayed represents the source data size for Oracle, SQL Server, and SAP ASE databases. Since this value is a measure of source size, it is not affected by Delphix constructs or operations, e.g. the sizes of dSources, VDBs, snapshots. Since the intent is to measure the size of the sources, data from replication or Selective Data Distribution are not included in the sum for the engine. The total usage is displayed in gigabytes.

When an admin (Delphix Management Application) or sysadmin (Delphix Setup) login to the Delphix Engine they will have access to the Managed Source Data tile (shown below) from their dashboard.

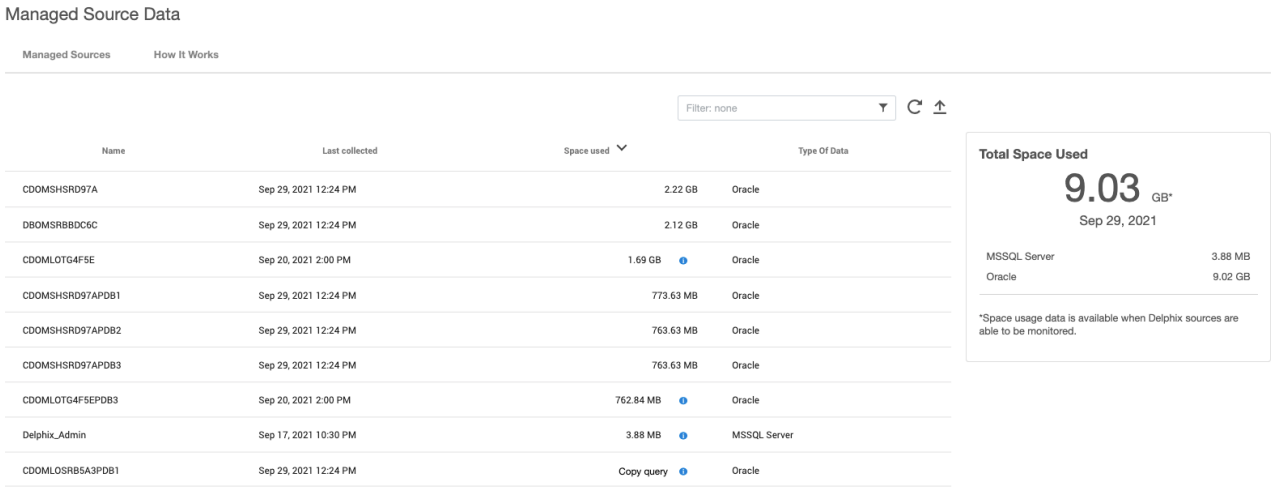


Selecting the **View** link located on the top-right takes users to the Managed Source Data page. This page provides a detailed breakdown of the total usage and can also be reached via **Resources** > **Managed Source Data** from the navigation bar.

Blue question marks indicate:

- A “fallback” value is being used in the sum: A fallback value is used when sources are unavailable at the time when the usage query is made (e.g., the source is disabled), a “fallback” value is used in the sum. The fallback value refers to the last usage value Delphix collected while connected to the source.
- Sources that are not included in the sum: These are source types for which Delphix does not currently collect data (for example, sources like EBS, DB2, HANA, PostgreSQL).

Hovering over the **blue question marks**, a tooltip appears to describe the exact situation (whether or not Delphix used historical data/fallback values or if the source for collection is unsupported).



To search the **Managed source Data** table, enter the name of the data source you are looking for. The grid will refresh to display the selected data source. Select to refresh data to get updated space usage information. To export the information provided in the grid to a .csv file select.

> **Backend data refresh**
>
> The backend data refreshes at a regular interval but does not trigger a UI refresh.

6.6.7 An overview of held space

6.6.7.1 Scenario description

The existence of a Held Space is not indicative of an issue but rather a representation of an underlying storage dependency.

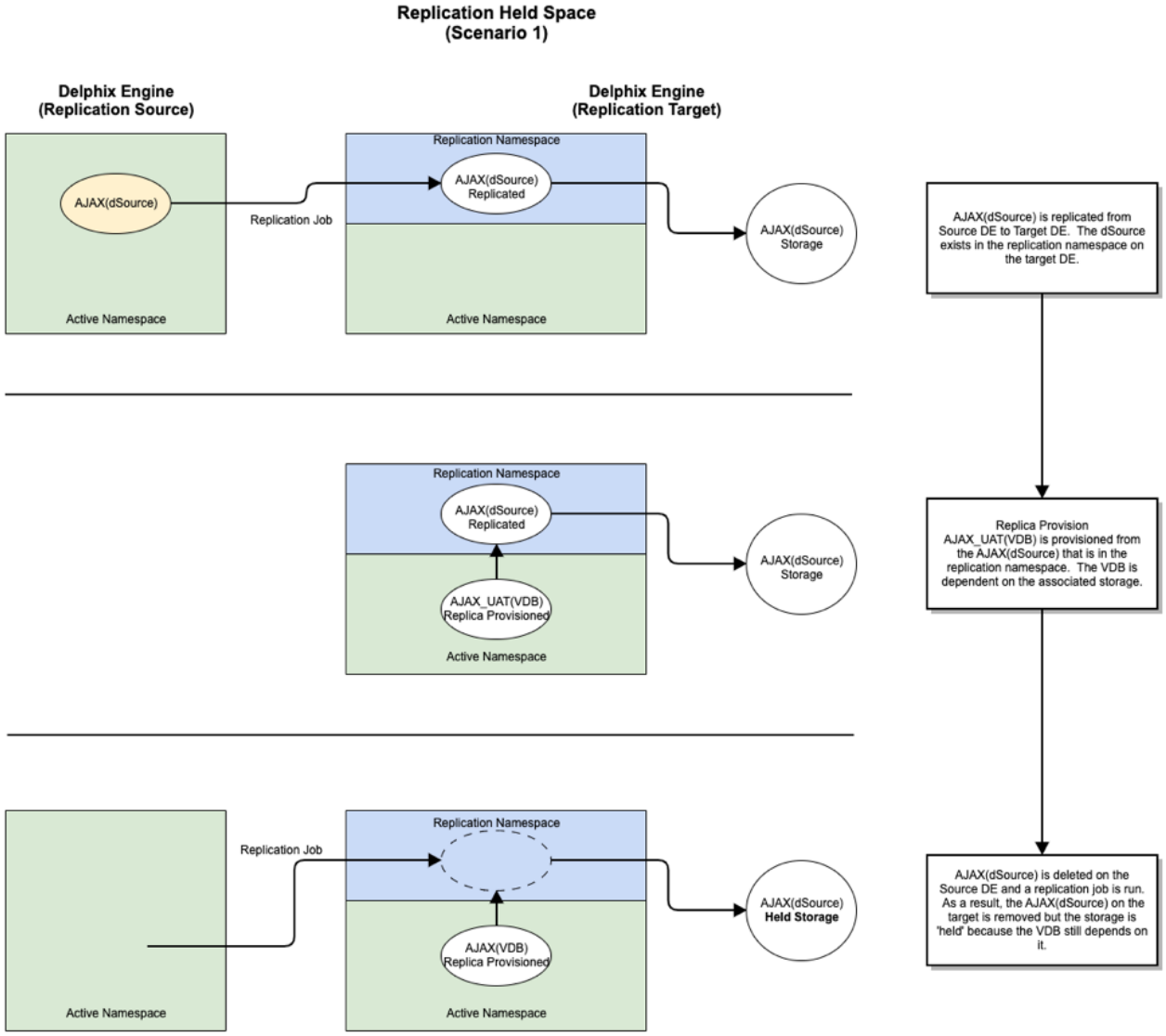
There are four scenarios that can create Held Space.

6.6.7.1.1 Scenario 1:

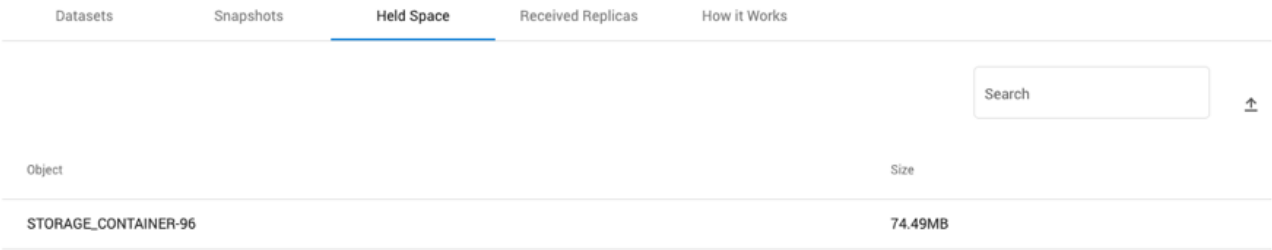
The first scenario can occur when you replicate a dSource to a second engine, which we'll call a Replication Target.

If you then provision a VDB from the Replication Target, and the dSource is subsequently deleted from the Replication Source, you will create a Held Space when that delete replicates to the Replication Target engine.

When this happens, the deleted dSource is removed from the target, but its storage remains held because it is needed by the replica provisioned VDB.



Held Space can be viewed in the **Storage Capacity** screen by selecting the Held Space tab.



6.6.7.1.2 Scenario 2:

The second scenario can occur in the context of a snapshot that is in use by a Delphix Self-Service branch or bookmark.

The snapshot may be removed via policy, but until the branches and/or bookmarks that leverage that snapshot are removed, its space will be held.

6.6.7.1.3 Scenario 3:

The third scenario can occur during Selective Data Distribution where a masked VDB that is provisioned from a dSource is replicated to a Replication Target.

The metadata objects relating to the dSource could contain sensitive information like hostnames, user names, and passwords. Hence they are not replicated to the Replication Target. The data belonging to the dSource is replicated after redacting any sensitive information. This redacted data shows up as held space.

6.6.7.1.4 Scenario 4:

The fourth scenario is caused by a failed or interrupted Replication job and results in held space on the Replication source engine. During replication, all data blocks to be replicated are serialized for transfer. Once the transfer is complete, the serialized data is released.

However, if a Replication job fails, the serialized data is retained until the next successful Replication job execution, or the Replication Profile is deleted - whichever happens first.

In a scenario where dSource and VDB snapshots in the serialized data reach the Retention Policy limit before the Replication job successfully executes, that snapshot data will become held space. This is most likely to happen if either of the following is true:

- A failed Replication job goes a long time before being successfully re-run.
- Replication schedules are configured with a period that is longer than one or more Retention Policies.

6.7 Monitoring and log management

Using both Delphix and its associated datasets will generate many types of logs. Monitoring these logs are a key component of good diagnosability. Here, we explain the types of logs that Delphix creates as well as the monitoring tool integrations we support, such as SNMP and Splunk.

This section covers the following topics:

- Configuring SNMP (see page 785)
- Viewing action status (see page 794)
- Viewing jobs (see page 796)
- System faults (see page 798)
- Accessing audit logs (see page 804)
- Creating support logs (see page 807)
- Setting support access control (see page 599)
- Setting syslog preferences (see page 810)
- Support access audit logs (see page 812)
- Diagnosing connectivity errors (see page 815)
- Email (SMTP) alert notifications (see page 817)
- Splunk integration (see page 835)
- Fluentd plugin service for API modules (see page 826)

6.7.1 Configuring SNMP

This topic describes how to configure SNMP.

Starting with version 6.0.6.0 the Delphix Engine adds support for SNMP version 3 by implementing the User Security Model (USM) of the SNMP version 3 (SNMPv3) specification. Version 3 of SNMP adds stronger security compared to version 2.

> When SNMP v2 is configured and enabled, SNMP v1 is also configured and enabled. Similarly, v2 is configured and enabled with v3. This is a known behavior of the net-snmp package used to implement SNMP on the engine.

USM replaces the community string as in SNMP version 2 with user records. The Delphix engine can both receive and send SNMP version 3 messages only by using the USM security model. The USM model provides more security over version 2 by hashing passwords and encrypting the payload.

The Delphix Engine includes an SNMPv3 agent that is only capable of responding to read-only operations, such as GET messages using the User Security Model. USM users can be configured via the GUI, CLI, or API. Each user is composed of a username, authentication password, authentication protocol, privacy password, and a privacy protocol.

The Delphix Engine is also capable of sending alerts and faults over SNMP with the User Security Model. Messages are sent identically as with SNMP version 2 except that the USM model is used to authenticate and encrypt the alert message against a customer's manager. Users will need to properly configure their managers to receive messages. The use of the test function can be used to test configuration.

6.7.1.1 Prerequisites

There are no prerequisites for enabling SNMP to provide system status. The following are prerequisites for sending alerts to an external SNMP manager.

- At least one SNMP manager must be available and must be configured to accept SNMPv2 InformRequest notifications.
- Delphix's MIB (Management Information Base) files must be installed on the SNMP manager or managers. These MIB files describe the information that the Delphix Engine will send out. They are attached to this topic.

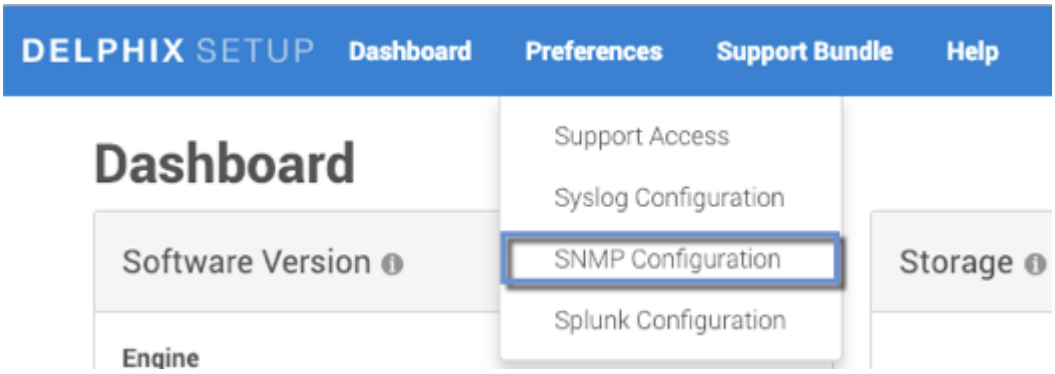
File name	Content-type	File Size
DELPHIX-ALERT-MIB.txt [243]	text/plain	5 kB
DELPHIX-MIB.txt [244]	text/plain	0.8 kB

6.7.1.2 Configuring SNMP for v2

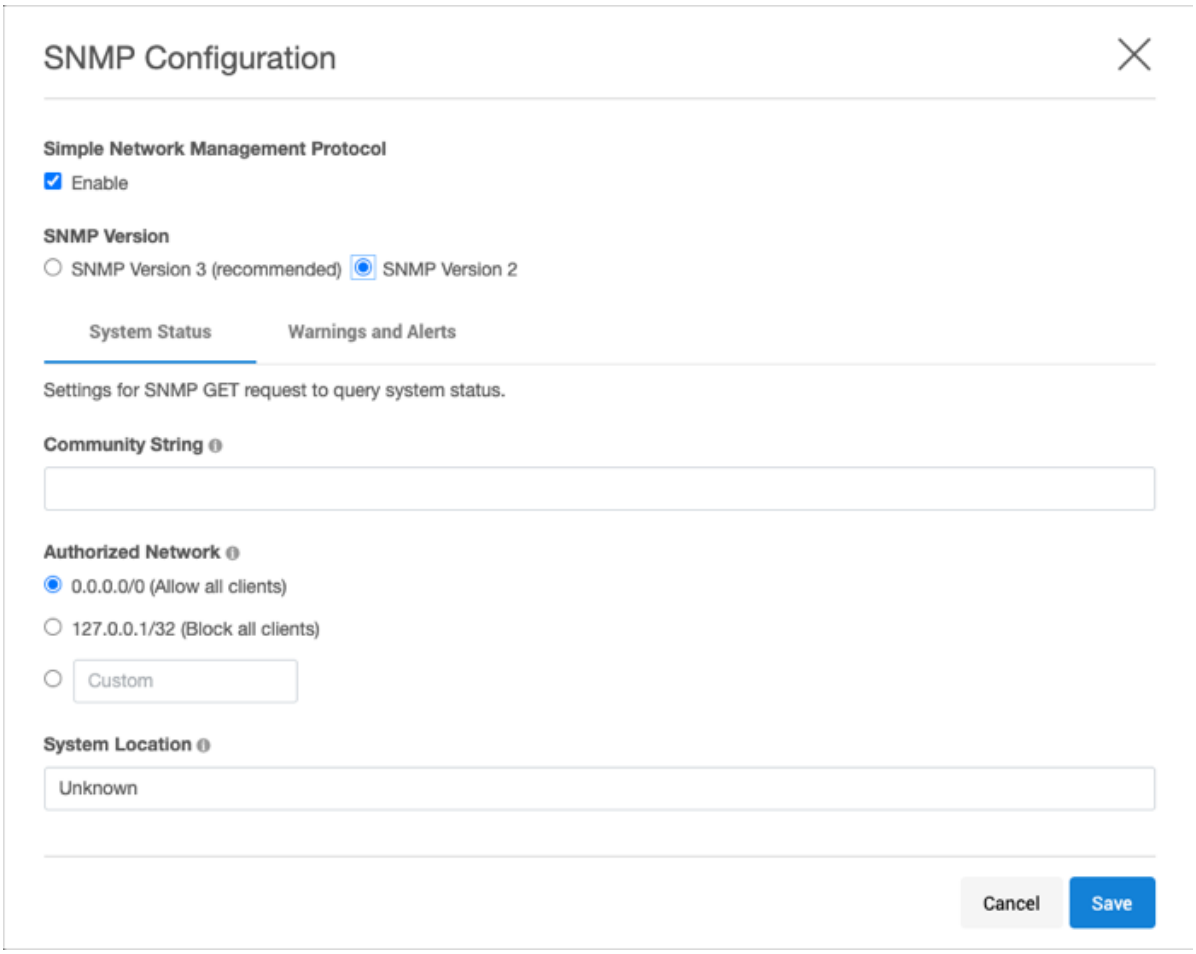
1. On the Delphix Engine login screen, select **Delphix Setup**.
2. In the Delphix Setup login screen, enter the **sysadmin** username and password.
3. Click **Log In**.

243 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149496711?preview=%2F149496711%2F149496726%2FDELPHIX-ALERT-MIB.txt
244 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149496711?preview=%2F149496711%2F149496729%2FDELPHIX-MIB.txt

4. From the **Delphix Setup**, select **Preferences > SNMP Configuration.**



5. Select **Enable**.
6. Select **SNMP Version 2**.



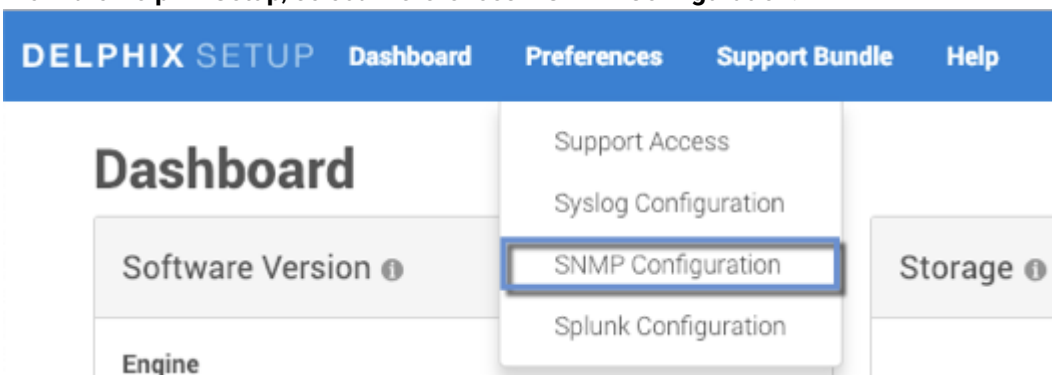
7. In **Community string**, enter the community string. It is the string that SNMP clients must provide in order to be authorized to retrieve SNMP information from the Delphix Engine.
8. OPTIONAL: change the following settings:

a. **Authorized network** – The set of client IP addresses (in CIDR notation) authorized to retrieve SNMP information from the Delphix Engine. To allow all clients, set this to 0.0.0.0/0 (the default). To prevent all clients from connecting, set this to the loopback address, 127.0.0.1/32.
b. **System location** – A free-form text description of the Delphix Engine's physical location. This is provided as the value for MIB-II OID.

9. Click **Save** to commit the SNMP configuration.

6.7.1.3 Configuring SNMP for v3

1. On the Delphix Engine login screen, select **Delphix Setup**.
2. In the Delphix Setup login screen, enter the **sysadmin** username and password.
3. Click **Log In**.
4. From the **Delphix Setup**, select **Preferences > SNMP Configuration**.

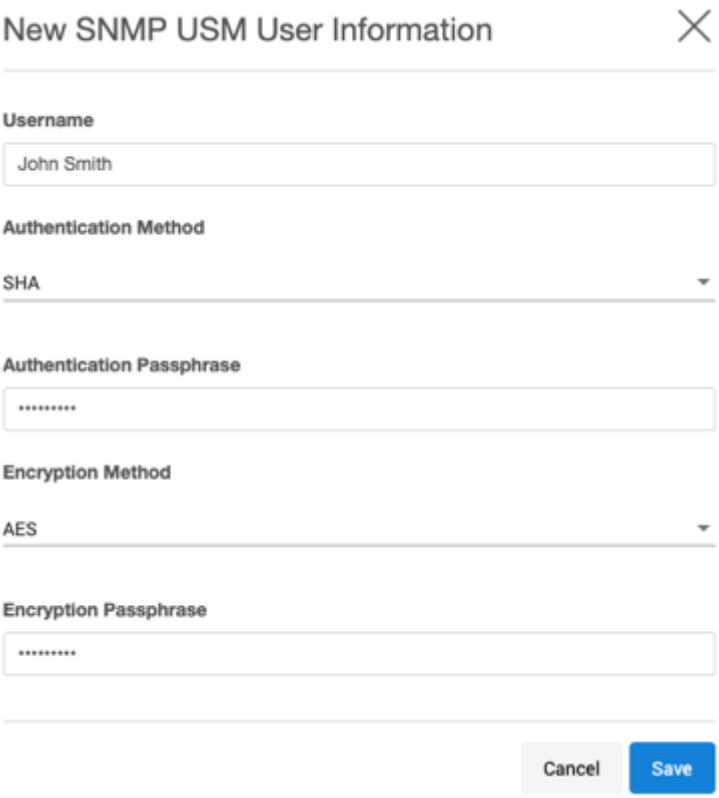


5. Select **Enable**.
6. Select **SNMP Version 3** (recommended).
7. Click **Save** to commit the SNMP configuration.

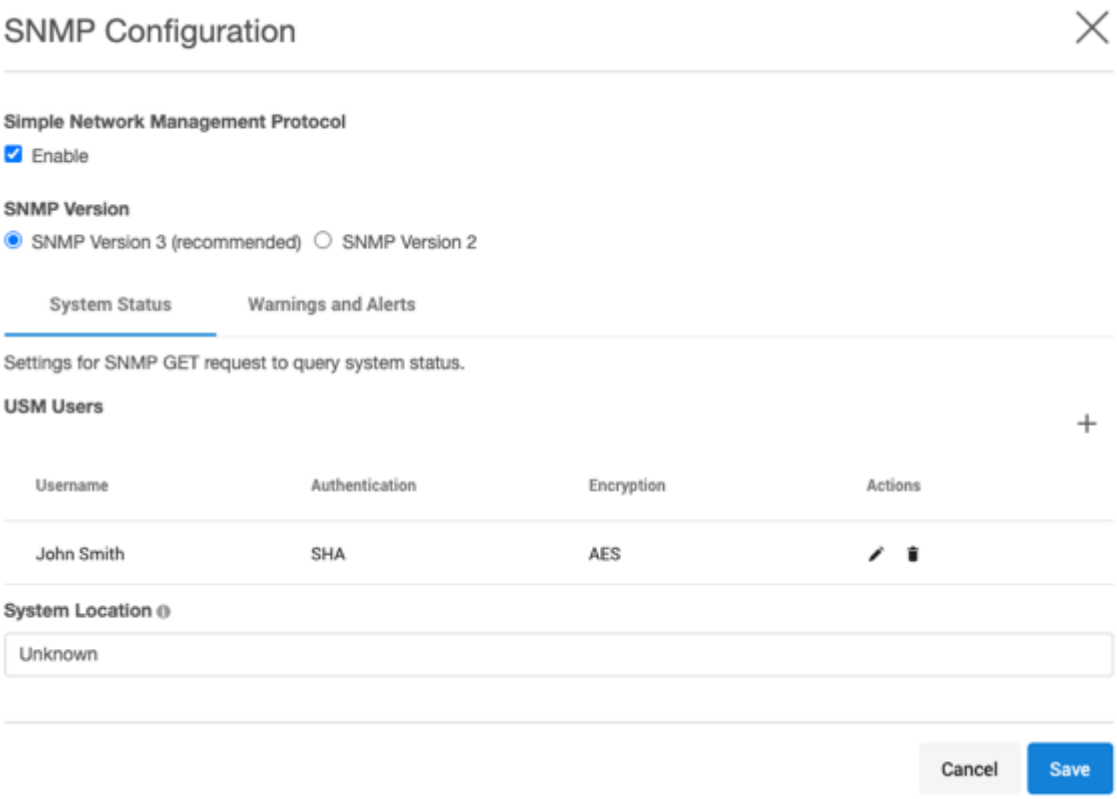
6.7.1.3.1 Configure SNMP version 3 agent

After enabling SNMP and choosing version 3, the SNMP agent can be configured with the User Security Model (USM) users.

1. Click on **System Status**.
2. Click on the **+** to show the **New SNMP USM User Information** dialog.



3. Provide the following information for the user:
   a. **Username -** A string representing the name of the user.
   b. **Authentication method -** An indication of whether messages sent on behalf of this user can be authenticated, and if so, the type of authentication protocol that is used. Users can select from:
      MD5 -the HMAC-MD5-96 authentication protocol.
      SHA -the HMAC-SHA-96 authentication protocol.
   c. **Authentication passphrase -** If messages sent on behalf of this user can be authenticated, this key will be used with the authentication key for use with the authentication. This field requires a minimum of 8 characters.
   d. **Encryption method -** Provides support for data confidentiality. The designated portion of an SNMP message is encrypted and included as part of the message sent to the recipient. Users can select from
      AES - Advanced Encryption Standard
      DES - Data Encryption Standard
   e. **Encryption passphrase** - If messages sent on behalf of this user can be encrypted, this key will be used with the encryption method for use with the authentication. This field requires a minimum of 8 characters.
4. Click **Save** to add your user.



6.7.1.3.2 Configuring SNMP version 3 managers

1. Click on the **Warnings and Alerts** tab.
2. Choose the **Severity** of the alerts that are sent via SNMP. The severity is ranked in order from most restrictive to least restrictive: critical, warning, and informational. Setting the severity critical will only include critical alerts while setting to informational will cause an informational, warning and critical alerts to be sent over SNMP to all managers.
3. Click **+** to add a manager to show the **New SNMP manager information** dialog**.**

New SNMP Manager Information

Username

Host Address

Port

Authentication Method

None

Authentication Passphrase

Encryption Method

None

Encryption Passphrase

☐ Use INFORM instead of TRAP.

Cancel Save

4. Provide the following information:
    a. **Username -** A string representing the name of the user.
    b. **Host address** - Host or IP address.
    c. **Port** - Port used by the SNMP Engine.
    d. **Authentication method -** An indication of whether messages sent on behalf of this user can be authenticated, and if so, the type of authentication protocol that is used. Users can select from:
    MD5 -the HMAC-MD5-96 authentication protocol.
    SHA -the HMAC-SHA-96 authentication protocol.
    e. **Authentication passphrase -** If messages sent on behalf of this user can be authenticated, this key will be used with the authentication key for use with the authentication. This field requires a minimum of 8 characters.
    f. **Encryption method -** Provides support for data confidentiality. The designated portion of an SNMP message is encrypted and included as part of the message sent to the recipient. Users can select from
    AES - Advanced Encryption Standard
    DES - Data Encryption Standard
    g. **Encryption passphrase** - If messages sent on behalf of this user can be encrypted, this key will be used with the encryption method for use with the authentication. This field requires a minimum of 8 characters.
5. By default, TRAP messages are sent and require the agent's authoritative engine id for the user to be configured with the SNMP engine ID from the Delphix Engine, which can be obtained via the CLI. Click Use **Inform instead of Trap** to use INFORM messages, which are more reliable and require less configuration at the expense of higher network traffic.
6. Click **Save**.

6.7.1.4 CLI: viewing SNMP engine ID

Viewing SNMP Engine ID

```
> service snmp v3
service snmp v3> ls
Properties
    type: SNMPV3Config
    enabled: true
    engineId: 0x80001f88801ad80b5e62a7f85f00000000
    location: Unknown
    securityModel: USM
    severity: WARNING

Children
manager
usm

Operations
```

```
update
```

6.7.1.5 Supported MIBs

The Delphix software can be configured to send SNMP traps when Delphix alerts are generated as described in the procedure above. In order to process these traps in your SNMP manager software, you will need the base Delphix MIB and the Delphix Alert MIB.

In addition to generating traps, the Delphix Engine supports read-only access to the following MIBs for basic system monitoring purposes.

- The following MIB-II object hierarchies are defined in RFC 1213[245]:
  - system (OID .1.3.6.1.2.1.1): Provides basic system identity information
  - interfaces (OID .1.3.6.1.2.1.2): Provides network interface information including I/O statistics
  - IP (OID .1.3.6.1.2.1.4): IP protocol information including IP addresses configured, routes, and IP statistics
- The following UCD objects defined in http://www.net-snmp.org/docs/mibs/ucdavis.html
  - memory usage (OID .1.3.6.1.4.1.2021.4)
  - CPU usage (OID .1.3.6.1.4.1.2021.11)
  - Disk I/O statistics (OID .1.3.6.1.4.1.2021.13.15)

6.7.1.6 Examples

The following examples assume that you have enabled SNMP on a Delphix Engine named example.company.com, and have set the community string to "public".

1. Walking the MIB-II objects using the net-snmp snmpwalk tool:

```
$ snmpwalk -v 2c -c public example.company.com
SNMPv2-MIB::sysDescr.0 = STRING: Delphix Engine 5.1.6.0 DelphixOS 5.1.2017.03.2
4
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.41028
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (476432) 1:19:24.32
SNMPv2-MIB::sysContact.0 = STRING: administrator@company.com
SNMPv2-MIB::sysName.0 = STRING: example.company.com
SNMPv2-MIB::sysLocation.0 = STRING: VM Host
...
```

2. Walking Disk read and write I/O statistics:

```
$ snmpwalk  -v 2c -c public example.company.com .1.3.6.1.4.1.2021.13.15.1.1.12
UCD-DISKIO-MIB::diskIONReadX.1 = Counter64: 11310593921
UCD-DISKIO-MIB::diskIONReadX.2 = Counter64: 334
```

245 https://tools.ietf.org/html/rfc1213

```
UCD-DISKIO-MIB::diskIONReadX.3 = Counter64: 334
UCD-DISKIO-MIB::diskIONReadX.4 = Counter64: 865912605
UCD-DISKIO-MIB::diskIONReadX.5 = Counter64: 867599133
UCD-DISKIO-MIB::diskIONReadX.6 = Counter64: 865339677
UCD-DISKIO-MIB::diskIONReadX.7 = Counter64: 11309258752
UCD-DISKIO-MIB::diskIONReadX.8 = Counter64: 0
UCD-DISKIO-MIB::diskIONReadX.9 = Counter64: 1822880256
seb-laptop:~$ snmpwalk  -v 2c -c public example.company.com .1.3.6.1.4.1.2021.1
3.15.1.1.13
UCD-DISKIO-MIB::diskIONWrittenX.1 = Counter64: 22337830400
UCD-DISKIO-MIB::diskIONWrittenX.2 = Counter64: 0
UCD-DISKIO-MIB::diskIONWrittenX.3 = Counter64: 0
UCD-DISKIO-MIB::diskIONWrittenX.4 = Counter64: 45118203392
UCD-DISKIO-MIB::diskIONWrittenX.5 = Counter64: 45137660928
UCD-DISKIO-MIB::diskIONWrittenX.6 = Counter64: 45139064320
UCD-DISKIO-MIB::diskIONWrittenX.7 = Counter64: 22337830400
UCD-DISKIO-MIB::diskIONWrittenX.8 = Counter64: 0
UCD-DISKIO-MIB::diskIONWrittenX.9 = Counter64: 33023515648
```

3. Retrieving the system uptime:

```
$ snmpget -v 2c -c public example.company.com .1.3.6.1.2.1.1.3.0
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (1453172) 4:02:11.72
```

6.7.2 Viewing action status

This page describes how to view the status of actions for the Data Engine.

To view the status of actions that are currently running on the Data Engine, open the **Action sidebar**. To view details of currently-running and completed jobs, open the **Dashboard**.

6.7.2.1 Action sidebar procedure

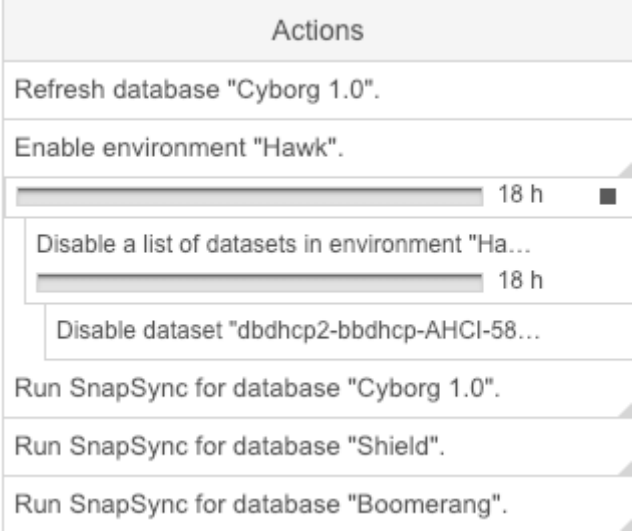
1. Login to the **Delphix Management** or **Delphix Setup** application.
   a. Depending on the width of the window, the **Action sidebar** may be automatically displayed on the right of the screen.
2. To see the **Action sidebar**, click **Actions** on the top navigation bar.

6.7.2.2 Description

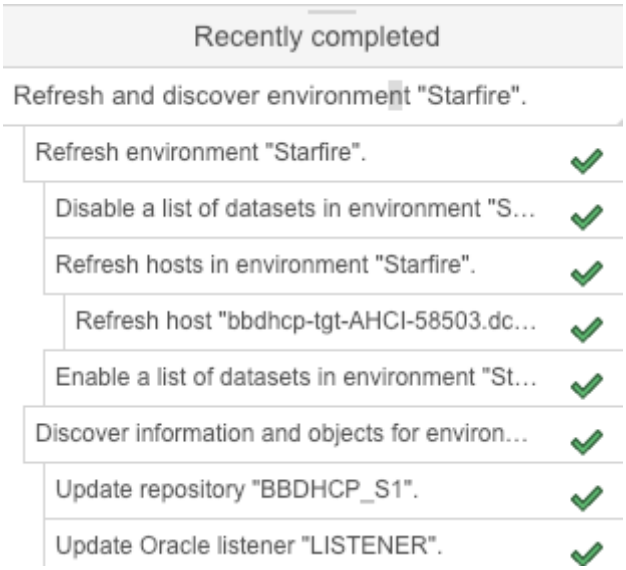
The **Action sidebar** consists of two sections. The top section lists actions that are currently running on the Data Engine. The bottom section, labeled **Recently completed**, contains actions that have recently been completed.

Each action is initially collapsed and only presents the title of the action. Click an action to expand it and see more details such as progress, elapsed time, and a description of the operation in progress.

The following is an example of the **Action sidebar** when an Enable action is running.



When an action has been completed it will move down to the bottom of the panel under **Recently completed**, as shown in the screenshot below with a Refresh action.

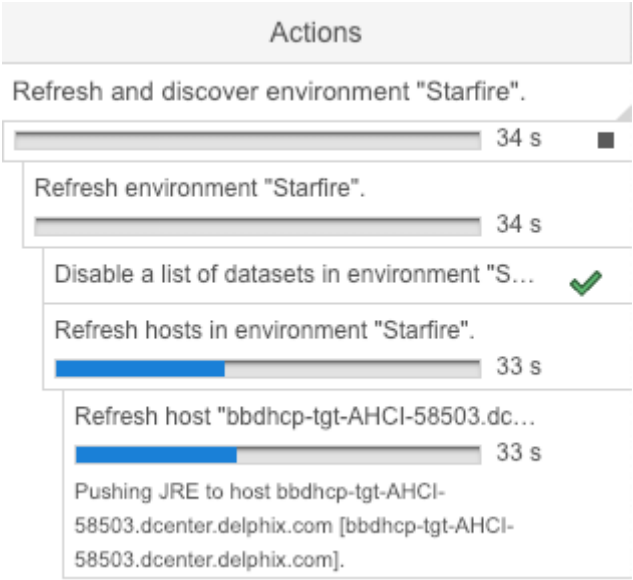


If you are a Delphix Admin or a System Admin, you will be able to see all the actions of your respective application. If you are not an admin user, you will only see actions you have permission to see.

6.7.2.3 Sub-action

Each action may contain one or several sub-actions which represent the execution of a subset of the action itself. Click an action to see its sub-actions and their respective details. Note that the list of sub-actions is created dynamically during the execution of the action.

The following is an example of an **Environment Refresh** action and its sub-actions.

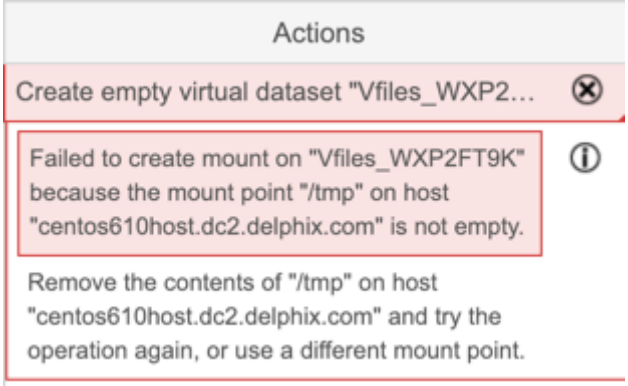


6.7.2.4 Errors

When an error condition occurs during the execution of an action, the background color of the action box becomes red, and the action remains in the top section until you dismiss it.

- Click the **action** title to expand it.
  - The action will expand to display a description of the error, suggestions to resolve it, and sometimes the raw output of command execution.
- To dismiss the action, click the **X** next to the action displaying an error.

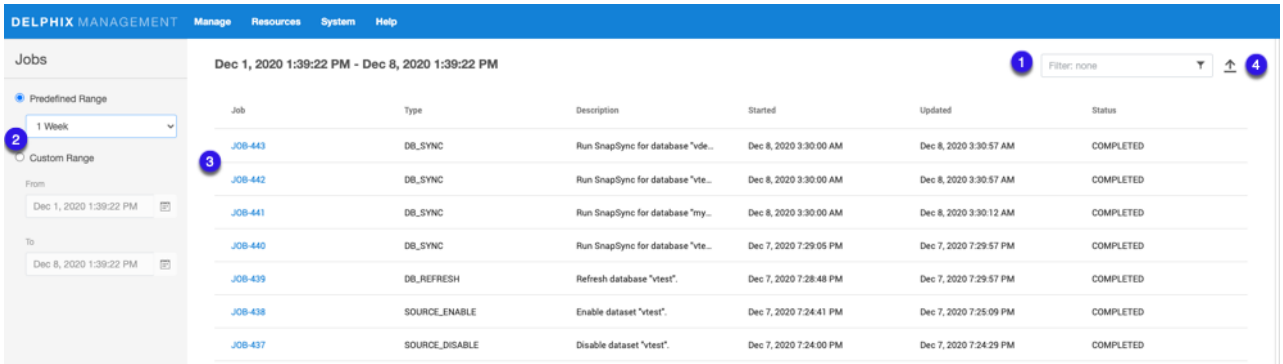
The following is an example of an action failure shown in the **Action sidebar**.



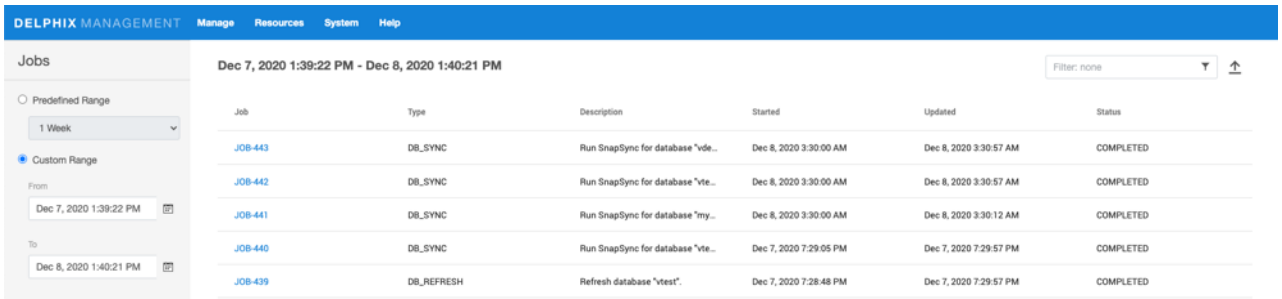
6.7.3 Viewing jobs

Login to the **Delphix management** application using **admin** credentials. From the **System** menu select **Jobs**.

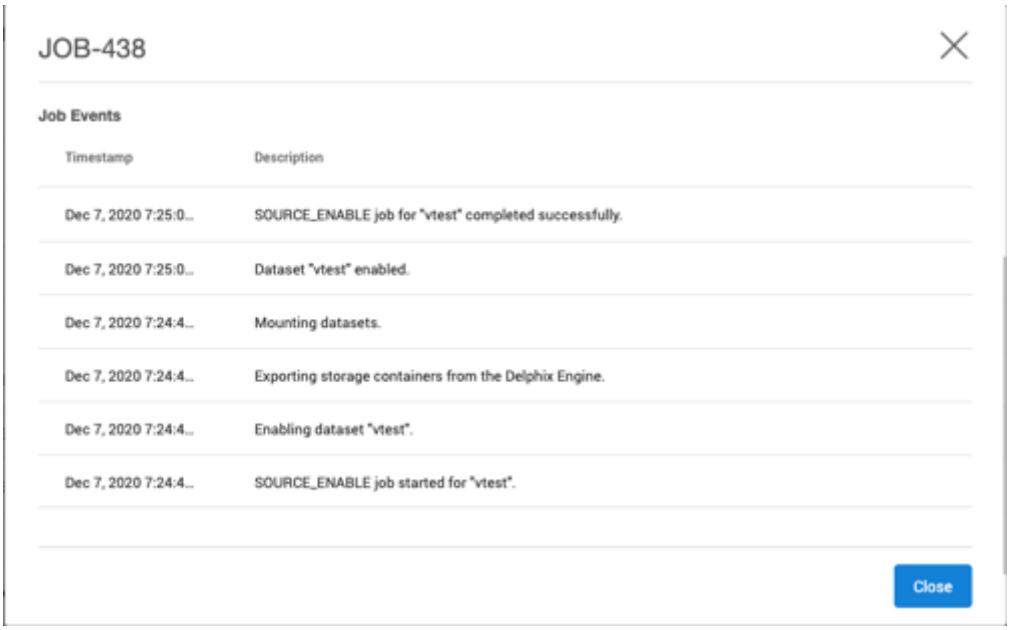
The **Jobs** panel displays all jobs that have been initiated by the Data Engine, in addition to their status.



1. Enter filter text to reduce the results to only those rows matching the text entered.
2. Customize results by selecting from a **Predefined Range** or from a **Custom Range**.



3. Select a job to view more details.



4. Click the **Export** button to download the current page of results to a file of comma-separated values (CSV).

Job retention

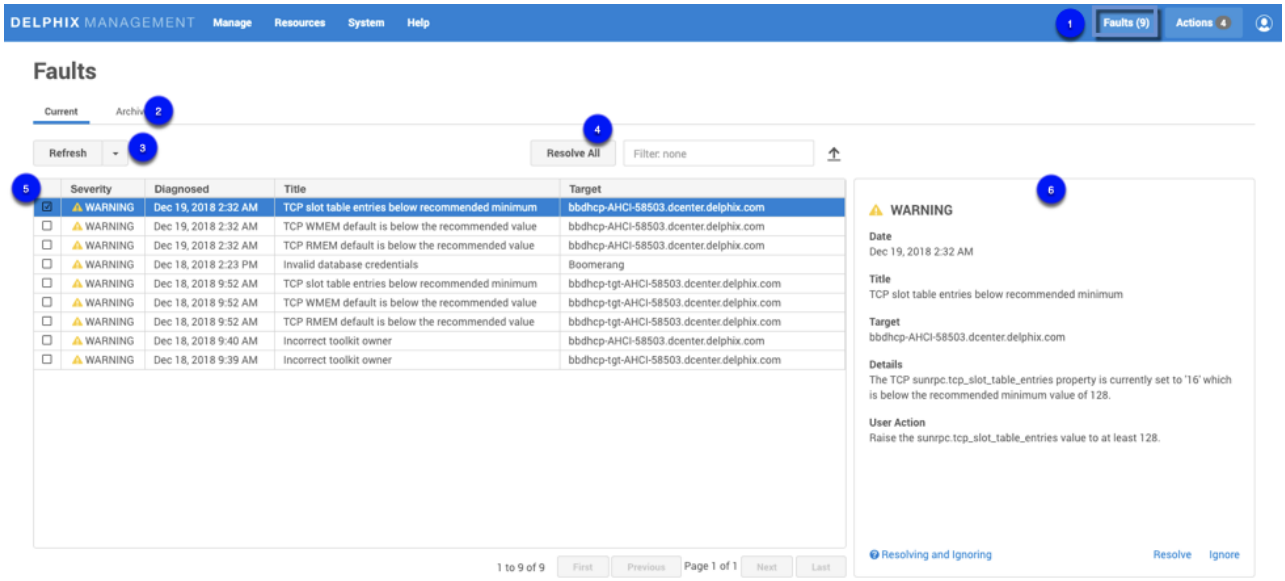
> The Jobs panel retains records of the most recent 10,000 jobs that have been initiated by the Data Engine. Delphix administrators may use the **job/retention** CLI to retain a larger number of jobs.

6.7.4 System faults

6.7.4.1 Overview

The Faults screen provides information on states and configurations that may negatively impact the functionality of the Delphix Engine and which can only be resolved through active user intervention. When you login to the **Delphix management** application as admin, the number of outstanding system faults appears on the right-hand side of the navigation bar at the top of the screen. Faults serve as a record of all issues impacting the Delphix Engine and can never be deleted. However, ignored and resolved faults are not displayed in the faults list.

The **Faults** screen as shown below has two tabs, **Current** and **Archive**.



System Faults screen

1. The number of system faults.

2. The **Faults** screen has two tabs, **Current** and **Archive**. Details of the selected fault are displayed on a card located to the right of the fault list. In the **Archive** tab, you can switch between **Resolved** or **Ignored** faults and reset all ignored faults.

3. Selecting **Refresh (Manual),** will refresh the faults table manually, or you can select one of the other available options from the drop-down menu, available options include; Manual, 1 Second, 1 Minute, 5 Minutes.

(Note: as there is no longer a Refresh button on the screen, you must select Refresh (Manual) to refresh the screen.

4. To search the **Faults** table, enter the name of the object you are looking for. The grid will refresh to display the selected object. You can also sort using the column headings.

Resolve All will resolve all the faults in your system.

Select to expand or close the objects in the grid.

Select to export the information provided in the grid to a.csv file.

5. You can select and resolve multiple faults; the card panel will display how many of each type are selected. For example:

There are 2 WARNING items selected

There are 0 CRITICAL items selected

6. Details for the selected fault are displayed on a card. You can resolve or ignore faults by selecting the appropriate link at the bottom of the card.

Resolving and Ignoring Faults

Ignoring a fault will also ignore future faults of that exact type against the same object, so that future fault conditions will not be re-diagnosed even if the fault condition persists or recurs. No further notifications will be received for that specific fault condition. It is advisable to only ignore faults when the following criteria are met:

- The fault is caused by a well-understood issue that cannot be changed.
- Its impact on the Delphix Engine is well understood and does not require action.

For example, if you think that knowing about this error in the future will be important, use "Resolve" rather than "Ignore". If you reset ignored items, this clears all ignored faults, but it leaves them as resolved and does not restore the actual fault. For reset ignored faults, new faults against the same object will no longer be ignored and you will again receive notifications. Examples: If you ignore a fault "Unable to ping host" for target "192.168.1.1", Delphix ignores "Unable to ping host" errors against target 192.168.1.1. You will never see the "Unable to ping host" fault again for that target 192.168.1.1 unless you reset ignored items. Similarly, some faults are raised against snapshots which are part of a dSource. Ignoring those errors only ignores similar errors for that exact snapshot. Tomorrow's snapshot could produce the fault again.

6.7.4.2 Delphix object-based environment monitor faults

Delphix now has a self-contained Java-based discovery infrastructure that consolidates with environmental monitoring, communicates via a common framework, and is able to provide feedback.

The environment monitor previously only created faults for "hosts" and "sources." There are several faults that more logically apply to other Delphix objects, such as repositories, which are DB install files. Posting them against sources results in fault duplication. The environment monitor now posts faults against -- and re-associates the offending faults with -- the correct objects. Consequently, users see fewer errors that are easier to diagnose.

6.7.4.3 Viewing faults

To view the list of active system faults:

1. In the top navigation bar, click **System** then **Faults**.
2. In the Faults screen, click any fault in the list to expand it and see its details.
   The details for the selected Fault will be displayed in the details card located on the right.

Each fault comprises six parts:

- **Severity** – How much of an impact the fault will have on the system. A fault can have a severity of either **Warning** or **Critical**.
  - A **Warning Fault** implies that the system can continue despite the fault but may not perform optimally in all scenarios.
  - A **Critical Fault** describes an issue that breaks certain functionality and must be resolved before some or all functions of the Delphix Engine can be performed.
- **Date** – The date that the Delphix Engine diagnosed the fault.
- **Title** – A short descriptive summary of the fault
- **Target** – The object against which the fault was posted. Faults will be posted against the host for incorrect environment configurations, sources for problems with the database, and repositories for issues with the installation.
- **Details** – A detailed summary of the cause of the fault
- **User action** – The action you can take to resolve the fault

6.7.4.4 Addressing faults

After viewing a fault and deciding on the appropriate course of action, you can address the fault through the user interface (UI). You can mark a fault as **Ignored** or **Resolved**. If you have fixed the underlying cause of the fault, mark it as **Resolved**. Note that if the fault condition persists, it will be detected in the future and re-diagnosed. You can mark the fault as **Ignored** if it meets the following criteria:

- The fault is caused by a well-understood issue that cannot be changed
- Its impact on the Delphix Engine is well understood and acceptable

In this case, the fault will not be re-diagnosed even if the fault condition persists. You will receive no further notifications.

To address a fault follow the steps below.

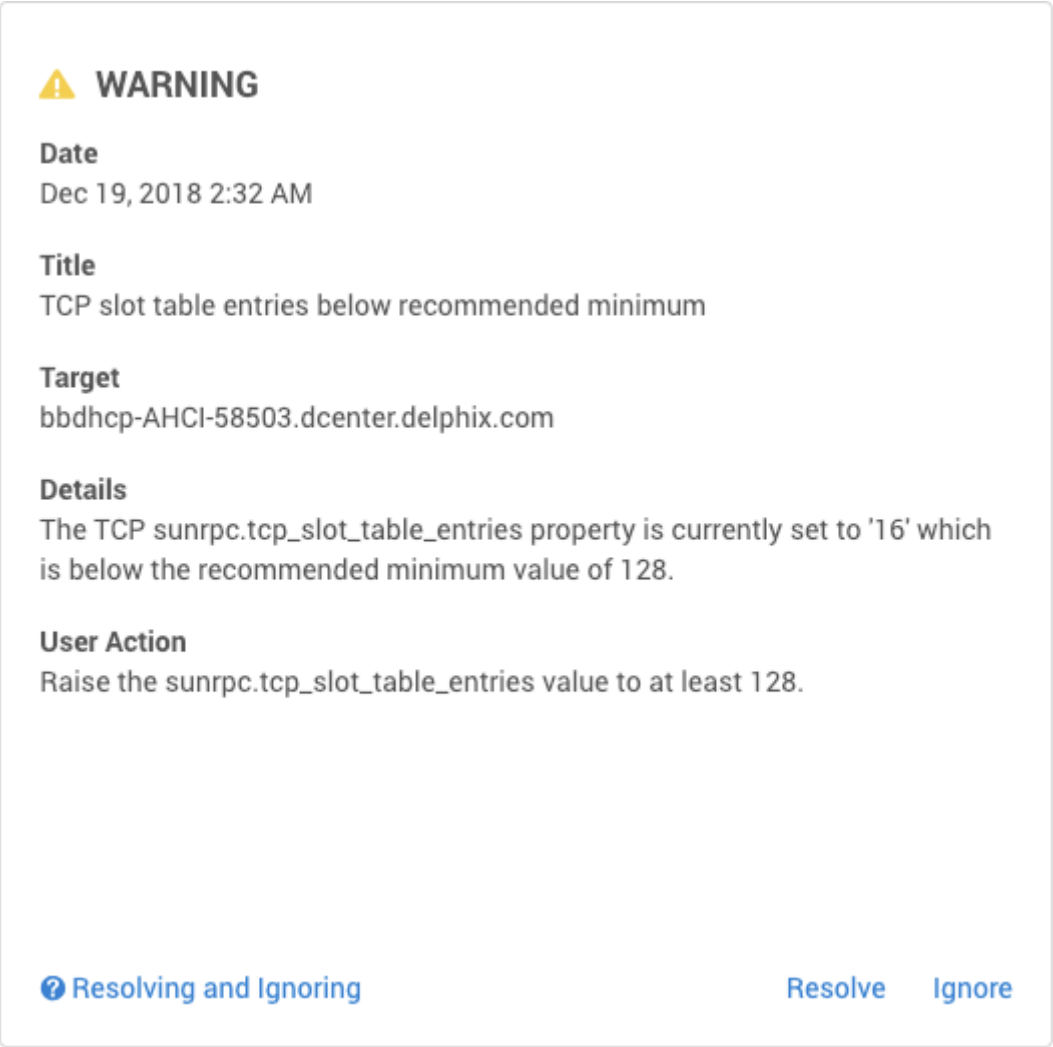
1. In the top menu bar, click **Faults**.
2. In the list of faults, click a **fault date/name** to view the fault details.
3. If the fault condition has been resolved, click **Resolve**.
   Note that if the fault condition persists it will be detected in the future and re-diagnosed.
4. If the fault condition describes a configuration with a well-understood impact on the Delphix Engine that cannot be changed, you can ignore the fault by clicking **Ignore**.
   Note that an ignored fault will not be diagnosed again even if the underlying condition persists.

By default, when a **critical or warning fault** occurs, the Delphix Engine immediately sends an email to the Engine Administrator (**admin**). Make sure you have configured an SMTP server and defined an appropriate email address for Engine Administrator (**admin**). See Initial Setup (see page 571) for more information.

Critical or warning alert emails
By default, emails will also be sent for **critical or warning alerts** (aka events). You can modify the default behavior by changing the alert profile with the CLI. See the CLI Cookbook Creating Alert Profiles (see page 2571) for more information.

6.7.4.5 Fault lifecycle example

Below is an image of the fault card for the fault "TCP slot table entries below the recommended minimum."



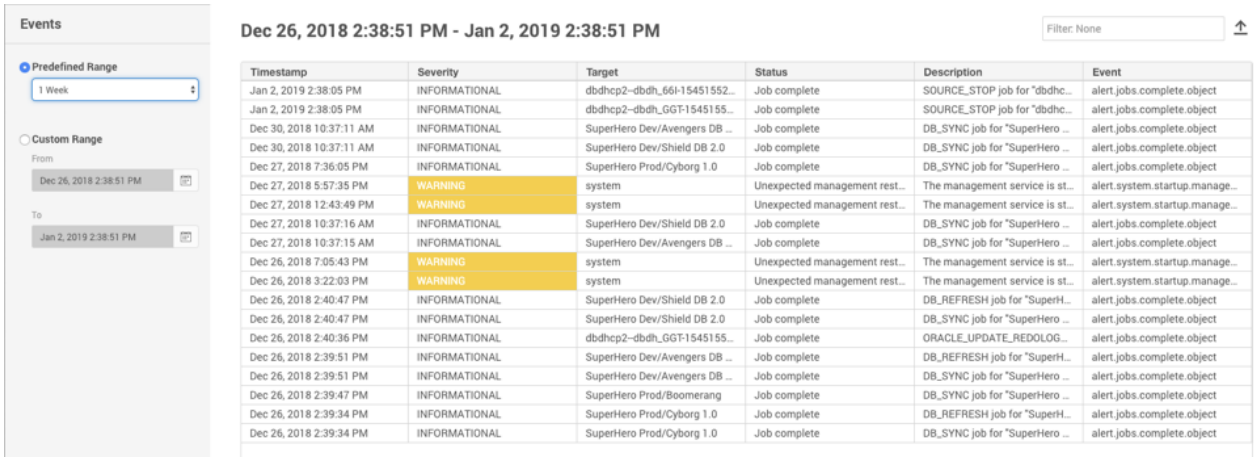
The **Details** section of the fault explains that the sunrpc.tcp_slot_table_entries property is set to a value that is below the recommended minimum of 128. The **User action** section instructs you to adjust the value of the sunrpc.tcp_slot_table_entries property upward to the recommended minimum. The process for adjusting this property differs between operating systems. To resolve the underlying issue, search "how to adjust

sunrpc.tcp_slot_table_entries" using a search engine and find that the second result is a link[246] to the Delphix community forum describing how to resolve this issue. After following the instructions applicable to your operating system, return to the Delphix UI and mark the fault **resolved**.

6.7.4.6 Viewing system faults

6.7.4.6.1 System events overview

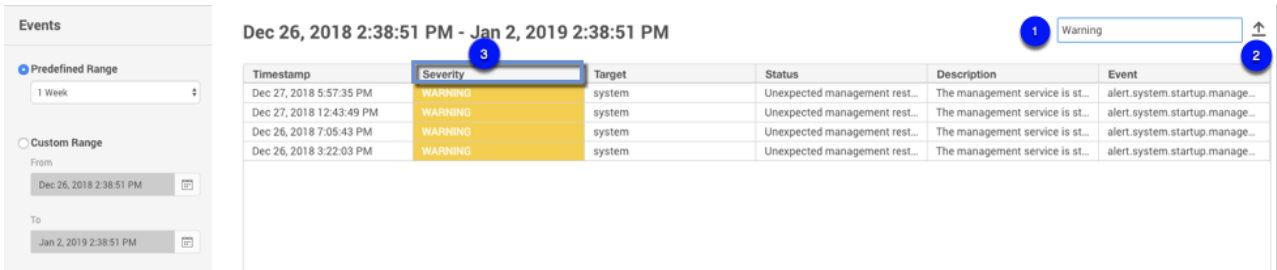
The event log interface has been improved to provide filtering, sorting, and exporting.



Event Viewer screen

As shown above, the **Event Viewer** window provides information about all the events that occurred for the selected time period. Text matching is limited to the following columns:

- Action
- Description



Event Viewer screen filtered for warning events

In the **Event Viewer** window, you can:

1. Enter filter text to reduce the results to only those rows matching the text entered. In the example above, we are filtering for “warning.”
2. Click the **Export** button to export your results to a .csv file.
3. Click a column header to sort rows by the values found in that column.

246 https://community.delphix.com/delphix/topics/tip_of_the_day_tuning_the_kernel_nfs_client_for_rhel4_through_rhel6_3_and_delphix

The first time you click a header, rows will sort in ascending order. Clicking the same header a second time will sort the rows in descending order. Clicking the same header a third time will restore the results to their default sort order.

6.7.4.6.2 Procedure

1. Launch the **Delphix Management** application.
2. Click **System**.
3. Select **Events**.
4. Select a time range.

6.7.4.6.3 Sorting and filtering

Optional: You can enter filter text to reduce the results to only those rows matching the text entered.

Text matching is limited to the following columns:

- Severity
- Status
- Description

You can click on a table column header to sort rows by the values found in that column.

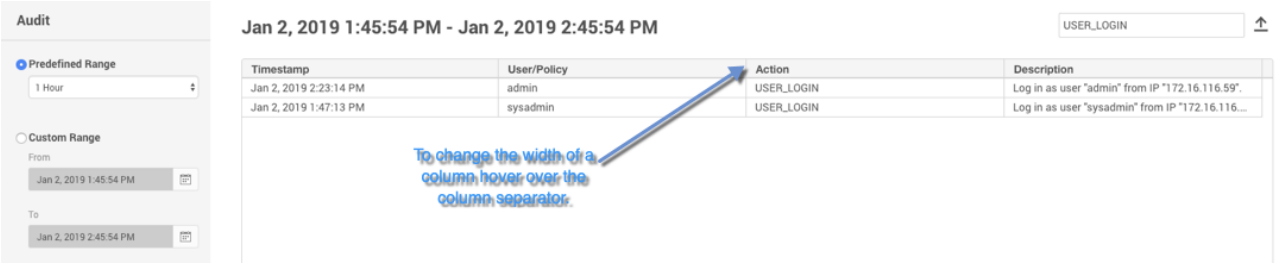
The first time you click a header, rows will sort in ascending order. Clicking the same header a second time will sort the rows in descending order. Clicking the same header a third time will restore the results to their default sort order.

Click the page navigation buttons to advance through large result sets.

6.7.4.6.4 Column resizing and tooltips

If you wish, you can resize column widths to better fit the data to the available screen space. To resize a column:

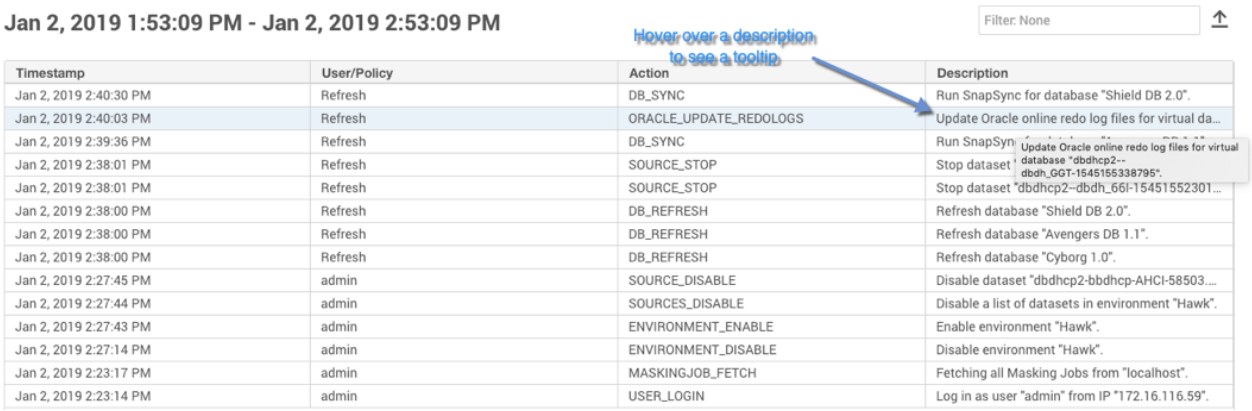
1. Hover the mouse over a column separator found in the header. This will cause the mouse pointer to change shape.
2. Click and drag and the column separator to the desired position. Dragging to the left will reduce the column width. Dragging to the right will increase the width.
3. Release the mouse button.



Alternatively, you can auto-size a column to fit the widest value of the current page:

1. Hover the mouse over a column separator found in the header. This will cause the mouse pointer to change shape.
2. Double click the column separator.

Values that do not fit within their column will be truncated with an ellipses (...). Hover the mouse over any value to see a tooltip rendering the complete, non-truncated value.



Timestamp	User/Policy	Action	Description
Jan 2, 2019 2:40:30 PM	Refresh	DB_SYNC	Run SnapSync for database "Shield DB 2.0".
Jan 2, 2019 2:40:03 PM	Refresh	ORACLE_UPDATE_REDOLOGS	Update Oracle online redo log files for virtual da...
Jan 2, 2019 2:39:36 PM	Refresh	DB_SYNC	Run SnapSyn
Jan 2, 2019 2:38:01 PM	Refresh	SOURCE_STOP	Stop dataset
Jan 2, 2019 2:38:01 PM	Refresh	SOURCE_STOP	Stop dataset "dbdhcp2--dbdh_66l-15451552301...
Jan 2, 2019 2:38:00 PM	Refresh	DB_REFRESH	Refresh database "Shield DB 2.0".
Jan 2, 2019 2:38:00 PM	Refresh	DB_REFRESH	Refresh database "Avengers DB 1.1".
Jan 2, 2019 2:38:00 PM	Refresh	DB_REFRESH	Refresh database "Cyborg 1.0".
Jan 2, 2019 2:27:45 PM	admin	SOURCE_DISABLE	Disable dataset "dbdhcp2-bbdhcp-AHCI-58503....
Jan 2, 2019 2:27:44 PM	admin	SOURCES_DISABLE	Disable a list of datasets in environment "Hawk".
Jan 2, 2019 2:27:43 PM	admin	ENVIRONMENT_ENABLE	Enable environment "Hawk".
Jan 2, 2019 2:27:14 PM	admin	ENVIRONMENT_DISABLE	Disable environment "Hawk".
Jan 2, 2019 2:23:17 PM	admin	MASKINGJOB_FETCH	Fetching all Masking Jobs from "localhost".
Jan 2, 2019 2:23:14 PM	admin	USER_LOGIN	Log in as user "admin" from IP "172.16.116.59".

6.7.4.6.5 Exporting results

Click the **Export** button to download the current page of results to a file of comma-separated values (CSV).

6.7.5 Accessing audit logs

This topic describes how to access audit logs. The audit log provides a record of all actions that were initiated by a policy or user, regardless of whether that action was successful.

6.7.5.1 Overview

Audit logs provide a record of all actions that were initiated by a policy or user, regardless of whether that action was successful. The audit log interface has been improved to provide filtering, sorting, and exporting.

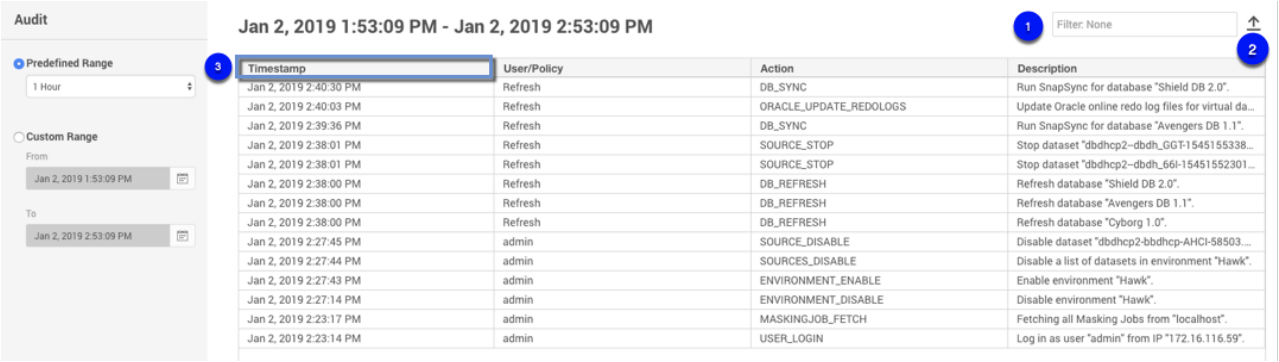
Timestamp	User/Policy	Action	Description
Jan 2, 2019 2:40:30 PM	Refresh	DB_SYNC	Run SnapSync for database "Shield DB 2.0".
Jan 2, 2019 2:40:03 PM	Refresh	ORACLE_UPDATE_REDOLOGS	Update Oracle online redo log files for virtual da...
Jan 2, 2019 2:39:36 PM	Refresh	DB_SYNC	Run SnapSync for database "Avengers DB 1.1".
Jan 2, 2019 2:38:01 PM	Refresh	SOURCE_STOP	Stop dataset "dbdhcp2--dbdh_GGT-1545155338...
Jan 2, 2019 2:38:01 PM	Refresh	SOURCE_STOP	Stop dataset "dbdhcp2--dbdh_66l-15451552301...
Jan 2, 2019 2:38:00 PM	Refresh	DB_REFRESH	Refresh database "Shield DB 2.0".
Jan 2, 2019 2:38:00 PM	Refresh	DB_REFRESH	Refresh database "Avengers DB 1.1".
Jan 2, 2019 2:38:00 PM	Refresh	DB_REFRESH	Refresh database "Cyborg 1.0".
Jan 2, 2019 2:27:45 PM	admin	SOURCE_DISABLE	Disable dataset "dbdhcp2-bbdhcp-AHCI-58503....
Jan 2, 2019 2:27:44 PM	admin	SOURCES_DISABLE	Disable a list of datasets in environment "Hawk".
Jan 2, 2019 2:27:43 PM	admin	ENVIRONMENT_ENABLE	Enable environment "Hawk".
Jan 2, 2019 2:27:14 PM	admin	ENVIRONMENT_DISABLE	Disable environment "Hawk".
Jan 2, 2019 2:23:17 PM	admin	MASKINGJOB_FETCH	Fetching all Masking Jobs from "localhost".
Jan 2, 2019 2:23:14 PM	admin	USER_LOGIN	Log in as user "admin" from IP "172.16.116.59".

Audit window

As shown above, the **Audit** window displays all actions that were initiated for the selected period of time. You can enter filter text to reduce the results to only those rows matching the text entered. In the figure below, we are filtering for “user.”

Text matching is limited to the following columns:

- Action
- Description



Audit window filtered for User events

1

Enter filter text to reduce the results to only those rows matching the text entered. In the example above, we are filtering for “user.”

2

Click the **Export** icon to export your results to a .csv file.

3

You can click a column header to sort rows by the values found in that column.

The first time you click a header, rows will sort in ascending order. Clicking the same header a second time will sort the rows in descending order. Clicking the same header a third time will restore the results to their default sort order.

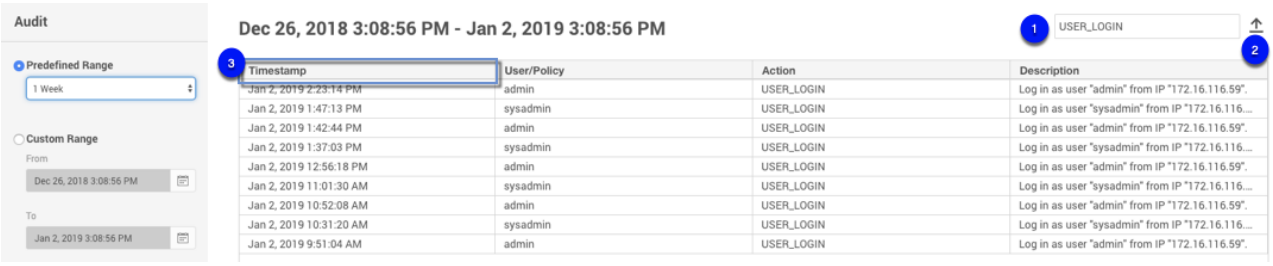
> Actions displayed in the Actions panel or on the Audit page are kept forever.

6.7.5.2 Procedure

1. Login to the **Delphix Management** application using **admin** credentials.
2. Click **System**.
3. Select **Audit**.

4. Select an audit log time range.

6.7.5.3 Sorting and filtering



1

Enter filter text (e.g. USER_LOGIN) to reduce the results to only those rows matching the text entered.

2

Click the **Export** icon to export your results to a .csv file.

3

You can click on a table column header to sort rows by the values found in that column. The first time you click a header, rows will sort in ascending order. Clicking the same header a second time will sort the rows in descending order. Clicking the same header a third time will restore the results to their default sort order. Text matching is limited to the following columns:

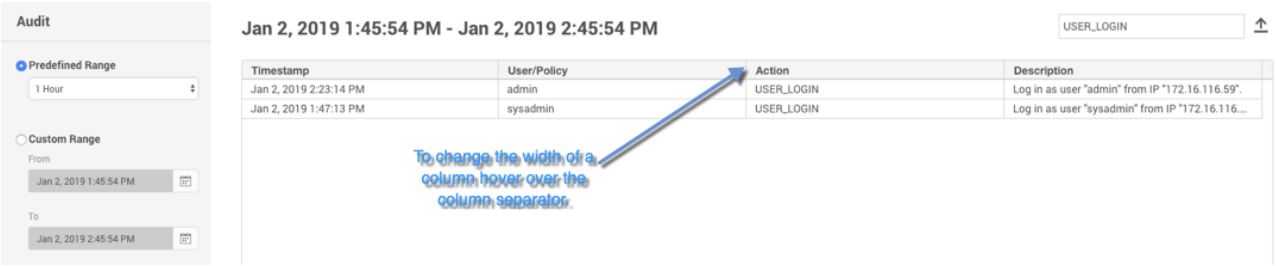
- Action
- Description

Click the page navigation buttons to advance through large result sets.

6.7.5.4 Column resizing and tooltips

If you wish, you can resize column widths to better fit the data to the available screen space. To resize a column:

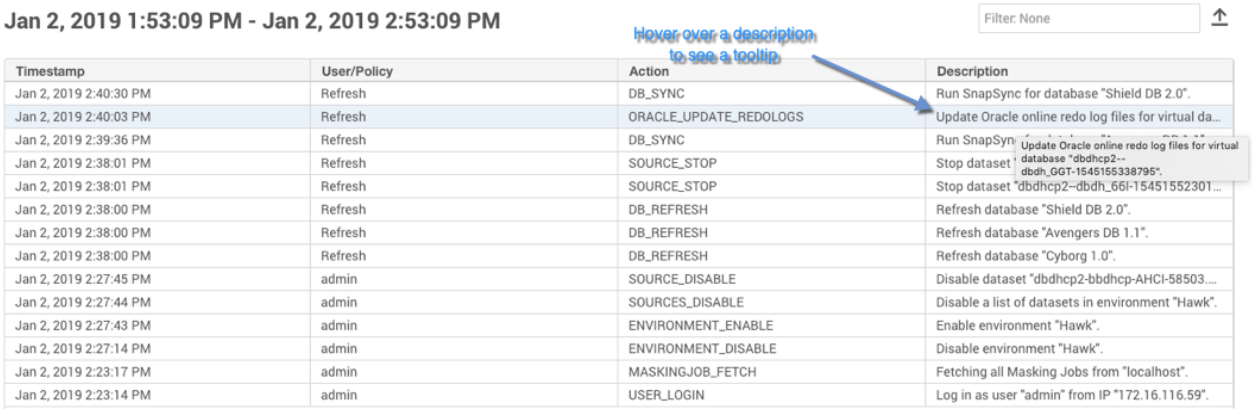
1. Hover the mouse over a column separator found in the header. This will cause the mouse pointer to change shape.
2. Click and drag and the column separator to the desired position. Dragging to the left will reduce the column width. Dragging to the right will increase the width.
3. Release the mouse button.



Alternatively, you can auto-size a column to fit the widest value of the current page:

1. Hover the mouse over a column separator found in the header. This will cause the mouse pointer to change shape.
2. Double click the column separator.

Values that do not fit within their column will be truncated with ellipses (...). Hover the mouse over any value to see a tooltip rendering the complete, non-truncated value.



6.7.5.5 Exporting results

Click the icon to download the current page of results to a file of comma-separated values (CSV).

6.7.6 Creating support logs

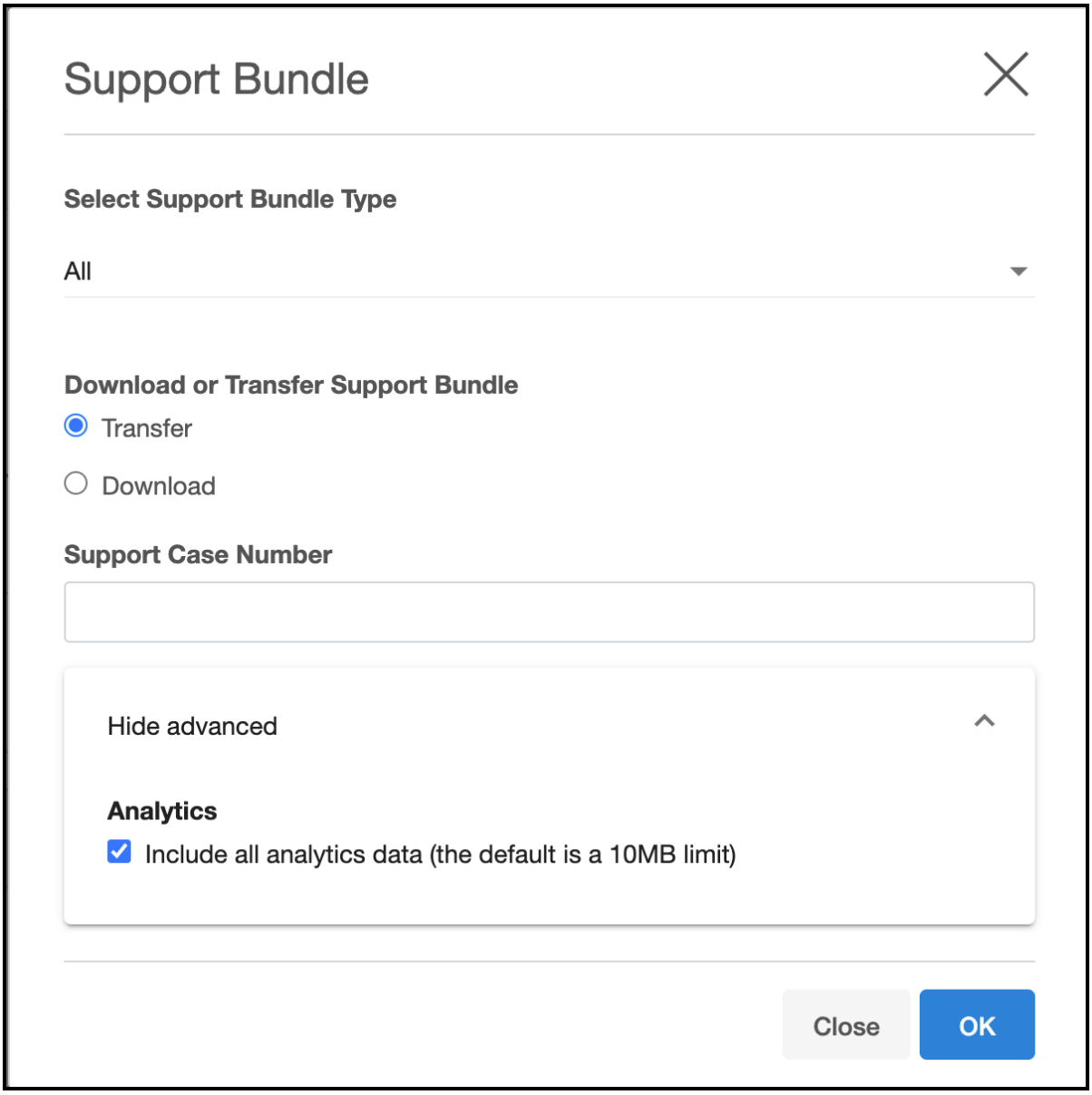
This topic describes how to create support bundles and manage server access control for Delphix Support.

Support bundles are used by Delphix as diagnostic tools for resolving Delphix Engine issues. Support bundles can be transferred directly to Delphix Support or downloaded. Customer-specific data is redacted or obfuscated where possible in the support bundle information. All passwords and personal data are either encrypted or omitted. This is an outbound-only connection from the Delphix Engine.

6.7.6.1 Using the GUI

1. Log into the **Delphix Management** appliance as an Engine administrator.
2. Click **Help**.
3. Select **Support Logs**.

4. The **Support Bundle** dialog appears.



5. Select **Download** or **Transfer**.
   a. If you select **Download**, then the support logs will be downloaded as a compressed `".tar"` file into a folder on your workstation.
   b. If you select **Transfer**, then the support logs will be uploaded over HTTPS to Delphix Support. If you have configured an HTTP Proxy, it will be used to send the support logs.
   c. If there is a support case involved, then please enter the case number to associate the logs to the case.
6. Click on **Show advanced** and select **Analytics**. This will include all the analytics data (default, up to 10MB) in the Support Bundle.
7. Click **OK**.

a. If you selected **Download** and have the compressed " `.tar` " file in a folder on your workstation, please **upload** that file to Delphix Support via the website at http://upload.delphix.com[247].
b. If there is a support case involved, then please enter the case number (again) to associate the logs to the case.

You can also access support log functionality in the **ServerSetup** application using **sysadmin** credentials. Click **Support Bundle** in the top menu bar.

6.7.6.2 Using the CLI

1. ssh into your Delphix Engine.

```
ssh <sysadmin_user>@<delphixengine>
```

2. Run the upload operation.

```
delphix > service
delphix service > support
delphix service support > bundle
delphix service support bundle > upload
```

3. Commit the operation.

```
delphix service support bundle upload *> commit
```

6.7.7 Setting support access control

This topic describes how to set the Support Access Control for Delphix Support. Support access control enables Delphix Support to access your instance of the Delphix Engine for a defined period of time using an access token.

1. Log into the **Delphix Setup** application using **sysadmin** credentials.
2. Click **Server Preferences**.
3. Select **Support Access**.
4. Click **Enable**.

247 http://upload.delphix.com/

5. Set the time period during which you want to allow Delphix Support to have access to your instance of the Delphix Engine.
6. Click **Generate Token**.
   Provide the token to Delphix Support to enable access to your server.

6.7.8 Setting syslog preferences

Syslog is a widely used standard for message logging. It permits the separation of the software that generates messages, the system that stores them, and the software that reports and analyzes them. Delphix makes use of Syslog as one of the standard mechanisms, along with SNMP and email, to distribute important user and system events, such as alerts, faults, and audits. In the case of Delphix, each Delphix Engine acts as a Syslog client which propagates the events to a centralized Syslog server.

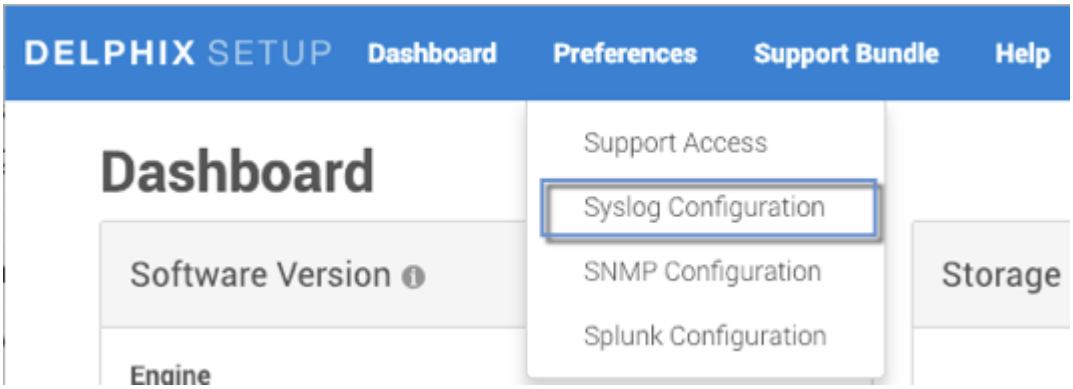
The network protocol over which the Delphix Engine communicates with the Syslog server is standardized in RFC 5424[248]. As a protocol, it supports using either UDP (RFC 5426[249]) or TCP (RFC 6587[250]) as the underlying transport and optional TLS mapping has been introduced to encrypt the messages over the wire for security purposes (RFC 5425[251]). However, as of this release, we only support Syslog over UDP with no encryption, which implies that Syslog messages are always sent in the clear and may be lost during transmission and delivered out of order due to the limitations of UDP.

To configure for Syslog support, you must specify the communication endpoint to which the Syslog server listens, which includes the hostname or IP address of the Syslog server and an optional port number. The latter defaults to 514 according to the Syslog standard but it can be changed if necessary.

System and user events generated by Delphix are always forwarded immediately to the Syslog server, which ensures the timely delivery of important events that may require immediate action.

A couple of different output formats are supported for messages delivered over Syslog, namely, TEXT and JSON. The TEXT format is the default. To change the message format, as of this release, you must do so via the CLI.

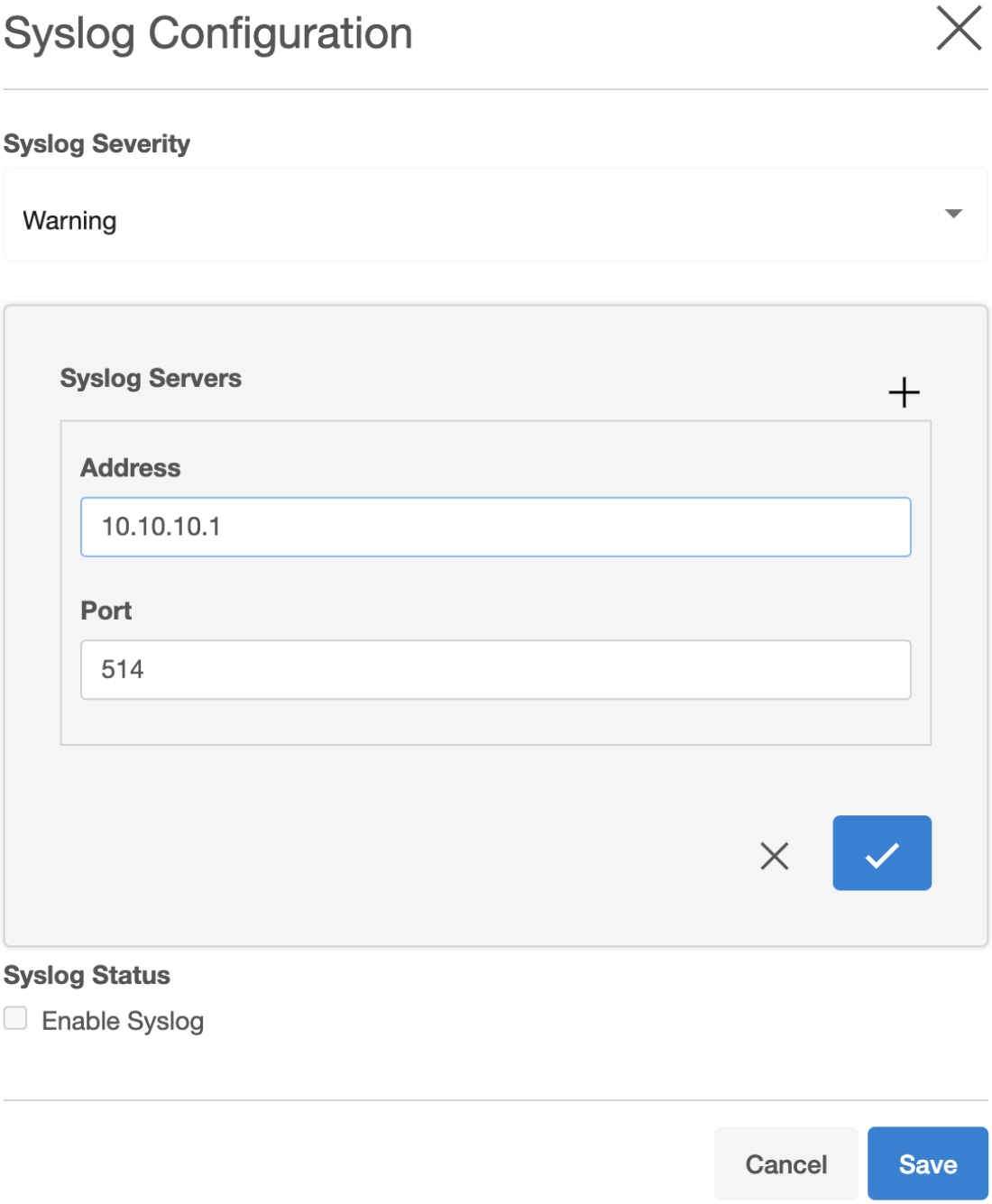
1. Log into the **Delphix Setup** application using **sysadmin** credentials.
2. Select **Preferences > Syslog Configuration**.



3. Select the severity level of the messages you want to be sent to the Syslog server.

248 https://tools.ietf.org/html/rfc5424
249 https://tools.ietf.org/html/rfc5426
250 https://tools.ietf.org/html/rfc65
251 https://tools.ietf.org/html/rfc5425

4. Click the **pencil** icon next to **Syslog Servers** and then in the **Syslog Configuration** window select the plus icon.
5. Enter the Syslog server hostname/IP address and port number.



6. Select **Enable Syslog**.
7. Click **Save**.

6.7.8.1 Severity levels for syslog messages

This topic discusses the Syslog reporting feature of the Delphix Engine, along with severity levels.

Syslog is a widely used standard for message logging. It permits the separation of the software that generates messages, the system that stores them, and the software that reports and analyzes them. Delphix makes use of Syslog as one of the standard mechanisms, along with SNMP and email, to distribute important user and system events, such as alerts, faults, and audits. In the case of Delphix, each Delphix Engine acts as a Syslog client which propagates the events to a centralized Syslog server.

Every Syslog message is attached to a severity level. As the name suggests, the severity level describes the severity of the event in question.

Audit Logs

Audit records are Informational Syslog messages. If you would like to forward Audit records, choose Severity Level Informational.

Severity levels

Every Syslog message is attached to a severity level number. Delphix defines the severity of Syslog messages in accordance with RFC 3164. There are eight severity levels available, as follows:

```
Numerical         Severity
Code

           0       Emergency: system is unusable
           1       Alert: action must be taken immediately
           2       Critical: critical conditions
           3       Error: error conditions
           4       Warning: warning conditions
           5       Notice: normal but significant condition
           6       Informational: informational messages
           7       Debug: debug-level messages
```

When setting up the Syslog settings for your Delphix Engine, you have the ability to choose what alerts to report. The severity levels above are available for users to select. Once you select a severity level, the Delphix Engine will send messages of the same or higher severity (i.e., the same or lower number) to your Syslog server. Therefore, there is no reason to select more than one severity. For example, if the "Notice" severity level is selected, all events less severe than Notice (Informational and Debug) will not be reported. If you want all events to be reported via Syslog, the Debug severity level should be chosen.

6.7.9 Support access audit logs

This topic describes how terminal session audit logging works within the Delphix OS. These logs contain keystroke by keystroke recordings of all terminal activity during a given shell session initiated by a super user (Delphix support).

6.7.9.1 Overview

Super user activity by Delphix support is recorded to an individual log file for each shell session. Each log file is named using the format **session_<shell user ip>_<epoch timestamp>**. The contents of the logs include commands entered into the shell and the output of those commands. Timestamps are additionally prepended to each line of the log to facilitate assessing the timeline of events.

> Session logs created during a super user shell session are kept forever unless deleted by a Delphix engine SYSTEM user.

6.7.9.2 Listing the session audit logs

Super user session logs can be reviewed/deleted through the CLI or API and downloaded through the API only. Any DOMAIN or SYSTEM user can list the current logs or download a given log file but only SYSTEM users can delete them. To review a list of the session logs currently present on a Delphix engine:

1. Login to the **Delphix CLI** using **admin** or **sysadmin** credentials.
2. Navigate to **superuser session** and press enter.
3. Use **list** or **ls** to view the files.

```
ip-12-345-678-90 superuser session> ls
Objects
NAME                                IPADDRESS      STARTTIMEUTC
DURATION
session_123.45.678.90_1686923517171  123.45.678.90  2023-06-16T13:51:57.171Z  20sec
session_123.45.678.90_1686923559856  123.45.678.90  2023-06-16T13:52:39.856Z  439sec
session_123.45.678.90_1686924008788  123.45.678.90  2023-06-16T14:00:08.788Z  87sec
```

Here is an example of calling the list API directly using curl:

```
curl -b ~/cookies.txt -X GET "http://mydelphixengine.myorg.com/resources/json/
delphix/superuser/session"
```

6.7.9.3 Downloading a session audit log

Any DOMAIN or SYSTEM user can download a super user session log file via the Delphix API. Here is an example of calling the download API using curl:

```
curl -v -O -J "http://mydelphixengine.myorg.com/resources/json/delphix/superuser/
session/download?sessionLogName=session_123.45.678.90_1686923517171" -b ~/cookies.txt
```

6.7.9.4 Reviewing a session audit log

It is recommended that session logs be viewed through a program such as **cat**, which is capable of interpreting control characters. This is because the logs not only include key strokes and terminal output, but also the control characters that dictate how that output was formatted and displayed, ensuring that the logs reflect what was actually seen during the shell session as accurately as possible. It is also possible to view the logs using any text editor, but in most cases this will be more difficult to read because the control characters themselves will be visible. Here is a snippet from a brief session log as it might be displayed by **cat**:

```
[2023-06-16T13:52:28.061Z] delphix:~$ echo testing 123
[2023-06-16T13:52:28.062Z] testing 123
[2023-06-16T13:52:30.599Z] delphix:~$ exit
```

Note that each line of the log includes a timestamp. This timestamp is prepended to each line as the log is written. It is not from the session terminal output, but rather is provided to more conveniently assess the timeline of a given session. The timestamp is generated in the instant before a given command is executed rather than when the prompt was first printed to the terminal to maximize its accuracy.

6.7.9.5 Limitations

Shell activity that involves opening a pager or buffer (e.g. **less, more, vi,** etc.) may not be fully reflected in the session log, though the command that initiates the pager/buffer will be present. For example, if a super user opens a file in **vi** for editing, the line to open the file would be present, followed by the next command run after **vi** was closed.

6.7.9.6 Deleting a session audit log

These audit logs are meant to live as long as they are needed, and thus are not governed by a retention policy. Should you wish to delete a log this can be done by SYSTEM users only through the Delphix CLI or API. To use the CLI:

1. Login to the **Delphix CLI** using **sysadmin** credentials.
2. Navigate to **superuser session** and press enter.
3. Review the current log files using **list** or **ls**
4. Select the log file that you wish to delete.
5. Use **list** or **ls** to review the log details and confirm this is the log you want to delete.
6. Type **delete** and press enter.
7. Type **commit** and press enter to delete the log.

```
ip-12-345-678-90 superuser session> select "session_123.45.678.90_1686923517171"
ip-12-345-678-90 superuser session 'session_123.45.678.90_1686923517171'> ls
Properties
    type: SuperuserSession
    name: session_123.45.678.90_1686923517171
    duration: 20sec
    ipAddress: 123.45.678.90
    reference: SUPERUSER_SESSION-session_123.45.678.90_1686923517171
    startTimeUTC: 2023-06-16T13:51:57.171Z

Operations
delete
ip-12-345-678-90 superuser session 'session_123.45.678.90_1686923517171'> delete
ip-12-345-678-90 superuser session 'session_123.45.678.90_1686923517171' delete *>
commit
ip-12-345-678-90 superuser session> ls
Objects
NAME                                IPADDRESS      STARTTIMEUTC
DURATION
session_123.45.678.90_1686923559856  123.45.678.90  2023-06-16T13:52:39.856Z  439sec
session_123.45.678.90_1686924008788  123.45.678.90  2023-06-16T14:00:08.788Z  87sec
```

The delete API can also be called directly. Unlike the download API, delete requires the session log reference, which is always SUPERUSER_SESSION-<log name>.Here is an example of calling the delete API using curl:

```
curl -X POST "http://mydelphixengine.myorg.com/resources/json/delphix/superuser/
session/SUPERUSER_SESSION-session_123.45.678.90_1686923517171/delete" -b ~/cookies.tx
t
```

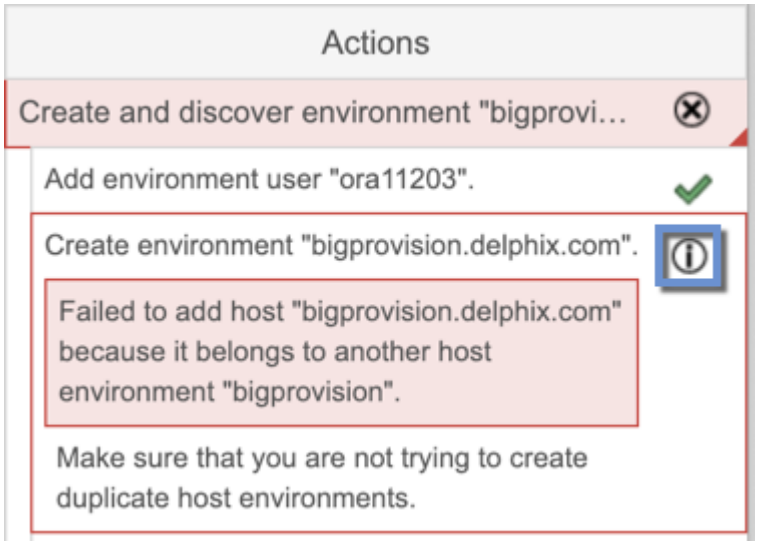
6.7.10 Diagnosing connectivity errors

Prior to the 5.1 release, when the Delphix Engine ran into an error operating on an external database or environment, it reported the immediate error that it had encountered; there was no mechanism for automatic analysis of the root causes of failures. The 5.1 release included infrastructure for automatic diagnosis of errors. When one of there errors occurs, the Delphix Engine now launches a set of tests to locate the root cause of the problem and present the result of the diagnosis. This will help you easily identify the true sources of errors such as closed ports or misconfigured router.

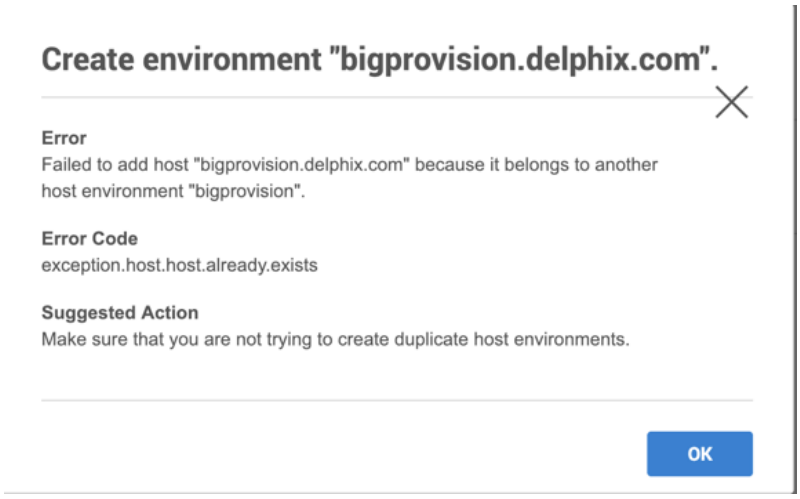
6.7.10.1 Failed actions

The Delphix Engine communicates failures in two different manners: actions that fail to complete, and faults. To view failed actions:

1. In the top right-hand corner of the Delphix Management application, click **Actions**.
2. For more information about why the action failed, click the (**i**) icon to show the error dialog.



The following shows a popup message with more information about the problem and what actions to take to resolve it. For some errors, the Delphix Engine will be able to diagnose the problem further and display this extra information under **Diagnosing Information**. In the screenshot above, the job failed because the Delphix Engine was unable to lookup the host address.



6.7.10.2 Viewing active faults

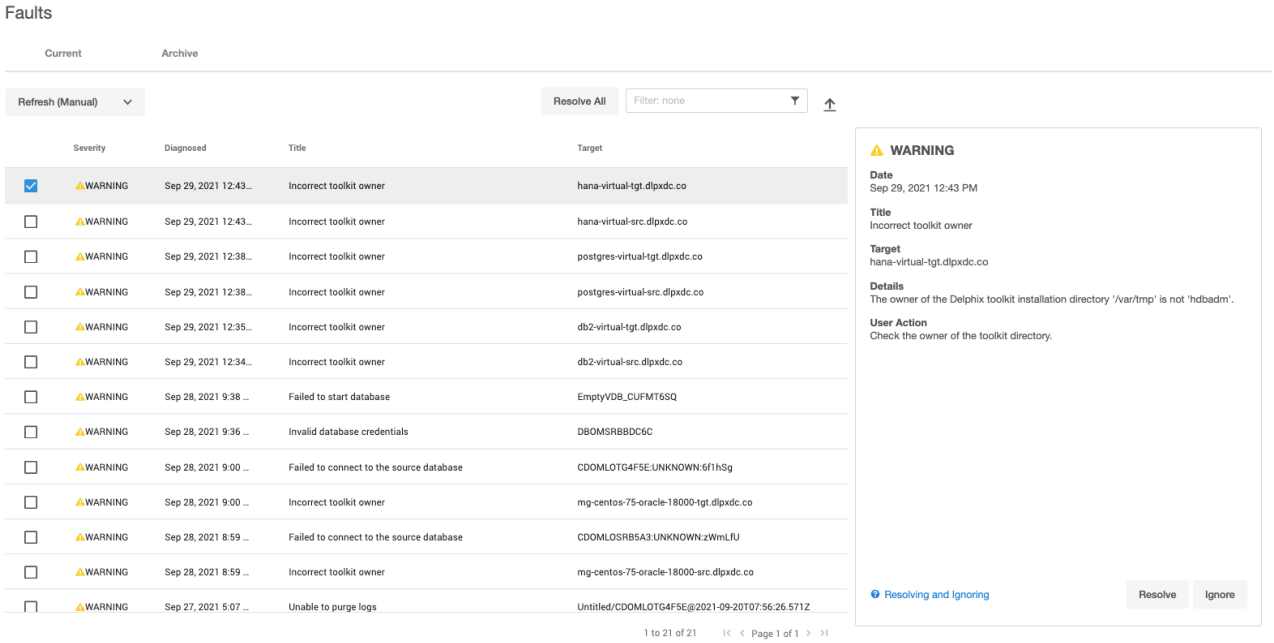
A fault symbolizes a condition that can affect the performance or functionality of the Delphix Engine and must be addressed. Faults can be either warnings or critical failures that prevent the Delphix Engine from functioning normally. For example, a problem with a source or target environment can cause SnapSync or LogSync policy jobs to fail. Faults will show up as active as long as:

- The error is still occurring, or
- You have chosen to manually resolve it or ignore it

For example, if a background job fails, it will create a fault that describes the problem. To view any active faults:

- In the top right-hand corner of the Delphix Management interface, click **Faults**.

This brings the Faults screen listing all active faults.



The screenshot above illustrates a fault with regard to a failure to a TCP slot table entry. The Delphix Engine will mark an object with a warning triangle to indicate that it is affected by an external problem. You can view more details of the fault by looking at the active faults and their fault effects.

6.7.11 Email (SMTP) alert notifications

6.7.11.1 Overview

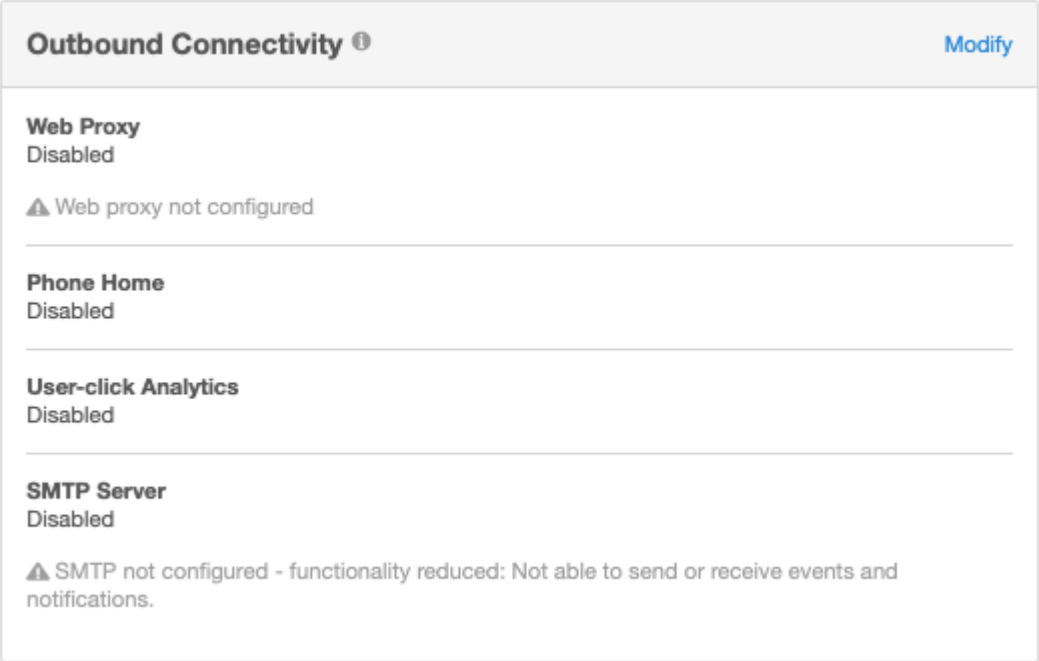
The configuration for SMTP-based alert notifications has two components:

- The configuration of an SMTP gateway by the Delphix system administrator
- The configuration of one or more alert profiles (if needed)

6.7.11.2 Configuring the SMTP gateway

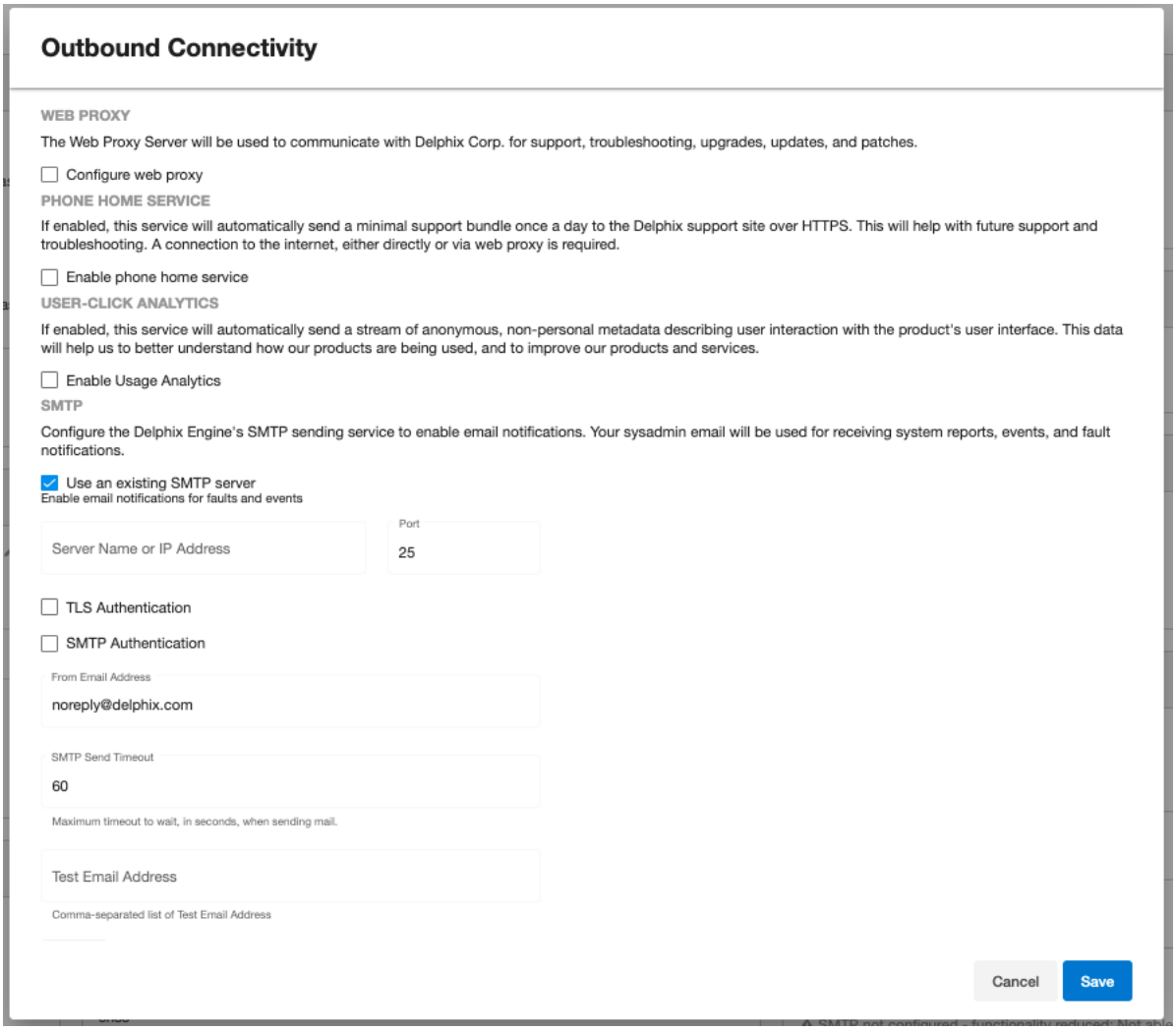
Before email-based alerts can function properly, many organizations require that an SMTP gateway is configured, through which all outbound email is sent.

1. Contact the appropriate administrator for your site in order to determine the SMTP gateway settings.
2. Login to the **Delphix setup** application as **sysadmin** or another user with system administrator privileges.
3. On the Dashboard screen, locate the **Outbound connectivity** section, and click **modify**.



Outbound Connectivity for SMTP Gateway

4. If not checked already, check the box next to **Use an existing SMTP server.**



5. At a minimum, enter the required information:
   a. SMTP Server Name of IP Address
   b. SMTP port
   c. From Email Address This will be the email address from which all alert emails will be sent.
6. In **Test Email Address**, enter the same email address to verify that you are able to receive email properly.
7. Click **Save** to save changes.

For further information, see the "Outbound Connectivity" section of Initial Setup (see page 571)

6.7.11.3 Alert profiles

The Delphix Engine can send out email notifications when alerts happen. Alert profiles control this functionality.

An alert profile is composed of two things:

- **Filter Specification:** A filter, or combination of filters, that specifies which alerts are of interest.

- **Alert Action:** This specifies the email addresses to which the Delphix Engine will send an email when an alert matches the filter specification.

By default, the Delphix Engine has a single alert profile configured with the following parameters:

- Filter Specification: Match any alert with a severity level of **CRITICAL** or **WARNING**.
- Alert Actions: Send an email to the address defined for user **admin**.
  Default domain user
  The default domain user created on Delphix Engines is now **admin** instead of delphix_admin. When engines created before 5.3.1 are upgraded to 5.3.1 or later they will retain their old username 'delphix_admin'. To avoid complications Delphix recommends creating users with an admin role and then Disabling delphix_admin.

Using the CLI, it is possible to:

- Modify the system default alert profile
- Create additional profiles in addition to the default one
- Set multiple actions for a single profile, such as "email dephix_admin" *and* "email user1@mycompany.com".

6.7.11.3.1 Simple filters

- Filtered by Owner of alerts target – for example, objects owned by user 1

6.7.11.3.2 Complex filters

Complex filters combine/modify other sub-filters:

- “And” filter – Used when all conditions defined must be met for the filter to notify the user with an email
- “Or” filter – Used when either one or the other of the conditions defined in the filters must be met for the filter to notify the user with an email
- “Not” filter – Used to exclude items

6.7.11.3.3 Limitations

- This is a CLI feature.
- Alert Profiles do not override permission settings. If you do not have Read permission on an object then your alert profile will never get triggered for that object's alerts, regardless of your filter settings.

The following CLI examples will run through how to create these three filters. Each example provides three different methods of setting up a profile. These include the following:

- A simple profile
- A profile with two filters
- A complicated profile

For more information, see CLI Cookbook: Creating Alert Profiles (see page 2571)

6.7.11.4 A simple profile

A simple profile approach matches the Delphix out-of-the-box default alert profiles. To create a simple alert profile using the CLI as seen in the figure below, go into the alert profile section of the command-line interface (CLI) and create a new profile. Line four prompts the engine to send an email when the filters are triggered. The following three command lines refer to the filter specifications. Follow two severity levels: warning and critical. This will trigger an email alert when any warning or critical events occur.

```
delphix > cd alert
delphix alert > cd profile
delphix alert profile > create
delphix alert profile create > set actions.0.type=AlertActionEmailUser
delphix alert profile create > set filterSpec.type=SeverityFilter
delphix alert profile create > set filterSpec.severityLevels.0=CRITICAL
delphix alert profile create > set filterSpec.severityLevels.1=WARNING
delphix alert profile create > commit
```

Simple Alert Profile example in the CLI

6.7.11.5 A compound alert profile

Creating a compound alert profile in the CLI will combine two filters together. This profile triggers an email about any alert on objects owned by the delphix_admin plus any other alert that is critical. The compound alert profile creates two filters. The first one will be the target owner filter, which in this case is **admin**. The second filter is the severity filter, allowing users to match anything that is critical. Combine these two filters using the OR logic so that if any of the sub-filters match, the whole filter matches. An example of this can be seen in the figure below.

Alert Profile using OR logic

While working in the CLI, the first four lines describe a simple alert profile. The distinction between simple and compound alert profiles is that in a compound profile, the top-level filter uses an OR filter with sub-filters for target owner and severity level, as seen in line five of the figure below.

```
delphix > cd alert
```

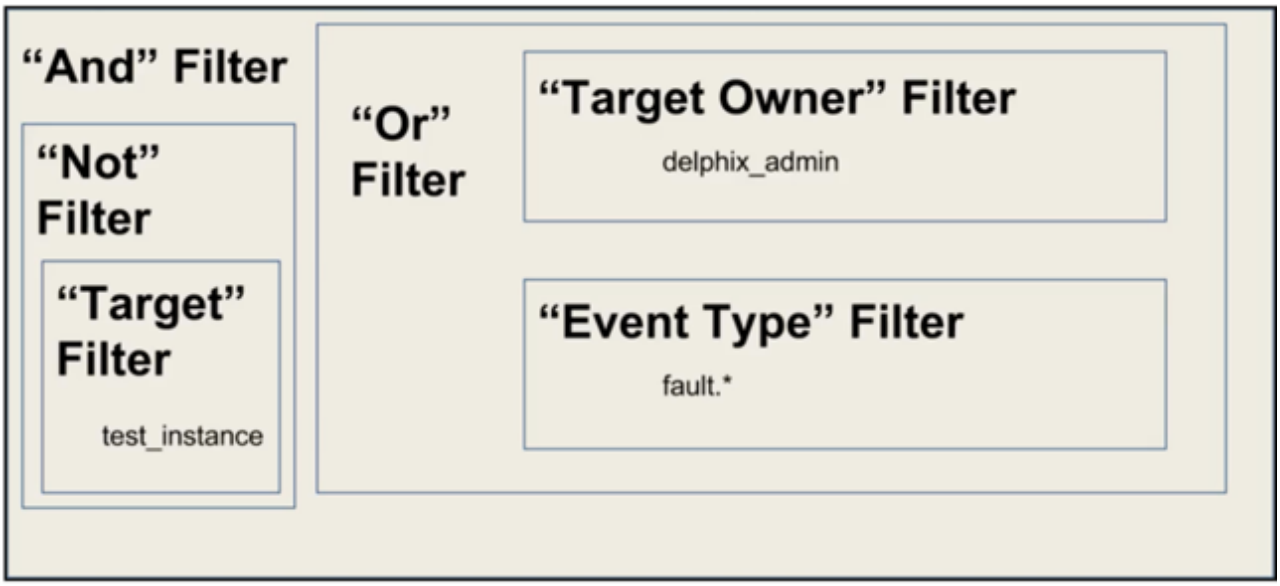
```
delphix alert > cd profile
delphix alert profile > create
delphix alert profile create > set actions.0.type=AlertActionEmailUser
delphix alert profile create > set filterSpec.type=OrFilter
delphix alert profile create > set filterSpec.subFilters.0.type=TargetOwnerFilter
delphix alert profile create > set filterSpec.subFilters.0.owners.0=delphix_admin
delphix alert profile create > set filterSpec.subFilters.1.type=SeverityFilter
delphix alert profile create > set filterSpec.subFilters.1.severityLevels.0=CRITICAL
delphix alert profile create > commit
```

A Compound Alert Profile

6.7.11.6 Complex alert profile

A complex alert profile uses the profile filter created in the compound alert profile and modifies it. For the example shown in the figure below, you have a VDB named test_instance that you do not need any emails about. The following commands will create an effective filter.

1. Create an OR filter with two sub filters: target owner and event type.
2. Create a NOT filter that will exclude the VDB (test_instance) from which you do not want to receive notifications.
3. Use the AND logic to combine all these filters together, as seen below.



Complex Alert Profile

Below is an example of the command lines used to set up this complex profile.

```
delphix > cd alert
delphix alert > cd profile
delphix alert profile > create
delphix alert profile create > set actions.0.type=AlertActionEmailUser
```

```
delphix alert profile create > set filterSpec.type=AndFilter
delphix alert profile create > set filterSpec.subFilters.0.type=NotFilter
delphix alert profile create > edit filterSpec.subFilters.0.subFilter
delphix alert profile create filterSpec.subFilters.0.subFilter > set
type=TargetFilter
delphix alert profile create filterSpec.subFilters.0.subFilter > set targets.0=test_i
nstance
delphix alert profile create filterSpec.subFilters.0.subFilter > back
delphix alert profile create > set filterSpec.subFilters.1.type=OrFilter
delphix alert profile create > set filterSpec.subFilters.1.subFilters.0.type=TargetOw
nerFilter
delphix alert profile create > set filterSpec.subFilters.1.subFilters.0.owners.0=delp
hix_admin
delphix alert profile create > set filterSpec.subFilters.1.subFilters.1.type=Severity
Filter
delphix alert profile create > set filterSpec.subFilters.1.subFilters.1.severityLevel
s=CRITICAL
delphix alert profile create > commit
```

Complex Alert Profile CLI

6.7.11.6.1 Creating alert profiles

1. SSH into your engine's CLI using your delphix_admin username and password

```
ssh delphix_admin@yourdelphixengine
```

2. Start creating your new profile. After logging in to get to the alert section enter alert.

```
delphix > alert
delphix alert > profile
delphix alert profile > create
delphix alert profile create > ls
```

3. Set Action(s) Use **AlertActionEmailList**if you want to specify a list of email addresses for this profile.

```
delphix alert profile create > set actions.0.type=AlertActionEmailList
delphix alert profile create > set actions.0.addresses.0=<email address to send
to>
delphix alert profile create > set actions.0.addresses.1=<additional email
address>
delphix alert profile create > set actions.0.addresses.2=<additional email
address>
```

Or, use **AlertActionEmailUser** if you just want the emails to go to the email address associated with this Delphix user.

```
delphix alert profile create > set actions.0.type=AlertActionEmailUser
```

It is possible to add more than one action here, so you may use both **AlertActionEmailList** and **AlertActionEmailUser** if desired.

4. Set filter Here is an example of setting a simple severity filter. With this filter, emails will be sent for any **CRITICAL** or **WARNING** alerts.

```
delphix alert profile create > set filterSpec.type=SeverityFilter
delphix alert profile create > set filterSpec.severityLevels.0=CRITICAL
delphix alert profile create > set filterSpec.severityLevels.1=WARNING
```

Here is an example of setting a simple target-owner filter. With this filter, emails will be sent for any alert whose target is owned by delphix_admin.

```
delphix alert profile create > set filterSpec.type=TargetOwnerFilter
delphix alert profile create > set filterSpec.owners.0=delphix_admin
```

Here is an example of a compound filter. With this filter, we combine the above two filters – an email is sent when an alert is **CRITICAL** or **WARNING**, and the alert's target is owned by delphix_admin.

```
delphix alert profile create > set filterSpec.type=AndFilter
delphix alert profile create > set filterSpec.subFilters.0.type=SeverityFilter
delphix alert profile create > set filterSpec.subFilters.0.severityLevels.0=CRI
TICAL
delphix alert profile create > set filterSpec.subFilters.0.severityLevels.1=WAR
NING
delphix alert profile create > set filterSpec.subFilters.1.type=TargetOwnerFilt
er
delphix alert profile create > set filterSpec.subFilters.1.owners.0=delphix_adm
in
```

5. Commit your changes

```
delphix alert profile create > commit
```

6.7.11.6.2 Profile filters

As seen above, you can use different filter types to customize which alerts the Delphix Engine will send emails about. The various filter types are listed below.

6.7.11.6.2.1 Simple filters

Filter type	Purpose	Example	Allowed Values
SeverityFilter	Match based on the alert's severity level (critical, warning, informational)	severityLevels.0=CRITICAL severityLevels.1=WARNING This would match any alert whose severity level is CRITICAL or WARNING.	1 or more of: • CRITICAL • WARNING • INFORMATIONAL
EventFilter	Match based on the alert's event type.	eventTypes.0=fault.* This would match any alert that is generated due to a newly-raised fault on the engine.	One or more text entries, optionally using the ***** wildcard.
TargetFilter	Match based on the alert's target.	targets.0=”Group/DB” This would match any alert whose target is the database “DB” located in the group “Group”.	Any object in the system. 1 or more objects may be specified.
TargetOwnerFilter	Match based on the owner of the alert's target.	owners.0=delphix_admin This would match any alert whose target's owner is the **delphix_admin** user.	Any user in the system. 1 or more users may be specified.

6.7.11.6.2.2 Compound filters

These filters combine/modify the behavior of other filters, called "subfilters". The subfilters may be of any type (simple or complex).

Filter Type	Purpose	Number of subfilters required
AndFilter	This filter matches if **all** subfilters match	2 or more
OrFilters	This filter matches if **any** subfilter matches.	2 or more
NotFilter	This filter matches if the subfilter does **not** match.	1

6.7.11.6.3 Action types

With the AlertActionEmailUser type, notification emails will be sent to the email address of the user who owns the alert profile.

With the AlertActionEmailList type, a list of email addresses must be specified in the "addresses" array. Notification emails will be sent to these addresses.

6.7.11.6.4 Email format options

You can send plain text as well as structured JSON. JSON can be useful for constructing solutions that will parse the email and perform further actions (notify, escalate, log).

To change the format, while updating an alert profile:

```
delphix alert profile ALERT_PROFILE-X update > set actions.0.format=<JSON|TEXT>
```

6.7.12 Fluentd plugin service for API modules

6.7.12.1 Overview

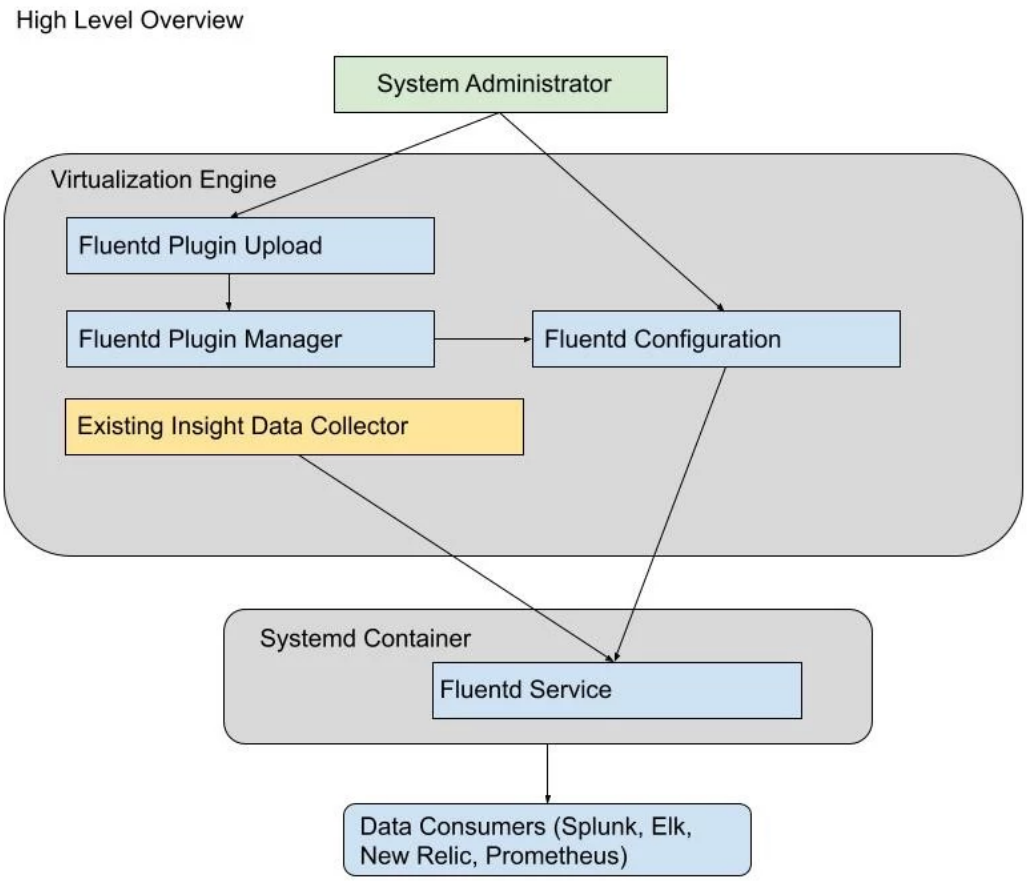
The Delphix Fluentd plugin service assisted in a feature that provided options for configuring a Delphix engine that forwards events and metrics to a Splunk host. Delphix has developed methods to extend this capability for use with other monitoring packages (ELK Stack, Datadog, etc.). Fluentd plugins provide a mechanism to customize the Delphix engine's forwarding capabilities for output to other data consumers.

The Fluentd plugin is a tar file with a configuration template, including any gem files that are required to execute the directives in the configuration file. Once the plugin is uploaded, the configuration occurs by collecting user input for the variables required by the template. The interfaces will have changed since the set of variables needed is now mutable from plugin to plugin. Splunk configuration will be available out of the box with no additional plugins. This feature uses one UI for all Fluentd configurations.

> The current implementation is limited to one data consumer at a time. Only one plugin, in addition to the default SplunkHec plugin, can be uploaded. This is an intended simplification for the initial Fluentd release version.

6.7.12.2 Technical details

6.7.12.2.1 Architectural diagram



6.7.12.2.2 GUI, API or CLI changes (if any)

APIs with a corresponding CLI for uploading plugins and Fluentd configuration have been added. The service/insight/plugins API displays the plugins that are available.

```
service insight plugins> ls

Objects   REFERENCE         PLUGIN       GEMS   ATTRIBUTEDEFINITIONS
SCHEMADEFINITION

FLUENTD_PLUGIN-1           splunkHec    …          …
…
```

```
Operations

requestUploadToken
```

The splunkHec plugin appears by default and cannot be deleted. Additionally, the requestUploadToken operation provides a token needed to upload a plugin via the data/upload API. If a second plugin is uploaded, it will appear in the list. Before a third plugin could be uploaded, the second must be selected and deleted.

The following curl commands illustrate the use of the API to upload a plugin:

```
curl -s -X POST -k --data @- http://<engine-name>/resources/json/delphix/session -c
~/cookies.txt -H "Content-Type: application/json" <<EOF

{

   "type": "APISession",

   "version": {

       "type": "APIVersion",

       "major": 1,

       "minor": 4,

       "micro": 3

   }

}

EOF

curl -s -X POST -k --data @- http://<engine-name>/resources/json/delphix/login -b ~/
cookies.txt -c ~/cookies2.txt -H "Content-Type: application/json" <<EOF

{

"type": "LoginRequest",

"username": "sysadmin",

"password": "sysadmin",

"target": "SYSTEM"

}

EOF
```

```
curl -s -X POST -k --data @- http://<engine-name>/resources/json/delphix/service/
insight/plugins/requestUploadToken -b ~/cookies2.txt -H "Content-Type: application/
json" <<EOF

Returns the token e.g:  {"type":"OKResult","status":"OK","result":{"type":"FileUpload
Result","url":"/resources/json/delphix/data/upload","token":"59346df5-3ecd-4ced-
afbf-97acefa156dc"},"job":null,"action":null}

curl  -b ~/cookies2.txt -X POST -F "file=@/Users/blewis/ws/far-dev/
splunk_host_port.far" -F "token=<token>" http:/<engine-name>/resources/json/delphix/
data/upload
```

The service/insight/plugins API displays the tokens that are available.

An example for creating a configuration with the default splunkHec plugin:

```
service insight configuration> create

service insight configuration create *> set plugin=splunkHec; set enabled=true; edit
attributes

service insight configuration create attributes *> add; set name=hec_host; set
value=http://vmname-splunkhost.delphix.com; back

service insight configuration create attributes *> add; set name=hec_port; set
value=8088; back;

service insight configuration create attributes *> add; set
type=FluentdSecretAttribute;

service insight configuration create attributes 2 *> set name=hec_token; set
secretValue=bb75c032-bdea-4c19-b152-d158cf13e019; back

service insight configuration create attributes *> add; set name=eventsIndex; set
value=delphix_events; back;

service insight configuration create attributes *> add; set name=protocol; set
value=https; back

service insight configuration create attributes *> add; set
name=metricsPushFrequency; set value=60; back

service insight configuration create attributes *> add; set name=metricsIndex; set
value=delphix_metrics; back

service insight configuration create attributes *> add; set name=eventsPushFrequency;
set value=60; back;

service insight configuration create attributes *> ls

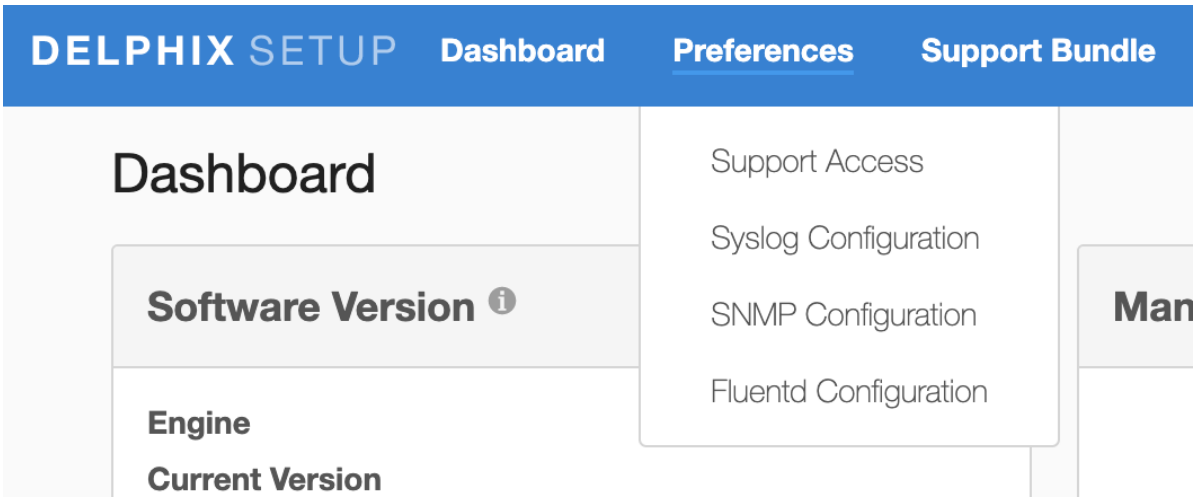
Properties
```

```
0:
    type: FluentdRegularAttribute (*)
    name: hec_host (*)
    value: http://vmname-splunkhost.delphix.com (*)
1:
    type: FluentdRegularAttribute (*)
    name: hec_port (*)
    value: 8088 (*)
2:
    type: FluentdSecretAttribute (*)
    name: hec_token (*)
    secretValue: ******** (*)
3:
    type: FluentdRegularAttribute (*)
    name: eventsIndex (*)
    value: delphix_events (*)
4:
    type: FluentdRegularAttribute (*)
    name: protocol (*)
    value: https (*)
5:
    type: FluentdRegularAttribute (*)
    name: metricsPushFrequency (*)
    value: 60 (*)
6:
    type: FluentdRegularAttribute (*)
```

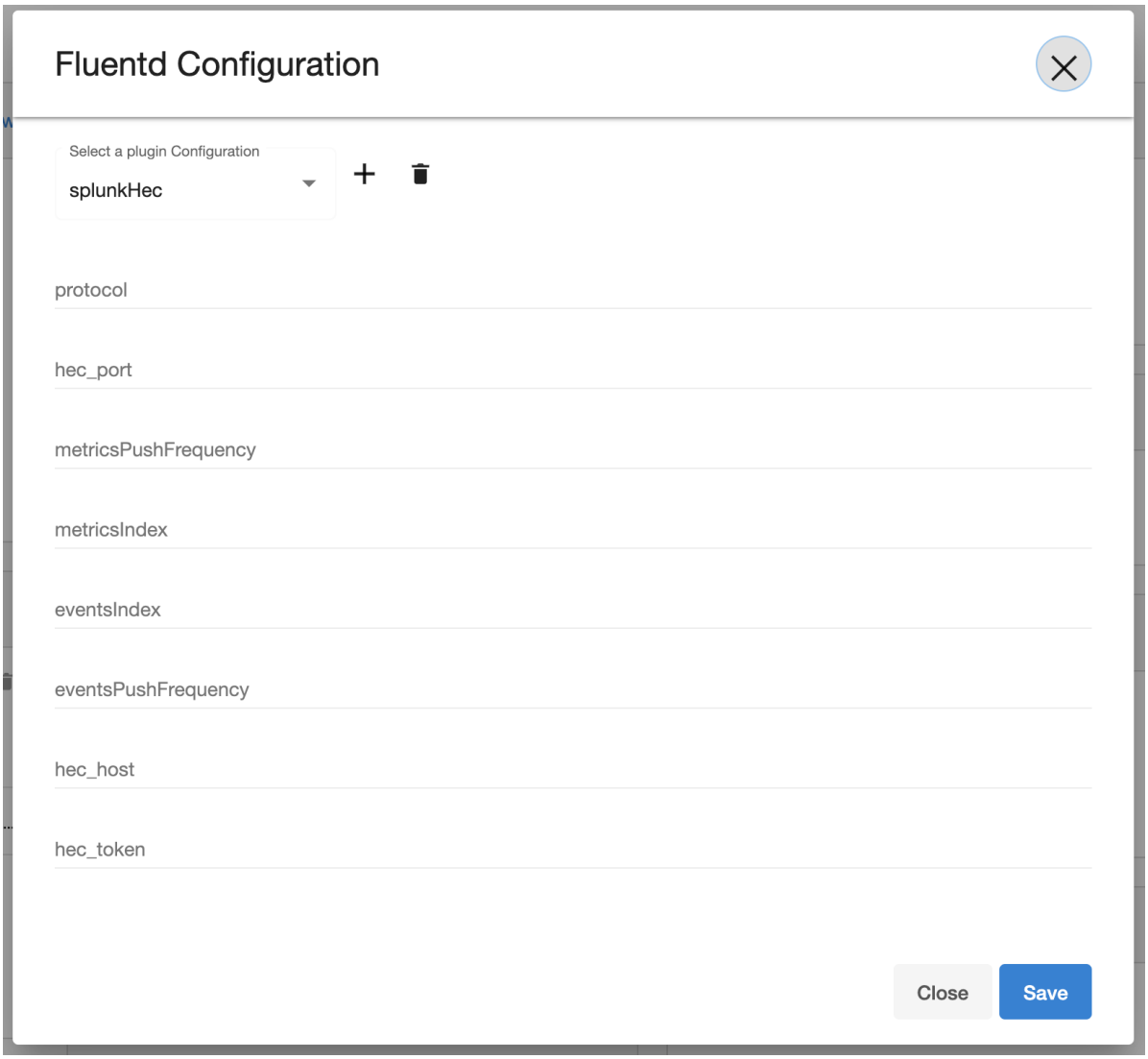
```
        name: metricsIndex (*)
        value: delphix_metrics (*)
    7:
        type: FluentdRegularAttribute (*)
        name: eventsPushFrequency (*)
        value: 60 (*)
## Use the "add" command to add an element to this array.
service insight configuration create attributes *> back
service insight configuration create *> ls
Properties
    type: FluentdConfig
    name: (unset)
    attributes: [ ... ] (*)
    enabled: true (*)
    plugin: splunkHec (*)
service insight configuration create *> commit
```

Creating default splunkHec configuration from GUI

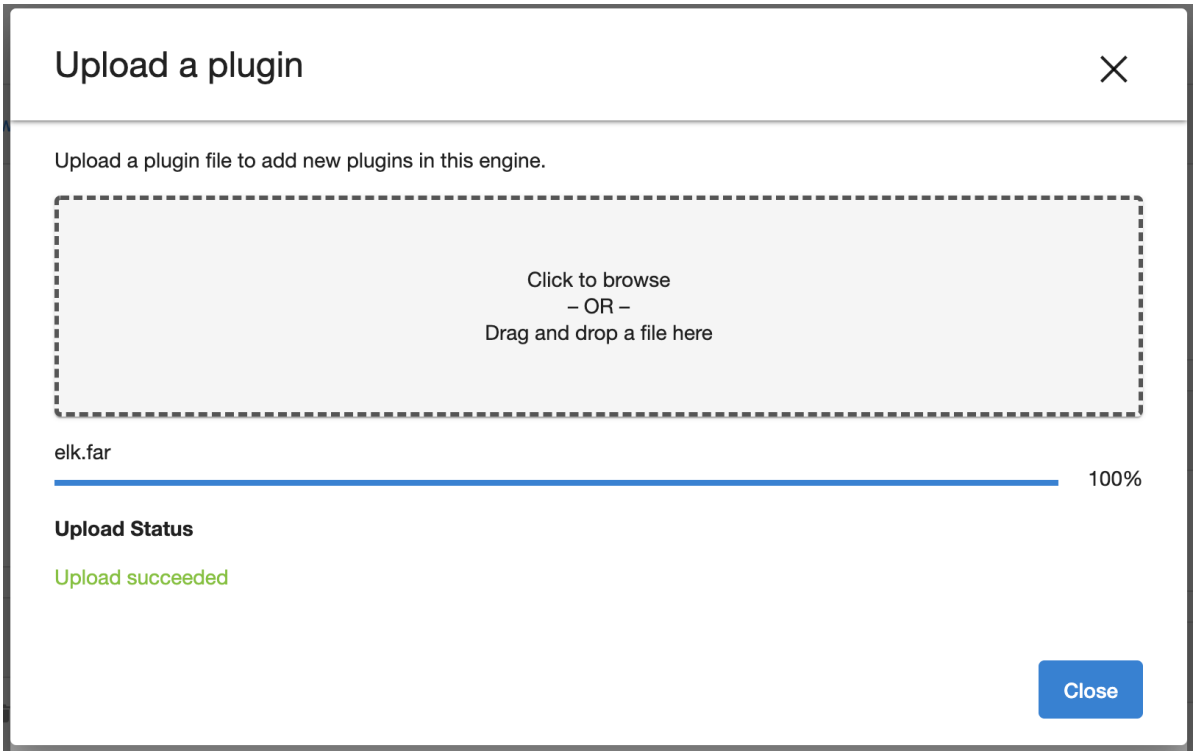
1. Login as a sysadmin user and select **Fluentd Configuration** under the Preferences menu.



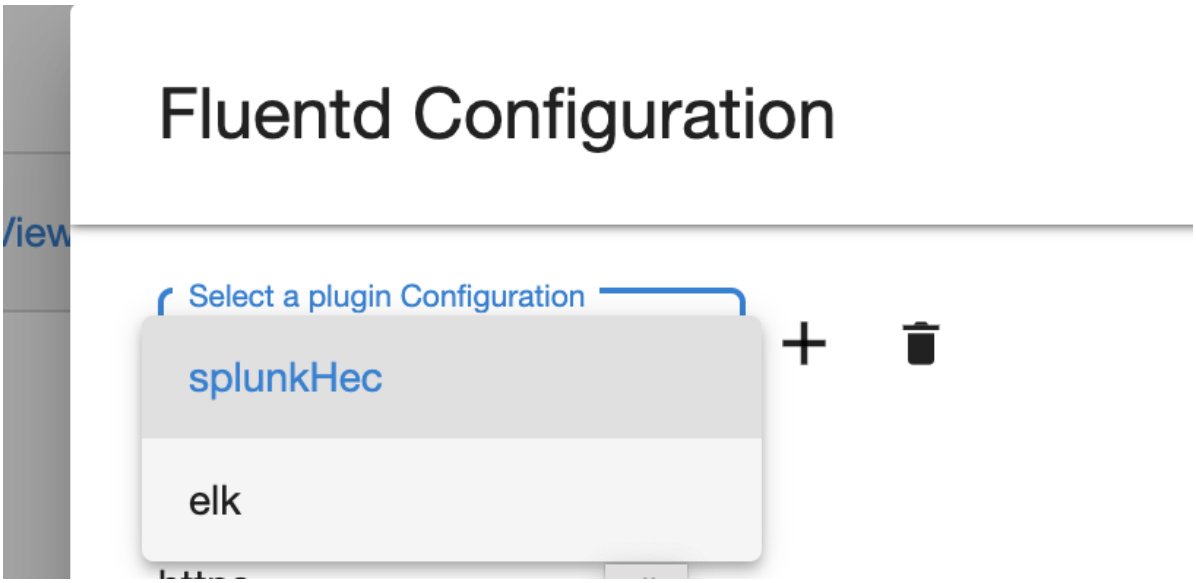
2. The splunkHec plugin appears in the dropdown by default.



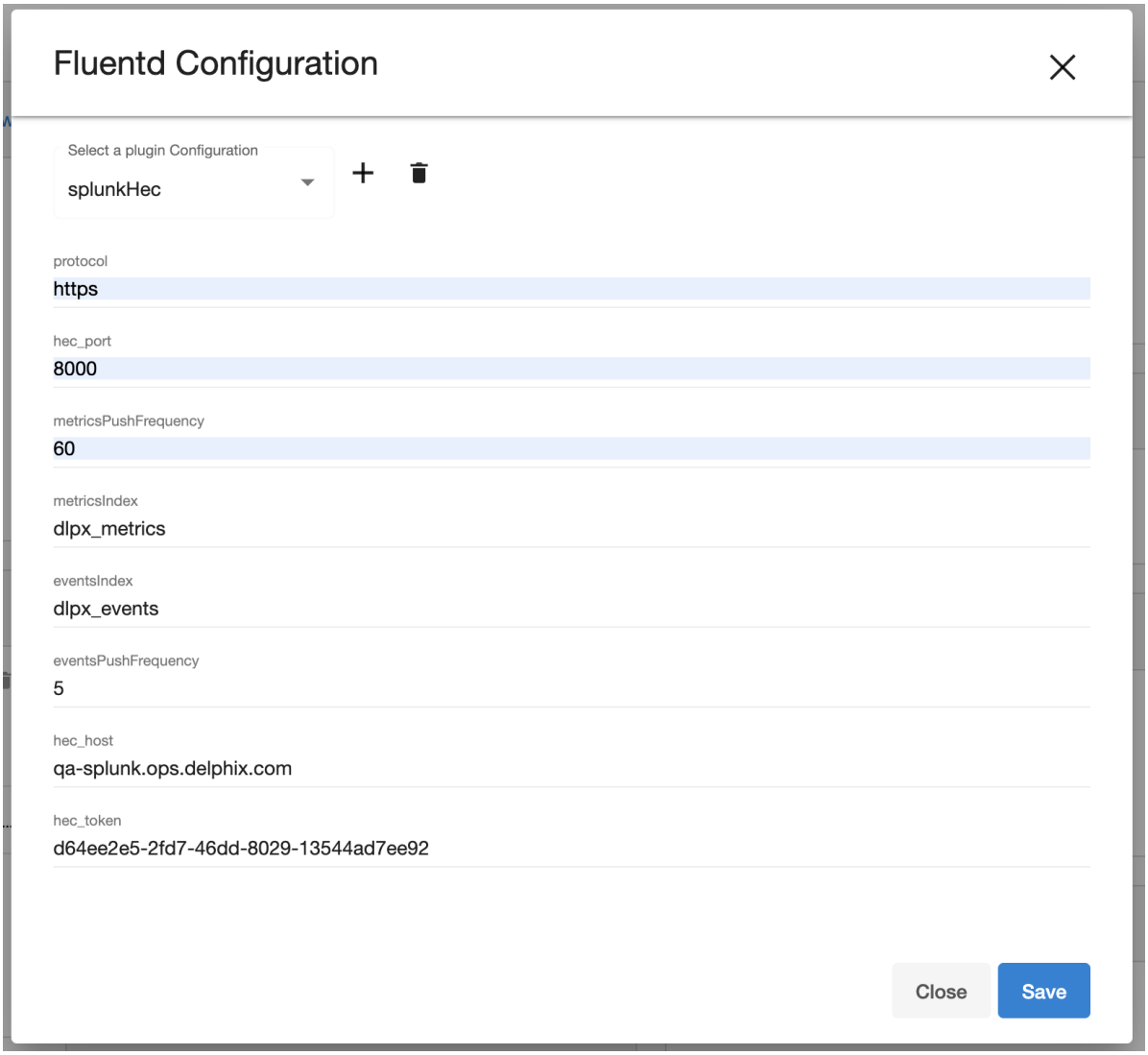
3. Additional plugins can be added by using the **+** button on the right.



4. The new plugin uploaded can be viewed in the dropdown menu and selected to add a configuration.



5. The delete button on the right can be used to delete a selected plugin.
6. The name of fields for configuration would be the same as those provided in the uploaded plugin file.



7. The configuration can be saved by clicking the save button on the bottom.
8. When loading an existing configuration on UI, the secret fields are masked.

6.7.12.2.3 Troubleshooting

Occasionally, the fluentd service might fail to start or encounter problems when sending logs to the remote data consumer. Typical explanations might include syntax errors in the plugin's config file or network configuration problems on the data consumer server. Such issues are generally reflected by error output in the `fluent.log` file, which can be accessed for download via the `/service/fluentd/plugins/downloadFluentdLog` API endpoint. Here is an example of downloading the log via curl:

```
curl -v -O -J "$delphix_engine/resources/json/delphix/service/fluentd/plugins/
downloadFluentdLog" -b ~/cookies.txt
```

> Fluentd rotates its logs once they have reached a certain maximum size to avoid memory issues should they grow too large. While unlikely, it is possible that the logs might be rotated in the midst of an ongoing issue you are trying to diagnose. If this occurs, it may be necessary to wait some time and retry the log download so that useful information is included.

6.7.12.2.4 Implementation

Secret attributes have been introduced in order to protect private data such as passwords and tokens. In the CLI and API, they are identified by the **FluentdSecretAttribute** type. In plugins, attribute names in the config file template beginning with **_secret** indicate secret data.

6.7.12.2.5 Known issues

Fluentd runs in a systemd container to limit negative side-effects from a bad plugin. More security can be added to this container by running as a non-root user, in addition to limiting the size of the log and buffer files it can write.

6.7.12.3 Splunk integration

Delphix enables self-monitoring/diagnosability of Delphix Engines by providing native integration with Splunk Enterprise. By providing details about your Splunk instance, you can allow Delphix to send structured JSON logs to Splunk that capture activity on the Delphix engine(s). These logs include Delphix events (Actions, Job Events, API Events, Faults, and Alerts) as well as performance metrics (CPU, disk, network, TCP, dataset, NFS, iSCSI) and capacity metrics. Delphix Insight enables extensible search and visualization of actionable information and provides a centralized, comprehensive view of Delphix activity (including the ability to cross-reference information from multiple Delphix engines) on a platform that allows building your own operational intelligence for your Delphix installation.

This section covers the following topics:

- Configuring splunk (see page 835)
- Using search (see page 838)

6.7.12.3.1 Configuring splunk

6.7.12.3.1.1 Prerequisites

Before you configure the Delphix Engine you will need to configure and make a note of the following in Splunk:

Please refer to the Splunk documentation[252] for detailed steps on how to configure your values.

Supported Splunk versions
Delphix only supports Splunk Enterprise 6.3.0 or higher.

1. In the Splunk web **UI Enable SSL** (this is optional but best practice for security) in your global HTTP Event Collector (HEC) settings.
2. The **Splunk hostname** or **IP Address**.
3. The **HEC Port** number for your Splunk instance (default 8088).
4. Enable the HTTP Event Collector on Splunk, and create a new **HEC Token** with a new Splunk index set as an allowed index for the token. Make sure **Enable Indexer Acknowledgement** is **unchecked** for the token. **Warning** : If you wish, you can use a separate Splunk index for performance and capacity metrics (otherwise, the same index will be used for both events and metrics). If you are using Splunk 7.0+, it is recommended that you create this second index as a special “Metrics” type index that is optimized for indexing and searching metrics data.
5. Note the **HEC Token Value** and the **Allowed Indexes** for the token.

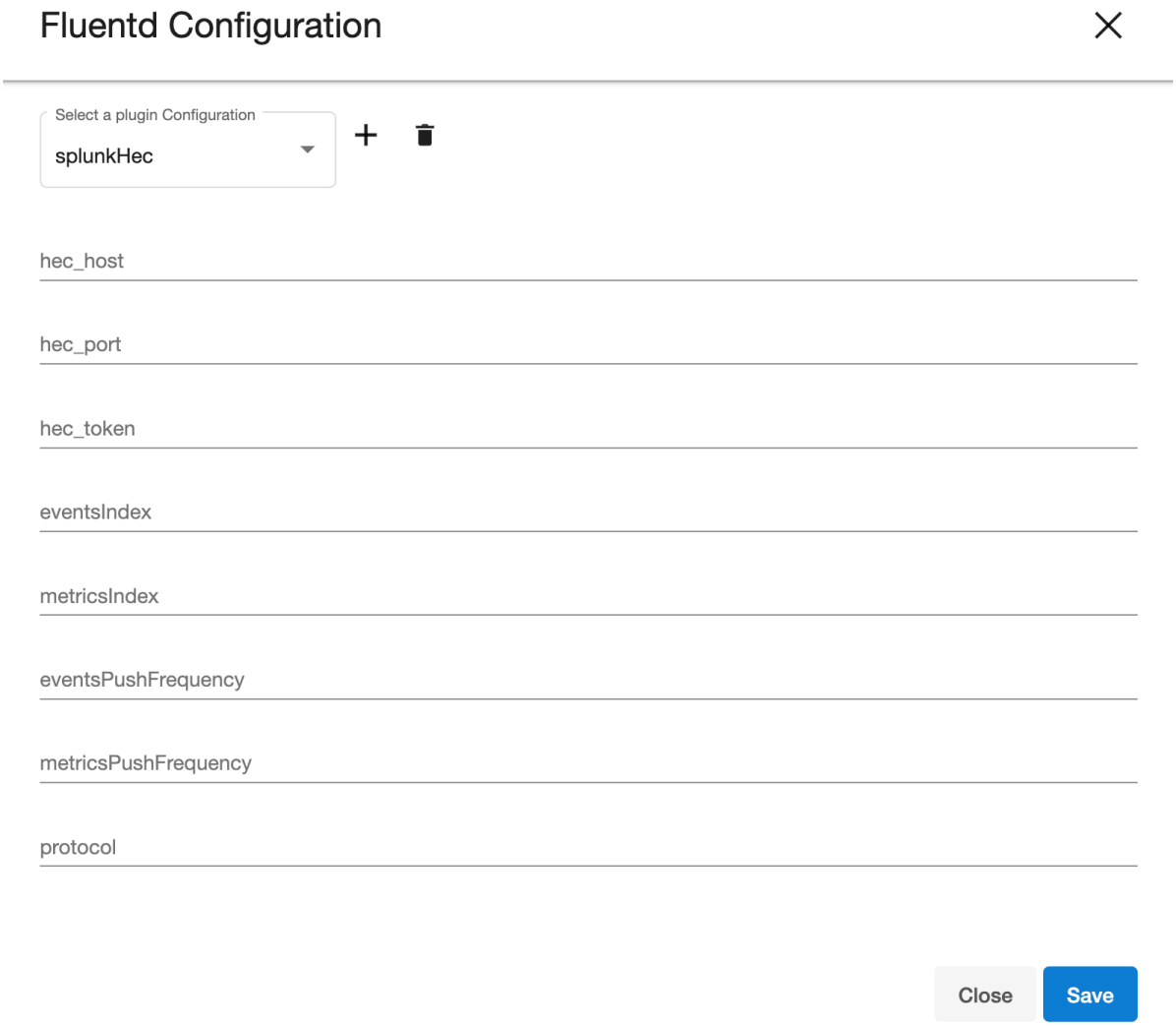
The following table provides an example of the data you will need.

Attribute	Sample data
Splunk Server IP address	192.168.8.8
Splunk Server HEC Port Number	8088
Splunk HEC Token	12345678-1234-1234-1234-1234567890AB
Index Name for Events	delphix_events
Index Name for Metrics	delphix_metrics

252 http://docs.splunk.com/Documentation/Splunk/7.1.1/Data/UsetheHTTPEventCollector#Configure_HTTP_Event_Collector_on_Splunk_Enterprise

6.7.12.3.1.2 Configuring Delphix for Splunk

1. Log in to the **Delphix Server Setup UI** as the sysadmin.
2. From the **Preferences** menu select **Fluentd Configuration**.
3. In the Fluentd Configuration window, enter your Splunk values, using the default **splunkHec** plugin configuration.



Host	Splunk hostname or IP address
HEC Port	The TCP port number for the Splunk HTTP Event Collector (HEC)
HEC Token	The token for the Splunk HTTP Event Collector (HEC)

Host	Splunk hostname or IP address
Events Index	The Splunk Index events will be sent to. It must be set as an allowed index for the HEC token.
Metrics Index	The Splunk Index metrics will be sent to. If none is specified then the Main Index will be used for metrics as well. It must be set as an allowed index for the HEC token.
Events Push Frequency	The frequency at which the Events will be pushed to Splunk. Specified in seconds.
Metrics Push Frequency	The frequency at which the Performance Metrics will be pushed to Splunk. Specified in seconds
Protocol	What protocol to use (HTTP or HTTPS) when connecting to Splunk. Must match your HTTP Event Collector settings in Splunk.

4. Click **Save** to enable the Splunk configuration and begin sending all new Actions, Job Events, Faults, Alerts, and Metrics to your Splunk instance.

6.7.12.3.1.3 Testing Your Connection to Splunk

Delphix will send a test event to Splunk once you have clicked **Save**. If this test does not receive an OK response, the configuration will not be saved and you will be prompted to fix the issue. There is no need to separately test the connection when using the Delphix UI. However, there is also an option for explicitly testing your SplunkHec configuration via the CLI at `service fluentd configuration testSplunkHec` or by using the API at the `/delphix/service/fluentd/configuration/testSplunkHec` endpoint.

6.7.12.3.2 Using search

Use the search to analyze your data and enumerate items in a metrics index. For more about searching a metrics index, refer to the Splunk[253] documentation.

6.7.12.3.2.1 Search examples - Metrics

The following examples provide information on viewing Metrics on Splunk 7.x

To get a list of all Metrics:

253 http://docs.splunk.com/Documentation/Splunk/7.0.3/Metrics/Search

```
| mcatalog values(metric_name)
```

To get a list of all dimensions of a given metric - say CPU utilization percentage:

```
| mcatalog values(_dims) where metric_name="system.cpu.util.pct"
```

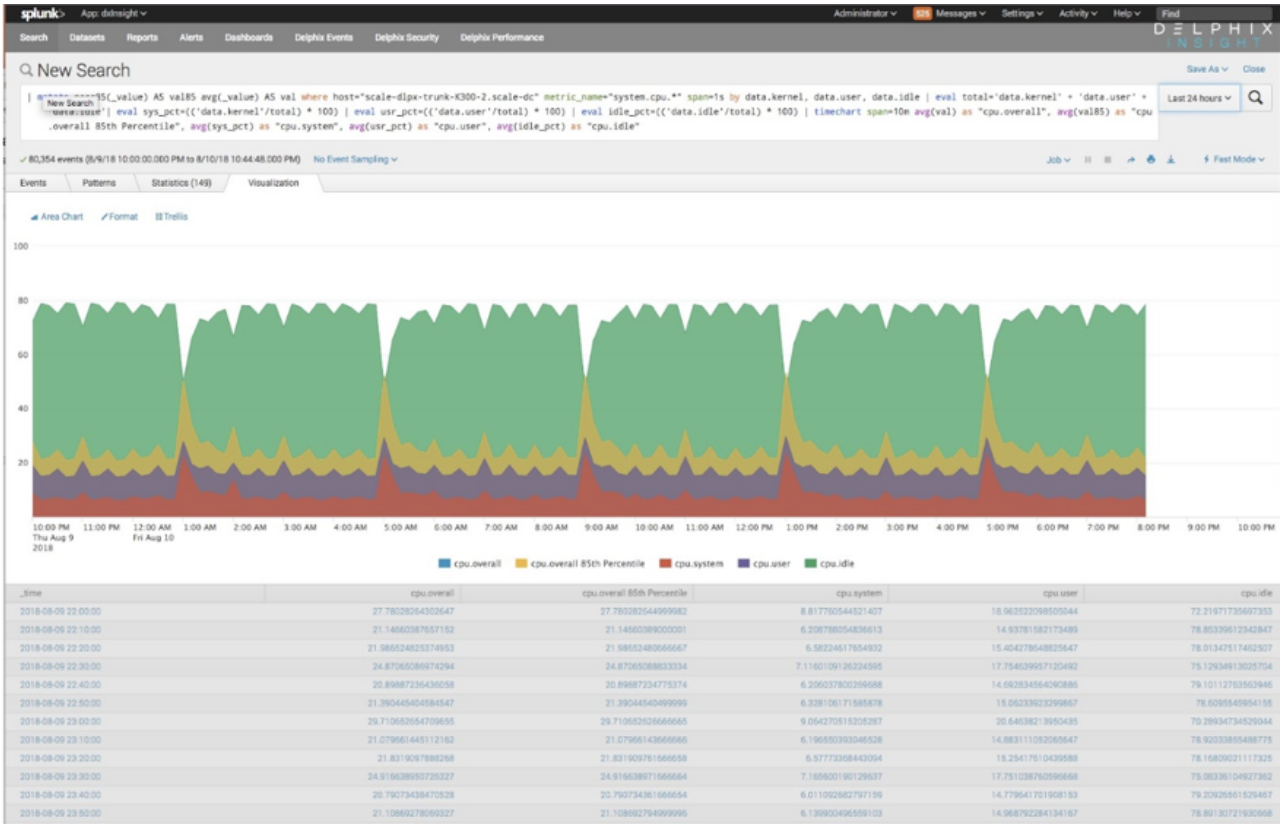
To view the average values of overall CPU utilization percentage across all hosts with a span of 30 seconds:

```
| mstats avg(_value) WHERE index=delphix_metrics AND metric_name=system.cpu.util.pct
span=30s
```

You can also display results in a chart with CPU wildcard:

```
| mstats perc85(_value) AS val85 avg(_value) AS val where metric_name="system.cpu.*"
 span=1s by data.kernel, data.user, data.idle
| eval total='data.kernel' + 'data.user' + 'data.idle'
| eval sys_pct=(('data.kernel'/total) * 100)
| eval usr_pct=(('data.user'/total) * 100)
| eval idle_pct=(('data.idle'/total) * 100)
| timechart span=10m avg(val) as "cpu.overall", avg(val85) as "cpu.overall 85th
Percentile", avg(sys_pct) as "cpu.system", avg(usr_pct) as "cpu.user", avg(idle_pct)
as "cpu.idle"
```

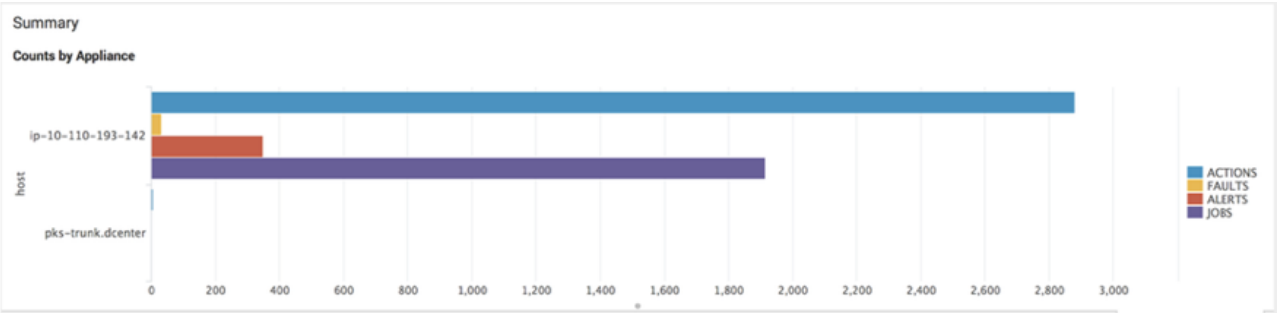
This type of search can be used to stack different CPU metrics that add up to 100%. Here is a sample screenshot of the above “stack different CPU metrics” from the Delphix Engines.



6.7.12.3.2.2 Search examples - events

The following queries demonstrate some basic visualizations of various Delphix events. The **delphix _index** value should be replaced with the name of the **Main Index** provided in during Delphix Setup. These examples serve as useful starting points that can be expanded to include other relevant data. See Events Format (see page 0) for a full description of the structure of each type of event.

Display event statistics per host



Search

```
index="delphix_index" | stats
count(eval(source="delphix.events.action.completed" OR
source="delphix.events.action.started" OR
```

```
source="delphix.events.action.waiting" )) AS ACTIONS
count(eval(source="delphix.events.fault.posted")) AS FAULTS
count(eval(source="delphix.events.alert" )) AS ALERTS
count(eval(source="delphix.events.job.event" )) AS JOBS BY host
```

List actions in descending order by the duration

Job Run Times

Jobs by Duration

reference	title	duration
ACTION-235	DB_EXPORT	231
ACTION-42	ENVIRONMENT_DISCOVER	211
ACTION-68	ENVIRONMENT_CREATE_AND_DISCOVER	206
ACTION-70	ENVIRONMENT_CREATE	176
ACTION-160	DB_SYNC	173
ACTION-72	HOST_ADD	165
ACTION-167	DB_REFRESH	161
ACTION-162	DB_PROVISION	132
ACTION-86	ENVIRONMENT_REFRESH_AND_DISCOVER	118
ACTION-146	DB_REFRESH	108

« prev 1 2 3 4 5 6 7 8 9 10 next »

Search

```
index="delphix_index" source="delphix.events.action.*" | transaction reference |
table reference title duration | sort duration
```

Lists faults

Faults

host	dateDiagnosed	details	reference
ip-10-110-193-142	2018-07-05T23:34:37.968Z	The TCP sunrpc.tcp_slot_table_entries property is currently set to '16' which is below the recommended minimum value of 128.	FAULT-35
ip-10-110-193-142	2018-07-05T23:34:37.902Z	The default send buffer that can be allocated for a TCP socket dictated by the second value in the net.ipv4.tcp_wmem property is currently set to '16384' which is below the recommended default of 4194304.	FAULT-34
ip-10-110-193-142	2018-07-05T23:34:37.612Z	The default receive buffer that can be allocated for a TCP socket dictated by the second value in the net.ipv4.tcp_rmem property is currently set to '87380' which is below the recommended default of 16777216.	FAULT-33
ip-10-110-193-142	2018-07-05T23:20:45.107Z	The owner of the Delphix toolkit installation directory is not 'sybase'.	FAULT-32
ip-10-110-193-142	2018-07-05T23:19:19.827Z	The owner of the Delphix toolkit installation directory is not 'sybase'.	FAULT-31

« prev 1 2 3 4 5 6 7 next »

Search

```
index="delphix_index"   source="delphix.events.fault.posted" | table host
dateDiagnosed details reference
```

Completed jobs

Events with "Job Complete" Status

host	timestamp	messageDetails	parentAction
ip-10-110-193-142	2018-07-05T23:36:50.018Z	SUPPORT_BUNDLE_DOWNLOAD job for "delphix_admin" completed successfully.	ACTION-1585
ip-10-110-193-142	2018-07-05T23:27:43.114Z	DB_UNDO job for "Untitled/dbdhcp3--dbdh_WSH-1530832953092" completed successfully.	ACTION-1386
ip-10-110-193-142	2018-07-05T23:27:14.739Z	SOURCE_STOP job for "dbdhcp3--dbdh_WSH-1530832953092" completed successfully.	ACTION-1387
ip-10-110-193-142	2018-07-05T23:27:02.994Z	DB_REFRESH job for "Untitled/dbdhcp3--dbdh_WSH-1530832953092" completed successfully.	ACTION-1383
ip-10-110-193-142	2018-07-05T23:27:02.977Z	DB_SYNC job for "Untitled/dbdhcp3--dbdh_WSH-1530832953092" completed successfully.	ACTION-1385

« prev 1 2 3 4 5 6 7 8 9 10 next »

Search

```
index="delphix_index"   source="delphix.events.job.event"| spath jobState | search
jobState=COMPLETED | table host timestamp messageDetails parentAction
```

Lists faults

Faults

host	dateDiagnosed	details	reference
ip-10-110-193-142	2018-07-05T23:34:37.968Z	The TCP sunrpc.tcp_slot_table_entries property is currently set to '16' which is below the recommended minimum value of 128.	FAULT-35
ip-10-110-193-142	2018-07-05T23:34:37.902Z	The default send buffer that can be allocated for a TCP socket dictated by the second value in the net.ipv4.tcp_wmem property is currently set to '16384' which is below the recommended default of 4194304.	FAULT-34
ip-10-110-193-142	2018-07-05T23:34:37.612Z	The default receive buffer that can be allocated for a TCP socket dictated by the second value in the net.ipv4.tcp_rmem property is currently set to '87380' which is below the recommended default of 16777216.	FAULT-33
ip-10-110-193-142	2018-07-05T23:20:45.107Z	The owner of the Delphix toolkit installation directory is not 'sybase'.	FAULT-32
ip-10-110-193-142	2018-07-05T23:19:19.827Z	The owner of the Delphix toolkit installation directory is not 'sybase'.	FAULT-31

« prev 1 2 3 4 5 6 7 next »

Search

```
index="delphix_index"   source="delphix.events.fault.posted" | table host
dateDiagnosed details reference
```

Completed jobs

Events with "Job Complete" Status

host	timestamp	messageDetails	parentAction
ip-10-110-193-142	2018-07-05T23:36:50.018Z	SUPPORT_BUNDLE_DOWNLOAD job for "delphix_admin" completed successfully.	ACTION-1585
ip-10-110-193-142	2018-07-05T23:27:43.114Z	DB_UNDO job for "Untitled/dbdhcp3--dbdh_WSH-1530832953092" completed successfully.	ACTION-1386
ip-10-110-193-142	2018-07-05T23:27:14.739Z	SOURCE_STOP job for "dbdhcp3--dbdh_WSH-1530832953092" completed successfully.	ACTION-1387
ip-10-110-193-142	2018-07-05T23:27:02.994Z	DB_REFRESH job for "Untitled/dbdhcp3--dbdh_WSH-1530832953092" completed successfully.	ACTION-1383
ip-10-110-193-142	2018-07-05T23:27:02.977Z	DB_SYNC job for "Untitled/dbdhcp3--dbdh_WSH-1530832953092" completed successfully.	ACTION-1385

« prev 1 2 3 4 5 6 7 8 9 10 next »

Search

```
index="delphix_index"   source="delphix.events.job.event"| spath jobState | search
jobState=COMPLETED | table host timestamp messageDetails parentAction
```

6.7.12.3.2.3 Search examples - event formats

The Actions, Job Events, Faults, and Alerts that Delphix sends to Splunk are structured according to predefined JSON schemas[254]

JSON schemas for events

The following set of JSON schemas[255] define the shape of each Splunk event, including which properties are expected to exist for each event type and what those properties mean. Some of these .json files are used as shared building blocks to define the other schemas; the top-level schemas which define each distinct event type are **Action.json**, **Alert.json**, **JobEvent.json**, **Fault.json**, and **FaultEffect.json.**

Below key-values are Splunk event metadata - This follows GeneralSplunkHeader.json - schema and GeneralSplunkEvent.json
GeneralSplunkHeader.json

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "time": {
            "description": "The time the event was logged. The default time format is
epoch time format, in the format <sec>.<ms>.",
```

254 http://json-schema.org/
255 http://json-schema.org/

```
            "type": "number"
        },
        "host": {
            "description": "The system's hostname. Will be the host value assigned to
the event data in Splunk.",
            "type": "string"
        },
        "source": {
            "description": "For example mgmt.event.action. Will be the source value
to assigned to the event data in Splunk.",
            "type": "string"
        },
        "sourcetype": {
            "description": "The sourcetype value to assign to the event data.",
            "type": "string"
        },
        "index": {
            "description": "The name of the index by which the event data is to be
indexed.",
            "type": "string"
        }
    },
    "required": ["time", "host", "source", "sourcetype", "index"]
}
```

GeneralSplunkEvent.json

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "properties": {
        "systemUniqueId": {
            "description": "The UUID of the system.",
            "type": "string"
        },
        "systemVersion": {
            "description": "The release version of the system.",
            "type": "string"
        }
    },
    "required": ["systemUniqueId", "systemVersion"]
}
```

Action.json

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "definitions": {
        "ActionEvent": {
            "type": "object",
```

```
            "properties": {
                "reference": {
                    "description": "The object reference of the action.",
                    "type": "string"
                },
                "title": {
                    "description": "Action title.",
                    "type": "string"
                },
                "details": {
                    "description": "Plain text description of the action.",
                    "type": "string"
                },
                "startTime": {
                    "description": "The time the action occurred. For a long running
process, this represents the starting time.",
                    "type": "string"
                },
                "endTime": {
                    "description": "The time the action completed.",
                    "type": "string"
                },
                "user": {
                    "description": "The user who initiated the action.",
                    "type": "string"
                },
                "userAgent": {
                    "description": "Name of the client software used to initiate the
action.",
                    "type": "string"
                },
                "originIp": {
                    "description": "Network address used to initiate the action",
                    "type": "string"
                },
                "parentAction": {
                    "description": "The parent action of this action.",
                    "type": "string"
                },
                "state": {
                    "description": "State of the action",
                    "type": "string"
                },
                "workSource": {
                    "description": "Origin of the work that caused the action.",
                    "type": "string"
                },
                "workSourceName": {
                    "description": "Name of the user or policy that initiated the
action.",
                    "type": "string"
                },
```

```
                "failureDescription": {
                    "description": "Details of the action failure.",
                    "type": "string"
                },
                "failureAction": {
                    "description": "Action to be taken to resolve the failure",
                    "type": "string"
                },
                "failureMessageCode": {
                    "description": "Message ID associated with the event.",
                    "type": "string"
                }
            },
            "required": ["reference", "title", "details", "state"]
        }
    },
    "type": "object",
    "title": "Action",
    "allOf": [{ "$ref": "GeneralSplunkHeader.json#" }],
    "properties": {
        "event": {
            "type": "object",
            "allOf": [{ "$ref": "#/definitions/ActionEvent" }, { "$ref":
"GeneralSplunkEvent.json#" }]
        }
    }
}
```

Alert.json

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "definitions": {
        "AlertEvent": {
            "type": "object",
            "properties": {
                "reference": {
                    "description": "The object reference of the alert.",
                    "type": "string"
                },
                "title": {
                    "description": "Title of the event which triggered the alert.",
                    "type": "string"
                },
                "code": {
                    "description": "Dotted descriptor of the type of event which
triggered the alert.",
                    "type": "string"
                },
                "eventSeverity": {
                    "description": "The severity of the event.",
```

```
                    "type": "string"
                },
                "details": {
                    "description": "Plain text description of the event which
triggered the alert.",
                    "type": "string"
                },
                "response": {
                    "description": "Automated response, if any, taken by the system.",
                    "type": "string"
                },
                "timestamp": {
                    "description": "The time the alert occurred.",
                    "type": "string"
                },
                "target": {
                    "description": "Reference to the target object.",
                    "type": "string"
                },
                "targetName": {
                    "description": "Name of the target object.",
                    "type": "string"
                }
            },
            "required": ["reference", "title", "code", "details", "timestamp"]
        }
    },

    "type": "object",
    "title": "Alert",
    "allOf": [{ "$ref": "GeneralSplunkHeader.json#" }],
    "properties": {
        "event": {
            "type": "object",
            "allOf": [{ "$ref": "#/definitions/AlertEvent" }, { "$ref":
"GeneralSplunkEvent.json#" }]
        }
    }
}
```

JobEvent.json

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "definitions": {
        "JobEvent": {
            "type": "object",
            "properties": {
                "job": {
                    "description": "The object reference of job associated with this
event.",
```

```
                    "type": "string"
                },
                "parentAction": {
                    "description": "The object reference of the parent action of the
associated job.",
                    "type": "string"
                },
                "jobState": {
                    "description": "The new state of the job.",
                    "type": "string"
                },
                "timestamp": {
                    "description": "The time the event occurred.",
                    "type": "string"
                },
                "percentComplete": {
                    "description": "Completion percentage of the job associated with
this event",
                    "type": "number"
                },
                "diagnoses": {
                    "description": "If job failed, a set of diagnoses of things that
may have caused the failure.",
                    "type": "array",
                    "items": { "$ref": "#/definitions/DiagnosisResult" }
                },
                "eventType": {
                    "description": "The type of this event (info, warning, or
error).",
                    "type": "string"
                },
                "messageCode": {
                    "description": "A message code describing this event.",
                    "type": "string"
                },
                "messageDetails": {
                    "description": "A message describing the details of this event.",
                    "type": "string"
                },
                "messageAction": {
                    "description": "Action to be taken by the user to repair or
remedy the situation.",
                    "type": "string"
                },
                "messageCommandOutput": {
                    "description": "Any command output generated by this event",
                    "type": "string"
                }
            },
            "required": ["job", "parentAction", "timestamp", "percentComplete",
"messageCode", "messageDetails", "eventType"]
        },
```

```
        "DiagnosisResult": {
            "type": "object",
            "properties": {
                "diagnosisCode": {
                    "description": "Message code associated with this diagnosis
check.",
                    "type": "string"
                },
                "diagnosisMessage": {
                    "description": "Description of this diagnosis check.",
                    "type": "string"
                },
                "failed": {
                    "description": "True if this diagnosis check did not pass.",
                    "type": "boolean"
                },
                "targetReference": {
                    "description": "Reference of the target object of this diagnosis
check, if applicable.",
                    "type": "string"
                }
            },
            "required": ["diagnosisCode", "diagnosisMessage", "failed"]
        }
    },
    "type": "object",
    "title": "JobEvent",
    "allOf": [{ "$ref": "GeneralSplunkHeader.json#" }],
    "properties": {
        "event": {
            "type": "object",
            "allOf": [{ "$ref": "#/definitions/JobEvent" }, { "$ref":
"GeneralSplunkEvent.json#" }]
        }
    }
}
```

AbstractFault.json

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "title": "AbstractFault",
    "properties": {
        "reference": {
            "description": "The object reference of the fault.",
            "type": "string"
        },
        "title": {
            "description": "Title of the event which triggered the fault.",
            "type": "string"
```

```
        },
        "code": {
            "description": "Dotted descriptor of the type of event which triggered
the fault.",
            "type": "string"
        },
        "details": {
            "description": "Plain text description of the event which triggered the
fault.",
            "type": "string"
        },
        "response": {
            "description": "Automated response, if any, taken by the system.",
            "type": "string"
        },
        "dateDiagnosed": {
            "description": "The date when the fault was diagnosed.",
            "type": "string"
        },
        "target": {
            "description": "Reference to the target object.",
            "type": "string"
        },
        "targetName": {
            "description": "Name of the target object.",
            "type": "string"
        },
        "state": {
            "description": "The state of the fault."
        },
        "eventSeverity": {
            "description": "The severity of the event.",
            "type": "string"
        }
    },

    "required": ["reference", "title", "details", "state", "target", "dateDiagnosed"]
}
```

Fault.json

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "definitions": {
        "FaultEvent": {
            "type": "object",
            "properties": {
                "dateResolved": {
                    "description": "The date when the fault was resolved.",
                    "type": "string"
                },
```

```
                "resolutionComments": {
                    "description": "Comments regarding the resolution of the fault.",
                    "type": "string"
                }
            }
        }
    },

    "type": "object",
    "title": "Fault",
    "allOf": [{ "$ref": "GeneralSplunkHeader.json#" }],
    "properties": {
        "event": {
            "type": "object",
            "allOf": [{ "$ref": "AbstractFault.json#" }, { "$ref": "#/definitions/
FaultEvent" }, { "$ref": "GeneralSplunkEvent.json#" }]
        }
    }
}
```

FaultEffect.json

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "definitions": {
        "AlertEvent": {
            "type": "object",
            "properties": {
                "reference": {
                    "description": "The object reference of the alert.",
                    "type": "string"
                },
                "title": {
                    "description": "Title of the event which triggered the alert.",
                    "type": "string"
                },
                "code": {
                    "description": "Dotted descriptor of the type of event which
triggered the alert.",
                    "type": "string"
                },
                "eventSeverity": {
                    "description": "The severity of the event.",
                    "type": "string"
                },
                "details": {
                    "description": "Plain text description of the event which
triggered the alert.",
                    "type": "string"
                },
                "response": {
```

```
                    "description": "Automated response, if any, taken by the system.",
                    "type": "string"
                },
                "timestamp": {
                    "description": "The time the alert occurred.",
                    "type": "string"
                },
                "target": {
                    "description": "Reference to the target object.",
                    "type": "string"
                },
                "targetName": {
                    "description": "Name of the target object.",
                    "type": "string"
                }
            },
            "required": ["reference", "title", "code", "details", "timestamp"]
        }
    },

    "type": "object",
    "title": "Alert",
    "allOf": [{ "$ref": "GeneralSplunkHeader.json#" }],
    "properties": {
        "event": {
            "type": "object",
            "allOf": [{ "$ref": "#/definitions/AlertEvent" }, { "$ref":
"GeneralSplunkEvent.json#" }]
        }
    }
}
```

6.7.12.3.2.4 Types of events

Delphix uses the **source** field in Splunk to designate the type of each event. Here is a full list of the possible values of the **source** field for events, along with an explanation of when each event is generated and the name of the corresponding JSON schema that describes the event structure.

source	Explanation	Schema
delphix.events.action.started	Action has started running.	Action.json
delphix.events.action.waiting	Action has moved to the WAITING state.	Action.json
delphix.events.action.completed	Action has completed successfully.	Action.json
delphix.events.action.failed	Action has failed.	Action.json

source	Explanation	Schema
delphix.events.action.canceled	Action has been canceled.	Action.json
delphix.events.job.event	Job Event has been generated in response to a Job progress update.	JobEvent.json
delphix.events.fault.posted	Fault has been posted.	Fault.json
delphix.events.fault.resolved	Fault has been resolved.	Fault.json
delphix.events.fault.ignored	User has chosen to ignore a fault.	Fault.json
delphix.events.fault.unignored	User has chosen to “unignore” a previously ignored fault.	Fault.json
delphix.events.fault.effect.posted	Fault Effect has been posted as a downstream effect of some Fault.	FaultEffect.json
delphix.events.fault.effect.resolved	Fault Effect has been resolved as a result of its cause being resolved.	FaultEffect.json
delphix.events.fault.effect.ignored	Fault Effect has been ignored when a user chose to ignore its cause.	FaultEffect.json
delphix.events.fault.effect.unignored	Fault Effect has been “unignored” when a user chose to unignore its previously ignored cause.	FaultEffect.json
delphix.events.alert	Alert has been posted.	Alert.json

6.7.12.3.2.5 Search examples - metric formats

JSON schemas for metrics

GeneralMetricEvent.json

```
{
   "$schema": "http://json-schema.org/draft-04/schema#",
```

```
    "type": "object",
    "properties": {
        "systemUniqueId": {
            "description": "The UUID of the system.",
            "type": "string"
        },
        "systemVersion": {
            "description": "The release version of the system.",
            "type": "string"
        },
        "event": {
           "description": "A tag that indicates this Event is a metric.",
           "type": "string",
           "enum": ["metric"]
        }
    },
    "required": ["systemUniqueId", "systemVersion", "event"]
}
```

A schema that is common for all metrics, the “data” nested JSON object contents varies depending on the metric:

CommonMetric.json

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "definitions": {
        "CommonMetric": {
            "type": "object",
            "properties": {
                "name": {
                    "description": "The name of the metric.",
                    "type": "string"
                },
                "time": {
                    "description": "The timestamp of the metric when it was collected.
The default time format is epoch time format, in the format <sec>.<ms>.",
                    "type": "number"
                },
                "value": {
                    "description": "The numeric value of the metric.",
                    "type": "number"
                },
                "type": {
                    "description": "The type of measurement.",
                    "type": "string",
                    "enum": ["counter", "g", "value", "summary"]
                },
                "data": {
                    "description": "JSON object having further details on the metric.
Contents depends on the actual metric.",
                    "type": "object"
```

```
                }
            }
        }
    },

    "type": "object",
    "title": "CommonMetric",
    "allOf": [{ "$ref": "GeneralSplunkHeader.json#" }],
    "properties": {
        "event": {
            "type": "object",
            "allOf": [{ "$ref": "#/definitions/CommonMetric" }, { "$ref":
"GeneralMetricEvent.json#" }]
        }
    }
}
```

6.7.12.3.2.6 Metrics format

Metrics Format is the combination of the two tables below (metadata + metric specific key-values).

JSON key	Value type	Description	Example	Comments
source	String	Dotted name hierarchy for insight source.	delphix.metrics.xyz	This is the “source” value assigned to an event data in Splunk.
index	String	Splunk metrics index name	insight_metrics	Splunk Index name
host	String	Hostname/IP	pks-insight.dc2.delphix.com	Could also serve as a tag in other time-series data (like opentsdb)
event	***"metric"***	Describes what kind of event this is.	n/a	Signifies this Splunk event is a “Metric”.
sourcetype	***“_json”***	Data format	n/a	Used for Splunk Indexed field extractions
systemUniqueId	String	The UUID of the system	"423f22db-4ee9-6ebe-ff0f-884ffdc351f7"	

JSON key	Value type	Description	Example	Comments
systemVersion	String	The release version of the system	5.2.5.0	

And the general key-values specific to an Insight metric are:

JSON key	Value type	Description	Example	Comments
name	String	Pseudo hierarchical dotted format of the metric name.	system.cpu.util.pct system.disk.ops.count system.net[256].total.bytes	A metric name: • is alphanumeric (underscore & dot allowed) • has *prefix* that points to the source of the data (like *system.cpu*) • has *suffix* that describes the unit (when it can) • an aggregate/summary metric would have *total* as part of suffix for example.
time	UNIX time notation (epoch)	Timestamp	1525399950	UNIX epoch time format.
value	Numeric value of the metric	The actual measurement	85.17 (system.cpu.util.pct)	Numeric only

256 https://system.net/

JSON key	Value type	Description	Example	Comments
type	String (Enum): • counter, • gauge, • summary	Type of the measurement	counter | gauge | summary	This field could be used to identify what kind of value a metric is presenting (like a “gauge” for cpu metric implies the values will fall into a range of 0-100% - similarly a “summary” would imply the value is an “accumulated value” like network utilization)
data	JSON Object	Dimensions of the metric (if any).	{ "read_latency": 0.98, "ops_write": 20, "ops_read": 30, "write_latency": 0.55, }	Has additional info about a metric (called “Dimensions” in Splunk) Has different key-vals depending on a given metric Like for CPU: it can have key-vals to like “kernel”, “user” For disk: can have latency/ throughput, operation name

6.7.12.3.2.7 Summary of all the current metrics

Metric name	Metric value	Metric dimensions (data.xxx below)
system.cpu.util.pct	cpu utilization percentage aggregated ***across all cpus***	user, kernel, idle
system.disk.ops.count	disk read/write operations aggregated ***across all disk instances***	count, op (read/write), latency, avgLatency, throughput
system.net[257]***.total.bytes***	network utilization - total bytes (in + out) of a given network interface	networkInterface, inBytes, outBytes, inPackets, outPackets
system.nfs.ops.count	nfs read/write operations aggregated ***across all instances***	count, op (read/write), latency, avgLatency, throughput

257 https://system.net/

Metric name	Metric value	Metric dimensions (data.xxx below)
system.iscsi.ops.count	iscsi read/write operations aggregated ***across all instances***	count, op (read/write), latency, avgLatency, throughput
system.tcp.total.bytes	tcp connection statistics aggregated by service (nfs/iscsi/dsp etc)	congestionWindowSize, inBytes, inUnorderedBytes, localAddress, outBytes, receiveWindowSize, remoteAddress, retransmittedBytes, roundTripTime, sendWindowSize, service unacknowledgedBytes, unsentBytes
system.dataset.total.bytes	dataset performance - total number of bytes read + written per dataset (dsource/vdb etc)	dataset, nread, nwritten, type (virtual, dsource, staging)
system.capacity.source.size	overall system capacity - actual used space in bytes	actualSpace, totalSpace, activeSpace, actualSpace, descendantSpace, logSpace, manualSpace, policySpace, syncSpace, timeflowUnvirtualizedSpace, unownedSnapshotSpace, unvirtualizedSpace
system.capacity.consumer.size	consumer capacity - actual used space in bytes per consumer (vdb/pdb etc)	syncSpace, descendantSpace, activeSpace, timeflowUnvirtualizedSpace, objectType, dSource, actualSpace, groupName, manualSpace, unownedSnapshotSpace, unvirtualizedSpace, logSpace, policySpace, consumerName

6.7.12.4 Creating Fluentd plugins

The Delphix Fluentd service includes built-in support for sending data to Splunk, but the feature can be extended to other data consumers through the use of plugins that can be uploaded to a Delphix engine for use by Fluentd. These plugins can be adopted from various widely available options, such as those found at http://rubygems.org, but only require a few specific gem files and a configuration file to tell Fluentd where and how to send metrics to your platform of choice.

6.7.12.4.1 Delphix Fluentd plugin structure

The expected file structure of a Delphix Fluentd plugin is simple: A root folder containing a properly configured `fluent.conf.stg` file, and a /gems subfolder with the necessary .gem files. Delphix Fluentd plugins are simpler than a typical Fluentd plugin because of Delphix's native Fluentd integration, which comes with a variety of Ruby gems that are used by many standard plugins. Additionally, the Delphix Fluentd service requires only the dependency gem files (i.e. the files ending with a .gem extension). Additional files that come with a downloaded plugin should be removed.

6.7.12.4.2 Pre-installed Fluentd ruby gems

To minimize the surface area of uploads to a given Delphix engine, all .gem files that are already present on the Delphix OS should also be removed from your plugin before upload. Here is a list of the provided gems that should be removed from your /gems folder, if present:

```
addressable (2.8.1)
async (1.30.3)
async-http (0.59.2)
async-io (1.34.0)
async-pool (0.3.12)
aws-eventstream (1.2.0)
aws-partitions (1.650.0)
aws-sdk-core (3.164.0)
aws-sdk-kms (1.58.0)
aws-sdk-s3 (1.116.0)
aws-sdk-sqs (1.51.1)
aws-sigv4 (1.5.2)
benchmark (default: 0.1.0)
bigdecimal (default: 2.0.0)
bindata (2.4.14)
bundler (2.3.18, default: 2.1.4)
cgi (default: 0.1.0.1)
cmetrics (0.3.3)
concurrent-ruby (1.1.10)
console (1.16.2)
cool.io (1.7.1)
csv (default: 3.1.2)
date (default: 3.0.3)
delegate (default: 0.1.0)
did_you_mean (default: 1.4.0)
digest-crc (0.6.4)
digest-murmurhash (1.1.1)
etc (default: 1.1.0)
excon (0.93.1)
faraday (1.10.2)
faraday-em_http (1.0.0)
faraday-em_synchrony (1.0.0)
faraday-excon (1.1.0)
```

```
faraday-httpclient (1.0.1)
faraday-multipart (1.0.4)
faraday-net_http (1.0.1)
faraday-net_http_persistent (1.2.0)
faraday-patron (1.0.0)
faraday-rack (1.0.0)
faraday-retry (1.0.3)
faraday_middleware-aws-sigv4 (0.6.1)
fcntl (default: 1.0.0)
ffi (1.15.5)
fiber-local (1.0.0)
fiddle (default: 1.0.0)
fileutils (1.6.0, default: 1.4.1)
fluent-config-regexp-type (1.0.0)
fluent-diagtool (1.0.1)
fluent-logger (0.9.0)
fluent-plugin-calyptia-monitoring (0.1.3)
fluent-plugin-flowcounter-simple (0.1.0)
fluent-plugin-kafka (0.18.1)
fluent-plugin-metrics-cmetrics (0.1.2)
fluent-plugin-opensearch (1.0.8)
fluent-plugin-prometheus (2.0.3)
fluent-plugin-prometheus_pushgateway (0.1.0)
fluent-plugin-record-modifier (2.1.1)
fluent-plugin-rewrite-tag-filter (2.4.0)
fluent-plugin-s3 (1.7.2)
fluent-plugin-sd-dns (0.1.0)
fluent-plugin-systemd (1.0.5)
fluent-plugin-td (1.2.0)
fluent-plugin-utmpx (0.5.0)
fluent-plugin-webhdfs (1.5.0)
fluentd (1.15.3)
forwardable (default: 1.3.1)
getoptlong (default: 0.1.0)
hirb (0.7.3)
http_parser.rb (0.8.0)
httpclient (2.8.3)
io-console (default: 0.5.6)
ipaddr (default: 1.2.2)
irb (default: 1.2.6)
jmespath (1.6.1)
json (2.6.2, default: 2.3.0)
linux-utmpx (0.3.0)
logger (default: 1.4.2)
ltsv (0.1.2)
matrix (default: 0.2.0)
mini_portile2 (2.8.0)
minitest (5.13.0)
msgpack (1.6.0)
multi_json (1.15.0)
multipart-post (2.2.3)
mutex_m (default: 0.1.0)
```

```
net-pop (default: 0.1.0)
net-smtp (default: 0.1.0)
net-telnet (0.2.0)
nio4r (2.5.8)
observer (default: 0.1.0)
oj (3.13.17)
open3 (default: 0.1.0)
opensearch-api (2.0.2)
opensearch-ruby (2.0.3)
opensearch-transport (2.0.1)
openssl (default: 2.1.3)
ostruct (default: 0.2.0)
parallel (1.22.1)
power_assert (1.1.7)
prime (default: 0.1.1)
prometheus-client (2.1.0)
protocol-hpack (1.4.2)
protocol-http (0.23.12)
protocol-http1 (0.14.6)
protocol-http2 (0.14.2)
pstore (default: 0.1.0)
psych (default: 3.1.0)
public_suffix (5.0.0)
racc (default: 1.4.16)
rake (13.0.6, 13.0.1)
rdkafka (0.11.1)
rdoc (default: 6.2.1.1)
readline (default: 0.0.2)
reline (default: 0.1.5)
rexml (default: 3.2.3.1)
rss (default: 0.2.8)
ruby-kafka (1.5.0)
ruby-progressbar (1.11.0)
ruby2_keywords (0.0.5)
rubyzip (1.3.0)
sdbm (default: 1.0.0)
serverengine (2.3.0)
sigdump (0.2.4)
singleton (default: 0.1.0)
stringio (default: 0.1.0)
strptime (0.2.5)
strscan (default: 1.0.3)
systemd-journal (1.4.2)
td (0.16.9)
td-client (1.0.8)
td-logger (0.3.28)
test-unit (3.3.4)
timeout (default: 0.1.0)
timers (4.3.5)
tracer (default: 0.1.0)
traces (0.7.0)
tzinfo (2.0.5)
```

```
tzinfo-data (1.2022.5)
uri (default: 0.10.0)
webhdfs (0.10.2)
webrick (1.7.0, default: 1.6.1)
xmlrpc (0.3.0)
yajl-ruby (1.4.3)
yaml (default: 0.1.0)
zip-zip (0.3)
zlib (default: 1.1.0)
```

Note that many gems have similar names but may still require your plugin to work. For example, `opensearch-api.gem` from the list above is distinct from `opensearch-transport.gem`. While gem version clashes can sometimes occur when there are duplicate gems found during installation, there is often no harm in including additional gem files, but it is recommended to eliminate as many non-essential gems as possible before uploading.

6.7.12.4.3 Setting up a Fluentd configuration file

Besides requiring the necessary gems to connect to a given data consumer, Fluentd needs a configuration file to know where and how to send the data it has received from Delphix. The Fluentd Configuration GUI in Delphix also uses the configuration file to determine what parameters should be displayed to the system administrator during setup. The configuration file should be named `fluent.conf.stg` and must be placed in your plugin's root folder. Each data consumer requires specific syntax to connect with Fluentd. Consult the documentation of your data consumer of choice when creating your `fluent.conf.stg` file. Here is an example configuration file template, which must be customized depending on the requirements of your chosen data provider (`Splunk` , `elasticsearch` , etc.):

```
/*
 * Copyright (c) 2023 Delphix. All rights reserved.
 */

delimiters "^", "^"

/*
 * This template is used by the Delphix management stack to auto-generate the final
configuration used
 * by the internal fluent service. User-editable fields are filled in with
information provided through
 * the GUI or API. Additional params specific to your needs can be substituted for
my_param.
 * buffer_flush_interval is a fluentd parameter that is dynamically populated based
on user input and
 * is provided here as an example.
 */

fluentConfig(my_param, buffer_flush_interval) ::= <<

<system>
```

```
  # equal to -v command line option
  log_level info
  <log>
    format json
    time_format %Y-%m-%dT%H:%M:%S.%NZ
  </log>
</system>

<source>
  @type forward
  port 24224
  bind 127.0.0.1 # only accept connections from localhost
</source>

<match delphix.events.**>
  ^commonFields(tagPrefix="delphix.events", index={^event_index^},
                flushInterval={^buffer_flush_interval^},
                retryTimeout="72h",
                totalLimitSize="10g",
                chunkLimitSize="1m", ...)^
  data_type event
</match>

<match delphix.metrics.**>
  ^commonFields(tagPrefix="delphix.metrics", index={^metrics_index^},
                flushInterval={^buffer_flush_interval^},
                retryTimeout="72h",
                totalLimitSize="10g",
                chunkLimitSize="1m", ...)^
</match>
>>

commonFields(tagPrefix, index, flushInterval, retryTimeout, totalLimitSize,
chunkLimitSize) ::= <<

/* Replace with syntax specific to your data consumer of choice */
@type myDataConsumer

/* Replace with parameters specific to your data consumer */
my_param ^my_param^

source ${tag} # Filled in at runtime by fluentd

<buffer>
  @type file
  path /var/lib/fluent/^tagPrefix^.*.buffer ^! buffer path must be unique for each
match section !^
  chunk_limit_size ^chunkLimitSize^
  total_limit_size ^totalLimitSize^
  flush_interval ^flushInterval^
  retry_timeout ^retryTimeout^
</buffer>
```

```
>>
```

Fluentd defaults to using JSON formatting for its output plugins. Refer to Fluentd documentation for including a formatter plugin in your config should you require an alternative.

6.7.12.4.4 Readying the plugin for upload

Here is an example of what a plugin for `elasticsearch 7` might look like after taking the above steps:

```
$ ls ./elasticsearch-7

fluent.conf.stg gems

$ ls ./elasticsearch-7/gems/

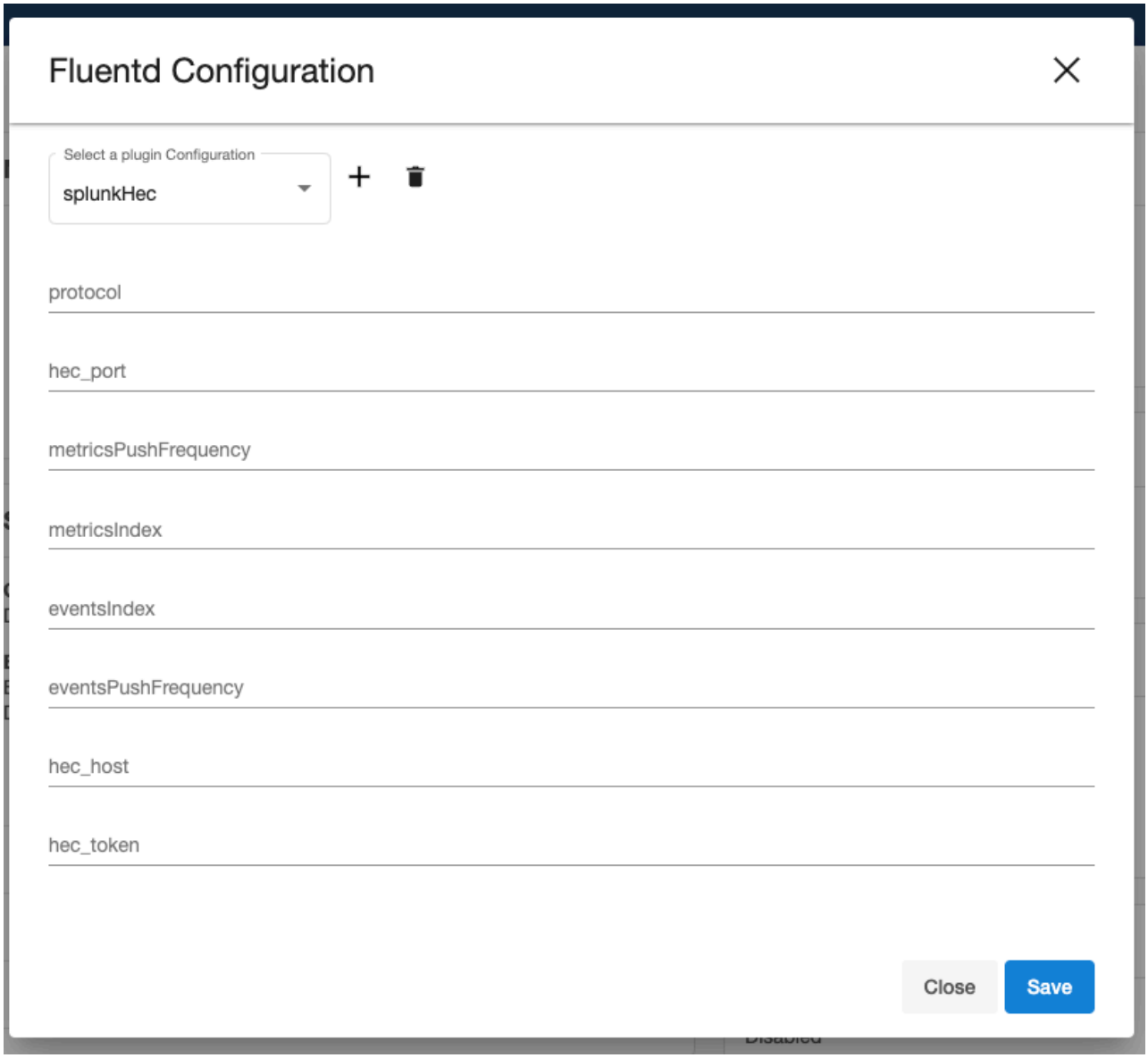
elasticsearch-7.17.1.gem elasticsearch-api-7.17.1.gem elasticsearch-transport-7.17.1.
gem fluent-plugin-elasticsearch-5.1.4.gem
```

Delphix Fluentd plugins must be stored in a .far archive file before uploading them to your engine. You can create a .far archive of your plugin files using a utility such as *tar*. Here is an example *tar* syntax for creating a plugin archive file:

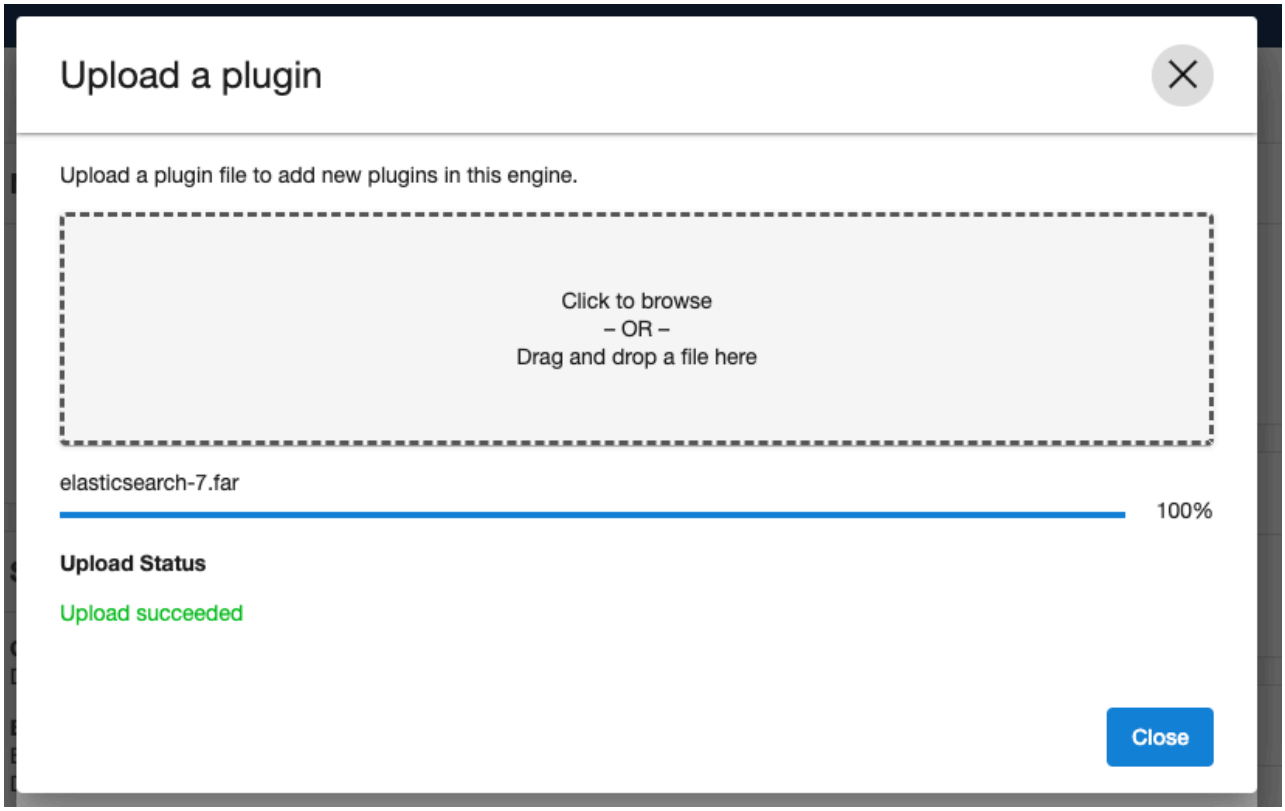
```
tar --create --verbose --owner=0 --group=0 --exclude-backups --exclude-vcs --one-
file-system --format=gnu --file=dlpx-example-plugin.far dlpx-example-plugin/
```

6.7.12.4.5 Uploading plugins to Delphix

Your new Fluentd plugin .far file can be uploaded through the API or the GUI in the Fluentd configuration wizard, by clicking the plus icon next to the plugin selection dropdown menu:

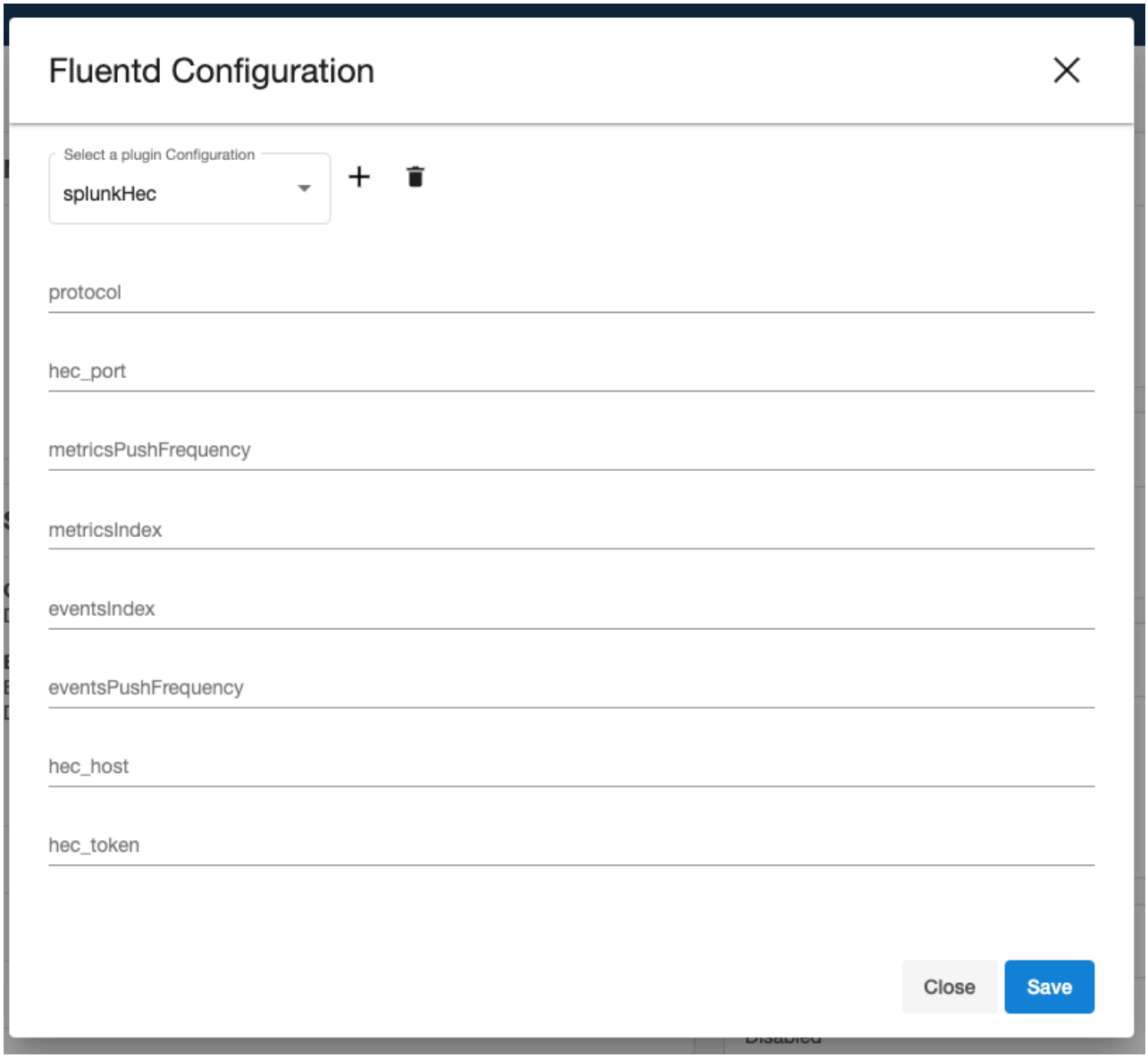


Click the gray box shown in the next window to locate your plugin .far file or drag it to the box to upload it to your Delphix engine:



6.7.12.4.6 Limitations

The Delphix Fluentd service supports sending logs to Splunk instances by default through use of the built-in splunkHec plugin. Only one additional plugin can be uploaded to a given Delphix engine at a time. If logs need to be sent to another service requiring a new plugin, the existing one needs to be removed from the engine, and a new plugin uploaded for the new data consumer. Removing an existing plugin can be done from the same window for uploading the plugin by selecting your plugin from the “Select a plugin Configuration” drop-down menu and clicking the trash can icon next to it:



SSL Verification does not currently support custom certificates.

These limitations may be relaxed in a future Delphix release.

6.8 Performance analytics management

Delphix offers various performance analytics tools to help users monitor throughput, latency, and other key metrics. Learn more about how to leverage these tools and how to architect your Delphix deployment for optimal performance.

This section covers the following topics:

- Performance analytics (see page 869)
- Storage performance configuration options (see page 896)
- Architecture for performance - hypervisors and host (see page 904)

- Target host OS and database configuration options (see page 907)

6.8.1 Performance analytics

This section covers the following topics:

- Performance analytics tool overview (see page 870)
- Working with performance analytics graphs in the graphical user interface (see page 872)
- Performance analytics statistics reference (see page 875)
- Performance analytics tool API reference (see page 876)
- Performance analytics case study: using a single statistic (see page 884)
- Performance analytics case study: using multiple statistics (see page 888)

6.8.1.1 Performance analytics tool overview

This topic describes the Performance Analytics tool and illustrates some basic uses of it.

6.8.1.1.1 Introduction

The performance analytics tool allows introspection into how the Delphix Engine is performing. The introspection techniques it provides are tuned to allow an iterative investigation process, helping to narrow down the cause associated with the performance being measured. Performance analytics information can be accessed through the Delphix Management application, as described in Working with Performance Analytics Graphs in the Graphical User Interface (see page 872), as well as the CLI and the web services API, as described in other topics in this section. The default statistics that are being collected on the Delphix Engine include CPU utilization, network utilization, and disk, NFS, and iSCSI IO operations (see Performance Analytics Statistics Reference (see page 875) for details).

The performance tool operates with two central concepts: **statistics** and **statistic slices**.

6.8.1.1.2 Statistics

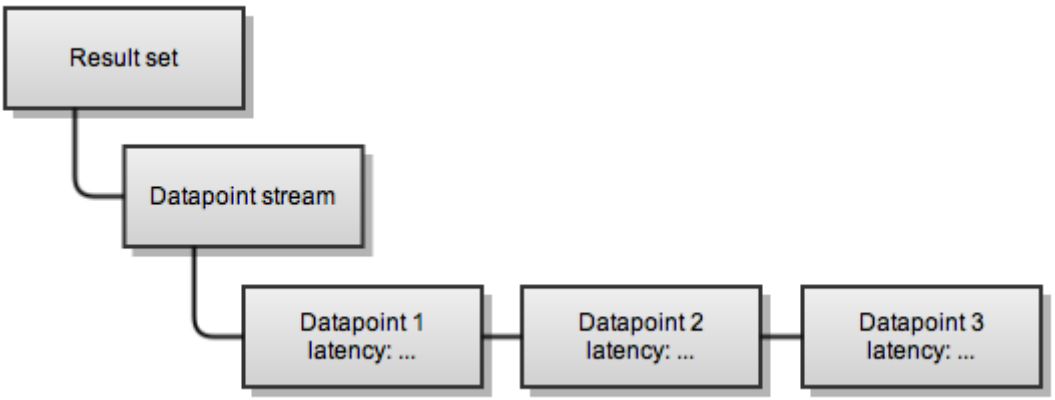
Each statistic describes some data that can be collected from the Delphix Engine. The first piece of information a statistic provides is its **type**, which you will use as a handle when creating a statistic slice. It also gives the **minimum collection interval**, which puts an upper bound on the frequency of data collection. The actual data a statistic can collect is described through a set of **axes**, each of which describes one "dimension" of that statistic. For example, the statistic associated with Network File System (NFS) operations has a **latency axis**, as well as an **operation type axis** (among many others), which allows users to see NFS latencies split by whether they were reads or writes.

Each axis has some important information embedded in it.

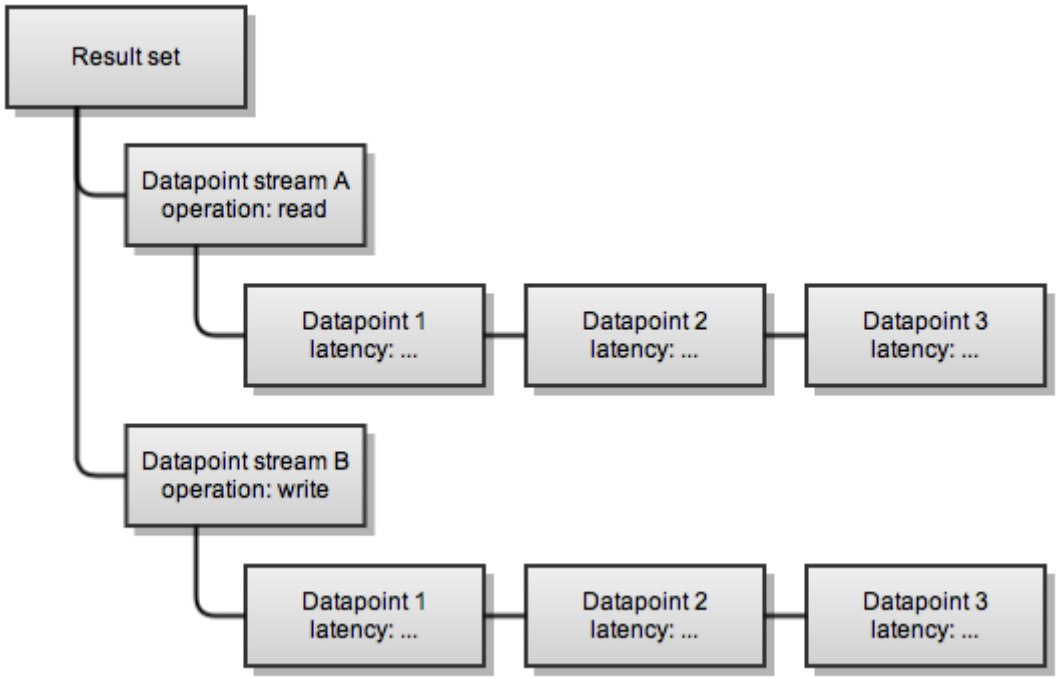
- The **name** of the axis provides a short description of what the axis collects and is used when creating a statistic slice
- A **value** type, which tells you what kind of data will be collected for this axis. The different value types are **integer**, **boolean**, **string**, and **histogram**. The first three are straightforward, but statistic axes with a histogram type can collect a distribution of all the values encountered during each collection interval. This means that instead of seeing an average NFS operation latency every collection interval, you can see a full distribution of operation latencies during that interval. This allows you to see

outliers as well as the average, and observe the effects of caching on the performance of your system more easily.
- A **constraint** type, which is only relevant while creating a statistic slice, and will be described in more detail below

One last bit of information that an axis provides makes the most sense after seeing how data points are queried. In the most basic situation, you would only collect one axis of a statistic, such as the latency axis from the NFS operations statistic. When you ask for data, you would get back a data point for every collection interval in the time range you requested. These data points would be grouped into a single stream.



However, if you had collected the operation type axis as well as the latency axis, you would get two streams of datapoints: one for reading operations, and one for write operations.



Because the operation axis applies to many data points, the data points returned are split into two streams, and the operation axis is stored with the top-level stream instead of with each data point in the streams.

However, the latency axis will be different for each data point in a stream, so it is not an attribute of the stream, but instead an attribute of the datapoint.

6.8.1.1.3 Statistic slices

Statistics describe what data can be collected and are auto-populated by the system, but statistic slices are responsible for actually collecting the data, and you must create them manually when you want to collect some performance data. Each slice is an instantiation of exactly one statistic, and can only gather data that is described by that statistic. "Slices" are so named because each one provides a subset of the information available from the parent statistic it is associated with. A statistic can be thought of as describing the axes of a multidimensional space, whereas you typically will only want to collect a simpler slice of that space due to the large number of axes available.

When you specify a slice, there are several fields that you must supply:

- The statistic type this slice is associated with. This must be the same type as the statistic of which this is an instantiation.
- The collection interval, which must be greater than the minimum collection interval the parent statistic gives
- The axes of the parent statistic this slice will collect

Finally, a slice can place constraints on axes of its parent statistic, allowing you to limit the data you get back. For instance, if you're trying to narrow down the cause of some high NFS latency outliers, it may be useful to filter out any NFS latencies which are shorter than one second. To do this, you would place a constraint on the latency axis of an NFS operation slice that states that the values must be higher than one second. You can constrain any axis in the same fashion, and each axis' description in the parent statistic gives a constraint type that can be applied to it. This allows you to place different types of constraints on the latency axis (which is a number measured in nanoseconds) than the operation type axis (which is an enum that can take the values "read" or "write").

6.8.1.1.4 Persisting analytics data

Data collected by slices is persisted temporarily on the Delphix Engine. Performance data is guaranteed to be available at the finest resolution for six hours, then is compressed to per-minute data and held for seven days, and finally compressed to per-hour data and held for 30 days. If data of a certain resolution will be needed longer than these limits, you should instruct the slice to remember the data permanently until you are done using it. The commands to manage this are listed in the Performance Analytics Tool API Reference (see page 876).

6.8.1.2 Working with performance analytics graphs in the graphical user interface

This topic describes the performance analytics graphs that are available in the Delphix Management application, and the controls for changing the views of those graphs.

6.8.1.2.1 Accessing the performance analytics graphs

1. Log into the Delphix Management application.
2. In the **Resources** menu, select **Performance Analytics**.

3. Use the controls described below to view statistics and their related graphs.

6.8.1.2.2 General graph display and controls

Control Name	Control description	Usage
Chart Tabs	Specifies which graphs/ charts are displayed.	Chart tabs have a multiple select drop-down which has the same three options checked by default (CPU, Network, Disk IO). Users can show/hide graphs by checking/unchecking drop-down options.
Timescale	Controls the time range of data displayed in the graph. Available values are 1 minute, 1 hour, and 1 day. By default, 1 minute is selected.	Timescale options are presented as a drop-down, select from **Minute**, **Hour**, or **Day** to change the Zoom Level.
Pause/Resume	Pause/Resume icons	All charts display live data so users can pause it by clicking the Pause button in the upper right corner of each graph. Once the chart is paused, the Pause button is replaced by the Resume button so a user can click it to see live data again.
Op. Type	Operation Type (used in Disk IO, NFS and iSCSI charts)	Options are presented as a drop-down.
Shown Data Timeline	Displays timestamps of data points in the graph.	
Available Data Timeline	Displays navigable time ranges for historical data.	

Control Name	Control description	Usage
Timeline Selector	Specifies the start and end time for the currently displayed data. The range displayed is controlled by the **Zoom Level**.	Drag the **Timeline Selector** to view statistics for a specific time in the past, or click the scroll bar arrows to view the desired time period. You can also use the slider controls within the Timeline Selector to change the length of time for which data is displayed. When the Timeline Selector is aligned to the right of the timeline, it represents live data that is updated every second. If the Timeline Selector is moved from the right alignment with the timeline, the data displayed is historical and no live updates are displayed. To resume live data updates, move the Timeline Selector back to the right-aligned position representing the current time. The data will be refreshed to the latest data, and live updates will resume every second.
Graph Legend	If more than one set of information is presented on the graph, the Graph Legend displays a description and color for each set and allows a user to toggle that set-off and on. For example, in the network graph there can be multiple network interfaces, and for each network interface the graph displays four statistics (bytes sent, bytes received, packets sent, packets received). When a user toggles off a specific network interface, all four statistics corresponding to that interface are hidden from the screen. The color for lines representing bytes sent and packets sent is the same. Similarly, the color for lines representing bytes received and packets received is the same. This makes it easier to correlate the number of bytes and the number of packets sent/received for a given network interface.	To hide a set of information, click on the set name within the Graph Legend. Data representing that set is removed from the graph, and the set's name is greyed out. To show a set that has been hidden, click on the set name.

6.8.1.2.3 Latency, timeline page, and tooltip graph display and controls

Control Name	Control Description	Usage
Timeline Page Left/Right Button	Scrolls Available Data Timeline by a specified time range depending on the current **Zoom Level**.	When the **Zoom Level** is set to **Minute**, click **Timeline Page Left**. The **Available Data Timeline** is changed to show the time period for the previous hour prior.
Graph Value Tooltip	Shows a value, along with the time stamp, for a specific data point.	Mouse over a data point on the graph to view the tooltip.
Latency Range Selector (shown on latency heatmaps only)	Controls the lower and upper limits for displayed latency buckets.	Drag the lower and upper controls to drill down into a specific range of latency buckets. Latency buckets that fall outside of the selected range are summarized, the lower row representing latency buckets that are below the lower limit, and the upper row representing latency buckets that are below the upper limit of the latency range selector. Use Latency Range Selector to view a more detailed distribution of latencies for a specific range.
Latency Outlier Selector (shown on latency heatmaps only)	Hides infrequent latencies (outliers) based on a percentage threshold. Its range is 0%-10%, with a default of 0%. The percentage establishes a threshold below which buckets are considered "outliers" and are hidden from the graph. Each bucket is assigned a percentage based on the ratio of its count vs the maximum count of any bucket in the graph.	Drag the control to the desired percentage threshold.

6.8.1.3 Performance analytics statistics reference

This topic describes the various performance statistics that are available for the Delphix Engine and how they can be used to analyze and improve performance.

The Delphix Engine is shipped with a default set of statistics that are collected on Delphix Engine virtual appliance, as listed below. The statistics are stored for up to 30 days for historical analysis.

Statistic	Description
CPU Utilization	Total CPU utilization for all CPUs. This statistic includes both kernel and user time.
Network Throughput	Measures throughput in bytes and packets, broken down by sent vs. received data and by the network interface. Each network interface shows four graphed lines: bytes sent, bytes received, packets sent, and packets received. To help easily correlate bytes and packets, the same color is used for both bytes and packet values.
Disk IO	Measures a number of IO operations, and the latencies and throughput of the underlying storage layer. The statistic is represented by the graphs - a column chart for IO operations, a heat map for latency distribution, and a line chart for throughput. IO operations are grouped by reads and write. A shaded rectangle on a latency heat map represents an IO operation (read or write) which falls within a particular time range (bucket). The shading of rectangles depends on the number of IO operations that fall within a particular bucket - the higher the count the darker the shading.
NFS	Measures a number of IO operations and the latencies and throughput of the NFS server layer in the Delphix Engine. Its graphical representation is similar to the Disk IO graph. It is useful to diagnose the performance of dSources and VDBs that use NFS mounts (Oracle, PostgreSQL).
iSCSI	Measures the number of IO operations, and the latencies and throughput, of the iSCSI server layer in the Delphix Engine. Its graphical representation is similar to the Disk IO graph. It is useful to diagnose the performance of Microsoft SQL Server dSources and VDBs.

6.8.1.4 Performance analytics tool API reference

This topic describes basic commands and command syntax for using the Performance Analytics tool.

6.8.1.4.1 Statistic types

More detailed information can be found about each statistic type through the command-line interface (CLI) and webservices API, but the following table provides more information about how similar I/O stack statistic types relate to each other.

Statistic type	Description	Axis name	Axis description	Axis value type
NFS_OPS	Provides information about Network File System operations. This is the entry point to the Delphix Engine for all Oracle database file accesses.	op	I/O operation type	STRING
		path	Path of the affected file	STRING
		size	I/O sizes in bytes	HISTOGRAM
		avgLatency	Average I/O latency in nanoseconds	INTEGER
		cached	Whether reads were cached	BOOLEAN
		latency	I/O latencies in nanoseconds	HISTOGRAM
		count	Number of I/O operations	INTEGER
		client	Address of the client	STRING
		throughput	I/O throughput in bytes	INTEGER
		sync	Whether writes were synchronous	BOOLEAN
iSCSI_OPS	Provides information about iSCSI operations. This is the entry point to the Delphix Engine for all SQL Server file accesses.	Same axes as NFS_OPS, except for path, cached, and sync.		

Statistic type	Description	Axis name	Axis description	Axis value type
VFS_OPS	This layer sits immediately below NFS_OPS and iSCSI_OPS. It should give almost exactly the same latencies, assuming no unexpected behavior is occurring.	Same axes as NFS_OPS, except for client.		
DxFS_OPS	This layer sits immediately below VFS_OPS, and the two of them should give almost exactly the same latencies.	Same axes as VFS_OPS.		
DxFS_IO_QUEUE_OPS	This layer sits below DxFS_OPS, but the latencies will differ from that layer because this layer batches together operations to increase throughput.	op	I/O operation type	STRING
		count	Number of I/O operations	INTEGER
		size	I/O sizes in bytes	HISTOGRAM
		avgLatency	Average I/O latency in nanoseconds	INTEGER
		latency	I/O latencies in nanoseconds	HISTOGRAM
		throughput	I/O throughput in bytes	INTEGER
		priority	Priority of the I/O	STRING

Statistic type	Description	Axis name	Axis description	Axis value type
DISK_OPS	This layer sits below DxFS_IO_QUEUE_OPS at the bottom of the I/O stack and measures interactions between the Delphix Engine an d disks.	op	I/O operation type	STRING
		count	Number of I/O operations	INTEGER
		size	I/O sizes in bytes	HISTOGRAM
		avgLatency	Average I/O latency in nanoseconds	INTEGER
		latency	I/O latencies in nanoseconds	HISTOGRAM
		throughput	I/O throughput in bytes	INTEGER
		error	Whether the I/O resulted in an error	BOOLEAN
		device	Device the I/O was issued to	STRING
CPU_UTIL	This is unrelated to the layers of the I/O stack. It measures CPU utilization on the Delphix Engine.	idle	Idle time in milliseconds (showAxes command may incorrectly state nanoseconds)	INTEGER
		user	User time in milliseconds (showAxes command may incorrectly state nanoseconds)	INTEGER

Statistic type	Description	Axis name	Axis description	Axis value type
		kernel	Kernel time in milliseconds (showAxes command may incorrectly state nanoseconds)	INTEGER
		dtrace	DTrace time in milliseconds (showAxes command may incorrectly state nanoseconds) Subset of time in kernel	INTEGER
		cpu	Which CPU was utilized	INTEGER
NETWORK_INTERFACE_UTIL	Network interface utilization on the Delphix Engine.	inBytes	Number of bytes received	INTEGER
		inPackets	Number of packets received	INTEGER
		outBytes	Number of bytes transmitted	INTEGER
		outPackets	Number of packets transmitted	INTEGER
		networkInterface	Which network interface was utilized	STRING
TCP_STATS	Statistics for all established TCP connections on the Delphix Engine.	localAddress	Local address for the TCP connection	STRING
		localPort	Local port for the TCP connection	INTEGER

Statistic type	Description	Axis name	Axis description	Axis value type
		remoteAddress	Remote address for the TCP connection	STRING
		remotePort	Remote port for the TCP connection	INTEGER
		inBytes	Data bytes received	INTEGER
		outBytes	Data bytes transmitted	INTEGER
		receiveWindowSize	The size of the local receive window	INTEGER
		sendWindowSize	The size of the peer's receive window	INTEGER
		congestionWindowSize	The size of the local congestion window	INTEGER
		retransmittedBytes	Bytes retransmitted	INTEGER
		inUnorderedBytes	Number of bytes received out of order. This is a subset of the "inBytes" value	INTEGER
		unacknowledgedBytes	Number of bytes sent but unacknowledged	INTEGER

Statistic type	Description	Axis name	Axis description	Axis value type
		roundTripTime	Smoothed average round-trip time in microseconds	INTEGER

6.8.1.4.2 Statistic axis value types

Values are returned when a slice's data is queried. Each axis has a value type, which specifies how the data will be returned.

Value type	Description
INTEGER	The value is returned as an integer. For information about what units the integer is measured in, read the documentation for the related datapoint or datapoint stream type.
BOOLEAN	The value is returned as a boolean.
STRING	The value is returned as a string. This is used for enum values as well, although the set of strings that can be returned is limited.

HISTOGRAM	The value is returned as a log-scale histogram. The histogram has size buckets whose minimum and maximum value get doubled. Histograms are returned as JSON maps, where the keys are the minimum value in a bucket and the values are the height of each bucket.
	Here is an example histogram. Notice that buckets with a height of zero are not included in the JSON object and that keys and values are represented as strings.
	`{`
	`"32768": "10",`
	`"65536": "102",`
	`"262144": "15",`
	`"524288": "2"`
	`}`

Axis constraints are used to limit the data which a slice can collect. Each axis specifies a constraint type which can be used to limit that axis' values.

Constraint type	Description
`BooleanConstraint`	A superclass in which constraints on boolean values must extend. Currently, the only subclass is `BooleanEqualConstraint`, which requires that a boolean axis equal either true or false (depending on user input).
`EnumConstraint`	A superclass in which constraints on enum values must extend. Currently, the only subclass is `EnumEqualConstraint`, which requires that an enum axis be equal to a user-specified value.
`IntegerConstraint`	A superclass in which constraints on integer values must extend. Subclasses include `IntegerLessThanConstraint`, `IntegerGreaterThanConstraint`, and `IntegerEqualConstraint`, which map to the obvious comparators for integers.
`NullConstraint`	This class signifies that an axis cannot be constrained. This makes the most sense for axes that provide an average value - placing a constraint on an average doesn't make sense because you are not able to include or discard a particular operation based on what its effects would be on the average of all operations.
`PathConstraint`	A superclass in which constraints on file path values must extend. Currently, the only subclass is `PathDescendantConstraint`, which requires that a path value must be a descendant of the specified path (it must be contained within it). This only applies to paths on the Delphix Engine itself, and all paths used must be canonical Unix paths starting from the root of the filesystem.
`StringConstraint`	A superclass in which constraints on string values must extend. Currently, the only subclass is `StringEqualsConstraint`, which requires that a string value must equal a user-specified string.

6.8.1.4.3 Statistic slice commands

Command	Description and Usage Examples
`getData`	This is used to fetch data from a statistic slice which has been collecting data for a while. It returns a datapoint set, which is composed of datapoint streams, which contain datapoints. For a full description, see the Performance Analytics Tool Overview (see page 870).

Command	Description and Usage Examples
`rememberRange`	This is used to ensure that data collected during an ongoing investigation doesn't get deleted unexpectedly. If this is not used, data is only guaranteed to be persisted for 24 hours. If it is used, data will be remembered until a corresponding call to `stopRememberingRange` is made.
`stopRememberingRange`	This is used to allow previously-remembered data to be forgotten. The data will be forgotten on the same schedule as brand new data, so you will have at least 24 hours before data which you have stopped remembering is deleted. This undoes the `rememberRange` operation.
`pause`	This command pauses the collection of a statistic slice, causing no data to be collected until `resume` is called.
`resume`	This command resumes the collection of a statistic slice, undoing a `pause` operation.

6.8.1.5 Performance analytics case study: using a single statistic

6.8.1.5.1 Overview

The Delphix Engine uses the Network File System (NFS) as the transport for Oracle installations. An increase in the NFS latency could be causing sluggishness in your applications running on top of Virtual Databases. This case study illustrates how this pathology can be root caused using the analytics infrastructure. This performance investigation uses one statistic to debug the issue, and utilizes the many axes of that statistic to filter down the probable cause of the issue. This technique uses an approach of iteratively drilling down by inspecting new axes of a single statistic and filtering the data to only include information about the operations that appear slow. This technique is valuable for determining which use patterns of a resource might be causing the system to be sluggish. If you isolate a performance issue using this approach but aren't sure what is causing it or how to fix it, Delphix Support can provide assistance for your investigation.

The following example inspects the statistic which provides information about NFS I/O operations on the Delphix Engine. This statistic can be collected a maximum of once every second, and the axes it can collect, among others, are:

- **latency**, a histogram of wait times between NFS requests and NFS responses
- **size**, a histogram of the NFS I/O sizes requested

- **op**, whether the NFS requests were reads or writes
- **client**, the network address of the NFS client which was making requests

> Roughly the same performance information can be obtained from the iSCSI interface as well.

6.8.1.5.2 Investigation

Because the NFS layer sits above the disk layer, all NFS operations that use the disk synchronously (synchronous writes and uncached reads) will have latencies that are slightly higher than those of their corresponding disk operations. Usually, because disks have very high seek times compared to the time the NFS server spends on CPU, disk operations are responsible for almost all of the latency of these NFS operations. In the graphical representation, you can see this by looking at how the slower cluster of NFS latencies (around 2ms-8ms) have similar latencies to the median of the disk I/O (around 2ms-4ms). Another discrepancy between the two plots is that the number of disk operations is much lower than the corresponding number of NFS operations. This is because the Delphix filesystem batches together write operations to improve performance.

If database performance is not satisfactory and almost all of the NFS operation time is spent waiting for the disks, it suggests that the disk is the slowest piece of the I/O stack. In this case, disk resources (the number of IOPS to the disks, the free space on the disks, and the disk throughput) should be investigated more thoroughly to determine if adding more capacity or a faster disk would improve performance. However, care must be taken when arriving at these conclusions, as a shortage of memory or a recently-rebooted machine can also cause the disk to be used more heavily due to fewer cache hits.

Sometimes, disk operations will not make up all of the latency, which suggests that something between the NFS server and the disk (namely, something in the Delphix Engine) is taking a long time to complete its work. If this is the case, it is valuable to check whether the Delphix Engine is resource-constrained, and the most common areas of constraint internal to the Delphix Engine are CPU and memory. If either of those is too limited, you should investigate whether expanding the resource would improve performance. If no resources appear to be constrained or more investigation is necessary to convince you that adding resources would help the issue, Delphix Support is available to help debug these issues.

While using this technique, you should take care to recognize the limitations that caching places on how performance data can be interpreted. In this example, the Delphix Engine uses a caching layer for the data it stores, so asynchronous NFS writes will not go to disk quickly because they are being queued into larger batches, and cached NFS reads won't use the disk at all. This causes these types of NFS operations to return much more quickly than any disk operations are able to, resulting in a very large number of low-latency NFS operations in the graph above. For this reason, caching typically creates a bimodal distribution in the NFS latency histograms, where the first cluster of latencies is associated with operations that only hit the cache, and the second cluster of latencies is associated with fully or partially uncached operations. In this case, cached NFS operations should not be compared to the disk latencies because they are unrelated. It is possible to use techniques described in the first example to filter out some of the unrelated operations to allow a more accurate mapping between disk and NFS latencies.

1. Begin the performance investigation by examining some high-level statistics such as **latency**.
    a. Create a slice with statistic type **NFS_OPS.**
    b. Set the slice to collect the **latency** axis.
    c. Do not add any constraints.

d. Set the collection interval. Anything over one second will work, but ten seconds gives good data resolution and will not use a lot of storage to persist the data that is collected. The rest of this example will assume a collection period of ten seconds for all other slices, but any value could be used.

```
/analytics

create
set name=step1
set statisticType=NFS_OPS
set collectionInterval=10
set collectionAxes=latency
commit
```

This will collect a time-series of histograms describing NFS latencies as measured from inside the Delphix Engine, where each histogram shows how many NFS I/O operations fell into each latency bucket during every ten-second interval. After a short period of time, read the data from the statistic slice:

```
select step1
getData
setopt format=json
commit
setopt format=text
```

The `setopt` steps are optional but allow you to see the output better via the CLI. The output looks like this:

```
{
    "type": "DatapointSet",
    "collectionEvents": [],
    "datapointStreams": [{
        "type": "NfsOpsDatapointStream",
        "datapoints": [{
            "type": "IoOpsDatapoint",
            "latency": {
                "32768": "16",
                "65536": "10"
            },
            "timestamp": "2013-05-14T15:51:40.000Z"
        }, ...]
    }],
    "resolution": 10
}
```

The data is returned as a set of datapoint streams. Streams hold the fields which are shared by all the datapoints they contain. Later on, in this example, the **opt** and **client** fields will be added to the streams, and multiple streams will be returned. Streams are described in more

detail in Performance Analytics Tool Overview (see page 870). The `resolution` field indicates the number of seconds that corresponds to each datapoint, which in our case matches the requested `collectionInterval`. The `collectionEvents` field is not used in this example, but lists when the slice was paused and resumed, to distinguish between moments when no data was collected because the slice was paused, and moments when there was no data to collect.

2. If the latency distributions show some slow NFS operations, the next step would be to determine whether the slow operations are reads or writes.
   a. Specify a new **NFS_OPS** slice to collect this by collecting the **op** and **latency** axes.
   b. To limit output to the long-running operations, create a constraint on the **latency**axis that prohibits the collection of data on operations with latency less than 100ms.

```
/analytics

create
set name=step2
set statisticType=NFS_OPS
set collectionInterval=10
set collectionAxes=op,latency

edit axisConstraints.0
set axisName=latency
set type=IntegerGreaterThanConstraint
set greaterThan=100000000
back

commit
```

   The `greaterThan` field is 100ms converted into nanoseconds.
   Reading the data proceeds in the same way as the first step, but there will be two streams of datapoints, one where `op=write,` and one where `op=read`.
   info : Because we constrained output to operations with latencies higher than 100ms, none of the latency histograms will all have any buckets for latencies lower than 100ms.

3. After inspecting the two data streams, you might find that almost all slow operations are writes, so it could be valuable to determine which clients are requesting the slow writes, and how large each of the writes is.
   a. To collect this data, create a new **NFS_OPS** slice which collects the **size** and **client** axes.
   b. Add constraints ensuring that the **op** axis should be constrained to only collect data for **write** operations, and the **latency** axis should be constrained to filter operations taking less than 100ms.
   info : Because the constraint on the op axis dictates that it will always have the value write, it is not necessary to collect the op axis anymore.

```
/analytics
```

```
create
set name=step3
set statisticType=NFS_OPS
set collectionInterval=10
set collectionAxes=size,client

edit axisConstraints.0
set type=IntegerGreaterThanConstraint
set axisName=latency
set greaterThan=100000000
back

edit EnumEqualConstraint
set type=StringEqualConstraint
set axisName=op
set equals=write
back

commit
```

Reading the data proceeds in the same way as the first two steps, but there will be one stream for every NFS client. The dataset collected by this will consist of a set of streams, one corresponding to each NFS client, and each stream will be a time-series of histograms showing write sizes that occurred during each ten-second interval.
Continuing to use this approach will allow you to narrow down the slow writes to a particular NFS client, and you may be able to tune that client in some way to speed it up.

6.8.1.6 Performance analytics case study: using multiple statistics

6.8.1.6.1 Overview

This case study illustrates an investigation involving more than one metric. In typical performance investigations, you will need to peel out multiple layers of the stack in order to observe the component causing the actual performance pathology. This case study specifically examines sluggish application performance caused due to slow IO responses from the disk subsystem. This example will demonstrate a technique of looking at the performance of each layer in the I/O stack to find which layer is responsible for the most latency, then looking for constrained resources that the layer might need to access. This technique is valuable for finding the most-constrained resource in the system, potentially giving actionable information about resources that can be expanded to increase performance.

For the following example, we will inspect latency at two layers: the Network File System (NFS) layer on the Delphix Engine, and the disk layer below it. Both of these layers provide the **latency** axis, which gives a histogram of wait times for the clients of each layer.

6.8.1.6.2 Investigation

The analytics infrastructure enables users to observe the latency of multiple layers of the software stack. This investigation will examine the latency of both layers, and then draw conclusions about the differences between the two.

6.8.1.6.3 Setup

To measure this data, create two slices. When attempting to correlate data between two different statistics, it can be easier to determine causation when collecting data at a relatively high frequency. The fastest that each of these statistics will collect data is once per second, so that is the value used.

1. A slice collecting the **latency** axis for the statistic type **NFS_OPS.**

```
/analytics

create
set name=slice1
set statisticType=NFS_OPS
set collectionInterval=1
set collectionAxes=latency
commit
```

2. A slice collecting the **latency** axis for the statistic type **DISK_OPS.**

```
/analytics

create
set name=slice2
set statisticType=DISK_OPS
set collectionInterval=1
set collectionAxes=latency
commit
```

After a short period of time, read the data from the first statistic slice.

```
select slice2
getData
setopt format=json
commit
setopt format=text
```

The same process works for the second slice. The `setopt` steps allow you to see the output better via the CLI. The output for the first slice might look like this:

```
{
    "type": "DatapointSet",
    "collectionEvents": [],
    "datapointStreams": [{
        "type": "NfsOpsDatapointStream",
        "datapoints": [{
            "type": "IoOpsDatapoint",
            "latency": {
                "512": "100",
                "1024": "308",
                "2048": "901",
                "4096": "10159",
                "8192": "2720",
                "16384": "642",
                "32768": "270",
                "65536": "50",
                "131072": "11",
                "524288": "64",
                "1048576": "102",
                "2097152": "197",
                "4194304": "415",
                "8388608": "320",
                "16777216": "50",
                "33554432": "20",
                "67108864": "9",
                "268435456": "2"
            },
            "timestamp": "2013-05-14T15:51:40.000Z"
        }, {
            "type": "IoOpsDatapoint",
            "latency": {
                "512": "55",
                "1024": "130",
                "2048": "720",
                "4096": "6500",
                "8192": "1598",
                "16384": "331",
                "32768": "327",
                "65536": "40",
                "131072": "14",
                "262144": "87",
                "524288": "42",
                "1048576": "97",
                "2097152": "662",
                "4194304": "345",
                "8388608": "280",
                "16777216": "22",
                "33554432": "15",
                "134217728": "1"
            },
            "timestamp": "2013-05-14T15:51:41.000Z"
```

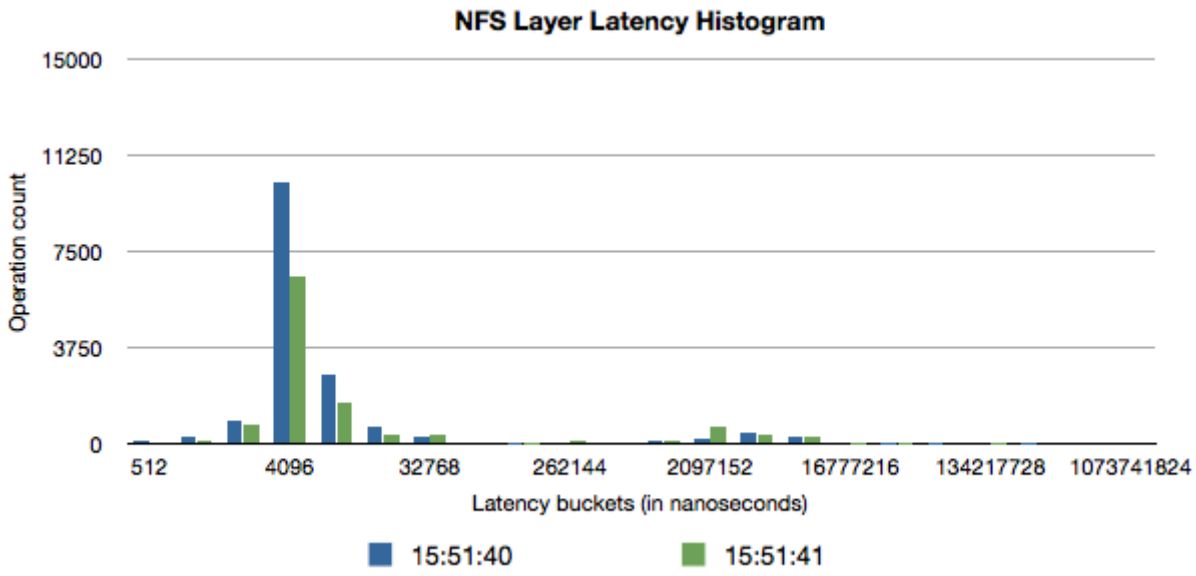
```
        }, ...]
    }],
    "resolution": 1
}
```

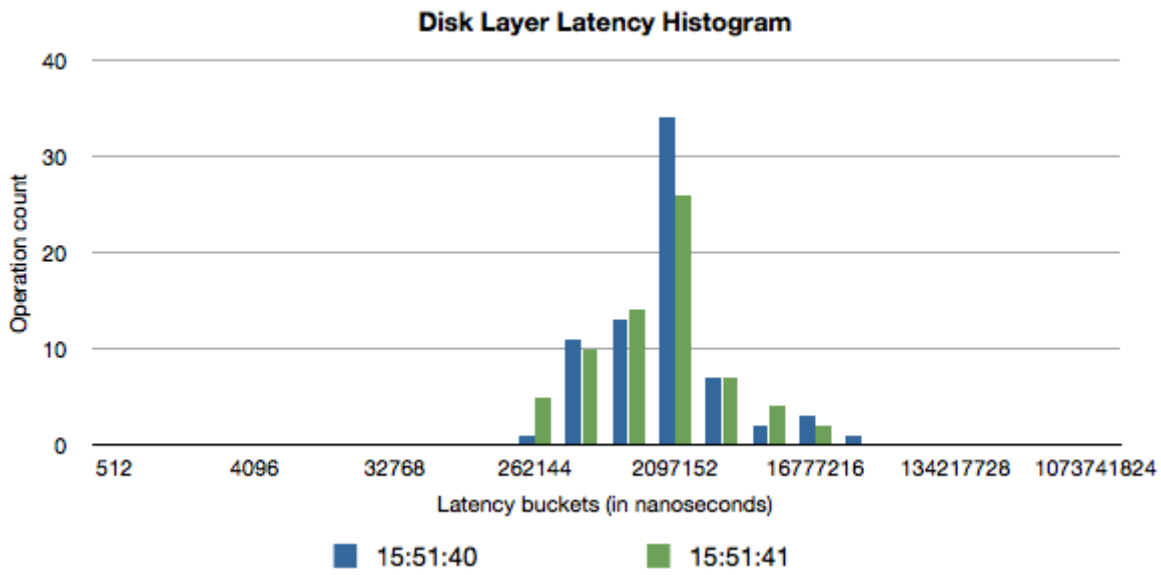
For the second slice, it might look like this:

```
{
    "type": "DatapointSet",
    "collectionEvents": [],
    "datapointStreams": [{
        "type": "DiskOpsDatapointStream",
        "datapoints": [{
            "type": "IoOpsDatapoint",
            "latency": {
                "262144": "1",
                "524288": "11",
                "1048576": "13",
                "2097152": "34",
                "4194304": "7",
                "8388608": "2",
                "16777216": "3",
                "33554432": "1"
            },
            "timestamp": "2013-05-14T15:51:40.000Z"
        }, {
            "type": "IoOpsDatapoint",
            "latency": {
                "262144", "5",
                "524288", "10",
                "1048576", "14",
                "2097152", "26",
                "4194304", "7",
                "8388608", "4",
                "16777216", "2"
            },
            "timestamp": "2013-05-14T15:51:41.000Z"
        }, ...]
    }],
    "resolution": 1
}
```

The data is returned as a set of datapoint streams. Streams hold the fields that would otherwise be shared by all the datapoints they contain, but only one is used in this example because there are no such fields. Streams are discussed in more detail in the Performance Analytics Tool Overview (see page 870). The `resolution` field indicates how many seconds each datapoint corresponds to, which in our case matches the requested `collectionInterval`. The `collectionEvents` field is not used in this example, but lists when the slice was paused and resumed to distinguish between moments when no data was collected because the slice was paused, and moments when there was no data to collect.

Graphically, these four histograms across two seconds look like this:





6.8.1.6.4 Analysis

Because the NFS layer sits above the disk layer, all NFS operations that use the disk synchronously (synchronous writes and uncached reads) will have latencies that are slightly higher than those of their corresponding disk operations. Usually, because disks have very high seek times compared to the time the NFS server spends on CPU, disk operations are responsible for almost all of the latency of these NFS operations. In the graphical representation, you can see this by looking at how the slower cluster of NFS latencies (around 2ms-8ms) have similar latencies to the median of the disk I/O (around 2ms-4ms). Another discrepancy between the two plots is that the number of disk operations is much lower than the corresponding number of NFS operations. This is because the Delphix filesystem batches together write operations to improve performance.

If database performance is not satisfactory and almost all of the NFS operation time is spent waiting for the disks, it suggests that the disk is the slowest piece of the I/O stack. In this case, disk resources (the number of IOPS to the disks, the free space on the disks, and the disk throughput) should be investigated more thoroughly to determine if adding more capacity or a faster disk would improve performance. However, care must be taken when arriving at these conclusions, as a shortage of memory or a recently-rebooted machine can also cause the disk to be used more heavily due to fewer cache hits.

Sometimes, disk operations will not make up all of the latency, which suggests that something between the NFS server and the disk (namely, something in the Delphix Engine) is taking a long time to complete its work. If this is the case, it is valuable to check whether the Delphix Engine is resource-constrained, and the most common areas of constraint internal to the Delphix Engine are CPU and memory. If either of those is too limited, you should investigate whether expanding the resource would improve performance. If no resources appear to be constrained or more investigation is necessary to convince you that adding resources would help the issue, Delphix Support is available to help debug these issues.

While using this technique, you should take care to recognize the limitations that caching places on how performance data can be interpreted. In this example, the Delphix Engine uses a caching layer for the data it stores, so asynchronous NFS writes will not go to disk quickly because they are being queued into larger batches, and cached NFS reads won't use the disk at all. This causes these types of NFS operations to return much more quickly than any disk operations are able to, resulting in a very large number of low-latency NFS operations in the graph above. For this reason, caching typically creates a bimodal distribution in the NFS latency histograms, where the first cluster of latencies is associated with operations that only hit the cache, and the second cluster of latencies is associated with fully or partially uncached operations. In this case, cached NFS operations should not be compared to the disk latencies because they are unrelated. It is possible to use techniques described in the first example to filter out some of the unrelated operations to allow a more accurate mapping between disk and NFS latencies.

1. Begin the performance investigation by examining some high-level statistics such as **latency**.
    a. Create a slice with statistic type **NFS_OPS.**
    b. Set the slice to collect the **latency** axis.
    c. Do not add any constraints.
    d. Set the collection interval. Anything over one second will work, but ten seconds gives good data resolution and will not use a lot of storage to persist the data that is collected. The rest of this example will assume a collection period of ten seconds for all other slices, but any value could be used.

    ```
    /analytics

    create
    set name=step1
    set statisticType=NFS_OPS
    set collectionInterval=10
    set collectionAxes=latency
    commit
    ```

    This will collect a time series of histograms describing NFS latencies as measured from inside the Delphix Engine, where each histogram shows how many NFS I/O operations fell into each latency bucket during every ten-second interval. After a short period of time, read the data from the statistic slice:

```
select step1
getData
setopt format=json
commit
setopt format=text
```

The `setopt` steps are optional but allow you to see the output better via the CLI. The output looks like this:

```
{
    "type": "DatapointSet",
    "collectionEvents": [],
    "datapointStreams": [{
        "type": "NfsOpsDatapointStream",
        "datapoints": [{
            "type": "IoOpsDatapoint",
            "latency": {
                "32768": "16",
                "65536": "10"
            },
            "timestamp": "2013-05-14T15:51:40.000Z"
        }, ...]
    }],
    "resolution": 10
}
```

The data is returned as a set of datapoint streams. Streams hold the fields which are shared by all the data points they contain. Later on, in this example, the **opt** and **client** fields will be added to the streams, and multiple streams will be returned. Streams are described in more detail in Performance Analytics Tool Overview (see page 870). The `resolution` field indicates the number of seconds that corresponds to each datapoint, which in our case matches the requested `collectionInterval`. The `collectionEvents` the field is not used in this example, but lists when the slice was paused and resumed, to distinguish between moments when no data was collected because the slice was paused, and moments when there was no data to collect.

2. If the latency distributions show some slow NFS operations, the next step would be to determine whether the slow operations are reads or write.
   a. Specify a new **NFS_OPS** slice to collect this by collecting the **op** and **latency** axes.
   b. To limit output to the long-running operations, create a constraint on the **latency**axis that prohibits the collection of data on operations with latency less than 100ms.

```
/analytics

create
set name=step2
set statisticType=NFS_OPS
```

```
set collectionInterval=10
set collectionAxes=op,latency

edit axisConstraints.0
set axisName=latency
set type=IntegerGreaterThanConstraint
set greaterThan=100000000
back

commit
```

The `greaterThan` the field is 100ms converted into nanoseconds.

Reading the data proceeds in the same way as the first step, but there will be two streams of data points, one where `op=write,` and one where `op=read`.

Info : Because we constrained output to operations with latencies higher than 100ms, none of the latency histograms will all have any buckets for latencies lower than 100ms.

3. After inspecting the two data streams, you might find that almost all slow operations are writes, so it could be valuable to determine which clients are requesting the slow writes, and how large each of the writes is.
   a. To collect this data, create a new **NFS_OPS** slice which collects the **size** and **client** axes.
   b. Add constraints ensuring that the **op** axis should be constrained to only collect data for **write** operations, and the **latency**axis should be constrained to filter operations taking less than 100ms. **Info**: Because the constraint on the op axis dictates that it will always have the value write, it is not necessary to collect the op axis anymore.

```
/analytics

create
set name=step3
set statisticType=NFS_OPS
set collectionInterval=10
set collectionAxes=size,client

edit axisConstraints.0
set type=IntegerGreaterThanConstraint
set axisName=latency
set greaterThan=100000000
back

edit EnumEqualConstraint
set type=StringEqualConstraint
set axisName=op
set equals=write
back

commit
```

Reading the data proceeds in the same way as the first two steps, but there will be one stream for every NFS client. The dataset collected by this will consist of a set of streams, one

corresponding to each NFS client, and each stream will be a time series of histograms showing write sizes that occurred during each ten-second interval.
Continuing to use this approach will allow you to narrow down the slow writes to a particular NFS client, and you may be able to tune that client in some way to speed it up.

6.8.2 Storage performance configuration options

This section covers the following topics:

- Optimal storage configuration parameters for the Delphix engine (see page 896)
- Storage performance test tool (fio) (see page 897)
- Storage performance expectations and troubleshooting (see page 903)

6.8.2.1 Optimal storage configuration parameters for the Delphix engine

This topic describes minimum capacity and throughput requirements for storage devices used with the Delphix Engine.

Storage for the Delphix Engine must be able to sustain the aggregated Input/Output Operations Per Second (IOPS) and throughput (MBPS) requirements of all its Virtual Databases. Throughput required for data source synchronization (SnapSync and LogSync) must also be supported.

The Delphix Engine requires storage for:

Item	Description
A copy of each Source Database	The copies are compressed.
Unique Block Changes in VDBs	When changes are made to a VDB, the Delphix Engine stores the changes in new blocks associated with only that VDB. The new blocks are compressed.
Timeflow for dSources and VDBs	The TimeFlow kept for each dSource and VDB comprises snapshots of the database (blocks changed since the previous snapshot) and archive logs. The retention period for this history of changes is determined by policies established for each dSource and VDB. The TimeFlow is compressed.

In addition to the storage for these items, the Delphix Engine requires 30% free space in its storage for the best performance. See An Overview of Capacity and Performance Information (see page 748) and related topics for more details on managing capacity for the Delphix Engine.

Best practices for storage performance include:

- Initial storage is equal to the size of the physical source databases. For high redo rates and/or high DB change rates, allocate an additional 10-20% storage.
- Add storage when storage capacity approaches 30% free
- Use physical LUNS allocated from storage pools or RAID groups that are configured for availability

- Never share physical LUNs between the Delphix Engine and other storage clients.
- Keep all physical LUNs the same size.
- Provision storage using VMDKs or RDMs operating in virtual compatibility mode.
- VMDKs should be **Thick Provisioned, Lazy Zeroed.** The underlying physical LUNs can be thin provisioned.
- Physical LUNs used for RDMs should be thick provisioned.
- Measure or estimate the required IOPS and manage the storage disks to provide this capacity. It is common to use larger numbers of spindles to provide the IOPS required.
- Physical LUNs carved from RAID 1+0 groups or pools with dedicated spindles provide higher IOPS performance than other configurations
- Maximize Delphix Engine vRAM for a larger system cache to service reads

Example

There are two production dSources, totaling 5 TB in size. 5 VDBs will be created for each. The Sum of read and write rates on the production source database is moderate (1000 IOPS), the sum of VDB read rate is moderate (950 IOPS), and the VDB update rate is low (50 IOPS).

- Initial storage equal to 5TB, provisioned as 5 x 1 TB physical LUNs, Thin Provisioned. Allow for expansion of the LUNs to 2TB.
- Provision as 5 x 950 GB Virtual Disks. VMDKs must be Thick Provisioned, Lazy Zeroed. Using 1 TB LUNs allows expansion to 2 TB (ESX 5.1 limit).
- The storage provisioned to the Delphix Engine storage must be able to sustain 1000 IOPs (950 + 50). For this reason, each physical LUN provisioned to the Delphix Engine must be capable of sustaining 200 IOPs. IOPs on the source databases are not relevant to the Delphix Engine.
- 64GB Delphix Engine vRAM for a large system cache

6.8.2.2 Storage performance test tool (fio)

6.8.2.2.1 Overview

This fio-based Storage Performance Tool executes a synthetic workload to evaluate the performance characteristics of the storage assigned to the Delphix Engine. The Storage Performance Tool is a feature that is only available from the command-line

6.8.2.2.2 Prerequisites

Prior to setting up the Delphix Engine, the admin can login to the Delphix CLI using a **sysadmin** account to launch the Storage Performance Tool. Because the test is destructive, it will only run against storage which has not been allocated to Delphix for use by the engine.

6.8.2.2.3 Running the storage test via CLI

1. Login as the sysadmin user to the Delphix Engine CLI.
    a. If the Delphix Engine has not been setup yet, the **network setup** prompt appears. Discard the command.

```
delphix network setup update *> discard delphix>
```

2. Create a storage test.

```
delphix> storage test
delphix storage test> create
delphix storage test create *>
```

3. Use 'get' to see other optional arguments. Modify the test parameters as needed and **commit** to start the test.

```
delphix storage test create *> get
   type: StorageTestParameters
   devices: (unset)
   duration: 120
   initializeDevices: true
   initializeEntireDevice: false
   testRegion: 128GB
   tests: ALL
delphix storage test create *> commit
    STORAGE_TEST-1
    Dispatched job JOB-1
    STORAGE_TEST_EXECUTE job started for "SYSTEM".
    Initializing storage test.
    Starting storage device initialization.
    ETA: 1:28:44.
    Storage device initialization complete.
    Starting storage benchmarking.
    Starting random read workload with 4 KB block size and 8 jobs.
    Starting random read workload with 4 KB block size and 16 jobs.
    Starting random read workload with 4 KB block size and 32 jobs.
    Starting random read workload with 4 KB block size and 64 jobs.
    Starting random read workload with 8 KB block size and 8 jobs.
    Starting random read workload with 8 KB block size and 16 jobs.
    Starting random read workload with 8 KB block size and 32 jobs.
    Starting random read workload with 8 KB block size and 64 jobs.
    Starting sequential write workload with 1 KB block size and 4 jobs.
    Starting sequential write workload with 4 KB block size and 4 jobs.
    Starting sequential write workload with 8 KB block size and 4 jobs.
    Starting sequential write workload with 16 KB block size and 4 jobs.
    Starting sequential write workload with 32 KB block size and 4 jobs.
    Starting sequential write workload with 64 KB block size and 4 jobs.
    Starting sequential write workload with 128 KB block size and 4 jobs.
    Starting sequential write workload with 1024 KB block size and 4 jobs.
    Starting sequential write workload with 1 KB block size and 16 jobs.
    Starting sequential write workload with 4 KB block size and 16 jobs.
    Starting sequential write workload with 8 KB block size and 16 jobs.
    Starting sequential write workload with 16 KB block size and 16 jobs.
```

```
    Starting sequential write workload with 32 KB block size and 16 jobs.
    Starting sequential write workload with 64 KB block size and 16 jobs.
    Starting sequential write workload with 128 KB block size and 16 jobs.
    Starting sequential write workload with 1024 KB block size and 16 jobs.
    Starting sequential read workload with 64 KB block size and 4 jobs.
    Starting sequential read workload with 64 KB block size and 8 jobs.
    Starting sequential read workload with 64 KB block size and 16 jobs.
    Starting sequential read workload with 64 KB block size and 32 jobs.
    Starting sequential read workload with 64 KB block size and 64 jobs.
    Starting sequential read workload with 128 KB block size and 4 jobs.
    Starting sequential read workload with 128 KB block size and 8 jobs.
    Starting sequential read workload with 128 KB block size and 16 jobs.
    Starting sequential read workload with 128 KB block size and 32 jobs.
    Starting sequential read workload with 128 KB block size and 64 jobs.
    Starting sequential read workload with 1024 KB block size and 4 jobs.
    Starting sequential read workload with 1024 KB block size and 8 jobs.
    Starting sequential read workload with 1024 KB block size and 16 jobs.
    Starting sequential read workload with 1024 KB block size and 32 jobs.
    Starting sequential read workload with 1024 KB block size and 64 jobs.
    Storage benchmarking complete.
    Generating results.
    Storage test completed successfully.
    STORAGE_TEST_EXECUTE job for "SYSTEM" completed successfully.
delphix storage test>
```

4. The job will be submitted and visible in the Delphix Management application.
5. Retrieve the test results

```
delphix storage test> select STORAGE_TEST-1
delphix storage test 'STORAGE_TEST-1'> result
delphix storage test 'STORAGE_TEST-1' result *> commit
Test Results
------------
Test ID:          1
Test System UUID: 564dc710-7bb1-c064-12c2-2659032acf1b
Start Time:       03-Feb-2015 10:52:31 -0800
End Time:         03-Feb-2015 12:20:25 -0800

Test Grades:

Test Name                                    Latency            Load Scaling
-------------------------------     --------------------------  ---------------
                                    Average  95th %ile   Grade  Scaling   Grade
                                    -------  ---------   -----  -------   -----
Random 8K Reads w/ 16 jobs             2.16       4.77     A-      0.89    poor
Random 4K Reads w/ 16 jobs             1.62       3.73     A       0.54    fair
Sequential 1M Reads w/ 4 jobs         62.60     182.00     D       1.40     bad
Sequential 1K Writes w/ 4 jobs         1.30       2.61     C       0.07    good
Sequential 128K Writes w/ 4 jobs      10.19      26.00     D       1.35     bad
```

```
Grading Key:

Test Name          Grade:  A+    A     A-    B     B-    C     C-     D
-----------------         ----  ----  ----  ----  ----  ----  ----  -----
Small Random Reads         2.0   4.0   6.0   8.0  10.0  12.0  14.0 > 14.0
Large Seq Reads           12.0  14.0  16.0  18.0  20.0  22.0  24.0 > 24.0
Small Seq Writes           0.5   1.0   1.5   2.0   2.5   3.0   3.5 >  3.5
Large Seq Writes           2.0   4.0   6.0   8.0  10.0  12.0  14.0 > 14.0

IO Summary:

Test Name                       IOPS  Throughput (MBps)          Latency (msec)
---------------------------  -------  -----------------
----------------------------------
                                                            Average      Min
Max    StdDev
                                                            -------  -------
-------  -------
Rand 4K Reads w/ 8 Jobs        15703              61.34     0.50     0.05
754.74     1.72
Rand 4K Reads w/ 16 Jobs       15631              61.06     1.00     0.05
1347.10     5.12
Rand 4K Reads w/ 32 Jobs       15972              62.39     1.95     0.05
1231.40    17.56
Rand 4K Reads w/ 64 Jobs       17341              67.74     3.62     0.05
1750.10    30.09
Rand 8K Reads w/ 8 Jobs        15151             118.37     0.52     0.05
45.18     0.27
Rand 8K Reads w/ 16 Jobs       16457             128.58     0.95     0.05
501.90     3.57
Rand 8K Reads w/ 32 Jobs       16908             132.10     1.84     0.05
1336.10    16.93
Rand 8K Reads w/ 64 Jobs       16865             131.76     3.71     0.05
1505.50    30.03
Seq 1K Writes w/ 4 Jobs        22053              21.54     0.18     0.04
168.14     0.26
Seq 4K Writes w/ 4 Jobs        24937              97.41     0.16     0.04
152.17     0.27
Seq 8K Writes w/ 4 Jobs        22946             179.27     0.17     0.04
120.19     0.28
Seq 16K Writes w/ 4 Jobs       18003             281.31     0.22     0.05
81.24     0.26
Seq 32K Writes w/ 4 Jobs       12993             406.05     0.30     0.05
40.33     0.38
Seq 64K Writes w/ 4 Jobs        6429             401.83     0.62     0.06
116.19     2.26
Seq 128K Writes w/ 4 Jobs       3614             451.86     1.10     0.08
200.12     4.75
Seq 1M Writes w/ 4 Jobs          388             388.83    10.28     0.27
832.57    42.24
```

```
Seq 1K Writes w/ 16 Jobs       25965              25.36     0.60     0.04
814.84     6.86
Seq 4K Writes w/ 16 Jobs       25610             100.04     0.61     0.04
1022.50     7.76
Seq 8K Writes w/ 16 Jobs       25183             196.75     0.62     0.04
910.55     7.91
Seq 16K Writes w/ 16 Jobs      23433             366.14     0.66     0.04
948.57     8.05
Seq 32K Writes w/ 16 Jobs      19327             604.00     0.81     0.05
1180.50     8.10
Seq 64K Writes w/ 16 Jobs       9313             582.08     1.71     0.06
711.96     4.40
Seq 128K Writes w/ 16 Jobs      3369             421.14     4.75     0.08
69.12     4.32
Seq 1M Writes w/ 16 Jobs         481             481.06    33.22     0.27
269.88    32.05
Seq 64K Reads w/ 4 Jobs        16912            1057.20     0.23     0.05
40.36     0.15
Seq 64K Reads w/ 8 Jobs        18862            1178.10     0.42     0.05
78.57     0.41
Seq 64K Reads w/ 16 Jobs       20352            1272.50     0.77     0.06
900.81     7.41
Seq 64K Reads w/ 32 Jobs       20750            1296.10     1.50     0.06
1231.60    20.02
Seq 64K Reads w/ 64 Jobs       21146            1321.70     2.95     0.06
2440.30    34.37
Seq 128K Reads w/ 4 Jobs       11649            1456.30     0.34     0.06
53.66     0.25
Seq 128K Reads w/ 8 Jobs       15995            1999.50     0.49     0.06
32.21     0.42
Seq 128K Reads w/ 16 Jobs      17413            2176.80     0.90     0.07
1057.60     6.46
Seq 128K Reads w/ 32 Jobs      17874            2234.30     1.76     0.07
1355.40    19.87
Seq 128K Reads w/ 64 Jobs      17523            2190.50     3.58     0.07
1926.20    36.79
Seq 1M Reads w/ 4 Jobs          1404            1404.20     2.84     0.31
64.38     0.75
Seq 1M Reads w/ 8 Jobs          2360            2360.70     3.38     0.32
17.60     0.46
Seq 1M Reads w/ 16 Jobs         3876            3876.50     4.10     0.33
429.44     3.20
Seq 1M Reads w/ 32 Jobs         4732            4732.60     6.69     0.29
1305.70    34.64
Seq 1M Reads w/ 64 Jobs         4730            4730.10    13.33     0.32
1847.90    54.39



IO Histogram:
```

```
Test Name                      us50  us100  us250  us500    ms1    ms2    ms4
ms10   ms20   ms50  ms100  ms250  ms500     s1     s2     s5
---------------------------  -----  -----  -----  -----  -----  -----  -----
-----  -----  -----  -----  -----  -----  -----  -----  -----
Rand 4K Reads w/ 8 Jobs          0      0      0      2     46     41      7
3      1      0      0      0      0      0      0      0
Rand 4K Reads w/ 16 Jobs         0      0      0      0     39     47     10
3      1      0      0      0      0      0      0      0
Rand 4K Reads w/ 32 Jobs         0      0      0      0      6     64     22
6      2      0      0      0      0      0      0      0
Rand 4K Reads w/ 64 Jobs         0      0      0      0      0      4     75
16      3      1      0      0      0      0      0      0
Rand 8K Reads w/ 8 Jobs          0      0      0      0     41     49      7
2      1      0      0      0      0      0      0      0
Rand 8K Reads w/ 16 Jobs         0      0      0      0      8     66     20
4      2      1      0      0      0      0      0      0
Rand 8K Reads w/ 32 Jobs         0      0      0      0      0      5     72
18      3      2      0      0      0      0      0      0
Rand 8K Reads w/ 64 Jobs         0      0      0      0      0      0      3
85      8      4      1      0      0      0      0      0
Seq 1K Writes w/ 4 Jobs          0      0      0      4     53     36      4
2      1      0      0      0      0      0      0      0
Seq 4K Writes w/ 4 Jobs          0      0      0      2     44     44      6
3      1      0      0      0      0      0      0      0
Seq 8K Writes w/ 4 Jobs          0      0      0      1     41     47      7
3      1      0      0      0      0      0      0      0
Seq 16K Writes w/ 4 Jobs         0      0      0      0     27     57     10
3      1      0      0      0      0      0      0      0
Seq 32K Writes w/ 4 Jobs         0      0      0      0      4     55     30
8      2      1      0      0      0      0      0      0
Seq 64K Writes w/ 4 Jobs         0      0      0      0      0      1     56
33      7      3      0      0      0      0      0      0
Seq 128K Writes w/ 4 Jobs        0      0      0      0      0      0      0
76     16      5      2      0      0      0      0      0
Seq 1M Writes w/ 4 Jobs          0      0      0      0      0      0      0
0      0     24     57     14      4      0      0      0
Seq 1K Writes w/ 16 Jobs         0      0      0      1     55     34      6
2      1      0      0      0      0      0      0      0
Seq 4K Writes w/ 16 Jobs         0      0      0      1     46     42      7
3      1      0      0      0      0      0      0      0
Seq 8K Writes w/ 16 Jobs         0      0      0      0      3     43     38
12      2      1      0      0      0      0      0      0
Seq 16K Writes w/ 16 Jobs        0      0      0      0      0      4     67
23      4      2      0      0      0      0      0      0
Seq 32K Writes w/ 16 Jobs        0      0      0      0      0      0      1
74     16      8      2      0      0      0      0      0
Seq 64K Writes w/ 16 Jobs        0      0      0      0      0      0      0
2     81     12      3      1      0      0      0      0
Seq 128K Writes w/ 16 Jobs       0      0      0      0      0      0      0
1      2     85      7      4      0      0      0      0
Seq 1M Writes w/ 16 Jobs         0      0      0      0      0      0      0
0      0      1      4     45     38      9      3      0
```

```
Seq 64K Reads w/ 4 Jobs          0      0      0      0      0     29     59
9      2      1      0      0      0      0      0      0
Seq 64K Reads w/ 8 Jobs          0      0      0      0      0      0     15
74      8      3      1      0      0      0      0      0
Seq 64K Reads w/ 16 Jobs         0      0      0      0      0      0      0
14     53     27      5      1      0      0      0      0
Seq 64K Reads w/ 32 Jobs         0      0      0      0      0      0      0
1     27     59      8      4      0      0      0      0
Seq 64K Reads w/ 64 Jobs         0      0      0      0      0      0      0
0      1     29     42     25      2      0      0      0
Seq 128K Reads w/ 4 Jobs         0      0      0      0      0      0     10
75      9      5      1      0      0      0      0      0
Seq 128K Reads w/ 8 Jobs         0      0      0      0      0      0      0
64     29      5      2      0      0      0      0      0
Seq 128K Reads w/ 16 Jobs        0      0      0      0      0      0      0
1     45     45      6      2      0      0      0      0
Seq 128K Reads w/ 32 Jobs        0      0      0      0      0      0      0
0      1     65     24      8      1      0      0      0
Seq 128K Reads w/ 64 Jobs        0      0      0      0      0      0      0
0      1      8     54     29      5      1      0      0
Seq 1M Reads w/ 4 Jobs           0      0      0      0      0      0      0
0      0     66     23      8      2      0      0      0
Seq 1M Reads w/ 8 Jobs           0      0      0      0      0      0      0
0      0      1     33     52     11      3      0      0
Seq 1M Reads w/ 16 Jobs          0      0      0      0      0      0      0
0      0      1      5     70     15      8      1      0
Seq 1M Reads w/ 32 Jobs          0      0      0      0      0      0      0
0      0      1      4     19     58     11      6      1
Seq 1M Reads w/ 64 Jobs          0      0      0      0      0      0      0
0      0      1      2     10     40     32     11      2

delphix storage test 'STORAGE_TEST-1'>
```

6.8.2.3 Storage performance expectations and troubleshooting

6.8.2.3.1 Overview

When you initially set up a Delphix Engine, there is a one-time opportunity (and requirement) to run your storage performance tests on unconfigured storage. Because of the test method we use, we cannot run it again in the same way after the engine is set up. When run from a Windows target host, DiskSpd[258] can help establish a performance baseline that combines the network and storage performance over **iSCSI**. This can be a significant help in determining whether your current or future performance is within normal expectations. Similarly, you can use fio[259] on the *nix side from a target system to establish a performance baseline that combines the network and storage performance over **NFS**.

258 https://gallery.technet.microsoft.com/DiskSpd-a-robust-storage-6cd2f223
259 http://freecode.com/projects/fio

6.8.2.3.2 Troubleshooting and information gathering questions

These questions are presented roughly in order of priority. The answers to these questions should help narrow down possible causes of poor performance.

- All storage traffic for virtualization Target hosts goes over the network. Have you reviewed Network Performance Expectations and Troubleshooting (see page 731)?[260]
- Were all of the storage best practices (see page 2348) applied?
    - In particular, check the IO Operation limit and round-robin settings.
- Was a baseline ever created for this host using DiskSpd or fio? (How does our current performance compare?)
- Gather storage specs from VM/storage administrator:
    - Vendor, Model (EMC VMAX 40k, HP 3PAR StoreServ 7000)
    - IO latency SLO (e.g. 5ms 99.999%)
    - IOPS/GB SLO (e.g. 0.68 IOPS/GB for EMC Gold-1)
    - Cache type and size (e.g. FAST cache 768GB)
    - Tier, #Pools; if auto – tiering; relocation schedule (e.g. Gold/Silver/Auto/3 pools/etc)
    - Pool detail: (#) drives, RPM, Type (e.g. Pool1: (20) EFD, (30) 15k SAS, Pool 2: (40) 10k SATA)
    - Connection (XXGb Fibre Channel)
    - Dedicated or Shared pool (how many applications/servers)

6.8.2.3.3 Conclusion

If you need further help, please contact Delphix Support or Professional Services to assist in getting the best performance possible from your environment.

6.8.3 Architecture for performance - hypervisors and host

The Delphix Engine is a "virtual appliance": a virtual machine guest running within a hypervisor on a physical host.
There are a few key best practices we need to keep in mind as we consider this architecture.

6.8.3.1 Architecture best practices hypervisor host ESX[261]

6.8.3.1.1 Hypervisor

1. ESXi 6.x[262] or 7.0 is recommended. ESXi 5.5 or earlier is no longer supported.

---

260 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/network-performance-expectations-and-troubleshooting
261 https://delphixdocs.atlassian.net/wiki/pages/resumedraft.action?draftId=5800521
262 https://www.vmware.com/files/pdf/techpaper/VMware-PerfBest-Practices-vSphere6-0.pdf

2. HyperThreading (HT) for Intel®-based servers (no HT on AMD CPUs).
    a. Disable HT in BIOS, on the ESXi Host, **and** disable HT Sharing[263] on the Delphix VM for consistency. This is our best practice, disable at all levels.
    Any other combination may result in a non-deterministic performance.
    b. When HT cannot be turned off for both the Host and Delphix VM, it should be turned on at all levels, not run in a "mixed-mode".
        i. HT disablement at a guest level only can result in non-deterministic performance.
        ii. A dedicated ESXi host, cluster, or DRA is recommended where consistent VDB performance is paramount.
    c. VM migration to a new host (e.g. VMware HA or vMotion) can create mismatched HT settings.
3. ESXi overhead (resources required for hypervisor cannot be reserved, they must be left unallocated).
    a. Memory Overhead - 10% of available RAM must not be allocated to guest VMs.
        i. *Example: 256GB RAM, allocate 230GB to Delphix VM, leave 26GB for ESX*
    b. CPU Overhead - At least 2 cores (ideally 4) must not be allocated to guest VMs.
        i. *Example: If 16 physical cores are available, allocate 12 to the virtual machines, leaving 4 for the hypervisor*
        ii. Why 4? Certain hypervisor functions require precedence over any virtualized system. If a hypervisor needs more CPU than the amount currently available, it can de-schedule all other virtual processes to ensure adequate CPU resources for the hypervisor. Ensuring the hypervisor will not have to de-schedule any running virtual processes (worlds) by setting aside and not over-subscribing CPUs for virtual functions will leave them available for hypervisor use.
    c. Even if the Delphix VM is the only VM on a host, the hypervisor is still active and essential; and still needs resources.
4. BIOS Power Management should be set to High Performance where ESXi controls power management.
    a. It can be impacted by VMware KB 1018206[264] - poor VM application performance caused by power management settings.
    b. Ensure that all BIOS-managed C-States other than C0 are disabled if power management is hardware controlled.
    c. Ensure that all ACPI sleep states above S0 are disabled in the BIOS.
    d. Examples of popular server lines from Cisco, HP, Dell are below. *Specific models will vary, use the appropriate spec sheet.*
        i. UCS[265]: disable the Processor Power States, disable Power Technology, set Energy Performance to "Performance"
        ii. HP Proliant[266]: set HP Power Regulator to HP "Static High Performance" mode

263 https://stromasys.atlassian.net/wiki/display/KBP/VMware+ESX+-+Disabling+hyper+threading+for+a+specific+virtual+machine
264 https://kb.vmware.com/kb/1018206
265 https://www.cisco.com/c/en/us/products/collateral/servers-unified-computing/ucs-b-series-blade-servers/whitepaper_c11-727827.html#HPC
266 https://h20566.www2.hpe.com/hpsc/doc/public/display?calledBy=Search_Result&docId=emr_na-c03031625&docLocale=en_US

iii. Dell[267]: set BIOS System Profile to "Performance Optimized" mode

5. VMware HA can be enabled; VMware DRS is generally disabled.
6. Blade/Rack Server Firmware and ESXi Drivers should be updated to the latest versions.
7. For Intel®-based servers with E5-2600 v2 processors.
8. Two typical server configurations:
   a. Blade Farm
   b. Rack Server
   Hyper-Converged configurations are possible for high performance.

6.8.3.1.2 Virtual machine guest

1. For VM machine settings, see Virtual Machine Requirements for VMware Platform (see page 604).
2. VMWare Guest Specifications:
   a. Minimum: 8 vCPU x 64 GB
   Small: 8 vCPU x 128GB
   Medium: 16 vCPU x 256 GB
   Large: 24 vCPU x 512 GB
   b. Reserve 100% of RAM and CPU:
      i. If the ESX host is dedicated to Delphix, CPU and RAM reservations are advised but not necessary, however, swap space will be required on the hypervisor to compensate for the lack of reserved RAM.
   c. Hyperthreading[268] - See the ESX host section at the top. Disable HT Sharing on VM, disable HT on ESX Host.
3. Assign single-core sockets for vCPUs in all cases. If there is a compelling reason to use multi-core vCPUs, reference the following article from VMware which describes matching virtual multi-core sockets to the hardware ESX is running on.
   VMware Article on CoresPerSocket[269]
   Example: *ESXi Host has 2 socket x 18 core Intel Xeon, Delphix Engine wants 16 vCPU.*
   Configure Delphix VM with 2 Virtual Sockets, 8 Cores Per Socket to utilize hardware architecture.
4. Avoid placing other extremely active VMs on the same ESX host.
5. Monitoring - vSphere Threshold Alerts for CPU, Network, Memory, Capacity.
6. To set the number of vCPUs per virtual machine via the vSphere client, please see "*Virtual CPU Configuration*" in the Administration guide:
   ESXi 6.0[270]

---

267 https://www.dellhpcsolutions.com/dellhpcsolutions/assets/Optimal_BIOS_HPC_Dell_12G.v1.0.pdf
268 https://stromasys.atlassian.net/wiki/display/KBP/VMware+ESX+-+Disabling+hyper+threading+for+a+specific+virtual+machine
269 https://blogs.vmware.com/vsphere/2013/10/does-corespersocket-affect-performance.html
270 https://docs.vmware.com/en/VMware-vSphere/index.html

   a. Delphix VM CPU Utilization[271] - Delphix KB article on what makes Delphix VMs similar to other resource-intensive applications
   b. Exchange on VMware Best Practices[272] - VMworld 2013 session
   c. ESXTOP Reference[273], Blog[274]

7. Ensure that the latest available VMware drivers and firmware versions are installed for HBAs, NICs, and any other hardware components configured on the Delphix virtual machine. This is a critical step that can have a massive impact on the performance and robustness of our solution.

6.8.4 Target host OS and database configuration options

This topic describes configuration options to maximize the performance of a target host in a Delphix Engine deployment. These network-tuning changes should improve performance for any data source

6.8.4.1 OS-specific tuning recommendations

6.8.4.1.1 Solaris

When exclusively using Oracle's Direct NFS Feature (dNFS), it is unnecessary to tune the native NFS client. However, tuning network parameters is still relevant and may improve performance.

6.8.4.1.1.1 Tuning the Kernel NFS client

On systems using Oracle Solaris Zones, the kernel NFS client can only be tuned from the global zone.

On Solaris, by default the maximum I/O size used for NFS read or write requests is 32K. For I/O requests larger than 32K, the I/O is broken down into smaller requests that are serialized. This may result in poor I/O performance. To increase the maximum I/O size:

1. As superuser, add to the `/etc/system` file:

```
* For Delphix: change the maximum NFS block size to 1M
set nfs:nfs3_bsize=0x100000
```

2. Run this command:

271 https://support.delphix.com/Delphix_Virtualization_Engine/Platforms/KBA1019_VMWare_and_Delphix_CPU_Utilization_Discrepancy_Explained
272 https://www.derekseaman.com/2013/08/vmword-2013-virtualize-microsoft-exchange-server.html
273 https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.monitoring.doc/GUID-D89E8267-C74A-496F-B58E-19672CAB5A53.html
274 https://www.yellow-bricks.com/esxtop/

```
# echo "nfs3_bsize/W 100000" | mdb -kw
```

It is critical that the above command be executed exactly as shown, with quotations and space. Errors in the command may cause a system panic and reboot.

6.8.4.1.1.2 Tuning TCP buffer sizes

On systems using Oracle Solaris Zones, TCP parameters, including buffer sizes, can only be tuned from the global zone or in exclusive-IP non-global zones. Shared-IP non-global zones always inherit TCP parameters from the global zone.

Solaris 10

It is necessary to install a new Service Management Facility (SMF) service that will tune TCP parameters after every boot. These are samples of the files needed to create the service:

File	Installation location
dlpx-tcptune[275]	`/lib/svc/method/dlpx-tcptune`
dlpx-tune.xml[276]	`/var/svc/manifest/site/dlpx-tune.xml`

1. As superuser, download the files and install them in the path listed in the **Installation location** in the table.
2. Run the commands:

```
# chmod 755 /lib/svc/method/dlpx-tcptune
# /usr/sbin/svccfg validate /var/svc/manifest/site/dlpx-tune.xml
# /usr/sbin/svccfg import /var/svc/manifest/site/dlpx-tune.xml
# /usr/sbin/svcadm enable site/tcptune
```

Verify that the SMF service ran after being enabled by running the command:

```
# cat `svcprop -p restarter/logfile tcptune`
```

275 https://delphix.atlassian.net/wiki/download/attachments/9039020081/dlpx-tcptune?version=1&modificationDate=1368535883352&cacheVersion=1&api=v2
276 https://delphix.atlassian.net/wiki/download/attachments/9039020081/dlpx-tune.xml?version=1&modificationDate=1368535896429&cacheVersion=1&api=v2

You should see output similar to this:

```
[ May 14 20:02:02 Executing start method ("/lib/svc/method/dlpx-tcptune
start"). ]
Tuning TCP Network Parameters
tcp_max_buf adjusted from 1048576 to 16777216
tcp_cwnd_max adjusted from 1048576 to 4194304
tcp_xmit_hiwat adjusted from 49152 to 4194304
tcp_recv_hiwat adjusted from 128000 to 16777216
[ May 14 20:02:02 Method "start" exited with status 0. ]
```

Solaris 11

As superuser

Run the following commands:

```
# ipadm set-prop -p max_buf=16777216 tcp
# ipadm set-prop -p _cwnd_max=4194304 tcp
# ipadm set-prop -p send_buf=4194304 tcp
# ipadm set-prop -p recv_buf=16777216 tcp
```

6.8.4.1.2 Linux/Redhat/CentOs

6.8.4.1.2.1 Tuning the Kernel NFS client

In Linux, the number of simultaneous NFS requests is limited by the Remote Procedure Call (RPC) subsystem. The maximum number of simultaneous requests defaults to 16. Maximize the number of simultaneous requests by changing the kernel tunable `sunrpc.tcp_slot_table_entries` value to 128.

To ensure that the interface does not drop packets because the driver is configured with one receive queue, use the following commands to view the adaptor policy/increase Rx queue length.

```
<LinuxHost> $ ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:22:BB:CC:DD:22
          inet addr:www.xxx.yyy.zzz  Bcast:www.xxx.yyy.zzz  Mask:255.255.255.0
          inet6 addr: feee::222:bbff:fffc:ddd/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:9000  Metric:1
          RX packets:760729910 errors:0 dropped:700 overruns:0 frame:0
```

```
          TX packets:309094054 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1023150307866 (952.8 GiB)  TX bytes:190673864056 (177.5 GiB)
```

RHEL4 through RHEL5.6

1. As superuser, run the following command to change the instantaneous value of simultaneous RPC commands:

   ```
   # sysctl -w sunrpc.tcp_slot_table_entries=128
   ```

2. Edit the file `/etc/modprobe.d/modprobe.conf.dist` and change the line:

   ```
   install sunrpc /sbin/modprobe --first-time --ignore-install sunrpc && { /bin/mount -t rpc_pipefs sunrpc / var /lib/nfs/rpc_pipefs > /dev/null 2>&1 || :;
   ```

   to

   ```
   install sunrpc /sbin/modprobe --first-time --ignore-install sunrpc && { /bin/mount -t rpc_pipefs sunrpc / var /lib/nfs/rpc_pipefs > /dev/null 2>&1 ; /sbin/sysctl -w sunrpc.tcp_slot_table_entries=128; }
   ```

   Improper changes to the modprobe.conf.dist file may disrupt the use of NFS on the system. Check with your system administrator or operating system vendor for assistance. Save a copy of the modprobe.conf.dist in a directory other than /etc/modprobe.d before starting.

RHEL 5.7 through RHEL 6.2

1. As superuser, run the following command to change the instantaneous value of simultaneous RPC commands:

   ```
   # sysctl -w sunrpc.tcp_slot_table_entries=128
   ```

2. If it doesn't already exist, create the file `/etc/modprobe.d/rpcinfo` with the following contents:

```
options sunrpc tcp_slot_table_entries=128
```

RHEL 6.3 onwards

Beginning with RHEL 6.3, the number of RPC slots is dynamically managed by the system and does not need to be tuned. Although the `sunrpc.tcp_slot_table_entries` tuneable still exists, it has a default value of 2, instead of 16 as in prior releases. The maximum number of simultaneous requests is determined by the new tuneable, `sunrpc.tcp_max_slot_table_entries` which has a default value of 65535.

6.8.4.1.2.2 Tuning TCP buffer sizes

Packages should install their configuration files in /usr/lib/(distribution packages) or /usr/local/lib/(local installs). Files in /etc/ are reserved for the local administrator, who may use this logic to override the configuration files installed by vendor packages. It is recommended to prefix all filenames with a two-digit number and a dash, to simplify the ordering of the files. The following is an example approach and should be tested beforehand.

```
echo "Target Kernel Parameter Tunings.  This is optional but highly recommended."
echo
echo "Tuning TCP Buffer Sizes - Parameters should be as below"
echo
echo "This script takes the recommended vendor approach of creating a file in /usr/
lib/sysctl.d and running \"sysctl -p\""
echo "This script will comment out identical settings in /etc/sysctl.conf, which
would otherwise override the Delphix settings"
echo "Admins can at their discretion set larger values, either in sysctl.conf or /
usr/lib/sysctl.d/60-sysctl.conf"
echo "----"
echo "net.ipv4.tcp_timestamps = 1"
echo "net.ipv4.tcp_sack = 1"
echo "net.ipv4.tcp_window_scaling = 1"
echo "net.ipv4.tcp_rmem = 4096 16777216 16777216"
echo "net.ipv4.tcp_wmem = 4096 4194304 16777216"
cat /dev/null > /usr/lib/sysctl.d/60-sysctl.conf
#Set NOW
NOW=$(date +"%m%d%Y%H%m%s")
echo "net.ipv4.tcp_timestamps = 1" >> /usr/lib/sysctl.d/60-sysctl.conf
echo "net.ipv4.tcp_sack = 1" >> /usr/lib/sysctl.d/60-sysctl.conf
echo "net.ipv4.tcp_window_scaling = 1" >> /usr/lib/sysctl.d/60-sysctl.conf
echo "net.ipv4.tcp_rmem = 4096 16777216 16777216" >> /usr/lib/sysctl.d/60-sysctl.conf
echo "net.ipv4.tcp_wmem = 4096 4194304 16777216" >> /usr/lib/sysctl.d/60-sysctl.conf
echo
echo "Running /sbin/sysctl -p /usr/lib/sysctl.d/60-sysctl.conf"
/sbin/sysctl -p /usr/lib/sysctl.d/60-sysctl.conf
if [ $? -ne 0 ]; then
    echo "Command \"sysctl -p /usr/lib/sysctl.d/60-sysctl.conf\" failed; aborting..."
```

```
    exit 1
fi
# Make a backup and Comment out similar lines in /etc/sysctl.conf if they exist
# sed will search for lines which are NOT comments ( ^[^#]* means starts with
anything other than a "#" ) but contain the parameters either with "." or "/"
 notation, and add a Comment.
cp /etc/sysctl.conf /tmp/sysctl.conf.$NOW
sed -i "/^[^#]*net[\./]ipv4[\./]tcp_timestamps/s/^/#Commented out by Delphix /" /etc/
sysctl.conf
sed -i "/^[^#]*net[\./]ipv4[\./]tcp_sack/s/^/#Commented out by Delphix /" /etc/
sysctl.conf
sed -i "/^[^#]*net[\./]ipv4[\./]tcp_window_scaling/s/^/#Commented out by Delphix /" /
etc/sysctl.conf
sed -i "/^[^#]*net[\./]ipv4[\./]tcp_rmem/s/^/#Commented out by Delphix /" /etc/
sysctl.conf
sed -i "/^[^#]*net[\./]ipv4[\./]tcp_wmem/s/^/#Commented out by Delphix /" /etc/
sysctl.conf
echo
echo "Running /sbin/sysctl -p /etc/sysctl.conf"
/sbin/sysctl -p /etc/sysctl.conf
if [ $? -ne 0 ]; then
    echo "Command \"sysctl -p /etc/sysctl.conf\" failed; aborting..."
    echo "We put a copy of the original at /tmp/sysctl.conf.$NOW"
    exit 1
fi
```

6.8.4.1.2.3 NFSv4 only - enabling recover lost locks

RHEL 6.6 onwards

By default, the Redhat NFSv4 client does not attempt to reclaim locks that were lost due to a lease expiration event. This can cause an application to encounter unexpected EIO errors on system calls such as write. The Delphix use case requires that the NFSv4 client attempt to reclaim lost locks that were due to lease expiration. An NFS client module parameter, '`recover_lost_locks`', is used to change the default behavior. Use the following command to check if the "recover_lost_locks" option is set to 1:

```
grep recover_lost_locks /etc/modprobe.d/*.conf
```

If the option is currently set to 0, change it to 1. If it is missing, proceed to the next paragraph where instruction is provided on how to add the option.

As superuser, run the following two commands to enable the NFS client to recover the lost locks feature:

```
# cat > /etc/modprobe.d/nfs4-locks.conf <<EOF
options nfs recover_lost_locks=1
EOF

# [ -d "/sys/module/nfs" ] && echo Y > /sys/module/nfs/parameters/recover_lost_locks
```

6.8.4.1.3 IBM AIX®

6.8.4.1.3.1 AIX NFSv4 configuration requirements (7.1 and 7.2)

1. An NFS Domain must be configured.
2. The nfsrgyd service must be running.
3. The NFS server IP address from the Delphix Engine must be mappable to an FQDN.

Configure the nfsv4 domain on the AIX target host

bash-3.2# chnfsdom test.com

Start the nfsrgyd service and confirm it is active

```
bash-3.2# startsrc -s nfsrgyd

bash-3.2# lssrc -s nfsrgyd
Subsystem         Group            PID          Status
 nfsrgyd          nfs              7536760      active
```

Confirm that IP address can be resolved

```
bash-3.2$ host 172.16.105.81
81.105.16.172.in-addr.arpa is  dcol1.delphix.com
```

Reference: IBM AIX: HOW TO SETUP NFSV4 MOUNT IN CLIENT AND SERVER[277]

6.8.4.1.3.2 Tuning the Kernel NFS Client

On AIX, by default the maximum I/O size used for NFS read or write requests is 64K. When Oracle does I/O larger than 64K, the I/O is broken down into smaller requests that are serialized. This may result in poor I/O performance. IBM can provide an Authorized Program Analysis Report (APAR) that allows the I/O size to be configured to a larger value.

1. Determine the appropriate APAR for the version of AIX you are using:

AIX Version	APAR Name
6.1	IV24594[278]

277 https://www.ibm.com/support/pages/ibm-aix-how-setup-nfsv4-mount-clinet-and-server
278 http://www-01.ibm.com/support/docview.wss?uid=isg1IV24594

AIX Version	APAR Name
7.1	IV24688[279]

2. Check if the required APAR is already installed by running this command:If the APAR is installed, you will see a message similar to this:

```
# /usr/sbin/instfix -ik IV24594
```

3. If the APAR is not yet installed, you will see a message similar to this:

```
All filesets for IV24594 were found.
```

```
There was no data for IV24594 in the fix database.
```

4. Download and install the APAR, as necessary. To find the APARs, use the main search function at http://www.ibm.com/us/en/, specifying the name of the APAR you are looking for from step 1. A system reboot is necessary after installing the APAR.
5. Configure the maximum read and write sizes using the commands below:

```
# nfso -p -o nfs_max_read_size=524288
# nfso -p -o nfs_max_write_size=524288
```

6. Confirm the correct settings using the command:
   You should see an output similar to this:

```
# nfso -L nfs_max_read_size -L nfs_max_write_size
NAME CUR DEF BOOT MIN MAX UNIT TYPE
   DEPENDENCIES
------------------------------------------------------------------------
------
nfs_max_read_size 512K 64K 512K 512 512K Bytes D
------------------------------------------------------------------------
------
nfs_max_write_size 512K 64K 512K 512 512K Bytes D
```

279 http://www-01.ibm.com/support/docview.wss?uid=isg1IV24688

```
------------------------------------------------------------------------
------
```

6.8.4.1.3.3 Tuning delayed TCP acknowledgements

By default, AIX implements a 200ms delay for TCP acknowledgments. However, it has been found that disabling this behavior can provide significant performance improvements.

To disable delayed ACKs on AIX the following command can be used:

```
# /usr/sbin/no -o tcp_nodelayack=1
```

To make the change permanent use:

```
# /usr/sbin/no -p -o tcp_nodelayack=1
```

6.8.4.1.4 HP-UX

6.8.4.1.4.1 Tuning the Kernel NFS client

On HP-UX, by default the maximum I/O size used for NFS read or write requests is 32K. For I/O requests larger than 32K, the I/O is broken down into smaller requests that are serialized. This may result in poor I/O performance.

1. As superuser, run the following command:

   ```
   # /usr/sbin/kctune nfs3_bsize=1048576
   ```

2. Confirm the changes have occurred and are persistent by running the following command and checking the output:

   ```
   # grep nfs3 /stand/system
   tunable nfs3_bsize 1048576
   ```

6.8.4.1.4.2 Tuning TCP buffer sizes

1. As superuser, edit the `/etc/rc.config.d/nddconf` file, adding or replacing the following entries:

```
TRANSPORT_NAME[0]=tcp
NDD_NAME[0]=tcp_recv_hiwater_def
NDD_VALUE[0]=16777216
#
TRANSPORT_NAME[1]=tcp
NDD_NAME[1]=tcp_xmit_hiwater_def
NDD_VALUE[1]=4194304
```

In this example, the array indices are shown as `0` and `1.` In the actual configuration file, each index used must be strictly increasing, with no missing entries. See the comments at the beginning of `/etc/rc.config.d/nddconf` for more information.

1. Run the command:

```
/usr/bin/ndd -c
```

2. Confirm the settings:

```
# ndd -get /dev/tcp tcp_recv_hiwater_def
16777216
# ndd -get /dev/tcp tcp_xmit_hiwater_def
4194304
```

6.8.4.2 OS-specific tuning recommendations for windows

6.8.4.2.1 Receive side scaling

Follow the instructions here to enable RSS: Enable Receive Side Scaling (RSS) on Staging/Target Network Interfaces (see page 2338)

hosted on the target, such as Oracle, SQL Server, Sybase, or vFiles. They should be applied to all Delphix targets.

6.9 Usage data management

The Delphix User-click Analytics feature is a lightweight method to capture how users interact with Delphix product user interfaces. The goal of capturing this data is to get a better understanding of product usage, engagement, and user behavior, and to use this data to improve Delphix products and customer experience. This feature is enabled by default for customers deploying on or upgrading to this version. User-click Analytics may also be disabled via the UI.

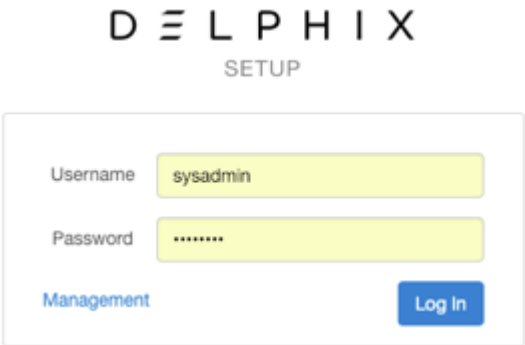
6.9.1 Disabling user-click analytics

> This procedure will disable user-click analytics on both the Delphix Engine and Delphix Self-Service.

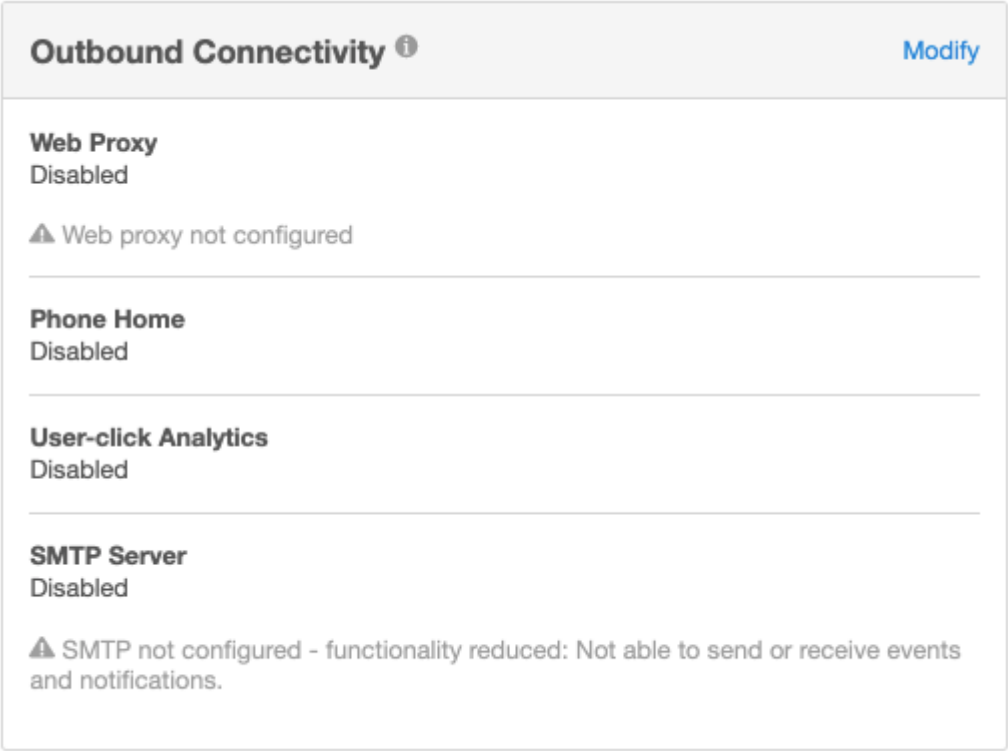
Procedure to disable user-click analytics via the GUI

Complete the following steps to disable user-click analytics via the GUI.

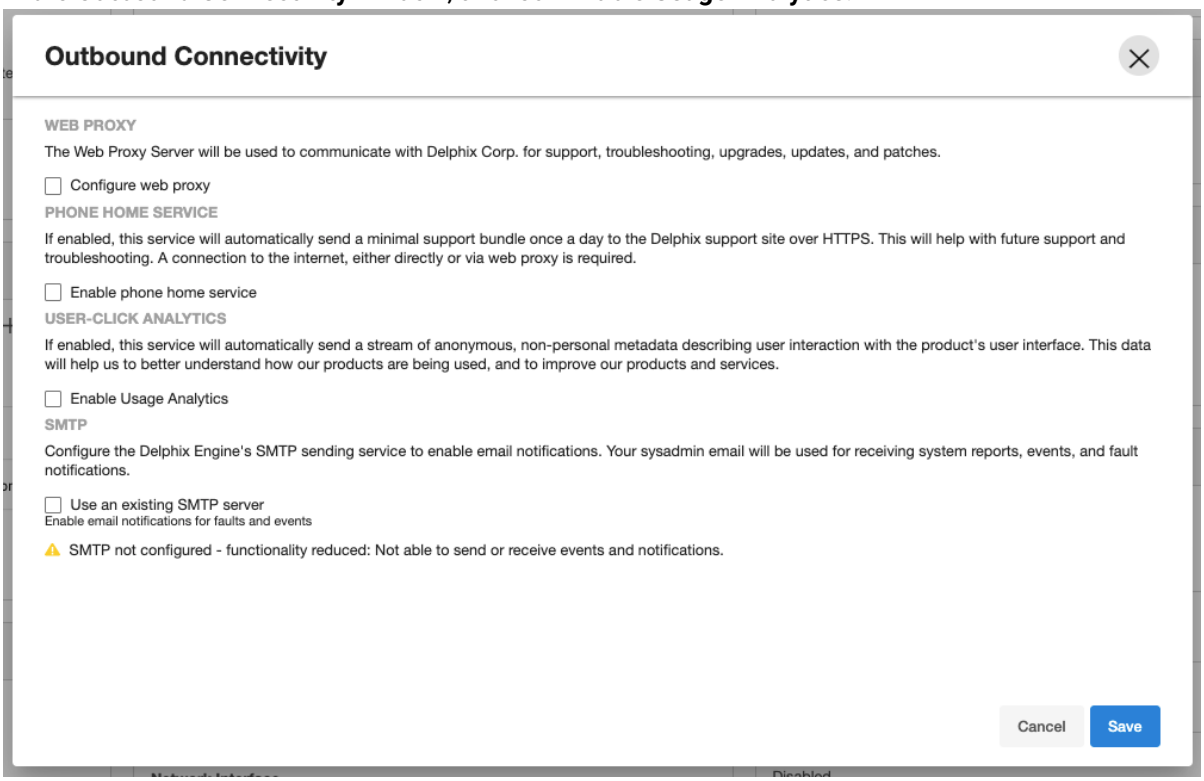
1. Login to the Delphix Setup application using the sysadmin **username** and **password**.



2. From the Outbound Connectivity panel, click **Modify**.



3. In the **Outbound Connectivity** window, uncheck **Enable Usage Analytics**.



4. Click **Save**.

6.10 Starting, stopping, and restarting your engine

Several options are available to start, stop, restart, or reset the Delphix engine for maintenance or debugging purposes.

6.10.1 Factory reset

This option returns the Delphix Engine to the "factory default" state. This removes all the data and configurations except for the network configuration and NTP settings. The network configuration and NTP settings are retained in order to make it is easy for a user to start the Delphix Engine otherwise you will need to perform the initial steps to configure the network via the VMware Hypervisor console.

It is recommended to shut down and remove all VDBs before resetting the Delphix Engine. Failure to do so could lead to stale data mounts in target environments. For the same reason, disable all dSources that use validated sync.

Use Factory Reset only when a complete reset and reconfiguration of the Delphix Engine is necessary, as all Delphix Engine objects will be de-allocated.

Complete the following steps to reset the Delphix Engine via GUI:

1. Connect to the Delphix Setup application and log in as a system administrator.
2. From the **Dashboard** panel, click the Actions menu (**...**) on the top right and select **Factory Reset**.

3. Click **OK**.

6.10.2 Restart

If the Delphix Java process is in a bad state, you may need to restart it to get it operational again.

You can restart the management process safely without shutting down VDBs or disabling dSources because restarting this process has no impact on running VDBs or dSources.

Complete the following steps to restart the Delphix Management Process via GUI:

1. Login to the Delphix Setup application as a system administrator.
2. From the **Dashboard** panel, click the Actions menu (**...**) on the top right and select **Restart**.

3. Click **OK**.

Complete the following steps to restart the Delphix Management Process via CLI:

1. Log in to the CLI as a system administrator.
2. Go to system > restart
3. delphix system restart *> commit

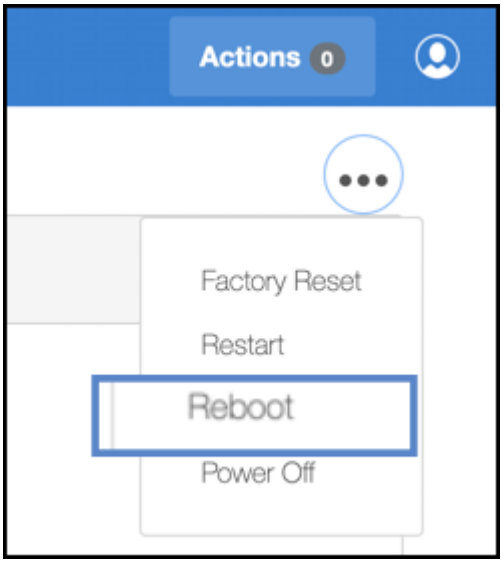
6.10.3 Reboot

If the management stack for your Delphix Engine hangs, you will need to perform a reboot.

Before performing a reboot, all your VDBs must be shut down and dSources disabled to maintain data integrity.

Complete the following steps to reboot the Delphix Engine via GUI:

1. Login to the Delphix Setup application as a system administrator.
2. From the **Dashboard** panel, click the Actions menu (**...**) on the top right and select **Reboot**.



3. Click **OK**.

Complete the following steps to reboot the Delphix Engine via CLI:

1. Log in to the CLI as a system administrator.
2. Go to system > reboot
3. delphix system reboot *> commit

> **Reboot vs restart**
> The only difference between the Restart and Reboot is that the Restart functions without shutting down the VDBs and disabling the dSources.

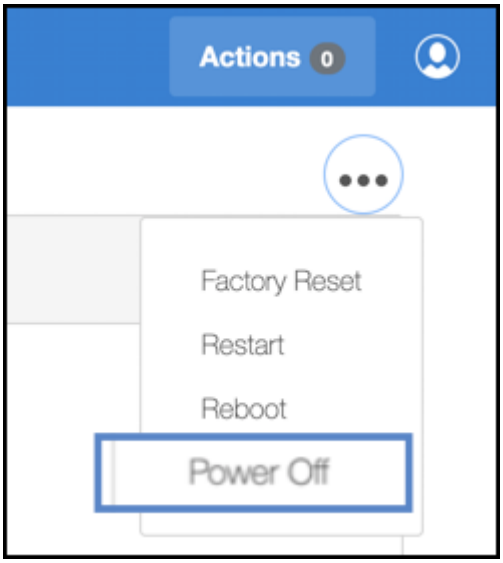
6.10.4 Power off

Occasionally, it is necessary to shut down the Delphix Engine for maintenance purposes. Before performing a shutdown, all your VDBs must be shut down and dSources disabled to maintain data integrity.

Once the Delphix Engine is powered off, you can power it back on through your hypervisor console.

Use the following steps to shut down the Delphix Engine via GUI:

1. Login to the Delphix Setup application as a system administrator.
2. From the **Dashboard** panel, click the Actions menu (**...**) on the top right and select **Power Off**.



3. Click **OK**.

Use the following steps to shut down the Delphix Engine via CLI:

1. ssh sysadmin@yourengine
2. delphix > system
3. delphix system > shutdown
4. delphix system shutdown *> commit

6.11 Introduction to privilege elevation profiles

This topic introduces the concept of Privilege Elevation Profiles, how they are managed, and how they are supported. Privilege Elevation Profiles exist to provide the Delphix Engine with a mechanism for running privileged commands in a secure way to achieve the following:

- Mount and Unmount NFS filesystems
- Create and Remove directories in paths not owned by the Delphix OS user
- Examine the running process list
- Run commands as root

Privilege Elevation Profiles is an advanced CLI topic and are not documented as part of the general Delphix Engine User Guide. Changes to the default sudo-based profile scripts, or the creation of new profiles that do not work as expected, can cause serious problems and render the Delphix Engine unusable. This article is aimed at advanced end-users and Delphix Professional Services consultants.

6.11.1 Support for privilege elevation profiles

> Writing and troubleshooting scripts, such as those required for Privilege Elevation Profiles, is out of scope and not covered by Delphix Support.

Privilege Elevation Profiles need to be tailor-made to work with non-standard environments that may use third-party or proprietary privilege elevation mechanisms other than sudo. Customers are strongly encouraged to work with Delphix Professional Services to formulate reliable profile scripts. There is nothing that prevents customers from creating their own profile scripts. However, customers bear full responsibility for supporting and troubleshooting their own profile scripts. Support for profile scripts created by our Professional Services consultants is still supported by Professional Services.

6.11.2 How do privilege elevation profiles work?

Privilege Elevation Profiles exist within a two-tier cascading hierarchy. This means there is one default profile for the entire Delphix Engine that should contain scripts for all the operations that require privilege elevation. Additional profiles may contain a subset of the scripts. When a non-default profile is used, the Delphix Engine uses that profile's scripts where they exist and reverts to the scripts in the default profile if no script for the operation exists. By default, the Delphix Engine ships with simple scripts that pass commands to the standard UNIX **sudo** command.

All Environments added to the Delphix Engine get added with the default Privilege Elevation Profile. The profile can be assigned on a per-host basis. Below shows how a host using a non-standard profile will use scripts in the cascading model.

default profile (sudo)	custom profile (myProfile)	host profile	script used
dlpx_mount	my_mount	myProfile	my_mount
dlpx_umount	my_umount		my_umount
dlpx_rmdir			dlpx_rmdir
dlpx_mkdir			dlpx_mkdir
dlpx_ps			dlpx_ps
dlpx_pfexec			dlpx_pfexec

default profile (sudo)	custom profile (myProfile)	host profile	script used
dlpx_pfexec_as_user			dlpx_pfexec_as_user

7 Upgrade

Version compatibility and support pre-checks

Please refer to Upgrade matrix (see page 454) before upgrading to a newer version.

Customers running version 5.3.9 and earlier that are requesting an upgrade to 6.0.0.0 and above, please contact Delphix Support to help coordinate this upgrade.

Upgrading from 6.0.x to 6.0.x includes pre-checks packaged in the upgrade image, contacting Delphix Support for this upgrade is **not required** (e.g. 6.0.0.0 → 6.0.9.0).

7.1 Upgrade

These topics describe processes for upgrading the Delphix Engine.

- Upgrading the Delphix Engine: Overview (see page 925)
- Upgrade prerequisites (see page 927)
- Downloading the upgrade image (see page 929)
- Uploading the upgrade image (see page 930)
- Applying the upgrade (see page 940)
- Post upgrade (see page 943)

7.2 Upgrading the Delphix Engine: Overview

Upgrading the Delphix Engine is a multi-step process that requires some preparation. The engine upgrade process will affect the availability of the Delphix Engine administrative interface and virtual datasets during the operation based on the type of upgrade chosen. For large configurations and Full/Apply Now upgrades, it can take one to two hours; Delay the Reboot upgrades are typically complete within 15 to 30 minutes. Please refer to Upgrade Matrix (see page 454) and the **version compatibility and support pre-checks** callout below before proceeding with an upgrade.

The following sections explain the steps involved in the upgrade process with links to detailed instructions for proceeding through each of them.

7.2.1 Types of upgrade

There are two types of upgrades, which are characterized by the impact they have on VDBs during the operation:

There is no need to reboot unless instructed to receive a specific fix from Delphix.

Upgrade type	Description
Delay the reboot	The user interface, API, and CLI (Command Line Interface) will only be available to the user performing the upgrade. dSources will stop refreshing from production. Policies execution will be delayed until after the upgrade has been completed. Jobs will be canceled (and resumed after upgrade if supported). Access to VDB data will not be affected by this upgrade and can be used normally.
Apply now	In addition to performing an application upgrade, DelphixOS, the operating system that runs Delphix, will be upgraded and the machine will reboot to the new OS as part of the upgrade process. The Delphix Engine will automatically disable all VDBs and dSources during the upgrade process in order to safely reboot to the new version, and thus you should schedule downtime for your VDB applications.

7.2.2 Outline of the upgrade process

Version compatibility and support pre-checks

Please refer to Upgrade Matrix (see page 454) before upgrading to a newer version.

Customers running version 5.3.9 and earlier that are requesting an upgrade to 6.0.0.0 and above, please contact Delphix Support to help coordinate this upgrade.

Upgrading from 6.0.x to 6.0.x includes pre-checks packaged in the upgrade image, contacting Delphix Support for this upgrade is **not required** (e.g. 6.0.0.0 → 6.0.9.0).

The following is an outline of the steps for upgrading the Delphix Engine:

1. Upload (see page 930) the upgrade image to the Delphix Engine.
2. Verify (see page 1432) and resolve (see page 935) the system requirements and known defects before starting the upgrade.

3. Schedule the appropriate downtime.
4. Start (see page 940) the upgrade and choose the upgrade type.
5. Address (see page 940) any runtime failures that happen as part of the upgrade.
6. Verify (see page 943) that the upgrade was completed successfully.

7.3 Upgrade prerequisites

Version compatibility and support pre-checks
Please refer to Upgrade matrix (see page 454) before upgrading to a newer version.
Customers running version 5.3.9 and earlier that are requesting an upgrade to 6.0.0.0 and above, please contact Delphix Support to help coordinate this upgrade.
Upgrading from 6.0.x to 6.0.x includes pre-checks packaged in the upgrade image, contacting Delphix Support for this upgrade is **not required** (e.g. 6.0.0.0 → 6.0.9.0).

7.3.1 Scheduling downtime

To determine if an upgrade will require a reboot and VDB downtime refer to Upgrade matrix (see page 454).

If the OS will not be updated as part of the upgrade, then the upgrade process will have no impact on the availability of VDBs, and you do not need to schedule any downtime for your VDB applications.

If the OS will be updated as part of the upgrade, then you should schedule appropriate downtime for your VDB applications. The Delphix Engine will automatically disable VDBs and dSources during the upgrade. The length of the downtime will be proportional to the number of VDBs.

Long-running jobs including replication and SnapSync will fail during any upgrade.

The upgrade file for the version to which you want to upgrade should be downloaded from the Delphix download site[280]. From 6.0.2.0 onwards, the upgrade images are packaged with the latest version of upgrade pre-checks. Please use the latest upgrade image from the download site since the checks will be updated on a need-by basis.

Delphix Upgrade images are approximately 5GB in size; it is recommended to have both a fast internet connection to the Delphix download site as well as to the Delphix Engine.

The upgrade image should be downloaded or moved to a location accessible to the computer used for navigating the Delphix Management application.

Upgrading replication source

280 https://download.delphix.com/

> Delphix Engines can only perform replication to engines on the same or higher version. Upgrading a replication source without upgrading its replication targets could cause replication between those peers to fail.

7.3.2 Verifying the integrity of the downloaded upgrade image

SHA256 checksums can be used to verify the integrity of files downloaded via download.delphix.com[281]. The `sha256sum` checksum listed next to each download will help ensure that a file has not changed as a result of a faulty file transfer, a disk error, or non-malicious meddling.

To calculate the checksum of downloaded files and compare them to the checksum listed on the download portal, the following utilities can be used:

- On Windows, `certutil -HashFile <upgrade.tar.gz> SHA256`
- On macOS or Unix servers with Perl installed, `shasum -a 256 <upgrade.tar.gz>`
- On Unix servers with `coreutils` installed, `sha256sum <upgrade.tar.gz>.`

7.3.3 Verifying connectivity to datasets and environments

For Upgrades from versions > 5.3.6.0 and <6.0.0.0, your Delphix Engine will automatically quiesce all VDBs and dSources during the upgrade process in order to safely reboot to the new version.

For Upgrades from versions >= 6.0.0.0, based on the Upgrade matrix (see page 454) you can choose to postpone the VDB downtime and update just the running stack and related packages by choosing "Delay the Restart" option. You can update the OS at a later date by completing the upgrade.

At the end of the upgrade process, the Delphix Engine will also update the Delphix platform toolkit on each connected environment.

To perform these tasks, the Delphix Engine must be able to connect to the environments in which datasets exist and must have credentials to connect to datasets or applications. Environments involved must be properly configured to enable script execution.

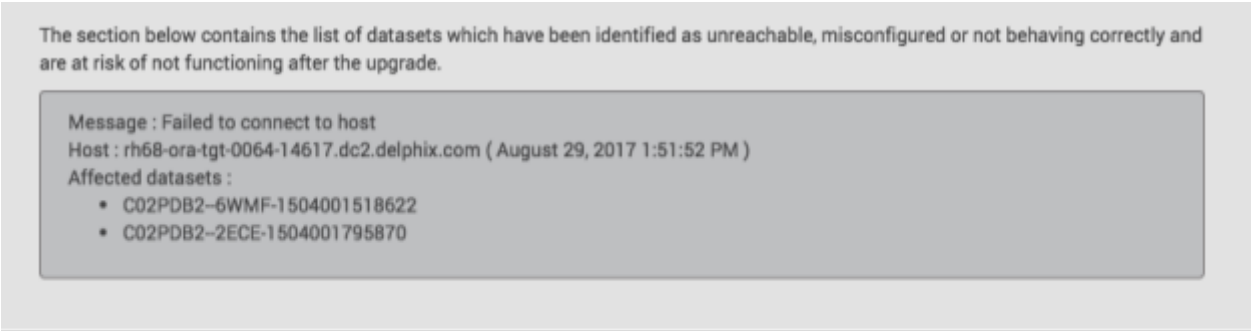
When an environment, dataset, or application is unreachable or misconfigured, the upgrade may encounter access errors, and may not be able to re-start it after the upgrade has been applied.

If you know that a dataset is unavailable for some reason (e.g. host not reachable on the network, or the hosting server is down), it is recommended that you **disable** it (if necessary utilize **Force Disable**).

The **Impact** of **Upgrade** section in the **Details** tab contains the list of datasets that have been identified as unreachable, misconfigured or not behaving correctly and are at risk of not functioning after the upgrade.

Review the warnings in this section (if any) and take appropriate actions before applying the upgrade.

281 http://download.delphix.com/



Screenshot of warning in the "Impact of Upgrade" section

7.4 Downloading the upgrade image

The upgrade file for the version to which you want to upgrade should be downloaded from the Delphix download site[282]. From 6.0.2.0 onwards, the upgrade images are packaged with the latest version of upgrade pre-checks. Please use the latest upgrade image from the download site since the checks will be updated on a need-by basis.

Delphix Upgrade images are approximately 5GB in size; it is recommended to have both a fast internet connection to the Delphix download site as well as to the Delphix Engine.

The upgrade image should be downloaded or moved to a location accessible to the computer used for navigating the Delphix Management application.

> **Version compatibility and support pre-checks**
> Please refer to Upgrade matrix (see page 454) before upgrading to a newer version.
> Customers running version 5.3.9 and earlier that are requesting an upgrade to 6.0.0.0 and above, please contact Delphix Support to help coordinate this upgrade.
> Upgrading from 6.0.x to 6.0.x includes pre-checks packaged in the upgrade image, contacting Delphix Support for this upgrade is **not required** (e.g. 6.0.0.0 → 6.0.9.0).

For upgrades from versions > 5.3.6.0 and < 6.0.0.0, you will download a migration image that is specific to the virtualization platform that your Delphix Engine is running on. The platforms for which images are provided include:

- Amazon AWS
- Microsoft Azure
- VMware ESX

For example, an ESX migration image will have a filename like `Delphix_6.0.0.0_2019-12-06-09-07_Standard_ESX_Migration.tar.gz`.

282 https://download.delphix.com/

For upgrades from versions >=6.0.0.0, you will download an upgrade image that is generic to all the virtualization platforms that your Delphix Engine is running on. The Upgrade image will have a filename like `Delphix_6.0.1.0_2020-03-17-00-39_Standard_Upgrade.tar`.

SHA256 checksums, which can be used to verify the integrity of any downloaded files, are available via download.delphix.com[283]. Each download link should list a `sha256sum` checksum underneath. Additionally, newer releases may also contain a downloadable SHA256SUMS file, containing the checksums of all files in that directory.

To calculate the checksum of downloaded files, and confirm that they match those in the SHA256SUMS file, the following utilities can be used:

- On Windows, `certutil -HashFile <upgrade.tar.gz> SHA256`
- On macOS or Unix servers with Perl installed, `shasum -a 256 <upgrade.tar.gz>`
- On Unix servers with `coreutils` installed, `sha256sum <upgrade.tar.gz>.`

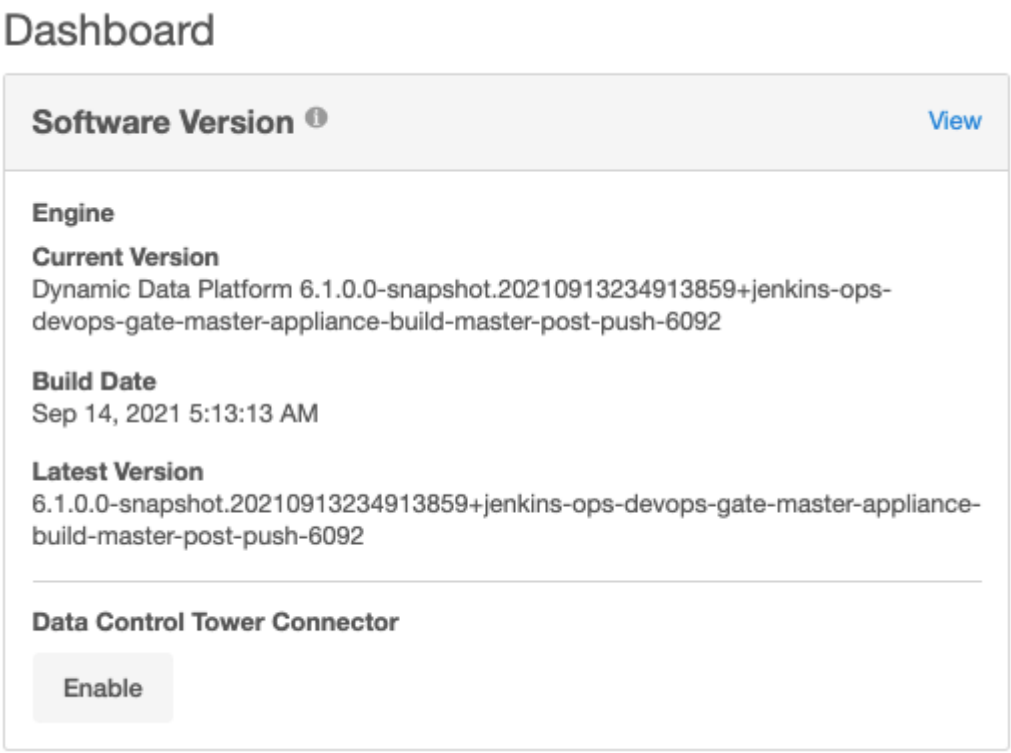
7.5 Uploading the upgrade image

Version compatibility and support pre-checks
Please refer to Upgrade matrix (see page 454) before upgrading to a newer version.
Customers running version 5.3.9 and earlier that are requesting an upgrade to 6.0.0.0 and above, please contact Delphix Support to help coordinate this upgrade.
Upgrading from 6.0.x to 6.0.x includes pre-checks packaged in the upgrade image, contacting Delphix Support for this upgrade is **not required** (e.g. 6.0.0.0 → 6.0.9.0).

The procedure for uploading an upgrade version to the Delphix Engine is:

1. Login to the **Delphix Setup**application.
2. In the Software Version panel, click View.

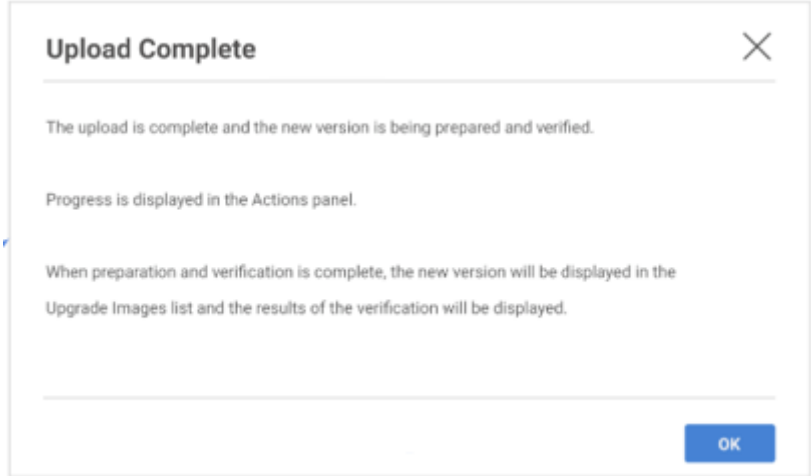
283 http://download.delphix.com



3. Click the **plus** icon to upload a new version.
4. Select the upgrade version you downloaded from the download site.

The upload begins on the drop or selection of a file (upgrade image) into the Upload Upgrade Image window. You will be able to view the progress in the Upload Status area.

5. In the Upload Complete dialog, click OK. You will see the new version appear on the page.



Once the upgrade image upload is complete, upgrade verification will start automatically

This verification process will notify you of any potential problems that require intervention from either you or Delphix support. The new version of the Delphix Engine cannot be installed before verification has completed. You can track the progress of verification from the **Actions** sidebar. If issues are found during the verification process, they will be listed on the version page. An upgrade will not be possible until the issues are resolved.

For more details, see Resolving upgrade checks (see page 935)

7.5.1 Upgrade verification

Version compatibility and support pre-checks
Please refer to Upgrade matrix (see page 454) before upgrading to a newer version.
Customers running version 5.3.9 and earlier that are requesting an upgrade to 6.0.0.0 and above, please contact Delphix Support to help coordinate this upgrade.
Upgrading from 6.0.x to 6.0.x includes pre-checks packaged in the upgrade image, contacting Delphix Support for this upgrade is **not required** (e.g. 6.0.0.0 → 6.0.9.0).

The Delphix Engine provides a feature that allows you to verify an upgrade before applying it. Each new version of the Delphix Engine can introduce new requirements for networking, hypervisor usage, the configuration of Delphix Engine objects and more. Verification can determine if the current Delphix Engine does not meet these requirements. It can also detect if the Delphix Engine is in an unexpected state before an upgrade, in which case the best solution might be to contact support. It is recommended that customers read the release notes[284] for any version they are upgrading to and look for new Delphix Engine requirements.

284 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/13107410/EBS+12.2+plugin+3.0.1+release+notes

As soon as the upgrade image is uploaded (see page 930), a verification job is started. This verification job will use information stored within the uploaded upgrade image to examine the state of the Delphix Engine and make a best effort to validate that applying the upgrade image will succeed. It also will notify the user of any potential problems that require either customer or support intervention.

After resolving problems noted by the verification process, verification can be run again. It is expected that customers will continue to fix problems and re-verify until no more problems remain, or until none of the remaining problems are critical.

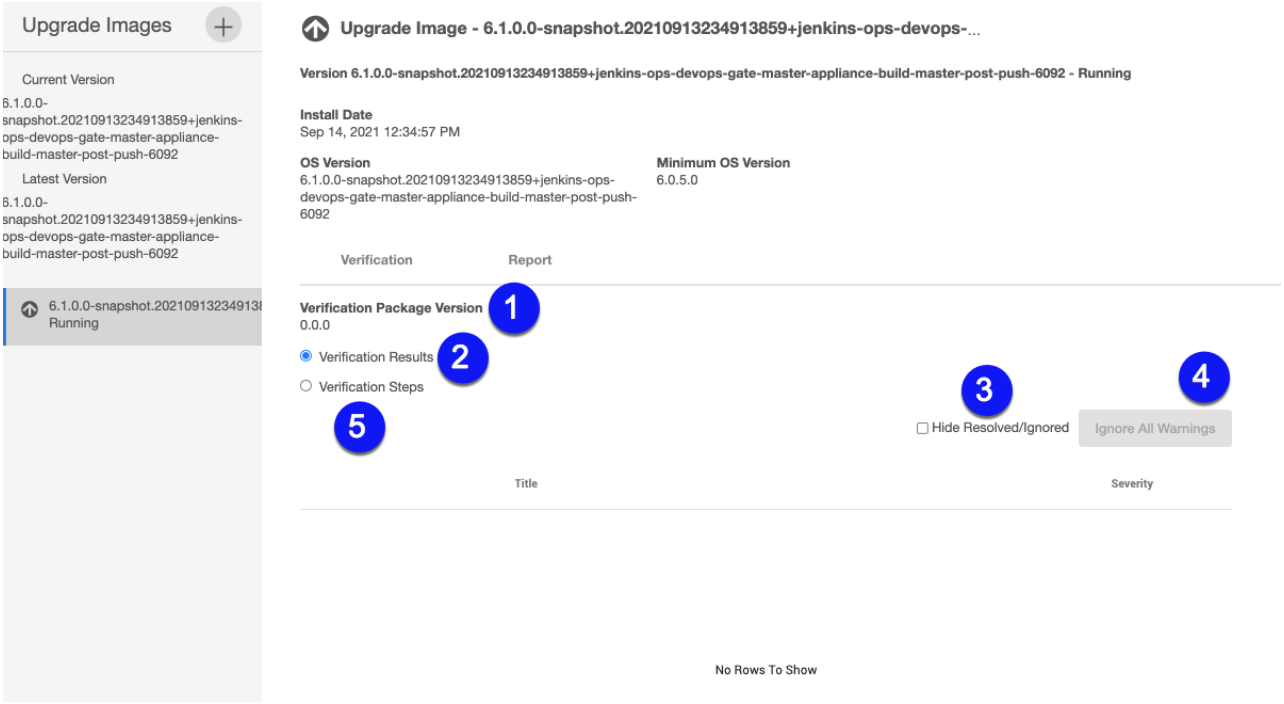
The verification does a “dry run” of some of the upgrade procedures in order to alert the administrator of potential problems before continuing with the upgrade. It is strongly recommended that you perform this verification a day or two in advance, before your upgrade downtime begins, in order to give yourself time to address any problems flagged by the verification. Perform a re-verify closer to the upgrade, when issues have been resolved.

The procedure for verifying an upgrade is:

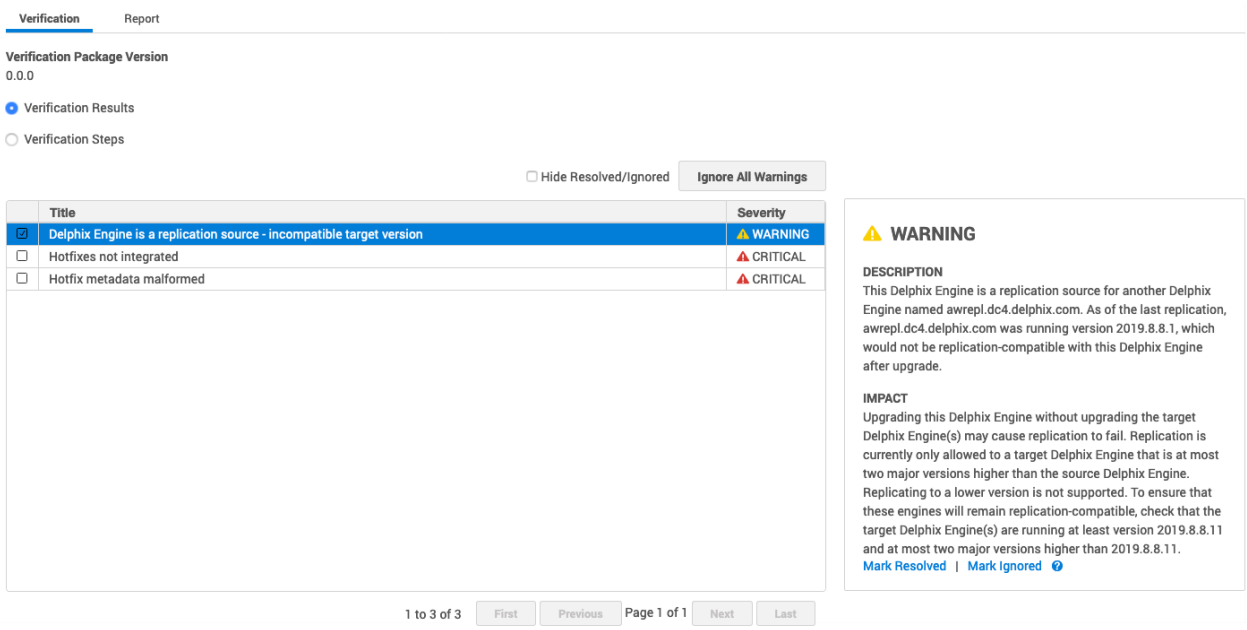
1. Login to the **Delphix setup** application.
2. In the **System upgrade management** panel, click **View**.
3. On the left-hand side, select the version to which you will be upgrading. Details on the version will be displayed on the right.
4. In the upper right-hand section of the Details tab, click **Verify upgrade**. Verification will run in the background. You can view the different stages with the progress during the upgrade process in the **Action** sidebar.
5. Click the **Verification results** option to view problems during verification.
6. Click the **Verification steps** option to view verification steps in detail.

7.5.1.1 Understanding the verification page

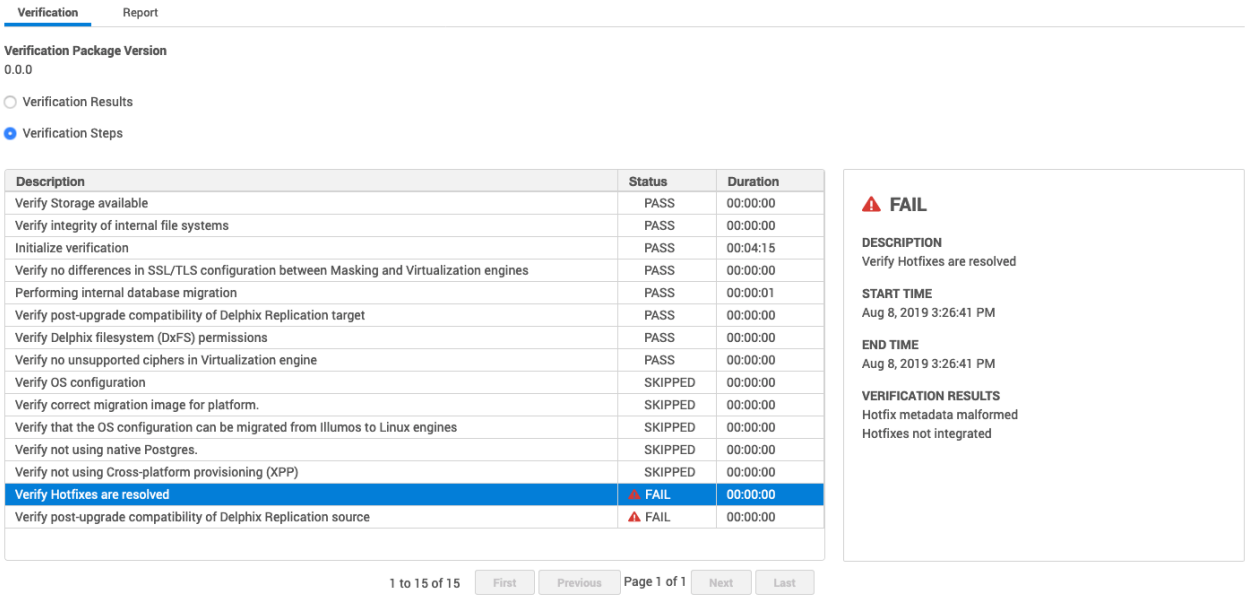
This section provides details about various components on the **Verification** page.



1. **Verification package version** displays the version of verification checks that are bundled with an upgrade version.
2. **Verification results** provides result details about the verification outcomes. This option is selected by default. The verification result detail includes upgrade severity, duration, and reason
3. **Hide resolved/ignored** enables hiding of all verification items that are Resolved or Ignored.
4. **Ignore all warnings** allows you to ignore all warning items during the upgrade process.



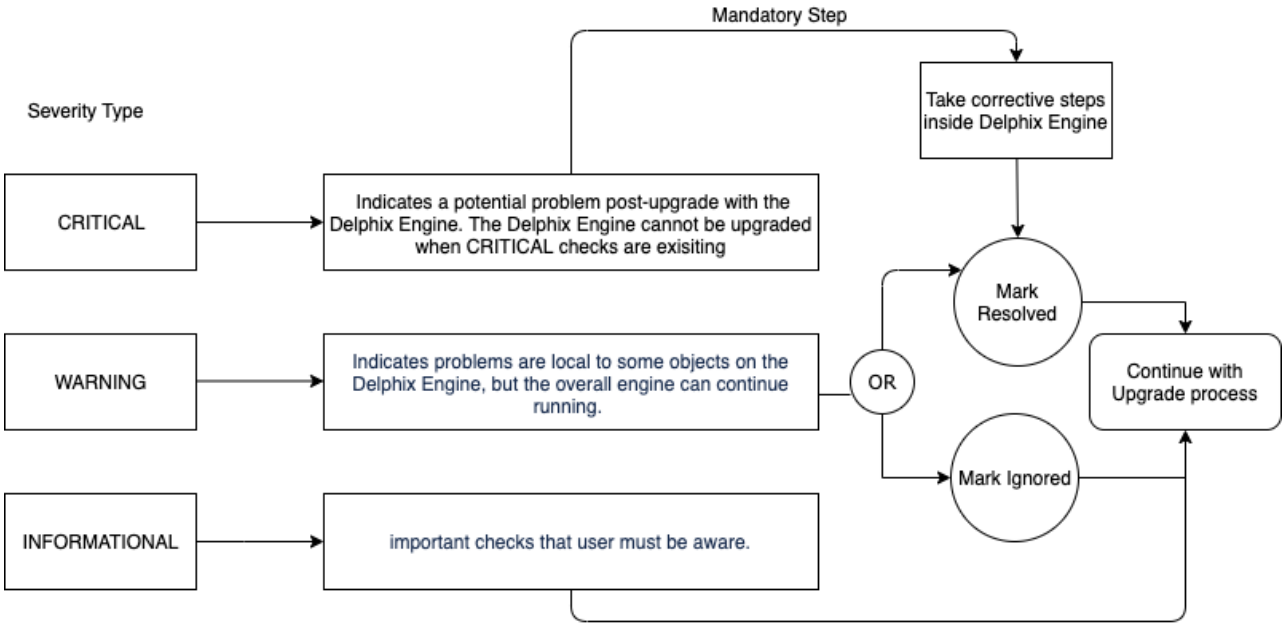
5. **Verification steps** allow you to view a complete list of verification actions, status, and durations.



7.5.2 Resolving upgrade checks

7.5.2.1 Overview

When verifying an upgrade, the verification job can sometimes detect particular action items called upgrade checks. These checks are items that the Delphix Engine is not capable of fixing on its own, and require customer and/or Delphix support action. Every upgrade check that appears must be resolved or ignored before the upgrade can proceed, with the exception of INFORMATIONAL checks which require no action by users. Below is a workflow for various severity types and upgrade check action.



Workflow for Severity Types and Upgrade Check Action

7.5.2.2 Upgrade severities checks

Severity	Description
CRITICAL	These checks indicate a potential problem post-upgrade with the Delphix Engine. The Delphix Engine cannot be upgraded while CRITICAL checks are present; it is likely that the engine will fail to upgrade or break catastrophically post-upgrade.
WARNING	These checks indicate problems that are local to some objects on the Delphix Engine, but the overall engine can continue running. For example, WARNING checks may indicate a problem with the configuration of a particular Windows environment and indicate that if an upgrade occurs, that environment will not function properly.
INFORMATIONAL	These checks provide important information about your upgrade but do not require immediate action.

The following is a list of checks that have been added to the Delphix Engine:

1. **OS tunable settings** (CRITICAL) - Upgrade verification checks that only a certain set of operating system tunables have been adjusted. If tunable settings not on this acceptable list have been changed, please contact Delphix support.
2. **Hotfix** (CRITICAL) - Upgrade verification checks if any hotfixes have been applied to the Delphix Engine, and if any exist that are not resolved in the Upgrade version, this check will generate a Critical result. In this scenario, the Engine may still be upgraded, but we ask that you contact Delphix support to assist in addressing the check.
3. **Snapshot directory visible** (CRITICAL) - This is an internal Delphix Engine problem that Delphix support can resolve for our customers.
4. **Replication** (WARNING) - It indicates that your Delphix Engine has replication set up as either a target or source. Delphix Engines can only perform replication to engines on the same version or higher, so it is recommended to upgrade all Engines in a Replication configuration at the same time, depending on the requirements. For more information on forward-compatible replication, refer to Replication overview (see page 2368).
5. **Storage available** (WARNING) - Upgrade verification checks that there is sufficient space to perform data operations.
6. **SSL/TLS configuration** (WARNING) - Upgrade verification checks that the security configuration of the engine is up to date and properly configured.

An Upgrade check can result in multiple check results. Check results are essentially a to-do list of action items required before performing an upgrade. For example, the OS Tunables will create one line item in the UI for each present tunable that is not on our acceptable list. There are two actions that can be performed on each check result.

7.5.2.3 Upgrade check actions

Action	Description
resolve	You have taken the action necessary to solve the problem and complete the action item. Marking a result as resolved means that the customer believes the check result indicated will no longer be a problem. The next verification run will check the result again, but if it is fixed will not create any new check results.
ignore	You have no intention of fixing the problem indicated by the check result. Future verification runs will not generate that check result again if it is present. CRITICAL severity checks cannot be ignored.

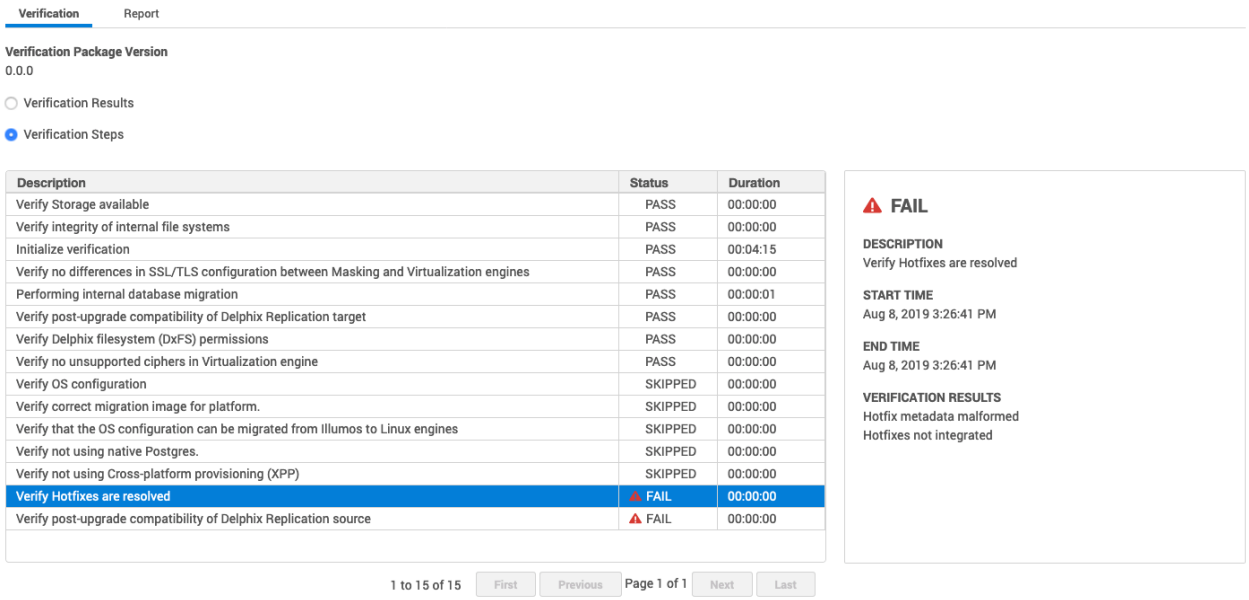
7.5.2.3.1 Examples of upgrade check actions

7.5.2.3.1.1 Case A: Verification results from critical severity check

Consider a case where Delphix Engine previously had a few hotfixes installed by Delphix Support to resolve an issue.

During the upgrade process, a critical severity message is displayed indicating the hotfixes that was used to resolve issue might be lost or reappear after the upgrade.

Hence new upgrade version is missing the root cause fix for that issue. In this example, the action is either to contact Delphix Support or upgrade to a new version that no longer needs these hotfixes. After taking the necessary action, you must mark resolved for the verification steps to complete for a successful upgrade.



1 It is mandatory to resolve all critical severity errors

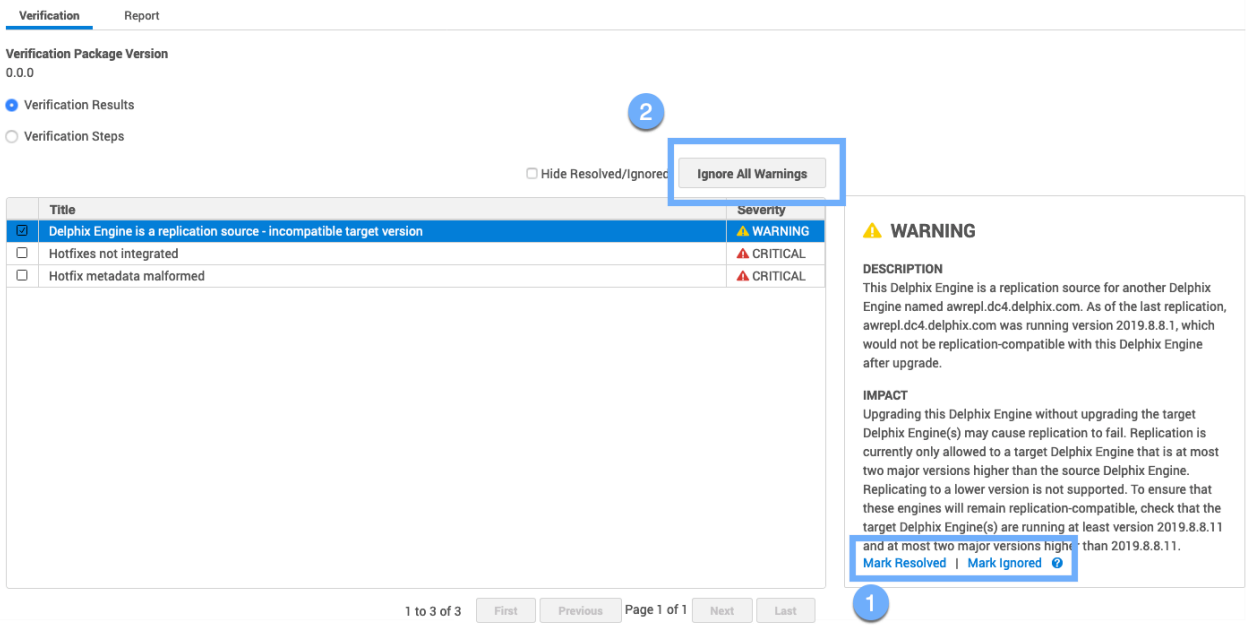
7.5.2.3.1.2 Case B: Verification results warning severity check with ignore all warnings

Consider a case where Delphix Engine is configured as a Delphix Replication source for another Delphix Engine.

During the upgrade process, a warning message appears because the target engine is on a version that is incompatible with the source engine version.

Hence replication is interrupted until the target engine is upgraded to a newer version.

In this example, the action steps are either to mark the issue as **Mark resolved** or **Mark ignored** because a replication-compatibility is acceptable with a few limitations (as indicated in below warning message).



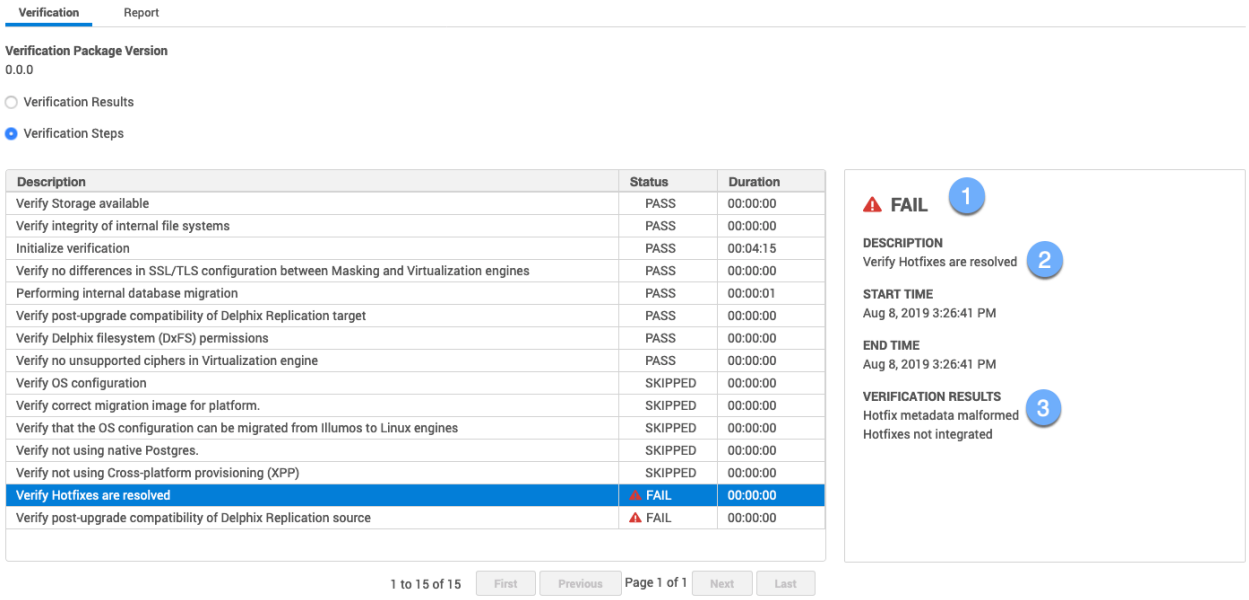
1 All Warnings must either be **Mark resolved** or **Mark ignored**

2 You also can use **Ignore all warnings** option to disregard all warning messages at once.

7.5.2.3.1.3 Case C: Verification steps upgrade failure

When you want to see more details about the verification process use the **Verification steps** radio button. Verification Steps is an alternate way to view the verification status. It includes all the steps that were performed by the verification process, the status, description and the duration of the step.

In this example below verification status, **FAIL** indicates that a verification result was generated. In the **Verification results** view, a **FAIL** status on this screen may correspond to a CRITICAL, WARNING, or INFORMATIONAL severity.



1 Failure severity

2 The description indicates why upgrade failure occurred

3 Verification results

7.6 Applying the upgrade

> **Version compatibility and support pre-checks**
> Please refer to Upgrade matrix (see page 454) before upgrading to a newer version.
> Customers running version 5.3.9 and earlier that are requesting an upgrade to 6.0.0.0 and above, please contact Delphix Support to help coordinate this upgrade.
> Upgrading from 6.0.x to 6.0.x includes pre-checks packaged in the upgrade image, contacting Delphix Support for this upgrade is **not required** (e.g. 6.0.0.0 → 6.0.9.0).

Once you have uploaded an upgrade version, verified the upgrade, optionally reviewed the warnings in the Impact of Upgrade section, scheduled downtime pertaining to the type of upgrade you are performing, you can apply the upgrade.

1. Login to the **Delphix setup** application.
2. In the **Upgrade images** panel, click **View**.
3. On the left-hand side, select the version to which you will be upgrading.
4. Click **Apply upgrade** to initiate the upgrade process.

a. For upgrades on engines already running 6.0.0.0 and greater, the user can choose the type of upgrade they want to perform.

The upgrade will run in the background. You can view the progress of the upgrade in the Action sidebar. Only the current system admin user can view the progress.

The status of the upgrade will be visible on the screen - if the upgrade is successful, the page will be redirected to the login view.

If an **Apply now** upgrade fails, the appliance will automatically roll back to the version running prior to the upgrade.

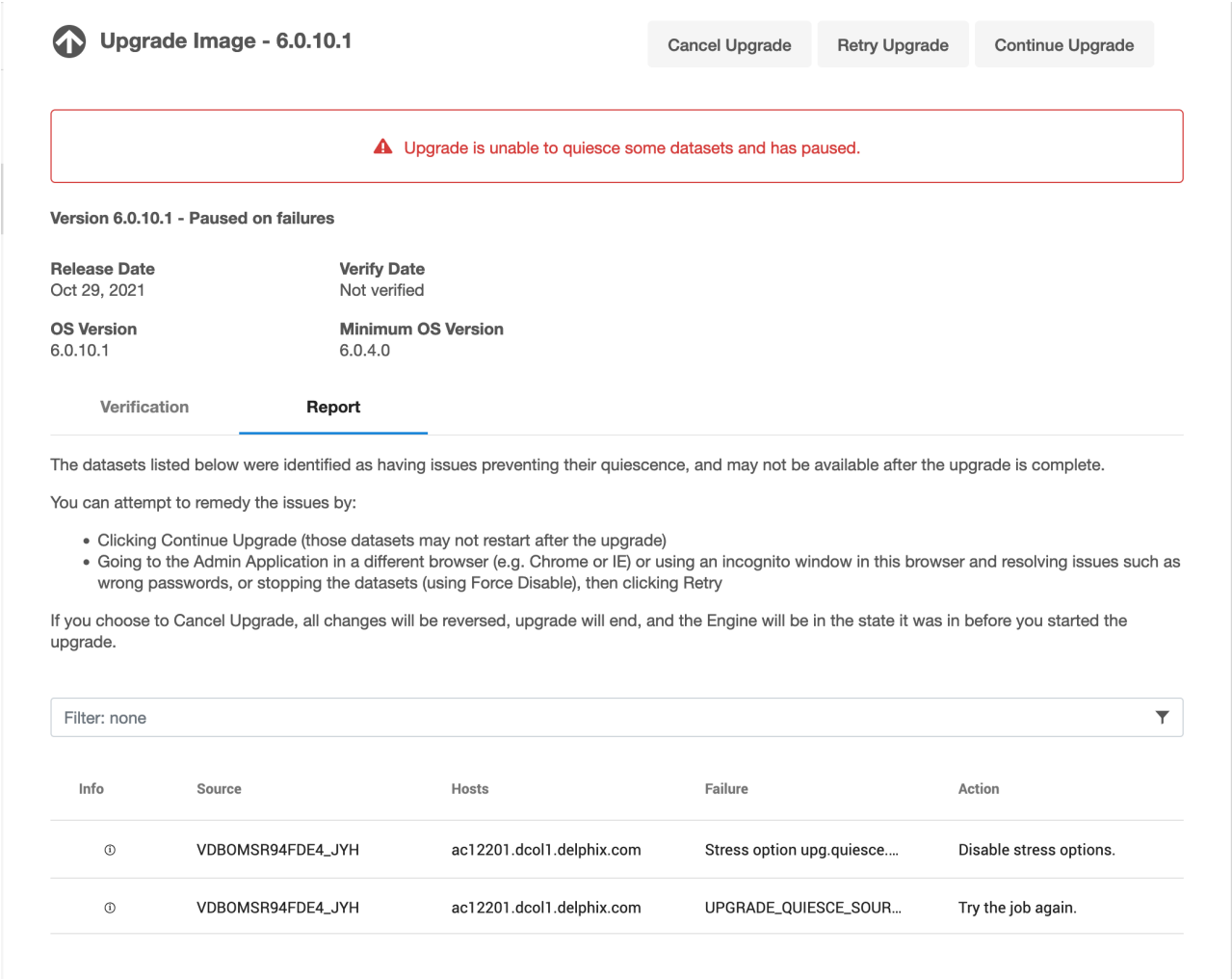
The version page will show the new version in an UPLOADED state and the Action sidebar will show that a rollback was performed. If automatic rollback was disabled through the CLI (not advised), you will have to contact support to proceed further, since you may not even be able to log in to the Delphix Engine.

7.6.1 Failure to quiesce a dataset

If Upgrade has failed to quiesce a dataset, it will pause and you will see the banner at the top of the **Upgrade** page as shown below:

While the upgrade is paused, datasets that have been quiesced are unavailable until you either roll back or continue the upgrade.

To review the list of failures, open the **Report** tab:



The datasets listed in the report were identified as having issues that prevented them from being quiesced, and may not be available after the upgrade is complete. Review the messages in the report and take the suggested corrective actions.

If you think that the errors may be the result of transient failures, you can click **Retry upgrade** to try again. Otherwise, it is recommended that you manually quiesce datasets that are still running. To do so:

1. Use a different browser or use an incognito window to go to the **Delphix management** application.

> Failure to do so might result in losing the current session and you may not be able to return to the Upgrade pause screen.

2. Either resolve issues such as a wrong password or stop the dataset using **Force disable**.
3. In the original browser or window, click **Retry upgrade** to try applying the upgrade again.

If you want to ignore the failures to quiesce datasets and proceed with the upgrade:

1. Click **Continue upgrade**. This will attempt to quiesce all datasets which have not yet been quiesced, but will not pause on failures.

   Note : This may result in datasets remaining unavailable after the upgrade is complete and the Delphix Engine restarts, since the underlying storage that backs the datasets will be unreachable during the upgrade. This may cause the databases or applications to failover or transition to a failure state, thus requiring administrator intervention to recover.
2. Review the messages in the report and take the suggested corrective actions.
3. If any of the listed datasets are critical, and you are unable to resolve the configuration errors in the report, you can **Rollback** the upgrade. If you choose **Rollback**, all changes will be reversed, the upgrade will end, and the Delphix Engine will be in the state it was in before you started the upgrade.

7.7 Post upgrade

After the upgrade is done, you will be redirected back to the login page.

Login to **Delphix setup** to make sure that the upgrade succeeded and that the new version is running.

A **post-upgrade cleanup** job is run automatically after the upgrade is done. This job refreshes environments and re-enables sources to bring objects back into a working state. VDB access will not be available at all until the environments have been refreshed and the objects are re-enabled by the job **Perform cleanup tasks following a Delphix engine upgrade**.

7.7.1 Failures

If you used **Continue upgrade** to force the upgrade as is described in Failure to quiesce a dataset (see page 940), the **Report** tab will contain the list of pre-upgrade quiesce failures. Upgrade will make a best effort to restore functionality to these datasets, but it may still hit the same errors that prevented the datasets from being quiesced successfully before the upgrade. You may need to manually bring up these datasets on the target hosts.

8 Security

Given that the Delphix software is meant to interact with some of your organization's most important application data, security is very important. The security section covers everything you need to know about how you can manage the Delphix software securely (user management, etc) as well as how Delphix works to ensure that the software is not susceptible to vulnerabilities. This section outlines standard hardening techniques that you should apply to every Delphix Engine in your environment. The Delphix Engine delivers its functionality through a web-based graphical user interface (GUI), web-based RESTful APIs, and a command-line interface (CLI) accessed over SSH. All three interfaces leverage the same accounts and privileges scheme. Access to the Delphix Operating System (DxOS) is restricted to Delphix Support and Delphix Professional Services and can be controlled through the Support Access Control mechanism, described below. Customer-specific software installations and modifications to the DxOS are not required or supported. Communications to/from connected systems ("sources" and "targets") should be limited to required ports, and encryption should be used wherever possible.

This section covers the following topics:

- Security principles (see page 944)
- Product security (see page 945)
- Replication security (see page 991)
- Object security (see page 991)
- System configuration (see page 995)
- GUI security (see page 996)
- Repave Delphix Engine (see page 997)
- Masking sensitive data (see page 1006)
- Audit logs (see page 1006)
- Support security (see page 1007)
- Password policies (see page 1007)
- Additional topics (see page 1009)

8.1 Security principles

The Delphix approach is based on:

Embrace separation of duties: Isolate and compartmentalize capabilities and privileges and never give or concentrate access to a single role.

Apply the principle of least privilege: Users should obtain only those privileges needed to do their jobs and only for as long as they are needed.

Use an open, simple design: Make security mechanisms simple and easy to use, and rely on proven, peer-reviewed solutions.

Use a layered defense: Provide no single point of failure; if one layer fails to catch an error, catch it in another layer.

Use complete mediation and authentication: Control and check every access point every time.

Use fail-safes: Deny access when not explicitly authorized. Prevent faults from causing an opportunity to compromise.

Protect data at rest and data in motion: Utilize common security protocols as well as features of the source database and database software to protect data at all times.

Minimize the attack surface: Present the minimum sockets, services, webpages, and accounts necessary to operate.

Don't rely on obscurity: Be secure even if everything but the key is known.

Audit and monitor everything: Provide a tamper-proof trail of evidence.

Leverage the environment: Design the Delphix Engine to leverage the security features offered by databases, operating systems, storage devices, and networks.

Anticipate external attack vectors: Combat attacks sourced from connected systems.

Enforce strong credentials: Define and enforce password policies.

8.2 Product security

This section covers the following topics:

- Delphix product security (see page 945)
- Software updates (see page 946)
- Network security (see page 946)
- Password vault support (see page 948)
- Certificate management (see page 965)
- Replication security (see page 991)

8.2.1 Delphix product security

8.2.1.1 Overview

Delphix takes an active approach towards the discovery of vulnerabilities by employing third-party VAPT (vulnerability and penetration testing) Teams on an annual basis, as well as actively reviewing additional tools to assess our products. We also actively monitor bugs and vulnerabilities against the list of open-source software which is integrated into our products.

Delphix investigates all known vulnerabilities, either found by customers or internally, or discovered by the VAPT Teams. During the course of this investigative process, Delphix works with the reporter of the vulnerability to gather the technical information and determine the appropriate remedial action.

Customers can perform their own security scans and audits using their access through application administrative accounts. Because operating system administrative access is not provided with the closed software virtual appliance, scans executed with this privilege level are consistent with the level of access available in the environment.

As needed, we provide compensating controls and mitigations to reduce the impact of security issues more quickly and deliver fixes as part of the standard software release process described above.

8.2.1.2 Software delivery security

We deliver our virtual software appliance and upgrade images to customers on a secure server with access control. Additionally, we provide cryptographic signatures for each image, and customers can use these signatures to ensure the software images have not been tampered with.

8.2.1.3 Ancillary components

The virtual appliance communicates with target servers running virtual databases (VDBs). The software is pushed to target servers to facilitate communication for Oracle and SQL Server databases. (Oracle toolkit and Delphix connector, respectively). This software lives outside of the DDDP product appliance, and new versions of the DDDP product will provide updated versions of this software as well.

8.2.2 Software updates

Keeping Software up to date is an important part of any hardening plan. Delphix software releases are cumulative and include bug fixes, new features, and security improvements. In addition, Delphix releases hotfixes, procedures, and workarounds for critical vulnerabilities.

8.2.2.1 Patch annually

To stay up to date, patch your Delphix Engine at least once per year. If you do not, you might have to upgrade twice to get the latest releases, and your old installed version will not be able to receive vulnerability fixes.

If possible, patch more frequently. Depending on the version you are upgrading from/to, you may be able to avoid or defer the reboot sequence, which defers downtime for your virtual databases (VDBs). This allows you to patch outside of downtime windows.

8.2.2.2 Subscribe to Delphix notifications

Delphix issues email notifications when critical vulnerabilities are discovered. Registered support accounts will automatically receive these notifications. To ensure that you receive these notifications:

- Register at least two Engine Admins with Delphix Support
- Add Delphix Support accounts when Engine Admins leave the company

8.2.3 Network security

Review the official documentation for the full list of required ports, *which depends on your database vendor*. Open only those ports that are required. The following table only lists generic requirements; you will need additional ports to integrate with databases.

Ports

Open only required ports.

8.2.3.1 General port allocation

The Delphix Engine makes use of the following network ports irrespective of the type of database objects on it:

8.2.3.1.1 General outbound port allocations

Protocol	Port numbers	Use
TCP	21	Passive FTP connections from the Delphix Engine to the Delphix FTP server. Used for sending logs to Delphix Support.
TCP	25	Connection to a local SMTP server for sending email.
TCP/UDP	53	Connections to local DNS servers
UDP	123	Connection to an NTP server
TCP/UDP	389	Standard access to an LDAP server
TCP/UDP	636	Secure access to an LDAP server
TCP	1023	Used to complete the passive FTP connection from the Delphix Engine to Delphix Support. These ports are not used if Delphix uses a proxy server to connect to the Internet.
TCP	8415	A Delphix replication source will connect to the replication target using this destination port

8.2.3.1.2 General inbound port allocations

Protocol	Port numbers	Use
TCP/UDP	22	SSH connections to the Delphix Engine
TCP	80	HTTP connections to the Delphix GUI
TCP	443	HTTPS connections to the Delphix GUI
TCP	8415	A Delphix replication target will accept incoming connections to this port from the replication

8.2.4 Password vault support

8.2.4.1 Overview

More and more organizations use Enterprise Password Vaults (EPV) such as CyberArk and HashiCorp Vault to store securely and centrally manage identities and credentials. Delphix has added CyberArk, HashiCorp and Azure Key Vault support to the Delphix Virtualization Engine as a new authentication option for environments and databases. This minimizes the number of places where credentials need to be stored and, therefore the risk of insecure storage.

The Delphix Engine uses various authentication methods such as username/password, username/ssh key, and Kerberos credentials when connecting to hosts and databases from the Engine. These credentials are stored on the Delphix Engine in an encrypted format and can be retrieved later to perform various operations. Delphix provides an additional authentication method by integrating Virtualization with the most common vault types (CyberArk, HashiCorp. Azure Key Vault). At runtime, Delphix retrieves the credentials (passwords, ssh keys) from the customer's vault servers via API calls and avoids managing customer passwords.

8.2.4.2 Configuring password vaults

In the Setup app, system administrators can manage (add, delete, modify, and validate) vault configurations during and after the initial setup. Each engine can have multiple vaults configured of any type.

The authentication method supported for CyberArk is Certificate-based. Configuring a CyberArk vault requires providing a host address, port number, application ID, and a client certificate (certificate chain and private key).

The authentication methods supported for HashiCorp vaults are Token-based, AppRole-based, and Certificate-based. Configuring a HashiCorp vault requires providing a token ID or a role ID and secret ID or a client certificate along with the host address and port number. For HashiCorp Enterprise vaults, a namespace can also be provided.

The authentication method supported for Azure Key Vault is Client-Secret-based. Configuring an Azure Key Vault requires providing the Azure tenant ID, client ID, and client secret.

8.2.4.3 Using password vaults

The Virtualization engine retrieves credentials at runtime from a vault using a unique identifier that locates a set of credentials in a configured vault. This occurs for any activity that requires Environment access (SnapSync, Validated Sync, LogSync, as well as Environment monitoring). This may result in a significant number of requests, so any existing connection rate limits should be evaluated and adjusted accordingly. For CyberArk, the unique identifier consists of a query string. For HashiCorp Vault, it consists of four parameters: engine, path, and a pair of keys that locate the username and secret (password or SSH key) in the key-value store at that engine and path. For Azure Key Vault, it consists of the Azure Vault Name (also known as the Resource Name within Azure Vault, not to be confused with the vault name that identifies the vault in the engine), username key, and secret key.

To set up an environment or database user to use a vault, use the credential type **VaultCredential** when adding/modifying such users and specify the vault and the unique identifier of the credentials.

8.2.4.4 Roles and privileges for CyberArk and HashiCorp users

Role	Privileges
System Administrator	Can add, modify, delete, and list vault configurations.
Delphix Administrator	Can list existing vault configurations and link environment and database users to vault credentials.

8.2.4.5 Supported environments and databases

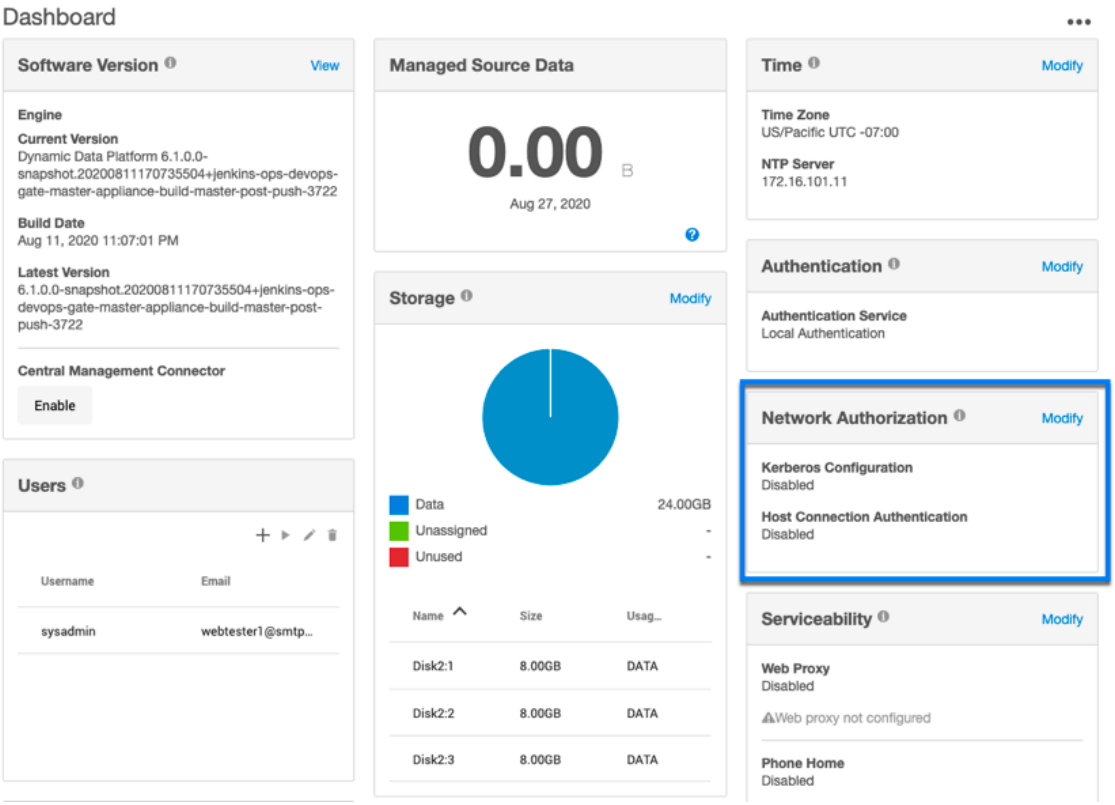
All environment users can use vault credentials. For Windows, the initial link via the Delphix Connector does not support vaults, but environment users can be subsequently updated to use vaults.

Vault integration is currently supported for SAP ASE database users, Oracle database users, and MSSQL domain users using 6.0.4 and later.

8.2.4.6 Setting up a vault via GUI

Complete setup via the GUI is available for CyberArk as of 6.0.3.0, for HashiCorp as of 6.0.4.0 and for Azure Key Vault as of 10.0.0.0.

1. Connect to the Delphix Engine http://
2. Add a CyberArk or HashiCorp CA certificate to the TrustStore as part of the initial configuration. Refer TrustStore Settings (see page 989) for steps to add a CA certificate.
3. Click on the **Modify** link in the top right of the **Network Authorization** panel.



4. On the Network Authorization window, click "**+**" to add a new vault.
5. Enter the following information: Depending upon your requirements, you can set the configurations for CyberArk, HashiCorp, or Azure Key vault.

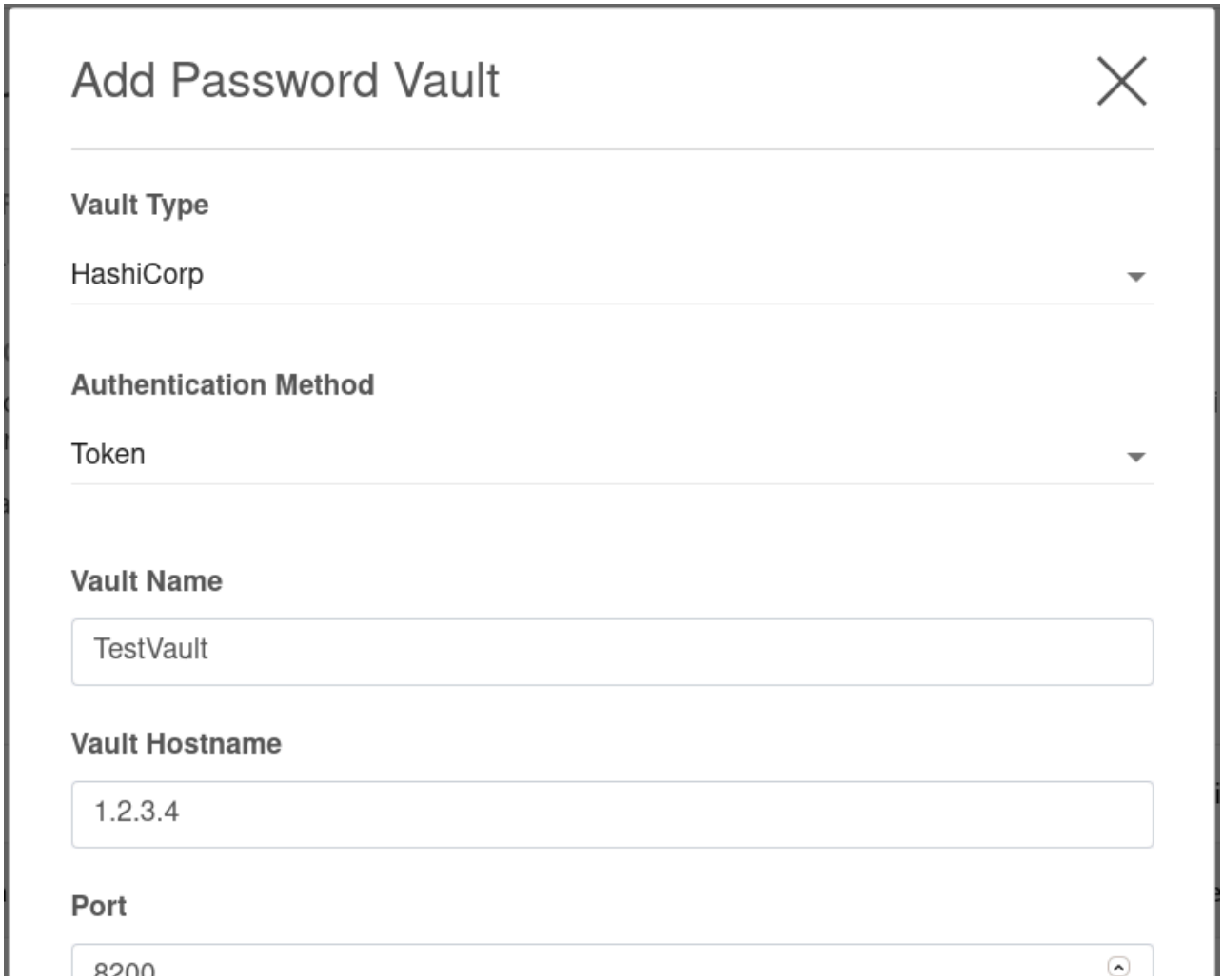
Field	Possible value and Data Type	Description
Vault Type - CyberArk		
Vault Name	*<user-specified>* Accepts a string value	Specifies the user-specified vault name
Vault Hostname	mycyberark.myorg.com Accepts a URL string value	Specifies the location of the user's vault server
Port	443 Accepts an integer value	Specifies the port number through which the communication will happen

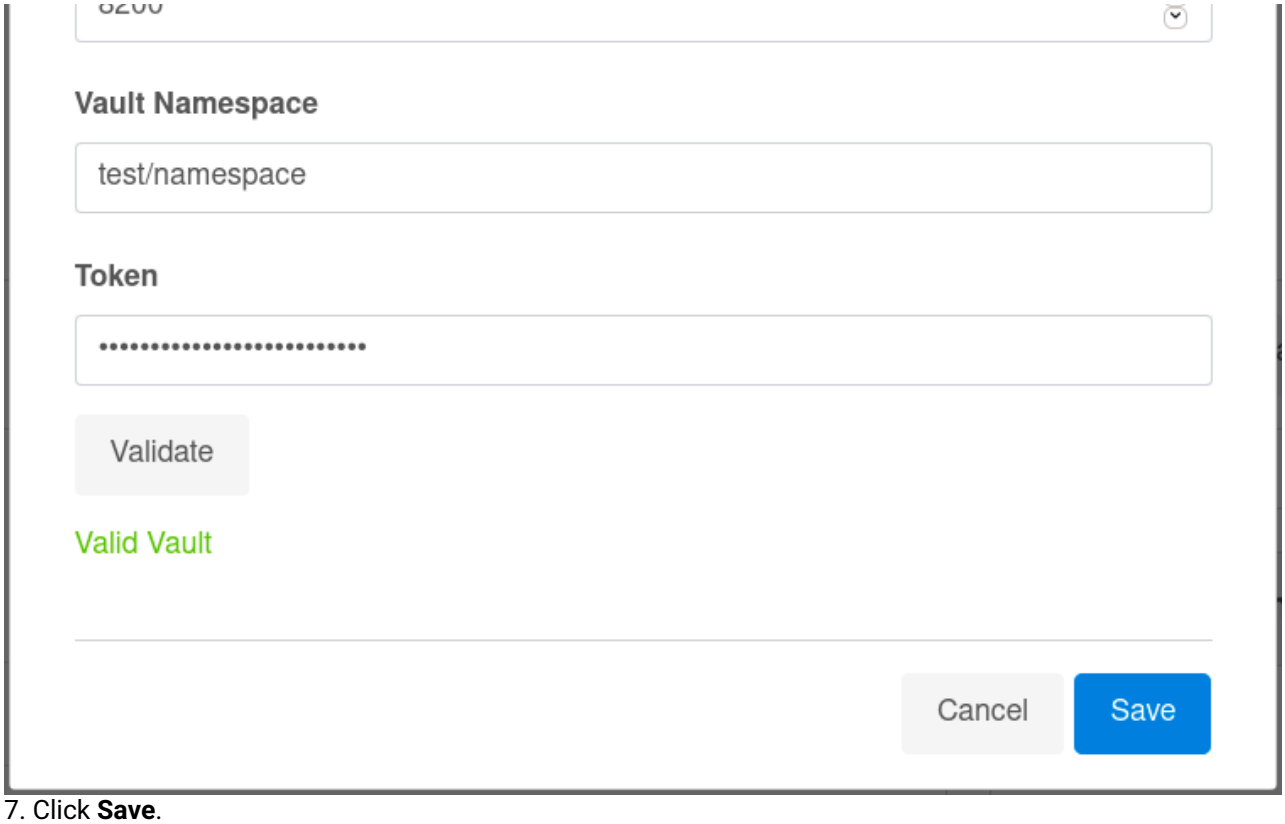
Field	Possible value and Data Type	Description
App ID	MyAppID Accepts a string value	Specifies an application ID registered with and provided by CyberArk
Authentication Certificate	——-BEGIN CERTIFICATE—-— *<certificate>* ——-END CERTIFICATE—-— Accepts a string value	Specifies the authentication certificate provided by CyberArk
Private Key	*<CyberArk-provided>* Accepts a string value	Specifies the private key provided by CyberArk for TLS based authentication
Vault-Type - HashiCorp		
Authentication method	**Token**	
Vault Name	*<user-specified>* Accepts a string value	Specifies the user-specified vault name
Vault Hostname	12.345.678.90 Accepts a URL string value	Specifies the location of the customer vault server
Port	8100 Accepts an integer value	Specifies the port number through which the communication will happen
Vault Namespace	purple Accepts a string value	Specifies the namespace configuration specific to the user environment that is provided by the HashiCorp Enterprise Platform
Token	s.abcdefghijklmnopqrstuvwxyz123.waR7a Accepts a string value	Specifies the token specific to the user environment that is provided by the HashiCorp Enterprise Platform
Authentication method	**AppRole**	
Vault Name	*<user-specified>* Accepts a string value	Specifies the user-specified vault name

Field	Possible value and Data Type	Description
Vault Hostname	12.345.678.90 Accepts a URL string value	Specifies the location of the customer vault server
Port	8100 Accepts an integer value	Specifies the port number through which the communication will happen
Vault Namespace	purple Accepts a string value	Specifies the namespace configuration specific to the user environment. This feature is provided with the HashiCorp Enterprise Platform
RoleID	abcdefg123-4a56-7890-a2bc-34567def8901 Accepts a string value	Specifies the RoleID specific to the user environment
SecretID	ab1cde0f-123g-4h56-i789-1234jk567890 Accepts a string value	Specifies the SecretID specific to the user environment
Authentication method	**Certificate**	
Vault Name	<*user-specified*> Accepts a string value	Specifies the user-specified vault name
Vault Hostname	12.345.678.90 Accepts a URL string value	Specifies the location of the customer vault server
Port	8100 Accepts an integer value	Specifies the port number through which the communication will happen
Vault Namespace	purple Accepts a string value	Specifies the namespace configuration specific to the user environment. This feature is provided with the HashiCorp Enterprise Platform

Field	Possible value and Data Type	Description
Authentication Certificate	——-BEGIN CERTIFICATE—-— *<certificate>* ——-END CERTIFICATE—-— Accepts a string value	Specifies the authentication certificate provided by HashiCorp for TLS based authentication
Private Key	abcdefg123-4a56-7890-a2bc-34567def8901 Accepts a string value	Specifies the private key specific to the user environment that is provided by HashiCorp for TLS based authentication
Role Name (Optional)	purple-admin-role Accepts a string value	Specifies the certificate role name for TLS based authentication

6. Click **Validate** to check the configurations before saving the vault details. The below screenshot shows an example of the HashiCorp Vault configuration.





7. Click **Save**.
The added configurations can be viewed in the Network Authorization window.

8.2.4.6.1 Editing a vault via the GUI

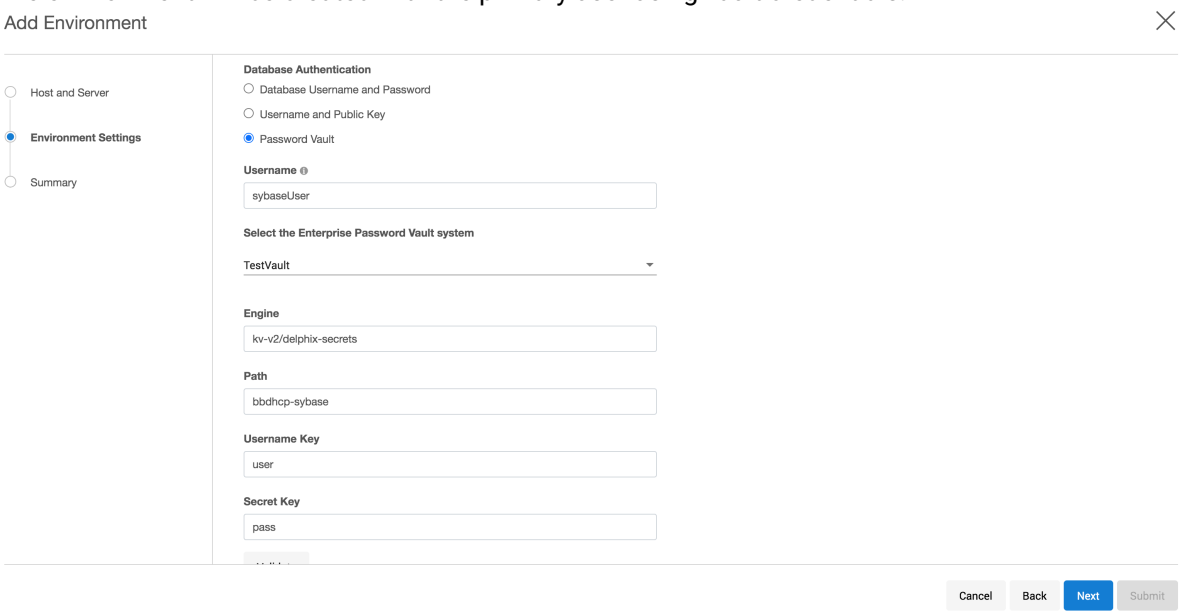
1. Connect to the Delphix Engine http:// <Delphix Engine>/login/index.html#serverSetup.
2. Click on the **Modify** link in the top right of the **Network Authorization** panel.
3. In the **Network Authorization** window, select a vault, then the **pencil** icon.
4. Edit your configuration.
5. Select **Edit**.

8.2.4.6.2 Deleting a vault via the GUI

1. Connect to the Delphix Engine http:// <Delphix Engine>/login/index.html#serverSetup.
2. Click on the **Modify** link in the top right of the **Network Authorization** panel.
3. In the **Network Authorization** window, select a vault, then select the **trashcan** icon.
4. Select **Yes** to delete the vault.

8.2.4.6.3 Adding a host user for HashiCorp

1. Login to the **Delphix Management Application** and select **Manage > Environments**.
2. Select Add **Environment**.
3. In the **Environment Setting** tab, select **Password Vault** as the Login Type.
4. Select the vault configuration and provide the secret engine name, path, and keys for the username secret and complete your environment configuration.
5. The environment will be created with the primary user using vault credentials.



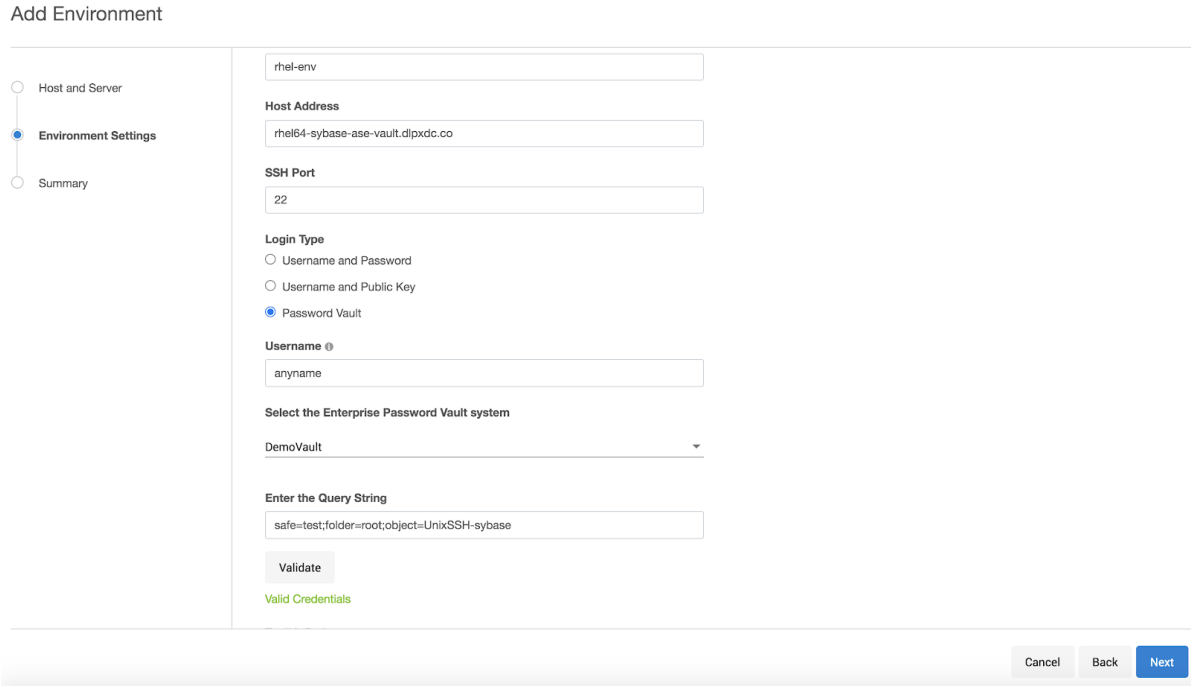
8.2.4.6.4 Adding a database user for HashiCorp

1. Login to the **Delphix Management Application**.

2. Add dSource using database credentials from HashiCorp vault by selecting **Password Vault** as the **Login Type**.
3. Provide the appropriate secret engine name, path, and keys for the username secret and complete configuration.

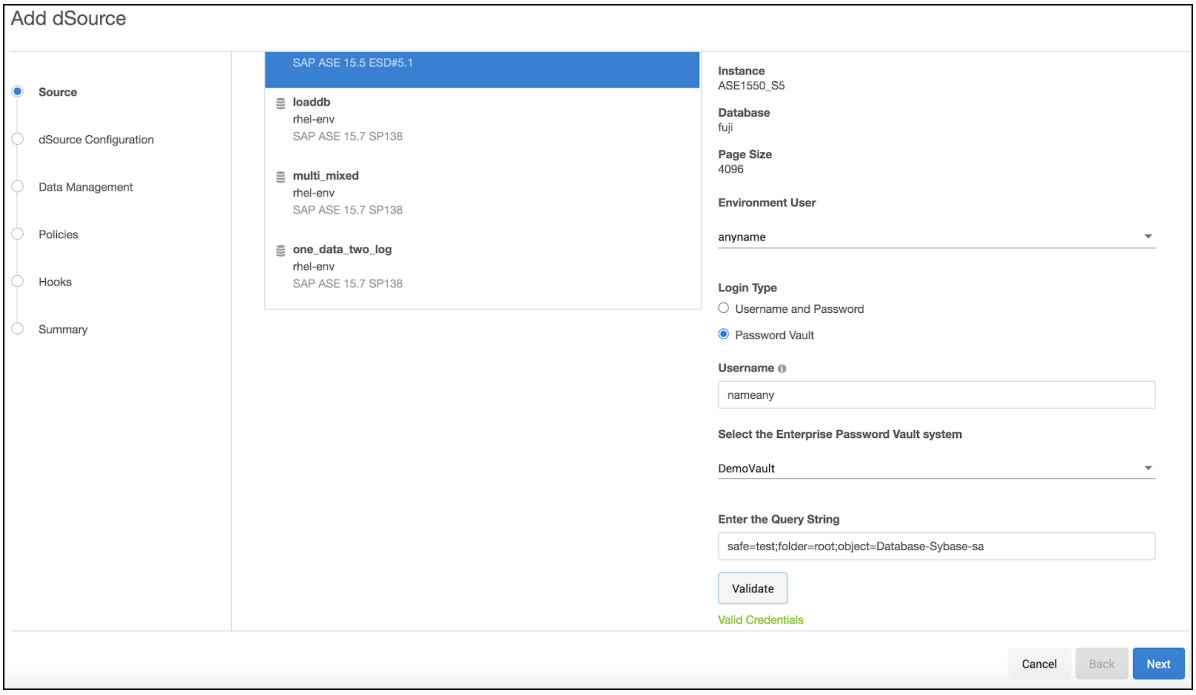
8.2.4.6.5 Adding a host user for CyberArk

1. Login to the **Delphix Management Application** and select **Manage > Environments**.
2. Select Add **Environment**.
3. In the **Environment Setting** tab, select **Password Vault** as the Login Type.
4. Select the vault configuration and provide the username, select the enterprise password vault system, and enter a query string that is a unique identifier pointing to the credentials to be retrieved and complete your environment configuration.
5. The environment will be created with the primary user using vault credentials.



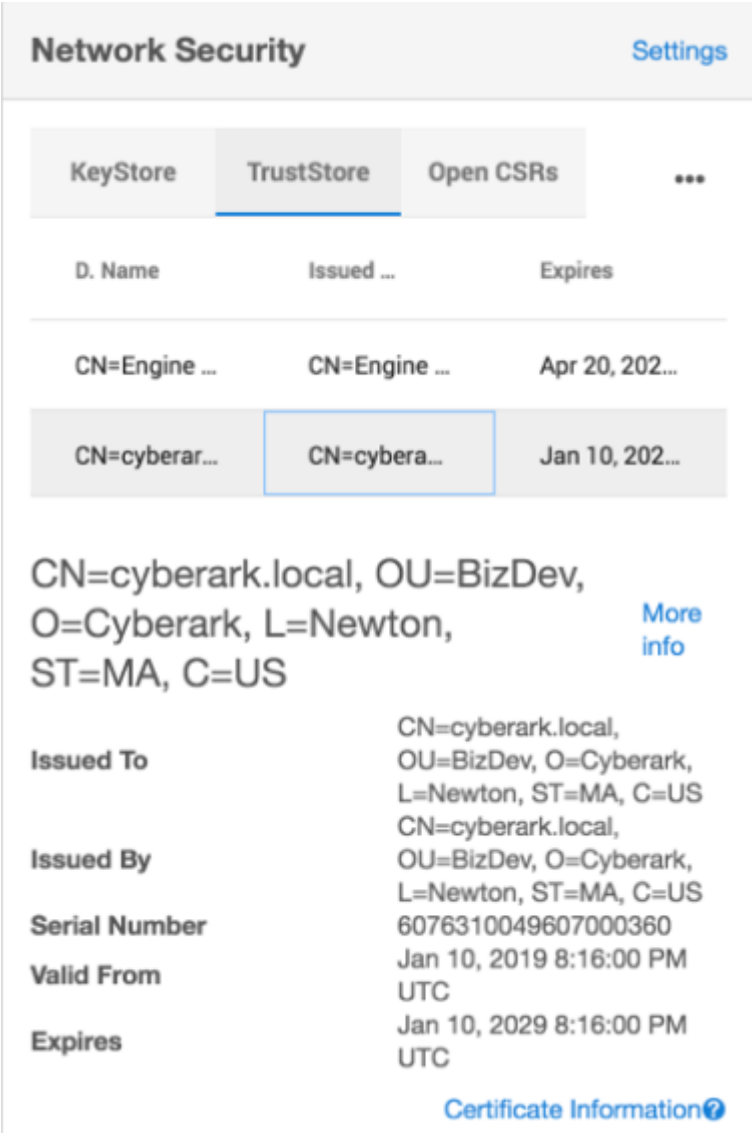
8.2.4.6.6 Adding a database user for CyberArk

1. Login to the **Delphix Management Application**.
2. Add dSource using database credentials from CyberArk vault by selecting **Password Vault** as the **Login Type**.
3. Provide the appropriate query string and complete configuration.



8.2.4.6.7 Setting up a vault via CLI

1. Login as a system administrator and add a CyberArk or HashiCorp CA certificate to the TrustStore as part of the initial configuration.



2. SSH to <Delphix Engine IP>service passwordVault and enter **create**.

```
ip-10-110-230-197 service passwordVault>create
```

3. Add a new vault configuration by entering a name, host, port, applicationId, client certificate, and private key.

```
ip-10-110-230-197 service passwordVault create
Properties
type: CyberArkPasswordVault
   name: DemoVault (*)
   applications: Delphix (*)
   clientCertificate:
```

```
      type: PemClientCertificate (*)
      clientCertificateChain: (required)
      privateKey:(required)
host: services-uscentral.sktyop.com (*)
port: 17993 (*)
```

4. Add a new HashiCorp vault configuration by entering a name, host, port, and other authentication information based on the authentication method (Token/AppRole/Certificate) selected.

   HashiCorp - Token Based Authentication

```
ip-10-110-230-197 service passwordVault create
Properties
    type: HashiCorpVault (*)
    name: HashiCorpDemoVault (*)
    authentication:
        type: HashiCorpTokenAuthentication (*)
        token: ******** (*)
    host: 10.119.132.40 (*)
    port: 8200 (*)
```

   HashiCorp - AppRole Based Authentication

```
ip-10-110-230-197 service passwordVault create
Properties
    type: HashiCorpVault (*)
    name: HashiCorpDemoVault (*)
    authentication:
        type: HashiCorpAppRoleAuthentication (*)
        roleId: 20d19a46-6fd9-c78b-b7e3-e43be4c8d5c2 (*)
        secretId: ******** (*)
    host: 10.119.132.40 (*)
    port: 8200 (*)
```

   HashiCorp - Certificate Based Authentication

```
ip-10-110-230-197 service passwordVault create
Properties
    type: HashiCorpVault (*)
    name: HashiCorpDemoVault (*)
    authentication:
        type: HashiCorpCertificateAuthentication (*)
        clientCertificate:
            type: PemClientCertificate (*)
            clientCertificateChain: (required)
            privateKey: (required)
        roleName: (unset)
    host: 10.119.132.40 (*)
```

```
    port: 8200 (*)
```

8.2.4.6.8 Updating an existing vault configuration

```
ip-10-110-230-197 service passwordVault> select DemoVault
ip-10-110-230-197 service passwordVault 'DemoVault'>update
ip-10-110-230-197 service passwordVault 'DemoVault'update *> set name=TestVault
ip-10-110-230-197 service passwordVault 'DemoVault'update *> commit
ip-10-110-230-197 service passwordVault 'TestVault'>
```

8.2.4.6.9 Deleting an existing vault configuration

```
ip-10-110-230-197 service passwordVault 'TestVault'> delete
ip-10-110-230-197 service passwordVault 'TestVault' delete *> commit
ip-10-110-230-197 service passwordVault>
```

8.2.4.6.10 Adding/Modifying host users

Add an environment with user credentials from CyberArk vault. When adding a host/database user with a vault credential, the name field would be a user identifier and not the actual username. In case this field is empty, a unique identifier is generated with a hash of vault credentials.

```
ip-10-110-230-197 environment create *> set hostEnvironment.name=bbh-env
ip-10-110-230-197 environment create *> set hostParameters.host.address=bbdhcp-vault-
demo.dlpx.co
ip-10-110-230-197 environment create *> set hostParameters.host.toolkitPath="/work"
ip-10-110-230-197 environment create *> set primaryUser.name=oracleUser
ip-10-110-230-197 environment create *> set
primaryUser.credential.type=VaultCredential
ip-10-110-230-197 environment create *> set primaryUser.credential.vault=DemoVault
ip-10-110-230-197 environment create *> set primaryUser.credential.vaultCredentialId="
safe-test;folder=root;object=UnixSSH-sybase"
ip-10-110-230-197 environment create *> commit
    `UNIX_HOST_ENVIRONMENT -6
    Dispatched job JOB-33
    ENVIRONMENT_CREATE_AND_DISCOVER job started for "bbh-env".
    ENVIRONMENT_CREATE_AND_DISCOVER job for "bbh-env" completed successfully.
```

8.2.4.6.11 Adding/Modifying database users

Add dSource using database credentials from CyberArk vault.

```
ip-10-110-230-197 database link *> set name=fuji
ip-10-110-230-197 database link *> set group=Untitled
ip-10-110-230-197 database link *> set linkData.config=ASE_SI_CONF-70
ip-10-110-230-197 database link *> set linkData.dbUser=sybaseUser
ip-10-110-230-197 database link *> set linkData.dbCredentials.type=VaultCredential
ip-10-110-230-197 database link *> set linkData.dbCredentials.vault=DemoVault
ip-10-110-230-197 database link *> set linkData.dbCredentials.vaultCredentialOd="safe
-test;folder=root;object=Database-Sybase-sa"
ip-10-110-230-197 database link *> set linkData.loadBackupPath='/opt/sybase/dumps"
ip-10-110-230-197 database link *> set linkData.sourceHostUser=HOST_USER-7
ip-10-110-230-197 database link *> set linkData.stagingHostUser=HOST_USER-7
ip-10-110-230-197 database link *> set linkData.stagingRepository=ASE_INSTANCE-6
ip-10-110-230-197 database link *> set
linkData.syncParameters.type=ASENewBackupSyncParameters
ip-10-110-230-197 database link *> commit
    `ASE_DB_CONTAINER-1
    Dispatched job JOB-39
    DB_LINK job started for "Untitled/fuji".
    DB_LINK job for "Untitled/fuji" completed successfully.
```

8.2.4.6.12 Update Existing Database Users

Convert an existing database to use vault credentials for the existing database user.

```
ip-10-110-230-197 > sourceconfig
ip-10-110-230-197 sourceconfig > select MyOraDB
ip-10-110-230-197 sourceconfig "MyOraDB" > update
ip-10-110-230-197 sourceconfig "MyOraDB" *> set
credentials.type=CyberarkVaultCredential
ip-10-110-230-197 sourceconfig "MyOraDB" *> set credentials.vault=MyVault
ip-10-110-230-197 sourceconfig "MyOraDB" *> set credentials.queryString="safe-
test;folder=root;object=UnixSSH-delphix_db"
ip-10-110-230-197 sourceconfig "MyOraDB" *> set db_user="Vault-User"
ip-10-110-230-197 sourceconfig "MyOraDB" *> commit
```

The set db_user="Vault-User" is an optional step. If the db_user field is not changed, then it will continue to hold the old value. This value may no longer be correct, or the change to Vault credentials may represent an increase in the customer's security stance, and they may not want their Delphix Admins to know the username.

8.2.4.6.13 Password Vault Cache

If it becomes necessary to reduce the number of vault access requests by the engine, you can enable caching in the vault configuration. To do this, enable the feature flag PASSWORD_VAULT_CACHE and set each vault's `cacheEnabled` to true, along with the desired `cacheExpiryTimeMinutes` value.

```
ip-10-110-230-197 service passwordVault create *> ls
Properties
    type: AzureVault
    name: (required)
    azureAuthentication:
        type: AzureCertificateAuthentication
        clientCertificate:
            type: KeystoreClientCertificate
            token: (required)
        clientId: (required)
        tenantId: (required)
    cacheEnabled: false
    cacheExpiryTimeMinutes: (unset)
```

```
ip-10-110-230-197 service passwordVault> select DemoVault
ip-10-110-230-197 service passwordVault 'DemoVault'>update
ip-10-110-230-197 service passwordVault 'DemoVault'update *> set cacheEnabled=true
ip-10-110-230-197 service passwordVault 'DemoVault'update *> set
cacheExpiryTimeMinutes=5
ip-10-110-230-197 service passwordVault 'DemoVault'update *> commit
ip-10-110-230-197 service passwordVault 'DemoVault'>
```

8.2.4.7 Setting up Vault via API

The vault API allows users to add, modify, delete, and list vault configurations and retrieving user credentials on a Delphix Engine.

Endpoint - *https://<Delphix Engine IP>/resources/json/delphix/service/passwordVault*

8.2.4.7.1 Sample API Request

```
{
    "type": "CyberArkPasswordVault",
    "name": "DemoVault",
    "host": "services-uscentral.skytap.com",
    "port": 17993,
    "applicationId": "Delphix",
    "clientCertificate": {
```

```
        "type": "PemClientCertificate",
        "privateKey": "-----BEGIN PRIVATE KEY-----<>-----END PRIVATE KEY-----",
        "clientCertificateChain": {
            "type": "PemCertificateChain",
            "chain": [
                {
                    "type": "PemCertificate",
                    "contents": "-----BEGIN CERTIFICATE-----<>-----END
CERTIFICATE-----"
                }
            ]
        }
    }
}
```

8.2.4.7.2 Deleting an existing vault configuration

```
{
    "type": "CyberArkPasswordVault",
    "name": "DemoVault",
    "host": "services-uscentral.skytap.com",
    "port": 17993,
    "applicationId": "Delphix",
    "clientCertificate": {
        "type": "PemClientCertificate",
        "privateKey": "-----BEGIN PRIVATE KEY-----<>-----END PRIVATE KEY-----",
        "clientCertificateChain": {
            "type": "PemCertificateChain",
            "chain": [
                {
                    "type": "PemCertificate",
                    "contents": "-----BEGIN CERTIFICATE-----<>-----END
CERTIFICATE-----"
                }
            ]
        }
    }
}
```

8.2.4.7.3 Adding/Modifying host users

Add an environment with user credentials from CyberArk vault. When adding a host/database user with a vault credential, the name field would be a user identifier and not the actual username. In case this field is empty, a unique identifier is generated with a hash of vault credentials.

```
{
"type": "HostEnvironmentCreateParameters",
```

```
"primaryUser": {
"type": "EnvironmentUser",
"credential": {
"type": "VaultCredential",
"vault": "CYBERARK_PASSWORD_VAULT-1",
"vaultCredentialId": "safe=test;folder=root;object=UnixSSH-sybase"
}
},
"hostEnvironment": {
"type": "UnixHostEnvironment",
"name": "bbh-env"
},
"hostParameters": {
"type": "UnixHostCreateParameters",
"host": {
"type": "UnixHost",
"address": "bbdhcp-vault-demo.dlpxdc.co",
"toolkitPath": "/work"
}
}
}
```

8.2.4.7.4 Adding/Modifying database users

Add dSource using database credentials from CyberArk vault.

The following a sample API link request for MSSQL Domain User.

```
{

    "type": "LinkParameters",

    "name": "ReportServer",

    "group": "GROUP-1",

    "linkData": {

        "type": "MSSqlLinkData",

        "config": "MSSQL_SINGLE_CONFIG-5",

        "sharedBackupLocations": [],

        "encryptionKey": "",

        "sourceHostUser": "HOST_USER-3",

        "mssqlUser":{

         "password": {
```

```
          "type": "VaultCredential",
          "vault": "CYBERARK_PASSWORD_VAULT-2",
          "vaultCredentialId": "safe=test;folder=root;object=Database-MSSql-addtully"
          },
          "type": "MSSqlDomainUser"
         },
         "pptRepository": "MSSQL_INSTANCE-4",
         "pptHostUser": "HOST_USER-3",
         "ingestionStrategy": {
          "validatedSyncMode": "TRANSACTION_LOG",
          "type": "ExternalBackupIngestionStrategy"
         },
         "sourcingPolicy": {
          "logsyncEnabled": false,
          "type": "SourcingPolicy"
         },
         "syncParameters": {
          "compressionEnabled": false,
          "backupPolicy": "PRIMARY",
             "type": "MSSqlNewCopyOnlyFullBackupSyncParameters"
         }
     }
 }
```


8.2.5 Certificate management

The Delphix Dynamic Data Platform uses SSL certificates to secure a number of different communications. Delphix provides users with the ability to manage their own certificates for HTTPS, DSP (Delphix Session

Protocol) and stunnel server connections to and from the Delphix Engine. Users are able to see multiple certificates, their attributes, and deployed status, they can also:

- Replace certificates
- Delete certificates
- Create CSRs (certificate signing requests)

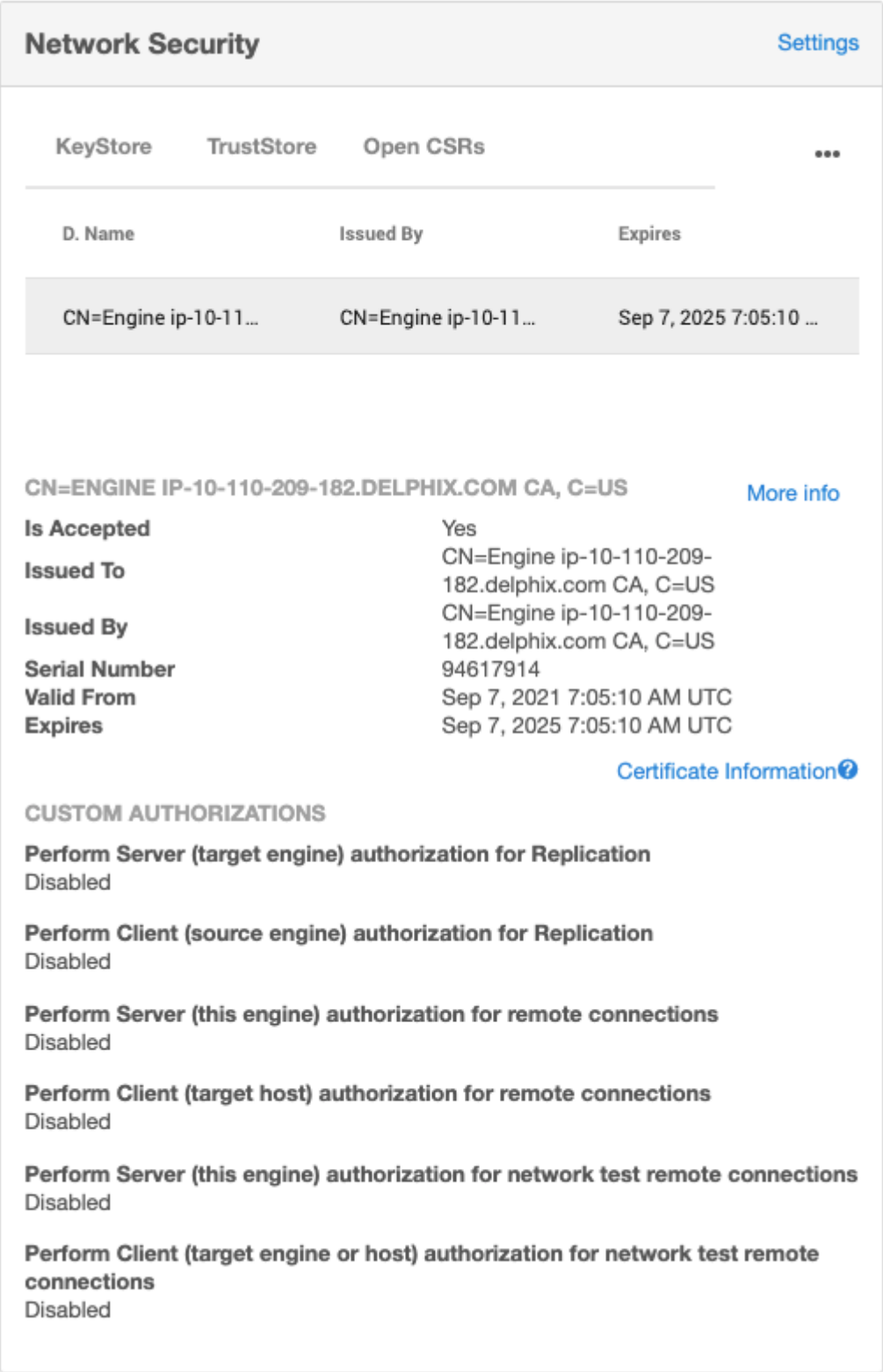
8.2.5.1 Configuring network security settings

To enable and modify the extent of security settings for HTTPS and DSP (Delphix Session Protocol), click on the “Settings” button in the top right of the “Network Security” panel. This will open a new wizard where you can modify the settings.

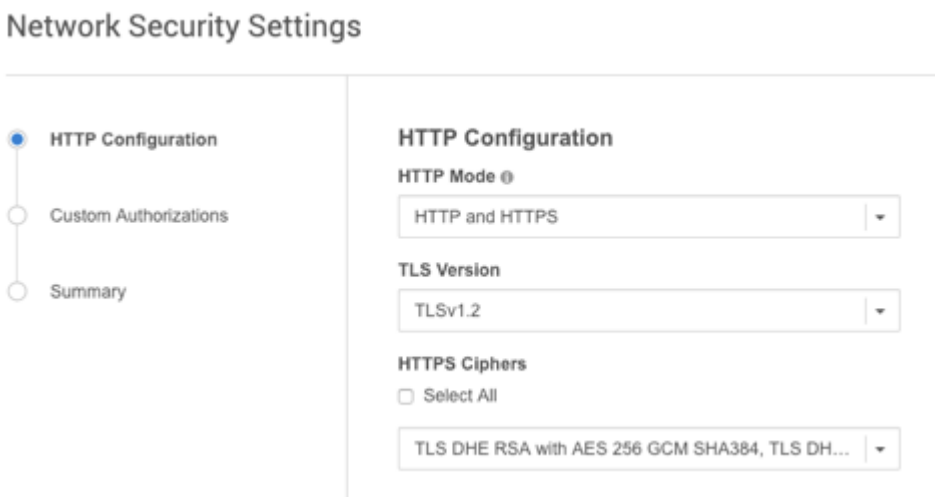
8.2.5.1.1 Configuring settings for HTTPS and DSP.

The following procedure guides you through the process of enabling or modifying security settings for HTTPS and DSP.

1. Connect to the Delphix Engine `http://<Delphix Engine>/login/index.html#serverSetup`
2. Click on the **Settings** link in the top right of the **Network Security** panel. This will open a new wizard where you can modify the settings.



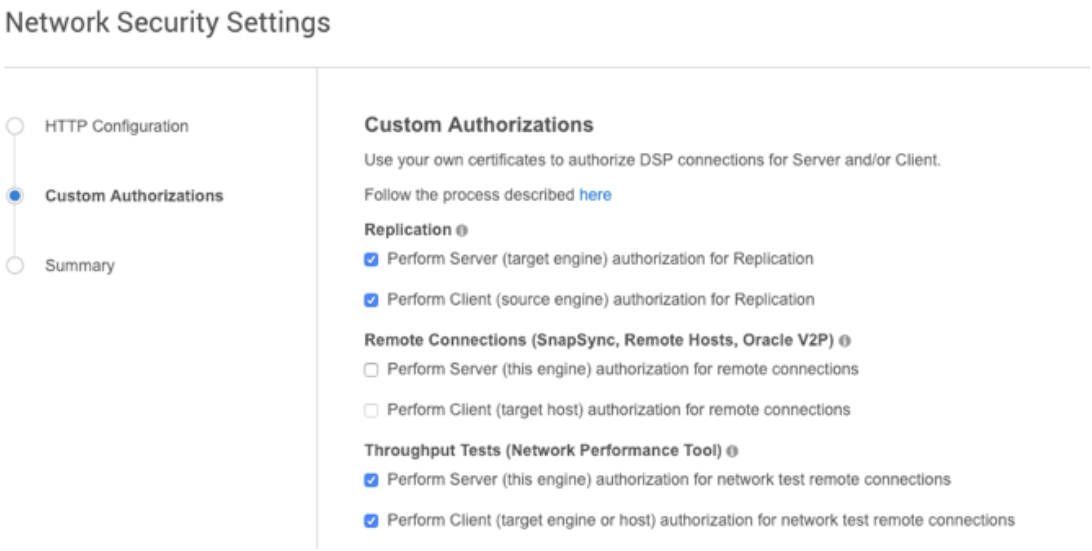
3. In the **Network Settings** wizard, you can modify the **HTTP mode**, **TLS Version**, and **TLS Cipher Suites** to be used.



4. Click **Next**.
5. By default, self-signed certificates are used for server authentication and username/password login acts as client authorization. You can enable the use of custom certificates and modify the extent of how they are used for the following features:
   - Replication (see page 2367)
   - Remote host connections (see page 969) (SnapSync, Remote hosts for Target or Server environments, and Oracle V2P)
   - Throughput tests (see page 971) using the Network Performance Tool and DSP

     Info: For all three features, Server authentication must be enabled for Client authentication to take effect.

     Note: Any configuration change requires a stack restart to take effect. Submitting the Network Security Settings in the GUI will automatically trigger a stack restart. All jobs will be stopped, but VDBs will continue to run.



6. Once the HTTP and DSP configurations have been set as desired, click **Next**.

7. You will be presented with a Summary tab. Clicking **Submit** to accept your changes will trigger a stack restart as this is necessary for the configuration changes to take effect. Note: all jobs will be stopped, but VDBs will continue to run.

8.2.5.1.2 Certificate management and remote connections

8.2.5.1.2.1 Overview

The server is the Delphix engine and the client is the remote host. This can be used for SnapSync, Oracle V2P (Virtual to Physical), and remote host connections. Once either of these options is enabled, the steps for adding certificate must be done for all environments in the engine.

8.2.5.1.2.2 Enabling server authentication

To enable server authentication, follow the below steps:

1. Replace the desired certificate for DSP (Delphix Session Protocol) in the engine KeyStore. For more details, refer to KeyStore Settings (see page 987)
2. Create a JKS or PKCS#12 keystore on the remote host with the full CA chain of the replaced certificate. Make sure the created keystore has permissions such that it is readable by all environment users configured in Delphix, and enter the keystore details into the host's truststore configuration on the engine. For more details, refer to Host DSP Configuration (see page 984)
3. Select **Perform server (this engine) authorization for remote connections**.

> Altering the authentication settings will require DSP keystore and truststore parameters to be configured for all existing environments, if not the refreshing of existing host environments will fail.

8.2.5.1.2.3 Enabling client authentication

1. DSP connector (for both Windows and Unix hosts)

To enable client authentication using DSP connector, first enable server authentication (refer to the above steps), then follow the below steps:

1. Create a JKS or PKCS#12 keystore on the remote host with the desired key pair. Make sure the created keystore has permissions such that it is readable by all environment users configured in Delphix, then enter the keystore details into the host's keystore configuration on the engine. For more details, refer to Host DSP Configuration (see page 984)

2. Add the full CA chain of the remote host's key pair to the TrustStore on the engine. For more details, refer to TrustStore Settings (see page 989)
3. Select **Perform Client (the target host) authorization for remote connections**.
4. Once the configurations have been set as desired, you will be presented with a summary page. Clicking **Submit** will trigger a stack restart, which is necessary for the configuration changes to take effect. Note: all jobs will be stopped, but VDBs will continue to run.

2. Connector installer connector (specific for Windows hosts)

There are two ways to generate self signed certificates :

a) By Installing the Delphix Connector (see page 2105), which will by default create certificates.

b) By using Self-signed Certificates (see page 2100)

To enable client authentication using connector installer, you must perform the below steps for all Windows hosts, which are being added to the Delphix Engine:

1. Execute the below command to generate the PEM file for the Delphix Connector (provided or self-signed (see page 2100))Java KeyStore file. Also, input the store password from the `DelphixConnector.properties` when prompted.

```
keytool -exportcert -alias DelphixConnector-
{UUID_From_DelphixConnector.properties} -keystore "{Installation_Dir}
\connector\DelphixConnector.jks" -rfc -file {Custom_PEM_File_Name}
```

2. Copy the PEM's file content and paste it while adding the certificate into the Delphix Engine.
3. Add the certificate to Delphix engine using the **sysadmin** login and select **Network Security**.
4. Select **Add Certificate** and upload the certificate.
5. Once the certificate is added, enable `validateWindowsConnectorCertificate` from the Delphix engine CLI. This will restart the Delphix engine.

8.2.5.1.3 Certificate management and replication

In Replication, the Server is the Target engine and the Client is the Source engine.

8.2.5.1.3.1 Enabling server authentication

To enable Server Authentication, do the following:

1. Replace the desired certificate for DSP (Delphix Session Protocol) in the Target engine KeyStore. For more details, refer to KeyStore Settings (see page 987)
2. Add the full CA chain of the replaced certificate from the Target engine to the TrustStore on the Source engine. The CA chain must match on both engines. For more details, refer to TrustStore Settings (see page 989)

3. Select the option **Perform Server (target engine) authorization for Replication** for both Target and Source engines.

8.2.5.1.3.2 Enabling client authentication

To enable Client Authentication, enable Server Authentication (refer to above steps), then do the following:

1. Replace the desired certificate for DSP in the Source engine KeyStore. For more details, refer to KeyStore Settings (see page 987)
2. Add the full CA chain of the replaced certificate from the Source engine to the TrustStore on the Target engine. The CA chain must match on both engines. For more details, refer to TrustStore Settings (see page 989)
3. Select the option Perform Client (source engine) authorization for Replication for both Target and Source engines.
4. Once the configurations have been set as desired, you will be presented with a summary page. Clicking **Submit** will trigger a stack restart as that is necessary for the configuration changes to take effect. Note: all jobs will be stopped, but VDBs will continue to run.

8.2.5.1.4 Certificate management for throughput tests

8.2.5.1.4.1 Enabling server authentication

To enable Server Authentication, do the following:

1. If using the engine to engine tests, follow the steps for Replication Server Authentication (see page 970)
2. If using the engine to host tests, follow the steps for Remote Connections Server Authentication (see page 969)
3. Select the option **Perform Server (this engine) authorization for remote connections**.

8.2.5.1.4.2 Enabling client authentication

To enable Client Authentication, enable Server Authentication (refer to above steps), then do the following:

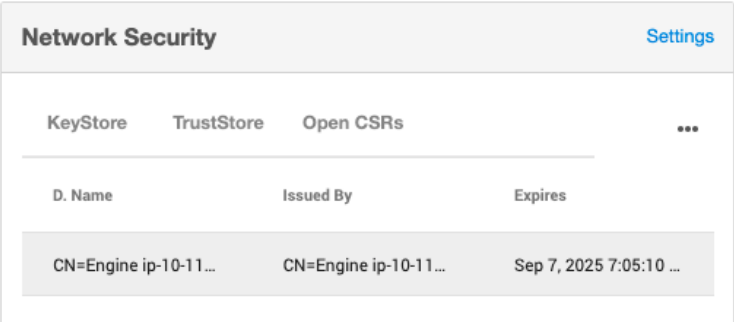
1. If using the engine to engine tests, follow the steps for Replication Client Authentication (see page 970)
2. If using the engine to host tests, follow the steps for Remote Connections Client Authentication (see page 969)
3. Select the option **Perform Client (target engine or host) authorization for remote connections.**
4. Once the configurations have been set as desired, you will be presented with a summary page. Clicking **Submit** will trigger a stack restart as that is necessary for the configuration changes to take effect. Note: all jobs will be stopped, but VDBs will continue to run.

> Delphix Engine generates alerts when certificates in the Keystore or truststore are expired or about to expire:
>
> - A warning level alert is generated if certificates are expiring in 60 days.
> - Critical level alerts are generated if certificates expire in 14 days or have already expired.

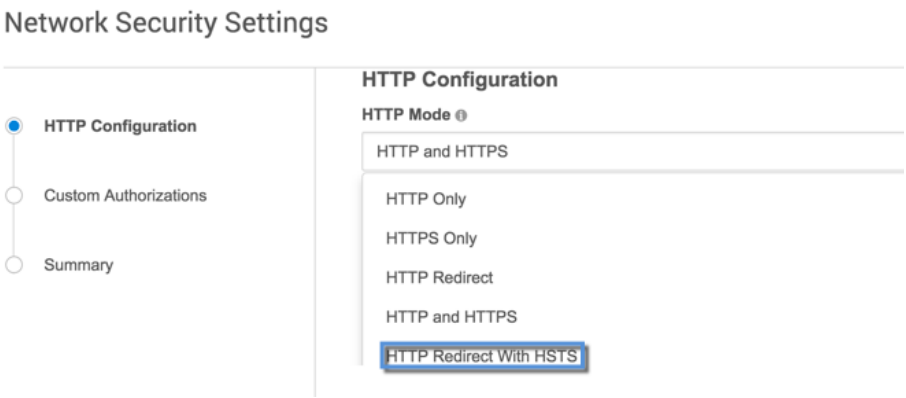
8.2.5.1.5 Configuring HTTP settings for the Delphix Engine

Use the following steps to configure HTTP settings for the Delphix Continuous Data Engine.

1. Login to the Continuous Data Setup UI as a sysadmin.
2. From the Dashboard, select **Settings** under **Network Security**. This will open a new wizard where you can modify the settings.



3. In the **Network Security Settings** screen, you can modify the **HTTP mode**, **TLS Version**, and **HTTPS Ciphers** to be used.
4. In **HTTP mode**, select one of the following:
   a. **HTTP Only** – accepts only HTTP connections.
   b. **HTTPS Only** – accepts only HTTPS connections.
   c. **HTTP Redirect** – redirect all requests made over HTTP to HTTPS.
   d. **HTTP and HTTPS** – accepts both HTTP and HTTPS connections.
   e. **HTTP Redirect with HSTS** – redirect all requests made over HTTP to HTTPS and add the “Strict-Transport-Security” header to all responses.
      i. HSTS is not required. By default, the Continuous Data Engine is configured to HTTP and HTTPS. This can be left as is.
      ii. If you set up HSTS and want to revert it, follow the same steps to this point and select your previous option.



5. In **TLS Version**, select the required TLS versions.
6. In **HTTPS Ciphers**, select the required option from the drop-down list. Click the check box next to **Select All** to choose all options.
7. Click **Next** and then click **Submit**, allowing your engine to restart for the changes to take effect.
8. When the Continuous Data Engine homepage is loaded after the restart, all the HTTP requests will be redirected to the selected HTTP mode after the first load.

The above steps could fail if:

1. The certificates are not trusted – a “not secure” warning on the search bar of the browser. Certificate setup should be completed before setting up HSTS headers in order for them to be useful. Otherwise, the redirect will still happen over 302 status.
2. Your site is not prepared for HTTPS setup – for example, each of the subdomains accessed should be supporting HTTPS; parts of the application might be inaccessible otherwise.
3. Every first request whenever the cache is cleared would redirect over 302. Any further requests after the first would redirect over 307

8.2.5.1.6 Regenerating self-signed end-entity and CA certificates

In many environments, the replacement of HTTPS and/or DSP may be unnecessary.

DSP certificate is only relevant if Custom Authorizations have been configured in Network Security settings, as discussed in the Configuring Network Security Settings[285] page. If these checkboxes are not applied, this means the DSP certificate is not being used.

285 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/83199469

HTTPS certificate replacement is only necessary if HTTPS connections are used for web browser access.

The following process will leverage Java `keytool` utility. This is commonly available in most Java JDK installations, including those installed in the Delphix Toolkit for Unix, Linux, and Windows Environments under **<toolkit directory> /*host/java/jdk/bin/**. In the following example, `/work` is the toolkit directory. The subdirectory naming conventions from 5.3.x and 6.0.x are illustrated as:

```
$ find /work -name keytool
/work/Delphix_COMMON_f126df603015_33e2f61712c3_2_host/java/jdk/jre/bin/keytool
/work/Delphix_COMMON_f126df603015_33e2f61712c3_2_host/java/jdk/bin/keytool
/work/Delphix_COMMON_564d56b0_26ad_e6ac_f782_d15213207664_oracle_host/java/jdk/bin/
keytool
/work/Delphix_COMMON_564d56b0_26ad_e6ac_f782_d15213207664_oracle_host/java/jdk/jre/
bin/keytool
```

By the end of this process, a PKCS#12 KeyStore file is generated containing the CA certificate, DSP, stunnel server, HTTPS certificate. This file will be used for upload twice in the System Setup interface.

Other notes:

- The recommended KeyStore password, **changeit**, is used.
- For `<domain>`, replace this string with the Engine FQDN in every command. This is used as the CN (Common Name). For instance, `-dname 'CN=Engine <domain>  ca,="" c="">>"=` would be replaced with `-dname 'CN=Engine  example.delphix.com  CA, C=US'`.
- The certificate aliases to be used are `tomcat` for HTTPS and `dsp` for DSP.
- The existing Delphix CA certificate in the truststore cannot be removed. Faults related to this certificate should be ignored.

8.2.5.1.6.1 Linux Version

1. Generate a new Delphix CA Certificate.

```
export PASSWORD_ENV='changeit'
keytool -genkeypair -noprompt  -alias delphixca  -keyalg RSA -keysize 2048
 -validity 397 -ext 1.3.6.1.5.5.7.3.1 -ext bc=ca:true -ext ku=kCS,cRLS -sigalg
SHA256withRSA -storepass:env PASSWORD_ENV  -storetype pkcs12 -startdate -10000M
-dname 'CN=Engine <domain> CA, C=US' -keypass:env PASSWORD_ENV -storetype
pkcs12 -keystore keystore
```

2. Generate the HTTPS/TLS certificate

```
keytool -genkeypair -alias tomcat -keyalg RSA -keysize 2048 -validity 397 -ext
1.3.6.1.5.5.7.3.1 -ext san=dns:<domain> -ext bc=ca:false  -sigalg SHA256withRSA
-storetype pkcs12 -storepass:env PASSWORD_ENV -startdate -10080M -dname
'CN=<domain>, C=US' -keypass:env PASSWORD_ENV -keystore keystore
```

3. Generate the DSP certificate

```
keytool -genkeypair -alias dsp -keyalg RSA -keysize 2048 -validity 397 -ext 1.3.6
.1.5.5.7.3.1 -ext san=dns:<domain> -ext bc=ca:false  -sigalg SHA256withRSA
-storetype pkcs12 -storepass:env PASSWORD_ENV -startdate -10080M -dname
'CN=<domain>, C=US' -keypass:env PASSWORD_ENV -keystore keystore
```

4. Generate the STUNNEL-SERVER certificate

```
keytool -genkeypair -alias stunnel-server -keyalg RSA -keysize 2048 -validity
397 -ext 1.3.6.1.5.5.7.3.1 -ext san=dns:<domain> -ext bc=ca:false  -sigalg
SHA256withRSA -storetype pkcs12 -storepass:env PASSWORD_ENV -startdate -10080M
-dname 'CN=<domain>, C=US' -keypass:env PASSWORD_ENV -keystore keystore
```

5. Sign the HTTP/TLS certificate

```
keytool -certreq -alias tomcat -keyalg RSA -sigalg SHA256withRSA -storetype
pkcs12 -keypass:env PASSWORD_ENV -storepass:env PASSWORD_ENV -keystore keystore
-file tomcat.csr
keytool -gencert -alias delphixca -ext 'san=dns:<domain>' -validity 397 -sigalg
SHA256withRSA -storetype pkcs12 -storepass:env PASSWORD_ENV -keystore keystore
-startdate -10080M -infile tomcat.csr -outfile tomcat.p12
keytool -importcert -alias tomcat -storetype pkcs12 -storepass:env PASSWORD_ENV
-keystore keystore -file tomcat.p12
```

6. Sign the DSP Certificate

```
keytool -certreq -alias dsp -keyalg RSA -sigalg SHA256withRSA -storetype pkcs12
-keypass:env PASSWORD_ENV -storepass:env PASSWORD_ENV -keystore keystore -file
dsp.csr
keytool -gencert -alias delphixca -ext 'san=dns:<domain>' -validity 397 -sigalg
SHA256withRSA -storetype pkcs12 -storepass:env PASSWORD_ENV -keystore keystore
-startdate -10080M -infile dsp.csr -outfile dsp.p12
keytool -importcert -alias dsp -storetype pkcs12 -storepass:env PASSWORD_ENV
-keystore keystore -file dsp.p12
```

8.2.5.1.6.2 Windows version

1. Generate a new Delphix CA Certificate.

```
$ENV:PASSWORD_ENV='changeit'
.\keytool -genkeypair -noprompt  -alias delphixca -keyalg RSA -keysize 2048
 -validity 397  -ext 1.3.6.1.5.5.7.3.1 -ext bc=ca:true -ext ku=kCS,cRLS -sigalg
SHA256withRSA -storepass:env PASSWORD_ENV  -storetype pkcs12 -startdate -10000M
```

```
-dname 'CN=Engine <domain> CA, C=US' -keypass:env PASSWORD_ENV -storetype
pkcs12 -keystore keystore
```

2. Generate the HTTPS/TLS certificate

```
.\keytool -genkeypair -alias tomcat -keyalg RSA -keysize 2048 -validity 397
 -ext 1.3.6.1.5.5.7.3.1 -ext san=dns:<domain> -ext bc=ca:false  -sigalg
SHA256withRSA -storetype pkcs12 -storepass:env PASSWORD_ENV -startdate -10080M
-dname 'CN=<domain>, C=US' -keypass:env PASSWORD_ENV -keystore keystore
```

3. Generate the DSP certificate

```
.\keytool -genkeypair -alias dsp -keyalg RSA -keysize 2048 -validity 397 -ext
1.3.6.1.5.5.7.3.1 -ext san=dns:<domain> -ext bc=ca:false  -sigalg SHA256withRSA
-storetype pkcs12 -storepass:env PASSWORD_ENV -startdate -10080M -dname
'CN=<domain>, C=US' -keypass:env PASSWORD_ENV -keystore keystore
```

4. Sign the HTTP/TLS certificate

```
.\keytool -certreq -alias tomcat -keyalg RSA -sigalg SHA256withRSA -storetype
pkcs12 -keypass:env PASSWORD_ENV -storepass:env PASSWORD_ENV -keystore keystore
-file tomcat.csr
.\keytool -gencert -alias delphixca -ext 'san=dns:<domain>' -validity 397
 -sigalg SHA256withRSA -storetype pkcs12 -storepass:env PASSWORD_ENV -keystore
keystore -startdate -10080M -infile tomcat.csr -outfile tomcat.p12
.\keytool -importcert -alias tomcat -storetype pkcs12 -storepass:env
PASSWORD_ENV -keystore keystore -file tomcat.p12
```

5. Sign the DSP Certificate

```
.\keytool -certreq -alias dsp -keyalg RSA -sigalg SHA256withRSA -storetype
pkcs12 -keypass:env PASSWORD_ENV -storepass:env PASSWORD_ENV -keystore keystore
-file dsp.csr
.\keytool -gencert -alias delphixca -ext 'san=dns:<domain>' -validity 397
 -sigalg SHA256withRSA -storetype pkcs12 -storepass:env PASSWORD_ENV -keystore
keystore -startdate -10080M -infile dsp.csr -outfile dsp.p12
.\keytool -importcert -alias dsp -storetype pkcs12 -storepass:env PASSWORD_ENV
-keystore keystore -file dsp.p12
```

At this point, the certificates can be installed by following the Customer Provided Key Pair method described in the User provided key pair configuration[286] page.

The same KeyStore generated will be uploaded twice, once using alias 'dsp' and once using alias 'tomcat', to replace DSP and HTTPS certificates, respectively.

286 https://cd.delphix.com/docs/latest/customer-provided-key-pair-configuration

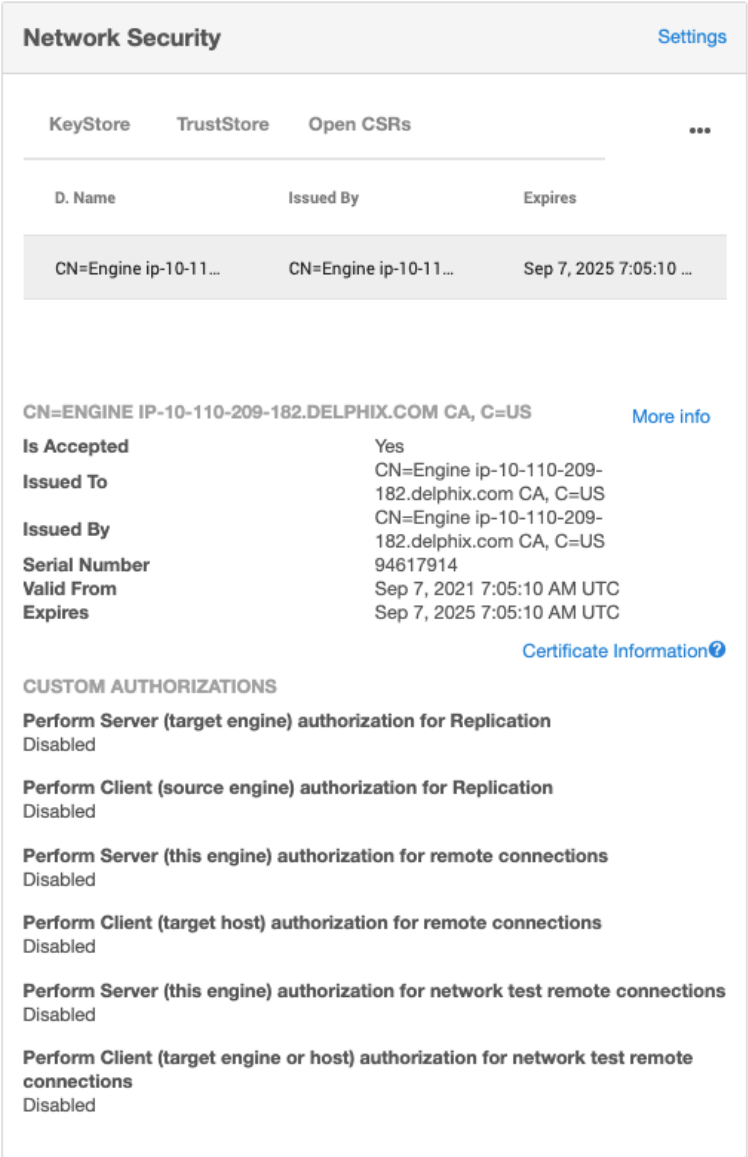
> If this error occurs, **Failed to read file with error "Invalid KeyStore format"**, ensure the **Upload certificate from a PKCS#12 KeyStore** radio button is selected.

8.2.5.1.6.3 Windows keytool distinctions

- Given the Delphix Connector installation directory **C:\Program Files\Delphix\DelphixConnector**, the keytool.exe executable can be found at **C:\Program Files\Delphix\DelphixConnector\jre\bin\keytool.exe**.
- PowerShell set environment variable with: `$ENV:PASSWORD_ENV='changeit'`
- Similar to the comment above with path assumption, on Windows, change ./keytool to .\keytool.exe if located in the jre\bin subdirectory.

8.2.5.2 Configuring network settings

This topic provides instructions on replacing the HTTPS (HTTP Secure) certificate, the DSP (Delphix Session Protocol) certificate, and the stunnel-server certificate used by the Delphix Continuous Data Engine. The KeyStore tab within the Network Security panel displays the keys that the Delphix Continuous Data Engine uses for its identity in HTTPS, DSP, and stunnel-server communication. For a selected service, the tab displays the complete certificate chain, starting with the end-entity certificate and ending with the root CA.



There are two methods of replacing a certificate. The key difference between the two is whether Delphix or the user is providing the key pair (public and private key).

For more information on how to configure these see:

- Delphix provided key pair configuration (see page 980)
- User-provided key pair configuration (see page 978)

8.2.5.2.1 Customer provided key pair configuration

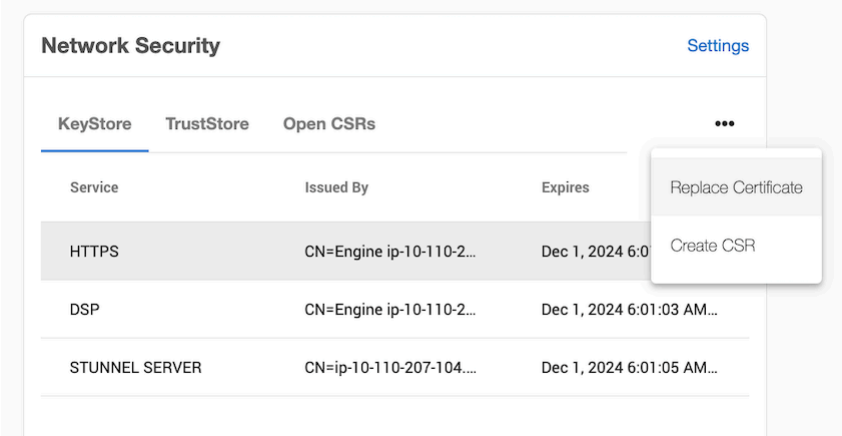
This section describes the steps to take if you are replacing the HTTPS or DSP (Delphix Session Protocol) or stunnel-server with your own key pair and certificate. To start, you need to add the key pair and full certificate chain as an entry in a file in JKS or PKCS #12 format.

If replacing the key pair for the DSP application, note that when the DSP Server or Client authenticates each incoming DSP connection (if enabled), it will validate that each certificate of the incoming connection's identity chain has a "Valid From (or Not Before)" date that is after its own time.

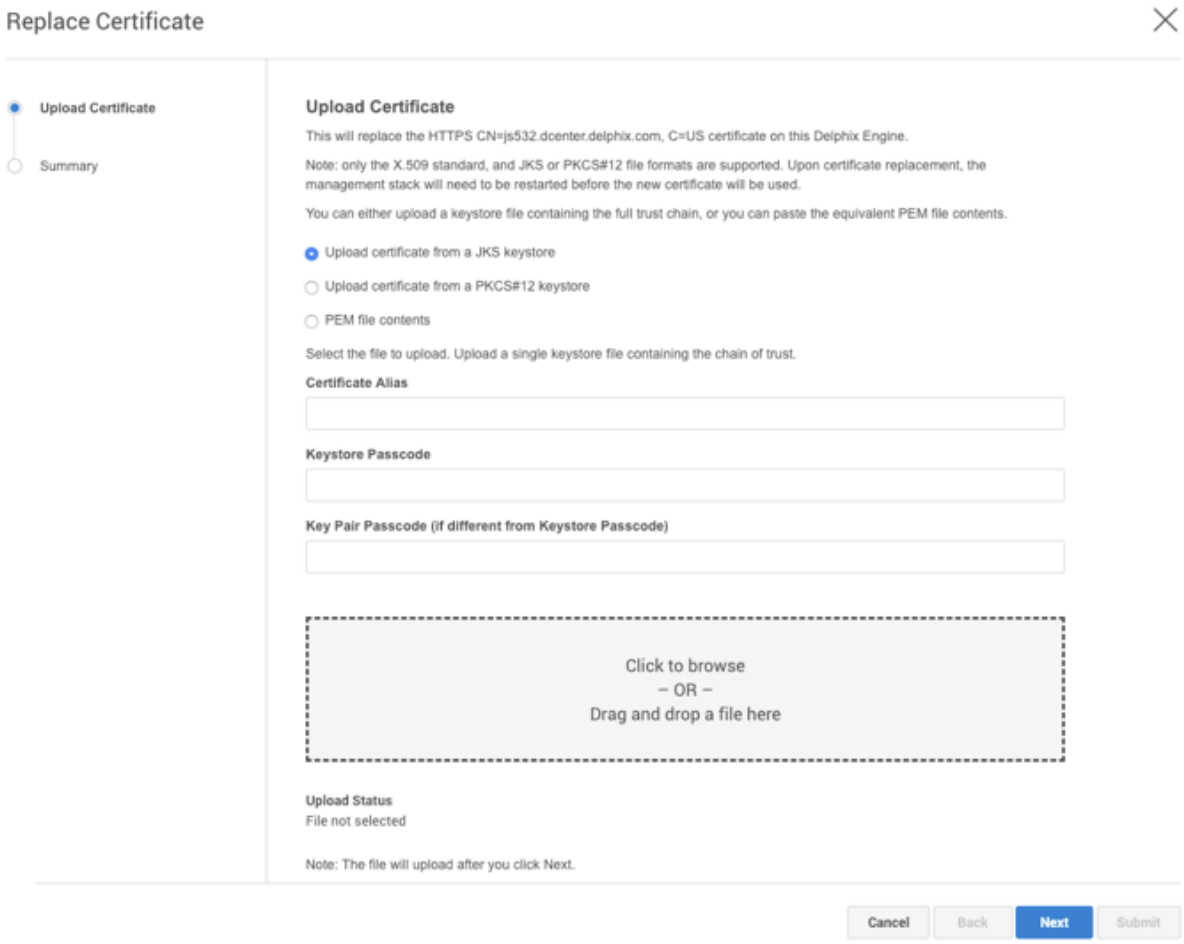
Thus, if your Delphix Engine or host environments are running off of an incorrect (slow) time configuration, then your DSP connections will not work until the offending engine or host's time advances past all incoming certificate's "Valid From (or Not Before)" time.

If correcting the Delphix Engine's or host environment's time configuration may cause issues, then you can workaround this issue by creating and using a certificate with a "Valid From (or Not Before)" date which is before your slowest Engine or host.

1. From the **Network Security** panel, select the **Actions** (**...**) menu, and select **Replace Certificate**.



2. In the **Replace Certificate** window, select **Upload certificate as files**.



a. The **Certificate Alias** field is where the key pair and certificate is saved in your JKS or PKCS #12 store.
b. The **Keystore Passcode** field is the keystore's password.
c. The **Key Pair Passcode** field is the password for the given alias' key. If not set, it uses the keystore's password.

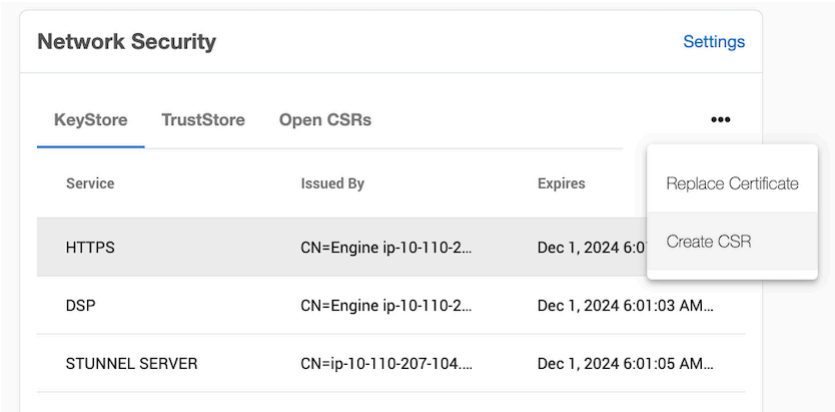
3. Click **Next** to view a summary. Then click **Submit**.

8.2.5.2.2 Delphix provided key pair configuration

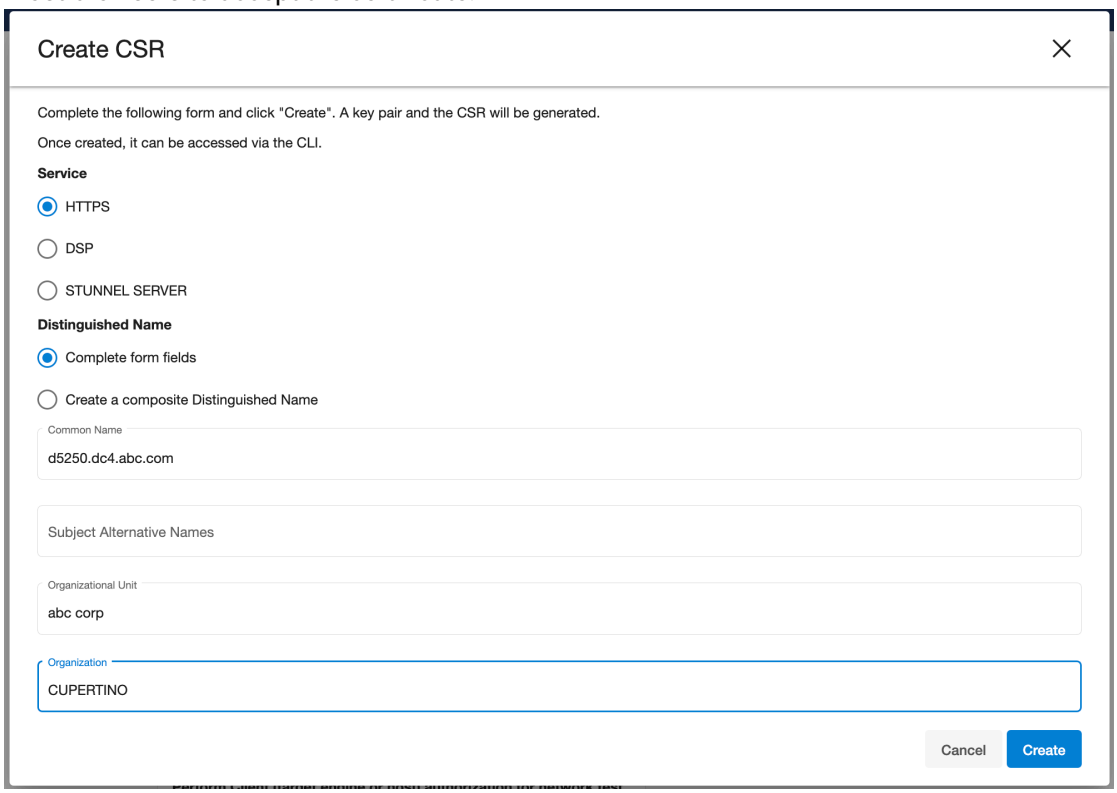
8.2.5.2.2.1 Creating CSR

Perform the following steps to provide an HTTPS or DSP (Delphix Session Protocol) or stunnel-server certificate chain for a key pair created by the Delphix Continuous Data Engine.

1. Connect to the Delphix Engine HTTP://< your engine>/login/index/html#serverSetup.
2. From the **Network Security** panel, select the **Actions** (**...**) menu, and select **Create CSR**.

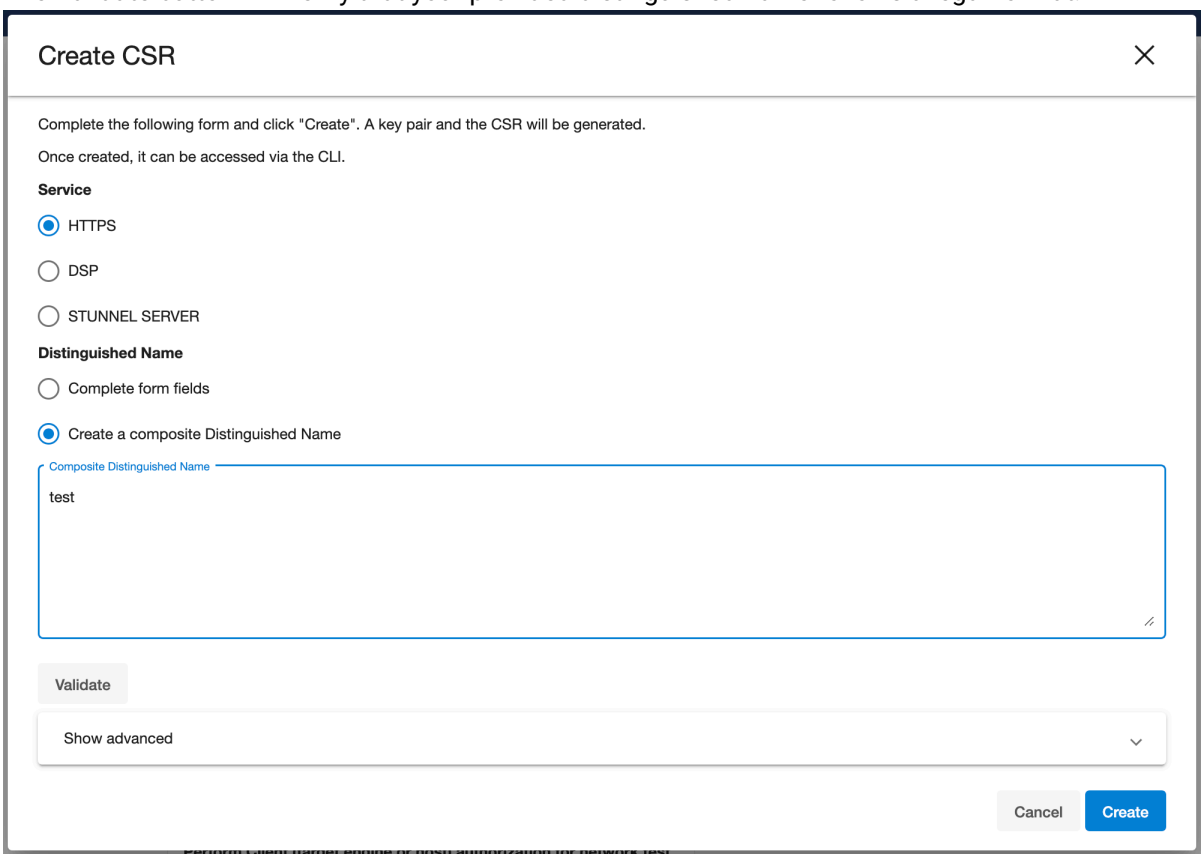


3. In the **Create CSR window** provide options for your CSR (certificate signing request). Fill up the following fields:
   a. The **Common Name** (CN) is required. If the CN is a valid hostname, then the **Subject Alternative Name** (SAN) can be left blank, otherwise a SAN should be provided. When the CN is a wildcard Common Name (e.g. *.abc.com), the SAN must be populated with every possible domain name covered by the certificate as a comma delimited list. A valid SAN is required for most browsers to accept the certificate.



   b. Select the service you want to create a CSR for (HTTPS or DSP).
   c. Select how to enter the distinguished name (using the provided fields, or as a composite string).

4. The Validate button will verify that your provided distinguished name follows a legal format.



5. [Optional] Select the **Show advanced link** to provide extra options.
   a. **Force Replace**: By default, this is false and means Delphix will not replace the active key pair and certificate with the newly generated key pair and self-signed certificate. If you want to replace the active key pair right away before the signed certificate has been created this can be set to true.
   b. **Keypair Algorithm**: Choose whether the created keypair will use the RSA or ECDSA algorithm. Once an algorithm has been chosen the user can customize the key size and the signature algorithm used.
      i. **Key size:** The valid sizes for RSA are between 2048 and 4096 inclusive. The valid sizes for ECDSA are between 256 and 571 inclusive.
      ii. **Signature algorithm:** The available signature algorithms for RSA are "`SHA256withRSA`", "`SHA384withRSA`", "`SHA512withRSA`". The available signature algorithms for ECDSA are "`SHA256withECDSA`", "`SHA384withECDSA`", "`SHA512withECDSA`".
6. Click **Create**. Once submitted, the CSR is created, and will show up in the **Open CSRs** tab. You can select the CSR and click on View PEM to obtain the CSR in a PEM format.

If creating a CSR for the DSP application, note that when the DSP Server or Client authenticates each incoming DSP connection (if enabled), it will validate that each certificate of the incoming connection's identity chain has a "Valid From (or Not Before)" date that is after its own time.

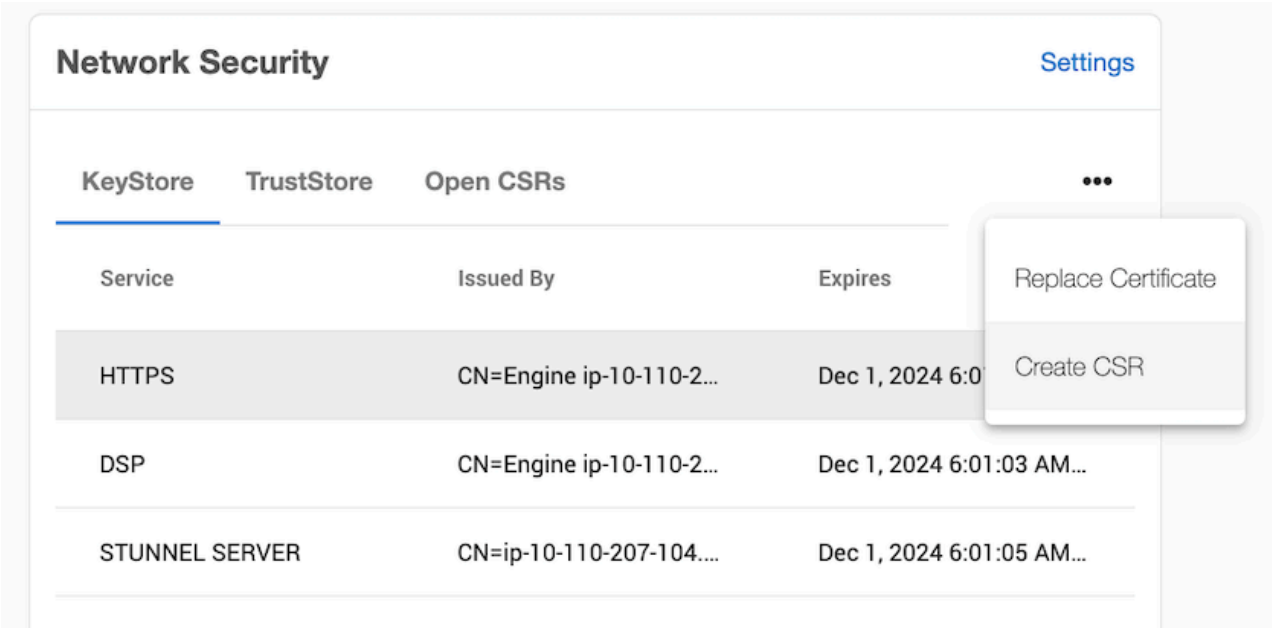
Thus, if your Delphix Engine or host environments are running off of an incorrect (slow) time configuration, then your DSP connections will not work until the offending engine or host's time advances past all incoming certificate's "Valid From (or Not Before)" time.

If correcting the Delphix Engine's or host environment's time configuration may cause issues, then you can workaround this issue by signing the CSR with a "Valid From (or Not Before)" date which is before your slowest Engine or host.

8.2.5.2.2.2 Replacing a certificate

After the CSR has been signed and turned into an X.509 Certificate, you can replace the certificate using the Replace Certificate option.

1. From the **Network Security** panel select the **Actions** (**...**) menu and select **Replace Certificate**.



2. In the Replace Certificate window, select PEM file contents. You can then paste the PEM response in the text box.

> The PEM contents must contain a list of the entire trust chain from the newly generated end-entity certificate to the root CA. You can do this by just pasting each certificate back to back in the text box as long as they are separated by the begin and end cert tags. The order that the PEM certificates are added to the list does not matter

3. Click **Next**.

4. The **Summary** tab describes your selections. Click **Submit**.

8.2.5.2.3 Host DSP configuration

In Delphix Engine to remote host DSP (Delphix Session Protocol) communication, the Server is the engine and the Client is the host. The Add Environment wizard allows you to tell the Delphix Engine how the KeyStores and TrustStores have been set up on the remote hosts.

> When the Server or Client authenticates each incoming DSP connection (if enabled), it will validate that each certificate of the incoming connection's identity chain has a "Valid From (or Not Before)" date that is after its own time.
>
> Thus, if your Delphix Engine or host environments are running off of an incorrect (slow) time configuration, then your DSP connections will not work until the offending engine or host's time advances past all incoming certificate's "Valid From (or Not Before)" time.
>
> If correcting the Delphix Engine's or host environment's time configuration may cause issues, then you can workaround this issue by creating and using certificates with a "Valid From (or Not Before)" date which is before your slowest Engine or host.

8.2.5.2.3.1 Adding a new single instance environment after DSP configuration changes

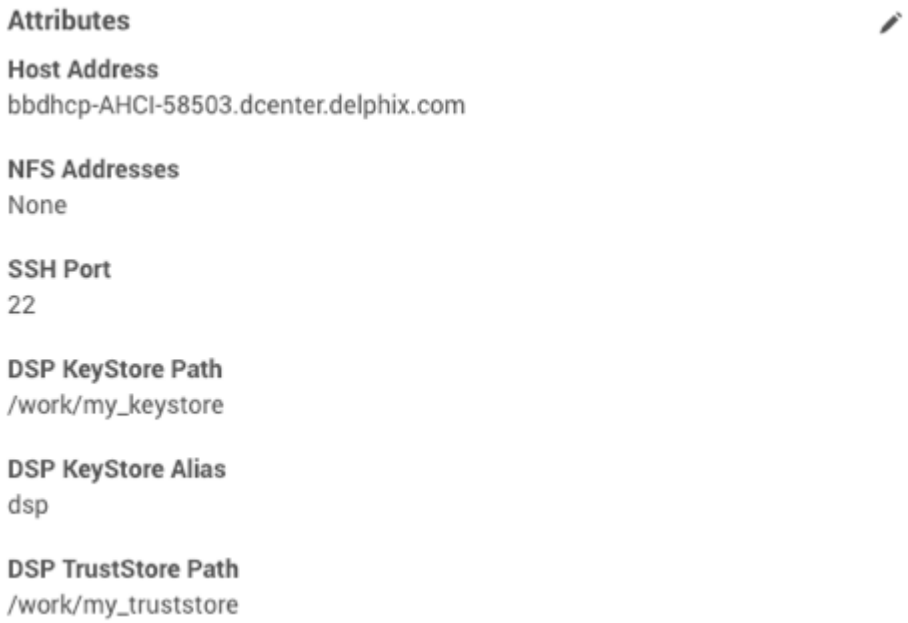
- If server authentication for remote host communication or engine to host throughput tests is desired, make sure the appropriate config is set. For more details refer to Configuring Network Security Settings (see page 966). You will need to create a JKS or PKCS#12 keystore on the remote host with the full CA chain of the DSP key in the keystore. By default, the key will just the signed by the Delphix CA, but you can replace the DSP key if you wish. Refer to KeyStore Settings (see page 987) for more details.
- If client authentication for remote host communication or engine to host throughput tests is also desired, make sure the appropriate config is set. For more details refer to Configuring Network Security Settings (see page 966). You will need to create another JKS or PKCS#12 keystore on the remote host with the desired key pair. Make sure the created keystore has permissions such that it is readable by all environment users. Then, add the full CA chain of the remote host's key pair to the TrustStore on the engine. For more details, refer to TrustStore Settings (see page 989)

Once the appropriate toggles are enabled, and the remote host is all set up, you can now add the environment as shown below. If only server authentication was desired, only the TrustStore fields need to be filled in. If client authentication was also desired, then the KeyStore fields will also need to be filled in

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Environments**.
4. Click the **Actions (...)** menu next to Environments and select **Add Environment**.
5. In the **Host and Server** tab, select **Unix/Linux**.
6. Select **Standalone Host** or **Oracle Cluster**, depending on the type of environment you are adding.
7. Click **Next.**
8. In the Environment Settings tab enter your DSP configurations.
9. Select **Submit.**

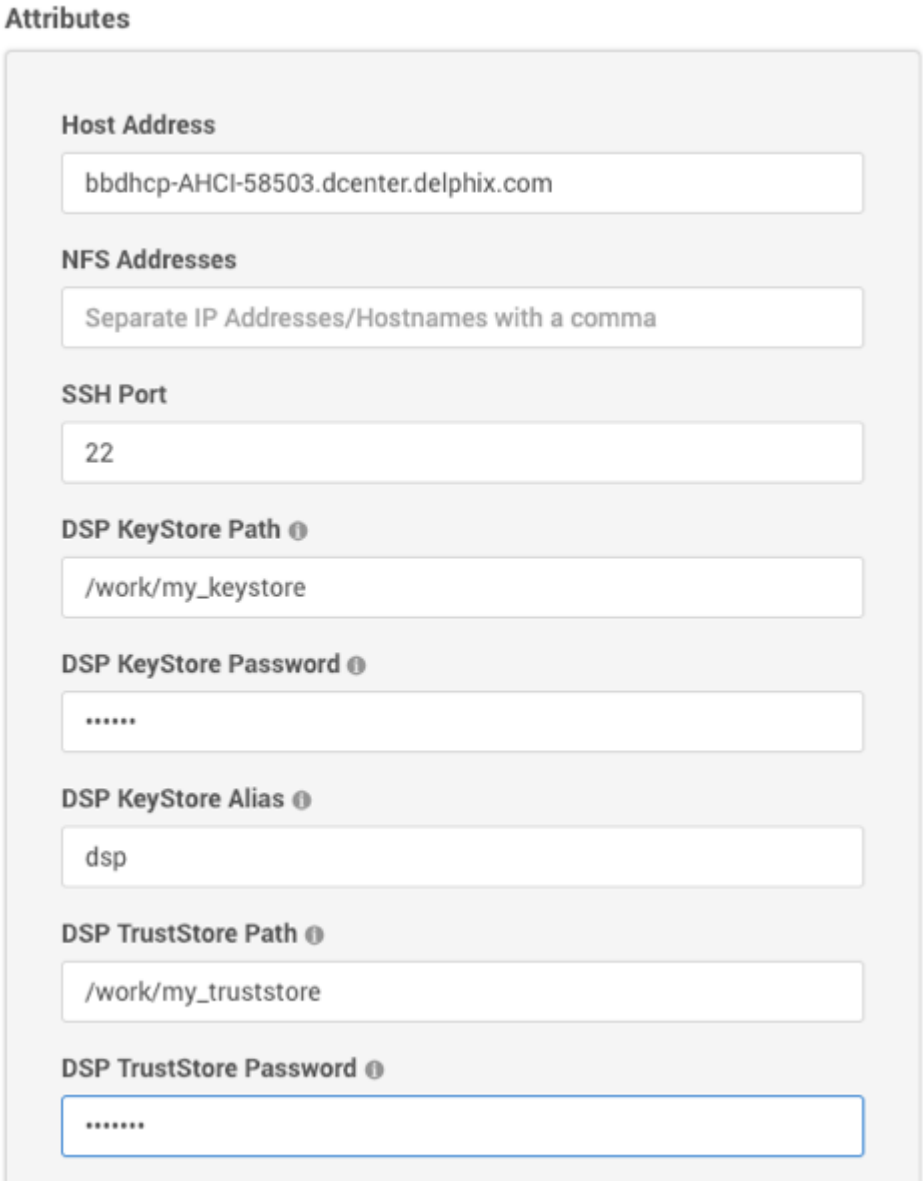
8.2.5.2.3.2 Modifying an existing single instance environment after DSP configuration changes

1. If an environment already exists after enabling server/client DSP authentication, you will need to modify its attributes for host communication to continue working. As detailed in the above section Adding a new single instance environment after DSP configuration changes (see page 984), you will need to set up the appropriate stores on the remote host.
2. Once this is done, the **Details** page of your environment will show the existing DSP attributes.
3. Click the **pencil** icon in the top right corner to edit the DSP KeyStore and TrustStore fields accordingly.



4. Edit the DSP fields as required.

5. Once you are done select the checkmark.



This will need to be done for all existing environments after DSP server/client authentication is enabled otherwise many host communication features will not work.

8.2.5.2.3.3 Adding a new Unix/Linux cluster environment after DSP configuration changes

Adding a new cluster environment is similar to adding a single instance environment, see Adding a new single instance environment after DSP configuration changes (see page 984). The steps for setting up the TrustStore and/or KeyStore will need to be done on ALL nodes of the cluster. For a new cluster, each node

must also be set up to have the exact same path, password, an alias, because the Delphix Engine will use the same configuration for every auto discovered node. If desired, the path, password, and alias configuration can be changed for each node, but only AFTER the cluster has been added. See (see page 0) Modifying an existing Unix/Linux cluster environment after DSP configuration changes (see page 984) for more details.

8.2.5.2.3.4 Modifying an existing Unix/Linux cluster environment after DSP configuration changes

Similar to single instance environments, an existing cluster can be modified if DSP server/client authentication is enabled. In the cluster environment's Details page, each node can be selected and modified individually. You can also use this to change the path, password, and/or alias to be different across nodes for a cluster that was just added.

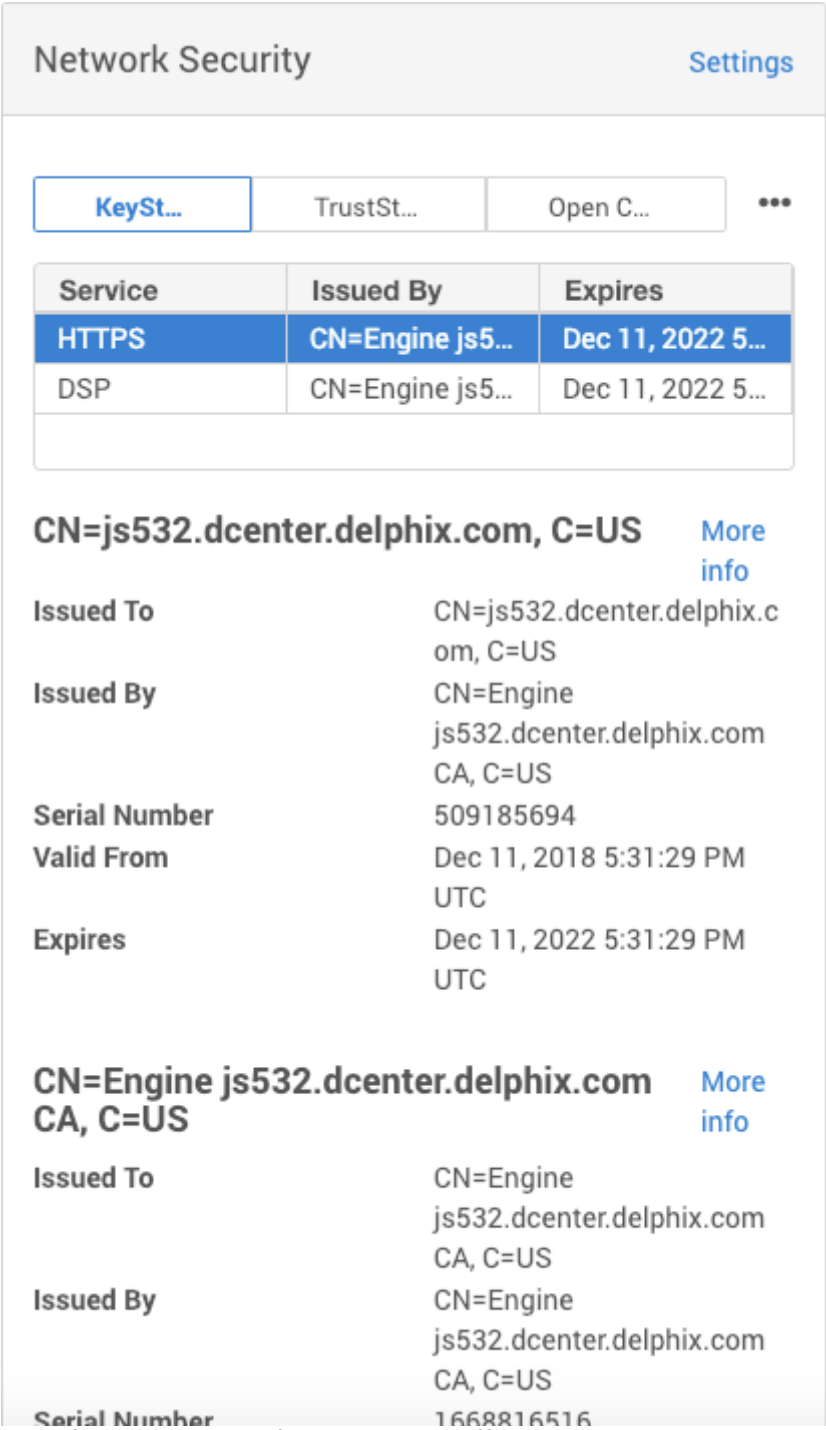
8.2.5.2.3.5 DSP for windows clusters

> This will need to be done for all existing environments after DSP server/client authentication is enabled otherwise many host communication features will not work.

Before adding a Windows cluster, each node must have already been added as a single instance. See Adding a new single instance environment after DSP configuration changes (see page 0) if adding a new Windows cluster. If modifying a Windows cluster after a DSP configuration change, you can modify each Windows node on its environment details page See Modifying an existing single instance (see page 0) for more details, or just use the cluster environment details which will allow you to select which node to modify. See Modifying an existing Unix/Linux cluster environment after DSP configuration changes (see page 0).

8.2.5.3 KeyStore settings

The KeyStore tab in the Network Security panel displays the public key certificates that the Delphix Continuous Data Engine uses for its identity in HTTPS, DSP (Delphix Session Protocol), and stunnel-server communication. For a selected service, the tab displays the complete certificate chain starting from the end-entity certificate and ending with the root CA.



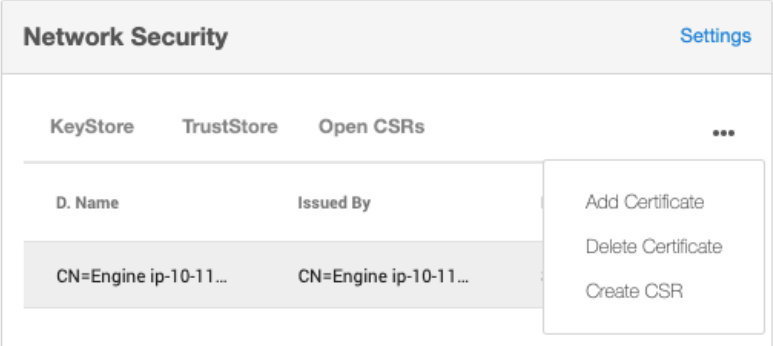
There are two methods of replacing a certificate. The key difference between the two is whether Delphix or the user is providing the key pair (public and private key). For more information see Configuring Network Settings (see page 977).

Delphix Engine generates alerts when certificates in the Keystore or truststore are expired or about to expire:

- A warning level alert is generated if certificates are expiring in 60 days.
- Critical level alerts are generated if certificates expire in 14 days or have already expired.

8.2.5.4 TrustStore settings

The TrustStore tab in the Network Security panel displays all the CA certificates that the Delphix Engine trusts. Click on the Actions (...) menu in the top right to reveal the options for editing the TrustStore contents:



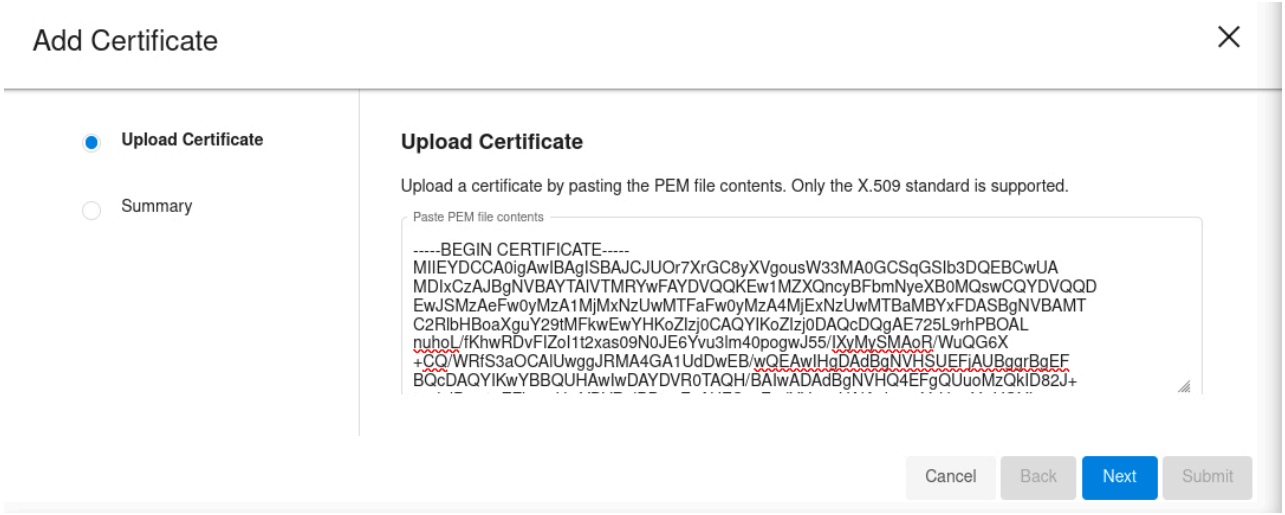
Delphix Engine generates alerts when certificates in the Keystore or truststore are expired or about to expire:

- A warning level alert is generated if certificates are expiring in 60 days.
- Critical level alerts are generated if certificates expire in 14 days or have already expired.

The truststore is used for a variety of connections from the engine to external hosts, such as secure SMTP, and HTTPS proxy connections.

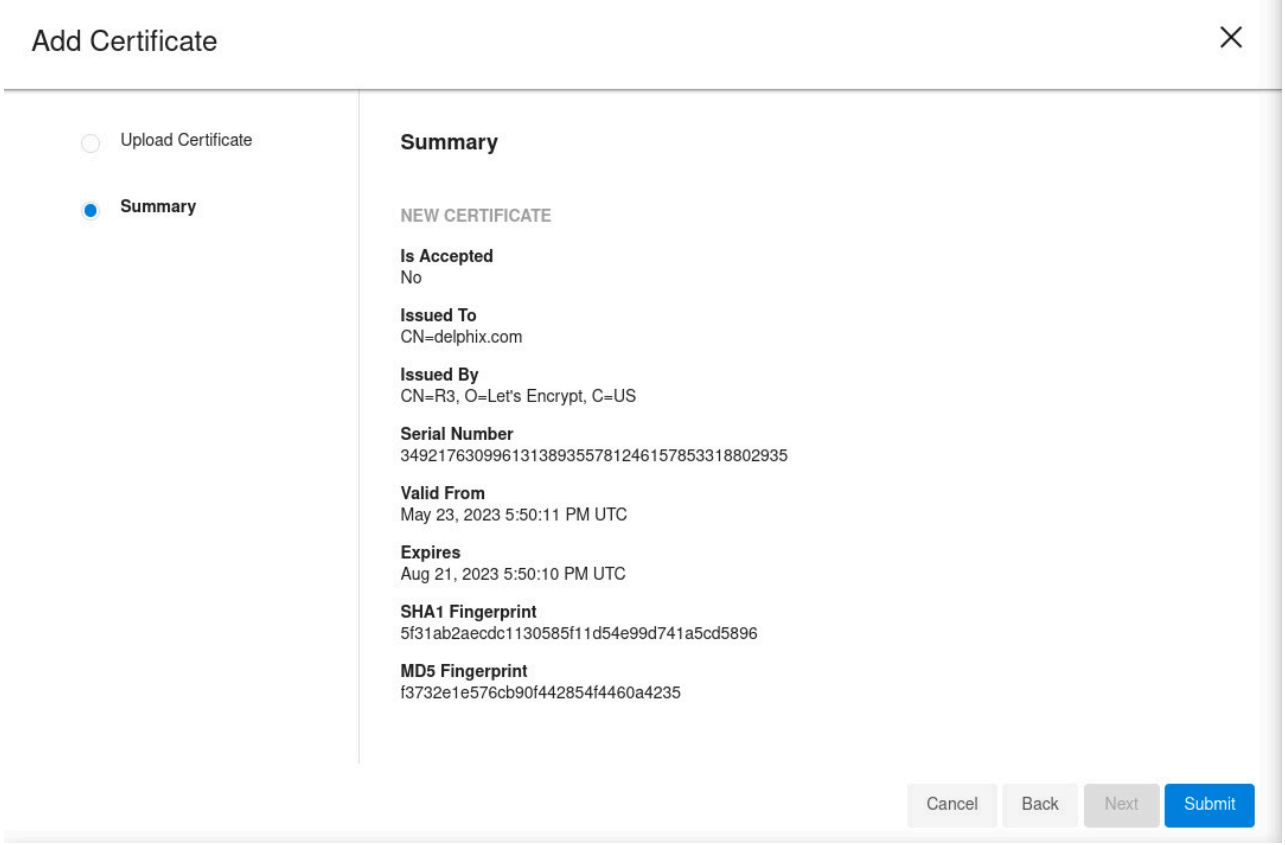
8.2.5.4.1 Adding a certificate

1. From the Actions menu select Add Certificate.
2. In the Add Certificate wizard paste the PEM contents of the CA certificate you want to add. The PEM contents must have the appropriate header and footer included.



3. If you are adding a non-root CA certificate, its signer must already exist in the truststore. So, if you are adding a chain with multiple certificates, you must add them individually starting from the root CA. If not, you will get an error saying that we could not establish a chain of trust.

4. Click **Next** to view a Summary tab where you can confirm the certificate contents.



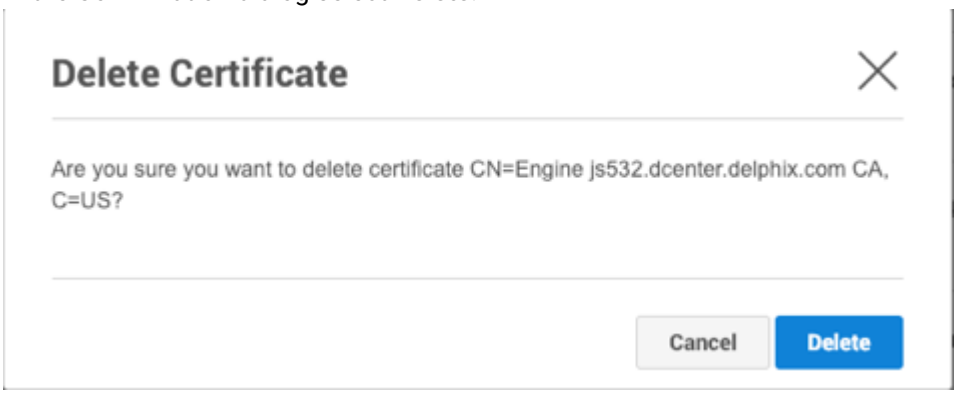
5. Click **Submit**.

8.2.5.4.2 Deleting a certificate

Use this option to delete the selected CA certificate.

> ⚠ Deleting any certificate that breaks an existing chain of trust is not allowed.

1. From the Actions menu select **Delete Certificate** .
2. The default Delphix CA cannot be deleted. Note that deleting any certificate that breaks an existing chain of trust is also not allowed.
3. In the Confirmation dialog select **Delete**.



> ⚠ Unlike the network security settings, any changes to the TrustStore will not require a stack restart.

8.3 Replication security

8.3.1 Choose encrypt and compress when replicating

Delphix provides encryption capabilities when replicating data from one Delphix Engine to another.

Upon selecting the **Encrypted** option under **Traffic Options** the Delphix Engine intelligently compresses before encrypting data. This ordering leads to lower CPU utilization and higher throughput compared to using encryption alone, with the same level of protection. See Configuring Replication (see page 2381) for details. The flag can be set at any time. Once set, it results in sending subsequent replication traffic streams as encrypted.

Train your Delphix Administrators to choose both options.

8.4 Object security

This section covers the following topics:

- User management (see page 992)

- Source database security (see page 993)
- Source and target host security (see page 994)

8.4.1 User management

8.4.1.1 Secure user management

Secure user management is best achieved by integration with your centralized authentication service. Once the integration is complete, create LDAP authenticated named users to facilitate separation of duties, least privileges, and auditing. Disable the out-of-the-box generic ADMIN and SYSADMIN accounts.

8.4.1.2 Use LDAP for authentication

As described under System Configuration above, enable LDAP authentication to leverage your enterprise authentication service and enable SSL/TLS to secure LDAP connections.

8.4.1.3 Create named users

Do not create generic functional accounts such as “QA,” “DEV,” or “TEST.” Such accounts will not leave a proper audit trail and violate the separation of duties principle. Instead, create LDAP authenticated named users.

8.4.1.4 Assign least privileges

Restrict the **admin** and **sysadmin** roles to 1-2 trusted named users each. These roles are highly privileged and must be carefully managed. These roles typically map to a **DBA** and **System Administrator** respectively.

For subordinate users who need to refresh VDBs, assign “Data Operator” privilege on the VDB and “Reader” privilege on the dSource.

For subordinate users who need to provision new VDBs from dSources, assign “Provisioner” privilege on the dSource and “Provisioner” privilege on the Group to which they will assign the VDB.

8.4.1.5 Consider Delphix self-service functionality

The Delphix Self-Service functionality is targeted towards developer and tester self-service, and it contains a more sophisticated privilege model. With this functionality, Delphix Self-Service users do not have access to the Administrator GUI.

Administrators can define multiple data sources as a complete template. They also allocate server resources as a “data container.” The end-user has the ability to update data from the source, from peers using the same source, and from prior images of the source that they have created.

8.4.1.6 Disable ADMIN and SYSADMIN

Once you have established named Delphix Administrators and Systems Administrators, disable the out-of-the-box `sysadmin` and `admin` accounts. You can disable accounts through the CLI.

> When engines created before 5.3.1 are upgraded to 5.3.1 or later they will retain their old username 'delphix_admin'. To avoid complications Delphix recommends creating users with an admin role and then Disabling delphix_admin.

8.4.2 Source database security

8.4.2.1 Choose minimum privileges for Delphix DB user

The Delphix Engine requires a DB user (**delphix_db)** on your source databases. This account is necessary to detect the state of the source and stay in sync. Do not give unnecessary privileges to this user. Leverage the script provided by Delphix in the hostchecker bundle to create a user with the minimum required privileges. The DB user (e.g., **delphix_db**, which is the example used on this page) account can have the same or different user name on each of your source databases.

8.4.2.2 Protect the Delphix DB user password

Because the **delphix_db** user has access to sensitive data dictionary information, take steps to protect access to this account.

8.4.2.3 Use database encryption functionality to encrypt sensitive data at rest

Database vendors provide tools to encrypt data at rest. This encrypts data on disk in order to provide protection at rest. Use the database encryption on your source databases to encrypt sensitive or all data. Delphix integrates seamlessly with database encryption to provide protection at rest.

8.4.2.4 Choose encrypt and compress when linking

Delphix provides encryption capabilities when linking against your source databases. Encrypting while linking can lead to higher CPU utilization and higher throughput. This overhead can be reduced by selecting both the Compress and Encrypt options, in order to compress the data before it is encrypted. Train your Delphix Administrators to choose both options.

8.4.3 Source and target host security

8.4.3.1 Oracle on UNIX

Delphix support for Oracle on UNIX requires an OS account (**delphix_os**) on source database servers and on target servers that will host virtual databases or files. The Delphix Engine uses SSH to send commands to this user, which performs operations on the host. Some of these commands require elevated privileges.

> There is no actual requirement that the account be named **delphix_os** on both sources and targets. You can name the account anything you want; you can also use separate accounts on every source and target.

- **Restrict su - delphix_os to named Delphix admins and system administrators**

Other users of the system do not need access to the delphix_os user. Your Delphix Admins and System Administrators should retain su ability to facilitate troubleshooting.

- **Use SSH Key exchange to allow the Delphix engine to communicate with targets**

Implement public/private key exchange instead of username/password. This allows you to keep the password of **delphix_os** completely secret.

- **Put delphix_os on password rotation** Rotate the delphix_os password in accordance with your enterprise security policy for application software accounts. You should either:
  - implement SSH Key exchange prior to placing delphix_os on password rotation, or
  - script CLI commands to update the password inside the engine as part of the rotation process.

Delphix Professional Services can assist you in integrating Delphix with your enterprise password rotation system.

- **Restrict elevated privilege commands to the lowest level needed** The Delphix Engine uses elevated privileges to provide core features as well as optional features. The Delphix docs describe in detail which privileges are absolutely necessary, as well as techniques for further restricting the commands that can be used. The Delphix Engine ships with support for “sudo” as the privilege elevation system, but also allows for integration with third-party and custom centralized privilege management systems.

8.4.3.2 Windows

Delphix support for SQL Server requires two OS accounts for Windows:

- **delphix_src** – used on the source database server
- **delphix_trgt** – used on the servers which host Virtual Databases Both are required for the Validated Sync target

> There is no actual requirement that the account is named **delphix_src/delphix_trgt**. You can name the account anything you want; you can also use separate accounts on every source and target. Finally, you can create a single account for use everywhere, but this is not recommended since it violates the separation of duties.

8.4.3.2.1 Restrict privileged commands to the lowest level needed

The Delphix user or domain account should have exactly the privileges required in the Delphix documentation. Do not grant additional privileges.

8.4.3.2.2 Put delphix_src and delphix_trgt on password rotation

Change the user or domain account password at regular intervals or in accordance with security policies for application software accounts. Use CLI scripts to quickly modify the password across the Delphix ecosystem. Delphix Professional Services can assist you in scripting and integrating Delphix with your enterprise password rotation system.

8.4.3.2.3 Use minimum privileges on your SMB share

Consult http://technet.microsoft.com/en-us/library/cc754178.aspx to understand how shared folder privileges work. Use the minimum privileges.

8.4.3.2.4 Use windows authentication for SQL server

SQL Server allows authentication via Windows or Mixed mode. Mixed mode allows authentication via Windows or SQL Server.

Windows authentication is more secure; it uses Kerberos security protocol, provides password policy enforcement with regard to complexity validation for strong passwords, provides support for account lockout, and supports password expiration. http://msdn.microsoft.com/en-us/library/ms144284.aspx

8.5 System configuration

There are a number of configuration options available in the System Administration area that help you to secure the Delphix Engine. You can configure these during installation through the setup wizard, or later by accessing the Delphix Setup screens.

8.5.1 Maintain system time with NTP

Establish at least one, but preferably three, corporate NTP servers and sync your Delphix Engine to them. This ensures that audit and error messages display the correct time.

When configuring Delphix Engine on VMware, be sure to configure the NTP client on the ESX host to use the same servers that you enter here. On a vSphere client, the NTP client is set in the **Security Profile** section of the configuration process.

8.5.2 Enable phone home

Phone Home service will send critical information about the Delphix Engine to Delphix Support using HTTPS, on a periodic basis. The use of a Web Proxy Server is fully supported. Phone Home data allows Delphix Support to proactively detect Delphix Engines affected by critical vulnerabilities.

8.5.3 Register your Delphix engine

Registration is fast and easy, and you can do it with or without Internet connectivity from the Delphix Engine. Failing to register the Delphix Engine will impact its supportability and security in future versions.

8.5.4 Enable LDAP for authentication

The LDAP protocol is used by enterprise authentication services. Enabling LDAP authentication allows your Delphix Engine to leverage the password control features of these products, such as expiration, lockout, and complexity.

Import your LDAP server certificate into your Delphix Engine, and enable SSL/TLS.

8.5.5 Enable SMTP and/or SNMP monitoring

When the Delphix Engine encounters errors, it issues **alerts**. Configure SMTP and/or SNMP to forward **alerts** to your central monitoring system.

8.6 GUI security

8.6.1 Overview

The sections in this article cover securing the Delphix GUI, which is similar to securing other web consoles. Some of these solutions include reducing the session timeout threshold, creating a signed certificate, and disabling HTTP access.

8.6.2 Reduce inactive session timeout to 15 minutes

This means that a user will be booted from the session after 15 minutes of inactivity. This is done with a CLI command on a per-user basis by modifying the `sessionTimeout` property of the **User** object, as shown below. The default inactive timeout happens after 30 minutes.

```
myhost.delphix.com>  cd  user
```

```
myhost.delphix.com user> select delphix_admin
myhost.delphix.com user 'delphix_admin'> update
myhost.delphix.com user 'delphix_admin' update *> set sessionTimeout=15
myhost.delphix.com user 'delphix_admin' update *> commit
```

8.6.3 Use a URL from your domain and create a signed certificate

Do not use IP Addresses to access the Delphix Engine. Create a hostname and DNS entry, such as "delphix1.mycompany.com". Delphix Support can assist in converting the engine from a self-signed certificate to a signed certificate that maps to your domain name. Please file a Support ticket to proceed.

8.6.4 Disable HTTP access

Disabling HTTP or configuring HTTP to redirect connections to HTTPS is recommended to protect in-flight user credentials and connections with the engine. This can be done via the command line[287] or through the GUI (see page 972).

8.7 Repave Delphix Engine

8.7.1 Overview

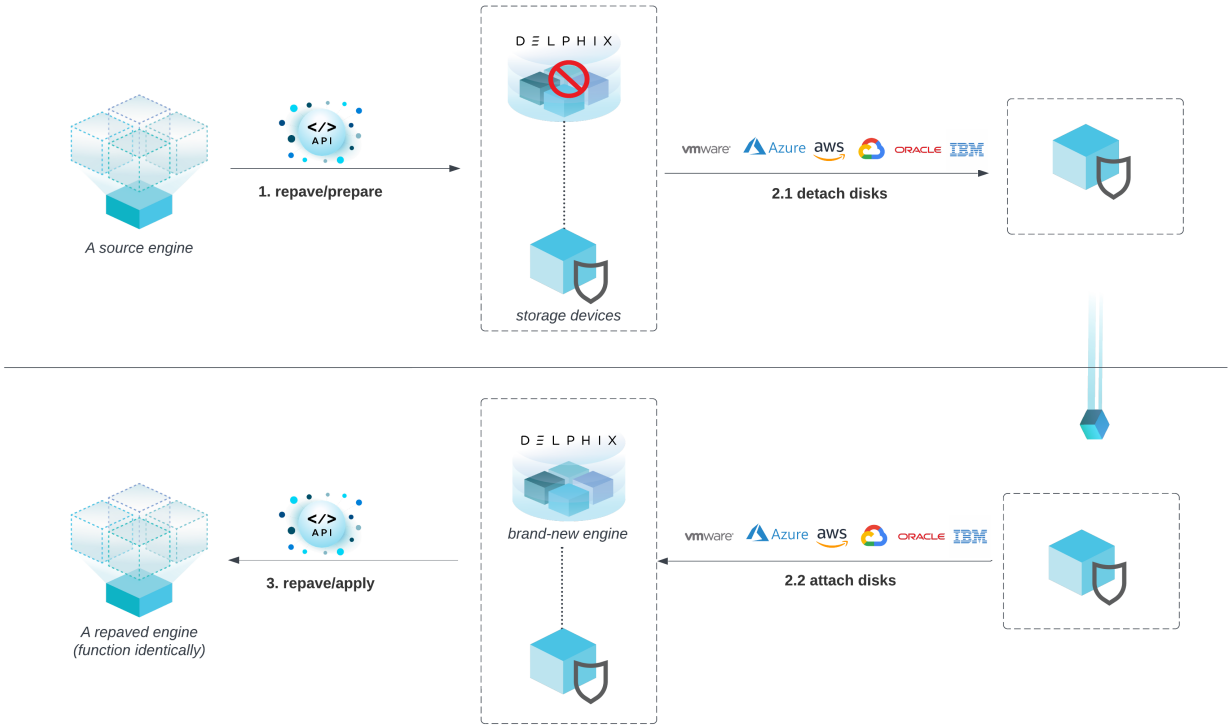
As the rate of new security vulnerabilities rise, Delphix continues developing solutions to mitigate potential risks. Repave is an added solution applied to a Delphix Engine to get rid of any potentially compromising complications. It transfers datasets and required metadata to a fresh, new target engine, without carrying over any compromised binaries. The target engine will function identically to the source engine.

8.7.1.1 Prerequisites

- Repave is supported on Continuous Data (virtualization) engine with block storage as of version 13.0.0.0 and Continuous Compliance (masking) engine with block storage as of version 16.0.0.0.
- You would need to enable the feature flag `REPAVE` before using it on engine version 13.0. After 14.0, the feature flag is enabled by default.
- The target engine shall be in the same version as the source engine.

287 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/cli-cookbook-changing-http-and-https-web-connections-1

8.7.2 How to Repave a Delphix Engine



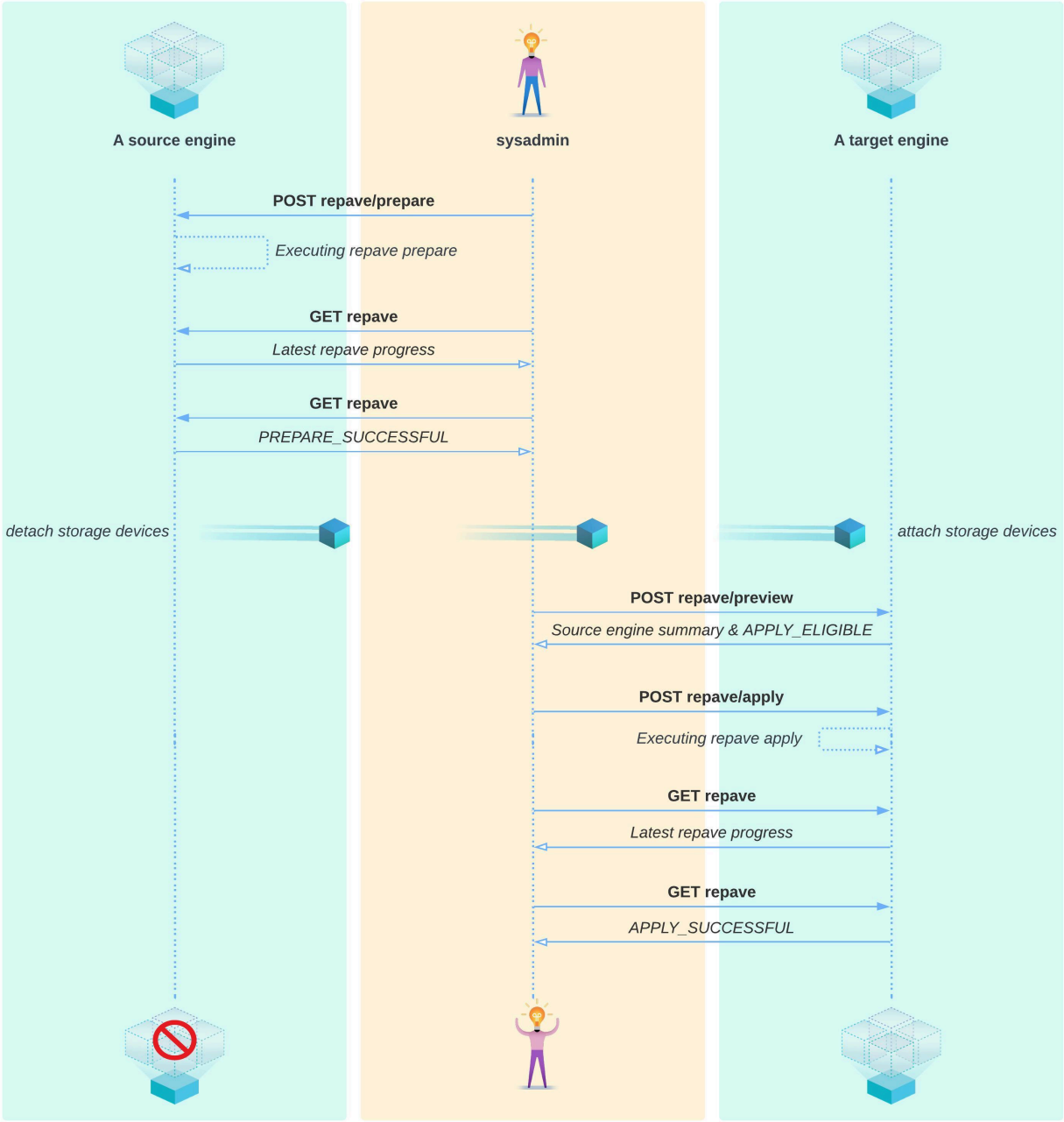
1. A sysadmin initiates **repave/prepare** on the source engine. All Delphix datasets are disabled and the Delphix management service is shut down.
2. A sysadmin detaches storage devices from the Delphix engine and attaches them to a new Delphix engine. This can be done through cloud APIs or manual operation.
3. A sysadmin calls **repave/apply** on the new engine. All previously disabled data sources are enabled again.

8.7.3 Repave API calls

- `POST repave/prepare` initiates the repave preparation on the source engine
- `POST repave/apply` starts the process of applying repave on the target engine.
- `GET repave` returns the repave state, configurable metadata, and engine summary of the current engine.
- `POST repave/preview` provides a preview of the configurable metadata and engine summary of the source engine, as well as determines its eligibility for applying repave on the target engine.

The representation below is a general API workflow of repave.

> You might need to call APIs `Establish Session`, `Login` and `Enable Feature Flag` before calling repave APIs.



8.7.3.1 Tips and usage guidelines

- `repave/prepare` and `repave/apply` returns a job id. However, when the Delphix management service is down, the job API will not work. The other way to check the repave status is by calling `GET repave`.
- All running sources on the engine will be disabled by repave. However, if an environment was disabled before repave, repave might fail to enable the corresponding dSources and vDBs back. Manually enabling the environment and then enabling the dSources and vDBs should resolve the problem.
- If an engine gets stuck in an ongoing repave state for a very long time (like a few hours), restarting the Delphix management stack can forcibly reset repave to a failed state so users are able to try again.
- If the source engine has a hot fix installed, the target engine shall have the same hotfix installed as well before applying repave.

8.7.4 POST calls

8.7.4.1 Establish session

`http://{{delphix_engine_url}}/resources/json/delphix/session`

Establish an API session with Delphix, which is **required** before calling repave APIs.

```
{
    "type": "APISession",
    "version": {
        "type": "APIVersion",
        "major": {{api_version_major}},
        "minor": {{api_version_minor}},
        "micro": {{api_version_micro}}
    }
}
```

8.7.4.2 Login

`http://{{delphix_engine_url}}/resources/json/delphix/login`

Login as `sysadmin`, which is **required** before calling repave APIs.

```
{
   "type": "LoginRequest",
```

```
    "username": "{{sysadmin_user}}",
    "password": "{{sysadmin_pwd}}"
}
```

8.7.4.3 Enable feature flag

`http://{{delphix_engine_url}}/resources/json/delphix/system/enableFeatureFlag`

Enable feature flag `REPAVE`, which is **required** before calling repave APIs.

```
{
    "type": "FeatureFlagParameters",
    "name": "REPAVE"
}
```

8.7.4.4 Repave prepare

`http://{{delphix_engine_url}}/resources/json/delphix/repave/prepare`

8.7.4.4.1 Introduction

Delphix sysadmin calls **repave/prepare** to start preparing for repave on the source engine compromised.

- All the sources will be disabled.
- All datasets and required metadata will be transferred to domain storage devices.
- The Delphix management service will be shut down.

8.7.4.4.2 Request parameters

- `ignoreDisableSourcesFailures`
  If `true`, a failure to disable sources will not block the repave. The default is `false`, if not provided.
- `enableSourcesOnFailure`
  If `true`, when repave fails, data source disabled by repave will be enabled again. The default is `false`, if not provided.

8.7.4.4.3 Response

A job id and action id will be returned if the API is called successfully.

> Please make sure repave state is `PREPARE_SUCCESSFUL` before detaching domain0 storage devices.

8.7.4.4.4 Technical details

The Delphix engine will go through four main steps during this phase:

- **Quiesce engine**
  All domain users will be forcibly kicked out. All automatic replication will be turned off. All running jobs will be cancelled or suspended. All policies will be paused. All sources will be disabled. A Repave state `PREPARE_QUIESCE_ENGINE_FAILED` indicates that the engine is failed to quiesce the engine.
- **Clean up environments**
  Windows environments will be cleaned up, this step is mainly to clean up the iSCSI setting of environment hosts. Any failure will be ignored since this step does not block repave, but users might see legacy iSCSI settings on their Windows environment hosts.
- **Extract metadata**
  Metadata that presents the identity of the current engine will be extracted and stored in domain0. A Repave state `PREPARE_EXTRACT_METADATA_FAILED` indicates that the engine has failed to extract metadata.
- **Export Domain0**
  Services like Delphix management stack and Postgres will be stopped before exporting the domain0 pool. The repave state `PREPARE_EXPORT_DOMAIN0_FAILED` indicates that the current engine has failed to stop services or export the domain0 pool.

```
{
    "type": "RepavePrepareParameters",
    "ignoreDisableSourcesFailures":false,
    "enableSourcesOnFailure":false
}
```

8.7.4.5 Repave apply

`http://{{delphix_engine_url}}/resources/json/delphix/repave/apply`

8.7.4.5.1 Introduction

Delphix sysadmin calls **repave/apply** on the new engine. The Delphix management service will be restarted and all previously disabled data sources will be enabled again.

- Before calling this API, make sure all domain0 pool storage devices have been attached to the target engine correctly.

8.7.4.5.2 Request parameters

One type of parameter is supported for version 13.0.

- `BlockStorageRepaveApplyParameters` is for Delphix Continuous Data engines with block storage.

8.7.4.5.3 Response

A job id and action id will be returned if the API is called successfully.

8.7.4.5.4 Technical Details

The engine will go through six main steps during the this phase. `APPLY_SUCCESSFUL` indicates the engine has applied repave successfully. You can always call `repave/apply` again if there is any failure during repaving.

- **Import Domain0**
  The domain0 pool from the source engine will be imported to the target engine. `APPLY_IMPORT_DOMAIN0_FAILED` indicates the repave apply failed to import domain0. Delphix management stack will be stopped after this step.
- **Check eligibility**
  Check if the target engine is eligible to apply repave. If the target engine version is different from the source engine, or any required hotfix is not installed, or not enough space is in rpool storage, checking eligibility will fail.
- **Setup engine metadata**
  Repave will shut down the Delphix management stack and call a backend task to set up engine metadata (MetaData Service) in this step, the engine metadata was originally from the source engine. It will also rsync `/var/delphix`, snapshot-based metadata and system tunable from domain0 to rpool. `APPLY_SETUP_MDS_FAILED` indicates repave fails in setting up engine metadata. The Delphix management stack will be restarted after this step.
- **Generate metadata**
  The configurable metadata stored in domain0 will be generated on the target engine so the target engine will behave identically as the source engine. `APPLY_GENERATE_METADATA_FAILED` indicates repave fails to generate metadata.
- **Refresh environments**
  All environments will be refreshed in parallel during this step. New iSCSI settings will be set up again on Windows environment hosts. `APPLY_REFRESH_ENV_FAILED` indicates repave fails to refresh environments.
- **Unquiesce engine**
  In this step, all sources disabled by repave will be enabled again, all scheduled replication jobs will be

resumed and policy execution will be resumed. `APPLY_UNQUIESCE_ENGINE_FAILED` indicates repave fails to unquiesce engine.

```
{
    "type": "BlockStorageRepaveApplyParameters"
}
```

8.7.4.6 Repave status

`http://{{delphix_engine_url}}/resources/json/delphix/repave`

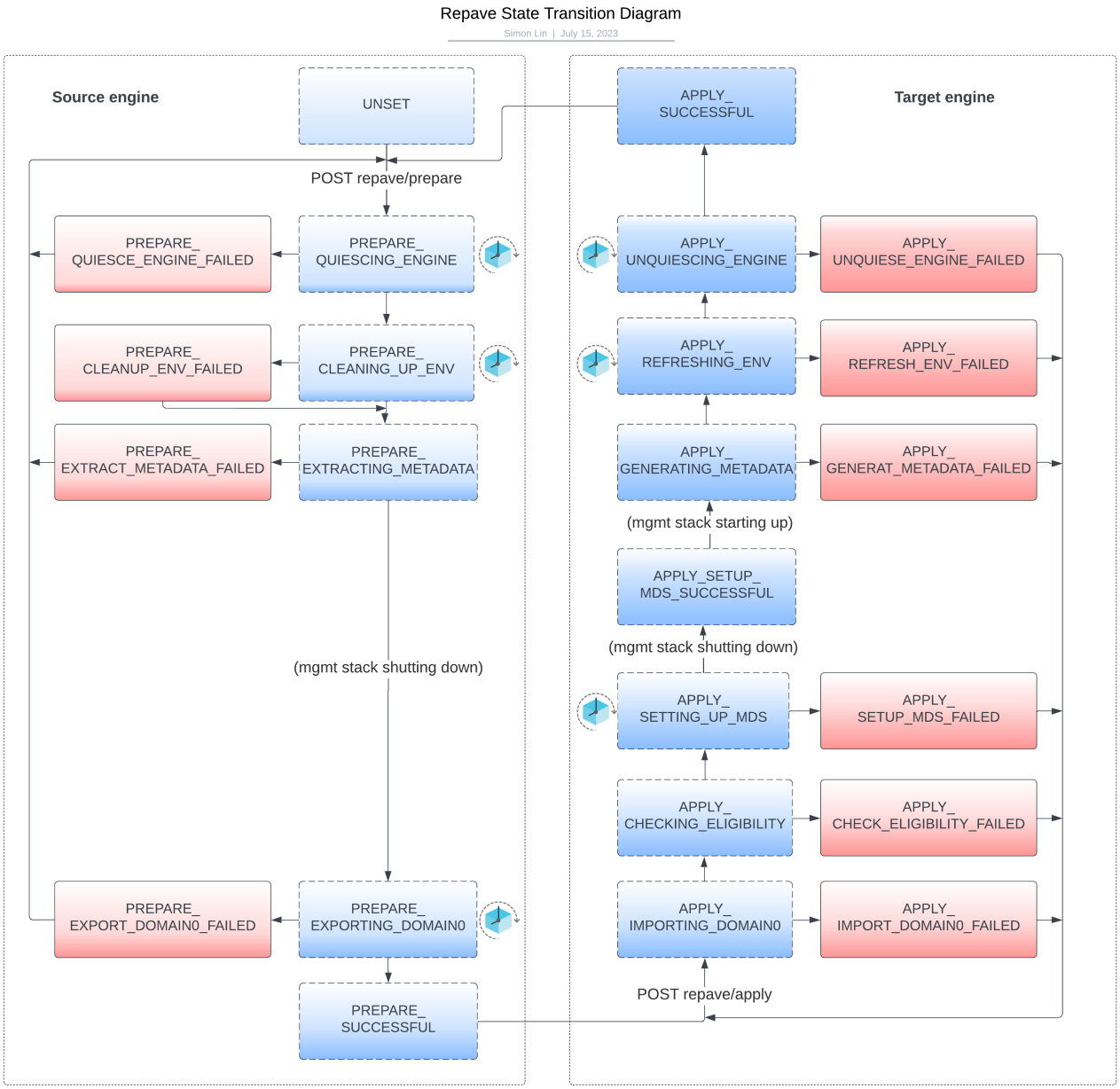
8.7.4.6.1 Introduction

`GET /repave` will show:

- Latest state of Repave.
- Summary of the current engine.
- Metadata of the current engine that will be migrated to the target engine.

8.7.4.6.2 Tips

- Always use `GET /repave` to check repave progress.
- When the Delphix management service is restarting, `GET /repave` might not respond, wait for a few minutes and try again.
- Use the repave state transition diagram below for reference.



8.7.4.7 Repave preview

`http://{{delphix_engine_url}}/resources/json/delphix/repave/preview`

8.7.4.7.1 Introduction

- `repave/preview` is for users to preview the summary and metadata of a source engine.
- The `APPLY_ELIGIBLE` state indicates the current engine is eligible to apply repave.

- The `APPLY_CHECK_ELIGIBILITY_FAILED` state indicates the current engine is not eligible to apply repave, the actual reason can be found from `stateDetail`.

8.7.4.7.2 Request parameters

One type of parameter is supported for version 13.0.

- `BlockStorageRepavePreviewParameters` is for engines with block storage.

8.7.4.7.3 Response

The engine summary and metadata of the source engine.

8.7.4.7.4 Technical details

- The engine summary and metadata of the source engine is stored in the domain0 rpool. `repave/preview` will try to import domain0 pool before previewing them, then domain0 pool will be exported again.

```
{
    "type": "BlockStorageRepavePreviewParameters"
}
```

8.8 Masking sensitive data

Encryption does not protect data that is accessed through applications and database clients, the most likely attack vector. Masking sensitive data before it gets to non-production systems is a critical tool in the security arsenal.

Delphix provides an add-on masking product for simple cost-effective integration. It is also possible to integrate Delphix with any masking technology.

There are many topologies to consider, and the complete explanation of their pros and cons is outside the scope of this document. Delphix Professional Services can assist you in analyzing various masking solutions.

8.9 Audit logs

8.9.1 Review audit logs monthly

Conduct a monthly review of audit logs on your Delphix Engine. Pay particular attention to provisioning operations of unmasked databases, which creates new copies of your production data. See Accessing Audit Logs (see page 804) for instructions.

8.9.2 Forward audit logs to central server via syslog

Forward audit logs to a central audit server using syslog techniques. Delphix Professional Services can assist you with scripts that facilitate this. See also Setting Syslog Preferences (see page 810) for configuration instructions.

8.10 Support security

At least every six months, regenerate the registration code and re-register the engine. See Regenerating the Delphix engine registration code (see page 672) for instructions and the reasons why this is important for security.

8.10.1 Delphix operating system (DxOS)

Delphix Support accesses the DxOS for deep diagnostics and troubleshooting. Access to the Delphix Engine requires access to your network. Typically this is granted via shared troubleshooting sessions over Webex, with full transparency.

If desired, you can enable additional control so that access can only take place when you provide a token.

If you disable Support Access, do not have the token, and you are unable to login as a system administrator, it can become impossible for Delphix Support to login to repair your system. Example: the management stack crashes, the login system becomes unavailable, and you have disabled Support Access and do not have the token. For this reason, Delphix strongly recommends leaving Support Access enabled at all times. If you wish to disable access, generate a unique token once a month and place it in a secure location separate from the Delphix Engine.

8.10.1.1 Disable support access (optional)

Support Access Control is managed as a system administrator in the Server Setup area.

- When set to DISABLED, it is impossible for Delphix Support to login to the DxOS.
- When set to ENABLED (the default), Delphix Support can login to the DxOS.
- When set to ENABLED and calendar time is set and a token generated, Delphix Support can only login with the token during that timeframe, which you provide. Generate the token once/month for an entire month and store it in a secure location separate from your Delphix Engine. When requested, provide the token through a secure means: in your support ticket, via email or SMS to a trusted entity, etc.

8.11 Password policies

8.11.1 Getting started

The password policy feature allows users to create their own custom password policies and enforce the password policy on non-LDAP Delphix Engine users.

8.11.2 Understanding password policies

A password policy is a named password policy that can be assigned to a user. It is a set of requirements that passwords must satisfy.

- minLength - A password must be longer than this length.
- reuseDisallowLimit - The user should not reuse old passwords. This tells the number of last used passwords disallowed to be reused as the new passwords.
- uppercaseLetter - A password must have at least one capital letter.
- lowercaseLetter - A password must have at least one lower case letter.
- digit - A password must have at least one digit.
- symbol - A password must have at least one symbol.
- disallowUsernameAsPassword - A password should not be the same as the user name.

Password policy requirements

When you set a password, it must differ from the most recent password and contain:

- at least 5 characters
- at least one uppercase letter
- at least one lowercase letter
- at least one numeric digit
- at least one symbol such as #, $, !
- do not use username or reverse username

This policy applies to non-LDAP Delphix Engine users. This includes the default users, **delphix_admin** and **sysadmin**. The password policy does not apply to LDAP users.

8.11.3 Default password policy

By default, the Delphix Engine enforces the password policy named **NONE**, which enforces the least possible constraint.

Passwords must contain at least one character.

8.11.4 Changing the password policy

To change the current password policy from the default policy **NONE**, create a custom password policy and select it instead of **NONE**.

8.11.4.1 Who can change password policy for whom

- Domain administrators can change the current password policy for all domain users.
- System users can change the current password policy for all system users.
- Domain regular users (non-administrators) users can only view the password policy.

8.11.5 What operations can be done by administrators

- Create custom password policies
- Update custom password policies
- Delete custom password policies
- Change the current password policy to any of the available password policies
- View available password policies
- View current password policy requirements

8.11.6 Password policy parameters

When you create a password policy, you can set the following parameters:

- Unique name for the password policy
- Minimum length of the password
- Whether password must differ from the last password
- Whether password must not contain the username or reverse user name
- Whether password must contain at least one uppercase letter
- Whether password must contain at least one lowercase letter
- Whether password must contain at least one numeric digit
- Whether password must contain at least one symbol such as #, $, !

8.11.7 Restrictions

- Restrictions for default password policy's modification (named **NONE**):
    - not allowed to delete the default password policy from available list of password policies.
    - not allowed to update any parameters of the default password policy.
- Cannot delete the password policy which is set as current password policy.

8.12 Additional topics

The Delphix Engine provides robust, enterprise-quality security controls. Performing the steps listed in this document will allow you to easily bring your Delphix Engines into compliance with your organization's security policies.

8.12.1 Perform a yearly audit

At least once annually, audit one or more Delphix Engines to ensure compliance with your security policies.

8.12.2 Port scan

Delphix fully supports network security scans, using a tool of your choosing.

8.12.3 Security testing

Many companies require security testing of applications in their environment using a Port Scanner or other Security Penetration Test tools. Delphix supports the use of these security tools with the application credentials available for the engine (e.g., delphix_admin). The Delphix Engine is a closed appliance, and OS credentials on the appliance are not provided for these tests.

8.12.4 Security banner

You can configure a custom security banner that will be displayed to all the users prior to login.

1. Login to the CLI using the sysadmin **username** and **password**.

```
ssh sysadmin@yourdelphixengine
```

2. Set the configuration and commit the changes.

```
delphix > service security
delphix service security > update
delphix service security update * > set banner="Your Message Here"
delphix service security update * > commit
```

The string set in the banner is in plain text only. You can also generate banners with multiple lines, however, this can only be done using the "securityConfig" API. Refer to the Delphix KB article[288] for additional information.

8.12.5 Virtual database security

The Delphix Engine provides advanced storage capabilities and automation to allow rapid provisioning of virtual databases (VDBs), which use only a fraction of the physical storage used by full database copies. Nonetheless, a VDB is equivalent to a physical database **and must be properly secured like any other database**.

By far the most dangerous attack vectors in the Delphix ecosystem are the same ones that existed pre-Delphix: unauthorized access to your non-production systems containing sensitive production data. **You must perform all the same actions to harden virtual databases as you would to harden physical clones**.

For information on securing your virtual databases, consult vendor-specific material and security guides.

288 https://support.delphix.com/Delphix_Virtualization_Engine/Sysadmin/Setting_up_a_Custom_Security_Banner_(KBA6215)

9 Datasets

A major factor of using Delphix (and where Delphix thrives) is integrating with various data sources, enabling the many facets available for getting the most out of your data.

Find a variety of topics to help understand the usage and architecture of data with Delphix in the **Getting started** section. This encompasses topics such as adding environments, creating their users, and using Delphix objects such as virtual databases (VDBs).

In addition, Delphix provides a vast offering of features specific to certain databases, covering a large range of potential data needs. To find detailed overviews, requirements, and workflows particular to specific data sources, select a section below that corresponds with the desired platform.

- Getting started with datasets (see page 1011)
- CockroachDB data sources (see page 1074)
- Couchbase data sources (see page 1113)
- IBM Db2 data sources (see page 1150)
- MongoDB data sources (see page 1228)
- MySQL data sources (see page 1280)
- Oracle data sources (see page 1332)
- Oracle E-Business Suite (EBS) environments and data sources (see page 1652)
- PostgreSQL data sources (see page 1767)
- SAP ASE data sources (see page 1846)
- SAP HANA data sources (see page 1953)
- SQL Server data sources (see page 2020)
- Unstructured files and app data (see page 2219)
- Yugabyte data sources (see page 2266)

9.1 Getting started with datasets

This section is tailored to provide a high-level overview of how to implement your datasets into the Delphix ecosystem. It covers the general data architecture in Delphix, management practices for environments, dSources, and VDBs specific to your datasets, getting the most out of hook operations, setting secure policies, and basic dataset troubleshooting. Select a topic below to get started.

- Datasets overview (see page 1012)
- Dataset support (see page 1012)
- General architecture (see page 1014)
- Environment management (see page 1017)
- dSource management (see page 1040)
- Virtual database (VDB) management (see page 1046)
- Hook operations (see page 1056)
- Shell operations (see page 1064)
- Other operations (see page 1065)
- Policies (see page 1068)
- Basic troubleshooting (see page 1073)

9.1.1 Datasets overview

This section outlines everything required to prepare and provision datasets (VDBs, files, etc) once you have successfully deployed the Delphix Continuous Data Engine. This includes setting up environments, dSources, and virtual databases (VDB). Before beginning, read through this section to better understand the overall ingestion strategy and the requirements to perform it.

- The General architecture (see page 1014) describes the layout of various components outside of Delphix, such as the source databases, environments, and other Continuous Data Engines. Determining how you get data into the Continuous Data Engine will determine where you should place certain environments and how many are required. Based on the ingestion configuration, an equivalent environment called the **target environment** is required to provision to.
- Learn how to create and manage environments in the Environment management (see page 1017) section. The section also touches on various technical requirements and how to ensure a reliable connection between your environments (or hosts) and the Continuous Data Engine. Once connected, the Continuous Data Engine will perform scans to affirm that dataset binaries and installation is available.
- Next, the formal ingestion phase begins with the creation of dSources (see page 1040). dSources are a central component in how Delphix's data virtualization is able to produce ephemeral copies. Once represented within the Continuous Data Engine, its state is then managed through Timeflow, Snapshots, and provisioning new database copies.
- The provisioning phase concludes with the creation of virtual databases (VDBs) (see page 1046). Once created, they can be easily distributed to your application teams without worry of overhead or broken infrastructure. Similar to a dSource, users can snapshot the VDB's state or provision new VDB's. If they get into a broken state or need to view an earlier point in time, the refresh capability can easily rollback time.
- The Hook operations (see page 1056) and Policies (see page 1068) sections provide functions that can be applied to all datasets and sources. Hooks and Policies help manage and optimize the interaction with datasets.
    - Hooks are scripts or sets of commands executed at specific events or stages of a dataset's lifecycle, to ensure custom processes are integrated seamlessly.
    - Policies define the rules and behaviors that govern the handling and maintenance of datasets within the Continuous Data Engine.

Due to differences between data source connectors, it is recommended to supplement each of these sections with connector-specific requirements. Further details can be found within each data source's documentation.

9.1.2 Dataset support

The data source connectors that Delphix Continuous Data can ingest and provision are divided into four categories. Categories differ by their support and development teams.

Category	Description

Standard Connector	Fully supported by Delphix Support. Entitled via Delphix Continuous Data.
Select Connector	Fully supported by Delphix Support. Entitled via the purchase of an add-on.
Premium Connector	Not supported by Delphix Support. Supported by a third party. Entitled via the purchase of an add-on.
Custom Connector	Not supported by Delphix Support. Supported by its open-source community or Delphix Services.

Users of Standard and Select Connectors can submit support tickets through the Delphix Support Portal[289]. Users of Premium Connectors can receive support through the third party that provided them with the connector. Users of Custom Connectors can receive support through the connector's open-source community or contact their Delphix Account Team for guidance.

The list below describes the category each Delphix-supported connector falls under. Reference this list to identify if Delphix Support supports a data source connector. Documentation for Standard and Select Connectors can be found within this documentation portal.

Connector Name	**Category**
Apache Cassandra	Select Connector
CockroachDB	Select Connector
Couchbase Server	Select Connector
IBM Db2 (for LUW)	Standard Connector
IBM Db2 (for z/OS)	Premium Connector (PopUp Mainframe)
Microsoft SQL Server	Standard Connector
Microsoft SQL Server Backup Ingestion	Select Connector
MongoDB	Select Connector

289 https://support.delphix.com/

MySQL	Select Connector
Oracle	Standard Connector
Oracle Backup Ingestion (OBI)	Select Connector
Oracle E-Business Suite (EBS)	Standard Connector
PostgreSQL	Standard Connector
SAP Adaptive Server Enterprise (ASE)	Standard Connector
SAP HANA	Standard Connector
SAP IQ	Select Connector
Unstructured Files (vFiles)	Standard Connector
YugabyteDB	Select Connector

9.1.3 General architecture

9.1.3.1 Ingestion overview

Delphix Continuous Data Engine automatically delivers copies of data to developers and testers to enable high-quality, on-time application development while mitigating infrastructure costs.

These virtual data copies are full read-write capable database instances that use a small fraction of the storage resources that a normal database copy would require. This helps promote a seamless flow of updates and changes to your data in real-time.

In order to create virtual data copies, Delphix Continuous Data Engine must ingest a copy of the source database into persistent storage on the engine, to later be provisioned as one or more copies.

After a copy is ingested, an imperative step is to preserve its current state by taking a snapshot. Subsequently, generating a VDB by provisioning it from the ingested copy establishes a replica of the source database. This VDB can be further enhanced by allowing snapshots or bookmarks, facilitating the capture of its current state for future reference.

Implementing functionality to rewind the VDB to a previous state is crucial. This action allows you to navigate and revert to specific points in the VDB's history. A dSource must consistently reflect the most recent changes from the source database. To do so, you must enable the refreshing of the dSource, maintaining synchronization with the source data.

Furthermore, for a VDB to stay current, it can be refreshed by updating it with the latest changes from a new snapshot of the dSource. Lastly, perform cloning of a VDB to create a child VDB.

Ingestion is the important process by which a consistent, ready-to-run copy of the data is prepared and captured, and persisted as a Delphix snapshot in Delphix storage, so that it can be provisioned to virtual copies as Delphix vDB(s). Before beginning, it's important to understand the recommended ways to ingest.

There are two architectural models by which Delphix performs ingestion:

9.1.3.1.1 Direct ingestion

Direct ingestion is an approach where the Delphix Continuous Data Engine is able to extract (directly from the true production source) the necessary information to reconstruct the source database and persist it into storage on the Delphix Continuous Data Engine. This direct methodology requires no intermediate host nor instance of the source database. The Delphix connector must interact directly with the true production source system to request the data to be persisted. There is no activity whatsoever on an intermediate staging host; instead the blocks are directly written to storage on the Delphix Continuous Data Engine.

The two primary examples of Direct ingestion are:

1. Unstructured files ingestion (does not require a connector).
2. Oracle database ingestion via direct integration with Oracle RMAN (Recovery MANager) utility.

For unstructured files ingestion, the Delphix Continuous Data Engine will connect directly to the true production source of the files and RSYNC them onto a filesystem in Delphix Storage.

Oracle database ingestion via RMAN integration is where the Oracle RMAN utility is able to ship the entire storage block map to Delphix when Delphix requests it, then later just the changed blocks since the previous ingestion (an incremental roll forward capability).

9.1.3.1.2 Staged ingestion

Staged ingestion dictates and expects reconstruction of the source database on an intermediate host (separate from the true source system) or instance of the source database. The intermediate platform is called Staging, and the persistent storage which backs the staging database is provided over the network by the Delphix Continuous Data Engine.

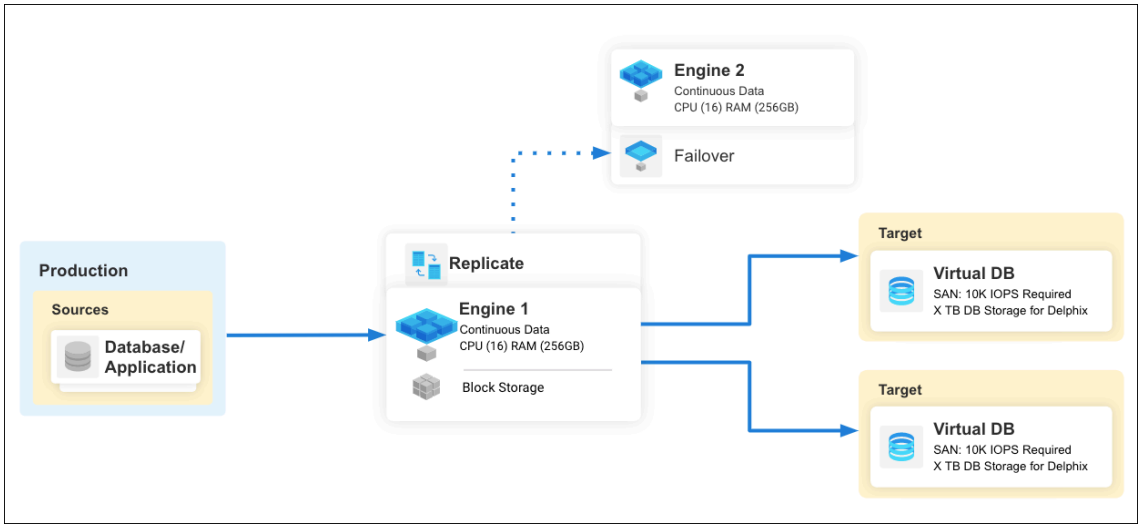
The majority of Delphix Continuous Data Connectors perform Staged ingestion.

Two common examples of Staged ingestion are:

1. Backups and logs.
2. Various forms of replication (physical or logical).

Backup and log ingestion is where database backups and transaction logs from a true production source database are recovered and rolled-forward on a staging host, where the persistent storage is provided by the Delphix Continuous Data Engine. The activity is completely independent and isolated from the true source database and system. This example of Staged ingestion requires zero communication between the Delphix solution, and the true production system, as long as the needed backups and logs are made accessible to the staging host.

Replication ingestion is where the staging host is configured as a replica to true production. Each DBMS system has its own specific facility and methodology for implementing replication. Typically some sort of direct communication is required to exist between the staging host and the true production source host.



9.1.3.2 Staging Pull vs. Staging Push

As described above, Staged ingestion involves an intermediate environment that mounts Delphix storage over the network for persistent storage (rather than locally-attached, physical storage) upon which the source database is reconstructed or prepared.

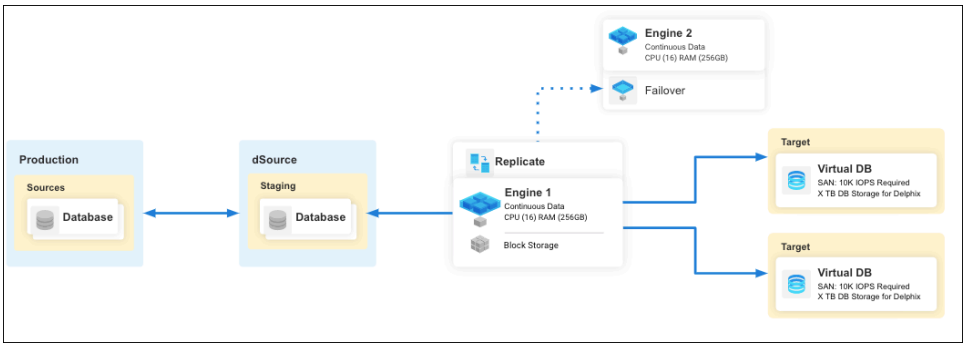
There are two forms of Staged ingestion:

1. Staging Pull
2. Staging Push

The distinction between the two hinges on whether or not Delphix Continuous Data performs the steps to prepare the data contents in the staging dataset.

- **Staging Pull**: Delphix Continuous Data performs the preparation and reconstruction (e.g. database restore/roll forward or replication configuration).
    - The process is entirely under the control of Delphix and cannot be modified nor substantially customized by the user. The benefit is that the heavy lifting to prepare the data on the staging host is automated and entirely taken care of by Delphix, simplifying the operational steps for the user. The downside is that the process is hard-coded and cannot be customized or modified, nor integrated with other third-party applications (like various Enterprise Backup solutions). This approach typically results in a logically identical copy of the pristine production data.
- **Staging Push**: The user, outside of Delphix Continuous Data, performs the steps of data preparation and reconstruction.
    - The user can tailor and customize the process, integrate with third-party tools, and is free to create the data contents however they wish. This approach is wide-open and can be performed however the user prefers, as long as the outcome results in a valid dataset (as determined by the corresponding DBMS) at the time that a dSource snapshot is taken in Delphix Continuous Data.
        - This approach is infinitely more flexible, in exchange for increased responsibilities placed upon the user. This approach can produce logically identical contents of the

pristine source data, but also offers flexibility to filter, modify, or reorganize the data so that it is not logically identical to the original source.



9.1.3.3 Provisioning

After the data source has been successfully ingested into Delphix Continuous Data Engine and represented internally as a dSource, it can be provisioned as a virtual database (VDB). VDBs are deployed onto a target environment and leverage the binaries and resources that are made available.

Once the VDB is running, a wide array of functionality is made available to it, such as refreshing it back to the latest dSource's snapshot, snapshotting a new branch of the dataset, or sharing it with team members. Then, when no longer needed, the virtual database can be destroyed and resources are instantly reclaimed in the target environment. Delphix Continuous Data Engine's virtualization capabilities drastically lower the effort required to manage datasets.

In advanced scenarios, virtual database management might be required to offload target environments or share datasets with other engines. For example, the migrate capability allows administrators to move the VDB from one target environment to another. This is particularly helpful if a target environment is overloaded or the end user is in a different region or zone.

Similarly, a virtual database can be copied to other engines through the process of Replication. In this scenario, the replica is a read-only virtual database in which other virtual databases can be provisioned from where the dSource is not directly available. This is valuable when combined with Delphix's Continuous Compliance's masking and production and non-production zones. In these scenarios, a masked virtual database is provisioned in production and then replicated over to the non-production zone to eliminate data leak risk. It is also broadly helpful to organize engines and limit access to certain datasets.

9.1.4 Environment management

9.1.4.1 Overview

In Delphix Continuous Data's architecture model, an environment is a single instance host or cluster of hosts that run database software. For example, a Linux system running PostgreSQL. Environments can either be a source (where data comes from), staging (where data is prepared/masked), or target (where data is delivered and used by developers and testers). To identify which environments are needed and how many, you must understand the general architecture documentation and the connector's supported ingestion models.

Each environment has its own properties and information depending on its source, staging, and target environment type and the dataset it will support. In this section, you can learn general the requirements to

configure an environment, how to add them to Delphix Continuous Data, and discover the environment dataset's binaries. If there is a configuration specific to certain connectors, it will be called out.

9.1.4.2 Environment configuration requirements

The following list outlines the requirements of an environment, such as name, host address, network requirements, or Delphix Toolkit path, to ensure a valid connection with Delphix Continuous Data. During the environment creation process, we recommend using this list as a guide along with the data source connector's specific environment requirements. These environments must be created before any configuration is performed within Delphix Continuous Data. Follow the other Environment Management sections for additional information on each of these requirements. In addition, please take note of the Operating System specific environment differences.

Requirement	**Description**
Environment Name	A unique, custom label to identify the environment within Delphix Continuous Data. This should be a user-friendly label to uniquely identify the environment.
Host Address	The fully qualified domain name (FQDN) or IP address for the environment.
Network Requirements	Communication between Delphix Continuous Data and environments occurs through various TCP and UDP which requires various ports to be open and available. Ports must be made available on the engine and the environment. Connector Specific: Many connectors require additional Ports to be opened to communicate with the database. Please consult your data source's connector for additional networking information. Unix/Linux: Configuration requires specification of the Secure Shell Protocol (SSH) port. By default, Delphix Continuous Data Engine will use port 22. Windows: Configuration requires the installation of the Delphix Connector on the target environment which is made available to the source or staging environments. Consult the Delphix Connector documentation **(LINK)** for more information.

Environment User(s)	The operating system (OS) user(s) for the environment. These are the users who have permission to ssh into an environment (Unix-based environments) or access the environment through the Delphix Connector (Windows-based environments). Multiple authentication methods are supported, such as Username/Password, Public Key, and Password Vault. Privilege elevation can also be configured in this phase.
Delphix Toolkit path (link to section in requirements area)	Unix/Linux: Location to install the Delphix Toolkit on the environment. <Link to glossary> Windows: This will be specified during the installation of the Delphix Connector.
Java Development Kit (Optional)	Each environment requires a JDK to communicate with Delphix Continuous Data. By default, the engine will provide the Eclipse Adoptium Java via the Delphix Toolkit. If the user would prefer to bring their own Java, follow these directions for details. **(link)**
Network File System (NFS) Addresses (Optional)	Unix/Linux: A list of authorized IP addresses that can mount a Delphix file system persisted by Delphix Storage. Additional IP addresses of the host. By default, Delphix Continuous Data Engine will automatically use the Host Address parameter above for NFS requests Windows: N/A
Delphix session protocol (DSP) Configuration (Optional)	If an environment already exists after enabling server/client DSP authentication, you will need to modify its attributes for environment communication to continue working, you will need to set up the appropriate stores on the remote environment.
Cluster (Optional)	Environments can be defined as part of a cluster. During this phase, the user will define one node of that overall cluster. Some of the above properties change slightly. Consult the data source's documentation for additional information.

9.1.4.3 Adding and discovering an environment

9.1.4.3.1 Adding an environment

Once the environments have been created and the network connectivity requirements are configured, they can be added to Delphix Continuous Data. Source or staging environments will contain the datasets made available for ingestion or provisioning. Target environments will provide the database binaries and hardware to perform the virtualization actions. Before proceeding we recommend installing the necessary dataset binaries on the environments.

To add an environment, navigate to the **Manage > Environments** tab and follow the steps of the configuration wizard. Additional information can be found in the Adding an environment (link). Once added, Delphix Continuous Data will maintain a connection to the environment and begin the discovery process.

9.1.4.3.2 Discovering an environment

Each environment added to Delphix Continuous Data is composed of two primary components: Installations (Dataset Homes) and Databases (Instances or Repositories). Installations contain the dataset's binaries which maintain the specific Databases and datasets you plan to ingest. Each Database then maps to a single dSource which is a primary object in Delphix Continuous Data's ingestion model. Delphix Continuous Data must keep track of all of these objects to perform ingestion and provisioning.

Once environments are added, the Discovering an Environment(link) process will automatically scan the source, staging, and target environment for available installations, databases, and files. Note: Target environments should only contain binaries as their purpose is to provision virtual databases, but not source datasets. In test scenarios and with proper organization, the same environment can be used as both a staging and target environment.

However, not all data source connectors fully support automated discovery. Adding a dataset home or Adding a database documentation describes how to manually add these objects. These processes are also useful if automated discovery misses a Dataset Home or Database. Consult the data source connector's documentation to understand which discovery mechanism each connector supports.

Once the Installations and Databases have been successfully discovered, you can continue to the dSource management phase. Follow the Environment operations section to learn more about the specific steps for each major action.

9.1.4.4 Network requirements

9.1.4.4.1 Overview

This topic covers the general network and connectivity requirements for connecting the Delphix Continuous Data Engine to the source, staging, and target environments. Ensure all ports are open and available before adding an environment to the engine.

9.1.4.4.2 Delphix Continuous Data Engine Connectivity

During the installation of the Delphix Continuous Data Engine, our documentation provided various ports that must be opened. Consult the Deployment section's Network connectivity requirements[290] and Deployment for additional engine-specific information.

9.1.4.4.2.1 NFS mounts on environments

Delphix Continuous Data Engine shares its storage space with the staging and target environments using NFS mounts.

Protocol	Port number	Use
TCP/UDP	111	Remote Procedure Call (RPC) port mapper used for NFSv3 mounts.
TCP	1110	Network Status Monitor (NSM) client from target environment to Delphix Continuous Data.
TCP	2049	Network File System (NFS) client from target environment to Delphix Continuous Data. (NFSv3 and NFSv4).
TCP	8341	Sending LogSync data from source to Delphix Continuous Data (for LogSync).
TCP	8415	SnapSync control and data from source to the Delphix Continuous Data. V2P control and data from the target environment to the Delphix Continuous Data.
TCP	54043	Client mount daemon (NFSv3 only).
TCP	54044	Lock state notification service (NFSv3 only).
TCP	54045	Network Lock Manager (NLM) client from target environment to the Delphix Continuous Data (NFSv3 only).
UDP	33434 - 33464	Traceroute from the target database server to the Delphix Continuous Data (optional).

290 https://cd.delphix.com/docs/latest/network-connectivity-requirements

9.1.4.4.3 General environment connectivity

Environments share a common set of ports that must be opened to establish communication with the Delphix Continuous Data engine.

9.1.4.4.3.1 Data source connector connectivity

Many connectors have their own set of connection requirements to consider due to database differences. We recommend navigating to the connector's specific documentation on environment creation to identify other network requirements.

Due to these differences, many connectors support a HostChecker (sometimes called HealthChecker) utility to ensure the validity of the newly created environment. Consult the connector's documentation for more information.

9.1.4.4.3.2 Secure Socket Shell (SSH) connectivity

The Delphix Continuous Data Engine connects to Linux and Unix environments using Secure Shell Protocol (SSH). This connection is long-running, performed through the `sshd` (OpenSSH Daemon) program.

Allow the following inbound port for SSH connection to each environment.

Protocol	Port numbers	Use
TCP	22	SSH connections to the source, staging, and target database environments.

9.1.4.4.3.3 Delphix connector connectivity

Windows environments require the Delphix Connector to communicate with the Delphix Continuous Data Engine. The Delphix Connector is installed on all target environments. Source and staging environments will use the target environment's connector for communication. Refer to the Delphix Connector documentation for additional information.

9.1.4.5 Environment users

9.1.4.5.1 Overview

Not to be confused with Delphix Continuous Data Engine administrator (admin) users and system administrator (sysadmin) users, environment users must be created for each environment to perform actions on each environment. These users are configured for each data platform to interface with the database instance on the host.

1. Login to the Delphix Management application using Admin credentials.

2. Click **Manage**.
3. Select **Environments**.
4. Click on the existing environment name you want to modify and open the environment information screen.
5. In the **Details** tab, click the **Plus** icon located next to Environment users.
6. There are four ways that can be used for the Delphix Continuous Data Engine to login into the environment.
    a. Enter the Username and Password for the OS user in that environment and click **Validate**.
    b. If you want to use a public key[291] for logging into your environment:
        i. Select 'Username and Public Key' for the Login Type.
        ii. Copy the public key that is displayed, and append it to the end of `~/.ssh/authorized_keys` file of the new user being added. If this directory or file does not exist, you will need to create it.
        iii. Run `chmod 600 authorized_keys` to enable only the file owner with read and write privileges.
        iv. Run `chmod 755 ~` to make your home directory writable only by your user and no other user may write to it.
        v. The public key needs to be added only once per user and per environment.
    c. Specify a Custom Key Pair
    d. Specify a Password Vault. Consult the Password Vault Support documentation[292] for more information.
7. Click the **Add** icon to save the new user.
8. To change the primary user for this environment, select the environment. Then click the **'star'** icon next to **Environment Users**. Only the primary user will be used for environment discovery.
9. To delete a user, click the **Trash** icon next to their username.

9.1.4.5.2 Privilege elevation profiles

Privilege elevation profiles exist to provide the Delphix Continuous Data Engine with a mechanism for running privileged commands in a secure way to achieve the following:

- Mount and unmount Network File System (NFS) file systems
- Create and remove directories in paths not owned by the Delphix Continuous Data Engine OS user
- Examine the running process list
- Run commands as root

291 https://cd.delphix.com/docs/13.0.0.0/cli-cookbook-configuring-ssh-host-verification-for
292 https://cd.delphix.com/docs/latest/password-vault-support

9.1.4.5.2.1 How privilege elevation profiles work

Privilege Elevation Profiles need to be tailor-made to work with non-standard environments that may use third-party or proprietary privilege elevation mechanisms other than sudo. You are strongly encouraged to work with Delphix Professional Services to formulate reliable profile scripts.

Privilege Elevation Profiles exist within a two-tier cascading hierarchy. This means there is one default profile for the entire Delphix Continuous Data Engine that should contain scripts for all the operations that require privilege elevation. Additional profiles may contain a subset of the scripts. When a non-default profile is used, the Delphix Continuous Data Engine uses that profile's scripts where they exist and reverts to the scripts in the default profile if no script for the operation exists. By default, the Delphix Continuous Data Engine ships with simple scripts that pass commands to the standard UNIX sudo command.

All environments added to the Delphix Continuous Data Engine get added with the default Privilege Elevation Profile. The profile can be assigned on a per-environment basis. Below description shows how an environment which is using a non-standard profile, will use scripts in the cascading model.

In order to create a privilege elevation profile, you must create both a profile and a profileScript. Scripts exist for particular operations, which include:

- dlpx_mount
- dlpx_umount
- dlpx_rmdir
- dlpx_mkdir
- dlpx_ps
- dlpx_pfexec
- dlpx_pfexec_as_user

There are three parameters to consider while creating a new profile:

1. name:
2. contents:
3. Profile:

9.1.4.5.2.2 Support for privilege elevation profiles

Writing and troubleshooting scripts, such as those required for Privilege Elevation Profiles, is out of scope and not covered by Delphix Support.

9.1.4.6 Delphix toolkit

9.1.4.6.1 Overview

The Delphix Toolkit is a group of files that allows communication between Delphix Continuous Data Engine and remote environments. Delphix Toolkit includes:

- Database and Host scripts
- Delphix Session Protocol (DSP) binaries

- Delphix HostChecker
- Java Development Kit (JDK)

The Delphix Toolkit is automatically installed on the environments at the specified toolkit path whenever an environment is added or refreshed.

9.1.4.6.2 Toolkit size and predicted growth

Delphix Toolkit's size depends on two factors:

1. The number of Delphix client applications running on the environment
2. The number of Virtual database(VDBs) on the environment (if any)

Currently, there are three Delphix client-side applications including the SnapSync, Delphix Connector, and Virtual to Physical (V2P) client. Each of the clients that run from the client-side toolkit generates their own logs - info, trace, debug, error. Each level of logging is restricted to a maximum of 10 log files and these log files are capped at 10 MB each.

Therefore the maximum space occupied by the toolkit directory on the Source server is its initial size of 400 MB + 1200 MB = 1.6 GB.

However, on the target server, the maximum toolkit size is initial size 400 MB + 800 MB + (Number of VDBs * 1MB)

9.1.4.7 Java Development Kit (JDK)

9.1.4.7.1 Overview

The Java Development Kit (JDK) is required on each environment to run the Delphix Toolkit. The JDK is provided by default with the Delphix Connector. Delphix Continuous Data Engine uses Eclipse Adoptium (formerly known as AdoptOpenJDK) by default.

Users can provide their own Java when adding an environment and in the environment configuration page. This includes Oracle JDKs, Adoptium Temurin JDKs, and JDKs specific to HP UX and AIX hosts.If you want to use a more recent version than the one that comes with Delphix Continuous Data Engine, you can do so when adding an environment to a given Delphix Continuous Data Engine.

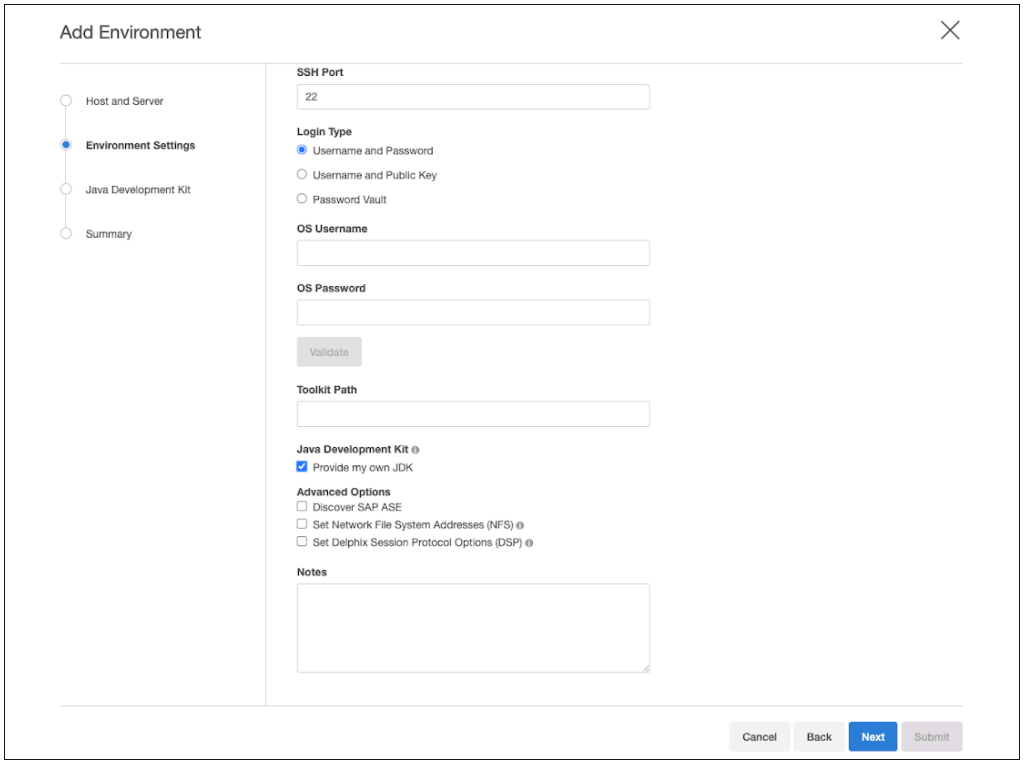
Note that the JDK is provided per node or host, not per cluster or environment. Initially, when creating a cluster all hosts must have Java at the same absolute path. However, once the cluster has been discovered in Delphix Continuous Data Engine each host's path can be adjusted. Supported JDKs include Adoptium Temurin OpenJDK, AIX, HPUX, and Oracle. If you have a Java license with Oracle then you can follow the instructions in this section.

- All Delphix environment users on that environment require read and execute permissions on the provided JDK, its subfolders, and files.
- Delphix only supports using custom JDKs for Oracle, Adoptium Temurin, AIX, and HPUX. All other JDKs are not supported.

9.1.4.7.2 Adding a JDK

By default, Delphix Continuous Data Engine comes with Adoptium's OpenJDK. To modify the JDK, follow the below steps:

1. Log in to the **Delphix management** application.
2. Click **Manage**.
3. Select **Environments**.
4. Click the **Actions (...)** menu next to **Environments and** select **Add environment.**
5. In the **Environment settings** tab, select the **Provide my own JDK** checkbox, and click **Next**.
   This action will remove the previous built-in JDK and will initiate an Environment refresh operation after the path is changed.



> Select Reset to Default to reset your JDK to the OpenJDK default.

6. In the **Java development kit** tab, the currently selected JDK kit will be shown (default is OpenJDK). Provide an absolute path to the JDK root and click **Next.**
7. Verify the configured JDK and environment settings and click **Submit.**

9.1.4.7.3 Updating a JDK path

For existing environments, you can add or update JDK paths by following the steps below:

1. Log in to the **Delphix management** application.
2. Select **Manage > Environments**.
3. Click on the name of an environment to view its basic information.
4. In the Details tab next to the **Java development kit**, click the **Pencil** icon.
5. In the **Java development kit dialog**, provide an absolute path to the JDK root and click **Next.**

> Do not place the JDK inside the Delphix Toolkit.

6. Click the **Check** icon to save your changes.

9.1.4.7.4 Finding a JDK

Custom JDKs need to be present on an environment which is being added to Delphix Continuous Data Engine and can be sourced directly from the supported vendors. These JDKs must be of Java major version 8 and come from one of the following:

- Oracle
- Adoptium Temurin OpenJDK
- HP-UX Java
- IBM Java for AIX

9.1.4.7.5 Supported operating systems

This feature is supported with use for Linux, Solaris, and Windows environments, as described in the support matrix. AIX and HP-UX provide us with a specific Java version to run on those hosts.

Only Java versions equal to or greater than U281 and equal to or lesser than U402 are tested and certified by Delphix based on below matrix.

	RHEL	SLES	Solaris	AIX	HP-UX	Windows
Oracle Java 8 u281	Supported	Supported	Supported	N/A	N/A	Supported
Oracle Java 8 u333	Supported	Supported	Supported	N/A	N/A	Supported
Oracle Java 8 u351*	Supported	Supported	Supported	N/A	N/A	Supported
Oracle Java 8 u381**	Supported	Supported	Supported	N/A	N/A	Supported
Oracle Java 8 u401***	Supported	Supported	Supported	N/A	N/A	Supported
AdoptOpenJDK 8 u382-b05	Supported	Supported	Supported	N/A	N/A	Supported
AdoptOpenJDK 8 u402-b06	Supported	Supported	Supported	N/A	N/A	Supported
AIX Java 8.0.0.821	N/A	N/A	N/A	Supported	N/A	N/A
HPUX Java 8.0.26	N/A	N/A	N/A	N/A	Supported	N/A

- * With Continuous Data Engine version 8.0.0.0 and above.
- ** With Continuous Data Engine version 15.0.0.0 and above.
- *** With Continuous Data Engine version 22.0.0.0 and above.

9.1.4.7.6 Delphix toolkit native Java support matrix

The below matrix describes the supported versions of Java we package with the Delphix toolkit for each operating system. This is the default option if you do not use the feature to provide your own Java for each host.

	RHEL	SLES	Solaris x64	Solaris sparc9	AIX	HP-UX	Windows
AdoptOpenJDK 8 u402-b06	Supported	Supported	Supported	Supported	N/A	N/A	Supported
IBM Java 8.0.821	N/A	N/A	N/A	N/A	Supported	N/A	N/A
HP Java 8.0.26	N/A	N/A	N/A	N/A	N/A	Supported	N/A

9.1.4.7.7 Java support policy

Delphix Continuous Data Engine is committed to testing and certifying the JDK versions that are included with the product, along with specific versions provided by Oracle. While using custom JDKs from Adoptium Temurin, AIX, and HPUX is generally supported, there is no certification process beyond those detailed in the native Java support matrix above Providing Your Own Oracle Java.

9.1.4.8 NFS encryptions

The Delphix Continuous Data Engine facilitates data communication with VDB target environments and staging environments via NFS and iSCSI. The system provides the flexibility to encrypt this communication as needed.

9.1.4.8.1 Feature limitations

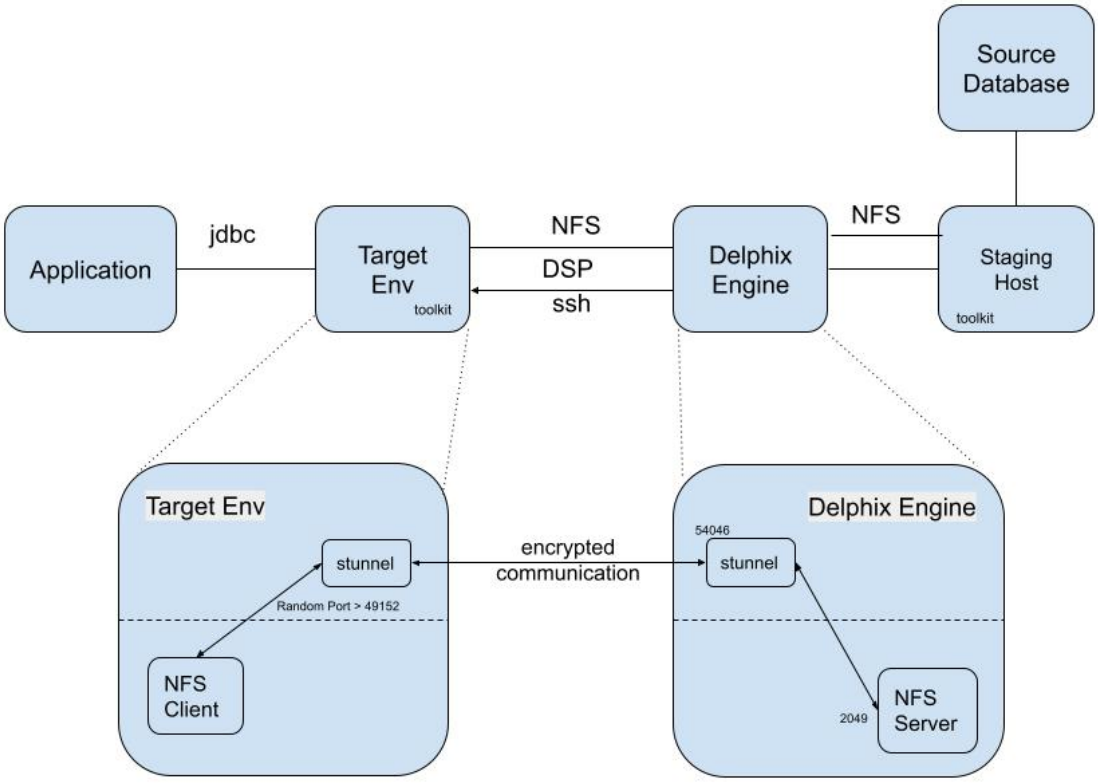
1. Supports only Linux-based host systems and Oracle clusters.
2. Windows systems and Windows clusters are not yet supported.
3. NFS encryption is not compatible with Oracle dNFS.
4. NFS encryption is not compatible with NFSv3.

9.1.4.8.2 Prerequisites

- The following operating systems are supported:
  - CentOS 7 and later
  - RHEL 7 and later
  - SUSE 12 and 15

- The designated network port for the specified feature in the Delphix Continuous Data Engine is 54046. The connection to this port is initiated from the host to the Delphix Continuous Data Engine, constituting an ingress connection from the perspective of the Delphix Continuous Data Engine. For more information on network connectivity requirements, refer to Network connectivity requirements[293].

9.1.4.8.3 Feature design



2 Image 1. NFS - Architecture

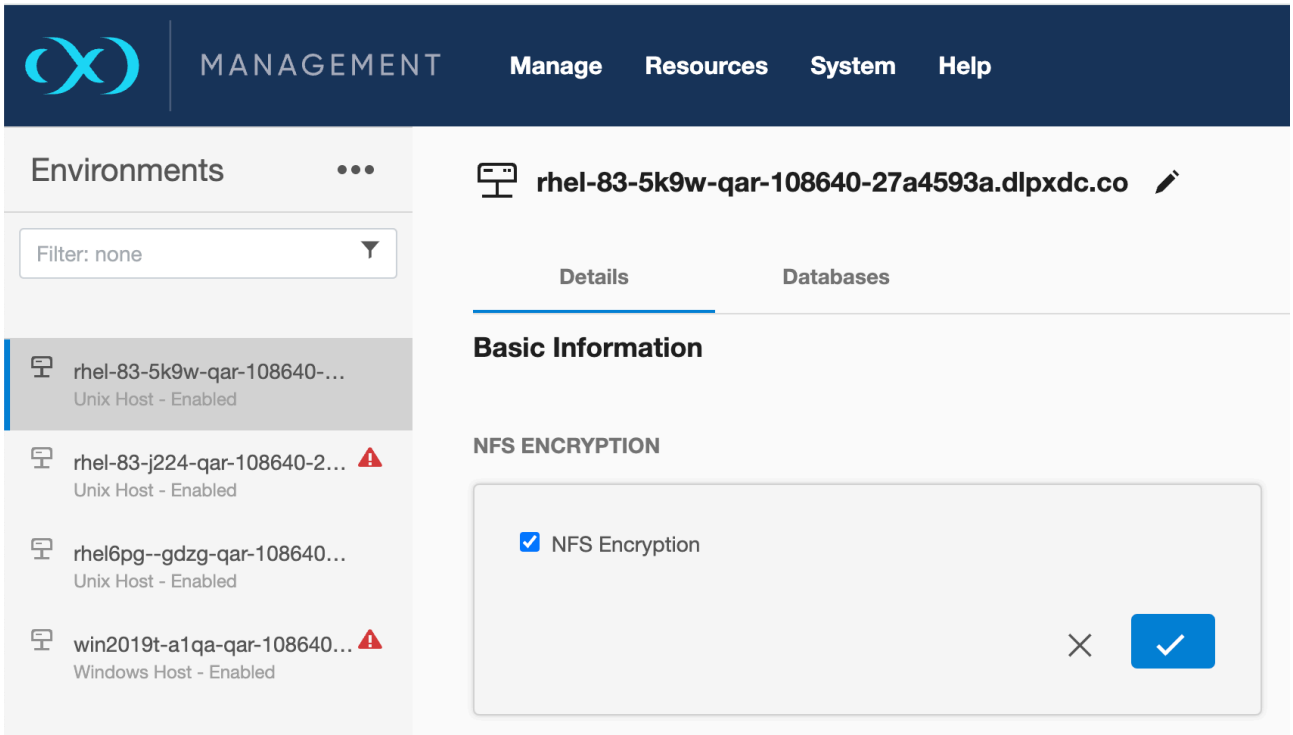
Encryption can be applied to secure data transfer between the Delphix Continuous Data Engine and the target VDB or staging source environment via NFS. This feature allows you the flexibility to enable or disable NFS communication encryption on a per-environment basis.

Stunnel is used for this purpose. Stunnel is a utility that provides TLS tunneling capabilities in user space. The implementation involves configuring two Stunnel services to establish a secure TLS connection: one acting as a server and the other as a client. The client is configured to listen on a local port, enabling secure tunneling of any connections made to that port to the remote peer Stunnel application running on the server.

293 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/71147588

9.1.4.8.4 Implementation

The NFS Encryption parameter can be activated for the specific environment either through the environment page or using the command-line interface (CLI). For more information about using the CLI, refer to the CLI[294] guide.

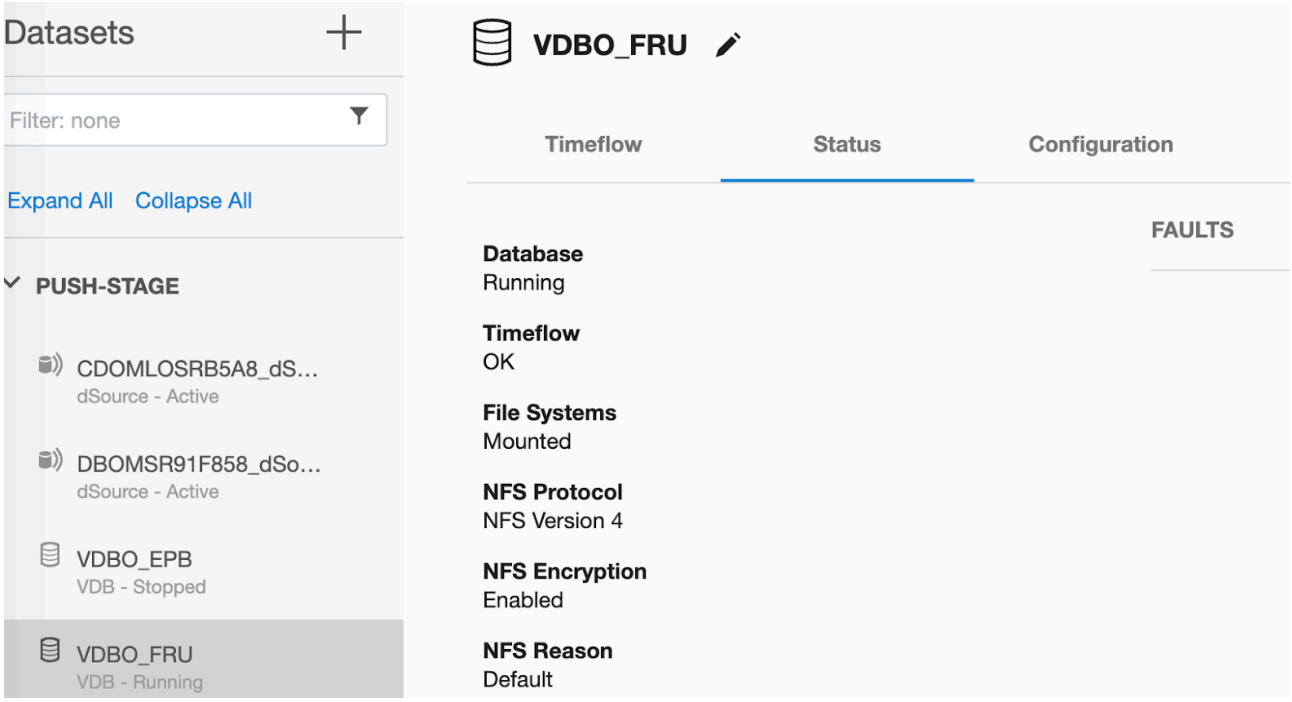


3 Image 2. Environment Page

Before activating encryption, it's essential to confirm that all datasets linked to the environment are in a disabled state.

Once environment encryption is enabled, you must enable all the relevant datasets. The encryption status of datasets can be viewed in its status tab. To disable or enable all the associated datasets, APIs are available at the environment level. For specific API details and changes, refer to API Changes in Delphix 17.0.0.0[295].

294 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/71074700
295 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/71145745



4 Image 3. Datasets Status Tab

9.1.4.8.5 Stunnel for NFS encryption (deployment and configuration)

Stunnel is employed for NFS encryption, as illustrated in the image for feature design above. The Stunnel executable and required resources are deployed to the host via the toolkit. A new working directory is created on the host and Stunnel functions from that location. A randomly generated port is employed to configure Stunnel on the host. Upon disabling this feature, the Stunnel process is terminated, and the working directory is removed from the host.

In cases where a host is connected to more than one Delphix Continuous Data Engine with NFS encryption enabled, it is possible to observe multiple Stunnel working directories on the host.

In the case of the Oracle cluster, the stunnel toolkit is set up, and the stunnel process is initiated on all active nodes when enabling this feature. Upon disabling the feature, the stunnel process is terminated, and the stunnel toolkit is uninstalled from all active nodes in the cluster.

Furthermore, when creating or enabling a node with this feature turned on for the cluster, the stunnel toolkit is configured, and the stunnel process is initiated on that specific node. If the feature is later turned off or the node is removed, the stunnel process is stopped, and the stunnel toolkit is removed from that individual node.

9.1.4.8.6 Certificate usage for Stunnel communication

On the setup page, a new certificate for the stunnel-server has been seamlessly integrated and this integration is fully automated, eliminating the need for any user intervention. The process for replacing this certificate is similar to the process of replacing another certificate on the setup page. For more details, refer to Customer provided key pair configuration[296].

296 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/71075999

Upon enabling NFS encryption for an environment, the stunnel working directory is established. Essential files are then transferred from the common toolkit directory to this newly created directory. Additionally, a required certificate is generated with a 730-day expiration period in real-time and is subsequently transferred to the host. This process facilitates the establishment of stunnel communication.

When you refresh the environment with NFS encryption enabled, a new certificate is generated and uploaded to the host. This process effectively extends the expiry, ensuring continued secure communication.

9.1.4.8.7 Configuring ciphersuites

The default ciphersuite used for stunnel communication is `TLS_CHACHA20_POLY1305_SHA256`. If needed, you can change it to one of the following options: "`TLS_AES_256_GCM_SHA384`", "`TLS_AES_128_GCM_SHA256`", "`TLS_AES_128_CCM_8_SHA256`", or "`TLS_AES_128_CCM_SHA256`" using the provided API.

Details	Path	Visibility	Availability
Env Encryption Ciphersuite	`GET /service/security/env_encryption`	System	CLI
Env Encryption Ciphersuite	`POST /service/security/env_encryption`	System	CLI

The end point `/service/nfs/encryption` is not available from this version.

When altering the ciphersuite used for Env Encryption, any changes to the ciphersuite automatically trigger adjustments in the transport layer. This process eliminates the need for additional modifications or manual service restarts.

9.1.4.8.8 Diagnostic data

The environment monitoring system is implemented to assess the operational status of active stunnel connections andtrigger the necessary alerts and fault notifications in case of any identified issues. However, assuming all configurations are in order, it can also restart the stunnel process if it unexpectedly stops.

Additionally, for diagnostic purposes, it should be noted that the NFS encrypted mounts will be associated with the loopback IP address (127.0.0.1), while non-encrypted mounts will use the Delphix Continuous Data Engine's IP.

9.1.4.8.9 API Usage

The following APIs are integrated for broad applicability, proving particularly advantageous when setting up NFS encryption in an environment.

Details	Path	Visibility	Availability
Enable all datasets for an environment	`POST /environment/<environment>/enableAllSources`	Admin	CLI
Disable all datasets for an environment	`POST /environment/<environment>/disableAllSources`	Admin	CLI

9.1.4.9 Environment operations

9.1.4.9.1 Adding an environment

Follow the steps below to add source, staging and target environments.

1. Login to the **Delphix management** application.
2. Click **Manage**.
3. Select **Environments**.
4. Click on the **Plus** icon next to **Environments**.
5. In the **Add environment** dialog, select **Unix/Linux**.
6. Select **Standalone host** or **Oracle cluster**, depending on the type of environment you are adding.
7. For a standalone environment enter the **Host IP** address.
8. For clustered environments, enter the **Node address** and **Cluster home.**
9. Enter an optional **Name** for the environment.
10. Enter the **SSH** port. The default value is **22**.
11. Enter a **Username** for the environment.
12. Select a Login Type:
    Username and Password - enter the OS username and password, or
    Username and Public Key - enter the OS username, and select Password Vault from an existing Enterprise Password Vault
    Info : For using Public Key Authentication
    If you want to use public-key encryption for logging into your Unix-based environment:
    a. Select **Public key** for the **Login type.**
    b. Click **View public key**.
    c. Copy the public key that is displayed, and append it to the end of your ~/.ssh/authorized_keys file. If this file does not exist, you will need to create it.

i. Run chmod 600 ~/.ssh/authorized_keys to allow only the file's owner to read and write to it (make sure the file is owned by the user).
ii. Run chmod 755 ~ to restrict access to the user's home directory so no other user may write to it.
iii. Run chmod 700 ~/.ssh so that others cannot write to it. The ~/.ssh directory cannot be writable by group or other users. Otherwise, authentication will fail.

13. The public key needs to be added only once per user and per environment.
14. For **Password login**, click **Verify credentials** to test the username and password.
15. Enter a **Toolkit path**. The toolkit directory stores scripts used for Delphix Continuous Data Engine operations, and should have a persistent working directory rather than a temporary one. The toolkit directory will have a separate subdirectory for each database instance. The toolkit path must have 0770 permissions and at least 345MB of free space.
16. Click **Submit**.

9.1.4.9.2 Refreshing an environment

After you make changes to an environment that you have already set up in the Delphix Management application, such as installing a new database home, creating a new database, or adding a new listener, the environment may need to be refreshed to reflect these changes.

During environment discovery and environment refreshes, Delphix Continuous Data Engine pushes a new copy of the toolkit to each staging and target environment and replaces the old toolkit with the newer one. The toolkit includes:

- A Java Runtime Environment (JRE)
- Delphix jar files
- The HostChecker utility
- Scripts for managing the environment and/or VDBs
- Delphix Connector log files

When you refresh the environment, it will push the toolkit back to the directory identified as the **Toolkit Path** for the given environment. Once this completes, the dSource can be brought back online.

Delphix Continuous Data Engine then executes some of these scripts to discover information about the objects in your environment (where the databases are installed, the database names, information required to connect to these databases, etc.). In some environments (Windows in particular), the scripts are customized to fit your environment.

An environment refresh or discovery operation does not alter the source configuration of manually added databases. If you have added the databases manually, then Delphix Continuous Data Engine does not update its source configuration upon discovering a change.

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Environments**.

4. In the Environments panel, click the name of the environment you want to refresh.
5. Select the **Refresh** icon.
6. In the Refresh confirmation dialog select **Refresh**.

9.1.4.9.3 Adding a dataset

While adding an environment to the Delphix Continuous Data Engine, all database installation homes on it are automatically discovered. However, if a database installation home is not automatically discovered, you can add it manually to the environment. This feature may vary for different connectors, for workflow related information, refer to the connector specific documentation.

Below is an example of how to add a dataset home for Oracle.

1. Login to the **Delphix management** application.
2. Click **Manage**.
3. Select **Environments**.
4. Select an Environment.
5. Click the **Databases** tab**.**
6. Click the **Add dataset home** button.
7. Enter the **Installation home**.
8. Enter the **Version** of the Installation Home.
9. Enter the **Oracle base** of the Installation Home.
10. Enter the **Bits** of the Oracle Home.
11. When finished, click **Add**.

9.1.4.9.4 Adding a database

An environment contains installations, that are database installations in the environment. Each environment may have any number of installations associated with it.

A database object defines the configuration of the dSource, sometimes called a Source Config or Instance, and is required to create a dSource. You can create any number of database objects using an installation, which represents known database instances. Databases only need to be identified on source or staging environments, not target environments.

For Oracle, SQL Server, and some other connectors, known databases are automatically added during discovery. However, not all support automatic database discovery, such as PostgreSQL, or databases can be missed by the discovery feature. In these scenarios, you must create the database manually.

Follow the steps below to create a database, or Source Config, on a source or staging environment:

1. Login to the **Delphix Management** application.
2. Click **Manage**.

3. Select **Environments**.
4. Select one of the configured source or staging environments.
5. Select the **Databases** tab.
6. Select one of the **Dataset Home** installations, in some connectors there will be no installation but a placeholder for installation.
7. Select the **Add Database** plus (+) button to specify the database configuration. Input fields for database installation may vary connector to connector. To know about the exact input fields you can refer to the connector specific documentation.

9.1.4.9.5 Adding a database using the CLI

The equivalent of adding a database in the CLI is creating a new `sourceconfig` object. Follow the directions below on how to do so.

Login to Delphix Continuous Data Engine as an admin user:

```
ssh admin@<CONTINUOUS-DATA-FQDN-or-IP>
```

Execute the commands below:

```
sourceconfig
create
set type=<AppDataStagedSourceConfig>
set databaseName=<dbname>
set repository="<repository name>"
set parameters="{\"name\":\"<source config name>\"}"
commit
```

9.1.4.9.6 Deleting an environment

Deleting an environment in Delphix Continuous Data will remove all data and configurations related to that environment. This action only affects the environment metadata stored in Delphix Continuous Data. It will not affect your database installations on the hosts or clusters that the environment is referencing.

Before deleting an environment, you must first delete all dependencies, such as dSources and virtual databases (VDBs)

1. Login to the **Delphix Management** application.
2. Select **Environments**.
3. In the Environments panel, select the environment you want to delete.
4. Click the Actions (...) menu, and select **Delete**.
5. In the Delete Environment confirmation dialog click **Delete**.

9.1.4.10 Validating host environments with HostChecker

9.1.4.10.1 Overview

Delphix Continuous Data Engine has developed a host checker for connectors utility that contains standardized checks for source, staging, and target hosts. These checks generally fall into two buckets:

- OS and Host permissions/access
- DB-specific functionality

Currently, the host checker utility is implemented for two types of data sources:

1. IBM DB2
2. Oracle EBS

OS and host permissions/access can (and should) be performed before adding the environment to Delphix Continuous Data Engine to ensure smooth deployment and virtualization workflows.

9.1.4.10.2 Procedure

1. Download the HostChecker tarball for the engine from https://download.delphix.com/ located in the **Delphix Product Releases > Continuous Data Connectors > Utilities > Host Checker Tool** directory.
2. Extract the HostChecker files from the HostChecker tarball.

```
[ dlpxqa@stagehost:~ ]$ tar -xvf hostchecker-for-connectors-1.1.0.tar
```

3. Grant execution permissions and run the shell script `main.sh` to initiate the Host checker utility.

```
[ dlpxqa@stagehost:~/hostchecker-for-connectors-1.1.0 ]$ chmod +x main.sh
[ dlpxqa@stagehost:~/hostchecker-for-connectors-1.1.0 ]$ ./main.sh
```

> Do not run the HostChecker as root. This will cause misleading or incorrect results from many of the checks.

4. Since Host checker is designed to run on different types of databases such as IBM Db2 and Oracle EBS, the user needs to select the type of dataset before starting the host checker.

```
[ dlpxqa@stagehost:~/hostchecker-for-connectors-1.1.0 ]$ ./main.sh
==============================================================================
```

```
                                        Delphix  Host Checker  Utility


================================================================================
Please select the type of dataset you want to run the host checker on:

1. DB2
2. Oracle EBS
Enter your choice (Default: 1): 1

Dataset type selected: DB2

--------------------------------------------------------------------------------
Starting host checker for DB2
--------------------------------------------------------------------------------
```

5. The utility prints the output of the various checks that it performs and also logs the details in a specific log directory under the utility's root directory. For example, IBM Db2 dataset environment logs are written under the `DB2-hostchecker-logs` directory.
6. When the script is running, the discovered issues or conflicts (text colored red) are always accompanied by the corresponding documentation URL for the customers to verify and fix the issues raised by the utility. However, if there are no accompanied documentation URLs, please forward them to Delphix Support along with the logs.
7. Repeat steps 3-7 until all the checks return no errors or warnings.

On successful completion, the scanned host should be added to Delphix Continuous Data and be used as a staging or target environment for your ingesting or provisioning data set.

9.1.4.11 Manually recovering a database after V2P

This page describes how to manually recover a database after the V2P process.

> For the Oracle database, while running V2P, Delphix copies the datafiles and other required files on the physical storage. After that it performs the database recovery, followed by some post recovery steps and openDatabase.
>
> - While running V2P from CLI, you can set `recoverDatabase=false` and `openDatabase=false`.
> - While running V2P from UI, you can select not to openDatabase.
> - If `recoverDatabase=false` is set, run both `recover-vdb.sh` and `open-vdb.sh`.
> - If the V2P was performed with `recoverDatabase=true` and `openDatabase=false`, only run `open-vdb.sh`.

9.1.4.11.1 Procedure

1. In the V2P target environment, navigate to the scripts directory for the exported database instance. You can find the scripts in a sub-directory named for that specific database instance. For Oracle databases, the path is `<target_directory>/<db_unique_name>/script/<instance name>`. For SQL Server databases, the path is `<target_directory>\<db_name>\scripts`.
2. For **Oracle** databases, locate the `recover-vdb.sh` and `open-vdb.sh` scripts, then run the required script(s) as described above (as `sh ./recover-vdb.sh` and `sh ./open-vdb.sh`). For **SQL server** databases, locate the script `Provision.ps1` and run it.
3. For **Oracle** databases, add the recovered database to /etc/oratab and **Refresh** the target environment for it to discover the recovered database. For **SQL server** databases, when the script completes, **Refresh** the target environment for it to discover the recovered database.

9.1.5 dSource management

9.1.5.1 Overview

dSource is the copy of a source database's persistent data layer that Delphix Continuous Data uses to create and update virtual databases (VDBs). Based on the ingestion model and data source type, the dSource could be exposed through a mount point and interacts with a Staging Database Instance, Database, or Files. Consult the data source connector documentation for specific details.

9.1.5.2 Linking dSources

Once you have discovered your data source environments, you will see a list of data sources on that host. The first step to sync data is to link a dSource from the host.

Linking a dSource will ingest data from the source and create a dSource object on the Delphix Continuous Data engine. As mentioned, the dSource is an object used to create and update virtual copies of your database. As a virtualized representation of your source data, it cannot be managed, manipulated, or examined by database tools.

Once a dSource has been linked, you can begin to perform dSource operations on it, such as enable, detach, delete, and more.

9.1.5.3 The dSource timeflow

Both dSources and VDBs have timeflows, which describes the timeline of data of a virtual database or dSource. Virtual databases can be provisioned from any snapshot in a Timeflow. In the UI, a Timeflow is represented vertically with a series of individual snapshots as rows.

9.1.5.4 The dSource snapshot

Snapshots of dSources preserve specific points in time for use later during the virtual database provisioning workflow. Taking a snapshot will create a new snapshot entry in the dSource's Timeflow.

9.1.5.5 dSource operations

9.1.5.5.1 Overview

In order to use dSource snapshots, you must create a virtual database (VDB), an independent, writable copy of a dSource snapshot.

When integrating an environment into the Continuous Data Engine, the system automatically locates and identifies any data sources within the host, listing them as dSources under the Databases tab specific for each environment. This section outlines the management of dSources and the synchronization of data from your sources, involving operations and concepts such as:

- **Link/Sync**: Establishing a connection with a data source.
- **Timeflows**: Describes the timeline of data of a virtual database or dSource.
- **Snapshots**: Captured states of a dSource at specific moments.
- **SnapSync and LogSync**: Processes to capture and synchronize data changes.
- **Enable/Disable**: Activating or deactivating a dSource.
- **Detach/Reattach**: Temporarily disconnecting and then reconnecting a dSource.
- **Delete**: Removing a dSource.
- **Upgrade**: Updating a dSource to a new version.

9.1.5.5.2 Adding an environment configuration

In the Delphix Continuous Data Engine, environments are composed of repositories, which correspond with your respective database management software (DBMS) installations. An environment can be linked with an unlimited number of these repositories.

To establish a dSource, you must define its configuration with a source environment configuration. Repositories serve as a foundation for creating this kind of object, each of which corresponds to a recognized database instance within the system.

For connectors, it is necessary to set up the environment configuration manually.

9.1.5.5.2.1 Steps to create an environment configuration in the staging environment

1. Access the Delphix Management application by logging in.
2. Navigate to the **Manage** section.
3. Go to **Environments** and select the appropriate repository.
4. Click the plus (**+**) icon to initiate a new configuration.
5. In the **Add database** area, fill in the necessary details.

6. Provide a name for your dSource in the **Name** field.

9.1.5.5.2.2 Setting up an environment configuration using the Command Line Interface (CLI)

To configure SourceConfig via CLI, follow these instructions:

1. Securely log into your Delphix Continuous Data Engine using the admin account with SSH:

```
ssh admin@<CONTINUOUS-DATA-FQDN-or-IP>
```

2. Execute the following commands:

```
sourceconfig
create
set type=<AppDataStagedSourceConfig>
set databaseName=<dbname>
set repository="<repository name>"
set parameters="{\"name\":\"<source config name>\"}"
commit
```

This process will create a SourceConfig for your dSource using the specified repository details.

9.1.5.5.3 Disabling a dSource

To ensure certain operations can be carried out, a dSource may need to be disabled. Disabling halts communication between the Delphix Continuous Data Engine and the source database without removing the underlying configuration.

9.1.5.5.3.1 Steps to disable a dSource

1. Log in to the Delphix Management application.
2. Select the dSource you wish to disable.
3. In the upper right-hand corner, click the Actions menu (...) and choose **Disable**.
4. In the confirmation dialog, click **Disable** again.

> Disabling is a prerequisite for operations such as:
> - Database migration
> - dSource metadata upgrading post-data source upgrade
> - Restoring the source database from backup

Disabling will cease any ongoing Delphix Continuous Data Engine operations related to the dSource. When you are ready to re-enable the dSource, simply select **Enable** from the Actions menu (...), and it will resume normal function.

9.1.5.5.4 Force disabling a dSource

There might be circumstances that require a dSource to be forcefully disabled, stopping all related Delphix Continuous Data Engine operations immediately.

9.1.5.5.4.1 Steps to force disable a dSource

1. Access the Delphix Management application.
2. Choose the dSource in question.
3. Click on the Actions menu (...) in the upper right corner and opt for **Disable**.
4. Select the **Force Disable** checkbox in the dialog that appears.

9.1.5.5.5 Detach and reattach a dSource

In certain scenarios, you may need to detach a dSource from its source database and later reattach it, possibly to a different source. This process is essential for maintaining the flexibility of your data management within the Delphix Continuous Data Engine.

9.1.5.5.5.1 Understanding dSource detachment

Detaching a dSource breaks its link with the current source database but preserves the existing data and snapshots within the Delphix Continuous Data Engine. Here are some items to note about detached dSources:

- **Continued use for VDBs**: A detached dSource can still provision a Virtual Database (VDB).
- **Re-linking capability**: You can re-link the original source database as a new dSource.
- **Child VDB refresh limitations**: VDBs that were created from the detached dSource can only refresh from the last snapshot taken before detachment.
- **Provisioning from new snapshots**: If a newer snapshot is needed, you will have to provision a new VDB. Subsequently, you may delete the outdated VDBs and the old dSource when they are no longer required.

9.1.5.5.5.2 Steps to detach a dSource

To detach a dSource from its source database:

1. Log into the Delphix Management application.
2. Select the dSource you wish to detach.
3. Click on the Actions menu (...) and choose **Unlink dSource**.
4. A warning message will pop up—confirm the action by clicking **Unlink**.

5. The status of the dSource will update to **Detached**.

9.1.5.5.5.3 Steps to reattach a dSource

When ready to re-establish a connection between a dSource and a source database:

1. Go back to the Delphix Management application.
2. From the Actions menu (...) next to the detached dSource, select **Link dSource**.
3. Follow the prompts to complete the re-linking process.

Remember, reattaching a dSource re-establishes its connection with a source database, allowing it to resume its role in data management operations according to your configurations.

9.1.5.5.6 Delete a dSource

Deleting a dSource will remove its metadata, snapshots, logs, and policies from the Delphix Continuous Data Engine. This action does not affect the source database.

9.1.5.5.6.1 Prerequisites for deleting a dSource

There must be no dependent virtual databases (VDBs) or virtual Pluggable databases (vPDBs). Ensure all dependent databases are deleted as per instructions in the **Deleting virtual databases** under Provisioning and managing virtual databases[297].

9.1.5.5.6.2 Steps to delete a dSource

1. Log in to the Delphix Management application.
2. Select the dSource you wish to delete.
3. From the Actions menu (...), select **Delete**.
4. Click **Delete** to confirm the action.

9.1.5.5.7 SnapSync a dSource

SnapSync is the process that captures the complete dataset on the initial load and only the incremental changes thereafter, creating efficient storage snapshots.

9.1.5.5.7.1 Understanding SnapSync

- The initial SnapSync pulls the complete data set using standard database protocols.
- Subsequent SnapSync operations fetch only incremental changes.
- Each SnapSync operation results in a snapshot, serving as a baseline for VDB provisioning.

297 https://cd.delphix.com/docs/latest/provisioning-and-managing-virtual-databases#Deleting-virtual-databases

9.1.5.5.8 Resync a dSource

Resync is a process similar to the initial SnapSync but is executed at intervals to ensure the dSource is updated with the complete data set from the source database.

9.1.5.5.8.1 Prerequisites for Resync

- The data directory under the mount path must be empty before starting a Resync operation.

9.1.5.5.8.2 Steps to Resync a dSource

1. If needed, ensure the data directory is empty to avoid errors.
2. The Resync will capture the complete data set, akin to the initial SnapSync process.

9.1.5.5.9 LogSync a dSource

LogSync complements SnapSync by periodically capturing log files from the source database, allowing for more granular recovery points between SnapSync snapshots.

9.1.5.5.9.1 LogSync details

- Log files are transferred using standard database protocols and stored separately from SnapSync data.
- These logs facilitate the provisioning of VDBs from points in time between SnapSync snapshots.
- The time taken to provision from these intermediate points is proportional to the volume of data change on the source database and the time elapsed since the last snapshot.

9.1.5.5.10 Upgrading a dSource

When the source database connected to a dSource is upgraded, it becomes necessary to update the dSource's metadata to match this new version. This section guides you through upgrading the dSource to align with the updated database.

The upgrade of a dSource is a crucial operation that ensures the Delphix Continuous Data Engine recognizes and correctly associates with the updated version of the source database. Note that this process does not upgrade the database itself; it merely updates the dSource metadata.

9.1.5.5.10.1 Prerequisites for upgrading a dSource

Before you begin the upgrade process:

- Confirm that the source database instance has been upgraded to the newer version.
- Ensure that the dSource is disabled. This is a mandatory step to proceed with the upgrade.

9.1.5.5.10.2 Upgrade procedure

The upgrade process will involve modifying the dSource to use the installation path of the new database version. The general steps are as follows:

1. Disable the dSource. This step is crucial and must be completed before you start the upgrade process. For details on disabling a dSource, refer to the **Disabling a dSource** section above.
2. Once the dSource is disabled, you can proceed with the upgrade according to the specific database you are working with.

9.1.5.5.10.3 Database-specific upgrade instructions

Each database has its own set of instructions for upgrading a dSource. Refer to the respective connector documentation for detailed procedures:

9.1.6 Virtual database (VDB) management

9.1.6.1 Overview

Virtual Database (VDB) is a full read-and-write copy of the source data that is provisioned from either a dSource or another VDB. A VDB is provisioned and managed by the Delphix Continuous Data Engine. Upon successfully adding a data source from an environment and generating snapshots of a dSource[298], you can initiate the provisioning of virtual databases (VDBs).

A dSource is a protected, virtualized representation of a source database which Delphix Continuous Data Engine maintains. It cannot be managed, manipulated, or examined by database tools. It is used to create and update virtual databases. To make use of the dSource snapshots, it is imperative to create a VDB.

It is important to note that the VDBs can also be created from existing VDBs. Once a VDB has been provisioned on the target environment, you have the ability to create a snapshot policy for that VDB. This policy enables you to capture and manage changes within the VDB in the same manner as any other conventional logical or physical database.

9.1.6.2 VDB Timeflow

Both dSources and VDBs have timeflows, which describes the timeline of data of a virtual database or dSource. Virtual databases can be provisioned from any snapshot in a timeflow. The virtual database's filesystem will be dependent upon the snapshot used to provision, and as such, that snapshot can not be removed until the dependency is removed. In the Delphix Continuous Data Engine UI, a timeflow is represented vertically with a series of individual snapshots as rows.

298 https://docs.google.com/document/d/1ZglUP1al5iZy-BN3dvhov0InOgfVR8U6sY5htaHGvqE/edit#heading=h.xl1ogq93hrpw

9.1.6.3 VDB Snapshot

Snapshots of dSources preserve specific points in time for use later during the virtual database provisioning workflow. Taking a snapshot will create a new snapshot entry in the VDBs timeflow.

9.1.6.4 VDB operations

VDB operations allows you to interact with and manipulate virtualized instances of databases. These operations are pivotal in ensuring the effective utilization, maintenance, and synchronization of virtual databases. Common VDB operations include:

9.1.6.4.1 Provision a VDB

Virtual databases are a significant part of the Delphix Continuous Data solution. In order to create or provision a virtual database, you will need a linked dSource from a source host and a compatible target environment, as described in the overview for Managing environments and hosts[299]. Since provisioning VDBs vary significantly across connectors, details on this process can be found within the documentation for the specific connector.

At the core, virtual databases are fully functional copies of data sources that can be created and managed at a fraction of the storage and time that are typically required. To end-users, VDBs act and perform just like a standard database.

9.1.6.4.2 Disable and enable a VDB

The act of disabling a VDB serves as a necessary prerequisite for procedures like VDB migration or upgrade. When a VDB is disabled, it consists of the complete removal of all its traces, including configuration files from the target environment in which it was originally provisioned. It's important to note that while the VDB itself, along with its associated metadata, remains intact on the Delphix Continuous Data engine, the configuration files are temporarily eliminated from the target environment.

Subsequently, when you decide to re-enable the VDB, these previously removed configuration files are reinstated within the target environment, facilitating the VDB's return to operational status with its settings and parameters restored.

9.1.6.4.2.1 Procedure

Perform the below steps to disable a VDB.

1. Login to the Delphix Management application.
2. Select the VDB you want to disable.
3. From the Actions menu, select Disable.
4. Click Disable to acknowledge the warning.

299 https://cd.delphix.com/docs/latest/managing-environments-and-hosts

When you are ready to re-enable the VDB again, select Enable from the Actions menu, and the VDB will continue to function as it did previously.

9.1.6.4.3 Start and stop a VDB

Stopping your virtual databases is essential when executing actions on the database that requires it to be in an offline state. Beyond simply ensuring the database's proper shutdown, the Delphix Continuous Data engine stop and start actions encompass additional functions, including unmounting and mounting the filesystems supplied by the Delphix Continuous Data engine.

9.1.6.4.3.1 **Procedure**

Perform the below steps to start/stop a VDB.

1. Login to the Delphix Management application.
2. Select the VDB you want to manage.
3. To stop an active VDB, click the stop icon at the top right-hand corner. Conversely, to start an inactive VDB, click the start icon at the top right-hand corner.
4. In cases where stopping or starting the VDB requires credentials distinct from the default environment user, then do the following:
    a. Check Provide Privileged Credentials.
    b. Enter the username and password.
    c. Click Validate Credentials.
5. Click Yes to stop or start the VDB.
6. To view the status of the stopping or starting of the VDB, you can use one of the below ways:
    a. View the Active Jobs on the right-hand corner of the screen
    b. monitor the progress bar from the VDB Status tab.

9.1.6.4.4 Refresh a VDB

Refreshing a VDB will re-provision it from the dSource. As with the normal provisioning process, you can choose to refresh the VDB from a snapshot or a specific point in time. However, you should be aware that refreshing a VDB will delete any changes that have been made to it over time. When you refresh a VDB, it will revert to its previous state in the specified snapshot or point in time.

When performing a VDB refresh, there is an option to choose between either the faster or more accurate point in time for that database.

Note: Although the VDB no longer contains the previous contents, the previous snapshots and timeflow still remain in the Delphix Continuous Data engine and are accessible through the Command Line Interface (CLI).

9.1.6.4.4.1 Prerequisites

To refresh a VDB, you must have the following permissions:

- **Provisioner** permissions on the dSource associated with the VDB

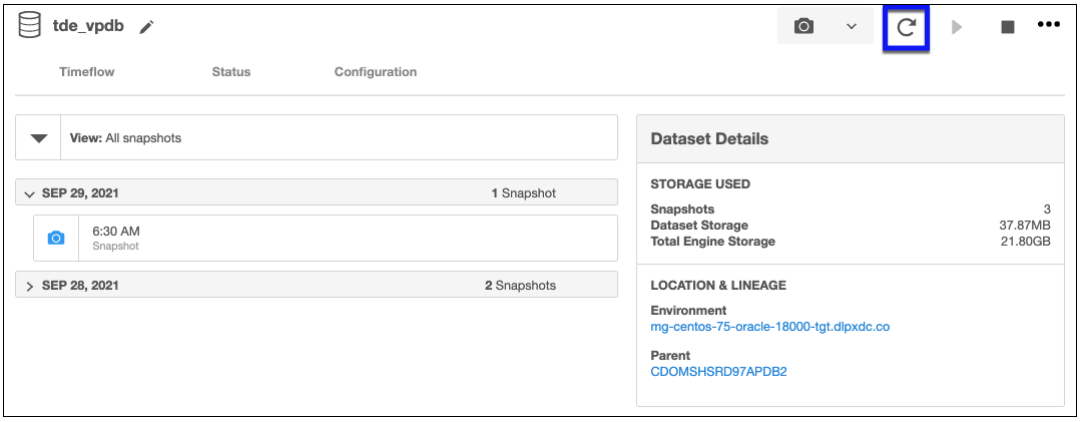
- **Provisioner** permissions on the group that contains the VDB
- **Owner** permissions on the VDB itself
- **Data** is a role that allows DB_ROLLBACK, DB_REFRESH, READ_ACTION, DB_SYNC, JOB_CANCEL.
- **Read** is a role that allows the user to inspect objects via the READ_ACTION permission.

A user with admin credentials can perform a VDB Refresh on any VDB in the system.

9.1.6.4.4.2 Procedure

Perform the following below to refresh a VDB.

1. Login to the Delphix Management application.
2. Navigate to **Manage** > **Datasets**.
3. Select the VDB that you want to refresh.
4. Click the **Refresh** VDB button. The Refresh VDB screen appears.
5. Select one of the refresh options and click **Next**.
   a. **Faster** - This option will utilize the most recent snapshot in the timeflow.
   b. **More accurate** - This option allows you to select a snapshot, a point in time, or a specific log ID.
6. Click **Submit** to confirm.
7. Click the **Actions** button to watch the progress of the refresh job.
8. If you want to know when the VDB was last refreshed/provisioned, check the **Time Point** on the Status page.



9.1.6.4.5 Rewind a VDB

Rewinding a VDB rolls it back to a previous point in its timeflow and re-provisions the VDB. The VDB will no longer contain changes after the rewind point.

- Although the VDB no longer contains changes after the rewind point, the rolled-over snapshots and timeflow still remain in Delphix Continuous Data engine and are accessible through the Command Line Interface (CLI). For instructions on how to use these snapshots to refresh a VDB to one of its later states after it has been rewound, refer to the topic CLI Cookbook: Rolling forward a VDB[300].
- Delphix clones a new timeflow from the closest snapshot older than or equal to the rewind point. This creates a dependency between the new timeflow and the parent snapshot and timeflow. The parent snapshot and timeflow cannot be deleted because of this dependency. The VDB must first be refreshed before the parent snapshot and timeflow can be removed.

9.1.6.4.5.1 Prerequisites

To rewind a VDB, you must have the following permissions:

- **Auditor** permissions on the dSource associated with the VDB
- **Owner** permissions on the VDB itself

You do not need owner permissions for the group that contains the VDB. A user with admin credentials can perform a VDB Rewind on any VDB in the system.

9.1.6.4.5.2 Procedure

Perform the below steps to rewind a virtual database.

1. Login to the Delphix Management application.
2. Select the VDB you want to rewind.
3. Select the **Timeflow** tab.
4. Select the rewind point as a snapshot or a point in time.
5. Click **Rewind**.
6. If you want to use login credentials on the target environment other than those associated with the environment user, click **Provide Privileged Credentials**.
7. Click **Rewind** to confirm.

You can use timeflow bookmarks as the rewind point when using the CLI. Bookmarks can be useful to:

- Mark where to rewind to - before starting a batch job on a VDB for example.

300 https://cd.delphix.com/docs/latest/cli-cookbook-rolling-forward-a-vdb

- Provide a semantic point to revert back to in case the chosen rewind point turns out to be incorrect.

For a CLI example using a timeflow bookmark, refer to the [CLI Cookbook: Provisioning a VDB from a Timeflow Bookmark](301).

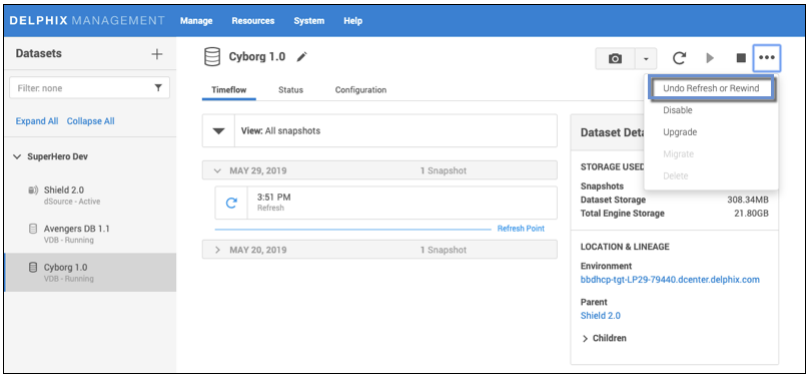
Undo rewind or refresh operation

An accidental refresh of one or more virtual databases (VDBs) from a source database can remove the important data. To restore the data back, you can undo VDB refresh or rewind actions.

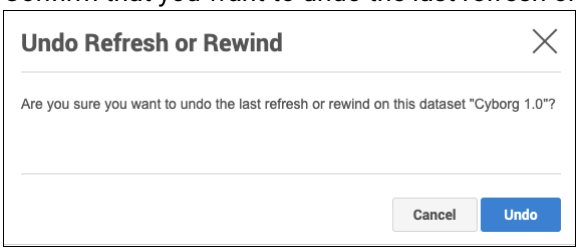
Procedure

Perform the below steps to undo a refresh or rewind operation:

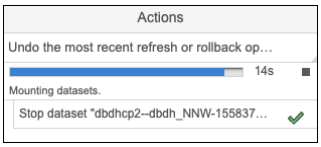
1. Login to the Delphix Management application.
2. Under Datasets, select the VDB you want to undo your rewind/refresh for.
3. From the **Actions** menu, select **Undo Refresh** or **Rewind**.



4. Confirm that you want to undo the last refresh or rewind for the selected VDB, by selecting **Undo**.



5. From the Action sidebar menu, you will be able to view the undo action.



301 https://cd.delphix.com/docs/latest/cli-cookbook-provisioning-a-vdb-from-a-timeflow-bo

9.1.6.4.6 Migrate a VDB

In certain scenarios, you might find it necessary to migrate a virtual database to a different target environment. This need can arise, for instance, when upgrading the host where the VDB is hosted or as part of a broader data center migration effort. Leveraging the VDB migration functionality enables you to relocate the VDB from its current location to a new target environment. This is easily accomplished by first disabling the database, then using the migrate VDB feature to select a new target environment.

9.1.6.4.6.1 Prerequisites

- Set up a new target environment that is compatible with the VDB that you want to migrate.
- Ensure that the VDB has first been disabled.

9.1.6.4.6.2 Procedure

Perform the below steps to migrate a virtual database.

1. Login to the Delphix Management application.
2. Select the VDB you want to migrate.
3. From the Actions menu, select **Migrate**.
4. Select the new target environment for the VDB, the user for that environment, and the database installation where the VDB will reside.
5. Select **Migrate**.
6. From the **Actions** menu, select **Enable**.
7. Click **Enable** to confirm.

Within a few minutes, your VDB will re-start in the new environment, and you can continue to work with it as you would any other VDB.

> This feature may vary for different connectors. For workflow related information, refer to the connector specific documentation.

9.1.6.4.7 Upgrade a VDB

Much like the process of upgrading a data source, upgrading a VDB entails modifying the installation path linked to the corresponding data platform. It's important to note that this feature doesn't upgrade the VDB itself; instead, it ensures that Delphix Continuous Data engine correctly associates the VDB with the correct database version to ensure proper functionality.

By performing this upgrade, you have the capability to update the database version used for running the VDB.

> This feature may vary for different connectors. For workflow related information, refer to the connector specific documentation.

9.1.6.4.7.1 Procedure for VDB In-Place Upgrade

Perform the below steps to upgrade a virtual database.

1. Remove any VDB refresh policy assigned to the VDB.
2. Upgrade the target database instance.
3. Refresh the target environment.

9.1.6.4.8 Delete a VDB

Deleting a VDB will remove it and its timeflow and snapshots from the Delphix Continuous Data engine entirely. This operation is non-reversible.

9.1.6.4.8.1 **Prerequisites**

You cannot delete a VDB that has dependent VDBs. Before deleting a VDBs, make sure that you have deleted all dependent VDBs. Otherwise, the delete option will be grayed out.

> Deleting a VDB is an unrecoverable operation. Proceed only if you want to permanently destroy the unique data that was created in the VDB.

9.1.6.4.8.2 **Procedure**

Perform the below steps to delete a virtual database.

1. Login to the Delphix Management application.
2. Select the VDB you want to delete.
3. From the **Actions** menu, select **Delete**.
4. Click **Delete** to confirm.

9.1.6.4.8.3 **Force Delete**

Deleting a vFiles may fail if it cannot be unmounted successfully from all target environments. You can use the force delete option to ignore all failures during unmount.

9.1.6.4.8.4 Using Force Delete

Deleting unused or outdated objects should be a regular part of Delphix Continuous Data engine administration. This is especially important to prevent low space errors, which can cause the Delphix Continuous Data Engine to stop. The Delphix Continuous Data engine holds a maximum of 400 objects.

Force Delete can be used when:

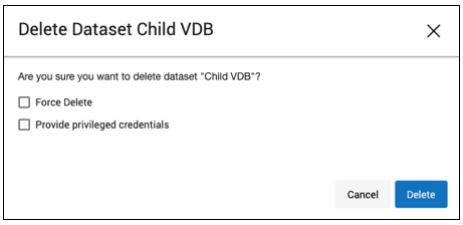
- The staging host/container is not accessible via Delphix Continuous Data engine anymore.
- The Delphix Connector service is not running (applicable to SQL Server only).
- The normal delete is throwing an error and not working.
- When the target or staging host is decommissioned.

> Force Delete should not be used as an alternative to **Delete** in normal circumstances.

Procedure

Perform the below steps to forcefully delete a virtual database.

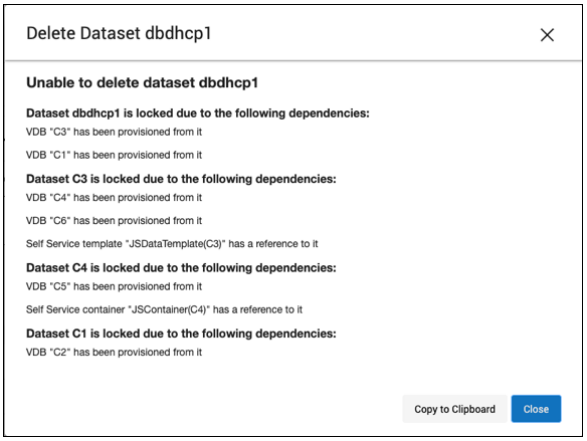
1. Log into the Delphix Management application.
2. Select **Resources** > **Storage Capacity**.
3. Select the Trash can next to the object you want to delete.
4. In the **Delete** dialog, select **Force Delete**. Oracle users will have the option to provide additional credentials.



5. Click **Delete**.

9.1.6.4.8.5 Dependencies

If there are dependencies on the snapshot, you will not be able to delete the snapshot free space; the dependencies rely on the data associated with the snapshot. These items are displayed with a lock icon next to the name.



9.1.6.4.9 Export a VDB (V2P)

V2P (Virtual to Physical) refers to the process of exporting a virtual database to a physical one. Once you have created a dSource or a VDB, you have the option to export its contents and log files to a physical database. This process creates a set of directories on the target environment and populates them with the database data, log files, and scripts that are used to recover the physical database.

You can automatically start the physical database recovery process as part of V2P, or you can use the scripts for manual recovery. When the export process completes, the target environment will contain a copy of the database in its unvirtualized size. Therefore, before you begin the process, ensure that the target directories specified during V2P operation have enough capacity to accommodate the database. For more information, refer to the connector specific documentation.

9.1.6.4.9.1 Provisioning with replication

The process for provisioning from replicated objects closely mirrors the standard VDB provisioning procedure, with one key distinction. Before initiating the provisioning process, you must first select the replica namespace containing the replicated object.

9.1.6.4.9.2 Prerequisites

- You must have a dSource or VDB that has been replicated from one Delphix Continuous Data engine to another, as described in Replication overview[302].
- The Delphix Continuous Data engine containing the replicated dSource or VDB must have a compatible target environment that it can use to provision a VDB from the replicated dSource or VDB.
- On the Delphix Continuous Data engine containing the replicated dSource or VDB, login to the Delphix Management application.

9.1.6.4.9.3 Procedure

Perform the below steps to perform provisioning from replicated objects .

302 https://cd.delphix.com/docs/latest/replication-overview

1. From the list of replica namespaces, select the replica namespace that contains the dSource or VDB from which you want to provision.
2. The provisioning process is now identical to the process for provisioning standard objects.

9.1.7 Hook operations

9.1.7.1 Overview

Hook operations enable the execution of custom tasks at specific points in the process of linking sources and managing virtual datasets. This feature is particularly useful in scenarios such as preventing monitoring systems from activating during the startup and shutdown of Virtual Databases (VDBs). Users can utilize pre and post-hooks to execute necessary scripts during VDB start/stop operations.

The execution environment for hook scripts varies depending on the operating system:

- **Unix-based** OS: Hook scripts will be executed using **bash**.
- **Windows** OS: Hook scripts will be executed using **PowerShell**.

Each Delphix Continuous Data object possesses a unique set of hooks. For efficient management across multiple datasets, you have the option to create hook templates. These templates allow you to apply the same hook configuration to numerous objects, streamlining the process (as discussed in the corresponding section later on this page).

9.1.7.2 Creating hook operations

9.1.7.2.1 Creating hook operations via the Delphix Continuous Data Engine UI

Hook operations can be configured either during the provisioning process or on existing virtual datasets.

9.1.7.2.1.1 During provisioning

1. In the **Add dSource** or **Add VDB** wizards, go to the **Hooks** tab.
2. Select the **Operation Type**, enter a **Name**, and input the **Script**.
3. To remove an operation, click the **Trash** icon next to it.
4. After setting all operations, click **Next** to proceed with provisioning.

9.1.7.2.1.2 To edit on existing virtual datasets

1. Log in to the Delphix Management application.
2. In the **Datasets** panel, select the virtual dataset.
3. Navigate to **Configuration** > **Hooks**.
4. Select a hook and click the **Pencil** icon to edit.

5. Use the **+** icon to add or the **Trash** icon to remove operations.
6. Save changes by clicking the checkmark icon.

9.1.7.2.2 Creating hook operations via the administrative Command-Line Interface (CLI)

9.1.7.2.2.1 To define hook operations

- During linking: Edit the array in **LinkingParameters** > **Source** > **Operations**.
- During provisioning: Edit the array in **ProvisionParameters** > **Source** > **Operations**.
- On existing dSources: Edit **Source** > **Operations**.
- On existing virtual datasets: Edit **Source** > **Operations**.

Detailed documentation on these CLI objects is available in the **Help** menu of the Delphix Management application, including `LinkedSourceOperations`, `VirtualSourceOperations`, `RunCommandOnSourceOperation`, and `RunExpectOnSourceOperation` API.

9.1.7.2.2.2 Example of editing hook operations through the CLI

To modify operations for a source, use the following commands:

1. Navigate to `VirtualSourceOperations` for the source.
2. Select a hook to edit. Example commands:

```
delphix> source
delphix source> select "pomme"
delphix source "pomme"> update
delphix source "pomme" update *> edit operations
delphix source "pomme" update operations *> edit postRefresh
```

3. To add an operation:

```
delphix source "pomme" update operations postRefresh *> add
delphix source "pomme" update operations postRefresh 0 *> set
type=RunCommandOnSourceOperation
delphix source "pomme" update operations postRefresh 0 > set command="echo Refresh
completed."
delphix source "pomme" update operations postRefresh 0 > ls
Properties
 type: RunCommandOnSourceOperation ()
 command: echo Refresh completed. ()
delphix source "pomme" update operations postRefresh 0 *> commit
```

4. To delete an operation:

```
delphix source "pomme" update operations postRefresh *> unset 1
delphix source "pomme" update operations postRefresh *> commit
```

9.1.7.3 Hook operations templates

Hook operation templates are designed to store commonly used operations. This allows for efficient reuse of operations across multiple virtual datasets, reducing repetitive work.

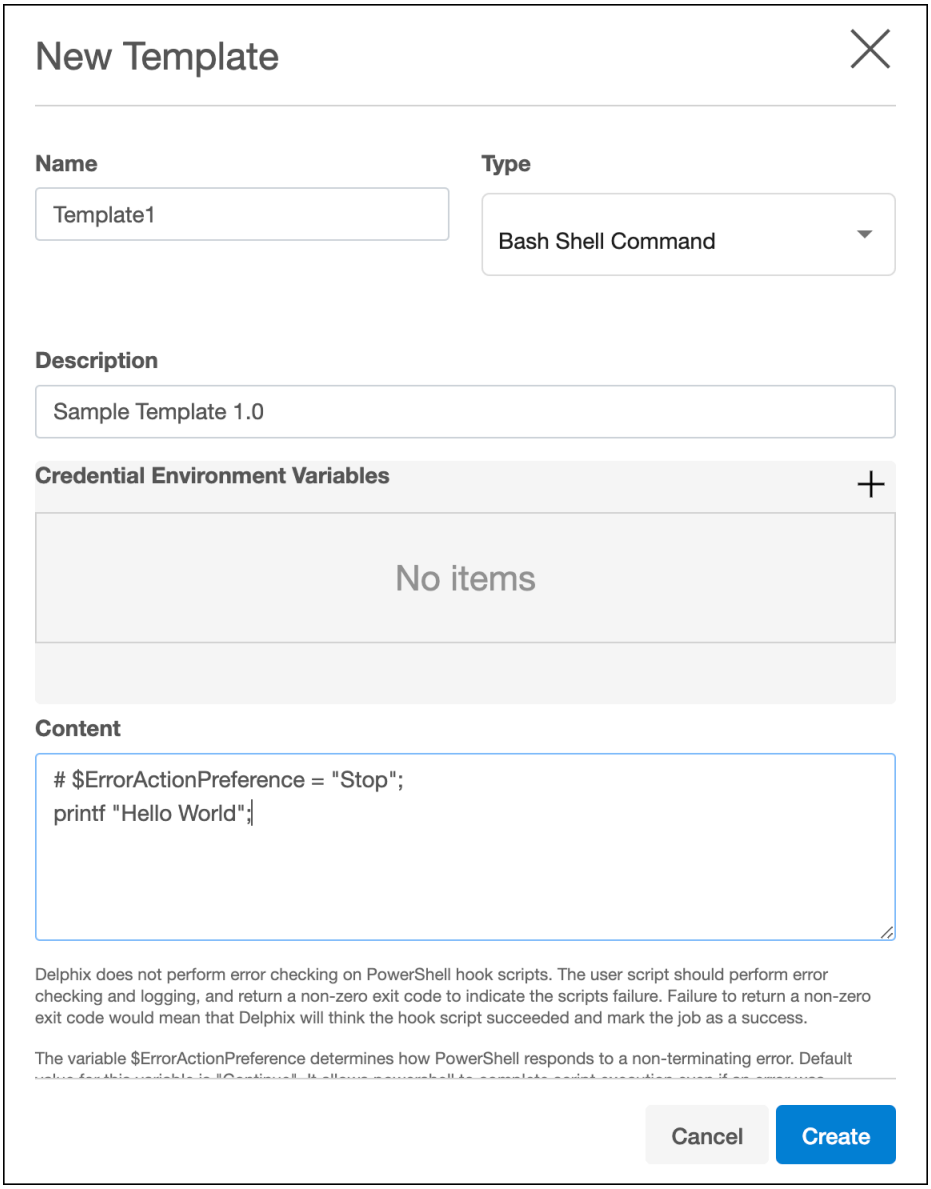
These templates are managed exclusively through the Delphix Continuous Data Engine UI; they are not available via the Command-Line Interface (CLI).

9.1.7.3.1 Creating a hook operation template

The provisioning wizard can import operations from these templates.

Steps to create a template:

1. Log in to the Delphix Management application.
2. Navigate to **Manage** > **Operation Templates**.
3. Click the **+** icon to initiate a new template creation.
4. Provide a **Name** for the template.
5. Select the **Operation Type**.
6. Enter a **Description** and the **Operation Contents**.
7. Click **Create** to finalize the template.



9.1.7.3.2 Importing a hook operation template

To import a template into a dataset:

1. Log in to the Delphix Management application.
2. In the **Datasets** panel, select the desired dataset.
3. Go to **Configuration** > **Hooks**.
4. Select the hook to be edited and click the **+** icon.
5. Choose **Import** and select the template.
6. Click **Import** again.

7. After setting all operations, click **Check** to save changes.

9.1.7.3.3 Exporting a hook operation template

To export a template:

1. Log in to the Delphix Management application.
2. In the **Datasets** panel, choose a dataset.
3. Navigate to **Configuration** > **Hooks**.
4. Select the hook, add a new operation, and select its type.
5. Edit the operation contents in the text area.
6. Click **Export**, then enter a **Name** and **Description** for the template.
7. Click **Export** to complete the process.

9.1.7.4 Hook operations list

9.1.7.4.1 dSource hook operations

Hook	Description
Pre-sync	• Executed before the SnapSync operation of a dSource. • These operations can quiesce data for capture during SnapSync or stop application processes that may interfere with SnapSync.
Post-sync	• Executed after the SnapSync operation of a dSource. • Operations undo changes made by pre-sync hooks and will run regardless of the success of pre-sync hooks or SnapSync.

9.1.7.4.1.1 Virtual Database hook operations

Hook	Description
Configure clone	• Operations performed after initial provision or after a refresh. • Runs after the virtual dataset has started. • During a refresh, this hook precedes the post-refresh hook.

Pre-refresh	• Operations performed before a refresh. • Runs before the virtual dataset is stopped. • Operations can cache data from the virtual dataset for restoration after the refresh.
Post-refresh	• Operations performed after a refresh. • Runs after the virtual dataset has started and after configure clone. • Will not run if the refresh or pre-refresh hook operation fails. • Operations can restore cached data after refresh completion.
Pre-rollback	• Operations performed before a rollback (rewind). • Runs before the virtual dataset is stopped. • Operations can cache data from the virtual dataset for restoration after rewind.
Post-rollback	• Operations performed after a rollback (rewind). • Runs after the virtual database has started. • Will not run if the rewind or pre-rewind hook operations fail. • Operations can restore cached data after rewind completion.
Pre-snapshot	• Operations performed before a snapshot. • Operations can quiesce data for capture during the snapshot or stop processes that may interfere with the snapshot.
Post-snapshot	• Operations performed after a snapshot. • Runs regardless of the success of the snapshot or pre-snapshot hook operations. • Operations can undo changes made by the pre-snapshot hook.
Pre-start	• Operations performed before the startup of a VDB or vFile. • Can be used to initialize configuration files or stop processes that might interfere with the virtual dataset.
Post-start	• Operations performed after the startup of a VDB or vFile. • Can clean up temporary files, restart processes stopped by pre-start hooks, or log notifications.
Pre-stop	• Operations performed before the shutdown of a VDB or vFile. • Can quiesce data or processes prior to the virtual dataset shutdown.

Post-stop	• Operations performed after the shutdown of a VDB or vFile. • Can be used to log notifications, clean up temporary files, or manage related processes.

9.1.7.4.2 Order of execution of hooks for various VDB operations

Provision	Refresh	Rollback	Snapshot	Start	Stop
Pre-start	Pre-refresh	Pre-rollback	Pre-snapshot	Pre-start	Pre-stop
Post-start	Pre-stop	Pre-stop	Post-snapshot	Post-start	Post-stop
Configure clone	Post-stop	Post-stop			
Run masking job	Pre-start	Pre-start			
Pre-snapshot	Post-start	Post-start			
Post-snapshot	Configure clone	Configure clone			
	Run masking job	Run masking job			
	Post-refresh	Post-rollback			
	Pre-snapshot	Pre-snapshot			
	Post-snapshot	Post-snapshot			

9.1.7.4.2.1 Staging server hooks

Staging Server hooks are currently applicable to SAP ASE and will execute on the configured staging server.

Hook	Description
Pre validated sync	Operations performed on a staging server before validated sync. This hook will run before the validated sync operation whenever the validated sync run is triggered.
Post validated sync	Operations performed on staging server after validated sync. This hook will run after the validated sync operation whenever the validated sync run is triggered.

9.1.7.4.2.2 Self-service interaction with hooks

These are the operations performed by the various Self-Service operations:

Rollback
Reset, Undo, and any Restore or Branch operation using source point-in-time or a bookmark that's part of the current container.

Refresh
Refresh and any Restore or Create Branch operation from any shared bookmark or when using the Template as the source of the restore (bookmark or point-in-time).

The Virtual Database Hooks described above will run accordingly. Snapshots are taken, for recovery purposes, before and after every operation that is performed. Bookmark operations will also initiate a snapshot.

9.1.7.5 Self-Service interaction with hooks

This page outlines the operations performed by various Self-Service actions:

9.1.7.5.1 Rollback

- Includes **Reset**, **Undo**, and any **Restore** or **Branch** operation that uses a source point-in-time or a bookmark part of the current container.

9.1.7.5.2 Refresh

- Comprises **Refresh** and any **Restore** or **Create Branch** operation from:
  - Any shared bookmark.
  - Using the template as the source of the restore (bookmark or point-in-time).
- The virtual database hooks described earlier will execute as per these operations.

9.1.7.5.3 Additional notes

- Snapshots are taken for recovery purposes before and after every operation performed.
- Bookmark operations will also initiate a snapshot.

9.1.8 Shell operations

9.1.8.1 RunCommand operation

- Executes a shell command in a Unix environment using `/bin/sh`.
- The environment user executes the command from their home directory.
- The Delphix Continuous Data Engine captures and logs all command output.
- In case of script failure, output is displayed in the Delphix Continuous Data UI and CLI for debugging.
- **Success criteria**: The command must exit with an exit code of 0. Any other exit codes indicate a failure.

9.1.8.1.1 Examples of RunCommand operations

- Direct command input:

```
remove_dir="$DIRECTORY_TO_REMOVE_ENVIRONMENT_VARIABLE"
if test -d "$remove_dir"; then
    rm -rf "$remove_dir" || exit 1
fi
exit 0
```

- Running existing scripts with executable permission:

```
/opt/app/oracle/product/10.2.0.5/db_1/dbs/myscript.sh "$ARG_ENVIRONMENT_VARIABLE"
"second argument in double quotes" 'third argument in single quotes'
```

9.1.8.2 RunBash operation

- Executes a Bash command in a Unix environment using a bash binary provided by Delphix, unless it's a Linux environment, in which case it uses the system's native bash binary.
- Follows the same execution and logging criteria as the `RunCommand` operation.
- **Success criteria**: The Bash command must exit with an exit code of 0.

9.1.8.2.1 Example of RunBash operation

```
remove_dir="$DIRECTORY_TO_REMOVE_ENVIRONMENT_VARIABLE"
# Bashisms are allowed here!
if [[ -d "$remove_dir" ]]; then
    rm -rf "$remove_dir" || exit 1
fi
exit 0
```

9.1.8.3 Running processes in the background

- To detach a process from the Delphix Continuous Data Engine, use the `nohup` command and process backgrounding.

Redirect `stdout` and `stderr` to ensure the Delphix Continuous Data Engine does not block.

9.1.8.4 Examples of running background processes

9.1.8.4.1 Acceptable examples

```
nohup python file.py 1>/dev/null 2>&1 & # both stdout and stderr redirected
```

9.1.8.4.2 Deniable examples

```
nohup python file.py & # no redirection
nohup python file.py 2>&1 & # only stderr is redirected
nohup python file.py 1>/dev/null & # only stdout is redirected
nohup python file.py 2>/dev/null & # only stdout is redirected
```

9.1.9 Other operations

This page introduces the `RunExpect` operation, designed for automating interactions with interactive programs like SSH in a Unix environment. It describes the use of environment variables for script execution, particularly those associated with dSources or VDBs. Guidance on passing credentials securely to hook operations and managing these credentials through the Delphix UI, API, and CLI, including steps for adding, modifying, or deleting credentials in hook operations is also included.

9.1.9.1 RunExpect operation

- Executes an `expect` script in a Unix environment, utilizing the **Expect** utility for automating interactions with typically interactive programs like `ssh`.
- Runs as the environment user from their home directory.
- The Delphix Continuous Data Engine logs all output, displaying it in both the Delphix Management application and CLI for debugging purposes.
- **Success criteria**: The script must exit with an exit code of 0. Other exit codes indicate failure.

9.1.9.1.1 RunExpect operation example

```
spawn ssh user@delphix.com
expect {
    -re {Password: } {
        send "${env(PASSWORD_ENVIRONMENT_VARIABLE)}\n"
    }
    timeout {
        puts "Timed out waiting for password prompt."
        exit 1
    }
}
exit 0
```

9.1.9.2 Environment variables

- Operations executing user-provided scripts have access to OS environment variables.
- For operations associated with dSources or VDBs, specific environment variables are set:
    - `DLPX_DATA_DIRECTORY`: The user-provided mount path on the UI.
    - `USER`: The OS user used for linking the dSource/VDB.

9.1.9.2.1 Environment variables available to hooks

- `<base name>_USER`: Username, present for vault credentials with a username.
- `<base name>_PASSWORD`: For PasswordCredential and vault credentials with a password.
- `<base name>_PRIVKEY`: Private key for KeyPairCredential and vault credentials with a private key.
- `<base name>_PUBKEY`: Public key for KeyPairCredential and vault credentials with a public key.

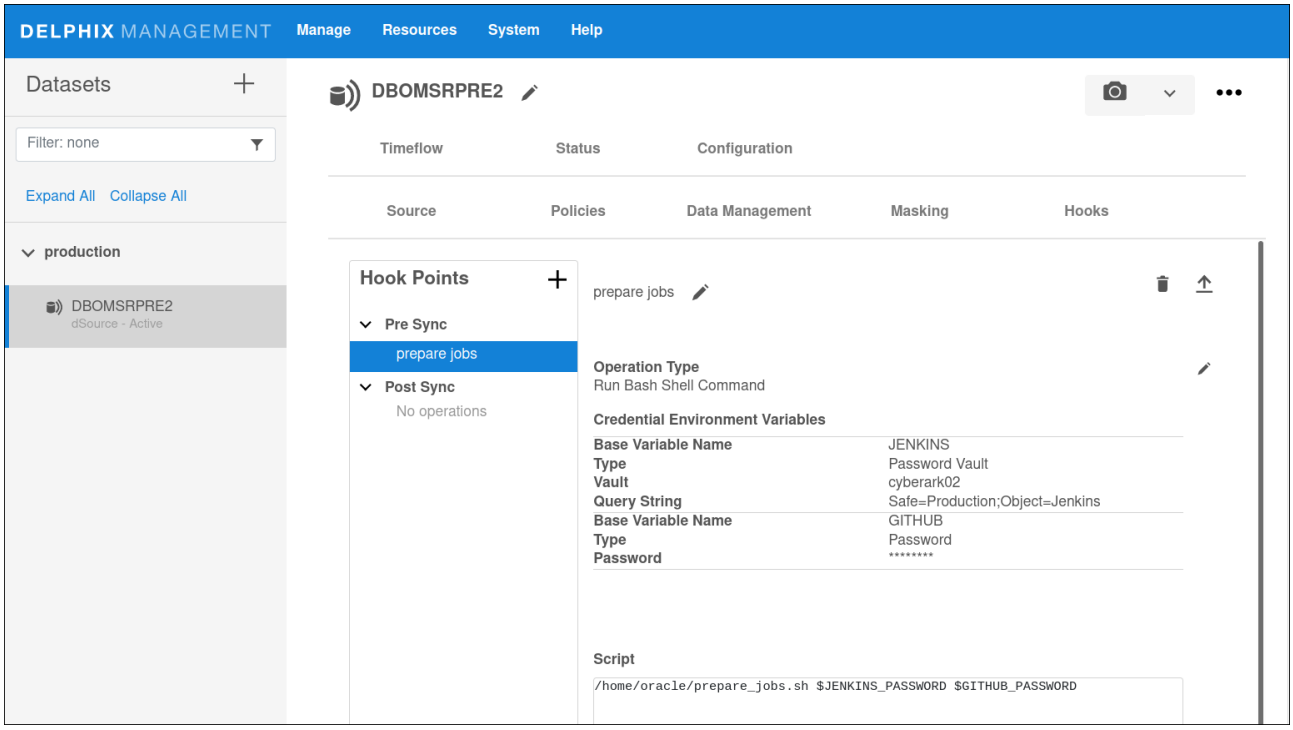
9.1.9.3 Secure credential management

- Pass credentials securely to hook operations to perform tasks like API calls or service management.

- Credentials are passed via environment variables in a hook operation's list property.
- View or update credentials in the **Credential Environment Variables** section in the Delphix Continuous Data Engine UI.

9.1.9.3.1 Credential management steps

1. Log in to the Delphix Continuous Data Engine UI.
2. Under Datasets, select a dSource or VDB.
3. Go to **Configuration** → **Hooks**, then select a hook operation.
4. To add, modify, or delete credentials, click the edit icon next to Operation type.



- Credentials in the Credential Environment Variables section are declared using a **Base Variable Name** followed by `_PASSWORD`.
- For example, variables JENKINS and GITHUB are used as `$JENKINS_PASSWORD` and `$GITHUB_PASSWORD`.

9.1.9.3.2 API and CLI credential configuration

- The credentials list property in the API and CLI is `credentialsEnvVarsList`.
- Credentials can be direct (password or key) or indirect (password vault).
- Direct credentials are managed securely and encrypted; vault credentials are retrieved just before execution.

- Four credential types: `CyberArkVaultCredential`, `HashiCorpVaultCredential`, `PasswordCredential`, `KeyPairCredential` (API and CLI only).

9.1.9.3.3 Example configuration through CLI

```
delphix source "pomme" update operations postRefresh *> edit 0.credentialsEnvVarsList
delphix source "pomme" update operations.postRefresh 0 credentialsEnvVarsList *>
adddelphix source "pomme" update operations.postRefresh 0 credentialsEnvVarsList 0 *>
set baseVarName=API1delphix source "pomme" update operations.postRefresh 0
 credentialsEnvVarsList 0 *> edit credentialsdelphix source "pomme" update
operations.postRefresh 0 credentialsEnvVarsList 0 credentials *> set
type=PasswordCredentialdelphix source "pomme" update operations.postRefresh 0
 credentialsEnvVarsList 0 credentials *> set password="API-KEY-02a0b73f"delphix
source "pomme" update operations.postRefresh 0 credentialsEnvVarsList 0 credentials
*> backdelphix source "pomme" update operations.postRefresh 0 credentialsEnvVarsList
*> adddelphix source "pomme" update operations.postRefresh 0 credentialsEnvVarsList 1
 *> set baseVarName=API2delphix source "pomme" update operations.postRefresh 0
 credentialsEnvVarsList 1 *> edit credentialsdelphix source "pomme" update
operations.postRefresh 0 credentialsEnvVarsList 1 credentials *> set
type=CyberArkVaultCredentialdelphix source "pomme" update operations.postRefresh 0
 credentialsEnvVarsList 1 credentials *> set queryString="Safe=Apis;Folder=Root\Servi
ce2;Object=ApiClient"delphix source "pomme" update operations.postRefresh 0
 credentialsEnvVarsList 1 credentials *> set vault=MyCyberArk1delphix source "pomme"
 update operations.postRefresh 0 credentialsEnvVarsList 1 credentials *> commit
```

Upon execution, the hook receives two sets of environment variables with corresponding credentials:

- `API1_PASSWORD` with value `API-KEY-02a0b73f`.

Variables:

- `API2_USER`
- `API2_PASSWORD` containing the values stored in the vault MyCyberArk1 at the location of the provided query string. The API2 credentials are retrieved from the vault just before the hook executes, so they will always contain the latest values.

9.1.10 Policies

9.1.10.1 Overview

Creating policies within the Delphix Continuous Data Engine is an effective method to automate dataset management. Users can synchronize data, capture snapshots, and configure both retention and replica retention policies. This ensures that virtual databases are refreshed and data remains ready for use.

To set up policies, go to the **Manage** dropdown menu and select **Policies**. The Delphix Continuous Data Engine offers five main policy categories in relation to dataset objects:

- **SnapSync**: Dictates the frequency of snapshot captures for a source database, known as a dSource.
- **VDB Snapshot**: Determines how often snapshots are taken for a Virtual Database (VDB).
- **Retention**: Specifies the duration for which snapshots and log files are kept for both dSources and VDBs.
- **VDB Refresh**: Automates the refreshing of a VDB using either the latest snapshot or Timeflow logs, with 'None' being the default setting.
- **Replica Retention**: Governs the length of time snapshots are preserved on replicated namespaces after deletion from the replication source. Snapshots are retained unless the source database is deleted. If the source is deleted, subsequent replication updates will remove the database and any snapshots retained on the target.

It is crucial to avoid setting conflicting policies. For instance, a NONE setting for the *Snapshot* policy can directly affect *retention* policies and lead to unexpected growth in VDB data. An example case could involve an engine that runs into a domain space issue because archive logs from a single VDB consume a significant portion of the storage pool.

This would be attributed to a NONE Snapshot policy being in place, coupled with a one-week log retention policy. The logs, necessary for provisioning from the only available snapshot, prevented the engine from enforcing the retention policy.

9.1.10.2 Default vs. Custom policies

There can be default or custom policies for each of the five categories above. Consider the benefits of both.

9.1.10.2.1 Default policies

- **Scope**: Implemented at the domain level; affect all objects within a category.
- **Modification**: Settings can be adjusted, but policy names are fixed.
- **Access**: Available to users with Delphix Admin credentials.

9.1.10.2.2 Custom policies

- **Flexibility**: Designed to cater to specific needs with customizable scheduling.
- **Scheduling**: Time intervals can vary widely, from minutes to days.
- **Access**: Can be created and managed by users with Delphix Admin credentials, as well as by group and object owners.

9.1.10.3 Setting different policies for objects in a group

Policies at the group level influence all contained objects. For unique requirements within a group, set the group-level policies initially, then apply object-specific policies.

9.1.10.3.1 SnapSync policy

SnapSync policies dictate the frequency of source database snapshots. The default SnapSync policy schedules daily snapshots at 3:30 AM local time, with a four-hour completion window. If SnapSync fails to complete in this timeframe, it will attempt again at the next scheduled time. Modify the default policy by

clicking the **Edit** icon or create new ones by clicking **Add**. Policies can be tailored using the **Schedule date** picker, by **Interval**, or with a `cron` expression.

9.1.10.3.2 SnapShot Policy

9.1.10.3.3 Retention policy

The Retention policy determines the duration for keeping snapshots and log files, which are essential for rewinding or provisioning objects. Snapshot retention must be at least as long as log retention. Allocating more storage to the Delphix Continuous Data Engine may be necessary for longer retention periods. The Retention policy, alongside the SnapSync policy, plays a critical role in the engine's performance and storage demands. Customize these to preserve data for a longer period of time, enabling access to earlier data points.

9.1.10.3.4 Replica Retention policy

The Replica Retention policy outlines the retention span for snapshots on replicated namespaces after their deletion from the source. Generally, deletion on the source leads to deletion on the target; however, the new Replica Retention policy extends the life of these snapshots on the target. Apply this policy to an entire namespace or specific groups or objects within. These snapshots support provisioning or refreshing VDBs, but point-in-time provisioning might be limited to optimize disk space. Replica Retention policies are enforced automatically, either on a set schedule or when a replication job is received.

Replica Retention policy is designed to lengthen snapshot lifespans only. They can be deleted when:

- Their policy duration is adjusted, and they fall outside the retention period.
- The original dSource or VDB is deleted.
- For Oracle **virtual pluggable databases** (vPDB), if the controlling **container database** (CDB) is removed or unreplicated, dependent snapshots on both the CDB and the vPDB are deleted. This could occur if a vPDB migrates from one CDB to another.

9.1.10.3.5 Benefits of longer retention

Extending the retention period for snapshots and logs enhances your ability to roll back to earlier data states. The advantages of a more prolonged retention timeline include:

- **Compliance with regulations**: Meets standards like SOX compliance, where data preservation is crucial.
- **Supports development**: Accommodates the needs of environments with frequent application updates and development cycles.
- **Controlled data progression**: Allows for careful and deliberate data changes.
- **Risk mitigation**: Reduces project risks by retaining critical data states.
- **Efficient recovery**: Facilitates a quicker rollback or restoration to previous points in time, enhancing operational resilience.

9.1.10.4 Create a custom policy

This section outlines the steps to create custom policies using `cron` expressions for specific database objects or groups.

Initiation: Custom policies are created by modifying an existing policy tied to a database object. This can be done during the object's initial creation or later through the Policy Management screen. For guidance on creating custom policies for dSources and VDBs during linking and provisioning, refer to the respective topics for each data platform.

Scheduling options: Remember, policies can be set using the Schedule date picker, by specifying an Interval, or by choosing 'Custom' to input a Quartz cron expression.

9.1.10.4.1 Procedure

1. Log in to the Delphix Management application.
2. Click on **Manage**.
3. Choose **Policies**.
4. Click on the tab for the object or group you are creating a policy for.
5. In the relevant tab, click the **Add** button. For instance, to create a new VDB Snapshot policy:
    a. Go to the VDB snapshot tab.
    b. Click on **+ VDB snapshot**.
6. Enter a **Name** for your policy.
7. Choose a **Time Zone** (by default, the system time zone is selected if none is specified).
8. Set a **Timeout** to determine how long a job should run before being terminated if not completed.
9. Pick a **Schedule** for the policy.
10. Click **Next** and select the object(s) to which this policy will apply.
11. Review your settings and click **Submit** to finalize the policy creation.

9.1.10.5 Policies and timezones

Time zones are a critical component for scheduling SnapSync, VDB Snapshot, and VDB Refresh policies. Here is how to adjust the time zone settings for these policies:

1. Log into the Delphix Continuous Data Engine UI.
2. Click on **Manage**.
3. Choose **Policies**.
4. Click on the **pencil icon** next to the policy you wish to edit.
5. Select the desired **Time Zone** from the drop-down list.
6. Click **Submit** to apply the new time zone setting.

Retention and Quota policies do not have scheduled times and therefore do not require a time zone configuration.

9.1.10.5.1 Post-upgrade policy cloning

When the Delphix Continuous Data Engine is upgraded, it automatically clones existing policies to maintain consistent behavior. These clones differ from the originals only in their time zone settings. After an upgrade, you may notice policy names include the time zones they are associated with.

Default policies are not cloned and will always follow the time zone settings of the Delphix Continuous Data Engine.

9.1.10.5.1.1 Example of Policy Cloning After an Upgrade

Consider dSources and VDBs that were set to operate under EST (America/New_York) or CST (America/Mexico_City). Post-upgrade, new policies reflecting these time zones will be created:

- **Original policy**: UserSnapSync
  - **New policies**: UserSnapSync (America/Mexico_City), UserSnapSync (America/New_York)
- **Original policy**: SnapshotTest
  - **New policies**: SnapshotTest (America/Mexico_City), SnapshotTest (America/New_York)
- **Original policy**: UserRefresh
  - **New policies**: UserRefresh (America/Mexico_City), UserRefresh (America/New_York)

After the system upgrade, check that the policies are operating as intended. This clarity ensures there is no confusion regarding the execution times of your policies.

9.1.10.6 Configure retention on individual snapshots

This section details how to set a custom retention period for individual snapshots, which can supersede the retention policy assigned to the container. For instance, selecting 'forever' ensures the snapshot is not purged by the retention policy.

9.1.10.6.1 Procedure

1. Login to the Delphix Management application.
2. Go to **Manage** > **Datasets**.
3. Find and expand the dSource or VDB you wish to modify.
4. Select the **Timeflow** tab.
5. Expand the snapshots section for which you wish to modify the retention time (Note, it may take a moment for individual snapshots to display).
6. Click on the three dots to expand the sub-menu.
7. Select **Keep for ...**.
8. In the **Keep For** dialog, specify the number of days, or select **forever**.

9. Click **Save** to apply the custom retention setting.

9.1.11 Basic troubleshooting

9.1.11.1 Environment

Connectivity issues between environments and engines:

- **SSH connectivity test**: To diagnose network connectivity issues, run an SSH connectivity test to the target host. This will validate the communication link and identify any potential network barriers.
- **DSP throughput test**: Conduct a DSP (Delphix Session Protocol) throughput test to the target host to ensure efficient data transfer rates and troubleshoot any connection issues.
- **Environment refresh**: Periodically, running an environment refresh can resolve detection issues related to installations and databases.

Unable to find installations and databases:

- If the engine fails to locate installations or databases, confirm network configurations and check for any updates or changes in the environment settings.
- Review management logs for signs of connectivity issues or environment detection failures. These logs can be accessed via the support bundle.

9.1.11.2 Delphix Toolkit

Ensure that the Delphix Toolkit is correctly installed and configured as per the environment's requirements. Issues with the toolkit can often lead to broader connectivity or operational problems.

9.1.11.3 dSource and virtual databases (VDB)

Ingestion issues:

- Ingestion problems, particularly with connectors like Oracle, MS SQL Server, etc., require a review of connector-specific logs.
- For AppData or non-Oracle/MS SQL Server/SAP ASE connectors, refer to the specific troubleshooting sections for guidance on managing unique issues.

9.1.11.4 Log management

Plugin logs:

- Plugin logs are crucial for diagnosing issues related to environment connectivity, dSource, and virtual databases.
- These logs are packaged within the standard Support bundle. For detailed information on creating Support bundles, refer to the Support logs[303] page.

303 https://cd.delphix.com/docs/latest/creating-support-logs

- The support bundle contains logs stored on the Delphix Continuous Data Engine side but does not include target side plugin logs. The location and management of these logs are detailed in the plugin-specific troubleshooting sections.

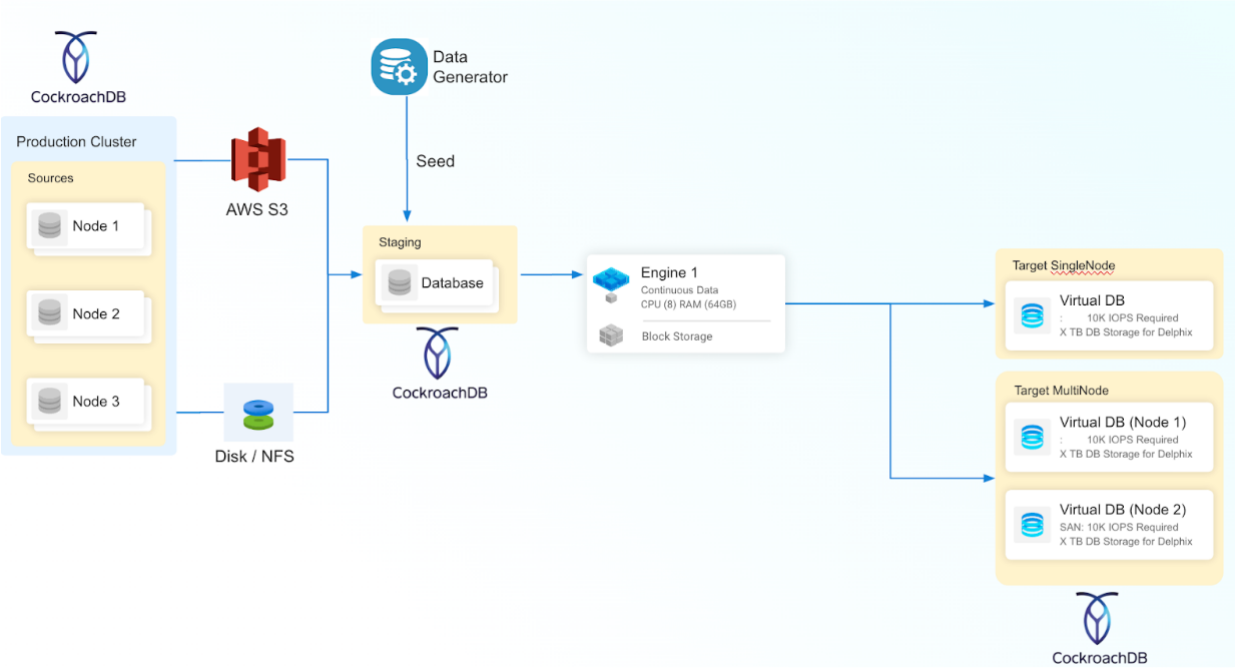
Locating logs on the engine:

- Users can find logs directly on the Delphix Continuous Data Engine. Visit the Monitoring and log management[304] documentation section for more information.

9.2 CockroachDB data sources

The Delphix CockroachDB virtualization solution uses a Staging Architecture Model. In this model, the source (production) database is copied to a staging environment. The staging environment's database is ingested by the Delphix Continuous Data Engine, which then enables virtualized copies on a target environment.

The CockroachDB plugin leverages CockroachDB's built-in backup and restore procedures to virtualize the data source. You can use the BACKUP[305] statements to efficiently backup the cockroach cluster and data to popular cloud services such as AWS S3 or NFS, and the RESTORE[306] statement to efficiently restore schema and data as necessary. The CockroachDB restore process is capable of restoring from either a full backup or from a combination of a full backup with incremental backups.

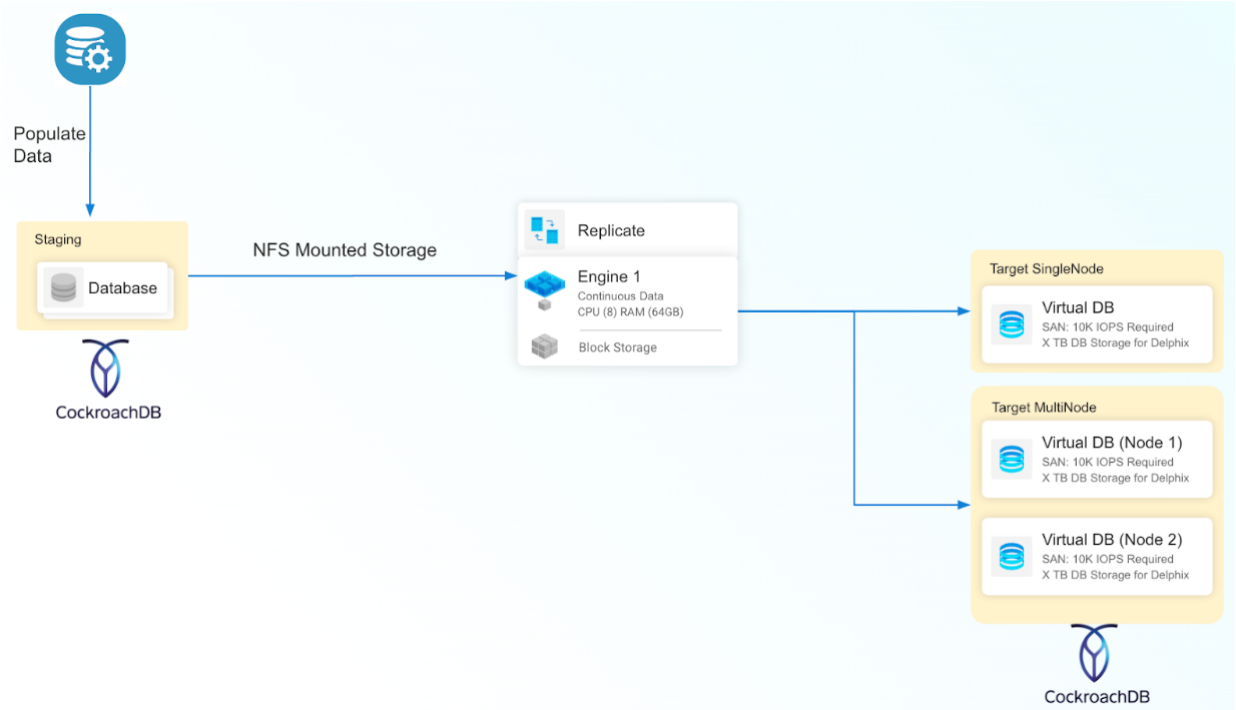


The CockroachDB connector supports the following ingestion mechanisms:

304 https://cd.delphix.com/docs/latest/monitoring-and-log-management
305 https://www.cockroachlabs.com/docs/v22.1/backup.html
306 https://www.cockroachlabs.com/docs/v22.1/restore

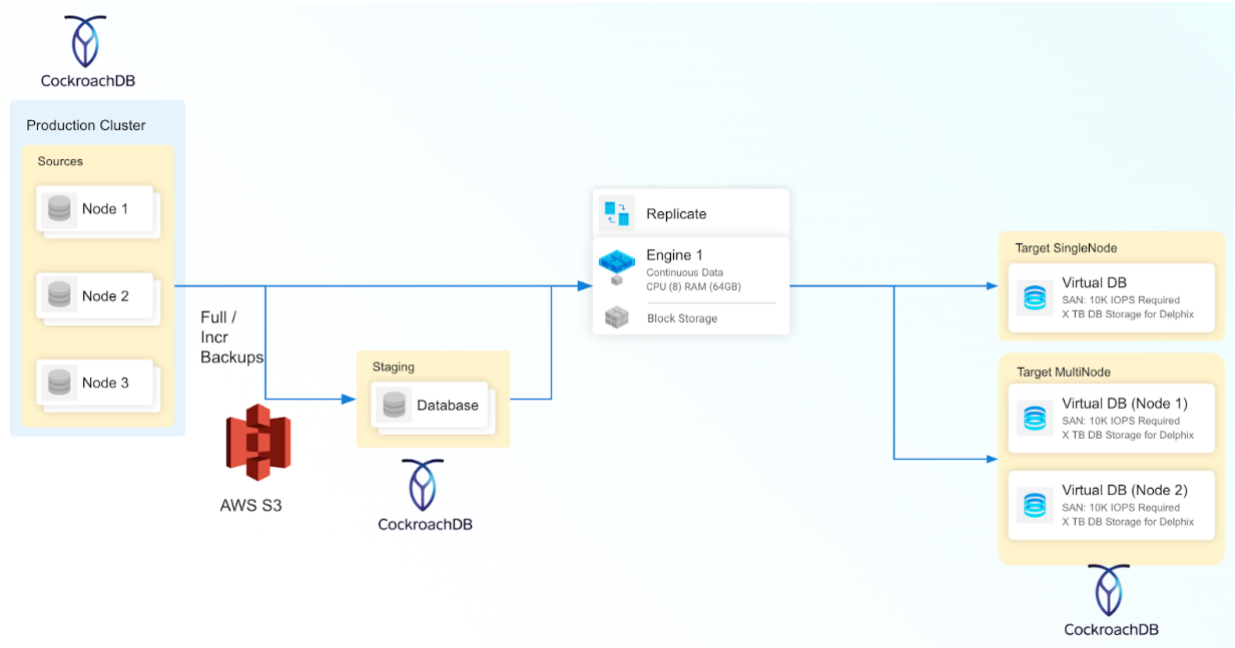
9.2.1 Seed

The Seed ingestion mechanism in the CockroachDB connector serves as a foundational approach for creating a seed database, allowing users to populate it with diverse data through various techniques such as built-in data generators, SQL statements, or the IMPORT command. This seed database serves as the initial dataset, providing a starting point for users to establish their CockroachDB environment.



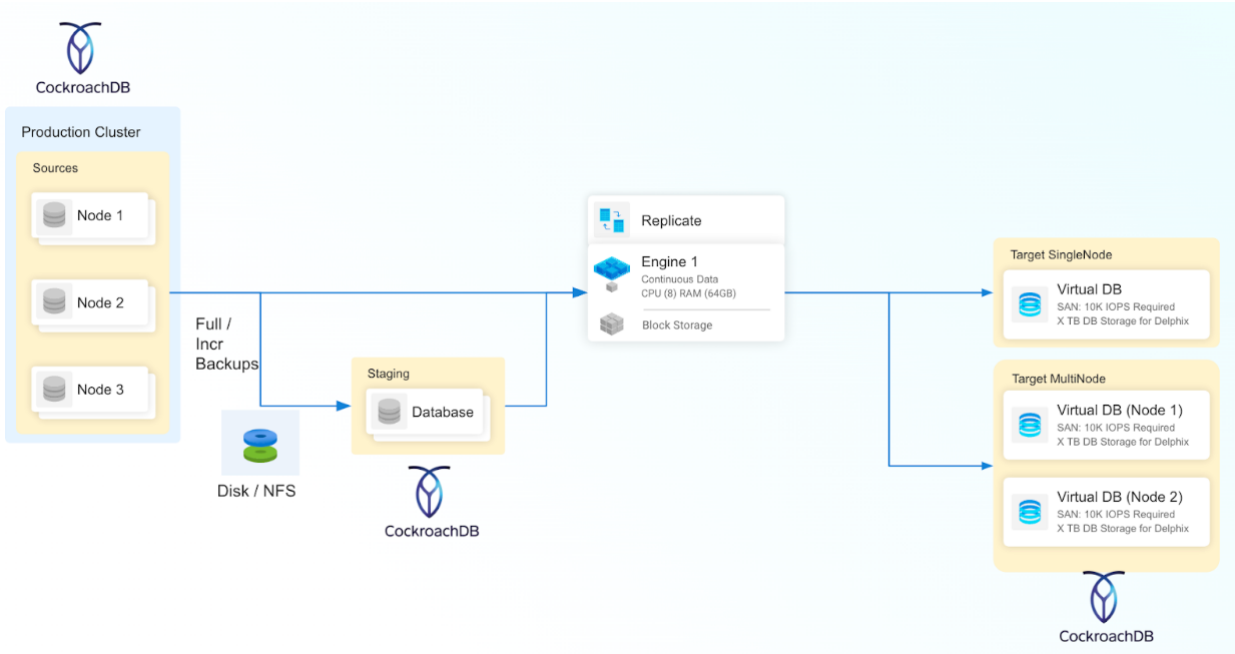
9.2.2 AWS S3 Backup

This ingestion type is generally used to create a staging database by utilizing backup files found within an AWS S3 location. This process facilitates the creation of a staging database through a zero touch production approach. Periodic backups stored on AWS S3 locations can be loaded to create a dSource timeflow.



9.2.3 Disk Backup

This ingestion type is generally used to create a staging database by utilizing backup files found within Disk/NFS locations which are accessible from the Staging host. This process facilitates the creation of a staging database through a zero touch production approach. Periodic backups stored on Disk/NFS locations can be loaded to create a dSource timeflow.



9.2.4 CockroachDB virtualization process

This topic serves as a quick guide that will walk you through your first implementation of the CockroachDB Connector, providing high-level steps. Each step briefly describes the task and the expected result. Refer to the documentation links for more detailed information.

1. Begin by reviewing the genera (see page 1014)l Delphix Continuous Data architecture diagrams (see page 1014) Identify existing or new CockroachDB source, staging, and target environments, the underlying Operating Systems, and CockroachDB database versions. This will give you a good understanding of the architectural strategy and confirm support.
2. Review the CockroachDB Support Matrices (see page 1078) Confirm that all system requirements are supported by the CockroachDB plugin, such as Continuous Data Engine, staging/target environment Operating Systems, and CockroachDB database versions. This will ensure that there are not any unexpected issues later.
3. Deploy and configure Delphix Continuous Data.
4. Install the CockroachDB plugin (see page 1086) The CockroachDB plugin establishes the connection between the Delphix Continuous Data Engine and the plugin version. It is recommended to stay up-to-date with the latest version.
5. Create staging and target environments (see page 1079) The staging environment will be a replica of the source environment that Delphix Continuous Data Engine will use to stay in sync. The target environment will provide the infrastructure to generate virtual database copies.
   a. Consult the source environment (see page 1079) and staging/target environments (see page 1079) requirements documentation for configuration details.
6. Discover Staging and Target Environments (see page 1089) To enable the Delphix Continuous Data Engine to recognize the provided infrastructure, search for CockroachDB installation, and then make it available for database linking and provisioning.
7. Linking the CockroachDB database (see page 1093) This configures a dSource which enables the syncing mechanism between the source database, staging database, and Delphix Continuous Data Engine. There are many ingestion methods that differ between plugins and solve different requirements. Therefore, we encourage you to identify the ingestion method that best suits your organization and infrastructure policies.
8. Provision virtual CockroachDB databases (see page 1100) Administrators and self-service users have the capability to provision an unlimited number of copies for their virtual databases.

Congratulations on successfully creating a virtual database using the Delphix Continuous Data Engine and the CockroachDB plugin! With the basics understood, we encourage you to learn about our more advanced features such as the Seed, AWS S3 and Disk ingestion methodology or our developer's APIs and CLIs.

9.2.5 CockroachDB requirements and prerequisites

- Network and connectivity requirements for CockroachDB (see page 1081)
- Sudo privileges for the CockroachDB environment OS user (see page 1084)
- CockroachDB support matrices (see page 1078)

- Requirements for cockroachDB source databases (see page 1079)

9.2.5.1 CockroachDB support matrices

9.2.5.1.1 Support Matrix

Refer to the following support matrices to ensure compatibility between the environment, database, connector, and Delphix Continuous Data Engine.

Compatible upgrades have a green **Y**, while incompatible upgrades show a red **N**

Color	Supported?
Y	Yes
N	No

9.2.5.1.2 Environment and Database Compatibility

Refer to this table to learn about supported Linux and Windows OS and DBMS versions of Cockroach DB:

Supported OS Version	Supported DBMS Versions				
	CRDB v21.1.x	CRDB v21.2.x	CRDB v22.1.x	CRDB v22.2.x	CRDB v23.1.x
RHEL 7.x	Y	Y	Y	Y	Y
RHEL 8.x	Y	Y	Y	Y	Y
Windows	N	N	N	N	N

9.2.5.1.3 Engine Compatibility Matrix

Delphix Engine Versions	Supported CockroachDB Connector Versions				
	CRDB v1.0.0	CRDB v1.1.0	CRDB v1.1.1	CRDB v1.1.2	CRDB v1.2.0

6.0.12.0 - 6.0.17.x	**Y**	Y	Y	Y	Y
7.0.0.0 - 20.0.0.0	**Y**	Y	Y	Y	Y

9.2.5.1.4 Ingestion / Deployment Compatibility Matrix

Ingestion Type	**CockroachDB Deployment**				
	Seed	Disk	AWS	Azure	GCP
Full	Y	Y	Y	N	N
Incremental	Y	Y	Y	N	N

Notes:

- The CockroachDB restore process is capable of restoring from either a full backup or from a combination of a full backup with incremental backups.
- CockroachDB does not support incremental-only restores as highlighted here[307].

9.2.5.1.5 CockroachDB Version compatibility across Delphix environments

- CockroachDB binaries in all environments (source, staging and target) must be of the same version for all data ingestion types (Seed / AWS S3 / Disk Backup).

9.2.5.1.6 CockroachDB data source limitations

- CockroachDB Ingestion from Azure and GCP are not supported.
- CockroachDB Point-in-time restore is not supported.
- Delphix V2P is not supported.
- Delphix Password Vault is not supported.

9.2.5.2 Requirements for cockroachDB source databases

The source database requirements are listed below:

307 https://www.cockroachlabs.com/docs/v23.2/restore#restore-types

1. Retrieve a comprehensive list of CockroachDB Backup's subdirectories absolute paths by executing the following command:

```
SHOW BACKUPS IN 's3://{bucket name}/{path}?AWS_ACCESS_KEY_ID={placeholder}
&AWS_SECRET_ACCESS_KEY={placeholder}';
```

```
SHOW BACKUPS IN 'nodelocal:/self/nfsmount/backups';
```

> The format of the CockroachDB Backup's subdirectory path follows a date-based naming scheme, specifically *<year>/<month>/<day>-<timestamp>*..

2. Before initiating the restoration process on the staging database, make sure to identify the following key information:

- Storage Bucket Name (AWS S3): Record the name of the storage bucket in AWS S3, including the absolute path where CockroachDB backups are situated in AWS S3.
- AWS_ACCESS_KEY_ID: The account access name or key associated with object storage holding CockroachDB backups, as it serves a crucial role in authentication.
- AWS_SECRET_ACCESS_KEY: The account secret access key associated with the account name field. This is essentially the password that gives access to CockroachDB backups.

This pertinent information can be extracted from the backup command utilized for backing up the CockroachDB full cluster on AWS S3 storage, exemplified by the following command:

```
BACKUP INTO 's3://{BUCKET NAME}?AWS_ACCESS_KEY_ID={KEY ID}
&AWS_SECRET_ACCESS_KEY={SECRET ACCESS KEY}' AS OF SYSTEM TIME '-10s';
```

9.2.5.2.1 Requirements for CockroachDB staging and target environments

The Staging and Target environments have the same set of requirements for CockroachDB.

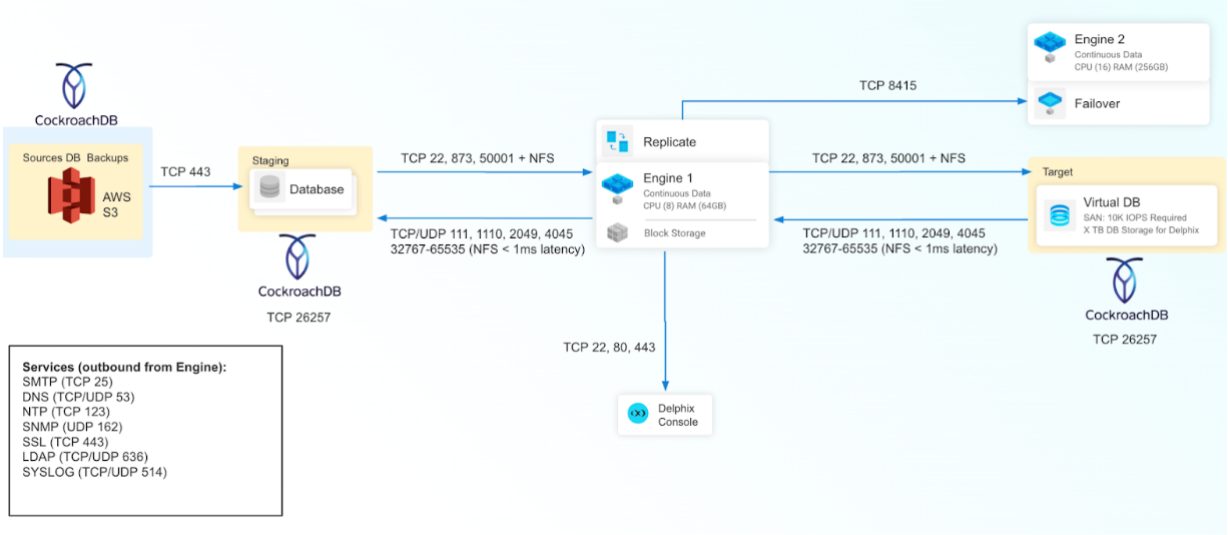
9.2.5.2.2 Network and Connectivity Requirements

Ensure that the requirements for Delphix Engine network connectivity[308] are met before proceeding further.

For CockroachDB connector specific network and connectivity requirements, see the relevant topics below.

308 https://cd.delphix.com/docs/latest/network-connectivity-requirements

9.2.5.2.2.1 Ports for AWS S3 / Disk Backup Ingestion



9.2.5.3 Network and connectivity requirements for CockroachDB

9.2.5.3.1 SSH connectivity to target host

1. Staging and target environments are required to have sshd running and configured such that the Continuous Data Engine can connect over ssh.
2. The Delphix platform expects to maintain long-running, highly performant ssh connections with the host environment.
3. Depending on the Ingestion type, once connected to a target environment through SSH, the Continuous Data Engine initiates a CockroachDB client connection(cockroach) from the staging environment to the source environment.

Whitelisting inbound port for SSH connection to target host

Protocol	Port Number	Use
TCP	22	SSH connections to the target database environment
TCP/UDP	111	Remote Procedure Call (RPC) port mapper used for NFSv3 mounts

TCP	1110	Network Status Monitor (NSM) client from target hosts to the Continuous Data Engine
TCP	2049	NFS client from target hosts to the Continuous Data Engine (NFSv3 and NFSv4)
TCP	54043	Client mount daemon (NFSv3 only)
TCP	54044	Lock state notification service (NFSv3 only)
TCP	54045	Network Lock Manager (NLM) client from target hosts to Continuous Data Engine (NFSv3 only)
UDP	33434 - 33464	Traceroute from the target database server to the Continuous Data Engine (optional)

9.2.5.3.2 Connecting to CockroachDB Instances [Source + dSource + VDBs]

Outgoing [From]	Incoming [To]	Protocol	Port Number	Use
Staging host	Source CockroachDB Instance	TCP	<source-port>	CockroachDB Source database instance port to connect to setting up replication pipeline. The default port for a CockroachDB instance is 26257.
End-user hosts	Staging / Target CockroachDB instance	TCP	<target-port>	Connecting to the created database instance on the target host.

9.2.5.3.3 Host environment requirements for CockroachDB

Staging/Target Host Requirements	Additional Details

Host Compatibility Check The operating system and architecture of the target environment must match the source environment.	It is required that the source and the target environments must have identical Operating system and hardware architecture configurations.
CockroachDB Installation and Compatibility Check There must be an installation of CockroachDB on the staging/target host that is compatible with the installation of CockroachDB on the source host.	The CockroachDB binary version on the staging/target must be identical to the source host.
There must be a directory on the staging/target host where the CockroachDB Connector support files can be installed (for example, /var/tmp) with the following properties: 1. The directory must be writable by the operating system user mentioned above. 2. The directory must have at least 1.5 GB of available storage.	This path is called the **"Toolkit Path"** on the Environment creation UI page.
During VDB provisioning, there must be a mount point directory (for example, /mnt) that will be used as the base for mount points that are created when provisioning a VDB with the following properties: 1. The mount point directory must be writable by the operating system user mentioned above. 2. The mount point directory should be empty.	

Hostname and IP must be correctly set in /etc/hosts file.	The output of "hostname -i" command should produce the correct result as the IP address of the server.
	For example:
	`[root@source ~]# hostname -i`
	`10.110.211.225`

Required Utilities and Tools
Ensure that either the **netstat** or **ss** utility is installed on the Staging/Target Host.
	CockroachDB Connector checks the status of the running CockroachDB instance and validates if ports are available on the staging/target host using the **netstat** or **ss** utility.
	Checking if **netstat** or **ss** utility is installed
	`which netstat`
	`which ss`

Storage Requirement	1. Staging Host must have enough storage space to hold the source backup file. 2. Empty folder on staging host to hold delphix toolkit [approximate 2GB free space]
Sizing Requirement	For comprehensive details and configuration guidelines on setting up staging/target CockroachDB clusters, refer to the official documentation: [CockroachDB Sizing Recommendations](309)

9.2.5.4 Sudo privileges for the CockroachDB environment OS user

Continuous Data Engine connects to the target host (via SSH) to perform the following operations:

1. Continuous Data Engine dynamically mounts and unmounts directories for linking and provisioning operations. This privilege is required because mount and unmount are typically reserved for users with superuser (root) permissions.
2. Continuous Data Engine dynamically creates and removes directories for linking and provisioning operations.

Operations (1) and (2) above will require sudo privileges for the OS user.

Configuring sudo access on target environments [as root user]:

1. The sudo configuration exists in /etc/sudoers file. We can edit this file directly or use a utility called visudo[310] to add the privilege.

```
vi /etc/sudoers
# Or
visudo
```

309 https://www.cockroachlabs.com/docs/stable/recommended-production-settings
310 https://www.sudo.ws/docs/man/1.8.13/visudo.man/

2. Adding the privileges

```
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD: /bin/mount,/bin/umount,/bin/mkdir,/bin/rmdir, /bin/ps
delphix_os ALL=(couchbase) NOPASSWD: ALL
```

> The examples used above are for illustrative purposes and the sudo file configuration options are subject to change.

3. To secure communication within your CockroachDB staging/target cluster, it is imperative to ensure that the certificate directory contains valid certificates encompassing the IP address and hostname of the CockroachDB staging/target server.
   a. List Certificates and Keys

```
cockroach cert list --certs-dir=[path-to-certs-directory]

$ cockroach cert list --certs-dir=/home/cockroach/certs
Certificate directory: /home/cockroach/certs
  Usage  | Certificate File |    Key File     |  Expires   |
Notes                            | Error
---------+------------------+-----------------+------------
+-----------------------------------------------------------------
+--------
  CA     | ca.crt           |                 | 2032/03/29 | num certs:
1                                                   |
  Node   | node.crt         | node.key        | 2027/03/26 | addresses:
crdb-node.example.com, 10.1.1.1                     |
  Client | client.root.crt  | client.root.key | 2027/03/26 | user: root
|
(3 rows)
```

   b. Verification of SSL/TLS Connection

```
cockroach sql --certs-dir=[certs-locations] --host=[ip-address]:[port]
```

> Make sure to replace **[certs-locations]**, **[ip-address]**, and **[port]** with the actual locations and details.

9.2.5.4.1 Requiretty settings for CockroachDB

The Continuous Data Engine requires that the requiretty setting be disabled for all Continuous Data users with sudo privileges.

It is required to specify the NOPASSWD qualifier within the "sudo" configuration file, as shown here: Sudo File Configuration Examples for CockroachDB Environments[311]. This ensures that the "sudo" command does not demand the entry of a password, even for the "display permissions" (i.e. "sudo -l") command.

- Additionally, requirements for the Delphix OS user are as follows:
  - Should be a regular o/s user with primary group as cockroach software installer O/S user.
  - Must have access to backup files of the source database stored on the staging host.
  - Must have read and execute access on all files within CockroachDB installation Path: locations where software binaries are installed and located on the staging host. Minimum permission level 775 recommended.
- Ensure to disable any login banners that resemble the one depicted in the image below on the staging host.

```
[staging_host]$ su - delphix
*******************************************************
*
*              Welcome to the Linux Terminal
*
*-------------------------------------------------------
*
*  You are now logged in as: delphix
*
*  Take caution with your actions and be responsible!
*
*******************************************************
Password:
```

9.2.6 CockroachDB installation and upgrade

9.2.6.1 Downloading the connector

The connector is provided as a downloadable .json file from the Delphix download site. Use the steps below to download it:

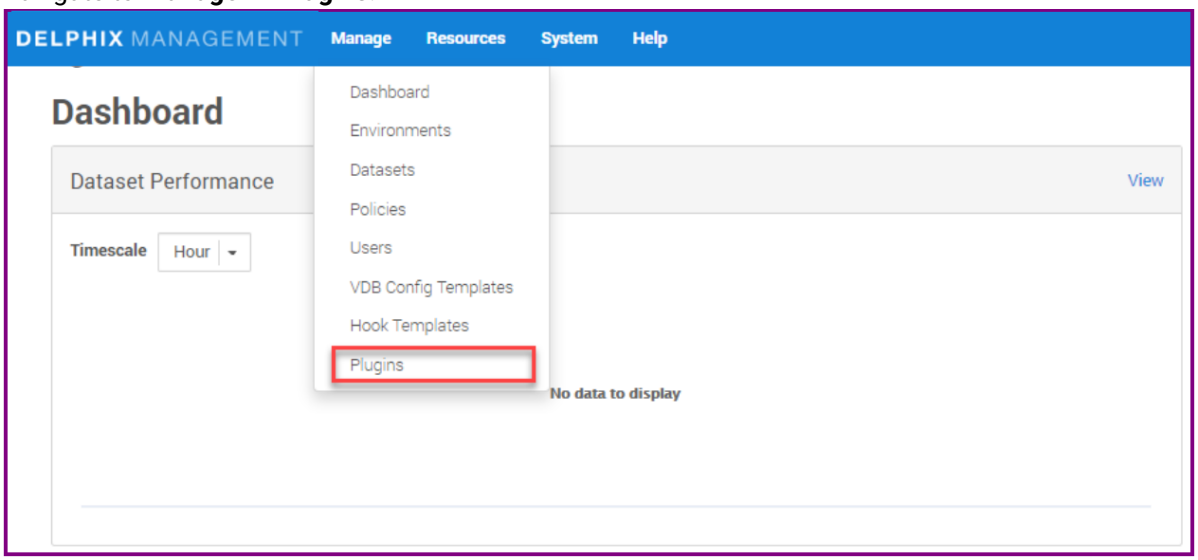
1. Navigate to the Delphix Download portal by clicking the following link or entering https://download.delphix.com/ into the address bar.
2. Login to the download portal using the same email and password credentials used for Delphix Support.

311 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/114529893

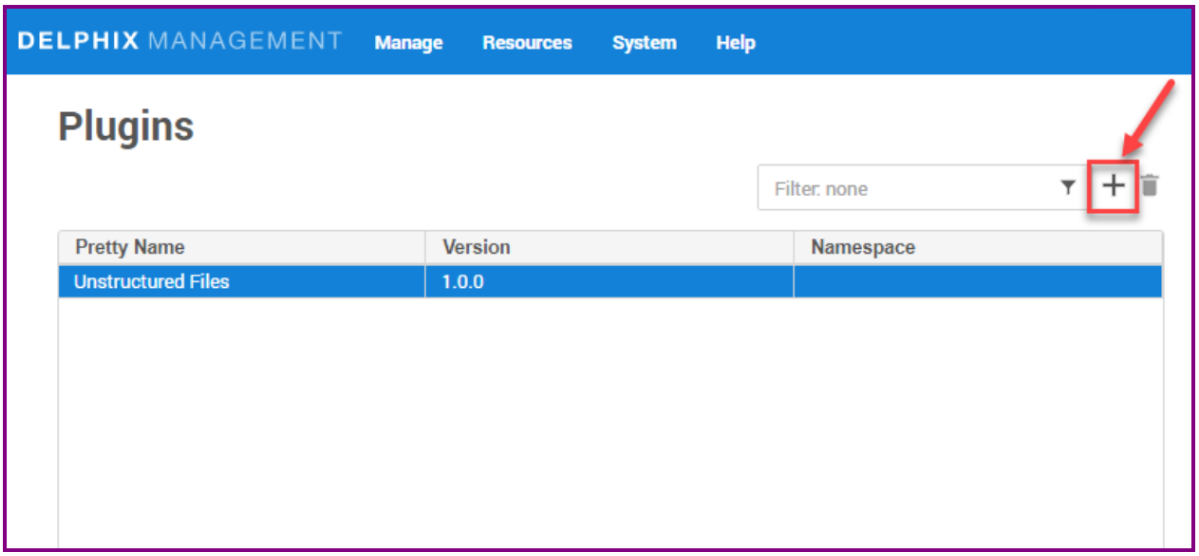
3. Navigate to **Delphix Product Releases** → **Continuous Data Connectors** → **CockroachDB** → **v1.x.x.x** -> **sc_crdb_v1.x.x.x.json** and the download should automatically begin.

9.2.6.2 Installing the connector

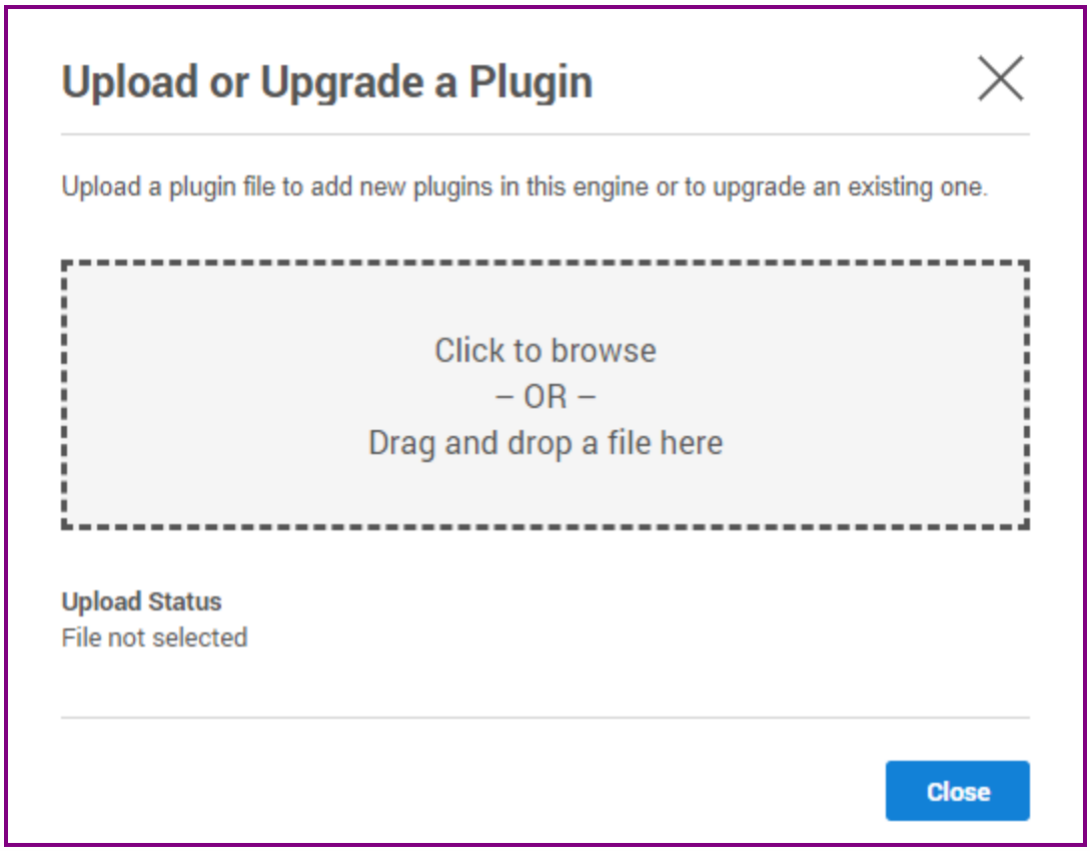
1. Login to the **Delphix Management** application.
2. Navigate to **Manage** → **Plugins**.



3. Click the **+** icon to upload the connector file.



4. This step shows the **Upload or upgrade a plugin** dialog. Upload the **sc_crdb_v1.x.x.x.json** file downloaded previously.



For more information, refer to the Continuous Data Engine connector management[312] section.

9.2.6.3 Upgrading the connector

The procedure for upgrading to a higher connector version is the same as installing a new CockroachDB connector. Repeat the previous steps using the .json file corresponding with the version being upgrade to.

9.2.6.3.1 After uploading the new connector version file

1. Refresh all (corresponding) environments.
2. Ensure a connector discovery is completed successfully.
3. Take a new snapshot of the dSource and VDB.

> While updating a plugin, disable & enable of datasets is not required. Only refresh of environments is required.

312 https://cd.delphix.com/docs/latest/delphix-engine-plugin-management

To upgrade to newer versions of the CockroachDB connector, the upgrade path as given below must be followed.

Upgrade Path

Released Versions	Recommended Upgrade Path
1.0.0, 1.1.0, 1.1.1, 1.1.2, 1.2.0	Supports Direct Upgrade to 1.2.0.

9.2.7 CockroachDB connector operations

To successfully ingest and virtualize a CockroachDB instance, you must perform the following connector operations in sequence:

- Provisioning a CockroachDB virtual database (see page 1100)
- Linking CockroachDB data sources (see page 1093)
- Discovering installations for CockroachDB (see page 1089)

The respective sections provide detailed information about prerequisites, limitations (if any), and the procedure to be followed for completing each of the operations.

9.2.7.1 Discovering installations for CockroachDB

Environment discovery/refresh is a process that enables the CockroachDB Connector to determine CockroachDB installation details on a host. Database discovery/refresh is initiated during the environment set up process.

Whenever there is any change (for example: installing a new database home) to an already set up environment in the Continuous Data application, an environment refresh must be performed.

- Prerequisite for the CockroachDB discovery operation (see page 1089)
- Adding an environment for CockroachDB (see page 1090)
- Add CockroachDB database (see page 1091)

9.2.7.1.1 Prerequisite for the CockroachDB discovery operation

The following requirements must be met before starting the discovery process. When you add an environment to Delphix Continuous Data Engine, CockroachDB Connector does an automatic discovery of all CockroachDB installed binaries on the server.

1. Login to the Staging Host as delphix os user.
2. Create a file name (if not present) named .bash_profile or .bashrc in home directory of delphix OS user.

```
touch $HOME/.bash_profile $HOME/.bashrc
```

3. Define the environment variable pointing to binary path of CRDB_REPO directory on the host for manual discovery of CockroachDB instances and other information. Example as below.

```
echo "CRDB_REPO=/u01/cockroach/cockroach" > $HOME/.bash_profile
```

4. In case of multiple CockroachDB versions installed on same host, provide all installations path separated with colon (:) to the environment variable defined in .bash_profile. Example as below.

```
echo "CRDB_REPO=/u01/cockroach/cockroach:/u02/cockroach/cockroach" >
$HOME/.ba
```

9.2.7.1.2 Adding an environment for CockroachDB

This topic describes how to add a CockroachDB Staging and target environment.

9.2.7.1.2.1 Prerequisites

To set up a CockroachDB staging environment, ensure all the essential requirements, as described in Requirements and Prerequisites (see page 1077) are met.

9.2.7.1.2.2 Procedure

Perform the following steps to add an environment to the Delphix Continuous Data Engine:

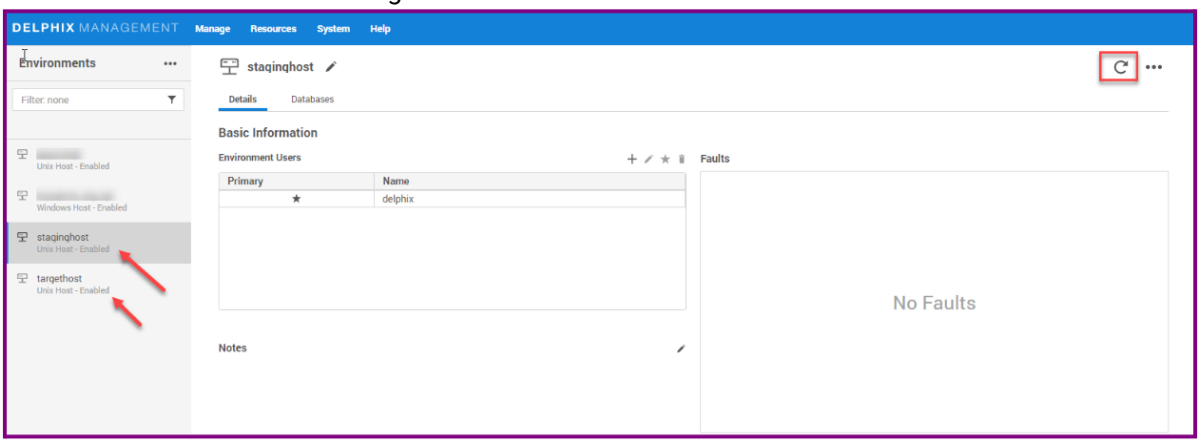
1. Login to the **Delphix Management** application.
2. Click **Manage** > **Environments**.
3. Click the **Actions ...** menu and select **Add Environment**.
4. From the Add Environment dialog window, select **Unix/Linux**.
5. Select **Standalone Host**
6. Click **Next**.
7. Provide the name of the environment.
8. Provide the **Host IP address** or **host FQDN**.
9. Provide the **SSH port**. The default value is 22.
10. Choose the **Login Type** that fits your setup.
    a. Username and Password – provide the environment username and password.
    b. Username and Public Key – provide the environment username.
    c. Password Vault – select from an existing Enterprise Password Vault.
11. Provide an **OS Username** for the environment.
12. If Applicable, Provide an **OS Password** for the environment..
13. For Password Login, click **Validate** to test the OS Username and OS Password.
14. Enter **Toolkit Path** (ensure that the toolkit path does not have any spaces).

15. To provide your own Oracle Java Development Kit, select the **Provide my own JDK** checkbox and click **Next**.
    a. On the **Java Development Kit** page, provide an absolute path to your Oracle JDK and click **Next**.
16. Click **Submit.**

9.2.7.1.2.3 Adding an Installation

If a new CockroachDB Installation is added to the environment and you want it to be reflected in the Continuous Data Engine UI, it is required to refresh the environment. Please refer to the below steps to perform "Environment Refresh" operation.

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Environments**.
4. In the **Environments** panel, click the name of the environment you want to refresh.
5. Select the **Refresh icon**.
6. In the Refresh confirmation dialog select **Refresh**.



9.2.7.1.3 Add CockroachDB database

As noted above, an environment contains repositories that are CockroachDB installations in the environment. Each environment may have any number of repositories associated with it.

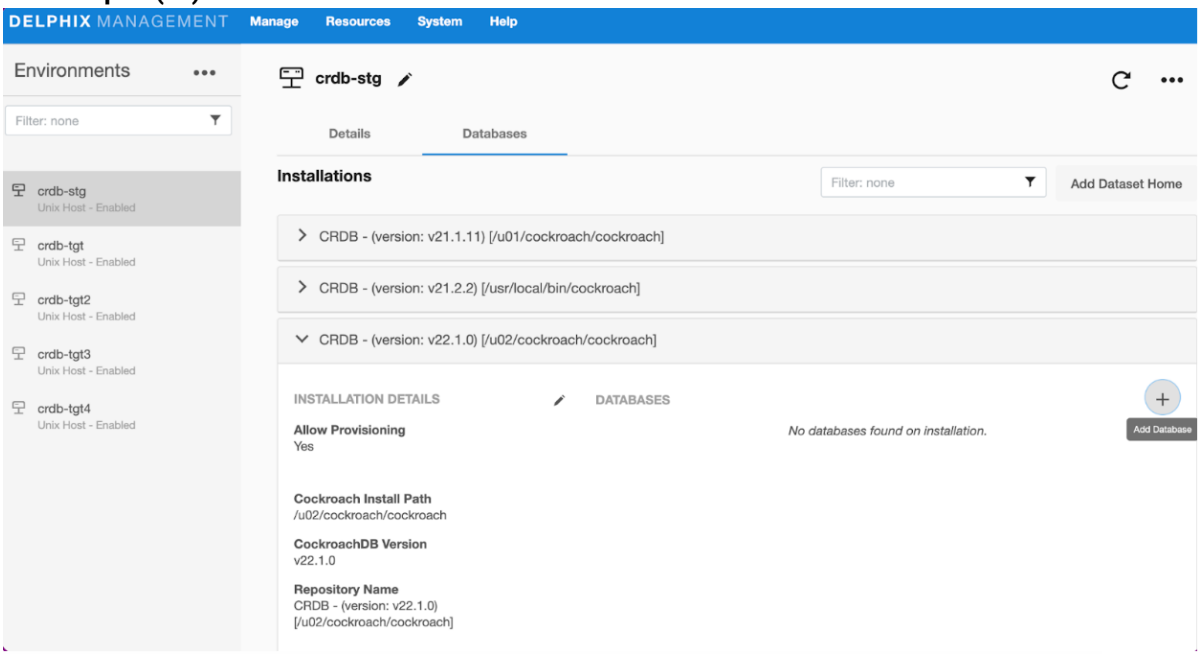
A SourceConfig object defines the configuration of the dSource and is required to create a dSource. You can create any number of SourceConfig objects using a repository, which represents known database instances.

For the CockroachDB plugin, the Source config must be created manually.

9.2.7.1.3.1 Procedure

Perform the following steps to create the Source Config on the Staging Environment:

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Environments**.
4. Select the repository.
5. Click on **plus(+)**icon.



6. Add required details in the Add **database** section.
7. **Identity field**: Provide unique name for staging database.
8. Click **Add** button.

9.2.7.2 Linking CockroachDB data sources

The Delphix Continuous Data Engine provides various ingestion methods for linking the dSource from a CockroachDB source database.

- Creating a dSource with ingestion type Seed (see page 1093)
- Creating a dSource with ingestion type AWS S3 (see page 1095)
- Creating a dSource with ingestion type Disk (see page 1098)

9.2.7.2.1 Creating a dSource with ingestion type Seed

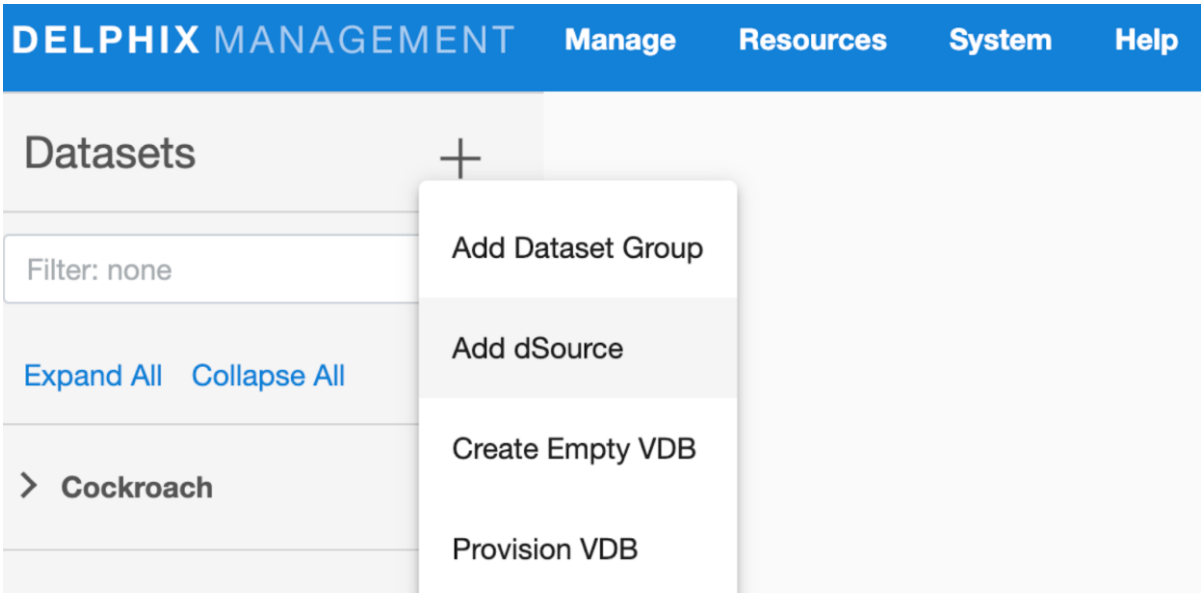
The following procedure provides details on creating a dSource with seed ingestion type.

9.2.7.2.1.1 Prerequisites

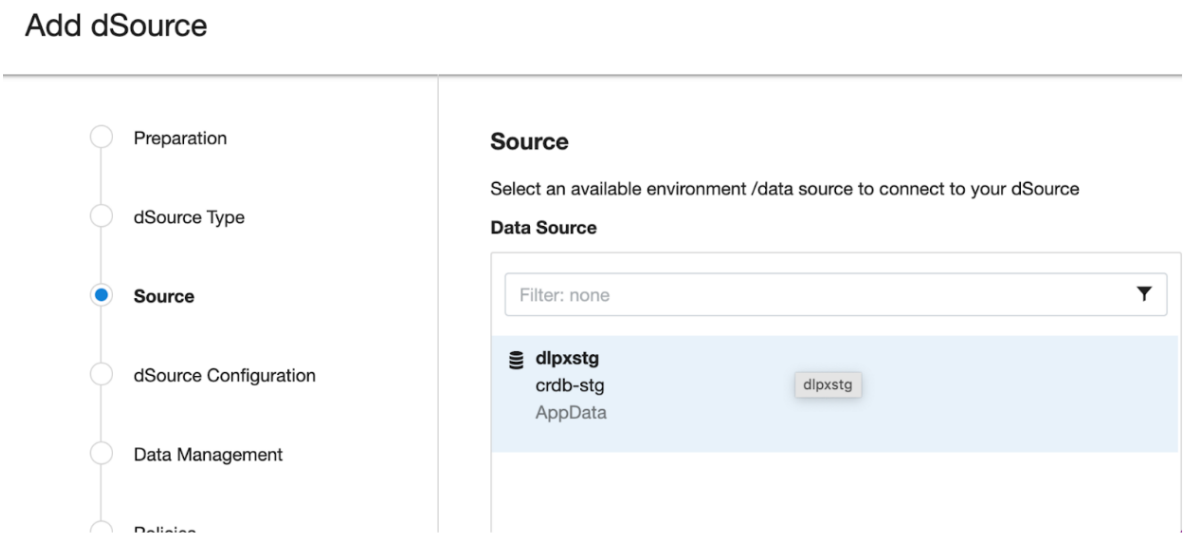
The source and staging instances must meet the host requirements as described in Requirements (see page 1079) for CockroachDB Staging/Target hosts.

9.2.7.2.1.2 Creating dSource

1. Login to **Delphix Management application**.
2. Click **Manage** > **Datasets**
3. Click on **Add dSource**.



4. In the Add dSource wizard, select the CockroachDB source config you just created on the staging host.



5. Select the dSource type from the drop-down available on dSource wizard. Refer different dSource Types[313].
6. Enter the CockroachDB specific parameters for your dSource configuration.
   a. **dSource Type** Type of dSource
   b. dSource (**Seed**) Details.
      i. **Mount Path** The path on the host where Delphix will mount the dSource files via NFS.
      ii. **IP Address / Hostname of the Staging Cluster** IP Address / Hostname of the interface for the staging Cluster (default 127.0.0.1).
      iii. **Staging CockroachDB Port Number** Staging DB port for cluster's inter-node communication.
      iv. **Staging CockroachDB Console Port Number** Staging Console port for HTTP requests.
      v. **Secure Mode** Option to run the CockroachDB server in secure mode.
      vi. **Certificate directory path** The absolute path to the certificate directory, it must have valid certificates containing the IP address or hostname for the CockroachDB server.
      vii. **CockroachDB Cluster Node Count** Number of nodes that consists of Staging CockroachDB cluster. Minimum: 3 and Maximum: 9
      viii. **CockroachDB Parameters** Additional CockroachDB configuration parameters to start the CockroachDB cluster.
         1. **Parameter Name** CockroachDB Cluster configuration parameter name
         2. **Parameter Value** CockroachDB Cluster configuration parameter value
      ix. Then click **Next** to proceed further.
7. Provide the details for **dSource Name** and **Target group** on the dSource configuration page and click **Next**.
8. On the Data management page, select the following and click **Next**.
9. Staging Environment: This will be your staging host where source config was created.
   a. User: Database OS user with required privileges for linking the dataset.

313 https://cd.delphix.com/docs/latest/dsource-management

b. On the next screens, configure a policy, hooks, and review the configuration, then click on **Next** button to view the summary.

10. Click the **Submit** button which will initiate the linking process

12. Once dSource is created successfully, you can review the datasets on **Manage** > **Datasets** > **dSource Name**.

Notes:

- Must specify the input "**Certificate directory path**" when the "**Secure Mode**" option is selected.
- For more detailed guidance on CockroachDB certificate generation, refer to the official documentation: CockroachDB Certificate Generation[314].

9.2.7.2.2 Creating a dSource with ingestion type AWS S3

The following procedure provides details on creating a dSource with AWS S3 ingestion type.

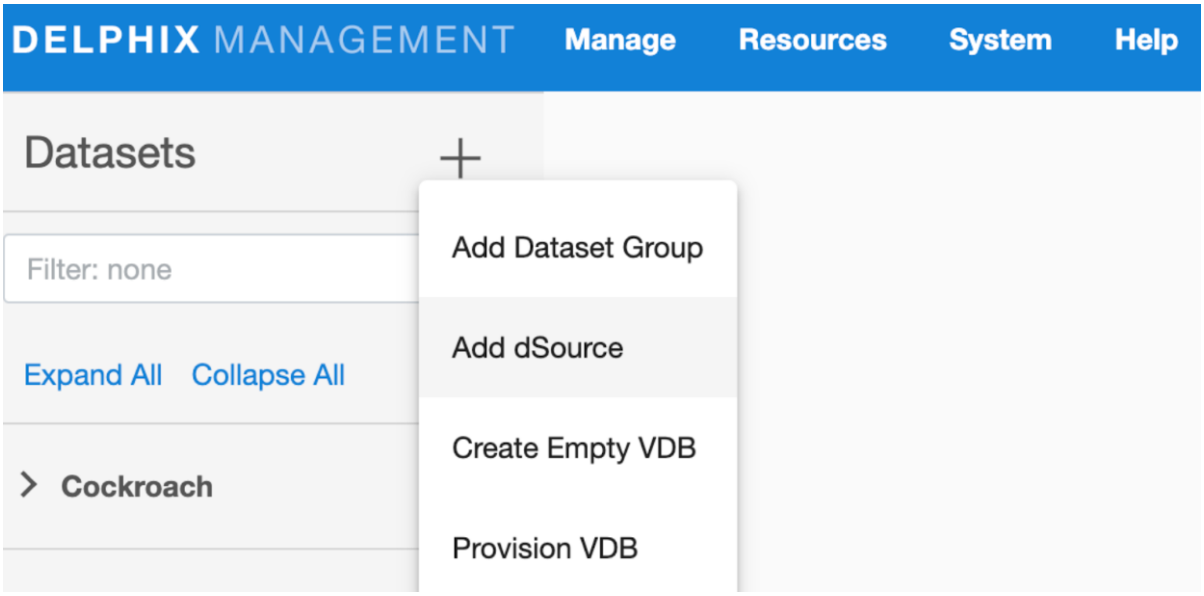
9.2.7.2.2.1 Prerequisites

- The source and staging instances must meet the host requirements as described in Requirements (see page 1079) for CockroachDB Staging/Target hosts.
- The accessibility of the backup location within the AWS S3 bucket from the Staging host must be verified prior to ingestion.

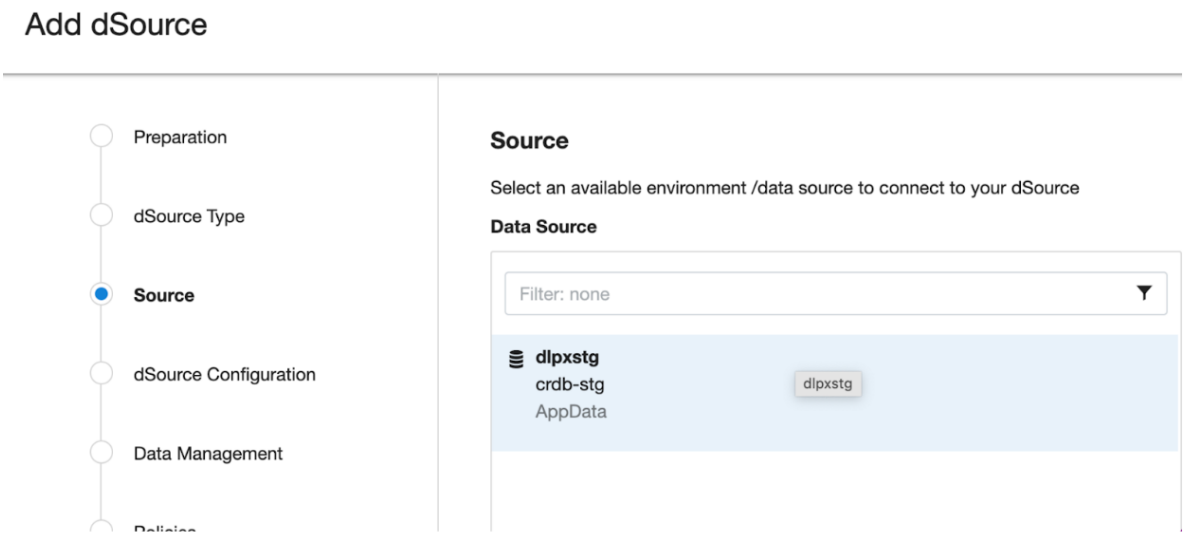
9.2.7.2.2.2 Creating dSource

1. Login to **Delphix Management application**.
2. Click **Manage** > **Datasets.**
3. Click on **Add dSource**.

314 https://www.cockroachlabs.com/docs/stable/cockroach-cert



4. In the Add dSource wizard, select the CockroachDB source config you just created on the staging host.



5. Select the dSource type from the drop-down available on dSource wizard. Refer different dSource Types[315]
6. Enter the CockroachDB specific parameters for your dSource configuration.
   a. **dSource Type** Type of dSource
   b. dSource (**AWS_S3**) Details
      i. **Mount Path** The path on the host where Delphix will mount the dSource files via NFS
      ii. **IP Address / Hostname of the Staging Cluster** IP Address / Hostname of the interface for the staging Cluster (default 127.0.0.1)

315 https://cd.delphix.com/docs/latest/dsource-management

iii. **Staging CockroachDB Port Number** Staging DB port for cluster's inter-node communication
iv. **Staging CockroachDB Console Port Number** Staging Console port for HTTP requests
v. **CockroachDB Backup's Subdirectory Path** CockroachDB Backup's Subdirectory Absolute Path where backups are located in S3 Bucket
vi. **CockroachDB Backup's Bucket or Container Name** Storage bucket or container name including absolute path where CockroachDB backups are located
vii. **Authentication Option for AWS S3 Storage Buckets**
    1. **Specified Authentication** Need to specify the AWS access key ID and secret access key in the dSource configuration fields:
        a. **Account Name or Access Key** Specify an account access name or key associated with object storage holding CockroachDB backups.
        b. **Account Secret Access Key** Specify the account secret access key associated with the account name field. This is essentially the password that gives access to CockroachDB backups.
    2. **Implicit Authentication** Store the AWS credentials as environment variables
        a. When using implicit authentication, ensure that aws credentials are configured on staging host using aws configure command.
        b. Configure an IAM role attached to the EC2 instance with the necessary permissions (e.g., s3:GetObject, s3:ListBucket, s3:PutObject) to access the S3 bucket where the backups are stored.
        c. Implement an S3 bucket policy to grant access to specific AWS accounts or IAM entities that need to restore the backup stored inside the S3 bucket. For example:

```
{
    "Version": "2012-10-17",
    "Id": "AWS_S3_Policy",
    "Statement": [
        {
            "Sid": "ExampleStatement01",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::22420070000:user/
delphix"
            },
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::dlpx-crdb-bucket",
                "arn:aws:s3:::dlpx-crdb-bucket/*"
            ]
        }
```

```
    ]
}
```

viii. **Skip Checking Localities** Select the checkbox to skip checking the localities of a cluster before a restore.
ix. **Secure Mode** Option to run the CockroachDB server in secure mode
x. **Certificate directory path** The absolute path to the certificate directory, it must have valid certificates containing the IP address or hostname for the CockroachDB server
xi. **CockroachDB Cluster Node Count** Number of nodes that consists of a Staging CockroachDB cluster. Minimum: 3 and Maximum: 9
xii. **CockroachDB Parameters** Additional CockroachDB configuration parameters to start the CockroachDB cluster
    1. **Parameter Name** CockroachDB Cluster configuration parameter name
    2. **Parameter Value** CockroachDB Cluster configuration parameter value
xiii. Then click **Next** to proceed further.

7. Provide the details for **dSource Name** and **Target group** on the dSource configuration page and click **Next**.
8. On the Data management page, select the following and click **Next**.
9. Staging Environment: This will be your staging host where source config was created.
    a. User: Database OS user with required privileges for linking the dataset.
    b. On the next screens, configure a policy, hooks, and review the configuration, then click on **Next** button to view the summary.
10. Click the **Submit** button which will initiate the linking process
11. Once dSource is created successfully, you can review the datasets on **Manage** > **Datasets** > **dSource Name**.

Note:

- You must specify the input "**Certificate directory path**" when "**Secure Mode**" option is selected.
- For more detailed guidance on CockroachDB certificate generation, refer to the official documentation: CockroachDB Certificate Generation[316].

9.2.7.2.3 Creating a dSource with ingestion type Disk

The following procedure provides details on creating a dSource with Disk ingestion type.

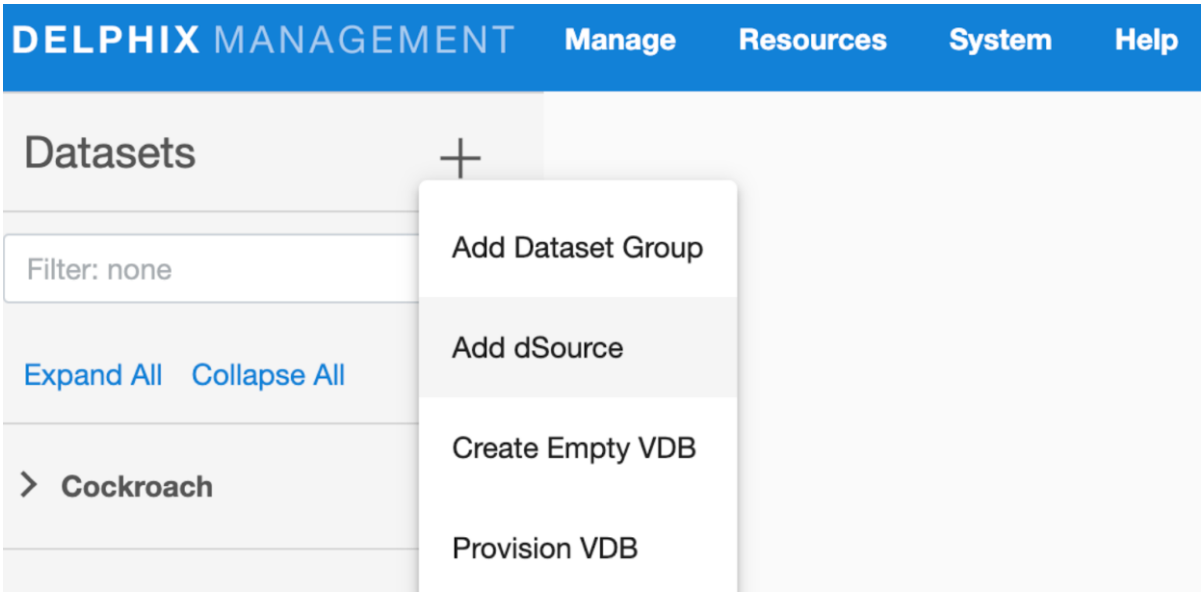
9.2.7.2.3.1 Prerequisites

The source and staging instances must meet the host requirements as described in Requirements (see page 1079) for CockroachDB Staging/Target hosts.

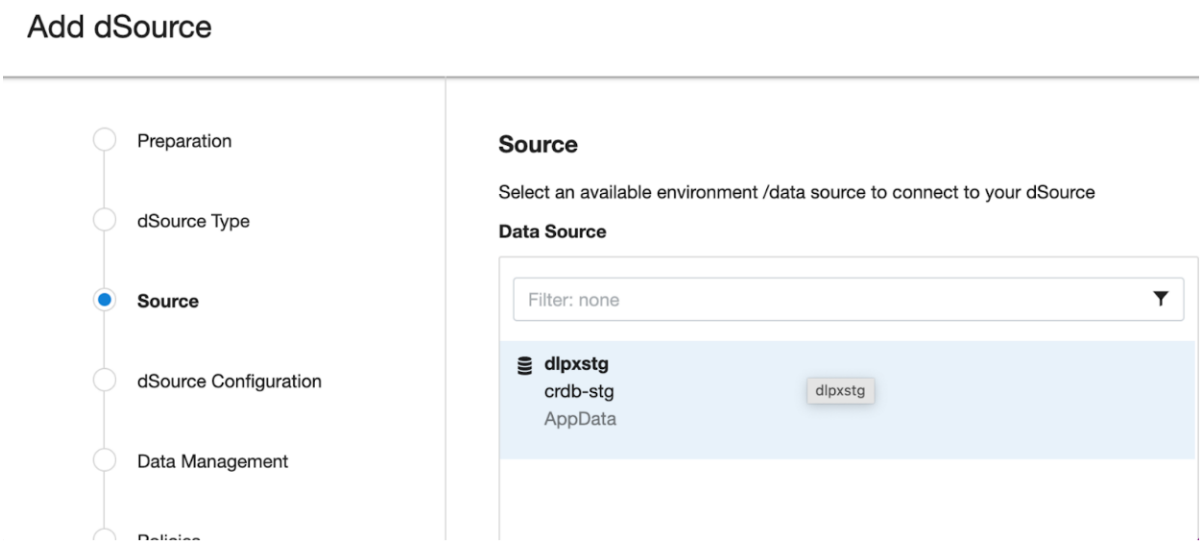
Creating dSource

316 https://www.cockroachlabs.com/docs/stable/cockroach-cert

1. Login to **Delphix Management application**.
2. Click **Manage** > **Datasets**
3. Click on **Add dSource**.



4. In the Add dSource wizard, select the CockroachDB source config you just created on the staging host.



5. Select the dSource type from the drop-down available on dSource wizard. Refer to different dSource Types[317]
6. Enter the CockroachDB specific parameters for your dSource configuration.
   a. **dSource Type** Type of dSource
   b. dSource (**AWS_S3**) Details

317 https://cd.delphix.com/docs/latest/dsource-management

i. **Mount Path** The path on the host where Delphix will mount the dSource files via NFS
ii. **IP Address / Hostname of the Staging Cluster** IP Address / Hostname of the interface for the staging Cluster (default 127.0.0.1)
iii. **Staging CockroachDB Port Number** Staging DB port for cluster's inter-node communication
iv. **Staging CockroachDB Console Port Number** Staging Console port for HTTP requests
v. **CockroachDB Backup Subdirectory Path** CockroachDB Backup's Subdirectory Absolute Path where backups are located in S3 Bucket
vi. **CockroachDB Backup's Directory Path** The absolute path on the staging host where CockroachDB backups are stored.
vii. **Secure Mode** Option to run the CockroachDB server in secure mode
viii. **Certificate directory path** The absolute path to the certificate directory, it must have valid certificates containing the IP address or hostname for the CockroachDB server
ix. **CockroachDB Cluster Node Count** Number of nodes that consists of Staging CockroachDB cluster. Minimum: 3 and Maximum: 9
x. **CockroachDB Parameters** Additional CockroachDB configuration parameters to start the CockroachDB cluster
    1. **Parameter Name** CockroachDB Cluster configuration parameter name
    2. **Parameter Value** CockroachDB Cluster configuration parameter value
xi. Then click **Next** to proceed further.

7. Provide the details for **dSource Name** and **Target group** on the dSource configuration page and click **Next**.
8. On the Data management page, select the following and click **Next**.
    a. Staging Environment: This will be your staging host where source config was created.
    b. User: Database OS user with required privileges for linking the dataset.
9. On the next screens, configure a policy, hooks, and review the configuration, then click on **Next** button to view the summary.
10. Click the **Submit** button which will initiate the linking process.
11. Once dSource is created successfully, you can review the datasets on **Manage** > **Datasets** > **dSource Name**.

Note:

- Must specify the input "**Certificate directory path**" when "**Secure Mode**" option is selected.
- For more detailed guidance on CockroachDB certificate generation, refer to the official documentation: CockroachDB Certificate Generation[318].

9.2.7.3 Provisioning a CockroachDB virtual database

To provision a virtual database (VDB), you can either use an existing dSource snapshot or use another VDB snapshot for VDB to VDB provisioning. A snapshot is a specific point in time on the timeflow.

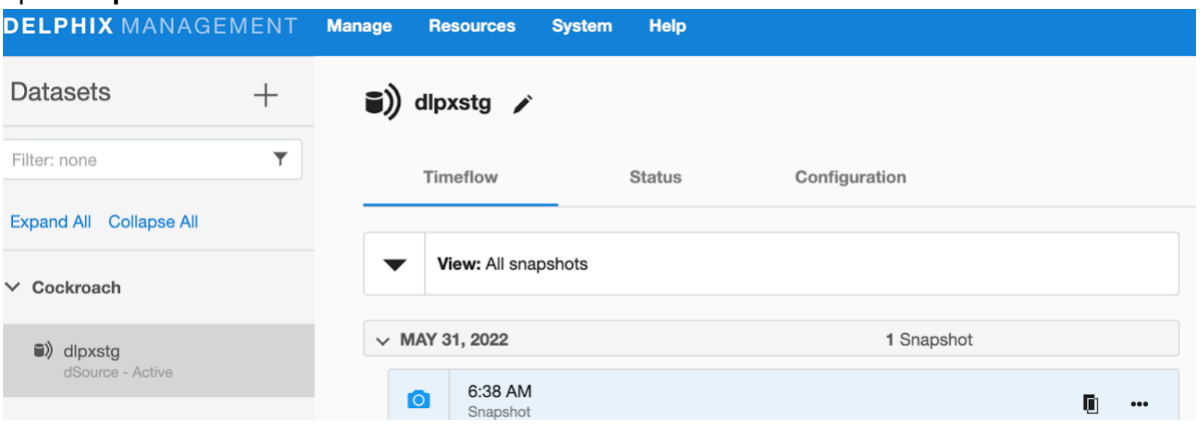
318 https://www.cockroachlabs.com/docs/stable/cockroach-cert

9.2.7.3.1 Prerequisites

- Must have linked a dSource from a staging instance following the instructions in "Linking a CockroachDB dSource (see page 1093)," or have already created a VDB from which you intend to provision another VDB.
- A CockroachDB target environment must be set-up with the necessary requirements as described in CockroachDB support and requirements (see page 1077).[319]
- A CockroachDB binary with the same version as the source database must be installed on the Target.

9.2.7.3.2 Procedure

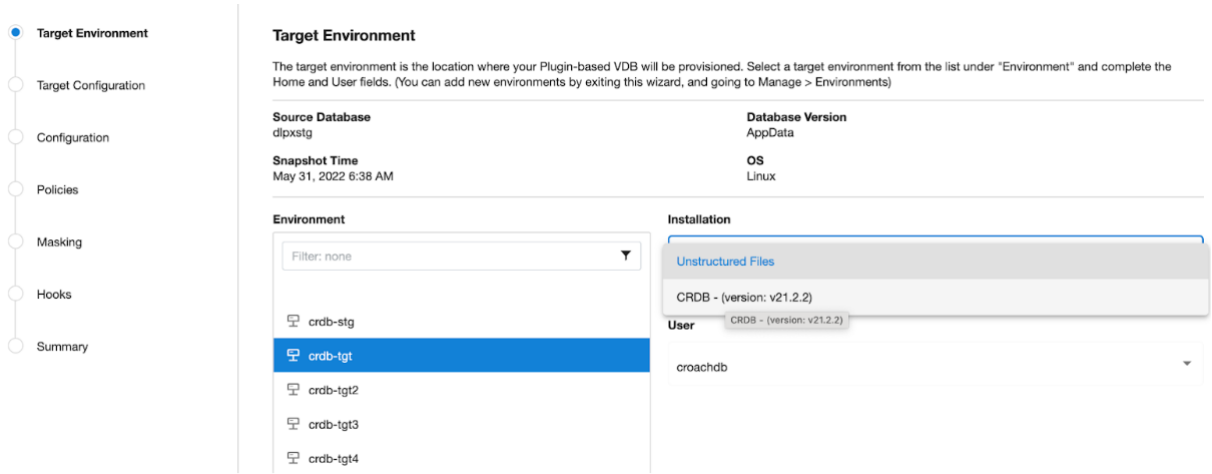
1. Navigate to **Manage**, and select **Datasets**.
2. Select the dataset and the snapshot that you want provision from. Click the provision VDB icon to open the **provision VDB** wizard.



3. Select a target environment from the left pane, and an Installation to use from the dropdown list of available CockroachDB instances on that environment.
4. Set the **Environment User** to be the Instance Owner.

> You can only select instance owners if you have multiple environment users set on that host.

319 https://cd.delphix.com/docs/latest/cockroachDB-requirements-and-prerequisites



5. Next, You will see the **Target Configuration** section where you need to enter VDB specific parameters.
    a. **Mount Path** Mount Path for VDB CockroachDB Cluster datafiles
    b. **IP Address / Hostname of the Primary Target Host** IP Address / Hostname of the interface for Primary Target Host
    c. **Target CockroachDB Port Number** Target DB port for cluster's inter-node communication
    d. **Target CockroachDB Console Port Number** Target Console port for HTTP requests
    e. **CockroachDB VDB Cluster Name** VDB Cluster Name to be configured on Target Host
    f. **Secure Mode** Option to run the CockroachDB server in secure mode
    g. **Certificate directory path** The absolute path to the certificate directory, it must have valid certificates containing the IP address or hostname for the CockroachDB server
    h. **CockroachDB Parameters** Additional CockroachDB configuration parameters to start the CockroachDB cluster
        i. **Parameter Name** CockroachDB Cluster configuration parameter name
        ii. **Parameter Value** CockroachDB Cluster configuration parameter value
    i. On the configuration page, select the Additional Nodes **(+Add)** button for adding new nodes to CockroachDB target cluster.
        i. **Environment** Select the target environment from drop-down to add the new CockroachDB node to cluster.
        ii. **Environment User** Select the target environment user to add the new CockroachDB node to cluster.
        iii. **IP Address / Hostname of the Additional Node** Target Host IP Address / Hostname of the interface for Secondary Target Host.
        iv. **Target CockroachDB Port Number** Target DB port for cluster's inter-node communication.
        v. **Target CockroachDB Console Port Number** Target Console port for HTTP requests.
        vi. **Certificate directory path** The absolute path to the certificate directory, it must have valid certificates containing the IP address or hostname for the CockroachDB server.
        vii. **Append CRDB Parameters** Option to append the existing primary node CRDB parameters to additional node.

j. Next, select **Additional Node CRDB Parameters** to overwrite additional node CockroachDB parameters configured in Target CockroachDB cluster. Click on **(+Add)** button to add CRDB parameters.
   i. **Parameter Name** CockroachDB Cluster configuration parameter name.
   ii. **Parameter Value** CockroachDB Cluster configuration parameter value.
k. Then click **Next** to proceed further.

6. On the Configuration page, Select the **VDB Name** and **Target Group** for the VDB.
7. Add a new group, if required.
8. Select a **Snapshot Policy** for the VDB.
9. Click **Next**.
10. Specify any desired hook operations.
11. Click **Next**.
12. Review the **Provisioning Configuration** and Data Management information.
13. Click **Submit**.

When provisioning starts, you can review the progress of the job in the Databases panel, or in the **Job History** panel of the Dashboard.

When the provisioning completes a new VDB timeflow will be created. Once the VDB is created successfully, you can review the datasets on the Manage > Datasets screen in the left hand panel. You have now provisioned a CockroachDB VDB. Moreover, the VDB will be included in the group you designated and listed in the Databases panel. If you select the VDB in the Databases panel and click the Open icon, you can view its card, which contains information about the database and its Data Management settings.

9.2.7.3.3 VDB Configuration Changes for CockroachDB

9.2.7.3.3.1 Prerequisites

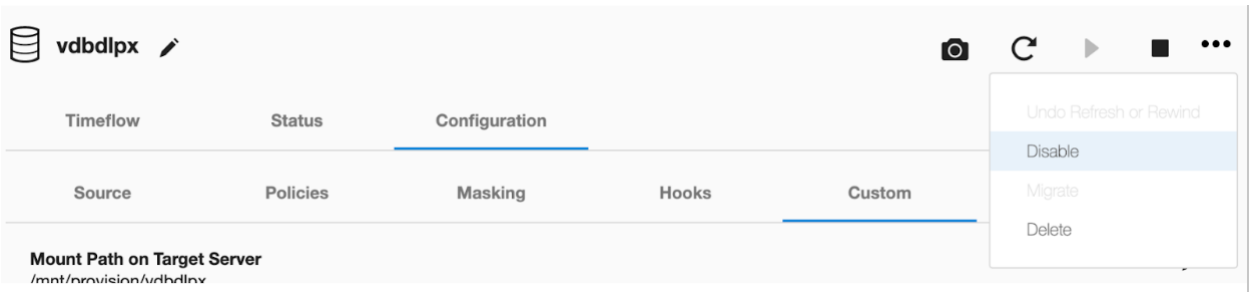
- Delphix highly recommends disabling the VDB before making any configuration changes to an existing CockroachDB Target cluster. This precautionary measure is advised due to the following reasons:
  - To ensure a smooth transition when altering the topology of the Cockroach cluster, such as scaling up or down the number of nodes, a graceful shutdown is required.
  - A service restart is necessary after applying any configuration changes to a CockroachDB target cluster.
  - From the Delphix workflow perspective, during the disable operation, cluster services are stopped, and the file system is unmounted. If changes to the configuration are made without disabling, it may lead to unforeseen issues when attempting to bring down the services.

Procedure

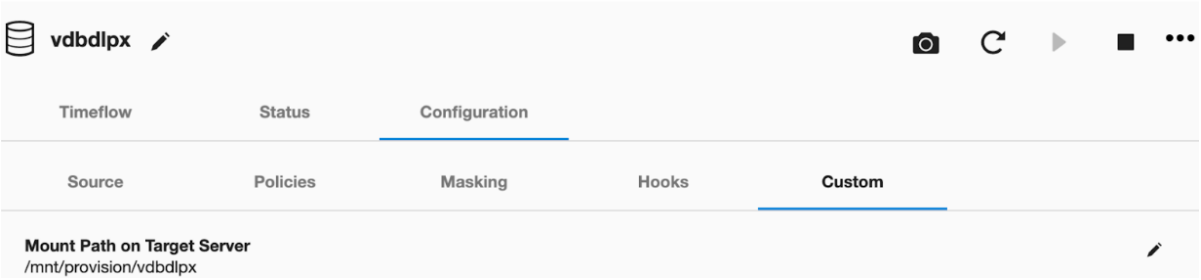
Steps to change the VDB configuration

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Datasets**.
4. Select the VDB for your CockroachDB target cluster.
5. From the Actions menu **(...)** select **Disable**.

> While updating a connector, it is not required to disable and re-enable datasets.

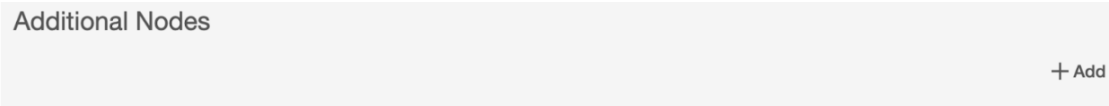


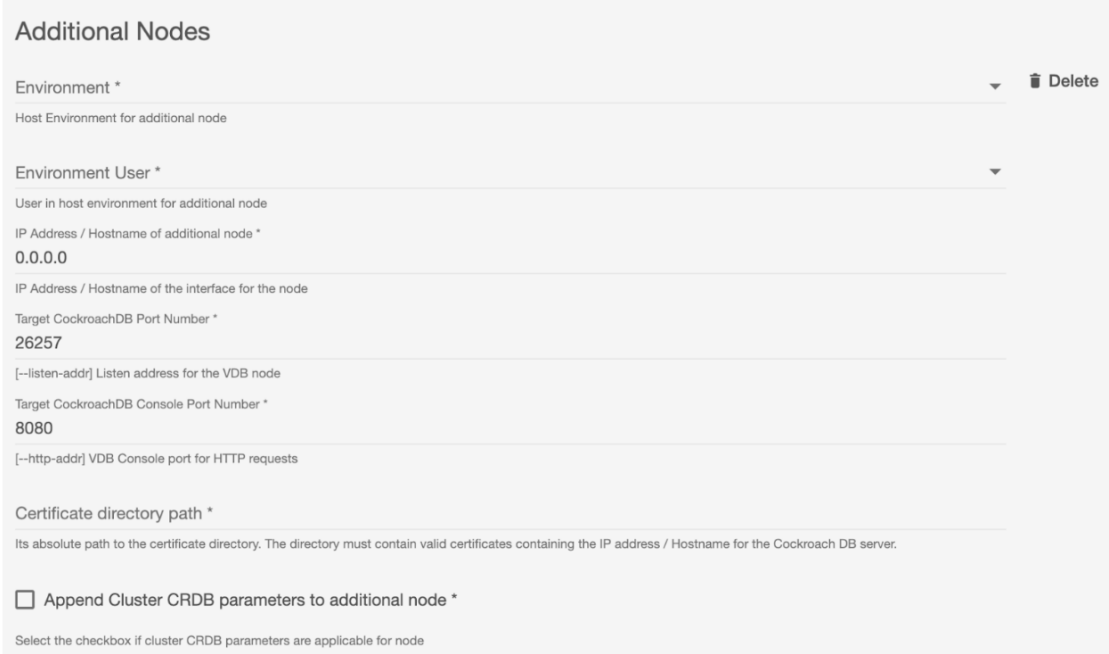
6. Select the VDB and click on **Configurations** -> **Custom** section. Click on edit (pencil) button.



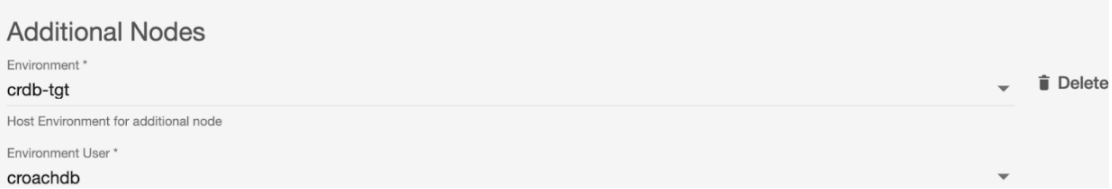
7. Modify the VDB specific parameters you wish to update for your CockroachDB target cluster.
   a. To update additional CRDB Parameters to CockroachDB Target Cluster.

   b. To add additional nodes to existing CockroachDB Target Cluster.



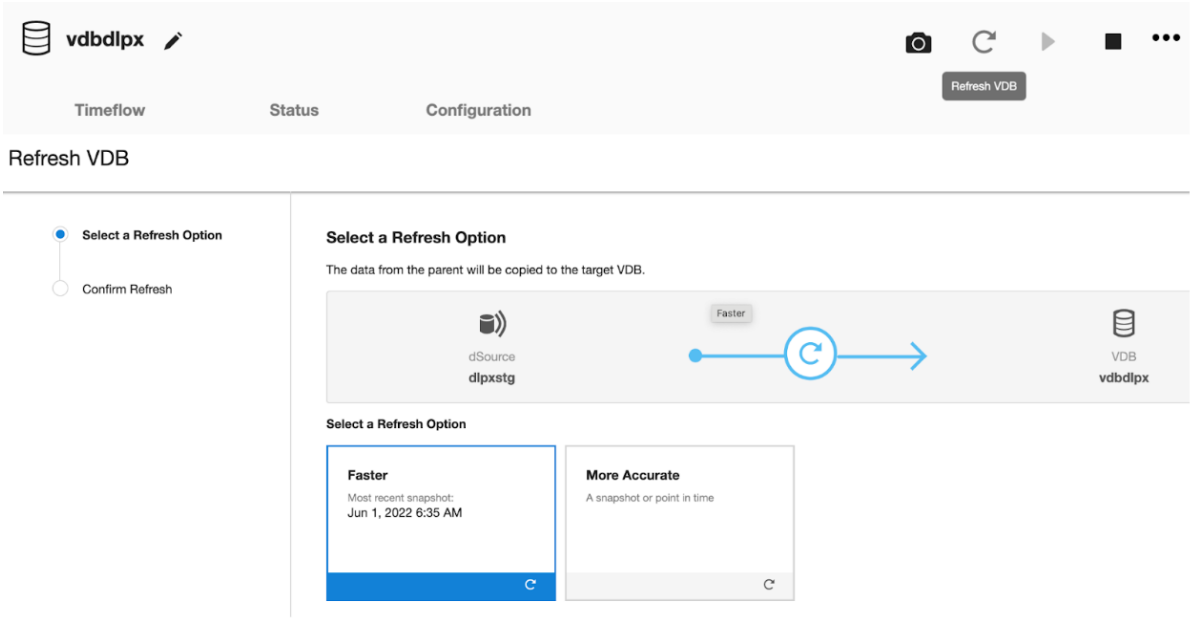


c. To Remove additional node from existing CockroachDB Target Cluster.



d. To overwrite additional node CRDB parameter for a specific node.

8. Apply the configuration changes by clicking the save button at the bottom.
9. Once the VDB configuration changes are saved. Click the Refresh button and select the Snapshot from which to refresh.



10. In the Refresh VDB dialog, confirm the Refresh by clicking the submit button.

9.2.7.3.4 Rewinding a Cockroach VDB

Rewinding a VDB rolls it back to a previous point in its Timeflow and re-provisions the VDB. The VDB will no longer contain changes after the rewind point.

9.2.7.3.4.1 Prerequisites

To rewind a VDB, you must have the following permissions:

A user with Continuous Data Engine Admin credentials can perform a VDB rewind on any VDB in the system.

9.2.7.3.4.2 Procedure

1. Login to the **Delphix Management** application.
2. Under **Datasets**, select the VDB you want to rewind.
3. On the **Timeflow** tab, select the rewind point as a snapshot or a point in time.
4. Click **Rewind**.
5. If you want to use login credentials on the target environment other than those associated with the environment user, click Provide Privileged Credentials.
6. Click **Yes** to confirm.

9.2.8 Troubleshooting CockroachDB data sources

The CockroachDB connector produces three sets of logs during normal runtime. Refer to the list below to understand what they are and how you can access them.

9.2.8.1 CockroachDB Plugin logs

1. The CockroachDB connector is built using the plugin framework. Therefore, it adheres to Delphix Continuous Data's standard plugin logging behavior. We recommend starting here for all troubleshooting. Refer to the Troubleshooting document (see page 1107) for more information.
2. Plugin logs for CockroachDB are found at the below locations.

```
/{toolkit_directory}/Delphix_COMMON_{appliance_id}/plugin/sc:cockroach_{plugin-
id}/CRDB/log_{os_user}/delphixCRDBDebug.log
```

9.2.8.2 CockroachDB Cluster logs

1. CockroachDB can write error messages to log files. The files are named using the following format:

```
cockroach.[host].[user].[start timestamp in UTC].[process ID].logCBBACKUPMGR.
```

2. To make it easier to watch a log without knowing the full filename, a symlink with the short filename cockroach.log is also created. `{delphix_mount_path}/node<number>/logs/cockroach.log`
3. This symlink points to the most recent log. The Cockroach log locations on staging or target host can be found at below locations:

```
{CockroachDB_install_path} version --build-tag
For example:
/usr/local/bin/cockroach version --build-tag
v21.2.2
```

4. Start CockroachDB multi-node cluster.

```
cockroach start --certs-dir=[path-to-certs-directory]
--advertise-addr=<node1 address>
--join=<node1 address>,<node2 address>,<node3 address>
--cache=.25 --max-sql-memory=.25
```

5. List CockroachDB certificate and keys.

```
cockroach cert list --certs-dir=[path-to-certs-directory]
```

6. Initialize a Cluster from any CockroachDB node.

```
cockroach init --certs-dir=[path-to-certs-directory]
--host=<address of any node on --join list>
```

7. Start interactive CockroachDB SQL shell.

```
cockroach sql --certs-dir=[path-to-certs-directory]
--host=<address of any cluster-node>:<cockroach_db_port>
```

> Use the *--certs-dir* option with the valid path to the certificate directory of the CockroachDB cluster.

8. Find full backup subdirectories details.

```
SHOW BACKUPS IN 's3://{bucket name}/{path}?AWS_ACCESS_KEY_ID={placeholder}
&AWS_SECRET_ACCESS_KEY={placeholder}';
```

9. Find full and incremental backups in a specific full backup subdirectory details.

```
SHOW BACKUP FROM LATEST IN 's3://{bucket name}?AWS_ACCESS_KEY_ID={placeholder}
&AWS_SECRET_ACCESS_KEY={placeholder}';
```

10. Create CockroachDB User to login into CockroachDB console to access secure clusters.

```
create user dlpxadmin WITH PASSWORD 'dlpxadmin';
```

11. Grant access to CockroachDB User to access data.

```
GRANT SELECT, INSERT ON TABLE vehicles to dlpxadmin;
```

9.2.9 CockroachDB developer's guide

- CockroachDB linking operations (see page 1109)
- CockroachDB provision operations (see page 1111)
- CockroachDB discovery operations (see page 1109)

- CockroachDB sourceconfig operations (see page 1109)

9.2.9.1 CockroachDB discovery operations

To discover an environment via CLI, login to the Delphix Continuous Data Engine using the admin user account.

```
ssh admin@<CONTINUOUS-DATA-FQDN-or-IP>
```

Discovering the environment

Run the following commands to discover the environment.

```
Example -
environment
create
set hostEnvironment.name=crdb-stg
set hostParameters.host.address=crdb-stg.dcol2.delphix.com
set hostParameters.host.toolkitPath=/home/delphix/toolkit
set primaryUser.name=delphix
set primaryUser.credential.password=delphix
commit
```

9.2.9.2 CockroachDB sourceconfig operations

Run the following commands to add the source config:

```
Example -
sourceconfig create
set type="AppDataStagedSourceConfig"
set name="dlpxstg"
set environmentUser="crdb-stg/delphix"
set parameters="{\"cluster_name\":\"dlpxstg\"}"
set repository="'CRDB - (version: v22.2.14) [/u03/cockroach/cockroach]'"
```

9.2.9.3 CockroachDB linking operations

9.2.9.3.1 dSource creation via CLI

To create a dSource via CLI, you must create a source configuration first. For more information on creating a source configuration, refer to Add Source Config using CLI (see page 1109).

To create a dSource and perform operations via CLI, login to the Delphix Continuous Data Engine using the admin user account.

```
ssh admin@<CONTINUOUS-DATA-FQDN-or-IP>
```

Set default linking parameters[This is common for all types of ingestions]:

Run the below commands to set the default linking parameters. The below example uses **testscfg** source config.

```
Example -
database link
set type=LinkParameters
set linkData.type=AppDataStagedLinkData
set name=dlpxstg
set group=cockroach
set linkData.config=dlpxstg
set linkData.environmentUser=crdb-stg/delphix
set linkData.stagingEnvironment=crdb-stg
set linkData.stagingEnvironmentUser=crdb-stg/delphix
set linkData.syncParameters.parameters={\"resync\":true}
```

Link parameters for Seed ingestion type:

Run the below commands to link parameters for seed ingestion type.

```
Example -
set linkData.parameters="{\"dsource_type\":\"Seed\",
\"mount_path\":\"/mnt/provision/dlpxstg\",
\"ip_addr\":\"10.43.35.72\",\"cockroach_port\":26257,
\"cockroach_db_console_port\":8080,\"cert_dir\":\"/home/delphix/toolkit\",
\"cluster_config\":3,\"crdb_parameters\":[{\"property_name\":\"max-sql-memory\",
\"value\":\"0.25\"}]}"
```

Link parameters for AWS_S3 ingestion type:

Run the below commands to link parameters for AWS_S3 ingestion type.

```
Example -
set linkData.parameters="{\"dsource_type\":\"Seed\",\"mount_path\":\"/mnt/provision/
dlpxstg-s3\",
\"ip_addr\":\"10.43.34.131\",\"cockroach_port\":36257,
\"cockroach_db_console_port\":9080,\"backup_piece\":\"/2022/04/12-105100.23\",
\"object_storage\":\"dlpx-crdb-backup/crdb_full\",\"account_name\":
\"****************\",\"account_secret_access_key\":\"*****************\",
\"enable_secure_mode\":true,\"cert_dir\":\"/home/delphix/toolkit\",
\"cluster_config\":3,\"crdb_parameters\":[{\"property_name\":\"max-sql-memory\",
\"value\":\"0.25\"}]}"
```

Link parameters for Disk ingestion type:

Run the below commands to link parameters for Disk ingestion type.

```
Example -
set linkData.parameters="{\"dsource_type\":\"Seed\",
\"mount_path\":\"/mnt/provision/dlpxstg-disk\",
\"ip_addr\":\"10.43.34.131\",\"cockroach_port\":56257,
\"cockroach_db_console_port\":5080,\"backup_piece\":\"/2023/10/04-150211.01\",
\"staging_host_backup_path\":\"/home/delphix/backups\",
\"enable_secure_mode\":true,\"cert_dir\":\"/home/delphix/toolkit\",
\"cluster_config\":3,\"crdb_parameters\":[{\"property_name\":\"max-sql-memory\",
\"value\":\"0.25\"}]}"
```

Commit

Run the below command to commit the change

```
commit;
```

Disable

Run the below command to disable the dSource.

```
Example -
source
select dlpxstg
disable
commit
```

Enable

Run the below command to enable the dSource.

```
Example -
source
select dlpxstg
enable
commit
```

9.2.9.4 CockroachDB provision operations

To create a VDB via CLI, you must first create a dSource. For more information on creating a dSource, refer to *Link operation via CLI*[320] section (see page 1109).

To create a VDB and perform operations via CLI, login to the Delphix Continuous Data Engine using the admin user account.

```
ssh admin@<CONTINUOUS-DATA-FQDN-or-IP>
```

320 https://cd.delphix.com/docs/latest/cockroachDB-discovery-operations

Provision

Run the below commands to provision a VDB. The below example uses **dlpxstg-s3** dSource.

```
Example -
database provision
set type=AppDataProvisionParameters
set container.name=dlpxtgt-s3
set container.group=cockroach
set sourceConfig.repository=APPDATA_REPOSITORY-9
set timeflowPointParameters.type=TimeflowPointSnapshot
set timeflowPointParameters.snapshot=APPDATA_SNAPSHOT-3
set sourceConfig.name=dlpxstg-s3
set source.allowAutoVDBRestartOnHostReboot=false
set source.parameters="{\"mount_path\":\"/mnt/provision/dlpxtgt-s3\",
\"ip_addr\":\"10.43.35.72\",\"cockroach_port\":26257,
\"cockroach_db_console_port\":8080,\"cockroach_cluster_name\":\"dlpxtgt-s3\",
\"enable_secure_mode\":true,\"cert_dir\":\"/home/delphix/certs\",
\"crdb_parameters\":[{\"property_name\":\"max-sql-memory\",
\"value\":\"0.25\"},{\"property_name\":\"cache\",
\"value\":\"0.25\"}]}"
commit
```

Disable

Run the below commands to disable a dSource. You must select the same sourceConfig name that was given during VDB creation.

```
Example -
source
select testscfgv
disable
commit
```

Enable

Run the below command to enable a dSource.

```
Example -

source
select testscfgv
enable
commit
```

Stop

Run the below command to stop a dSource.

```
Example -

source
select testscfgv
stop
commit
```

Start

Run the below command to start a dSource.

```
Example -

source
select testscfgv
start
commit
```

9.3 Couchbase data sources

This section contains the following pages that will help with creating Couchbase data sources (dSources) in Delphix Continuous Data.

- Couchbase virtualization process (see page 1113)
- Couchbase requirements and prerequisites (see page 1114)
- Couchbase installation and upgrade (see page 1122)
- Couchbase connector operations (see page 1124)
- Couchbase troubleshooting (see page 1141)
- Couchbase developer's CLI guide (see page 1141)
- Couchbase data source limitations (see page 1150)

9.3.1 Couchbase virtualization process

This page serves as a quick reference overview of the steps used to perform an implementation of the CouchbaseDB Connector. Each step briefly describes the task and the expected result. Refer to the respective documentation links for more detailed information.

1. Begin by reviewing the general Delphix Continuous Data architecture diagrams[321]. Identify existing or new CouchbaseDB source, staging, and target environments, the underlying operating systems, and the CouchbaseDB database versions. This provides a good understanding of the architectural strategy to confirm support.
2. Review the Couchbase requirements and prerequisites[322]. Please verify the CouchbaseDB Connector complies with all system prerequisites, including those of Continuous Data, staging/target

321 https://cd.delphix.com/docs/latest/general-architecture
322 https://cd.delphix.com/docs/latest/couchbase-requirements-and-prerequisites

environments, Operating Systems, and CouchbaseDB database versions. Verifying this information is essential to align with the features and functionalities of the connector.

3. Install the CouchbaseDB Connector[323]. The CouchbaseDB Connector establishes the connection between the Delphix Continuous Data Engine and the connector version. It is recommended to stay up-to-date with the latest version.
4. Adding staging and target environments[324].[325] The staging environment will host a database that the Delphix Continuous Data Engine keeps in sync with the source environment database, at an interval chosen by the user. The target environment will provide the infrastructure to generate virtual database (VDB) copies.
    a. Consult the Couchbase host environment requirements[326] documentation for configuration details.
5. Adding a Couchbase installation to an environment[327]. To enable the Delphix Continuous Data Engine to recognize the provided infrastructure, search for a CouchbaseDB installation to make available for database linking and provisioning.
6. Linking the CouchbaseDB database[328]. This configures a dSource that enables the syncing mechanism between the source database, staging database, and the Delphix Continuous Data Engine. There are many ingestion methods that differ between connectors and solve different requirements. It is best to identify the ingestion method that best suits your organization and infrastructure policies.
7. Provisioning Couchbase VDBs[329]. Administrators and Self-Service users can provision one or more copies of their virtual databases (VDBs).

9.3.2 Couchbase requirements and prerequisites

9.3.2.1 Requirements for source databases

1. For **XDCR** ingestion:
    a. Source database admin credentials with the following privileges:
        i. XDCR_ADMIN
        ii. DATA_MONITOR
2. For **Couchbase backup manager** ingestion:
    a. There are no specific requirements for the source database, as the connector does not establish a connection with it.

---

323 https://cd.delphix.com/docs/latest/couchbase-installation-and-upgrade
324 https://cd.delphix.com/docs/latest/adding-a-couchbase-staging-target-environment
325 https://docs.google.com/document/d/19UPeBn_JQqaH09ofS5RcXdpH4viprJ_KfIP-y8avkoY/edit#heading=h.2jok6zkqgsem
326 https://cd.delphix.com/docs/latest/couchbase-host-environment-requirements
327 https://cd.delphix.com/docs/latest/adding-a-couchbase-installation-to-an-environment
328 https://cd.delphix.com/docs/latest/linking-couchbase-data-sources
329 https://cd.delphix.com/docs/latest/provisioning-couchbase-vdbs

9.3.2.2 Requirements for staging and target databases

The staging and target environments have the same set of requirements for CouchbaseDB.

9.3.2.3 Staging prerequisites for ingestion using backups

The backup can exist on the staging server or can be accessed over NFS.

For performing the backup operation, the following commands must be executed. This example uses a backup location of */u01/couchbase_backup* and the repository name *delphix*.

- Create a config file using the following command, this file will be required at the time of dSource creation using `CBBACKUPMGR`.

```
/opt/couchbase/bin/cbbackupmgr config --archive /u01/couchbase_backup --repo
delphix
```

- Backup from the source host to the backup directory (archive) mentioned in the previous step.

```
/opt/couchbase/bin/cbbackupmgr backup -a /u01/couchbase_backup -r delphix -c
couchbase://<production server> -u user -p password
```

9.3.2.4 Couchbase support matrices

9.3.2.4.1 OS and Database compatibility

Supported OS version	**Supported DBMS versions**			
	Couchbase 6.0.x	Couchbase 6.5.x	Couchbase 6.6.x	Couchbase 7.1.x
RHEL 6.x	Y	Y	Y	Y
RHEL 7.x	Y	Y	Y	Y
RHEL 8.x	Y	Y	Y	Y
Windows	N	N	N	N

9.3.2.4.1.1 Engine and Connector compatibility

Delphix Engine versions	Supported Couchbase connector versions	
	Version 1.2.1	version 1.2.2
6.0.7.x - 6.0.11.x	Y	N
6.0.12.x - 6.0.15.x	Y	N
7.x - 16.x	Y	Y

9.3.2.4.1.2 Database and Connector compatibility

Couchbase version	Supported Couchbase connector versions	
	Version 1.2.1	Version 1.2.2
Couchbase 6.0.x	Y	Y
Couchbase 6.5.x	Y	Y
Couchbase 6.6.x	Y	Y
Couchbase 7.1.x	N	Y

9.3.2.4.1.3 Database and Database Edition compatibility

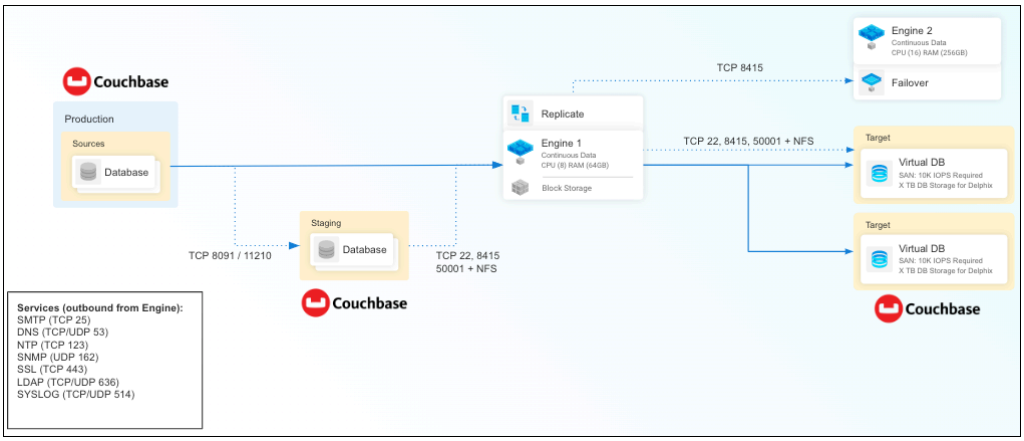
CouchbaseDB version	Couchbase Community Edition (XDCR Only)	CouchbaseDB Enterprise Edition
Couchbase 6.0.x	Y	Y

Couchbase 6.5.x	Y	Y
Couchbase 6.6.x	Y	Y
Couchbase 7.1.x	Y	Y

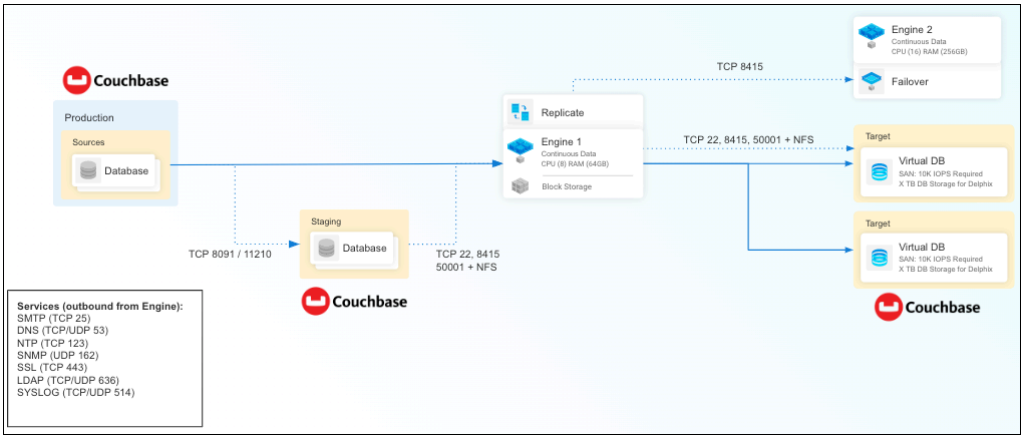
9.3.2.5 Couchbase network and connectivity requirements

Ensure that the requirements for Delphix Engine network connectivity[330] are met before proceeding further. This page addresses CouchbaseDB connector specific network and connectivity requirements.

9.3.2.5.1 Ports for XDCR



9.3.2.5.2 Ports for Ingestion using backups



330 https://cd.delphix.com/docs/latest/network-connectivity-requirements

9.3.2.5.3 SSH connectivity to staging and target host

1. Staging and target environments are required to have `sshd` running and configured for Continuous Data to connect over SSH.
2. The Delphix platform expects to maintain long-running, highly performant SSH connections with the host environment.
3. Depending on the **ingestion type**, once connected to a target environment through SSH, the Continuous Data initiates a Couchbase client connection (couchbase-cli) from the staging environment to the source environment.

9.3.2.5.4 Whitelisting the inbound port for SSH connection to the target host

9.3.2.5.4.1 Creating NFS mounts on the target host

Delphix Continuous Data Engine shares its storage space with the target host using NFS mounts.

Protocol	Port Number	Use
TCP	22	SSH connection to the target database environment.
TCP/UDP	111	Remote Procedure Call (RPC) port mapper used for NFSv3 mounts.
TCP	1110	Network Status Monitor (NSM) client from target hosts to the Continuous Data Engine.
TCP	2049	NFS client from target hosts to the Continuous Data Engine (NFSv3 and NFSv4).
TCP	54043	Client mount daemon (NFSv3 only).
TCP	54044	Lock state notification service (NFSv3 only).
TCP	54045	Network Lock Manager (NLM) client from target hosts to Continuous Data Engine (NFSv3 only).
UDP	33434 - 33464	Traceroute from the target database server to the Continuous Data Engine (optional).

9.3.2.5.4.2 Connecting to CouchbaseDB instances (Source, dSource, and VDBs)

Outgoing (from)	Incoming (to)	Protocol	Port number	Use
Staging host	Source CouchbaseDB Instance	TCP	<source-port>	CouchbaseDB Source database instance port to connect to setting up the replication pipeline. The default port for a CouchbaseDB instance is 8901.
End-user hosts	Staging / Target CouchbaseDB instance	TCP	<target-port>	Connecting to the created database instance on the target host.

9.3.2.6 Couchbase host environment requirements

Staging/target host requirements	Explanation
Host compatibility check The operating system and architecture of the target host must match the source environment.	It is required that the source and the target environments should have identical OS and hardware configurations.
There must be a directory on the staging/target host where the Couchbase Connector support files can be installed (for example, **/var/tmp**) with the following properties: 1. The directory must be writable by the operating system user mentioned above. 2. The directory must have at least 1.5 GB of available storage.	This path is called the “Toolkit Path” in the Environment creation UI page.

CouchbaseDB installation and compatibility check There must be an installation of CouchbaseDB on the staging/target host that is compatible with the installation of CouchbaseDB on the source host.	The CouchbaseDB binary version on the staging/target must be identical to the source side.
There must be a mount point directory (for example, **/mnt**) that will be used as the base for mount points that are created when provisioning a VDB with the following properties: 1. The mount point directory must be writable by the operating system user mentioned above. 2. The mount point directory should be empty.	

Hostname and IP must be correctly set in **/etc/hosts** file.	The output of the hostname `-i` command should produce the correct result as the IP address of the server.
	For example:
	`[root@source ~]# hostname -i`
	`10.110.211.225`

Required utilities and tools Ensure the expect utility is installed on the staging/target host.	CouchbaseDB Connector runs password sensitive commands by making use of expect utility. Checking if the expect utility is installed with: `which expect`
Storage requirement	1. The Staging Host must have enough storage space to hold the source backup file OR NFS mount Source backup filesystem. 2. An empty folder on the staging host to hold the delphix toolkit (approximately 2GB free space).

9.3.2.6.1 Sudo privileges for the environment OS user

Continuous Data Engine connects to the target host (via SSH) to majorly perform the following operations:

1. Continuous Data dynamically mounts and unmounts directories for linking and provisioning operations. This privilege is required because mount and unmount are typically reserved for superuser.

2. Continuous Data dynamically creates and removes directories for linking and provisioning operations.

The operations above will require sudo privileges for the OS user.

Configuring sudo access on target environments (as root user):

1. The sudo configuration exists in the **/etc/sudoers** file. The file can be edited directly or a utility called `visudo` can be used to add the privilege.

```
vi /etc/sudoers
# Or
visudo
```

2. Adding the privileges

```
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD: /bin/mount,/bin/umount,/bin/mkdir,/bin/rmdir, /bin/ps
delphix_os ALL=(couchbase) NOPASSWD: ALL
```

> The examples used above are for illustrative purposes and the sudo file configuration options are subject to change.

9.3.2.6.1.1 Requiretty settings

Continuous Data requires that the requiretty setting be disabled for all Continuous Data users with sudo privileges.

It is required to specify the NOPASSWD qualifier within the sudo configuration file. This ensures the `sudo` command does not demand the entry of a password, even for the display permissions (i.e. `sudo -l`) command.

- Additionally, Delphix OS user should be:
  - Regular OS user (e.g., delphix_os)
  - Add OS user to couchbase OS group
  - Must have access to backup files of the source database stored on the staging host.
  - Must have access to the path create and modify files in `/opt/couchbase/var/lib/couchbase`.
  - An empty folder with approximately 2GB of free space on the host, to hold Delphix Toolkit (dxtoolkit).

    - An empty folder on the host to mount the NFS filesystem. This is just an empty folder with no space requirements and acts as a base folder for NFS mounts.
    - Sudo privileges for mount and umount. The sample above uses delphix_os as delphix user.
    - If Couchbase service is installed using a Couchbase user, the Delphix OS user (e.g., delphix_os) must have privilege to run any command as a Couchbase user, using sudo.
- Please ensure any login banners similar to the one in the image below must be disabled on the staging host.
- Couchbase binaries installed and configured:
    - Disable auto start using OS services:
        - `systemctl disable couchbase-server.service`
        - `systemctl stop couchbase-server.service`

```
[staging_host]$ su - delphix
*******************************************************
*
*             Welcome to the Linux Terminal
*
*-----------------------------------------------------
*
*  You are now logged in as: delphix
*
*  Take caution with your actions and be responsible!
*
*******************************************************
Password:
```

9.3.3 Couchbase installation and upgrade

9.3.3.1 Prerequisites

- Consult the Couchbase support matrices[331] page to ensure your Delphix Continuous Data Engine is on a version compatible with the Connector version of choice.
- Install Couchbase binaries on the source, staging, and target servers.

9.3.3.2 Downloading the connector

The connector is provided as a downloadable .json file from the Delphix download site. Use the steps below to download it:

1. Navigate to the Delphix Download portal by clicking the following link or entering https://download.delphix.com/ into the address bar.
2. Login to the download portal using the same email and password credentials used for Delphix Support.
3. Navigate to **Delphix Product Releases** → **Continuous Data Connectors** → **Couchbase** → **v1.x.x.x** -> **sc_couchbase_v1.x.x.x.json** and the download should automatically begin.

331 https://cd.delphix.com/docs/latest/couchbase-support-matrices

9.3.3.3 Installing the connector

1. Login to the **Delphix Management** application.
2. Navigate to **Manage** → **Plugins**.
3. Click the **+** icon to upload the connector file.
4. This step shows the **Upload or upgrade a plugin** dialog. Upload the **sc_couchbase_v1.x.x.x.json** file downloaded previously.

For more information, refer to the Continuous Data Engine connector management[332] section.

9.3.3.4 Upgrading the connector

The procedure for upgrading to a higher connector version is the same as installing a new CouchbaseDB connector. Repeat the previous steps using the .json file corresponding with the version being upgrade to.

9.3.3.4.1 After uploading the new connector version file

1. Refresh all (corresponding) environments.
2. Ensure a connector discovery is completed successfully.
3. Take a new snapshot of the dSource and VDB.

> While updating a connector, it is not required to disable and re-enable datasets.

To upgrade to newer versions of the CouchbaseDB connector, the upgrade path as given below must be followed.

Connector upgrade path: 1.2.0 -> 1.2.2

9.3.3.5 Upgrading Couchbase version on source, staging, and target host

If the Couchbase version at the staging or target host changes, the staging database and target database have to be recreated.

> If the existing staging database or target database is refreshed with an upgraded installation, the database might not function properly.

332 https://cd.delphix.com/docs/latest/delphix-engine-plugin-management

9.3.4 Couchbase connector operations

To successfully ingest and virtualize a Couchbase instance, you must perform the following connector operations in sequence:

1. Discovering installations[333]
2. Linking data sources[334]
3. Provisioning virtual databases[335]

The respective sections provide detailed information about prerequisites, limitations (if any), and the procedure to be followed for completing each of the operations.

9.3.4.1 Discovering Couchbase installations

Environment discovery (or refresh) is a process that enables the Couchbase Connector to discover Couchbase installations on a host and determine their details. Database discovery is initiated during the environment setup process. Whenever there is any change (installing a new database home) to an environment that's already setup in the Delphix application, an environment refresh must be performed.

9.3.4.1.1 Adding a Couchbase staging/target environment

This page describes the steps for adding a CouchbaseDB staging and target environment.

9.3.4.1.1.1 Prerequisites

- A Couchbase staging environment must be setup with the necessary requirements, as described in Couchbase requirements and prerequisites[336].
- Installation of the Couchbase Connector is required before the discovery.
- Install Couchbase binaries on the staging and target servers.
    - Couchbase installed at the path `/opt/couchbase`.
    - Environment variable `$COUCHBASE_PATH` should be set on the staging/target host (which contains the binary path of Couchbase) for the user who added the environment (in our case, delphix_os).

9.3.4.1.1.2 Procedure

Perform the following steps to add an environment (staging/target) to the Delphix Continuous Data Engine:

1. Login to the **Delphix Management** application.

333 https://cd.delphix.com/docs/latest/adding-a-couchbase-installation-to-an-environment
334 https://cd.delphix.com/docs/latest/linking-couchbase-data-sources
335 https://cd.delphix.com/docs/latest/provisioning-couchbase-vdbs
336 https://cd.delphix.com/docs/latest/couchbase-requirements-and-prerequisites

2. Navigate to the **Manage** → **Environments** page.
3. Click the **Actions** ··· menu and select **Add Environment**.
4. From the **Add Environment** dialog window, select **Unix/Linux.**
5. Select **Standalone Host**.
6. Click **Next**.
7. Provide the name of the environment.
8. Provide the **Host IP address** or **host FQDN**.
9. Provide the **SSH port**. The default value is 22.
10. Provide an **Environment Username** for the environment.
11. Choose the **Login Type** that fits your setup.

- Username and password: provide the environment username and password.
- Username and public key: provide the environment username.
- Password Vault: select from an existing Enterprise Password Vault.

> **Using Public Key authentication**
> If you want to use public key authentication for logging into your Unix-based environment, there are two options:
>
> 1. Use the Delphix Continuous Data Engine's key pair. For more information on how to do this, refer to the How to use the Continuous Data Engine's key-pair[337] section.
> 2. Provide a key pair for this environment. As an alternative, a key pair specific for this environment can be provided via the API or CLI. For instructions, refer to CLI Cookbook: Setting up SSH key authentication for UNIX environment users[338].

12. For Password, enter the password associated with the user in step 11.
13. If you want to use Public Key Encryption for logging into your environment:
    a. Select Public Key for the Login Type
    b. Click View Public Key
    c. Copy the public key that is displayed, and append it to the end of your `~/.ssh/authorized_keys` file. If this file does not exist, it must be created.

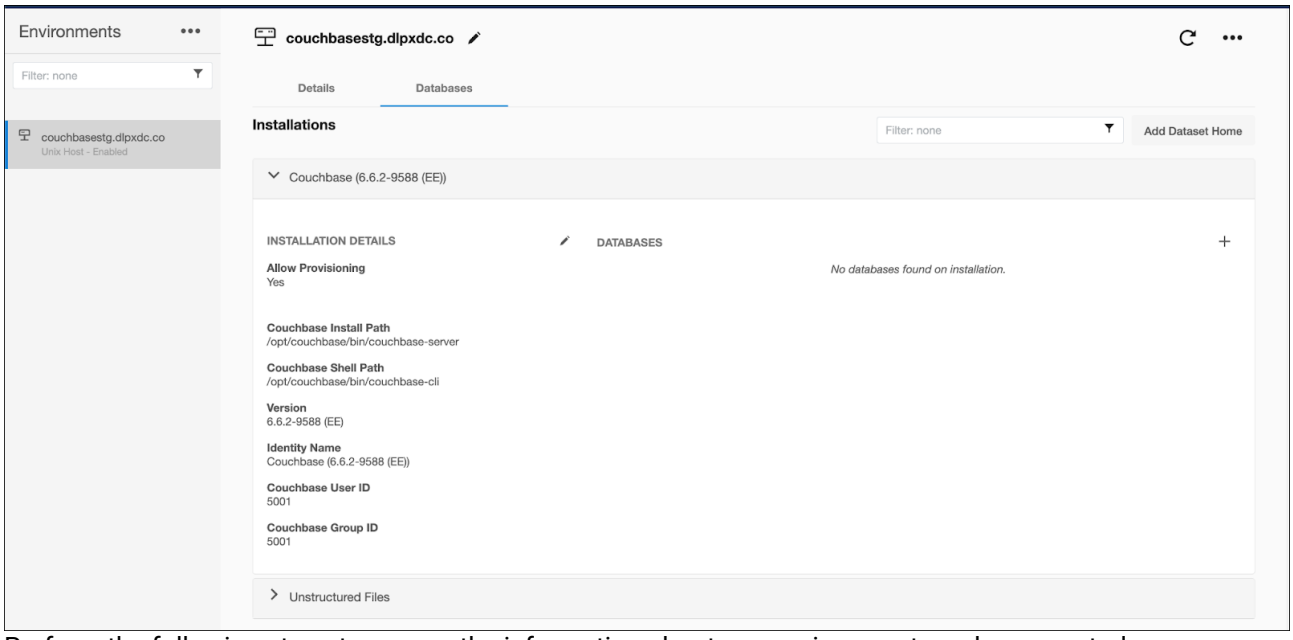
337 https://cd.delphix.com/docs/latest/how-to-use-the-delphix-continuous-data-engine-key-
338 https://cd.delphix.com/docs/latest/cli-cookbook-setting-up-ssh-key-authentication-for

> The addition of the public key is a one-time requirement per user and per environment. Additionally, you have the option to add public key authentication to an environment user's profile via CLI (for details, refer to the CLI Cookbook: Setting up SSH key authentication for UNIX environment users[339] topic).

14. For Password Login, click Verify Credentials to test the username and password.
15. Enter **Toolkit Path** (ensure that the toolkit path does not have any spaces).
16. To provide your own Oracle Java Development Kit, select the **Provide my own JDK** checkbox and click **Next**.
    a. On the Java Development Kit page, provide an absolute path to your Oracle JDK and click **Next**.
17. Click **Submit**.

After successfully adding and discovering the environment, you should be able to view the newly added environment listed under the Environments tab. In case it does not appear, click the refresh icon in your browser.

After the environment has been successfully discovered, if you wish to establish further links, add a source configuration to the previously discovered installation.



Perform the following steps to access the information about any environment you have created:

1. Navigate to **Manage** > **Environments**.
2. Select the environment name.

339 https://cd.delphix.com/docs/latest/cli-cookbook-setting-up-ssh-key-authentication-for

9.3.4.1.2 Adding a Couchbase installation to an environment

If a new CouchbaseDB Installation is added to the environment, then in order to reflect it into the Continuous Data UI, it is required to refresh the environment. Please refer to the "Environment Refresh" and “prerequisites” section at Managing environments and hosts[340] for more details.

9.3.4.1.3 Adding a Couchbase SourceConfig

An environment contains repositories that are CouchbaseDB installations in the environment. Each environment may have any number of repositories associated with it.

A SourceConfig object defines the configuration of the dSource and is required to create a dSource. You can create any number of SourceConfig objects using a repository, which represents known database instances.

For the CouchbaseDB connector, the Source config must be created manually.

9.3.4.1.3.1 Procedure

Source Config is created on the Staging Environment

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Environments**.
4. Select the **repository**.
5. Click on plus **+** icon.
6. Add the required details in the **Add** database section.
7. Enter the **port number** in the **Source Couchbase** port section.
8. Enter the **source host address** in the **Source Host** (this will be used as the Source cluster for XDCR replication).
9. Enter the **unique name** for the **staging database** in the **Identify Field** section.
10. Enter the **Couchbase data path** of the **staging host** in **Database Data Path** section. The DB data path is the designated directory where Couchbase Server stores essential data files, encompassing data files, metadata, logs, and internal database files.
11. Click the **Add** button to complete.

9.3.4.2 Linking Couchbase data sources

Linking a data source will create a dSource object on the engine and allow Delphix to ingest data from this source. The dSource is an object that the Delphix Virtualization Engine uses to create and update virtual copies of your database.

Note: An environment can be associated with only one dSource/VDB at a time

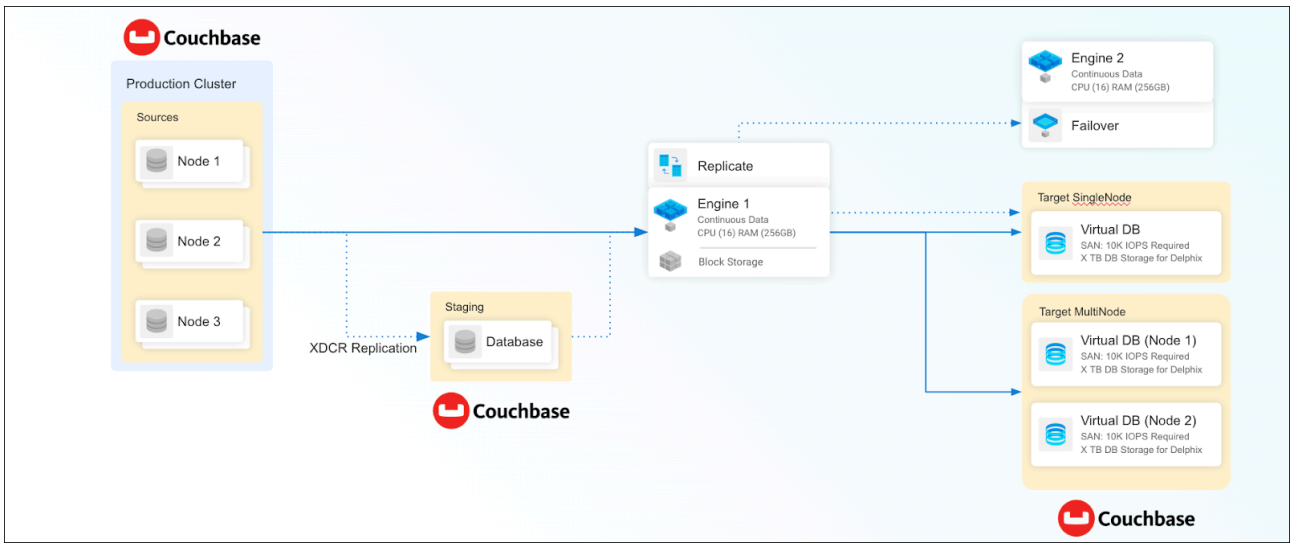
340 https://cd.delphix.com/docs/latest/managing-environments-and-hosts

9.3.4.2.1 Prerequisites

Staging environment created and source database configured in discovered repository.

9.3.4.2.2 Creating a Couchbase dSource using XDCR

The image below is an illustrated representation of the architecture of Continuous Data Engine support for Couchbase databases using XDCR.



9.3.4.2.2.1 Steps to create the dSource

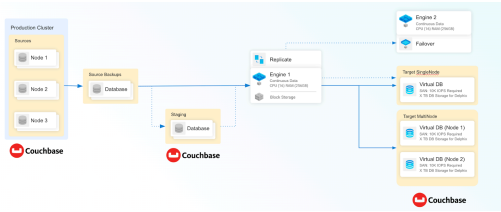
1. Login to **Delphix Management** application.
2. Click **Manage** → **Datasets**.
3. Select **Add dSource**.
4. In the Add dSource wizard, select the Couchbase source configuration which is created on the staging host.
5. Enter the Couchbase-specific parameters for your dSource configuration.
6. Select the dSource type XDCR from the drop-down available on dSource wizard.
7. Enter the details for **Staging Couchbase host** - FQDN or IP address recommended.
8. Enter the details for **Staging Port Number** available on the staging host. The default port for couchbase is 8091.
9. Enter the details for **Mount Path** available on the staging host. This empty folder acts as a base for NFS mounts.
10. Enter the details for **Staging Cluster Name** to setup new cluster on the staging host.
11. Enter the configuration details for your staging cluster as per resource availability on the staging host.

- Cluster RAM Size
- Cluster Index RAM Size
- Cluster FTS RAM Size
- Cluster Eventing RAM Size - this should be 0
- Cluster Analysis RAM Size - this should be 0

12. Enter the details for **Staging Cluster Admin User** and **Staging Cluster Admin Password**
13. Enter the details for **Source Cluster Admin User** and **Source Cluster Admin Password**
14. Enter Bucket Size. A bucket size of zero means the connector will extract bucket size from the source database and apply the same in staging. Giving it any other value, would mean imposing that value on all buckets irrespective of the size in the source database.
15. If not all, buckets need to be replicated, click on **+** plus symbol to modify configuration settings. Mention bucket list for which cross datacenter replication (XDCR) only be enabled.
16. Enter the details of **Bucket Name** to be part of XDCR. Then click **Next**.
17. Provide the details for **dSource Name** and **Target group** on the dSource configuration page.
18. On the **Data management** page, select the following:
    - Staging Environment: This will be your staging host where source config was created.
    - User: Database OS user with required privileges for linking the dataset.
19. On the next screens, configure a policy, hooks, and review the configuration, then click on **Next** button to view the summary.
20. Click the **Submit** button which will initiate the linking process.
21. Once dSource is created successfully, you can review the datasets on **Manage** > **Datasets** > **dSource Name**.

9.3.4.2.3 Creating a Couchbase dSource using an offline backup (Couchbase Backup Manager)

Below is the brief architecture of Continuous Data Engine support for Couchbase database



Prerequisites: Access to production backup e.g.,:

/backup - archive

PROD - repository

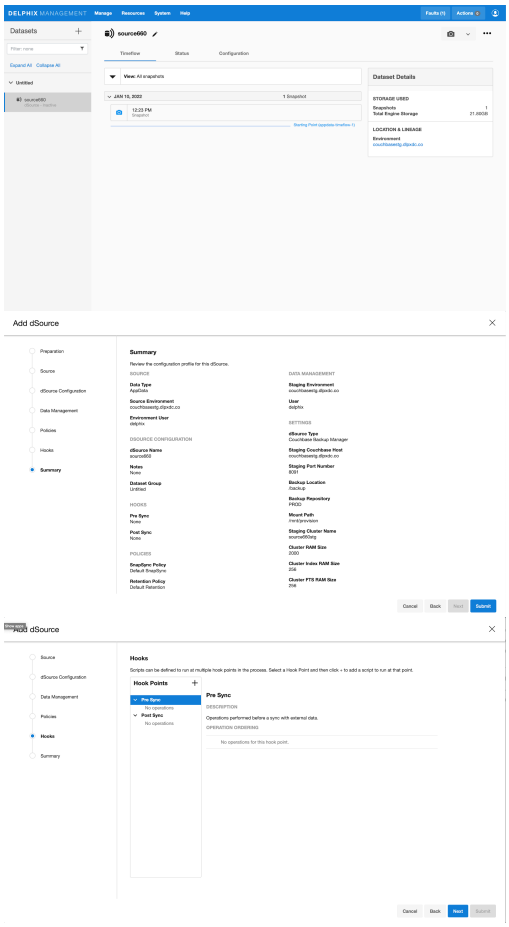
/opt/couchbase/bin/cbbackupmgr info -a /backup -r PROD

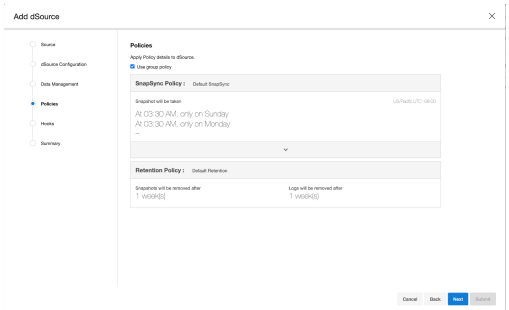
```
 Name  | Size     | # Backups  |
 PROD  | 58.03MB  | 1          |
```

```
+ Backup                    | Size    | Type  | Source                    | Cluster UUID                | Range  |
Events  | Aliases  | Complete  |
+ 2022-01-10T11_29_26.465860528-05_00  | 58.03MB  | FULL  | http://couchbasesrc.dlpxdc.co:8091  |
08f7937a26b2d20178a5ed16d7a2dd1c  | N/A    | 0      | 0       | true     |
```

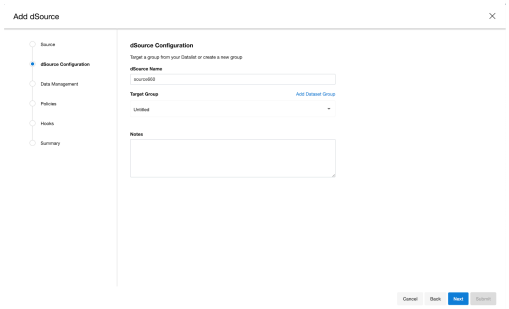
1. Login to **Delphix Management** application.
2. Click **Manage** > **Datasets**.
3. Select **Add dSource**.
4. In the Add dSource wizard, select the Couchbase source configuration which is created on the staging host.
5. Enter the Couchbase-specific parameters for your dSource configuration.
6. Select the dSource type **Couchbase Backup Manager** from the drop-down available on dSource wizard.
7. Enter the details for **Staging Couchbase host** - FQDN or IP address recommended.
8. Enter the details for **Staging Port Number** available on the staging host. The default port for couchbase is 8091.
9. Enter the details for **Backup Location** available on the staging host. For instance, if your Backup Path is specified as "/u01/couchbase/backups/," the corresponding **Backup Location** should be set as "/u01/couchbase".
10. Enter the details for **Backup Archive Name**.
    - If Backup Path is specified as "/u01/couchbase/backups/," the corresponding **Backup Archive Name** should be set as "backups".
    - If the **Backup Archive Name** field left empty then it will automatically ingest new full backups. The plugin will search for backup folders in the format %Y%m%d%H%M%S within **Backup Location**.
11. Enter the details for **Backup repository**
12. Enter the details for **Mount Path** available on the staging host. This empty folder acts as a base for NFS mounts.
13. Enter the details for **Staging Cluster Name** to setup new cluster on the staging host.
14. Enter the configuration details for your staging cluster as per resource availability on the staging host.
    - Cluster RAM Size
    - Cluster Index RAM Size
    - Cluster FTS RAM Size
    - Cluster Eventing RAM Size - this should be 0
    - Cluster Analysis RAM Size - this should be 0
15. Enter the details for **Staging Cluster Admin User** and **Staging Cluster Admin Password.**

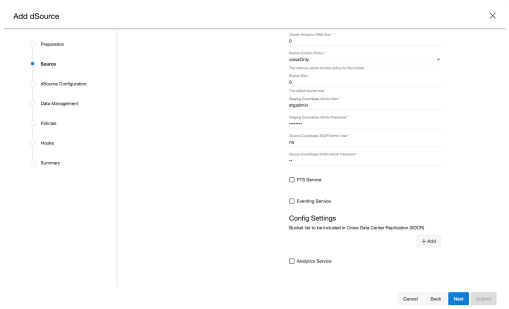
16. Enter **Bucket Size**. A bucket size of zero means the connector will extract bucket size from the source database backup and apply the same in staging. Giving it any other value, would mean imposing that value on all buckets irrespective of the size in the source database.
17. Then click on **Next** button
18. Provide the details for **dSource Name** and **Target group** on the dSource configuration page.
19. On the **Data management** page, select the following:
    - Staging Environment: This will be your staging host where source config was created.
    - User: Database OS user with required privileges for linking the dataset.
20. On the next screens, configure a policy, hooks and review the configuration and click on **Next** button to view the summary.
21. Click the **Submit** button which will initiate the linking process.
22. Once dSource is created successfully, you can review the datasets on **Manage** > **Datasets** > **dSource Name**.

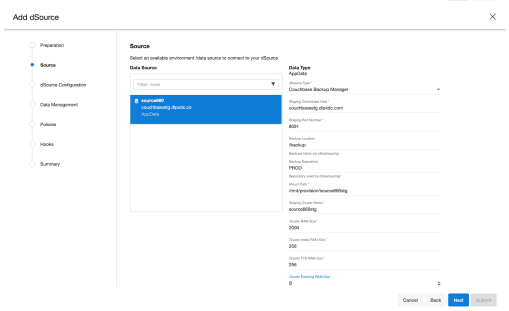



Add dSource
Source
dSource Configuration
Data Management
Policies
Hooks
Summary
Policies
Apply Policy details to dSource.
Use group policy
SnapSync Policy : Default SnapSync
Snapshot will be taken
At 03:30 AM, only on Sunday
At 03:30 AM, only on Monday
Retention Policy : Default Retention
Snapshots will be removed after
1 week(s)
Logs will be removed after
1 week(s)
Cancel
Back
Next
Submit

Add dSource
Source
dSource Configuration
Data Management
Policies
Hooks
Summary
Data Management
Configure and Administer data details.
Staging Environment
User
delphix
Cancel
Back
Next
Submit


Add dSource
Source
dSource Configuration
Data Management
Policies
Hooks
Summary
dSource Configuration
Target a group from your Datalist or create a new group
dSource Name
Target Group
Add Dataset Group
Untitled
Notes
Cancel
Back
Next
Submit


Add dSource
Preparation
Source
dSource Configuration
Data Management
Policies
Hooks
Summary
FTS Service
Eventing Service
Config Settings
Analytics Service
Cancel
Back
Next
Submit


Add dSource
Preparation
Source
dSource Configuration
Data Management
Policies
Hooks
Summary
Source
Select an available environment /data source to connect to your dSource
Data Source
Data Type
AppData
Cancel
Back
Next
Submit

9.3.4.3 Provisioning Couchbase VDBs

9.3.4.3.1 Provisioning a Couchbase virtual database

To provision a virtual database (VDB), you can either use an existing dSource snapshot or use another VDB snapshot for VDB to VDB provisioning. A snapshot is a timeflow created in a dSource or VDB.

> An environment can be associated with only one dSource/VDB at a time.

9.3.4.3.2 Prerequisites

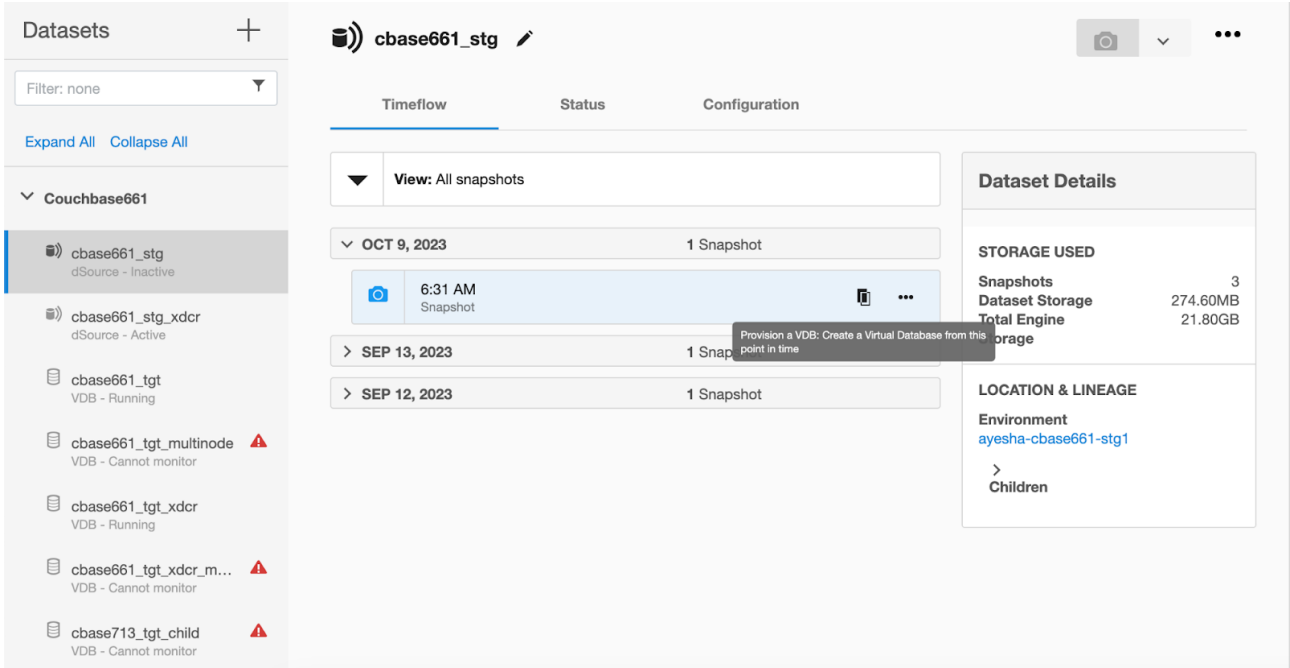
- You should either have linked a dSource from a staging instance following the instructions in Linking a CouchbaseDB dSource[341], or you should have already created a VDB from which you intend to provision another VDB.
- A CouchbaseDB target environment must be set-up with the necessary requirements as described in CouchbaseDB support and requirements[342].
- A CouchbaseDB binary with the same version as the source database must be installed on the Target host.

9.3.4.3.3 Provisioning a VDB

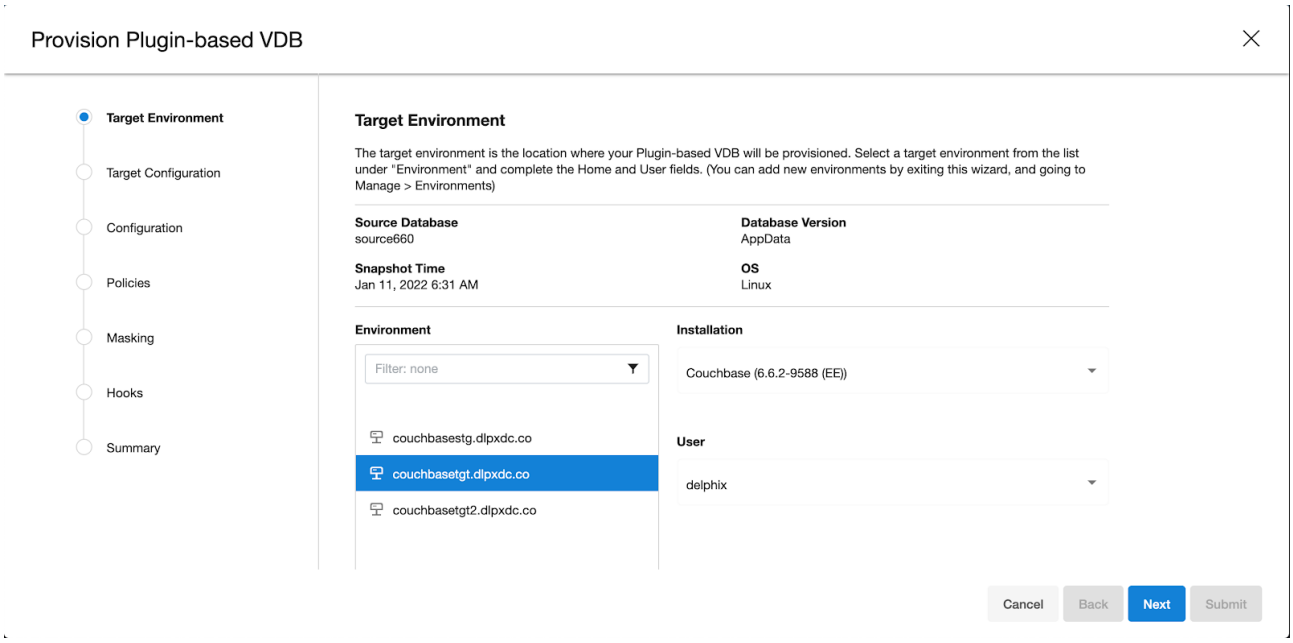
1. Navigate to Manage, and select Datasets.
2. Select the dataset and the snapshot that you want provision from. Click the provision VDB icon to open the provision VDB wizard.

341 https://docs.google.com/document/d/1S48-GbC7zUvE-jses4aRonuVRSYHAIzSLjjjqBmRCM4/edit#heading=h.r479rwk1pche

342 https://docs.google.com/document/d/19UPeBn_JQqaH09ofS5RcXdpH4viprJ_KfIP-y8avkoY/edit#heading=h.i7626pa6lniq

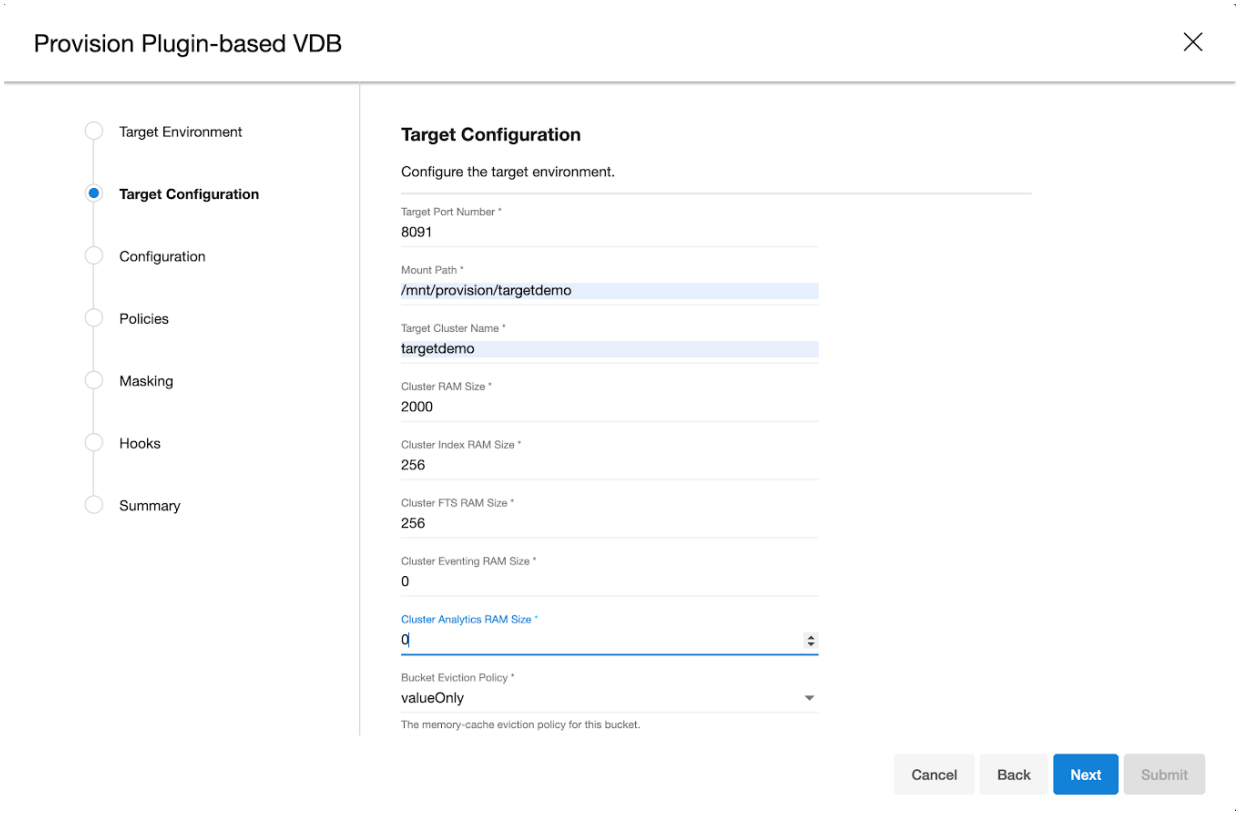


3. Select the target host from the dropdown on which VDB needs to be created.

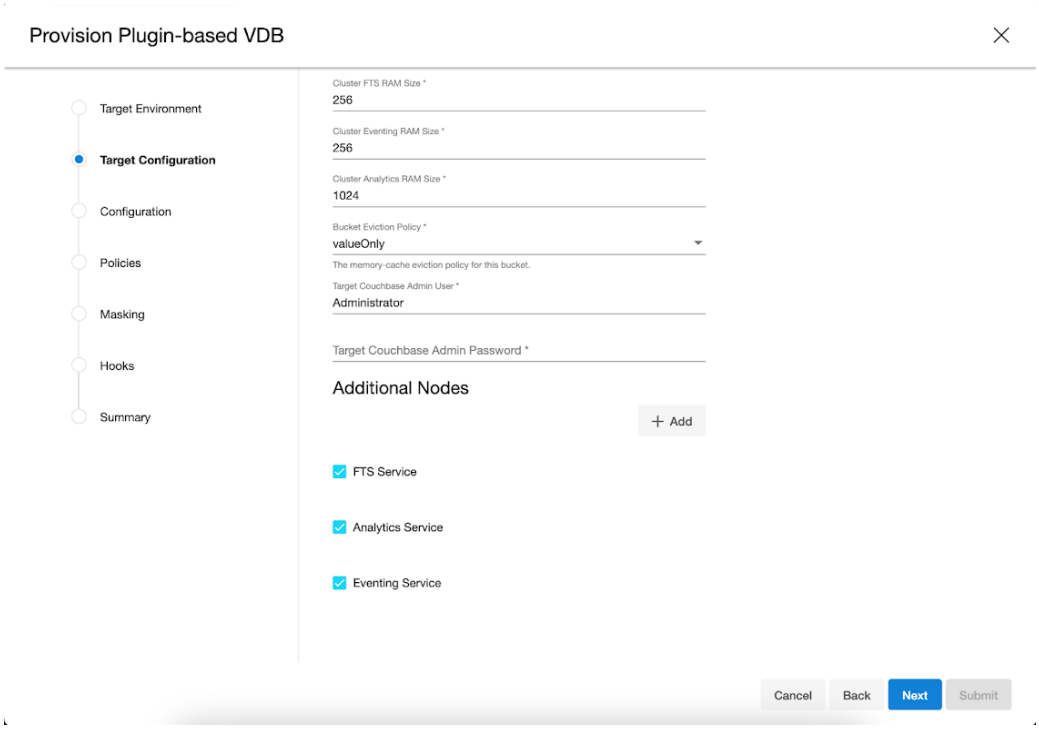


4. Enter the following values for the target configuration:
   - **Target Port Number**: Port number on which Couchbase services will be started. (ex. 8091).
   - **Mount Path**: NFS mount path where dSource snapshot will be mounted by Engine (ex. /mnt/provision/targetdemo).
   - **Target Cluster name**: Cluster name which is required to be set up on the target host. (ex. targetdemo).
   - **Cluster Ram Size**: Whole Cluster memory.

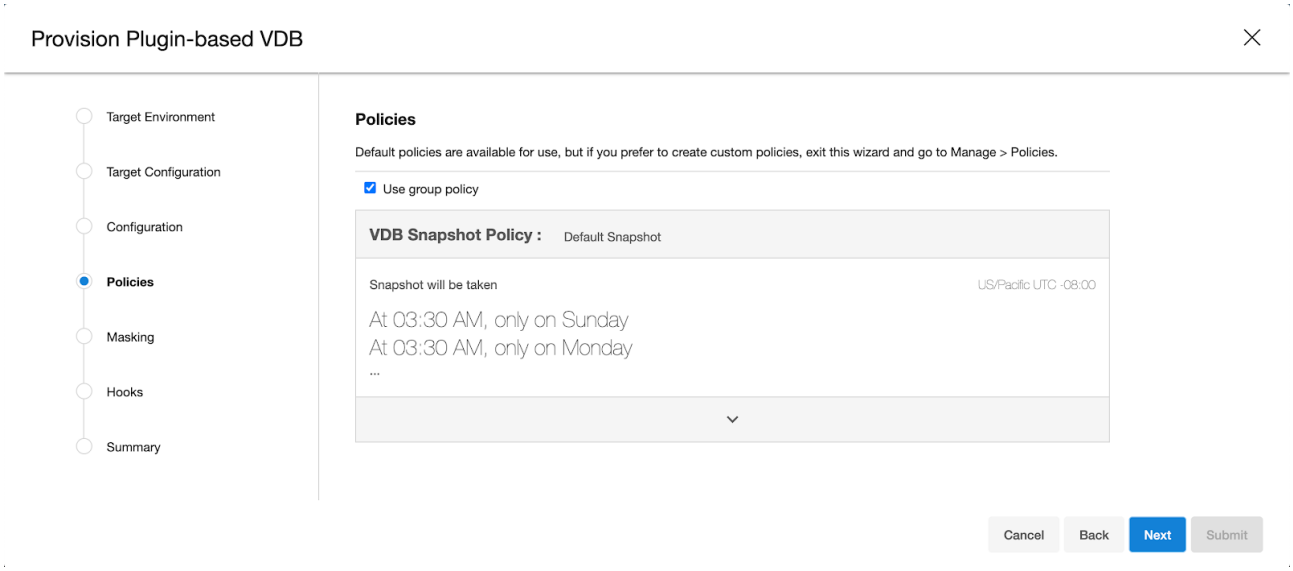
- **Cluster Index Ram Size**: Cluster indexer memory.
- **Cluster FTS Ram Size**: Cluster FTS memory (if needed and FTS service will be configured).
- **Cluster Eventing Ram Size**: Cluster Eventing memory (if needed and Eventing service will be configured).
- **Cluster Analytics Ram Size**: Cluster Analytics memory (if needed and Analytics service will be configured).



5. Enter the following values for the target configuration:
   - **Target couchbase Admin User**: Target Cluster admin username.
   - **Target couchbase Admin password**: Target Cluster admin password.
   - Select services needed on the target cluster (FTS, Eventing, Analytics).

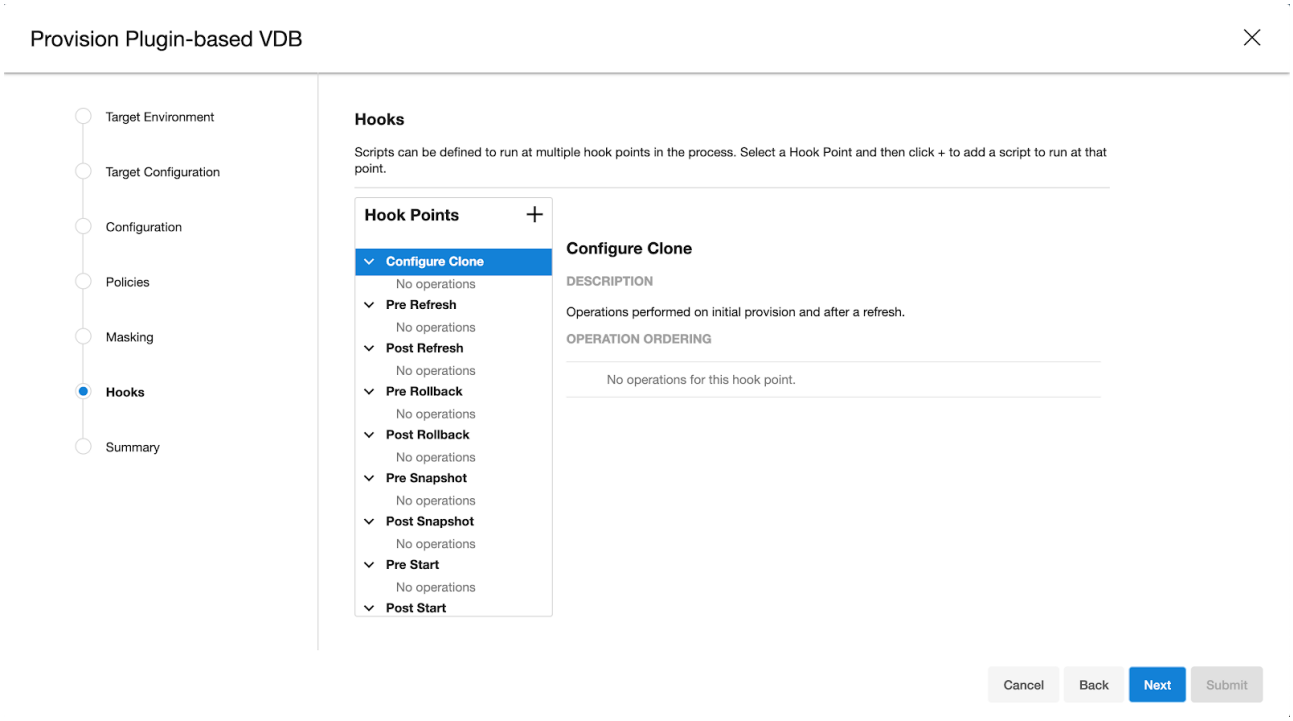


6. Provision connector based VDB. Enter the following value:
   - **VDB Name**: Delphix Target Cluster name.
   - **Target group**: Delphix Target Cluster group.
7. Select a policy for VDB, select **Next**.
8. Select masking for VDB if needed, select **Next**.
9. Add hooks for VDB if needed, select **Next**.



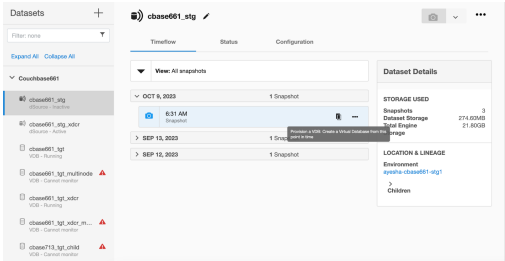
10. Preview the summary and select **Submit**.

11. Once the VDB is created successfully, you can review the datasets on **Manage** > **Datasets** > **vdb Name**.



9.3.4.3.4 Provisioning a multi-node Couchbase VDB

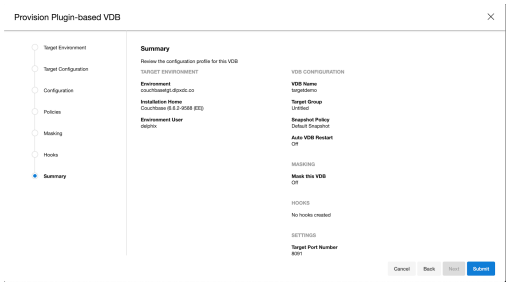
1. Navigate to Manage, and select Datasets.
2. Select the dataset and the snapshot that you want provision from. Click the provision VDB icon to open the provision VDB wizard.

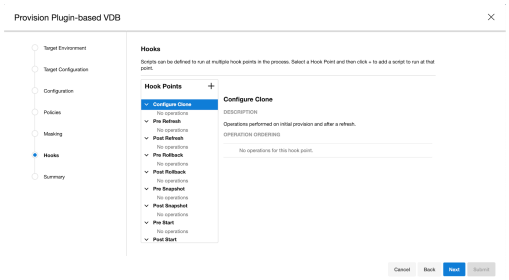


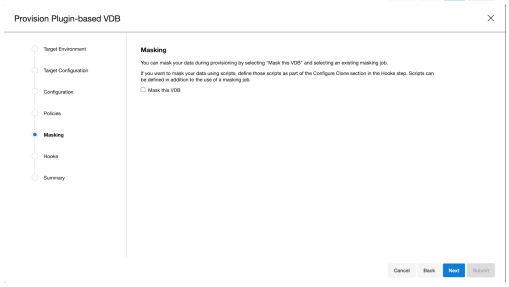
3. Select the target host from the dropdown on which VDB needs to be created.
4. Enter the following values for the target configuration:
   - **Target Port Number**: Port number on which Couchbase services will be started. (ex. 8091)
   - **Mount Path**: NFS mount path where dSource snapshot will be mounted by Engine (ex. /mnt/provision/targetdemo)
   - **Target Cluster name**: Cluster name which is required to be set up on the target host. (ex. targetdemo)
   - **Cluster Ram Size**: Whole Cluster memory

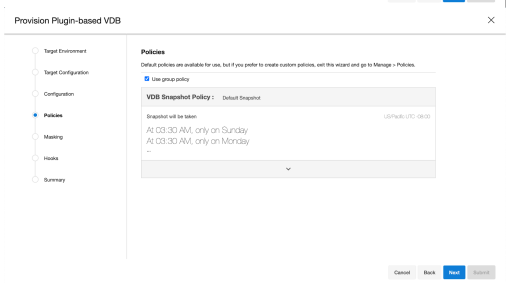
- **Cluster Index Ram Size**: Cluster indexer memory
- **Cluster FTS Ram Size**: Cluster FTS memory (if needed and FTS service will be configured)
- **Cluster Eventing Ram Size**: Cluster Eventing memory (if needed and Eventing service will be configured)
- **Cluster Analytics Ram Size**: Cluster Analytics memory (if needed and Analytics service will be configured)

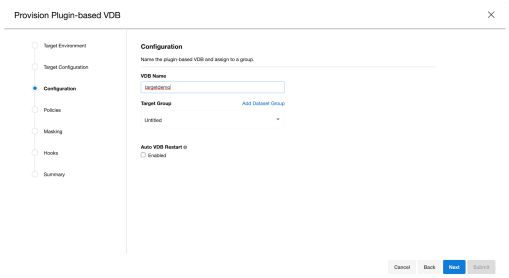
5. Enter the following values for the target configuration:
   - **Target couchbase Admin User**: Target Cluster admin username
   - **Target couchbase Admin password**: Target Cluster admin password
   - Select services needed on the first node of the cluster (FTS, Eventing, Analytics)
   - Click Add buton to open a dialog box for additional node. If you need more nodes, click Add button again to add more nodes
   - Enter the following values for each node:
     - **Delphix Environment Name**: Select an additional node environment from drop-down menu
     - **Delphix Environment User**: Select an environment user
     - **Node hostname / IP**: Enter a hostname or IP of the new node - it will be used as a server name in Couchbase configuration
     - Select services needed on the additional node of the cluster (FTS, Eventing, Analytics)
6. Provision connector based VDB. Enter the follwing value:
   - **VDB Name**: Delphix Target Cluster name
   - **Target group**: Delphix Target Cluster group
7. Select a policy for VDB, select **Next**.
8. Select masking for VDB if needed, select **Next**
9. Add hooks for VDB if needed, select **Next**
10. Preview the summary and select **Submit**

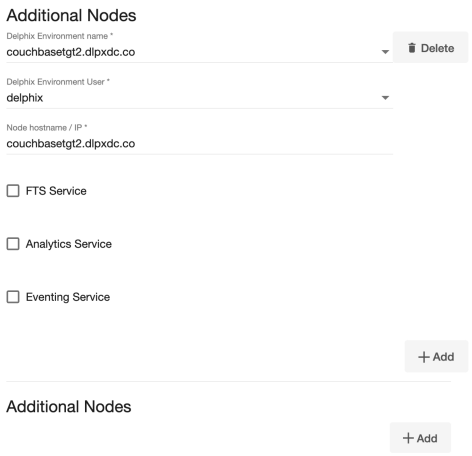


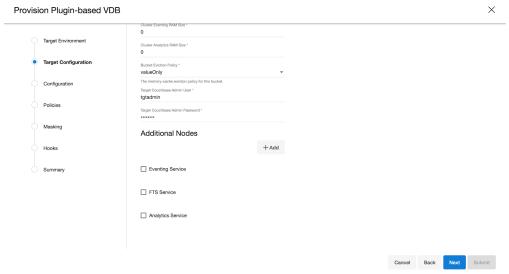

Provision Plugin-based VDB
Target Environment
Target Configuration
Configuration
Policies
Masking
Hooks
Summary
Hooks
Scripts can be defined to run at multiple hook points in the process. Select a Hook Point and then click + to add a script to run at that point.
Hook Points
Configure Clone
No operations
Pre Refresh
No operations
Post Refresh
No operations
Pre Rollback
No operations
Post Rollback
No operations
Pre Snapshot
No operations
Post Snapshot
No operations
Pre Start
No operations
Post Start
Configure Clone
DESCRIPTION
Operations performed on initial provision and after a refresh.
OPERATION ORDERING
No operations for this hook point.
Cancel
Back
Next
Submit

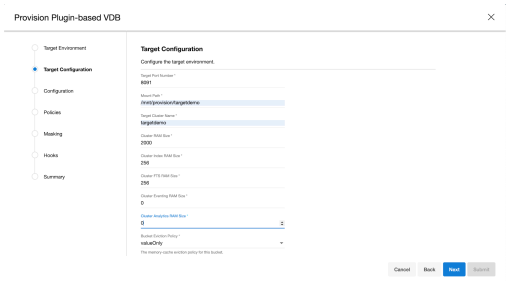

Provision Plugin-based VDB
Target Environment
Target Configuration
Configuration
Policies
Masking
Hooks
Summary
Masking
You can mask your data during provisioning by selecting "Mask this VDB" and selecting an existing masking job.
If you want to mask your data using scripts, define those scripts as part of the Configure Clone section in the Hooks step. Scripts can be defined in addition to the use of a masking job.
Mask this VDB
Cancel
Back
Next
Submit

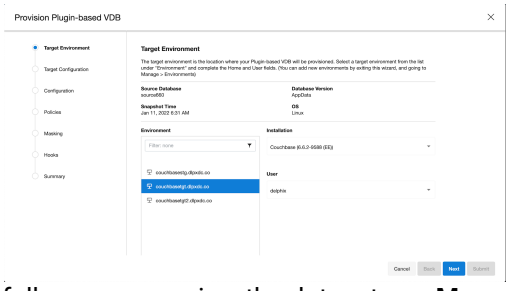

Provision Plugin-based VDB
Target Environment
Target Configuration
Configuration
Policies
Masking
Hooks
Summary
Policies
Default policies are available for use, but if you prefer to create custom policies, exit this wizard and go to Manage > Policies.
Use group policy
VDB Snapshot Policy : Default Snapshot
Snapshot will be taken
US/Pacific UTC -08:00
At 03:30 AM, only on Sunday
At 03:30 AM, only on Monday
Cancel
Back
Next
Submit


Provision Plugin-based VDB
Target Environment
Target Configuration
Configuration
Policies
Masking
Hooks
Summary
Configuration
Name the plugin-based VDB and assign to a group.
VDB Name
Target Group
Add Dataset Group
Untitled
Auto VDB Restart
Enabled
Cancel
Back
Next
Submit

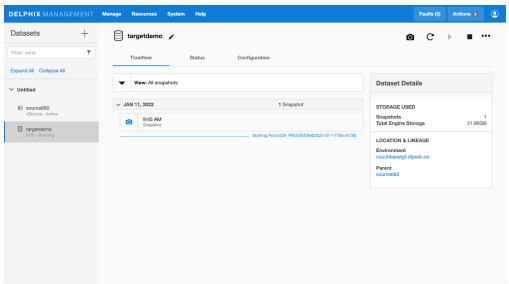

Additional Nodes
Delphix Environment name *
couchbasetgt2.dlpxdc.co
Delete
Delphix Environment User *
delphix
Node hostname / IP *
couchbasetgt2.dlpxdc.co
FTS Service
Analytics Service
Eventing Service
Add
Additional Nodes
Add







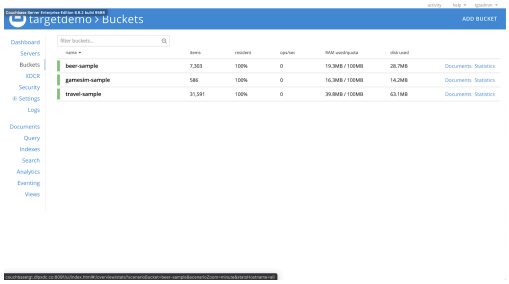
Once the VDB is created successfully, you can review the datasets on **Manage** > **Datasets** > **vdb Name**.



9.3.4.3.5 Accessing target Couchbase VDB cluster

Use an IP / Hostname of the target environment and VDB port defined above to access Target Cluster VDB. Admin user name and password are defined based on input from point 4.

An example screenshot of Couchbase Console of the provisioned VDB can be find below:



9.3.5 Couchbase troubleshooting

9.3.5.1 Logging

There are the following sets of logs for the CouchbaseDB connector.

1. The connector logs are part of the Delphix Engine. Refer to [How to Retrieve Logs] (https://developer.delphix.com/References/Logging/#how-to-retrieve-logs) to find out how to get the Delphix connector Logs.
2. CouchbaseDB Instance logs
    a. CouchbaseDB Instance logs are located at /opt/couchbase/var/lib/couchbase/logs (for staging/target instances)

9.3.6 Couchbase developer's CLI guide

- Discovering a Couchbase environment (see page 1141)
- Couchbase SourceConfig operations (see page 1142)
- Couchbase linking operations (see page 1143)
- Couchbase provisioning operations via CLI (see page 1147)

9.3.6.1 Discovering a Couchbase environment

To discover an environment via CLI, login to the Delphix Continuous Data Engine using the admin user account.

[Connection to Continuous Data with FDQN "dlpx-engine.delphix.com[343]"]

```
Syntax -
ssh admin@dlpx-engine.delphix.com
```

9.3.6.1.1 Discovering the environment

Run the following commands to discover the environment.

[Adding environment with FQDN "rh74-cbase.delphix.com[344]", hostname "rh74-cbase", toolkit-path "/var/tmp", os-user "delphix_os", os-password "delphix"]

```
Example -
```

343 http://dlpx-engine.delphix.com
344 http://rh74-cbase.delphix.com

```
> environment
> create
> set hostEnvironment.name=rh74-cbase.delphix.com
> set hostParameters.host.address=rh74-cbase
> set hostParameters.host.toolkitPath=/var/tmp
> set primaryUser.name=delphix_os
> set primaryUser.credential.password=delphix
> commit
```

Run the following commands to discover the environment.

[Adding environment with FQDN "rh74-cbase.delphix.com[345]", hostname "rh74-cbase", toolkit-path "/var/tmp", os-user "delphix_os", os-password "delphix"]

```
Example -

> environment
> create
> set hostEnvironment.name=rh74-cbase.delphix.com
> set hostParameters.host.address=rh74-cbase
> set hostParameters.host.toolkitPath=/var/tmp
> set primaryUser.name=delphix_os
> set primaryUser.credential.password=delphix
> commit
```

9.3.6.2 Couchbase SourceConfig operations

Run the following command to add the source config.

```
Example -
> sourceconfig create
> set type=AppDataStagedSourceConfig
```

345 http://rh74-cbase.delphix.com

```
> set name=testscfg
> set parameters="{ \
\"couchbaseSrcHost\":\"rh74-src-cbase.delphix.com\", \
\"couchbaseSrcPort\":8091, \
\"dbPath\":\"/opt/couchbase/var/lib/couchbase\", \
\"prettyName\":\"testscfg\"
}"
> set repository="rh74-cbase/Couchbase (7.1.3-3479 (EE))"
> commit
```

9.3.6.2.1 Delete

Run the below command to delete sourceconfig.

```
Example -
> sourceconfig create
> select testscfg
> delete
> commit
```

9.3.6.3 Couchbase linking operations

9.3.6.3.1 Couchbase dSource creation via CLI

To create a dSource via CLI, you must create a source configuration first. For more information on creating a source configuration, refer to Add Source Config using CLI[346].

To create a dSource and perform operations via CLI, login to the Delphix Continuous Data Engine using the admin user account.

```
Syntax -
ssh admin@dlpx-engine.delphix.com
```

346 https://docs.google.com/document/d/1darUpOgXmGaDNCMyQ_gCLHhVgEAQoyg9s3h2GFYk-JY/edit#heading=h.9l0qtkvk9xvv

9.3.6.3.2 Set default linking parameters for Couchbase

Run the below commands to set the default linking parameters. The below example uses testscfg source config.

```
Example -
> database link
> set type=LinkParameters
> set linkData.type=AppDataStagedLinkData
> set name=testds
> set group=Untitled
> set linkData.config=testscfg
> set linkData.environmentUser=rh74-cbase/delphix_os
> set linkData.stagingEnvironment=rh74-cbase
> set linkData.stagingEnvironmentUser=rh74-cbase/delphix_os
> set linkData.syncParameters.parameters={\"resync\":true}
```

9.3.6.3.2.1 Link parameters for the Couchbase XDCR ingestion type

Run the below commands to link parameters for XDCR ingestion type.

```
Example -

> set linkData.parameters="{ \
\"dSourceType\":\"XDCR\", \
\"couchbaseHost\":\"rh74-cbase.delphix.com\", \
\"couchbasePort\":8091, \
\"couchbaseBakLoc\":\"\", \
\"couchbaseBakRepo\":\"\", \
\"mountPath\":\"/mnt/provision/testds\", \
\"stgClusterName\":\"dlpx_staging_cluster\", \
\"clusterRAMSize\":2000, \
\"clusterIndexRAMSize\":256, \
```

```
\"clusterFTSRAMSize\":256, \
\"clusterEventingRAMSize\":256, \
\"clusterAnalyticsRAMSize\":1024, \
\"bucketEvictionPolicy\":\"valueOnly\", \
\"bucketSize\":0, \
\"couchbaseAdmin\":\"Administrator\", \
\"couchbaseAdminPassword\":\"delphix\", \
\"xdcrAdmin\": \"Administrator\", \
\"xdcrAdminPassword\":\"\", \
\"fts_service\":false, \
\"eventing_service\":false, \
\"analytics_service\":true, \
\"config_settings_prov\":[]
}"
```

9.3.6.3.2.2 Link parameters for Couchbase Backup Manager ingestion type

Run the below commands to link parameters for Couchbase Backup Manager ingestion type.

```
Example -

> set linkData.parameters="{ \
\"dSourceType\":\"Couchbase Backup Manager\", \
\"couchbaseHost\":\"rh74-cbase.delphix.com\", \
\"couchbasePort\":8091, \
\"couchbaseBakLoc\":\"/home/delphix_os/backups\", \
\"couchbaseBakRepo\":\"delphix_backup_repo\", \
\"mountPath\":\"/mnt/provision/testds\", \
\"stgClusterName\":\"dlpx_staging_cluster\", \
```

```
\"clusterRAMSize\":2000, \
\"clusterIndexRAMSize\":256, \
\"clusterFTSRAMSize\":256, \
\"clusterEventingRAMSize\":256, \
\"clusterAnalyticsRAMSize\":1024, \
\"bucketEvictionPolicy\":\"valueOnly\", \
\"bucketSize\":0, \
\"couchbaseAdmin\":\"Administrator\", \
\"couchbaseAdminPassword\":\"delphix\", \
\"xdcrAdmin\": \"Administrator\", \
\"xdcrAdminPassword\":\"\", \
\"fts_service\":false, \
\"eventing_service\":false, \
\"analytics_service\":true, \
\"config_settings_prov\":[]
}"
```

9.3.6.3.3 Commit, disable, and enable

9.3.6.3.3.1 Commit

Run the below command to commit the change

```
> commit;
```

9.3.6.3.3.2 Disable

Run the below command to disable the dSource.

```
Example -
> source
> select testdb
> disable
```

```
> commit
```

9.3.6.3.3.3 Enable

Run the below command to enable the dSource.

```
Example -
```

```
> source
```

```
> select testdb
```

```
> enable
```

```
> commit
```

9.3.6.4 Couchbase provisioning operations via CLI

To create a VDB via CLI, you must first create a dSource. For more information on creating a dSource, refer to Link operation via CLI section[347].

To create a VDB and perform operations via CLI, login to the Delphix Continuous Data Engine using the admin user account.

```
ssh admin@dlpx-engine.delphix.com
```

9.3.6.4.1 Provision

Run the below commands to provision a VDB. The below example uses testdb dSource.

```
Example -
```

```
> database provision
```

```
> set type=AppDataProvisionParameters
```

```
> set container.name=testvdb
```

```
> set container.group=Untitled
```

```
> set source.allowAutoVDBRestartOnHostReboot=true
```

```
> set sourceConfig.repository="rh74-cbase.delphix.com/delphix_os vFiles (15.2)"
```

347 https://docs.google.com/document/d/1S48-GbC7zUvE-jses4aRonuVRSYHAIzSLjjjqBmRCM4/edit#heading=h.guuursro9gzw

```
> set timeflowPointParameters.container=testdb
> set sourceConfig.name=testscfgv
> set source.parameters="{ \
\"couchbasePort\":8091, \
\"mountPath\":\"/mnt/provision/testvdb\", \
\"tgtClusterName\":\"dlpx_tgt_cluster\", \
\"clusterRAMSize\":2000, \
\"clusterIndexRAMSize\":256, \
\"clusterFTSRAMSize\":256, \
\"clusterEventingRAMSize\":256, \
\"clusterAnalyticsRAMSize\":1024, \
\"bucketEvictionPolicy\":\"valueOnly\", \
\"couchbaseAdmin\":\"Administrator\", \
\"couchbaseAdminPassword\":\"delphix\", \
\"fts_service\":false, \
\"analytics_service\":false, \
\"eventing_service\"true, \
\"node_list\":[]
}"
> commit
```

9.3.6.4.2 Disable, enable, start, and stop

9.3.6.4.2.1 Disable

Run the below commands to disable a dSource. You must select the same sourceConfig name that was given during VDB creation.

`Example` `-`

```
> source
```

```
> select testscfgv
> disable
> commit
```

9.3.6.4.2.2 Enable

Run the below command to enable a dSource.

```
Example -

> source
> select testscfgv
> enable
> commit
```

9.3.6.4.2.3 Stop

Run the below command to stop a dSource.

```
Example -

> source
> select testscfgv
> stop
> commit
```

9.3.6.4.2.4 Start

Run the below command to start a dSource.

```
Example -

> source
> select testscfgv
```

```
> start
```

```
> commit
```

9.3.7 Couchbase data source limitations

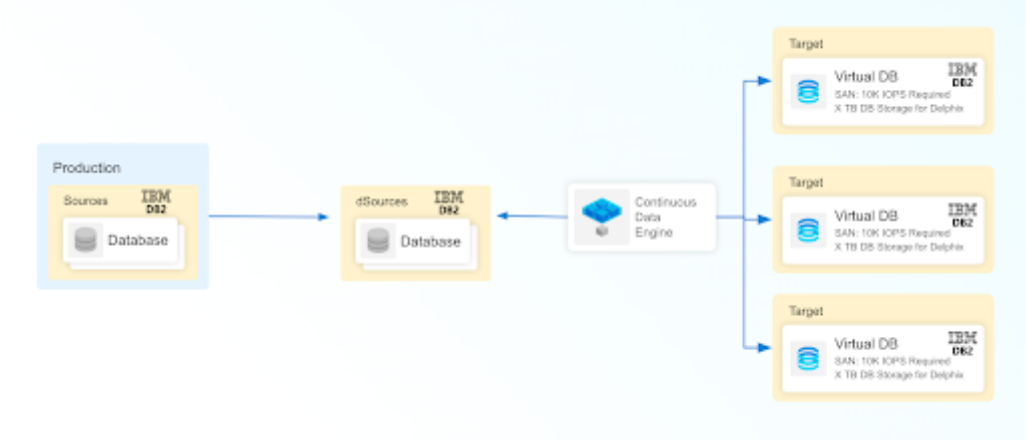
Note the following limitations for Couchbase data sources.

- Child VDB cannot be provisioned from Multi-node VDB.
- Multi-node VDB can't be cloned.
- V2P is not supported
- Point in time recovery is not supported. Provision / refresh / rewind time is based on snapshot only

9.4 IBM Db2 data sources

9.4.1 Overview

The IBM Db2 virtualization solution consists of the Delphix Continuous Data Engine, and the connector for IBM Db2 which must be uploaded and installed into the Delphix Continuous Data Engine. This solution ingests source data using the staging architecture[348], which copies the production database from the source environment into a staging environment. The Delphix Continuous Data Engine captures the persistent storage layer of the staging database, which then allows Delphix Continuous Data Engine to create virtualized copies in target environments.



9.4.1.1 IBM Db2 connector architecture

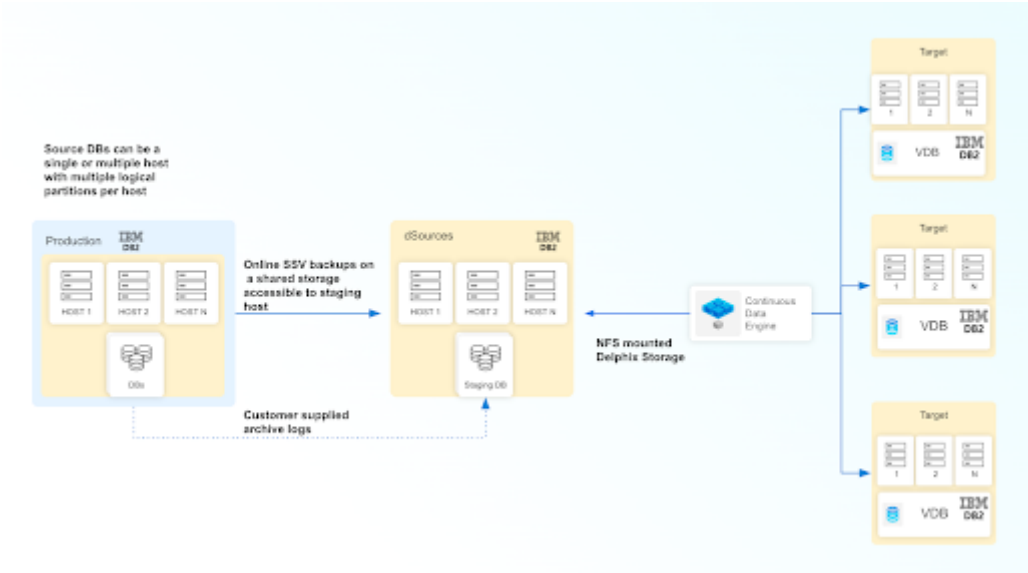
IBM Db2 connector supports ingestion of both single partition (non-DPF) AND multi-partition (logical) also known as DPF (Database Partition Feature) databases. (see page 1219)

IBM Db2 architecture for single partition (non-DPF) databases:

348 https://cd.delphix.com/docs/latest/general-prerequisites

9.4.1.1.1 IBM Db2 architecture with multi-partition database (DPF):

A multi-partition, or partitioned, database is a database with more than one partition. Depending on your hardware environment, there can be several configurations in which you can partition your database. During DPF dSource ingestion, the production database can be physically partitioned across multiple hosts. However, for the dSource ingestion or VDB provisioning process, the staging and target hosts will have to be configured using logical partitions only.



The IBM Db2 connector supports both staging pull and staging push.

The following specific ingestion methods are supported:

- **Staging Pull:** This ingestion method allows Delphix Continuous Data Engine to execute the necessary operations to prepare the data content in the Staging database. Staging pull supports the below ingestion methods for both DPF and non-DPF databases.
    - HADR (High Availability Disaster Recovery)
    - Backup only
    - Backup with Archived logs
        - Rollforward to Point-in-Time (RFPIT)

- **Staging Push:** This ingestion method requires you to execute the necessary operations to prepare the data content in the staging database for both DPF and non-DPF databases.

9.4.1.2 Limitations of the IBM Db2 solution

Delphix Continuous Data Engine and the IBM Db2 connector should not be considered a replacement for backup and recovery of your source instances and databases. The Delphix Continuous Data Engine solution is intended for very different purposes, primarily to create non-production database copies.

The following are not supported:

- Architecture other than 64-bit OS
- Windows OS type
- Db2 Versions below 10.5 i.e (10.1, 9.8 etc)
- Db2 Offline backup at staging for staging push
- Migrating existing VDBs to a different host
- Virtual to Physical (V2P) VDB operation
- Db2 Pure Scale databases
- Multi-node configuration of dSource or VDB databases

9.4.1.3 Support

License for the IBM Db2 connector is included with the standard Delphix License Agreement, and the solution is supported by Delphix Technical Support. For any questions, bugs, or feature requests, contact us via Delphix Support[349] or Delphix Community[350] portals.

9.4.2 IBM Db2 virtualization process

The following material introduces you to the overall process of deploying the IBM Db2 virtualization solution. A sequence of high-level steps is presented. Each step briefly describes the task and the expected result. For more information, refer to the embedded links.

1. Begin by reviewing the general Delphix Continuous Data Engine architecture diagrams (see page 1014). Identify existing or new IBM Db2 source, staging, target environments, the underlying operating systems, and IBM Db2 database versions. This will give you a good understanding of the architectural strategy and confirm support.
2. Review the IBM Db2 support matrix (see page 1153). Confirm that your combination of IBM Db2 software, host operating system, Delphix Continuous Data Engine version, and IBM Db2 connector version are supported for the various environments that will integrate into your IBM Db2 deployment. Early validation will reduce unexpected issues later.
3. Deploy and configure the Delphix Continuous Data Engine.

349 https://support.delphix.com/
350 https://community.delphix.com/contactus

4. Install the IBM Db2 connector (see page 1176). The IBM Db2 connector extends the Delphix Continuous Data Engine to IBM Db2 datasets. It is recommended to stay current with the latest version of the connector.
5. Create staging and target environments (see page 1180). The staging environment will host an IBM Db2 instance that the Delphix Continuous Data Engine will ingest from. The target environment will host one or more IBM Db2 instance(s) in which the virtual database copies will be created.
    a. Consult the source environment (see page 1161) and staging/target environments (see page 1164) requirements documentation for configuration details.
6. Add staging and target environments in the Delphix Continuous Data Engine (see page 1180). Use the "Add Environment" operation in the Delphix Continuous Data Engine for each host, which causes the Delphix Continuous Data Engine to register the specific host and catalog the IBM Db2 software installation(s) on that host.
7. Link the IBM Db2 source database (see page 1185). Configure and create a dSource which allows the Delphix Continuous Data Engine to capture a copy of the source data. There are many ingestion methods that can offer a variety of methods to capture the data to be virtualized by the Delphix Continuous Data Engine. Therefore, we recommend identifying the ingestion method best suits your organization and infrastructure policies.
8. Provision virtual IBM Db2 databases (see page 1208). Administrators and users can provision multiple copies of the source data as virtual databases.

9.4.3 IBM Db2 requirements and prerequisites

This section contains the following pages about IBM Db2 requirements and prerequisites in Delphix Continuous Data Engine.

- IBM Db2 support matrix (see page 1153)
- Requirements for IBM Db2 source databases (see page 1161)
- Requirements for IBM Db2 staging and target environments (see page 1164)
- Sudo privileges for the environment users (see page 1170)

9.4.3.1 IBM Db2 support matrix

Refer to the following support matrices to ensure compatibility between the environment, database, connector, and Delphix Continuous Data Engine.

Compatible upgrades have a green **Y**, while incompatible upgrades show a red **N**.

Following Delphix support policies, when a minor release is certified, all the patch releases falling under that minor release are also considered certified.

9.4.3.1.1 Environment and database compatibility

9.4.3.1.1.1 Database editions

The following are the IBM Db2 database editions supported by connector version 4.0.0 and above.

- Standard edition
- Advanced edition
- Advanced enterprise server edition
- Enterprise server edition
- Community edition
  - Only available for IBM Db2 v11.5+.
- IBM Db2 with BLU Acceleration Pattern 1.2.19.0
  - Db2 v11.5.8.0 Special fix_29133[351].

> For more information on support for BLU or column-organized tables with DB2 connector, please reference IBM Db2 additional information (see page 1219).

9.4.3.1.1.2 Operating system

The IBM Db2 connector is certified for use in source, staging, and target environments with the following operating systems and database versions.

9.4.3.1.2 Red Hat Enterprise Linux (RHEL)

> Support for RHEL 8.x requires installing the libncurses.5 library onto the host. Please reference KBA 5622[352] for actionable steps.

IBM Db2 connector supports the following OS and Db2 Database versions.

OS version	Db2 Database version

351 https://www.ibm.com/docs/en/cloud-pak-system-w4600/2.3.3?topic=patterns-db2-blu-acceleration-pattern

352 https://support.delphix.com/Delphix_Virtualization_Engine/Delphix_Admin/Environment_Operations_Fail_With_%22bash%3A_error_while_loading_shared_libraries%3A_libncurses.so.5%3A_cannot_open_shared_object_file%3A_No_such_file_or_directory%22_(KBA5622)

	10.5	11.1	11.5
RHEL 6.9	Y	Y	N
RHEL 7.0	Y	Y	N
RHEL 7.1	Y	Y	N
RHEL 7.2	Y	Y	N
RHEL 7.3	Y	Y	N
RHEL 7.4	Y	Y	N
RHEL 7.5	Y	Y	N
RHEL 7.6	Y	Y	N
RHEL 7.7	N	Y	N
RHEL 7.8	N	Y	Y 6.0.5+
RHEL 7.9	N	Y	Y
RHEL 8.0	N	Y	Y
RHEL 8.1	N	Y	Y
RHEL 8.2	N	N	Y
RHEL 8.3	N	N	Y
RHEL 8.4	N	N	Y
RHEL 8.5	N	N	Y
RHEL 8.6	N	N	Y 6.0.16+

RHEL 8.7	N	N	Y
RHEL 8.8	N	N	Y 12.0.0.0+
RHEL 9.2	N	N	Y 19.0.0.0+

9.4.3.1.3 Advanced Interactive eXecutive (AIX)

IBM Db2 connector supports below mentioned OS and Db2 versions.

	Database version		
OS Version	**10.5**	**11.1**	**11.5**
AIX 7.1	Y	Y	Y
AIX 7.2	Y	Y	Y

9.4.3.1.4 IBM Db2 connector and Delphix Continuous Data Engine compatibility

For more information, refer to the Delphix End of Support[353] policy to determine whether a connector version is in primary, extended, or legacy support. Contact Delphix customer support with any questions.

353 https://support.delphix.com/Support_Policies_and_Technical_Bulletins/Support_Policies/Software_Product_Line_Policies_(KBA1003)#:~:text=Delphix%20may%20discontinue%20a%20product,18%20months%20of%20Extended%20Support

Engine Version	IBM Db2 Connector version																				
	3.1.1	4.0.0	4.0.1	4.1.0	4.1.1	4.1.2	4.1.3	4.2.0	4.2.1	4.3.0	4.3.1	4.3.2	4.4.0	4.4.1	4.4.2	4.5.0	4.5.1	4.6.0	4.6.1	4.7.0	4.7.1	4.7.2
6.0.0.x	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.1.x	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.1.x	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.3.x	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.4.x	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.5.x	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N

6.0.6.x	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.7.x	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.8.x	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.9.x	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.10.x	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.11.x	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.12.x	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.13.x	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N	N

6.0.14.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N
6.0.15.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N
6.0.16.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N
6.0.17.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N
7.0.0.x	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N
8.0.0.x	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N
9.0.0.x	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N
10.0.0.x	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N

11.0.0.x	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N
12.0.0.x	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N
13.0.0.x	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N
14.0.0.x	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
15.0.0.x	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
16.0.0.x	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
17.0.0.x	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

18.0.0.x	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
19.0.0.x	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
20.0.0.x	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
21.0.0.x	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

9.4.3.2 Requirements for IBM Db2 source databases

IBM Db2 connector is based on the zero touch production concept, meaning the production environment and staging environments are mutually exclusive except for HADR (High Availability Disaster Recovery) where the staging environment acts as a standby for the production. Source Db2 hosts are considered primary/production servers that have Db2 binaries installed and have Db2 instances created on them. The hosts that contain the data that you wish to ingest are referred to as the source environment. This topic describes the requirements for creating connections between the Delphix Continuous Data Engine and Db2 hosts and instances.

Requirements	**Explanation**	
	Single Partition (Non-DPF)	**Multi Partitions(DPF)**

HADR Ports	The configuration values for HADR_LOCAL_SVC and HADR_REMOTE_SVC should be set. These ports are used at the end user's discretion and need to be specified during the linking process. It is highly recommended that these ports also be defined in the `/etc/services` file to ensure that they are only used by Db2 for the specified databases.	N/A
SSL certificate	For encrypted HADR transmission, SSL certificate should be made available on source host.	N/A
HADR source state	Source instance and database must be in active state to create HADR connection.	N/A

dbadm and *secadm* privileges for database	The database instance users must have *dbadm*[354] and *secadm*[355] permissions to enable cross instance restore.
	Delphix Continuous Data Engine uses the Db2 instance owner account on the dSource for multiple operations, including querying data inside the databases. In order to ingest a database on the staging server with different instances, you need permissions on the source database in order to perform a restore on the staging server. For example, if you have an instance named "auto1051" at the source with a database name "delphix" and you want to create a dSource on the "auto1052" instance on the staging server, then you must explicitly grant DBADM and SECADM to the dSource instance "auto1052" on the source instance using the following steps:
	• Connect to the source database as the source instance owner.
	  • `connect to <DB_NAME> user <SOURCE_INSTANCE_OWNER>`.
	• Issue database grant command,
	  • `grant DBADM, SECADM on the database to user <DSOURCE_INSTANCE_OWNER>`
	  • `grant DBADM, SECADM on the database to user <VDB_INSTANCE_OWNER>`
	• Repeat step 2 for every database to be included in the dSource, on the corresponding source database.
		Same as non-DPF

Database container requirement	It is recommended not to use special characters for source database container paths as this may create problems while parsing such container paths and may also result in failures during dSource and VDB creation.	Same as non-DPF

354 https://www.ibm.com/docs/en/db2/11.5?topic=authorities-database-administration-authority-dbadm
355 https://www.ibm.com/docs/en/db2/11.5?topic=authorities-security-administration-authority-secadm

Banners (see page 1219)	For connector versions < 4.2.0, Banners are not supported For connector versions >= 4.2.0, `.profile`, `MOTD`, and `ssh` banners are supported. **Note:** Print/echo statements in `.bashrc` are not supported by the connector.	Same as non-DPF

9.4.3.3 Requirements for IBM Db2 staging and target environments

This section elaborates on the various host requirements for the staging host where the dSource is linked and the target where the VDB is provisioned based on the type of ingestion method being used.

Requirements	Explanation	
	Single Partition (Non-DPF)	**Multi Partitions (DPF)**
Access to database backups	The staging environment must have access to a full online backup of the source database on disk to create the first full copy. Delphix Continuous Data Engine recommends using compressed backups as that will reduce storage needs and speed up ingestion. It is recommended to test the integrity of the backup before using it for ingestion using the utility *db2ckbkp*[356]. Once this first fill backup is ingested into the staging database, representing the Db2 dSource, subsequent synchronizations can be accomplished via HADR or Archive Logs.	Same as non-DPF
HADR Ingestion - port specific	The configuration values for `HADR_LOCAL_SVC` and `HADR_REMOTE_SVC` should be set. These ports are used at the end user's discretion and need to be specified during the linking process. It is highly recommended that these ports also be defined in the `/etc/services` file to ensure that they are only used by Db2 for the specified databases.	NA

356 https://www.ibm.com/docs/en/db2/11.5?topic=commands-db2ckbkp-check-backup

Db2 instance	The staging and target Db2 instances that you wish to use must already exist on the host. We can use the same instance for the dSource and VDB creation and can also have multiple VDBs on the same instance.	Same as non-DPF
Db2 instance configuration parameters	Instance level configuration values such as the bufferpool value will need to be managed by the end users independent of Delphix Continuous Data Engine.	Same as non-DPF
Db2 instance compatibility	The instances used for staging and target environments must be identical with the source Db2 instance in terms of installation version.	Same as non-DPF
No of partitions (Logical)	NA	The staging and target hosts must be configured with the same number of logical partition nodes as the source.

Logical partition configuration	NA	The logical partition configuration must be added in the `$HOME/sqllib/db2nodes.cfg` file where $HOME is the home directory of the instance user. The hostname to be used in the db2nodes.cfg file should be a fully qualified domain name.
		Example:
		• db2 `nodes.cfg` file configured for 4 logical partitions should look like:
		`0 <FQDN hostname> 0`
		`1 <FQDN hostname> 1`
		`2 <FQDN hostname> 2`
		`3 <FQDN hostname> 3`

/etc/services entry	NA	• Each logical partition must have a corresponding port entry in `/etc/services` of hosts.
		• In the example below, db2inst1 is the instance where the user and port number from 60000 to 60003 are reserved for inter-partition communication.
		`DB2_db2inst1 60000/tcp`
		`DB2_db2inst1_1 60001/tcp`
		`DB2_db2inst1_2 60002/tcp`
		`DB2_db2inst1_END 60003/tcp`

Registry variable setting	Registry variables `DB2RSHCMD` and `DB2COMM` should be updated with values as shown below:
	`SSHCMD=$(which ssh)`
	`db2set DB2RSHCMD="$SSHCMD"`
	`db2set DB2COMM=TCPIP`
		Same as non-DPF

db2_all utility test	NA	After the environment is configured on staging and target, execute `db2_all echo hello` on both staging and target servers to add host key in the known_hosts file. This will also help to check for any ssh issues on these servers.
Archive logs	Instance users should have permission to read logs.	The backup files (or log directory if using Archive Logs) should be consistent with the partitions configured on staging.

SSH configuration parameters	NA	When configuring staging and target environments for multiple partitions (dpf), the `ClientAliveInterval` option must be set to 0 or commented out to prevent the SSH connection from getting severed in the midst of command executions.
Primary user permissions	There must be an operating system user (delphix_os) with these privileges: • Ability to login to the staging and target environment via SSH • Ability to run mount, umount, mkdir, and rmdir as a super-user. If the staging and target host is an AIX system, permission to run the nfso command as a super-user.	Same as non-DPF

Toolkit directory and its permissions	There must be a directory on the staging and target environment where you can install the IBM Db2 connector. For example, `/var/opt/delphix/toolkit`. • The primary user (delphix_os) must own the toolkit path directory (`__/var/opt/delphix/toolkit`). • If the primary user and instance users are sharing any common group, then the toolkit path directory (`__/var/opt/delphix/toolkit`) must have permissions rwxrwx-- (0770), but you can also use more permissive settings. This allows instance users who are part of the same group as the delphix_os user's group to be able to create directories inside the toolkit path directory. • If the delphix_os user and instance users are not sharing any common group, then the toolkit path directory must have -rwxrwxrwx (0777) permissions. • The directory should have 1.5GB of available storage: 400MB for the connector and 400MB for the set of logs generated by each Db2 instance that runs on the host • The connector directory will be used as the base location for the mount point by default if no other custom mount base is provided.	Same as non-DPF
Common group between primary and instance users	The primary environment user should share its primary group with the instance user. For example, if the delphix_os is the primary environment user which is used for environment addition and its primary group is also delphix_os then instance users (responsible for the Delphix Continuous Data Engine operations such as linking and provisioning) should share group delphix_os as their secondary group	Same as non-DPF

Instance user requirement	• The instance owner of each instance you wish to use within staging or a target host must be added as an environment user within the Delphix Continuous Data Engine.
	• For HADR synced dSources the staging instance owner must be able to "read" the ingested database contents as Delphix Continuous Data Engine will check the validity of the database by querying tables before each dSource snapshot.
	Note: If a Db2 database tablespace container is added or deleted on primary/source database, the dSource will have to be resynced.
	• Ensure that the following database configurations are set to default values:
	  • LOGPRIMARY, LOGSECOND - default values are set by DB2 configuration advisor and shouldn't be changed. For more information, refer to logsecond - Number of secondary log files configuration parameter[357].
	  • LOGARCHCOMPR1, LOGARCHCOMPR2- the default values are set to OFF so that the workflows can work properly.
		Same as non-DPF

DB2_RESTORE_GRANT_ADMIN_AUTHORITIES	If the instance user doesn't have access to the source database and the staging host uses a different instance user to ingest the dSource, the user can set the DB2 registry variable `db2set DB2_RESTORE_GRANT_ADMIN_AUTHORITIES=ON`. Db2 will then grant administrative authorities to the staging host instance after the restore is complete. However, since this only works during restore operation, VDBs can only be provisioned on instance with the same name as staging database.	Same as non-DPF

357 https://www.ibm.com/docs/en/db2/11.5?topic=parameters-logsecond-number-secondary-log-files

Database level requirement	• The staging and target hosts must have the empty instances created prior to Delphix Continuous Data Engine using them • The desired OS user to execute commands related to dSource and VDB operation for each instance has been added as an environment user. • Make sure the staging instance used for linking doesn't have an existing restoring database with the same name. • The dSource ingestion or VDB provisioning should not be performed on the source instance to maintain the integrity of the source database.	Same as non-DPF
Banners (see page 1219)	For connector versions < 4.2.0, Banners are not supported For connector versions >= 4.2.0, `.profile`, `MOTD`, and `ssh` banners are supported. **Note:** Print/echo statements in `.bashrc` are not supported by the connector.	Same as non-DPF

It is highly recommended that the Database Partitioning Feature (DPF) for IBM Db2 staging and target should be configured on separate hosts due to the reason that the DPF instances in staging or target environments consume a lot of resources.

9.4.3.4 Sudo privileges for the environment users

This topic describes the rationale behind specific `sudo` privilege requirements for virtualizing IBM Db2 databases along with a few default commands. As required by the IBM Db2 connector, you must grant this OS account the following key privileges:

1. **Disable requiretty:** The Delphix Continuous Data Engine requires the `requiretty` setting to be disabled. This allows the OS user to run sudo commands without a logged-in terminal session (tty).

2. **Add NOPASSWD:** By including the NOPASSWD qualifier, you ensure that the "sudo" command does not prompt for a password when executing certain commands such as mount, umount, mkdir, and rmdir.

In the example configurations, the OS username **delphix_os** is used. Make sure to modify the username based on the OS user available in the environment for performing the linking and/or provisioning operations.

9.4.3.4.1 Procedure

Perform the following steps on both the staging and target environments.

1. The sudo configuration is located in the `/etc/sudoers` file. Run the following commands to open the `sudoers' file:

```
vi /etc/sudoers
```

> Third party tools like `visudo`, can be helpful in minimizing syntax errors.

2. Next, update the file with the following snippet to grant the required privileges to the `delphix_os` user:

```
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD: \
/bin/mount, /bin/umount, /bin/mkdir, /bin/rmdir
```

9.4.3.4.2 Configuring `sudo` access on AIX for IBM Db2 staging and target environments

In addition to sudo access to the `mount`, `umount`, and `ps` commands on AIX target hosts, Delphix Continuous Data Engine also requires `sudo` access to `nfso`. This is required on the staging/target hosts for the Delphix Continuous Data Engine to monitor the NFS read-write sizes configured on the AIX system. A super-user access level is needed to run the `nfso` command.

Example: AIX /etc/sudoers file for a Delphix Continuous Data Engine target

```
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD: \
/usr/sbin/mount, \
/usr/sbin/umount, \
/usr/sbin/nfso, \
/usr/bin/ps, \
/bin/mkdir, \
/bin/rmdir
```

9.4.3.4.3 Example 1

This example restricts the **delphix_os** user's use of `sudo` privileges to the directory `/db2`.

Note that the wildcards are allowed for the mount and `umount` options because those commands expect a fixed number of arguments after the options. The option wildcard on the `mount` command also makes it possible to specify the file system being mounted from the Delphix Continuous Data Engine.

However, wildcards are not acceptable on `mkdir` and `rmdir` commands because they can have any number of arguments after the options. For those commands, you must specify the exact options (`-p`, `-p -m 755`) used by the Delphix Continuous Data Engine.

Example `/etc/sudoers` file configuration on the target environment for sudo Privileges on the VDB mount directory only (Linux OS)

```
Defaults:delphix_os !requiretty
delphix_os ALL=(root) NOPASSWD: \
/bin/mount  *        /db2/*, \
/bin/umount *        /db2/*, \
/bin/umount          /db2/*, \
/bin/mkdir -p        /db2/*, \
/bin/mkdir -p -m 755 /db2/*, \
/bin/mkdir           /db2/*, \
/bin/rmdir           /db2/*
```

9.4.3.4.4 Example 2

This example restricts the **delphix_os** user's use of `sudo` privileges to the directory / `db2`, restricts the mount commands to a specific Delphix Continuous Data Engine hostname and IP, and does not allow user-specified options for the `umount` command.

This configuration is more secure, but there is a tradeoff with deployment simplicity. This approach would require a different sudo configuration for targets configured for different Delphix Continuous Data Engines.

Example: Configuring the `/etc/sudoers` file on the target environment for privileges on the VDB mount directory only, and allows mounting only from a single server (Linux OS)

```
Defaults:delphix_os !requiretty
delphix_os ALL=(root) NOPASSWD: \
/bin/mount           <delphix-server-name>* /db2/*, \
/bin/mount *         <delphix-server-name>* /db2/*, \
/bin/mount           <delphix-server-ip>*   /db2/*, \
/bin/mount *         <delphix-server-ip>*   /db2/*, \
/bin/mount "", \
/bin/umount          /db2/*, \
```

```
/bin/umount *        /db2/*, \
/bin/mkdir [*]       /db2/*, \
/bin/mkdir           /db2/*, \
/bin/mkdir -p        /db2/*, \
/bin/mkdir -p -m 755 /db2/*, \
/bin/rmdir           /db2/*
```

9.4.3.4.5 Privilege elevation requirements

The optional privilege elevation feature allows a non-privileged OS user (e.g., delphix_os) to gain the permissions of a privileged OS user (e.g., dlpxqa) for operations on the staging or target environments. This configuration can be valuable to maximize an environment's security posture. Privilege elevation can be used to work with Kerberos environments.

9.4.3.4.5.1 Procedure

First, the Privilege Elevation script, dlpx_db_exec, must be created and uploaded onto the Delphix Continuous Data Engine. Then, the non-privileged OS user must be granted the appropriate permissions.

Upload the privilege elevation script.

Perform one of the following methods to upload the privilege elevation script:

a) CURL file upload method

Before running any of the following commands, update <CONTINUOUS-DATA-FQDN-or-IP>, <USERNAME>, and <PASSWORD> placeholder values.

1. Create a session with the Delphix Continuous Data Engine.
    a. Also, update the version values for the `major`, `minor`, and `micro` fields.
        i. The correct values for your Delphix Continuous Data Engine version can be found in the API version information (see page 2753).

```
curl -i -c cookies.txt -X POST -H "Content-Type:application/json"
 https://<CONTINUOUS-DATA-FQDN-or-IP>/resources/json/delphix/
session -d '{
    "version":{
        "major":1,
        "minor":11,
        "micro": 5,
        "type":"APIVersion"
    },
    "type":"APISession"
}'
```

2. Login to Delphix Continuous Data Engine as the `admin` user.

```
curl -i -c cookies.txt -b cookies.txt -X POST -H "Content-Type:application/
json" https://<CONTINUOUS-DATA-FQDN-or-IP>/resources/json/delphix/login -d '{
    "username":"<USERNAME>",
    "password":"<PASSWORD>",
    "type":"LoginRequest",
    "target":"DOMAIN"
}'
```

3. Copy `DLPX_DB_EXEC` contents to the Delphix Continuous Data Engine.

```
curl -i -b cookies.txt -X POST -H "Content-Type:application/json" https://
<CONTINUOUS-DATA-FQDN-or-IP>/resources/json/delphix/host/privilegeElevation/
profileScript/HOST_PRIVILEGE_ELEVATION_PROFILE_SCRIPT-7 -d '{"type":
"HostPrivilegeElevationProfileScript","contents": "#\n# Copyright (c) 2017 by
Delphix. All rights reserved.\n#\n\n#\n# This script allows customization of
command execution with an alternate user\n# account.\n# Arg $1 contains \"-
u<optional user account>\" for the desired user under\n# which database
commands will be executed.\n# By default this argument is ignored and the
script is executed as the default\n# account.\n#\n\nif [[ $1 != -u* ]]; then\n
echo \"Incorrect command line paramaters, -u<optional user account> is required
as the first parameter\"\n    exit 1\nfi\nuser_id=`echo $1 | sed -e \"s\/^-u\/
\/\"`\necho \"$user_id\" >> \/tmp\/test.log\nshift 1\n#echo \"$user_id and
$DB2_DB_NAME\" >> \/tmp\/test.log\nif [[ $user_id != \"delphix_os\" ]];
then\ncommand=$(printf \"%s \" \"$@\")\nsudo su - $user_id -c \"cd \/home\/
delphix_os;export DB2DBDFT=$DB2DBDFT;$command\"\nelse\n\"$@\"\nfi"}'
```

> If the dlpx_db_exec script is updated after the environment(s) is added, refresh the environment to propagate the changes.

b) CLI Method

For steps on creating a privilege elevation profile, refer to [CLI Cookbook: How to create or edit privilege elevation profiles and profile scripts]((see page 2594)).

9.4.3.4.5.2 Configure sudo privileges to grant privilege elevation

With privilege elevation, all IBM Db2 connector commands are run via the `dlpx_db_exec` script with the first parameter being the user to execute, and the remaining parameters being the command to execute and within this script, the commands are run using 'sudo'(or some other custom elevation utility). This script may be customized by the end user, but it must always return the results of the executed command and exit with the return code from the executed command.

When a command is invoked with `dlpx_db_exec`, instance IDs that were discovered during the discovery phase will be passed to the script. All Db2 changes are based on top of the Db2 DB level changes.

To execute the connector commands as the high-privileged OS user, sudoers entries are required.

To add additional sudoers entries you need to edit the `/etc/sudoers` file. Run the following commands to open the `sudoers' file:

```
vi /etc/sudoers
# or
visudo
```

> Third party tools like `visudo`, can be helpful in minimizing syntax errors.

Entries required for both linking and provisioning via low-privileged user (delphix_os):

```
Defaults:delphix_os !requiretty
delphix_os ALL=(ALL) NOPASSWD: /bin/mkdir, /bin/rmdir, /bin/mount, /bin/umount
delphix_os ALL=(ALL) NOPASSWD: ALL
```

The dlpx_db_exec script

The formatted `dlpx_db_exec` script is presented here for completeness.

```
# This script allows customization of command execution with an alternate user
account.

# Arg $1 contains "-u<optional user account>" for the desired user under

# which database commands will be executed.

# By default this argument is ignored and the script is executed as the default

# account.

if [[ $1 != -u* ]]; then
echo "Incorrect command line parameters, -u<optional user account> is required as the
first parameter"
exit 1
fi

user_id=`echo $1 | sed -e "s/^-u//"`
echo "$user_id" >> /tmp/test.log
shift 1
echo "$user_id and $DB2_DB_NAME" >> /tmp/test.log
if [[ $user_id != "delphix_os" ]]; then
command=$(printf "%s " "$@")
sudo su - $user_id -c "cd /home/delphix_os;export DB2DBDFT=$DB2DBDFT;$command"
```

```
else
$@
fi
```

9.4.4 IBM Db2 installation and upgrade

The IBM Db2 connector is managed through the Delphix Continuous Data Engine's plugin framework. Plugins are distributed separately from the Delphix Continuous Data Engine. The plugin must be installed before IBM Db2 environments can be added, and before attempting any IBM Db2 virtualization operations.

9.4.4.1 Downloading the connector

Perform the following steps to download the IBM Db2 connector to your desktop system.

1. Log in to the Delphix download portal[358] using your credentials from the Delphix support portal.
2. Select **Delphix Product Release** > **Continuous Data Connectors** and choose IBM Db2 folder[359].
3. Locate the desired version of the plugin you need, and download the corresponding .zip file. The file format should appear as follows: `Plugin_DB2_<version>.zip`.
4. After the download is complete, extract the contents of the downloaded .zip file to obtain a JSON file. For example, `db2db.json`.

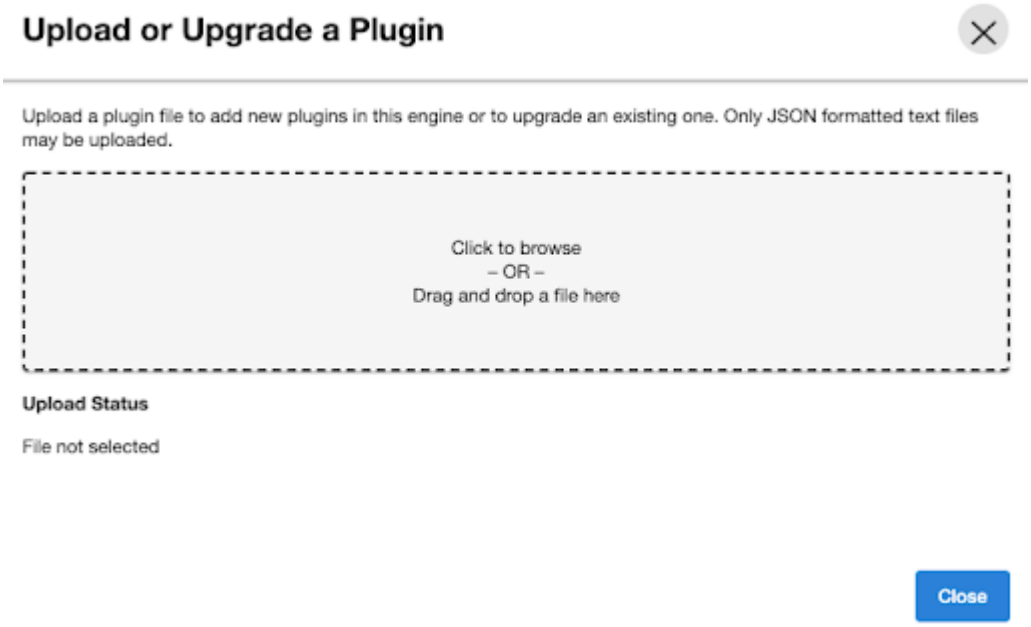
9.4.4.2 Installing the connector

Perform the following steps to upload and install a new connector.

1. Log in to the **Delphix Management application**.
2. Navigate to the **Manage** > **Plugins** page.
3. To upload a plugin, click the plus icon. This action displays a **Upload or Upgrade a Plugin** dialog window.
4. Browse to the JSON file on your desktop which you downloaded and extracted in the previous step
5. Upload the downloaded JSON (for example, db2db.json) file.
6. Once the upload is complete, close the popup window and proceed with adding the environment.

358 https://download.delphix.com/

359 https://download.delphix.com/folder/5353/Delphix%20Product%20Releases/Continuous%20Data%20Connectors/IBM%20Db2



For more information, refer to the Delphix Continuous Data Engine connector management[360] section.

9.4.4.3 Upgrading the connector

Users must upgrade their connectors to use new features and apply bug fixes.

Connectors are not upgraded through engine upgrades. However, some connector upgrades are only compatible with a minimum Delphix Continuous Data Engine version. Please ensure your engine is upgraded to a compatible version before upgrading the connector. For full compatibility details, refer to the IBM Db2 support matrix[361].

Perform the following process to upgrade your IBM Db2 connector to a later version.

> ⚠ Since v4.0.0, connector upgrades have been streamlined. However, skipping a version upgrade step before then may lead to configuration issues. If upgrading from a pre-v4.0.0, follow the Upgrade matrix (see page 0) instructions closely.

9.4.4.3.1 Upgrade process

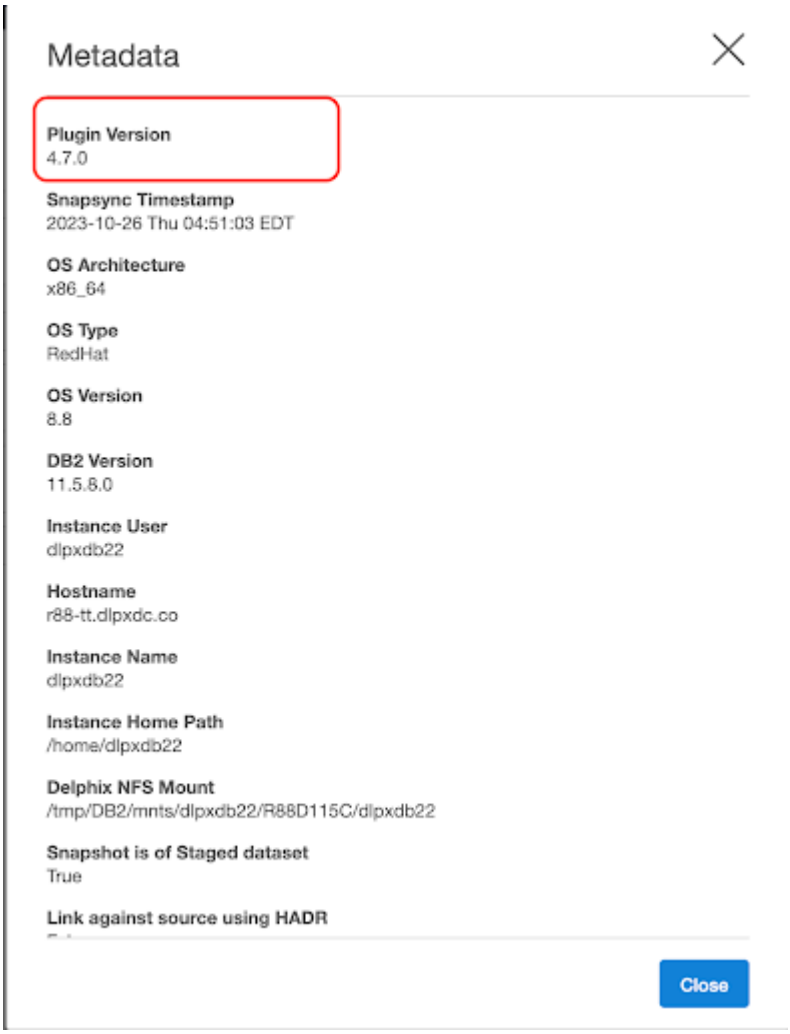
Perform the following steps when upgrading your IBM Db2 connector.

1. Disable all IBM Db2 dSources and VDBs. (Only required for IBM Db2 connector versions lower than v4.0.0).

360 https://cd.delphix.com/docs/latest/delphix-engine-plugin-management
361 https://cd.delphix.com/docs/latest/ibm-db2-support-matrix

Note: Consult the Upgrade matrix (see page 1176) instructions for more details.

2. Upload the newer connector version into the Delphix Continuous Data Engine. For more information, refer to the Installing the IBM Db2 connector[362] documentation.
3. Refresh all the associated IBM Db2 environments to trigger the discovery workflow. Verify that the workflow is completed.
4. Enable dSources and VDBs, if disabled, and create new dSource and VDB snapshots.
5. Review the newly created snapshot's metadata for the new plugin version to confirm the upgrade was successful.



362 https://cd.delphix.com/docs/latest/ibm-db2-installation-and-upgrade

9.4.4.3.2 Upgrade matrix

This section lists IBM Db2 connector versions that a user can upgrade from and to.

> You must adhere to the following upgrade considerations. Ignoring the below upgrade matrix may result in incompatible IBM Db2 data sources.
> - It is mandatory to upgrade to the v3.1.1 version of the IBM Db2 connector and then upgrade to v4.0.0. of the IBM Db2 connector.
> - After upgrading to v4.0.0, you may upgrade directly to any higher version.

Follow the matrix below to avoid connector corruption:

Your installed version	Required upgrade path	dSource and VDB Downtime Required?
v3.0.0 v3.0.1 v3.0.2	Upgrade to v3.1.0.	Yes
v3.1.0	Upgrade to v3.1.1	Yes
v3.1.1	Upgrade to v4.0.0.	Yes
v4.0.0+ (vSDK)	Upgrade to any more recently released version.	No

9.4.5 IBM Db2 connector operations

To successfully ingest and virtualize a IBM Db2 instance, you must perform the following connector operations in sequence:

- Discovering IBM Db2 installations (see page 1180)
- Linking IBM Db2 data sources (see page 1185)
- Provisioning IBM Db2 virtual databases (see page 1208)

9.4.5.1 Discovering IBM Db2 installations

Discovery is a process that allows the IBM Db2 connector to gather information about the IBM Db2 installation details in an environment. The database's discovery occurs during the environment creation or refresh operations.

Whenever there is any modification, such as installing a new database home to an environment, it is necessary to perform an environment refresh operation.

9.4.5.1.1 Prerequisites for discovery operation

The following requirements must be met before starting the discovery process.

When you add or refresh an environment to Delphix Continuous Data Engine, the IBM Db2 connector will automatically perform a discovery of all the IBM Db2 installed binaries on the server.

1. Install the IBM Db2 connector.
2. The environment user must be able to execute the `db2` command.
3. Verify environment and host compatibility with Delphix Continuous Data Engine and IBM Db2. For more information on the procedure, refer to the Host Checker documentation (see page 603).

9.4.5.1.2 Adding an IBM Db2 environment

This topic describes the process of adding IBM Db2 staging and target environment.

9.4.5.1.2.1 Prerequisites

To set up an IBM Db2 staging environment, ensure all the essential requirements as described in Requirements and Prerequisites (see page 1153) are met.

In addition, we recommend verifying environment and host compatibility with Delphix Continuous Data using the Host Checker tool. You can learn more about this in the Host Checker (see page 1038) section.

9.4.5.1.2.2 Procedure

Perform the following steps to add an environment to the Delphix Continuous Data Engine:

1. Login to the **Delphix Management** application.
2. Navigate to the **Manage > Environments** page.
3. Click the **Actions** menu and select **Add Environment**.
4. From the **Add Environment** dialog window, select **Unix/Linux.**
5. Select **Standalone Host**.
6. Click **Next**.
7. Provide the name of the environment.

8. Provide the **Host IP address** or **host FQDN**.
9. Provide the **SSH port**. The default value is 22.
10. Choose the **Login Type** that fits your setup.
    a. **Username and Password** – provide the OS username and password.
    b. **Username and Engine Public Key** – provide the OS username.
    c. **Username and Custom Key Pair** - provide the OS username.
    d. **Password Vault** – select from an existing Enterprise Password Vault.

11. Based on your **Login Type**, authenticate the environment by following the directions below:
    a. For **Login Type** 'Username and Password', enter the password associated with the user in step 11.
    b. For **Login Type** 'Username and Engine Public Key',
        i. Enter the **OS username**.
        ii. Click **View Public Key**.
        iii. Copy the public key that is displayed, and append it to the end of your `~/.ssh/authorized_keys` file. If this file does not exist, it must be created. **Note:** If you want to use public-key authentication for logging into your Unix-based environment, you have two options:
            - Use the Delphix Continuous Data Engine's key pair. For more information on how to use the Delphix Continuous Data Engine's key pair, refer to the How to use the Continuous Data Engine's key-pair (see page 1225) section.
            - Provide a key pair for this environment. As an alternative, a key pair specific for this environment can be provided via the API or CLI. For instructions, refer to CLI Cookbook: Setting up SSH key authentication for UNIX environment users (see page 2548).
        iv. For **Login Type** 'Username and Custom Key Pair':
            1. Enter the **OS username**.
            2. Provide **Private Key** for the environment.
            3. Provide **Public Key** for the environment.

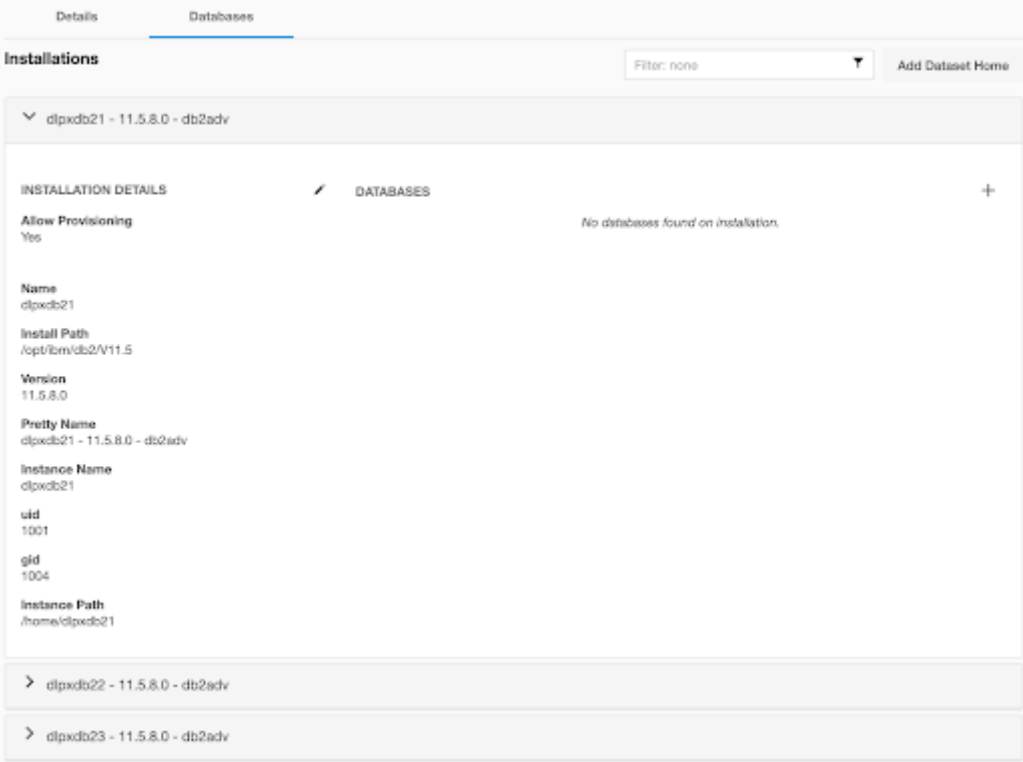
> Provide a Custom key pair for this environment. This key pair is specific for the environment and can be provided via the API or CLI as well. For instructions, refer to CLI Cookbook: Setting up SSH key authentication for UNIX environment users (see page 2548).

> - The addition of the public key is a one-time requirement per user and per environment. Additionally, you have the option to add public key authentication to an environment user's profile via CLI (for details, refer to the CLI Cookbook: Setting up SSH key authentication for UNIX environment users (see page 2548) topic).

- If you intend to use a low-privileged OS user, ensure that you have fulfilled the Prerequisites for Privilege Elevation using DLPX_DB_EXEC Script. (see page 1780)

12. Enter **Toolkit Path** (ensure that the toolkit path does not have any spaces).
13. Click on **Validate** to verify the user Credentials.
14. To provide your own Oracle Java Development Kit, select the **Provide my own JDK** checkbox and click **Next**.
    a. On the Java Development Kit page, provide an absolute path to your Oracle JDK and click **Next**.
15. Review the environment configuration **Summary** and click **Submit**.

After successfully adding and discovering the environment, you should be able to view the newly added environment listed under the Environments tab. In case it does not appear, click the refresh icon in your browser.



After the environment has been successfully added to the Delphix Continuous Data Engine, you should be able to see all the discovered repositories under the **Databases** tab.

Perform the following steps to access the information about any environment you have created:

1. Navigate to **Manage** > **Environments**.
2. Select the environment name.

9.4.5.1.3 Adding an IBM Db2 installation

If you have added a new IBM Db2 Installation to your environment and want it to be reflected in the Delphix Continuous Data Engine UI, you must refresh the environment. For more information, refer to the "Environment Refresh" section on the Environment management (see page 1017) page.

9.4.5.1.4 Adding IBM Db2 environment users

All Db2 Delphix Continuous Data Engine operations such as linking, snapshot, and provisioning always occur on the database level. As a result, we require that any DB2 instance that you wish to use either as staging or target must have its associated instance owner added as an environment user to the host within the Delphix Continuous Data Engine. Perform the following steps to add each IBM DB2 instance owner you use on the host.

It is important to note that Delphix Continuous Data Engine will not create new instances on the host and can only use existing DB2 instances that it finds on the host machine.

9.4.5.1.4.1 Procedure

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Environments**.
4. Click on the existing environment name you want to modify and open the environment information screen.
5. In the **Details** tab, click the Plus icon located next to **Environment users**.
6. Enter the **Username** and **Password** of the user.
7. Click on **Validate** to verify the user credentials
8. Click on **Add** to add the environment user.
9. To change the primary user for this environment, click the Star icon next to Environment Users. Only the primary user will be used for environment discovery.
10. To delete a user, click the Trash icon next to their username.

9.4.5.1.5 Environment-related IBM Db2 operations via CLI

There are many operations related to a Delphix Continuous Data Engine environment:

- Create/Discover
- Update
- Refresh
- Disable
- Enable
- Delete

CLI operations for each will appear below:

To discover an environment via CLI, log in to the Delphix Continuous Data Engine using the admin user account.

```
ssh admin@<CONTINUOUS-DATA-FQDN-or-IP>
```

9.4.5.1.5.1 Discovering the environment

Run the following commands to discover the environment.

```
environment
create
set hostEnvironment.name=rh88-stg
set hostParameters.host.address=rh88-stg.dlpxdc.co
set hostParameters.host.toolkitPath=/tmp
set primaryUser.name=delphix
set primaryUser.credential.password=delphix
commit
```

9.4.5.1.5.2 Updating the environment

Run the following commands to update the environment. The name of the environment can be updated from `rh88-stg` to `rh88-stg.dlpxdc.co` using the following command.

```
environment
select rh88-stg
update
set name=rh88-stg.dlpxdc.co
commit
```

9.4.5.1.5.3 Refreshing the environment

Run the following commands to refresh the environment.

```
environment
select rh88-stg.dlpxdc.co
refresh
commit
```

9.4.5.1.5.4 Disabling the environment

Run the following commands to disable the environment.

```
environment
select rh88-stg.dlpxdc.co
disable
commit
```

9.4.5.1.5.5 Enabling the environment

Run the following commands to enable the environment.

```
environment
select rh88-stg.dlpxdc.co
enable
commit
```

9.4.5.1.5.6 Deleting the environment

Run the following commands to delete the environment.

```
environment
select rh88-stg.dlpxdc.co
delete
commit
```

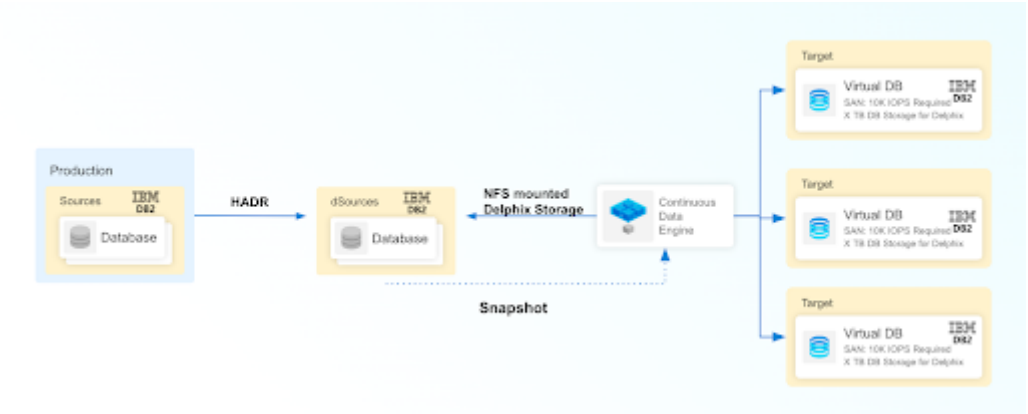
9.4.5.2 Linking IBM Db2 data sources

The Delphix Continuous Data Engine provides various ingestion methods for linking the dSource from an IBM Db2 source database.

- HADR (see page 1185)
- Backup only (see page 1191)
- Backup with archived logs (see page 1193)
- IBM Db2 staging push (see page 1197)
- Optional parameters (see page 1200)
- Take a IBM Db2 snapshot (see page 1201)
- Source sizing implementation for IBM Db2 dSource (see page 1203)
- IBM Db2 link operations via CLI (see page 1204)

9.4.5.2.1 HADR

The following diagram demonstrates the architecture of the Delphix Continuous Data Engine support for the IBM Db2 database with HADR between the source and staging database. HADR ingestion type is supported only with non-partitioned databases and the terms staging and standby can be used interchangeably in this context.



9.4.5.2.1.1 Prerequisites

- Be sure that the source and staging instances meet the host requirements and the databases meet the container requirements described in the requirements section (see page 1153).[363]
- The environment user or the privileged user must be able to run mount, unmount, mkdir and rmdir.
- The Instance owner/user must have `dbadm` and `secadm` permission.
- The instance owner/user must have a common group with the primary/os user.

HADR single standby database

- The following database configuration settings must be set at the source:
  - `db2 update db cfg for <DB_NAME> using HADR_LOCAL_HOST <PRIMARY_IP> HADR_LOCAL_SVC <PRIMARY_PORT > immediate`
  - `db2 update db cfg for <DB_NAME> using HADR_REMOTE_HOST <STANDBY_IP> HADR_REMOTE_SVC <STANDBY_PORT> immediate`
  - `db2 update db cfg for <DB_NAME> using HADR_REMOTE_INST <STANDBY_INSTANCE_NAME> immediate`
  - `db2 update db cfg for <DB_NAME> using HADR_SYNCMODE SUPERASYNC immediate`
- If the database configuration parameter LOGINDEXBUILD is set to OFF, do the following:
  - `db2 update db cfg for <DB_NAME> using LOGINDEXBUILD ON`
  - Turn off all connections to the database and reactivate the database
- If the database configuration parameter LOGARCHMETH1 is set to OFF, do the following:
  - `db2 update db cfg for <DB_NAME> using LOGARCHMETH1 XXXX (must be a valid log archiving method)`
  - Take an offline backup

363 https://docs.google.com/document/d/1cz2PZHJ62kRLjgQC6XgPLqBXklok6hoHtBVdzkLbkN0/edit#heading=h.ywjz4zydmkep

    - If LOGARCHMETH1 points to a third-party backup server (i.e. TSM or Netbackup) define LOGARCHMETH2 to disk
        i. `db2 update db cfg for <DB_NAME> using LOGARCHMETH2 DISK:<full path to archive log directory>`
            - Log files in the directory must be available from the time of the backup until the restore has successfully completed on the dSource.
- `db2 start hadr on db <DB_NAME> as primary by force`
- Take a full online backup as defined in the "How to take a full online backup of a db2 database" (see page 1225) section.
- Record the following information, as it must be entered on the Delphix Continuous Data Engine while creating the dSource.
    - HADR Primary hostname
    - HADR Primary SVC[364]
    - HADR Standby SVC[365] (auxiliary standby port)

HADR multiple standby databases

This assumes a single standby database HADR setup already exists. The existing standby will be referred to as the main standby. The new standby will be referred to as the auxiliary standby.

- The following database configuration settings must be set on the primary database:
    - `db2 update db cfg for <DB_NAME> using HADR_SYNCMODE <SYNC MODE> immediate -- set whichever sync mode you wish to use on your main standby`.
    - `db2 update db cfg for <DB_NAME> using HADR_TARGET_LIST "<MAIN_STANDBY_IP:MAIN_STANDBY_PORT|AUXILIARY_STANDBY_IP:AUXILIARY_STANDBY_PORT>" immediate`
        - You may have up to two auxiliary standbys defined and separated by a '|'. One of the auxiliary standbys must be the dSource.
- `db2 stop hadr on db <DB_NAME>`.
- `db2 start hadr on db <DB_NAME> as primary by force`.
- Take a full online backup as defined in the "Backup Source Database (see page 1225)" section below. While this backup is running, you may continue with the next step.

> Best practices to follow while taking a backup (see page 1225).

364 https://www.ibm.com/docs/en/db2/11.5?topic=parameters-hadr-local-svc-hadr-local-service-name
365 https://www.ibm.com/docs/en/db2/11.5?topic=parameters-hadr-remote-svc-hadr-remote-service-name

- The following database configuration settings must be set on the existing main standby database:
  - `db2 update db cfg for <DB_NAME> using HADR_SYNCMODE <same mode as defined in 1.a above.> -- It must be the same value used for the primary database`.
  - `db2 update db cfg for <DB_NAME> using HADR_TARGET_LIST "<PRIMARY_IP:PRIMARY_PORT|MAIN_STANDBY_IP:MAIN_STANDBY_PORT>"`.
- `db2 stop hadr on db <DB_NAME>`.
- `db2 start hadr on db <DB_NAME> as standby`.
- Record the following information, as it must be entered on the Delphix Continuous Data Engine while creating the dSource (the auxiliary standby database):
  - HADR Primary hostname
  - HADR Primary SVC
  - HADR Standby SVC (auxiliary standby port)
  - `HADR_TARGET_LIST <PRIMARY_IP:PRIMARY_PORT|MAIN_STANDBY_IP:MAIN_STANDBY_PORT>`

Enable SSL for HADR

- Create a local keystore in a dedicated directory with suitable permissions on both the primary and standby instances, using the provided command. Ensure that the Db2 instance user grants 700 permissions to the directory, while for privileged elevation or low-privileged users, 750 permissions should be assigned to the directory.
  - `gsk8capicmd_64 -keydb -create -db <KEYSTORE_NAME> -pw <PASSWORD> -type pkcs12`
    Note:
    - Use an extension like ".ps12," ".p12," or ".pfx" for the keystore file name.
    - Use `-stash` option to create a stash file at the same path as the keystore, with a file extension of `.sth`. At instance start-up, GSKit uses the stash file to obtain the password to the keystore.
- Update the database manager configuration for `SSL_SVR_KEYDB` and `SSL_SVR_STASH` (if the stash file is created) on both the primary and standby instances:
  - `db2 update dbm cfg using SSL_SVR_KEYDB <KEYSTORE_FILE_PATH>`
  - `db2 update dbm cfg using SSL_SVR_STASH <STASH_FILE_PATH>`
    Note:
    - You must restart the instance to apply the configuration changes on both the primary and standby instances.
    - While it's possible to configure HADR with SSL without specifying the stash file, it's generally recommended to use the stash file to enhance security and automation. If

you decide not to use a stash file and do not specify the `SSL_SVR_STASH` parameter in the dbm cfg, you would be prompted to enter the password manually each time you start the Db2 instance. This can be done using the db2start command, and you will be prompted to enter the password.

- You must restart the instance to apply the configuration changes on both the primary and standby instances.
- You must add certificates on both the primary and standby instances. You can create a self-signed digital certificate or use a CA-signed certificate.
  - Create a self-signed digital certificate:
    - `gsk8capicmd_64 -cert -create -db <KEYSTORE_NAME> -pw <PASSWORD> -label <CERT_LABEL_NAME> -dn "CN=<COMMON_NAME>"`
  - CA-signed certificate:
    - `gsk8capicmd_64 -cert -import -db <CA_SIGN_CERT> -pw <CA_SIGN_CERT_PWD> -target <KEYSTORE_NAME> -target_pw <PASSWORD>`
      Note: On the standby instance, use the label naming convention as `DB_NAME>_*_STANDBY`. If using CA signed certificate, rename the label based on a naming convention for the standby instance.
- Extract the respective certificate's public key to a certificate file on both the primary and standby instances:
  - `gsk8capicmd_64 -cert -extract  -db <KEYSTORE_NAME> -pw <PASSWORD> -label <CERT_LABEL_NAME> -target <CERT_NAME>.crt`
- Share the extracted certificates between the primary and standby instances.
- Add the shared certificates to the Keystore on both the primary and standby instances:
  - `gsk8capicmd_64 -cert -add  -db <KEYSTORE_NAME> -pw <PASSWORD> -label <CERT_LABEL_NAME> -file <CERT_NAME>`
- Check and confirm that the labels used for the certificates are present in both the primary and standby instance's keystore:
  - `gsk8capicmd_64 -cert -list -db <KEYSTORE_NAME> -pw <PASSWORD>`

9.4.5.2.1.2 Limitation

- All source, staging, and target environments must be on the same fix pack version.
- For HADR SSL/TLS, a local keystore of type "PKCS12" with either a password provided by the user or a stashed password is supported.

9.4.5.2.1.3 Procedure

1. Login to the **Delphix Management** application.

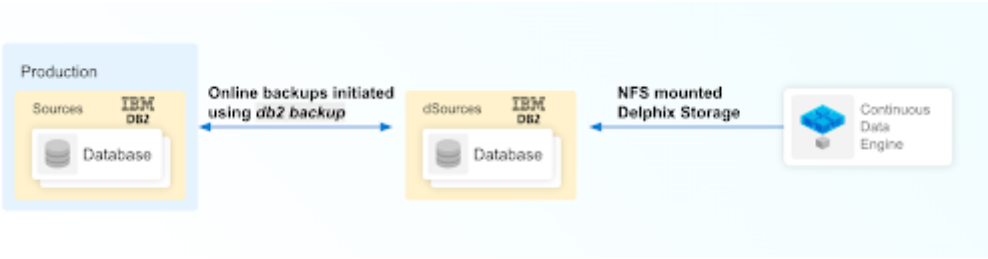
2. On the **Databases** tab of the Environment Management screen, add a source config (see page 1034) against the discovered staging instance.
3. Then, click **Manage** and select **Datasets.**
4. Click the **Plus (+)** icon and select **Add dSource**, you'll get a list of available source configs using which you can go for dSource creation.
5. In the **Add dSource** wizard, select the required source configuration.
6. Provide the Database Name.
7. Optionally, provide a custom mount base using the **Mount Base** (see page 1200) parameter.
8. Enter the complete **Backup Path** where the database backup file resides. If no value is entered, the default value used is the instance home directory. If there are multiple backup files for a non-DPF database on the backup path, the most current one will be used.
9. Optionally, provide multi-part backup using **Additional Backup Path** (see page 1200) parameter.
10. Optionally, you can add the **Parallelism factor for Database restore** (see page 1200) parameter**.**
11. Check the **Link against the source using HADR** checkbox. Selecting this checkbox reveals a collapsible group where the HADR-related details such as Primary hostname, service ports, target list and SSL/TLS related options that the end user wishes to configure HADR using SSL/TLS protocol.
12. If the dSource is set up to use **HADR** please enter the following fields. For more information about HADR please view High Availability Disaster Recovery (see page 1212)under IBM Db2 glossary.
    a. Enter a fully qualified HADR Primary Hostname. This is a required field for HADR and must match the value set for `HADR_LOCAL_HOST` on the master.
    b. Enter the port or `/etc/services` name for the HADR Primary SVC. This is a required field for HADR and uses the value set for `HADR_LOCAL_SVC` on the master.
    c. Enter the port or `/etc/services` name for the HADR Standby SVC. This is a required field for HADR and uses the value set for `HADR_REMOTE_SVC` on the master.
13. If the dSource is to use HADR with SSL, check the **Configure SSL** checkbox under **HADR DETAILS**. If a stash file is not available, provide the password for Key Store.
14. Optionally, you can set the **database configuration** (see page 1200) parameters using Staging Database Configuration Settings.
15. Optionally, you can provide a bash script using the **Use Plugin Hooks** (see page 1200) parameter.
16. Click **Next**.
17. Enter a name and select a group for your dSource. Adding a dSource to a dataset group lets you set Delphix Domain user permissions for that database and its objects, such as snapshots. See the topics under Users and Groups (see page 674) for more information.
18. Click **Next** and set the **Staging Environment** to be the same as the dSource host. Select the **Staging Environment User** to be the same as the instance owner of the dSource instance.
    Info: If you need to change or add an environment user for the dSource instance, follow the instructions here (see page 1183).

a. Ensure that the Staging environment must be the environment that you have chosen to create the source config.
   Select the **User** from the drop-down and set the **SNAPSHOT PARAMETERS** for the dSource snapshot.
   i. Select **Resynchronize dSource** to resynchronize the dSource. This will force a non-incremental load of data from the source similar to when the dSource was created. This action avoids duplicating storage requirements and maintains a Timeflow history.

19. In the **Policies** tab, apply policy details to the dSource if needed, and then click **Next** to continue.
20. In the **Hooks** tab, select a Hook Point and then click **+** to add a script that you want to run at that hook point. You have the flexibility to define scripts to run at multiple hook points in the process.
21. In the **Summary** tab, review the configuration profile for your dSource.
22. Click **Submit**. The Continuous Data Engine will initiate two jobs, namely DB_Link and DB_Sync to create the dSource. You can monitor the progress of these jobs on the Running Actions section on the top right panel, or by navigating to **System > Jobs**. Upon successful completion of the jobs, the database icon will change to a dSource icon on the **Environments > Host > Databases** screen. The dSource will also be listed under Datasets within its assigned group.

9.4.5.2.2 Backup only

The following diagram demonstrates the architecture of the Delphix Continuous Data Engine support for the IBM Db2 database with full online backup-only ingestion between the source and staging database.



9.4.5.2.2.1 Prerequisites

- Be sure that the source and staging instances meet the host requirements and that the databases meet the container requirements described in the Requirements and prerequisites (see page 1153) section.
- The environment user or the privileged user must be able to run mount, unmount, mkdir, and rmdir.
- The Instance owner/user must have `dbadm` and `secadm` permissions.
- The instance owner/user must have a common group with the primary/os user.

9.4.5.2.2.2 Limitation

- You can not apply archived logs and rollforward to a given time.
- To update the database, you need to re-ingest with newer backup files.

9.4.5.2.2.3 Procedure

1. Login to the **Delphix Management** application.
2. On the **Databases** tab of the Environment Management screen, add a source config (see page 1034) against the discovered staging instance.
3. Then, click **Manage** and select **Datasets.**
4. Click the **Plus (+)** icon and select **Add dSource**, you'll get a list of available source configs using which you can go for dSource creation.
5. In the **Add dSource** wizard, select the required source configuration.
6. Provide the Database Name.
7. Optionally, provide a custom mount base using the **Mount Base** (see page 1200) parameter.
8. Enter the complete **Backup Path** where the database backup file resides. If no value is entered, the default value used is the instance home directory. If there are multiple backup files for a non-DPF database on the backup path, the most current one will be used. If there are multiple sets of backup files for a DPF database on the backup path, the most recent one will be used.
9. Optionally, provide multi-part backup using **Additional Backup Path** (see page 1200) parameter.
10. Optionally, you can add the **Parallelism factor for Database restore** (see page 1200) parameter**.**
11. Optionally, users can set the **database configuration** (see page 1200) parameters using Staging Database Configuration Settings.
12. Optionally, users can provide a bash script using the **Use Plugin Hooks** (see page 1200) parameter.
13. Click **Next**.
14. Enter a name and select a group for your dSource. Adding a dSource to a dataset group lets you set Delphix Domain user permissions for that database and its objects, such as snapshots. See the topics under Users and Groups (see page 674) for more information.
15. Click **Next** and set the **Staging Environment** to be the same as the dSource host. Select the **Staging Environment User** to be the same as the instance owner of the dSource instance.
    Info: If you need to change or add an environment user for the dSource instance, follow the instructions here (see page 1183).

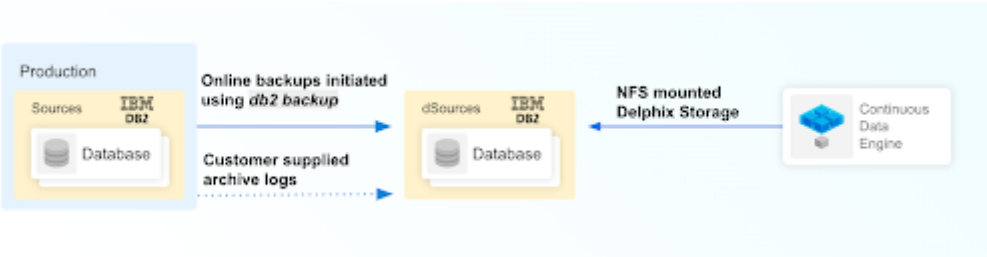
> Ensure that the Staging environment must be the environment that you have chosen to create the source config.
> Select the **User** from the drop-down and set the **SNAPSHOT PARAMETERS** for the dSource snapshot.
>
> 1. Select **Resynchronize dSource** to resynchronize the dSource. This will force a non-incremental load of data from the source similar to when the dSource was created. This action avoids duplicating storage requirements and maintains a Timeflow history.

16. In the **Policies** tab, apply policy details to the dSource if needed, and then click **Next** to continue.
17. In the **Hooks** tab, select a Hook Point and then click **+** to add a script that you want to run at that hook point. You have the flexibility to define scripts to run at multiple hook points in the process.
18. In the **Summary** tab, review the configuration profile for your dSource.
19. Click **Submit**. The Continuous Data Engine will initiate two jobs, namely DB_Link and DB_Sync to create the dSource. You can monitor the progress of these jobs on the Running Actions section on the top right panel, or by navigating to **System > Jobs**. Upon successful completion of the jobs, the database icon will change to a dSource icon on the **Environments > Host > Databases** screen. The dSource will also be listed under Datasets within its assigned group.

9.4.5.2.3 Backup with archived logs

The following diagram demonstrates the architecture of the Delphix Continuous Data Engine support for the IBM Db2 database with full online backup with end user-supplied archive logs ingestion between source and staging database.

With archived logs, either the database can be roll forwarded to the current time or to a specific point in time using the Rollforward to PIT (RFPIT) checkbox during ingestion. In case DPF is enabled, you need to place archive logs inside a folder with a name as NODE<Partition number> where <Partition number> is a four-digit number. For example, if the source environment has two partitions, then the user-provided log path will have folder names NODE0000 and NODE0001. Both these folders will have respective archive logs. Snapshot operation will be used to apply the archive logs.



9.4.5.2.3.1 Prerequisites

- Be sure that the source and staging instances meet the host requirements and the databases meet the container requirements described in the requirements section. (see page 1153)
- The environment user or the privileged user must be able to run mount, unmount, mkdir, and rmdir.
- The Instance owner/user must have `dbadm` and `secadm` permissions.
- The instance owner/user must have a common group with the primary/os user.
- A full backup must be accessible on the staging server, it could be a shared location but must be mounted on the staging host and visible as a file system.
- Archived logs must also be accessible on the staging server
    - The following command can be used to generate archive logs from the source database.
        i. `db2 archive log for database <database name>`
    - Instance users must have read permissions on logs so that Delphix-managed recovery can apply the logs.

    - Instance users must have read permissions on logs and additional write permissions on end user-provided log locations if the Delphix Continuous Data Engine is planning to use end user-provided log locations for applying the logs.
    - If the user has placed the archive logs at a user-provided log location, then the logs must be valid and the first active log of the staging database must be present in the user-provided log location.
- If the database configuration parameter LOGINDEXBUILD is set to OFF, do the following:
    - `update db cfg for <DB_NAME> using LOGINDEXBUILD ON`
    - Force off all connections to the database and reactivate the database.
- The end user does log purging from the provided log location. Delphix Continuous Data Engine only provides the list of logs that are eligible for purging.

9.4.5.2.3.2 Limitation

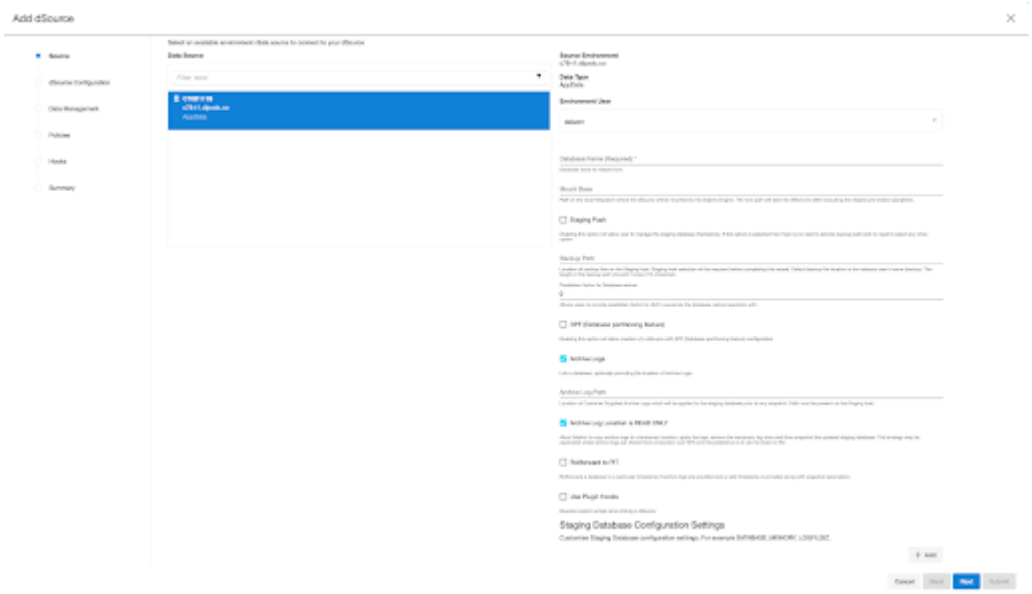
- Continuous updates to the dSource are not available. The snapsync operations apply new logs from the log location and keep the dSource updated with changes on the source.
- Read access to the staging database is not available.
- RFPIT timestamp is supported only with UTC timezone. If no time zone is provided, a local time zone will be applied.

9.4.5.2.3.3 Procedure

1. Login to the **Delphix Management** application.
2. On the **Databases** tab of the Environment Management screen, add a source config (see page 1034) against the discovered staging instance.
3. Then, click **Manage** and select **Datasets.**
4. Click the **Plus (+)** icon and select **Add dSource**, you'll get a list of available source configs using which you can go for dSource creation.
5. In the **Add dSource** wizard, select the required source configuration.
6. Provide the Database Name.
7. Optionally, provide a custom mount base using the **Mount Base** (see page 1191) parameter.
8. Enter the complete **Backup Path** where the database backup file resides. If no value is entered, the default value used is the instance home directory. If there are multiple backup files for a non-DPF database on the backup path, the most current one will be used. If there are multiple sets of backup files for a DPF database on the backup path, the most recent one will be used.
9. Optionally, provide multi-part backup using **Additional Backup Path** (see page 2623).
10. Optionally, you can add the **Parallelism factor for Database restore** (see page 1200)**.**
11. If you want to roll forward a snapshot to a user-specified timestamp, then you must select the **Rollforward to PIT** checkbox. You can also manually do this through the snapshot option. This option is visible only if you have selected the Archive logs checkbox since DB2 requires archive logs to

rollforward to a specific timestamp.
Info: The user-specified timestamp needs to be provided along with the timezone. For example, 2021-11-14-04.15.00 **UTC**. If the timezone is not entered, then the user's local timezone is applied.

12. If you are using **Archive Logs** for dSource creation complete the following fields:
    a. Check the **Archive logs** checkbox.
    b. Provide the log location of the archive logs. In this location, you can supply load copy files by appending schema and table name to it so that connector can process those load copy files during the rollforward process. Archive logs will be used for applying the logs during the snapshot operation. This location must be a part of the staging server and this location must be different from a full backup location.



    c. The **Archive Log Location is READ ONLY**(default) option indicates whether Delphix should copy the log files to the dSource's file system prior to the log apply process. If this is not selected the staging database will read the log files directly from the user-provided log location and perform the log apply operation.
    Note:

    - (Special considerations for Archive Log Location is READ ONLY) If this operation is performed on the source database and the user wants the Connector to process the load files, place the load copy files in your location and append the schema and table names (for example: `<DB Name>.<Number>.<Instance>.DBPART<partNumber>.<Timestamp>.001.<schema name>.<table name>`) to the load copy file names. If you do not want to process load copy files via the Connector place the load copy files in a similar structure as it is on the source or set a DB2 registry parameter `DB2LOADREC` .
    - Special consideration during dSource upgrade from older version connector to Archive Logs feature enabled connector. Suppose you have a dSource with an older

version of the connector and now you want to upgrade that dSource to **Archive Logs** feature, then you need to perform the following steps :

1. Go to the configuration tab of dSource and then click on the custom sub-tab.
2. Click on the edit icon.
3. Check the **Archive Logs** and **Archive Log Location is the READ ONLY** checkbox.

13. Click on submit and then go for RESYNC operation to upgrade the dSource.
14. Optionally, you can set the **database configuration** (see page 1200) parameters using Staging Database Configuration Settings.
15. Optionally, users can provide a bash script using the **Use Plugin Hooks** (see page 1200) parameter.
16. Click **Next**.
17. Enter a name and select a group for your dSource. Adding a dSource to a dataset group lets you set Delphix Domain user permissions for that database and its objects, such as snapshots. See the topics under Users and Groups (see page 674) for more information.
18. Click **Next** and set the **Staging Environment** to be the same as the dSource host. Select the **Staging Environment User** to be the same as the instance owner of the dSource instance.
    Info: If you need to change or add an environment user for the dSource instance, follow the instructions here (see page 1183).

> Ensure that the staging environment must be the environment that you have chosen to create the source config.
> Select the **User** from the drop-down and set the **SNAPSHOT PARAMETERS** for the dSource snapshot.
>
> 1. Click on the **Timestamp** input field to enter the timestamp at which the snapshot will be taken.
> The user-specified timestamp needs to be provided along with the timezone in YYYY-MM-DD-HH.MM.SS format. For example, 2022-03-12-04.15.00**UTC**. If the timezone is not entered, then the user's local timezone is considered.
> Providing the timestamp will only work if the user has checked the **Rollforward to PIT** option.
> 2. Select **Resynchronize dSource** to resynchronize the dSource. This will force a non-incremental load of data from the source similar to when the dSource was created. This action avoids duplicating storage requirements and maintains a Timeflow history.

19. In the **Policies** tab, apply policy details to the dSource if needed, and then click **Next** to continue.
20. In the **Hooks** tab, select a Hook Point and then click **+** to add a script that you want to run at that hook point. You have the flexibility to define scripts to run at multiple hook points in the process.
21. In the **Summary** tab, review the configuration profile for your dSource.

22. Click **Submit**. The Delphix Continuous Data Engine will initiate two jobs, namely DB_Link and DB_Sync to create the dSource. You can monitor the progress of these jobs on the Running Actions section on the top right panel, or by navigating to **System > Jobs**. Upon successful completion of the jobs, the database icon will change to a dSource icon on the **Environments > Host > Databases** screen. The dSource will also be listed under Datasets within its assigned group.

9.4.5.2.4 IBM Db2 staging push

The staging push transfers the responsibility of preparing the staging database to the end user. This ingestion method allows you to manually push the source data into the mount path backed by the Delphix Continuous Data Engine storage. It offers an alternative approach to the staging pull model where the Delphix Continuous Data Engine takes care of preparing the staging database.

9.4.5.2.4.1 Backups for staging push ingestion

Although the method used for pushing the source data into the mount point can be completely your choice, there are a few limitations when it comes to the type of backups used for the ingestion. The backups will have to be online backups since Delphix Continuous Data Engine relies on the archive logs to rollforward the database to specific timestamps and offline backups can be created by databases running circular logging rather than an archive logging mechanism.

The staging push ingestion method has been verified with the following mechanisms for both DPF and non-DPF databases:

- Backups Only with rollforward to MRT (Minimum Recovery Time)
- Backups with Archive logs (Rollforward to PIT)
- HADR (Only for non-DPF databases)

> The first snapshot created will be an empty directory and doesn't contain any database and hence a VDB cannot be provisioned using this VDB.

Prerequisites

Ensure that you meet the following requirements before you begin creating a dSource.

- The environment requirements are met by the source and staging environments as described Requirements for staging/target environments. (see page 1164)
- To use privilege elevation, you must meet the Privilege elevation requirements (see page 1164).

For more information, refer to Requirements and Prerequisites (see page 1153).

Limitation

- You cannot rollforward the staging database to the end of logs.

Procedure

Perform the following steps to create a dSource using the staging push ingestion method.

1. Login to the **Delphix Management** application.
2. Navigate to the **Manage > Environments** page.
3. On the **Databases** tab, choose a repository for your dSource and click the plus icon.
4. In the **Add Database** dialog window, provide the **name** of your source configuration and click **Add**.
5. Select your source configuration and click the **Add dSource** link located on the right.
6. In the **Source** tab, do the following:
   a. Provide the database name.
   b. Select the **Use Staging Push** checkbox. To use the Database partition feature along with staging push also check the Use DPF checkbox.
   c. Click **Next**.
7. In the **dSource Configuration** tab, provide a dSource name and then select a group for your dSource. Adding a dSource to a dataset group allows you to manage user permissions for that database and its objects, including snapshots. For more information, refer to the topics under Users and Groups (see page 674).
8. In the **Data Management** tab, specify your staging environment, user, and snapshot parameters. Select the **Resynchronize dSource** checkbox if you want to resynchronize the dSource. This operation will force a non-incremental load of data from the source. This operation is similar to creating a new dSource but prevents duplication of the storage requirements and maintains timeflow history. Click **Next** to proceed.
9. In the **Policies** tab, apply policy details to the dSource if needed, and then click **Next** to continue.
10. In the **Hooks** tab, select a Hook Point and then click **+** to add a script that you want to run at that hook point. You have the flexibility to define scripts to run at multiple hook points in the process.
11. In the **Summary** tab, review the configuration profile for your dSource.
12. Click **Submit**. The Continuous Data Engine will initiate two jobs, namely DB_Link and DB_Sync to create the dSource. You can monitor the progress of these jobs on the **Running Actions** section on the top right panel or by navigating to **System > Jobs**. Upon successful completion of the jobs, the database icon will change to a dSource icon on the **Environments > Host > Databases** screen. The dSource will also be listed under Datasets within its assigned group.
13. Look in connector logs generated at `<toolkit path>/Delphix_COMMON_<long id>/plugin/DB2_18f4ff11-b758-4bf2-9a37-719a22f5a4b8/logs/<instance_name>/<database_name>.diag.log` and follow the steps mentioned under Steps to restore and rollforward the staging database to restore and rollforward the database and bring the database to a consistent state.
14. Once the database has been restored and rollforwarded to a specific point in time, take a snapshot of the dSource as explained here (see page 1201).

9.4.5.2.4.2 Staging push with HADR

Staging push ingestion can also be used along with HADR to make sure the staging database is always in sync with the source database. The advantage of this method is that the user doesn't have to provide the path to archive logs and run rollforward commands manually. Instead, the staging database starts syncing with the source database once it is connected to the source database by following the steps provided in here. (see page 1185)

Prerequisites

Ensure that you meet the following requirements before you begin creating a dSource.

- The environment requirements are met by the source and staging environments as described Requirements for staging/target environments. (see page 1164)
- To use privilege elevation, you must meet the Privilege elevation requirements (see page 1164).

For more information, refer to Requirements and prerequisites (see page 1153).

Limitations

- HADR configuration should be set at source and staging manually
- You need to stop HADR on the staging database before performing any operation on the dsource.

Procedure

1. Create staging push dSource as described in Procedure above.
2. Set up HADR configuration at source and staging as mentioned here.[366]
3. Start HADR on source followed by staging:
   a. For Source: `db2 start hadr on db <DB_NAME> as primary by force`.
   b. For staging: `db2 start hadr on db <DB_NAME> as standby`.
4. Once the logsync is complete or satisfactory between the source and staging or when the customer is ready to take a snapshot, HADR needs to be stopped in standby by running:
   a. `db2 deactivate db <DB_NAME>`
   b. `db2 stop hadr on db <DB_NAME>`

> The logsync between primary (production) and staging can be verified by running db2pd -hadr on the staging host and looking for the parameter "HADR_CONNECT_STATUS" and "HADR_LOG_GAP".

366 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/31443272

5. Take a snapshot of the the dSource by following the steps here (see page 1201).

9.4.5.2.5 Optional parameters

9.4.5.2.5.1 Custom mount base

Users can specify their own mount path to host the dSource dataset by specifying the base path in the **Mount Base** field. Make sure this path exists on the staging host.

9.4.5.2.5.2 Additional backup paths

Once the Backup path is entered, a new field will pop up below the Backup path namely the **Additional Backup path** where you can provide multiple backup paths if the backup is split into multiple filesystems during the backup operation.

Parallelism factor for database restore

Optionally, you can add the **Parallelism factor for Database restore** to provide the degree of parallelization to run the database restore operation. If no value is entered, the IBM Db2 connector inherently assigns a parallelism factor.

- Parallelism factor depends on the resources provided on the staging host; users may not see any difference in performance if there aren't enough resources for the IBM Db2 connector to implement the parallelism.
- You can provide any positive integer value for the parallelism factor. IBM Db2 connector will provide that level of parallelism given that the system has appropriate resources to offer.
- Parallelism factor is inversely proportional to the Database restore time, which implies that a higher parallelism factor will result in reduced restoration time.
- For optimal performance during restoration, make sure to provide the same parallelism factor while creating the backups.

Config Settings

Optionally, you can set the database configuration parameters during the linking operation in the **Config Settings** section. For example `DATABASE_MEMORY`, `LOGFILSIZ`.

Connector defined hooks

Optionally, users can provide a bash script as a connector-defined hook. Delphix Continuous Data Engine has two connector managed Hooks:

- **Plugin Pre-Restore Hook:** The bash script entered in this field will get executed before restoration. This hook is useful in such cases where the users want to prepare the full backup before the restore operation.
- **Plugin Post-Restore Hook**: The bash script entered in this field will get executed after completion of the restore operation. This hook is useful in such cases where the user wants to purge the full backup file after the restore operation.

Plugin-managed pre-Hook - You can copy the required full backup file from some remote location to the staging host. Below is an example of such a bash script (the name of the below script is copyBackup.sh)

```
#!/bin/bash


# Copying backup file from a remote host remote.delphix.com
scp auto1051@remote.delphix.com:/backuplocation/R74D105A.0.auto1051.DBPART000.2018051
0073639.001 /db2backup/


The user can provide the above script in Plugin managed pre-Hook field as "sh
copyBackup.sh"


2. Plugin managed post-Hook - User can purge the full backup file from the staging
host. Below is an example of such bash script (name of below script is
purgeBackup.sh)
#!/bin/bash


# Purging full backup file from staging host for saving the staging storage
rm -f /db2backup/R74D105A.0.auto1051.DBPART000.20180510073639.001


The user can provide the above script in Plugin managed post-Hook field as "sh
purgeBackup.sh"
```

9.4.5.2.6 Take a IBM Db2 snapshot

Taking a snapshot creates a new snapshot entry in the dSource's Timeflow. The staging pull dSource allows you to select the parameters before taking the snapshot. These parameters are provided as input to the snapshot operation. The default snapshot operation uses parameters defined in the dSource environment.

9.4.5.2.6.1 Snapshot (Default)

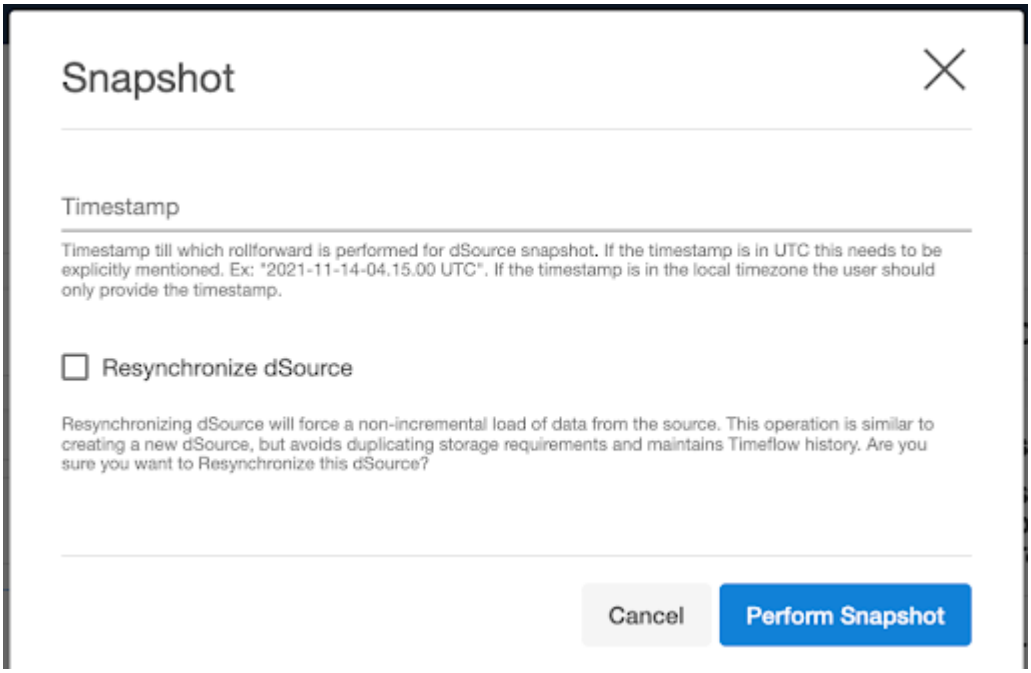
This section lists the steps to take a snapshot and delete the same.

1. Login to the **Delphix Management** application.
2. Click **Manage** and select **Datasets** from the dropdown list.
3. Select the dSource you want to Snapshot.

4. Click the **Camera** icon. Alternatively, click the arrow next to the Camera icon and select **Snapshot (default)**.
5. From the Snapshot dialog prompt, click on **Yes**.
6. Click **View:** under **Timeflow** to verify the Snapshot you just created.
7. To delete, select the snapshot you just created, and from the Actions menu (...) select **Delete Snapshot**.
8. In the Delete Snapshot dialog, select **Delete**.
9. Click **View:** under **Timeflow** to verify the Snapshot, you just deleted.

9.4.5.2.6.2 Snapshot (with params)

1. Login to the **Delphix Management** application.
2. Click **Manage** and select **Datasets** from the dropdown list.
3. Select the dSource you want to Snapshot.
4. Click the arrow next to the Camera icon and select **Snapshot with Params..**.
5. From the Snapshot dialog provide the required options and select **Perform Snapshot.**
    a. Provide **Timestamp,** when the database needs to be rollforwarded to a specific timestamp
    b. Select the **Resynchronize the dSource** checkbox when the database needs to be resynced.
6. Click **View:** under **Timeflow** to verify the Snapshot you just created.
7. To delete, select the snapshot you just created, and from the Actions menu (...) select **Delete Snapshot**.
8. In the Delete Snapshot dialog, select **Delete**.
9. Click **View:** under **Timeflow** to verify the Snapshot, you just deleted.

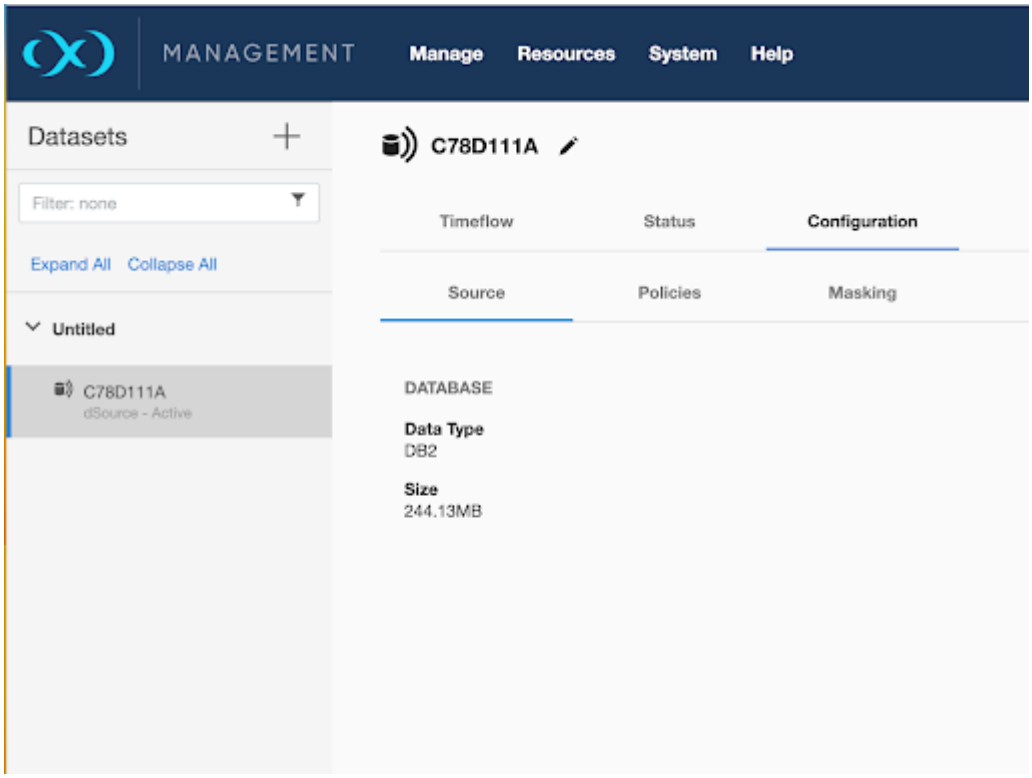


9.4.5.2.7 Source sizing implementation for IBM Db2 dSource

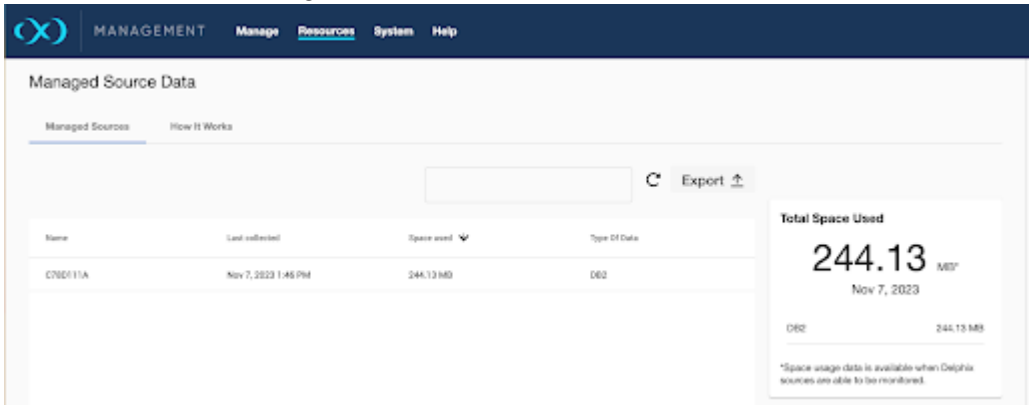
The source sizing feature enables you to visualize the dSource size based on the total allocated space on the mount point provided by the Delphix Continuous Data Engine for the dSource.

9.4.5.2.7.1 Procedure

- For all Non-AIX hosts, the size of dSource can be calculated using `du -sb <staging db mount path>` command that provides the correct volume and apparent size of the dSources.
- For AIX hosts, the size of the dSource can be calculated by executing any one of the following commands depending on whether the database is open for connections or not.
  a. `select sum(TBSP_TOTAL_SIZE_KB)*1024 from sysibmadm.TBSP_UTILIZATION`, if the database is open for connections and it is possible to connect to the database.
  b. `db2dart <dbname> /DTSF`, if the database is not open for connections and is in a rollforward pending state.
- For AIX hosts, the connector won't be able to display the size of the database during the creation of dSource, since it needs to be restored to read the size of the database.
- If the connector is unable to list the database or the database is yet to be restored, then the status tab under dSource displays **Not available**.
- The value can be found in two distinct locations within the Delphix Continuous Data Engine's UI.
  - Under the Configuration tab of dSource.



- Under **Resources** > **Managed Source Data**.



9.4.5.2.8 IBM Db2 link operations via CLI

9.4.5.2.8.1 dSource creation from CLI

To create a dSource via CLI, you must create a source configuration first. For more information on creating a source configuration, refer to Add Source Config using CLI (see page 1041).

To create a dSource and perform operations via CLI, log in to the Delphix Continuous Data Engine using the admin user account.

```
ssh admin@<CONTINUOUS-DATA-FQDN-or-IP>
```

9.4.5.2.8.2 Set default linking parameters (This is common for all types of ingestions)

Run the following commands to set the default linking parameters. The following example uses **testds** source config.

```
database link;
set linkData.type=AppDataStagedLinkData;
set name=testds
set group=Untitled
set linkData.config=testds
set linkData.environmentUser=dlpxdb21
set linkData.stagingEnvironment=rh88-stg.dlpxdc.co
set linkData.stagingEnvironmentUser=dlpxdb21
set linkData.syncParameters.parameters={\"resync\":true}
```

9.4.5.2.8.3 Link parameters for backup only

Run the following commands to link parameters for backup-only ingestion type.

```
set linkData.parameters="{\"dbName\":\"R88D115A\",\"backupPath\":\"/db2backup/11_5\",
\"stagingPushFlag\":false,\"dpfFlag\":false,\"monitorHADR\":false,
\"toolkitHookFlag\":false,\"logIngestionFlag\":false,\"copyToApplyFlag\":false,
\"rfPIT\":false}";
```

For DPF

```
set linkData.parameters="{\"dbName\":\"R88D115A\",\"backupPath\":\"/db2backup/11_5\",
\"stagingPushFlag\":false,\"dpfFlag\":true,\"monitorHADR\":false,
\"toolkitHookFlag\":false,\"logIngestionFlag\":false,\"copyToApplyFlag\":false,
\"rfPIT\":false}";
```

9.4.5.2.8.4 Link parameters for backup with archived logs

Run the following commands to link parameters for backup and archived logs.

```
set linkData.parameters="{\"dbName\":\"R88D115A\",\"backupPath\":\"/db2backup/11_5\",
\"stagingPushFlag\":false,\"dpfFlag\":false,\"monitorHADR\":false,
\"toolkitHookFlag\":false,\"logIngestionFlag\":true,\"logPath\":\"/home/dlpxdb21/
logs\",\"copyToApplyFlag\":false,\"rfPIT\":false}";
```

For DPF

```
set linkData.parameters="{\"dbName\":\"R88D115A\",\"backupPath\":\"/db2backup/11_5\",
\"stagingPushFlag\":false,\"dpfFlag\":true,\"monitorHADR\":false,
\"toolkitHookFlag\":false,\"logIngestionFlag\":true,\"logPath\":\"/home/dlpxdb21/
logs\",\"copyToApplyFlag\":false,\"rfPIT\":false}";
```

9.4.5.2.8.5 Link parameters for backup with archived logs (Using RFPIT option)

Run the following commands to link parameters for backup with archived logs with RFPIT.

```
set linkData.parameters="{\"dbName\":\"R88D115A\",\"backupPath\":\"/db2backup/11_5\",
\"stagingPushFlag\":false,\"dpfFlag\":false,\"monitorHADR\":false,
\"toolkitHookFlag\":false,\"logIngestionFlag\":true,\"logPath\":\"/home/dlpxdb21/
logs\",\"copyToApplyFlag\":false,\"rfPIT\":true}";

set linkData.syncParameters.parameters={\"resync\":true,\"rtimestamp\":\"2023-10-25-0
8.50.00\"}
```

For DPF

```
set linkData.parameters="{\"dbName\":\"R88D115A\",\"backupPath\":\"/db2backup/11_5\",
\"stagingPushFlag\":false,\"dpfFlag\":true,\"monitorHADR\":false,
\"toolkitHookFlag\":false,\"logIngestionFlag\":true,\"logPath\":\"/home/dlpxdb21/
logs\",\"copyToApplyFlag\":false,\"rfPIT\":true}";

set linkData.syncParameters.parameters={\"resync\":true,\"rtimestamp\":\"2023-10-25-0
8.50.00\"}
```

9.4.5.2.8.6 Link parameters for HADR

Run the following commands to link parameters for HADR.

```
set linkData.parameters="{\"dbName\":\"C78d111A\",\"backupPath\":\"/db2backup/11_1\",
\"additionalBackupPath\":[],\"stagingPushFlag\":false,\"dpfFlag\":false,
\"monitorHADR\":true,\"toolkitHookFlag\":false,\"logIngestionFlag\":false,
\"copyToApplyFlag\":false,\"rfPIT\":false,\"hadrDetails\":{\"hadrPrimaryHostname\":
\"c78-s1.dlpxdc.co\",\"hadrPrimarySVC\":\"60055\",\"hadrStandbySVC\":\"60056\",
\"hadrTargetList\":\"\",\"configureSSL\":false}}"
```

9.4.5.2.8.7 Link parameters for HADR with SSL

Run the following commands to link parameters for HADR with SSL.

```
set linkData.parameters="{\"dbName\":\"C78d111A\",\"backupPath\":\"/db2backup/11_1\",
\"additionalBackupPath\":[],\"stagingPushFlag\":false,\"dpfFlag\":false,
\"monitorHADR\":true,\"toolkitHookFlag\":false,\"logIngestionFlag\":false,
\"copyToApplyFlag\":false,\"rfPIT\":false,\"hadrDetails\":{\"hadrPrimaryHostname\":
\"c78-s1.dlpxdc.co\",\"hadrPrimarySVC\":\"60055\",\"hadrStandbySVC\":\"60056\",
\"hadrTargetList\":\"\",\"configureSSL\":true,\"keyStorePassword\":\"delphix\"
}}"
```

9.4.5.2.8.8 Link parameters for additional backup path

```
set linkData.parameters="{\"dbName\":\"R88D115C\",\"backupPath\":\"/db2backup/11_5/
backup_path1\",\"additionalBackupPath\":[{\"bpath\":\"/db2backup/11_5/
backup_path2\"}, {\"bpath\":\"/db2backup/11_5/backup_path3\"}],
\"stagingPushFlag\":false,\"dpfFlag\":false,\"monitorHADR\":false,
\"toolkitHookFlag\":false,\"logIngestionFlag\":true,\"logPath\":\"/home/dlpxdb21/
logs\",\"copyToApplyFlag\":false,\"rfPIT\":false}";
```

For DPF

```
set linkData.parameters="{\"dbName\":\"R88D115C\",\"backupPath\":\"/db2backup/11_5/
backup_path1\",\"additionalBackupPath\":[{\"bpath\":\"/db2backup/11_5/
backup_path2\"}, {\"bpath\":\"/db2backup/11_5/backup_path3\"}],
\"stagingPushFlag\":false,\"dpfFlag\":true,\"monitorHADR\":false,
\"toolkitHookFlag\":false,\"logIngestionFlag\":true,\"logPath\":\"/home/dlpxdb21/
logs\",\"copyToApplyFlag\":false,\"rfPIT\":false}";
```

9.4.5.2.8.9 Link parameters for staging push

Run the following commands to link parameters for staging push.

```
set linkData.parameters="{\"dbName\":\"R88D115A\",\"stagingPushFlag\":true,
\"dpfFlag\":false,\"monitorHADR\":false,\"toolkitHookFlag\":false,
\"logIngestionFlag\":false,\"copyToApplyFlag\":false,\"rfPIT\":false}";
```

For DPF

```
set linkData.parameters="{\"dbName\":\"R88D115A\",\"stagingPushFlag\":true,
\"dpfFlag\":true,\"monitorHADR\":false,\"toolkitHookFlag\":false,
\"logIngestionFlag\":false,\"copyToApplyFlag\":false,\"rfPIT\":false}";
```

9.4.5.2.8.10 Commit

Run the following command to commit the change

```
commit;
```

9.4.5.2.8.11 Disable

Run the following command to disable the dSource.

```
source
select testdb
disable
commit
```

9.4.5.2.8.12 Enable

Run the following command to enable the dSource.

```
source
select testdb
enable
commit
```

9.4.5.3 Provisioning IBM Db2 virtual databases

The virtual database (VDB) is a Delphix Continuous Data Engine virtualized copy of your source database that runs on your target environment. A VDB is provisioned from a dSource or VDB snapshot.

9.4.5.3.1 Prerequisites

You must meet the following requirements to create a VDB:

1. Identify an originating dataset and snapshot.
    a. Link a dSource by following the instructions in Linking a IBM Db2 dSource (see page 1185).
    b. Select a VDB that has been previously provisioned.

2. Set up an IBM Db2 target environment as described in IBM Db2 support and requirements. (see page 1153)
3. To use privilege elevation, you must meet the Privilege elevation requirements (see page 1164).
4. Once you have restored the databases into your target environment, If the version of IBM® Db2® in your source environment is at a different fix pack level than the version of Db2 in your target environment, then you need to update the 11.1 version databases with `db2updv111` command and 11.5 databases with `db2updv115` command utility. Run the below command as a Pre-snapshot hook:

```
db2updv111 -d <DBNAME> -a
db2updv115 -d <DBNAME> -a
```

More details on the above can be found here (see page 1225).

9.4.5.3.2 Limitations

By default, the IBM Db2 connector will not allow provisioning a VDB on an instance that contains a database that has an identical name to the source database.

For example,

1. When we are provisioning a VDB from a VDB then the source VDB will be treated as the source database.
2. When we are provisioning a VDB from a dSource then a staging database will be treated as the source database.

If you want to provision a VDB on an instance that contains a source database, then you need to follow the steps mentioned here. (see page 1024)

9.4.5.3.3 Procedure

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Datasets**.
4. Select a **dSource**.
5. Select a **snapshot** from which you want to provision.
6. Click the Provision **VDB** icon to open the **Provision VDB** wizard.
7. Select a target environment from the left pane.
8. Select an **Installation** to use from the dropdown list of available DB2 instances on that environment.
9. Set the **Environment User** to be the Instance Owner. Note: The picking of instance owners is only possible if you have multiple environment users set on that host.

10. Provide VDB Name as database name as parameter.
11. Optionally, provide the VDB Mount Base for the VDB.
12. Optionally, set the database configuration parameters for the VDB.
13. Click **Next**.
14. Select a **Target Group** for the VDB. Click the green **Plus** icon to add a new group, if necessary.
15. Select a **Snapshot Policy** for the VDB.
16. Click **Next**.
17. Specify any desired hook operations.
18. Click **Next**.
19. Review the **Provisioning Configuration** and **Data Management** information.
20. Click **Submit**. When provisioning starts, you can review the progress of the job in the **Databases** panel, or in the **Job History** panel of the **Dashboard**. When provisioning is complete, the VDB will be included in the group you designated and listed in the **Databases** panel. If you select the VDB in the Databases panel and click the **Open** icon, you can view its card, which contains information about the database and its Data Management settings. Once the VDB provisioning has successfully completed, if the source and target instance ids are not the same, you may want to grant secadm and dbadm on the database to the target instance id. For more information, refer to Requirements and prerequisites (see page 1153).

9.4.5.3.4 Provisioning operations of IBM Db2 via CLI

To create a VDB via CLI, you must first create a dSource. For more information on creating a dSource, refer to Link operation via CLI section (see page 1204).

To create a VDB and perform operations via CLI, log in to the Delphix Continuous Data Engine using the admin user account.

```
ssh admin@<CONTINUOUS-DATA-FQDN-or-IP>
```

9.4.5.3.4.1 Provision

Run the following commands to provision a VDB. The following example uses `testdb` dSource.

Common commands

```
database provision
set type=AppDataProvisionParameters
set container.name=testvdb
set container.group=Untitled
set source.allowAutoVDBRestartOnHostReboot=true
set sourceConfig.name=""
```

```
set sourceConfig.repository="dlpxdb21 - 11.5.8.0 - db2adv"
set sourceConfig.environmentUser=dlpxdb21
set timeflowPointParameters.container=testdb
```

9.4.5.3.4.2 With default mount base

```
set source.parameters="{\"databaseAliasName\":\"testvdb\",\"config_settings_prov\":
[{\"propertyName\":\"NUM_DB_BACKUPS\",\"value\":\"11\"}]}";
```

9.4.5.3.4.3 VDB with custom mount base

```
set source.parameters="{\"databaseAliasName\":\"testvdb\",\"VDBmountLocation\":\"/
vdb_mount_base/test\",\"config_settings_prov\":[{\"propertyName\":\"NUM_DB_BACKUPS\",
\"value\":\"11\"}]}";
```

9.4.5.3.4.4 Commit

Run the following command to commit the change.

```
commit
```

9.4.5.3.4.5 Disable

Run the following commands to disable a VDB.

```
source
select testvdb
disable
commit
```

9.4.5.3.4.6 Enable

Run the following command to enable a VDB.

```
source
select testvdb
enable
commit
```

9.4.5.3.4.7 Stop

Run the following command to stop a VDB.

```
source
select testvdb
stop
commit
```

9.4.5.3.4.8 Start

Run the following command to start a VDB.

```
source
select testvdb
start
commit
```

9.4.5.3.4.9 Refresh

Run the following to refresh a VDB to the latest snapshot, where the parent dsource is `testdb`.

```
database
select testvdb
refresh
set timeflowPointParameters.container=testdb
commit
```

9.4.6 IBM Db2 troubleshooting

This section covers the following topics:

- IBM Db2 advanced configuration (see page 1212)
- IBM Db2 logging (see page 1218)
- Managing IBM Db2 log rotation and log level (see page 1218)

9.4.6.1 IBM Db2 advanced configuration

This topic provides details on customizing IBM Db2 connector logging and controlling supported dataset configurations. You can modify the following parameters at any stage of the dataset lifecycle. The IBM Db2 connector will ignore values for a parameter if the values do not fit the description below and will continue the operation with the default set of values for that particular parameter.

9.4.6.1.1 Configuration parameters for connector versions 4.x and later

Parameter Value	Usage
usersHaveCommonGroup	Sets the permission for the ***"mnts"*** directory located under the `/<toolkit_path>/DB2/` and for the `logs` directory located under `<toolkit path>/Delphix_COMMON_<long id>/plugin/DB2_18f4ff11-b758-4bf2-9a37-719a22f5a4b8/logs/`. The value is set to True, if the primary group of the primary environment user (user which is used to do discovery) is shared with the instance user otherwise it is set to False. Once the necessary changes are made, refresh the environment in order to apply the changes. **Default Value: True**
allowSourceDbOnSameInstance	IBM Db2 connector restricts the provisioning of a VDB on a DB2 instance which contains a database with the same name as the VDB's source database For example: • When provisioning a VDB from a VDB then the latter is treated as the source database. • When provisioning a VDB from a dSource then the staging database representing the dSource is treated as the source database. If the user wants to provision a VDB on an instance which contains a database with the same name as the VDB's source database, then the parameter allowSourceDbOnSameInstance must be set to true. **Default value: False**
restorePipelineLimit	Db2 DPF connector performs restoration of all non-catalog partitions in parallel. The parameter `restorePipelineLimit` allows the user to configure the number of partitions that could be restored in parallel by the connector. The valid value for this field is any positive integer. **Default value: 10**

Parameter Value	Usage
skipArchiveLogValidation	This option allows the user to set the connector to skip the validation process of end user supplied archive logs and perform a partial validation.To[367] be able to use this functionality, the parameter value must be set to True. **Default value: False**
skipDataPathVerification	This option allows the user to set the connector to skip the data path verification process during VDB snapshot operation.[368] The parameter should be used when upgrading from connector versions <= 3.1.1 and users do not wish to perform VDB re-provision or refresh after the connector upgrade. To be able to use this functionality, the parameter value must be set to True. **Default value: False**
level	Sets the connector debug log levels. **Default value: INFO**
logFileSize	Sets the maximum connector log file size in MB. **Default value: 1**
retention	Sets the maximum number of connector log files. **Default value: 2**

9.4.6.1.1.1 Connector Config file

After adding the environment to Delphix Continuous Data Engine, you can find the connector configuration file, `db2_plugin.conf` on the remote environment. It is typically located at the following specified path:

```
<toolkit path>/Delphix_COMMON_<long id>/plugin/DB2_18f4ff11-
b758-4bf2-9a37-719a22f5a4b8/db2_plugin.conf
```

This file is divided into 2 sections - “plugin_custom_parameters” and “plugin_logging_parameters”.

```
# Sample of file db2_plugin.conf
#
# The parameters must not be left padded with spaces. The plugin will otherwise not
accept the specified parameter
```

367 http://validation.To
368 http://validation.to/

```
# values and will use the default values.
#
[plugin_custom_parameters]
#
# If the user has not shared a common group between primary environment user and
instance user.
# Then the user needs to set parameter usersHaveCommonGroup as False.
#
usersHaveCommonGroup=True
#
# During provision operations the plugin checks if the target instance contains a
database name which is identical to
# the source DB name from which the VDB is to be created. By default the plugin will
never allow creating a VDB on the
# instance where the source database already exists, or on any instance that has a
database with the same name. If the
# users still want to create a VDB on an instance which contains a database with the
same name as the source DB then
# they need to set parameter allowSourceDbOnSameInstance as True.
#
allowSourceDbOnSameInstance=False
#
# If the users want to control the number of parallel restores then they can tune the
parameter restorePipelineLimit.
# By default the value of this parameter would be 10.
#
restorePipelineLimit=10
#
# The archive logs validation process can be skipped if the end user so chooses.
Instead, a speedier technique can be used,
# in which the archive logs are only validated for the first and last logs and the
file sequences in between are confirmed.
#
skipArchiveLogValidation=False
#
# When upgrading from versions 3.1.1 or below, the verification of paths can be
skipped if the end user chooses to avoid
# the VDB snapshot error due to changed data path structure
#
skipDataPathVerification=False
[plugin_logging_parameters]
#
# This flag will set the debug level of plugin logs on the remote server (both
staging and target). There are two levels:
# Info
# Debug
# The above are the only valid values that can be assigned. If any other value is
assigned, the plugin will set the
# default level as Info. the string is case insensitive so info, Info and INFO are
acceptable.
#
level=INFO
#
```

```
# This parameter will set the maximum size of the active log file in MB. Once this
 limit is reached, the plugin will
# rotate the log as per the retention property defined below. The minimum value of
this parameter is 1 MB and maximum
# value is 10 MB. This parameter only takes in a positive integer value. The default
 value is 1 MB
# logFileSize=<positive integer>
#
logFileSize=1
#
# Whenever the size of <DB Name>.diag.log file exceeds the value provided by
logFileSize parameter then the plugin will
# rename the active log file to <DB Name>.diag.log.<number> and a new log file with
name <DB Name>.diag.log will be
# generated. For example, if LogRetention is set to 4, the plugin will have the
following log files: <DB Name>.diag.log,
# <DB Name>.diag.log.1, <DB Name>.diag.log.2, <DB Name>.diag.log.3, <DB
Name>.diag.log.4.
# File <DB Name>.diag.log.4 will be the oldest one.
# File <DB Name>.diag.log.1 will be the most recent archive log file.
# File <DB Name>.diag.log will be the active log file.
# The minimum and default value for retention flag is 2.
# retention=<positive integer>
#
retention=2
```

Please adhere to the configuration parameters provided above. Adding unknown parameters in this file could result in workflow errors.

9.4.6.1.2 Configuration parameters for connector versions 3.x and prior

The advanced config file is named `advanceConfFlag.conf` and is created under the `<Toolkit directory>` during discovery or environment refresh operation if the file does not pre-exist already. This file will have the following configurable parameters:

Parameter Value	Usage
notCommonGroupFlag	Sets the permission for the "mnts", "logs" and "code" directory located under the `/<toolkit_path>/DB2/`. The value is set to True, if the primary group of the primary environment user (user which is used to do discovery) is shared with the instance user otherwise it is set to False. Once the necessary changes are made, refresh the environment in order to apply the changes. **Default Value: True** By default, the `notCommonGroupFlag` parameter is commented out in the file. This means that the connector implicitly assumes its value as false.
allowSourceDBonSameInst	DB2 connector restricts the provisioning of a VDB on a DB2 instance which contains a database with the same name as the VDB's source database For example: • When provisioning a VDB from a VDB then the latter is treated as the source database. • When provisioning a VDB from a dSource then the staging database representing the dSource is treated as the source database. If the user wants to provision a VDB on an instance which contains a database with the same name as the VDB's source database, then the parameter allowSourceDbOnSameInstance must be set to true. **Default value: False**
restorePipelineLimit	Db2 DPF connector performs restoration of all non-catalog partitions in parallel. The parameter `restorePipelineLimit` allows the user to configure the number of partitions that could be restored in parallel by the connector. The valid value for this field is any positive integer. **Default value: 10**

9.4.6.1.2.1 Connector Config file

After adding the environment to Delphix Continuous Data Engine, you can find the connector configuration file, `advanceConfFlag.conf` on the remote environment. It is typically located at the following specified path:

```
<toolkit path>/advanceConfFlag.conf
```

Sample config file

```
# If the user is not sharing a common group between primary environment user and
instance user.
# Then the user needs to set parameter notCommonGroupFlag as true.
# notCommonGroupFlag=true

# During provision operation check if instance contains a database name which is
identical to the source DB name  used for provisioning operation.
# By default plugin will never allow creating a VDB on the instance where source
database (or other database which is identical in name with source DB) already
exists.
# If the user still wants to create a VDB on the instance which contains a database
name which is identical to the source DB name used for provisioning operation,
# then the user needs to set parameter allowSourceDBonSameInst as true.
# allowSourceDBonSameInst=true

# Limit for parallel restores
restorePipelineLimit=10
```

9.4.6.2 IBM Db2 logging

You can find the IBM Db2 connector logs within the scratch path on the remote environment, located at the following path:

```
<toolkit path>/Delphix_COMMON_<long id>/plugin/DB2_18f4ff11-
b758-4bf2-9a37-719a22f5a4b8/logs/<os_user>/diag.log
```

You can configure the logging mechanism by modifying the following parameters defined in the `db2_plugin.conf` file. This feature offers options to provide customized connector logging.

1. **level** - Sets the connector debug log levels.
2. **logFileSize** - sets the maximum connector log file size in MB.
3. **retention** - Sets the maximum number of connector log files.

9.4.6.3 Managing IBM Db2 log rotation and log level

9.4.6.3.1 Setting the log level (level)

Delphix users can set the logging level of the connector-generated logs by setting the parameter *level* under the plugin_logging_parameters header. Higher logging levels will help to expedite debugging issues. There are two levels of logging which are: Info and Debug. The configured logging level will apply to all the objects

(both dSources and VDBs) present on a staging/target host. For example; if we have 4 dSources on a host associated with a Delphix Engine then the log level will apply to all the dSources.

Log level description:

- level=INFO: This level will print only informational logs. This is the default log level.
- level=DEBUG: This level will print informational logs and debug statements.

9.4.6.3.2 Setting maximum size (logFileSize)

Delphix users can set the maximum size of the active log file by defining the value of the `logFileSize` parameter under the plugin_logging_parameters header of the db2_plugin.conf file. Once this maximum size limit is reached, the connector will rotate the log as per the retention property described below. This parameter only takes in a positive integer value. The minimum value of this parameter is 1 MB and the maximum value is 10 MB. The default value is 1 MB.

9.4.6.3.3 Setting the number of log files to retention (retention)

Delphix users can set a retention level for connector-generated logs for each dataset (`<Db name>.diag.log`) using the parameter retention under the plugin_logging_parameters header. As per this parameter, the log files are moved (archived), renamed, or deleted once they reach the value set in the logFileSize parameter. New incoming log data is directed into a new fresh file (at the same location).

By default, the value for this parameter will be set to a minimum value of 2. The user can change this value and set its value within the range 2 and 50. For example; if *retention* is set to 4, the connector will have the following log files: <DB Name>.diag.log, <DB Name>.diag.log.1, <DB Name>.diag.log.2, <DB Name>.diag.log.3, <DB Name>.diag.log.4.

- File <DB Name>.diag.log is the active log file
- File <DB Name>.diag.log.1 is the most recent archive log file
- File <DB Name>.diag.log.4 is the oldest one

For known issues, refer to the Known issues (see page 482) section from the release notes.

9.4.7 IBM Db2 additional information

9.4.7.1 Delphix Virtualization glossary

9.4.7.1.1 Rollforward to Point-in-Time (RFPIT)

Since Delphix virtualization doesn't support VDB point-in-time for DB2, the DB2 connector provides a feature called dSource Rollforward to Point-in-Time (or dSource Timestamp) where the customer can rollforward the dSource to a particular point-in-time so that they can provision a VDB out of the dSource snapshot where the database was rollforwarded to a specific timestamp thereby providing the functionality of Point-in-time VDB provisioning.

9.4.7.1.2 Support for BLU or column-organized tables with DB2 connector

DB2 BLU Acceleration is a feature of IBM's DB2 relational database management system that is designed to deliver high performance for analytical workloads.

It is a high-performance, scalable, and easy-to-use platform for running analytical workloads on large datasets to derive insights from the data quickly and efficiently.

Based on a single registry setting, `DB2_WORKLOAD=ANALYTICS`, DB2 automatically adapts resources, configuration, workload management, and storage parameters to optimize resource consumption for the platform you are running on.

BLU Acceleration reuses the same SQL compiler and optimizer. Most utilities—including LOAD, INGEST, EXPORT, BACKUP and RESTORE, ROLLFORWARD, and many others—work as usual. In addition there is the ability to mix row-organized and column-organized tables in the same storage (tablespace), bufferpool, schema, and even within the same SQL statement.

Prerequisites in Staging and target environments -

- Users must set the DB2 registry variable to `DB2_WORKLOAD=ANALYTICS` before dsource/VDB dataset workflows.
- Users must make sure memory settings are set according to the target server's instance memory.
- Restore operation restores the database configuration file from the backup image.

For more information, read IBM documentation[369].

9.4.7.2 IBM Db2 glossary

9.4.7.2.1 Db2 tablespace Container

In the db2 connector context, DB2 uses containers to physically store data. A container can be a file, a directory, or a raw device. Since containers are objects that actually exist on a disk or a raw device, they are considered as physical database objects. In contrast to containers, tablespaces are logical database objects.

9.4.7.2.2 IBM Db2 authentication

Db2 User and group authentication is managed in a facility external to Db2 LUW, such as the operating system, a domain controller, or a Kerberos security system. This is different from other database management systems (DBMSs), such as Oracle and SQL Server, where user accounts may be defined and authenticated in the database itself, as well as in an external facility such as the operating system.

Any time a user ID and password is explicitly provided to Db2 LUW as part of an instance attachment or database connection request, Db2 attempts to authenticate that user ID and password using this external security facility. If no user ID or password is provided with the request, Db2 implicitly uses the user ID and password that were used to login to the workstation where the request originated. More information on Db2 authentication and authorization is available via IBM documentation[370].

369 https://www.ibm.com/docs/en/db2/11.5?topic=organization-column-organized-tables
370 https://www.ibm.com/developerworks/data/library/techarticle/dm-0508wasserman/

9.4.7.2.3 Delphix Db2 authentication

Delphix for Db2 requires that the staging and target hosts must already have the necessary users and authentication systems created/installed on them. Delphix will neither create users nor change database passwords as part of the provisioning process.

9.4.7.2.4 Db2 database level support

Delphix supports a number of Db2 database-level features. An overview of Db2 database level support is as follows:

- Support for multiple databases linking in a Single Instance, which allows Delphix Engine users to utilize an available instance on the target more efficiently.
- Support for using an end user supplied directory for the Delphix Connector, Db2 Mount Points, and Db2 Delphix files and logs.
- Support for Kerberos environments.
- Support for VDB provisioning of the same OS version level or one version higher than the source and staging instances.
- Intelligent handling of HADR logs.

9.4.7.2.5 High Availability Disaster Recovery (HADR)

The HADR feature of IBM Db2 provides a high availability solution for both partial and complete site failures. It protects against data loss by replicating data changes from a source database, called the primary, to one or more target databases, called the standby.

9.4.7.2.5.1 Delphix HADR support

HADR replication takes place at a database level, not at the instance level. Therefore, a standby instance can have multiple databases from multiple different primary servers/instances on it. If the instance ID on the Delphix standby is not the same as the instance ID on the primary, the Delphix standby instance ID must have database permissions **secadm** and **dbadm** granted to it on the primary database. These permissions and all HADR settings must be implemented on the primary database before you take the backup on the primary database.

9.4.7.2.5.2 Log transmitting

All changes that take place at the primary database server are written into log files. The individual log records within the log files are then transmitted to the secondary database server, where the recorded changes are replayed to the local copy of the database. This procedure ensures that the primary and the secondary database servers are in a synchronized state. Using two dedicated TCP/IP communication ports and a heartbeat, the primary and the standby databases track where they are processing currently, the current state of replication, and whether the standby database is up-to-date with the status of the primary database. When a log record is "closed" (still in memory, but has yet to be written to disk on the primary), it is immediately transmitted to the HADR standby database(s). Transmission of the logs to the standbys may also be time-delayed.

9.4.7.2.5.3 Multiple standby

Beginning in Db2 v10.1, the HADR feature supports multiple standby databases. This enables an advanced topology where you can deploy HADR in multiple standby mode with up to three standby databases for a single primary. One of the databases is designated as the principal HADR standby database, with the others termed as auxiliary HADR standby databases. As with the standard HADR deployment, both types of HADR standbys are synchronized with the HADR primary database through a direct TCP/IP connection. Furthermore, both types support the reads on standby feature and can be configured for time-delayed log replay. It is possible to issue a forced or unforced takeover on any standby, including the delphix auxiliary standby. However, you should never use the Delphix auxiliary standby as a primary, because this will impact Delphix performance.

9.4.7.2.5.4 Delphix HADR synchronization

The Delphix for Db2 uses the HADR capability of Db2 to synchronize data from a production Db2 database into a Delphix-controlled Db2 "standby" server. By using this mature and existing Db2 capability, the Delphix Engine is able to ingest data and keep the standby server in sync with only a minimal impact on production. The HADR connection is configured to Super-Asynchronous (SUPERASYNC) mode where log writes are considered successfully transmitted when the log records are sent from the primary database. Because the primary database does not wait for acknowledgments from the standby database, there is no delay on the primary and transactions are considered committed regardless of the state of the replication of that transaction. For further information on Delphix synchronization, see Data sychronization section for more information.

9.4.7.2.5.5 SSL/TLS support for HADR

Starting from Db2 version 11.1 and onwards, the introduction of the HADR_SSL_LABEL configuration parameter marks a significant advancement in Db2's security capabilities. It grants you the ability to establish SSL encryption for your Db2 databases, significantly bolstering security and data protection, particularly during critical operations like replication and synchronization.

With the inclusion of HADR_SSL_LABEL support in the IBM Db2 connector, you gain a streamlined method to seamlessly integrate SSL security within their Db2 HADR setups. This entails that all data replication and communication between primary and standby databases can now benefit from robust encryption in-transit, effectively securing sensitive information from potential threats.

Utilizing this feature within the IBM Db2 connector ensures that you can maintain the utmost security and encryption for their data while it's in transit between the primary and standby databases, reinforcing the integrity and confidentiality of their Db2 environment.

9.4.7.2.6 Database partitioning feature

Database Partitioning Feature (DPF) lets you partition your database across multiple servers. Since you can add new machines and spread your database across them, this allows users to scale their database. This means more CPUs, more memory, and more disks from each of the additional machines are available for your database. DPF can be used to manage large databases for a variety of use cases including data warehousing, data mining, online analytical processing (OLAP), or online transaction processing (OLTP) workloads.

DPF enables the user to divide a database into database partitions, a database partition is a part of a database that consists of its own data, indexes, configuration files, and transaction logs. Each database partition can be configured on the different physical server having its own set of computing resources, including CPU and storage. When a query is processed, the request is divided so each database partition processes the rows that it is responsible for. DPF can maintain consistent query performance as the table grows by providing the capability to add more processing power in the form of additional database partitions. This capability is often referred to as providing linear scalability using Db2s shared-nothing architecture.

DPF is an approach to sizing and configuring an entire database system. Please follow the recommended practices for optimal performance, reliability, and capacity growth. Please refer to IBM documentation of DPF for more details in IBM knowledge center[371].

9.4.7.2.6.1 Backup types supported in DPF

The type of backups supported by IBM Db2 connector when used with DPF are:

- SSV backups through backup files or named pipes.
- Non-SSV backups through backup files or named pipes.

Example backup command for DB2 SSV backups: db2_all `db2 BACKUP DATABASE<DB_NAME> ON all dbpartitionnums online to <Backup File Location> compress include logs`.

Example backup command for DB2 Non SSV backups: `db2_all db2 backup database <DB_NAME> online to <Backup File Location> compress include logs`.

9.4.7.2.7 Source database with Raw DEVICE type storage

Several users use raw device-based tablespaces for source Db2 databases. To leverage these environments with Delphix, Delphix has built a workflow using Db2's native tools that allow Delphix to discover and convert a raw device-based tablespace into an automatic storage-based tablespace during ingestion. Once the data is ingested into staging, end users will be able to provision VDBs of the automatic storage-based database.

9.4.7.2.8 Support for named pipe type Db2 backup

Users can also provide Db2 backups of type named-pipe. IBM Db2 connector will check the type (type will be either named-pipe or a file) of backup and will use the restore syntax accordingly.

9.4.7.2.9 Purging of archive logs after snapshot operation

Logs can be purged from your log location after the snapshot operation. The Connector will preserve the list of logs that are eligible for purging in the snapshot's metadata. The snapshot metadata can be accessed via the Delphix CLI or API.

371 https://www.ibm.com/developerworks/data/library/techarticle/dm-0608mcinerney/index.html

9.4.7.2.10 Pipelining logic for implementing parallel restores

The connector employs a parallel pipeline methodology so that the restore operation of non-catalog partitions can be performed in parallel in the Database Partitioning Feature (DPF). The number of parallel restores is determined by the value of “restorePipelineLimit” (default value is 10) in *<toolkit path>/Delphix_COMMON_<long id>/plugin/DB2_18f4ff11-b758-4bf2-9a37-719a22f5a4b8/db2_plugin.conf*. For more details on this please refer to Requirements for Db2 Hosts and Databases (see page 1153). The parameter “restorePipelineLimit” is configurable by end-users. The connector performs parallel restore for all non-catalog partitions. For e.g. If the total number of non-catalog partitions is 15, and the "restorePipelineLimit" parameter is set to 10, the first set of 10 restores will happen in parallel. The connector will track the restore of each partition present in the pipe. Whenever a restore of a partition completes, it will move out from the pipe and a new partition will enter into that pipe. Thus, the connector ensures that the pipe will always have the user-configured number of partitions being restored (default=10).

In case of a restore failure for a particular partition, the connector will track that failure and will print the same in the connector-generated logs so that the user can take the necessary action. If the user received failure during restore operation for a few partitions and then decided to manually initiate the restores for failed partitions, the connector will verify the datapaths during snapshot operation. In case it finds wrong path information (such as paths outside the mount point), then the connector will error out of the operation.

The connector will track the status of each restore in a file named <DB Name>_metadata.json The connector can handle a scenario where the staging host reboots in-between the parallel restores. In this case, the user can perform a “Resynchronisation” operation again. At this time, the remaining restores will resume from scratch. For more details on this please refer to Requirements for Db2 Hosts and Databases (see page 1153)

9.4.7.2.11 Data synchronization

Delphix provides the following options to keep your dSource in sync with the source database:

1. During the linking process, you can set up a HADR connection between the original source database and the Standby instance. This allows the Standby instance using HADR for log shipping to always keep its databases in sync with the source. It is important to note that a single Standby instance (dSource) can contain multiple databases from multiple different servers and instances as long as each database has a unique name.
2. Users can opt to configure SSL with the HADR connection to enhance security by encrypting data-in-transit between the primary and the standby databases.
3. The users can opt for the "Staging Push" feature by selecting the checkbox "Use Staging Push". With this feature, the user can manage the restore and roll forward of a staging database without involving the Delphix Engine.
4. Users can opt for a non-HADR method of database ingestion. The database will be ingested to a standalone staging host and the user can trigger a resynchronization operation with a most recent online full backup to keep the staging database up to date with the source database.
5. Users can opt for the Delphix Recovery with the end user Supplied Archive Logs method. During an update, the dSource user can trigger a snapshot operation where user-provided archive logs will get applied to the staging database.

6. Optionally, when the DPF feature is in use, users can use the “Archive Logs” feature. Users need to place archive logs inside a folder with a name as NODE<Partition number> where <Partition number> is a four-digit number. For example, if the source environment has two partitions then the user-provided log path will have folder names NODE0000 and NODE0001. Both the folders will have respective archive logs. Snapshot operation will be used to apply the archive logs.

Delphix HADR standby configuration may not meet Db2 DR requirements and should only be used for Delphix use cases.

9.4.7.2.12 Banners

Banners are any kind of messages that are printed from commands or scripts that are executed on a new session. They are mainly used to print welcome messages or information regarding the host when you start a new session. There are four ways to accomplish this,

1. Print statements in .profile `/etc/motd`.
2. SSH banner configured under `/etc/ssh/sshd_config`.
3. Print statements in `.bashrc`.
4. Adding print statements or commands that return strings in `.bashrc`.
5. can hurt the execution of the connector’s workflows and also basic unix commands such as scp or rsync, etc. Therefore, it is extremely important to avoid using print statements in the `.bashrc` file.

9.4.8 IBM Db2 FAQS

9.4.8.1 What are the best practices for taking a backup?

The following best practices can help improve backup and restore performance:

1. Compression should be enabled
2. The following parameters should be optimally configured:
   - Utility Heap Size (UTIL_HEAP_SZ)
   - No. of CPUs
   - No. of Table Spaces
   - Extent Size
   - Page Size
3. Parallelism & Buffer configuration may be used to improve backup performance. Parameters that should be configured are:
   - Parallelism
   - Buffer Size
   - No. of Buffers

More information about backup best practices is available in IBM Knowledge Center[372].

9.4.8.2 How to take a full online backup of a db2 database?

For the examples below we are using the database name as TEST and location as '/backup_dir'. If no path is provided backups will be generated in the current directory.

Non-DPF backups

```
db2 backup db TEST  online to  /backup_dir compress include logs
```

DPF, SSV backups

```
db2 backup db TEST ON ALL DBPARTITIONNUMS online to /backup_dir compress include logs
```

DPF, NONSSV backups

```
db2_all "db2 backup db TEST online to /backup_dir compress include logs"
```

Multi-part backup

```
b2 backup db TEST  online to  /backup_dir1,/backup_dir2,/backup_dir3 compress include
logs
```

9.4.8.3 What is the user-side validation prior to the snapshot?

User-side validation prior to a snapshot operation:

- The user must validate the backup files using db2ckbkp utility on the staging host.
- The user must validate the archive logs using db2cklog utility on the staging host.
- The authenticity of archive logs would be managed by the user.
- The user has to provide the first active log at the staging database log directory location after applying the logs.
- If there is a scenario where a user performed some load copy (non-logged transactions) operations on the source side then it's the responsibility of the user to make those non-logged transactions available on the staging database.

9.4.8.4 How to use the Delphix Continuous Data Engine key pair?

Perform the following steps to use the Delphix Continuous Data Engine key pair:

372 https://www.ibm.com/support/knowledgecenter/en/SSEPGG_10.5.0/com.ibm.db2.luw.admin.ha.doc/doc/c0006205.html

1. Select **Public Key** for the **Login Type.**
2. Click **View Public Key**.
3. Copy the public key that is displayed, and append it to the end of your `~/.ssh/authorized_keys` file. If this file does not exist, you must create it.
   a. Run `chmod 600 ~/.ssh/authorized_keys` to allow only the file's owner to read and write to it (make sure the file is owned by you).
   b. Run `chmod 755 ~` to restrict access to your home directory so that no other may write to it.
   c. Run `chmod 700 ~/.ssh` so that others cannot write to it. The `~/.ssh` directory is not writable by group or others. Otherwise, authentication will fail.

9.4.8.5 How to update the Db2 version level?

This is applicable only when the target environment is using a different Db2 fix pack level than the source. Consider the following scenarios for the optional prerequisite mentioned above:

1. if your source environment uses Db2 version 11.5.5 and your target environment uses Db2 11.5.8 or a later fix pack, you need to update the databases in the target environment.
2. If your databases were created or upgraded to Db2 Version 11.1 GA and you applied Db2 Version 11.1 Fix Pack 1 or later, then running the above command will automatically apply all updates required from Version 11.1 GA up to and including the fix pack level that you are installing.
3. To know how to update databases, see db2updv115 - Update database to Version 11.5 mod pack command[373]
4. Certain database or instance operations are restricted while an online fix pack update is in progress.
5. See db2updv111 - Update database to Version 11.1 fix pack command[374], which mentions it is mandatory to run db2updv111 on the database if one is coming from a higher version of fix pack to Lower OR vice versa.
6. For more information, you can also refer: Update the databases for a Db2 11.1.x fix pack[375]

If your source environment uses version 11.1.x and your target is to be upgraded to 11.5.x, following are the steps to perform for an in-place upgrade of dsource and VDB.

9.4.8.6 How to match the DB2 common directory when the target host is added in multiple Delphix Continuous Data Engines?

The connector path directory structure is as follows in the target host `<toolkit path>/<Delphix_COMMON>/DB2_18f4ff11-b758-4bf2-9a37-719a22f5a4b8/logs/<instance name>`. However, if there are multiple Delphix Continuous Data Engines using this target, then multiple common directories are created on the target host like this (three engines access the same target and plugin code):

373 https://www.ibm.com/docs/en/db2/11.5?topic=sc-db2updv115-update-database-version-115-fix-pack-command
374 https://www.ibm.com/docs/en/db2/11.1?topic=sc-db2updv111-update-database-version-111-fix-pack-command
375 https://www.ibm.com/docs/hr/opw/8.2.0?topic=db2-update-databases-111x-fix-pack

```
delphix@support-tools:/nas/support/truist/69459$ ls -lart
total 875888
drwxrwxr-x  8 delphix delphix      8192 Oct 20 11:54
 Delphix_COMMON_17b1429568c0_290c7f251296_28_host/
drwxrwxr-x  8 delphix delphix      8192 Oct 20 11:59
 Delphix_COMMON_ce55f7389c2d_7e296fa53e07_24_host/
drwxrwxr-x  8 delphix delphix      8192 Oct 20 12:04
 Delphix_COMMON_cd1f039b057f_63d00dbd3271_22_host/
drwxrwx---  5 delphix delphix       152 Oct 27 11:59
 Delphix_cd1f039b057f_63d00dbd3271_23_host/
```

To identify the common directory corresponding to the Delphix Continuous Data Engine:

- Collect the Engine UUID (You can get "Server ID" from the GUI "About" menu). In this case Engine UUID is:

  ```
  421de915-558f-68a1-fac8-cd1f039b057f
  ```

- Take the last group of numbers:

  ```
  cd1f039b057f
  ```

- Match this group with the FIRST group of numbers following the "COMMON_" characters in the directory name, and in this case it is:

  ```
  Delphix_COMMON_cd1f039b057f_63d00dbd3271_22_host
  ```

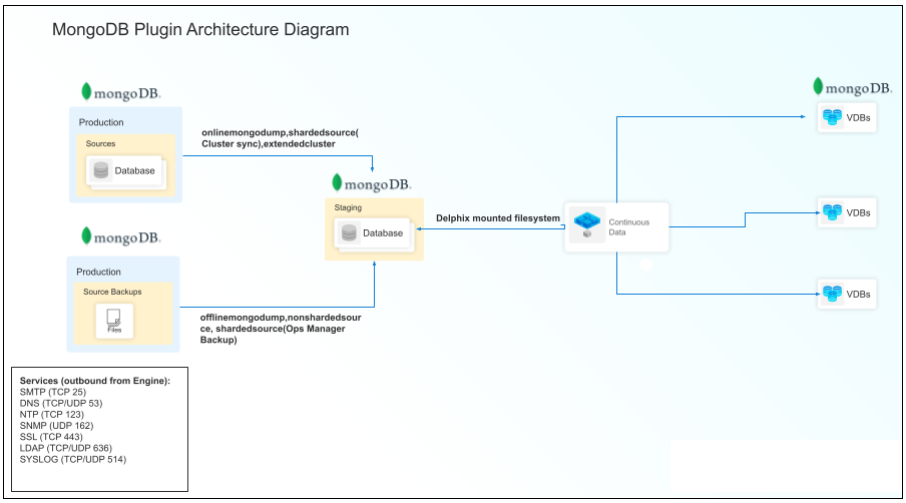
9.5 MongoDB data sources

This section contains the following pages that will help with creating MongoDB data sources (dSources) in Delphix Continuous Data.

- MongoDB virtualization process (see page 1229)
- MongoDB ingestion types (see page 1233)
- MongoDB requirements and prerequisites (see page 1235)
- MongoDB installation and upgrade (see page 1244)
- MongoDB connector operations (see page 1247)
- Provisioning MongoDB virtual databases (VDBs) (see page 1268)
- MongoDB logging (see page 1272)
- MongoDB developer's guide (see page 1272)

9.5.1 MongoDB virtualization process

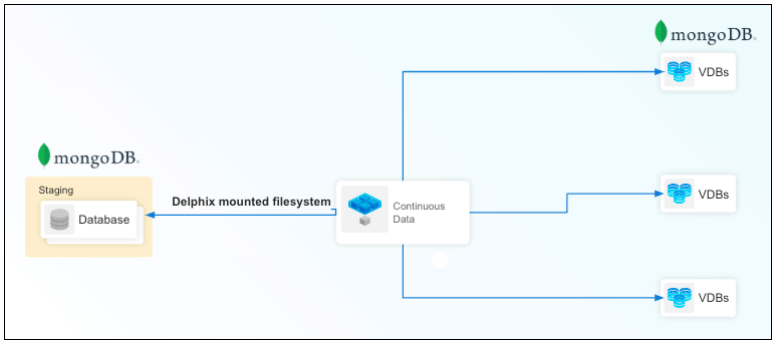
The Delphix MongoDB connector solution uses a staging architecture. In this solution, the source (production) database is copied to a staging environment. The staging environment's database is ingested by the Delphix Continuous Data Engine, which then enables virtualized copies on a target environment.



The MongoDB connector supports the following ingestion mechanisms:

9.5.1.1 seed

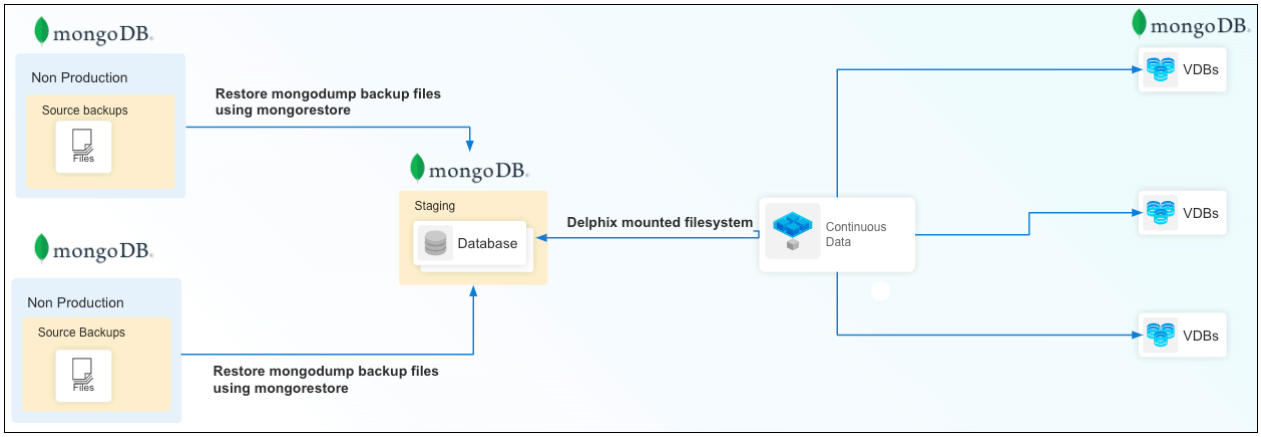
This data ingestion type is used for pure development purposes. There is no source instance associated with it. It creates an empty instance, which is managed by the Delphix continuous Data Engine, and helps to create virtual MongoDB instances to avail benefits of all its features.



9.5.1.2 offlinemongodump

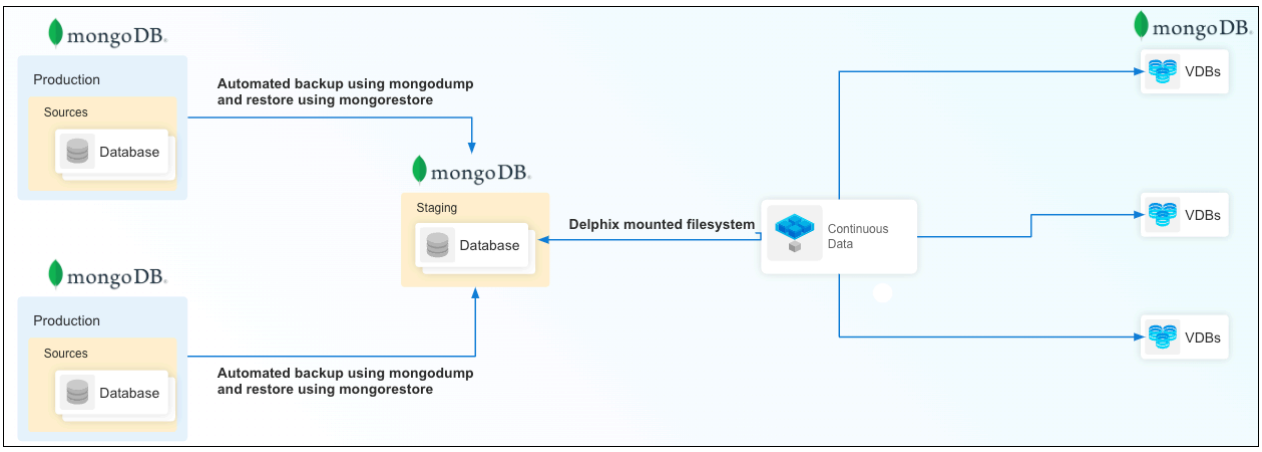
This staging database is created using `mongodump` backup file(s) of the source MongoDB instance. It helps to create a staging database using zero touch production (defined in the MongoDB ingestion types[376] page). You can load periodic backups to create a dSource timeflow.

376 https://cd.delphix.com/docs/latest/mongodb-ingestion-types



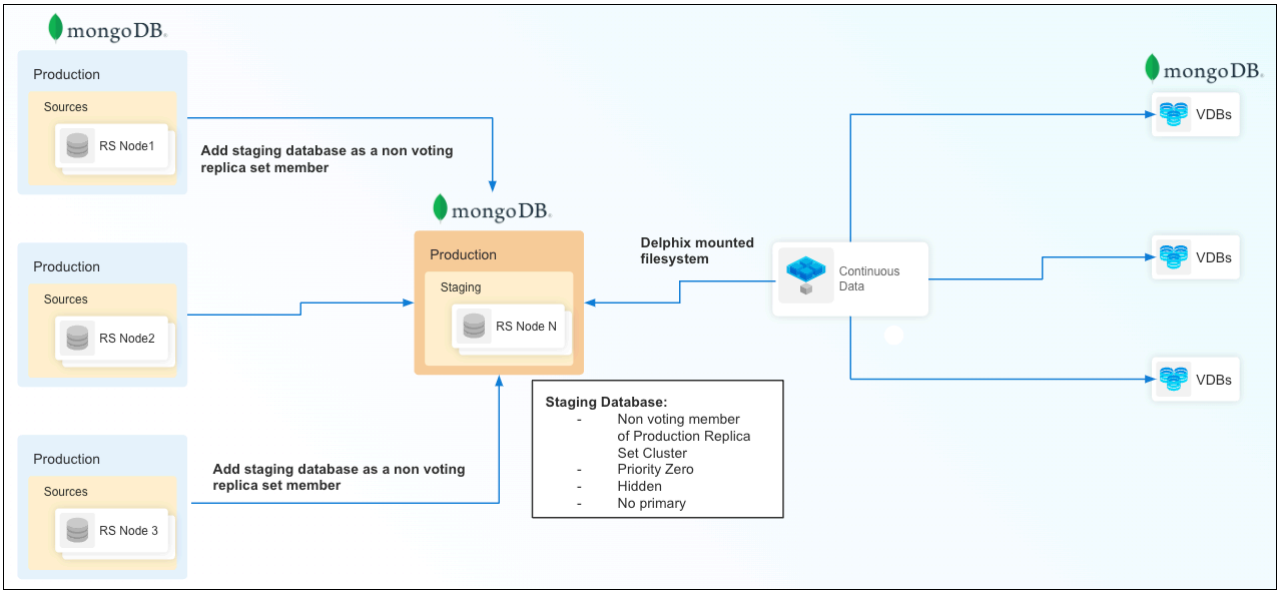
9.5.1.3 onlinemongodump

This staging database is created using a `mongodump` online backup of source MongoDB instance. You can schedule a policy to capture periodic backups and generate a dSource timeflow.



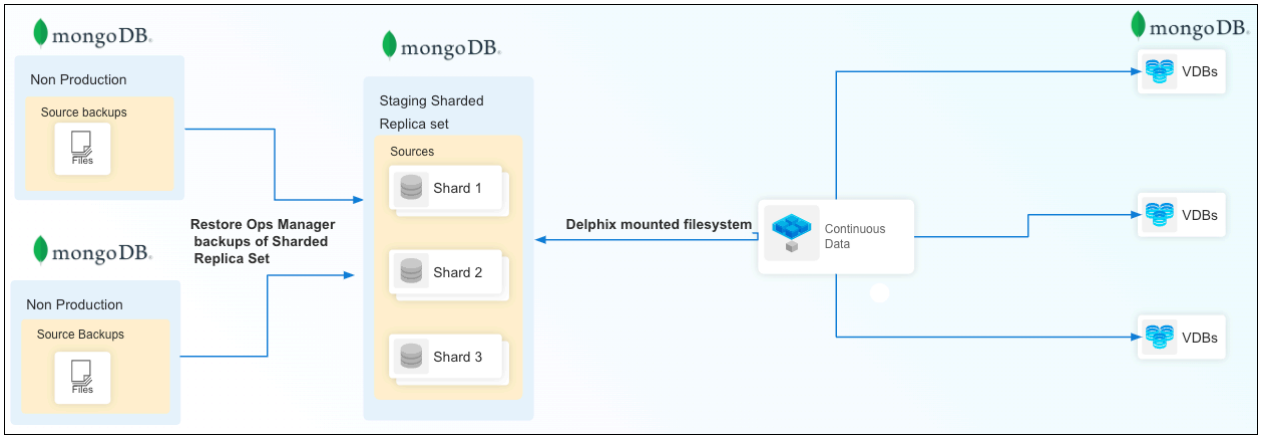
9.5.1.4 extendedcluster

This staging database is created by adding a secondary member to the existing source cluster. This member does not participate in voting and never becomes primary, nor serves any read operations. It's the fastest way of capturing incrementals. You can setup a policy to take snapshots to generate a dSource timeflow.



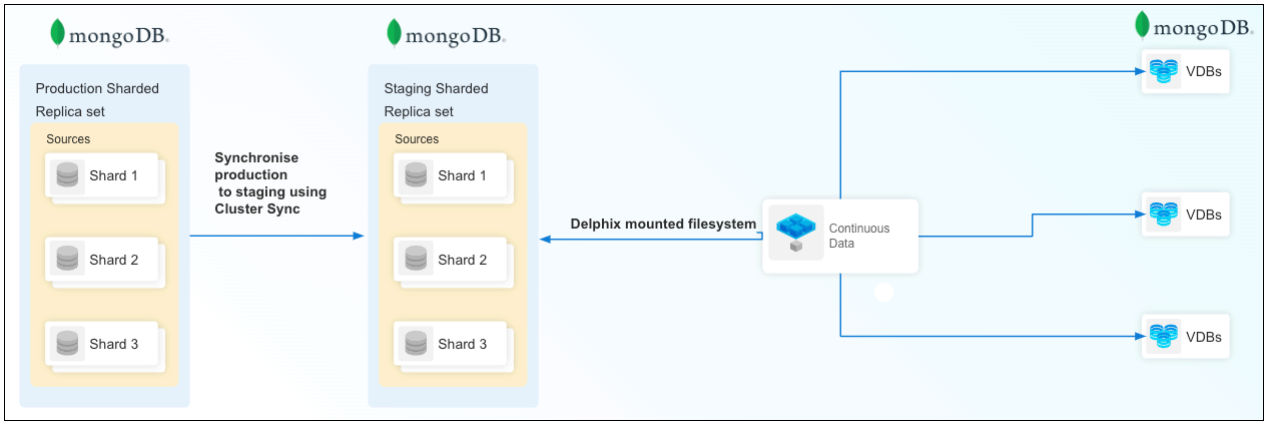
9.5.1.5 shardedsource (using MongoDB Ops Manager backup files)

This staging database is created using backup files of the source MongoDB instances created by MongoDB Ops Manager. It helps to create a staging database using zero-touch production. You can load periodic backups to create a dSource timeflow.



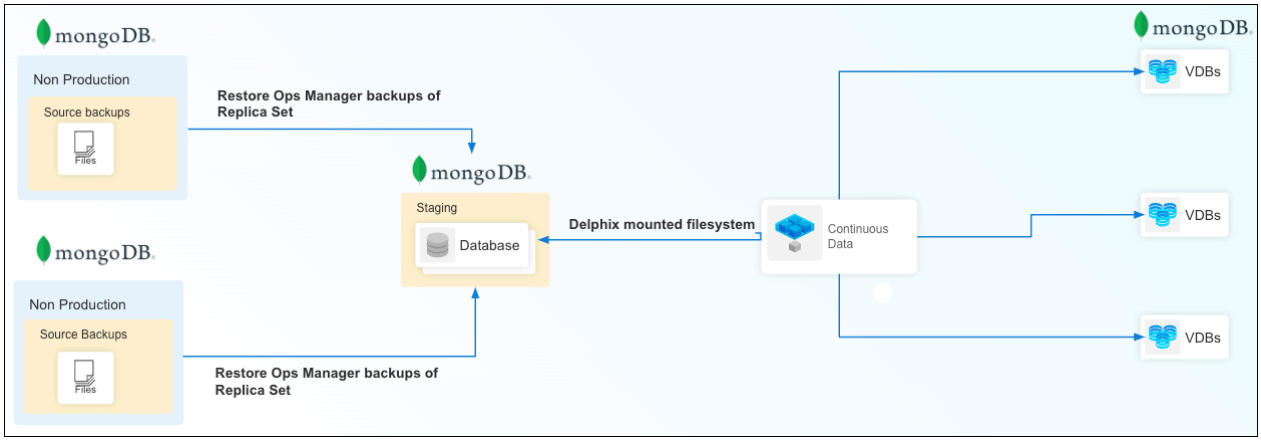
9.5.1.6 shardedsource (using cluster-to-cluster sync)

This staging database is created by setting up a MongoSync synchronization pipeline between the staging database and source sharded cluster. It is the fastest way of capturing incrementals of a sharded cluster. You can set a policy to take snapshots to generate a dSource timeflow.



9.5.1.7 nonshardedsource (using MongoDB Ops Manager backup files)

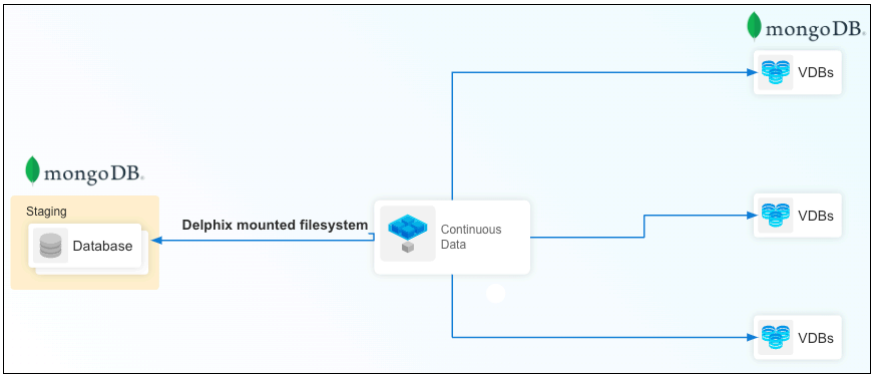
This staging database is created using a backup file of a source MongoDB instance created by MongoDB Ops Manager. It helps to create staging database using zero touch production. You can load periodic backups to create a dSource timeflow.



9.5.1.8 stagingpush

This staging database can be created in multiple ways. It varies based on the use cases. It can be created manually, by copying files at o/s level, or using tools like MongoDB Ops Manager.

> Delphix Continuous Data Engine only exhibits the operations needed to capture data. It is your responsibility to create or delete the MongoDB instance on the staging host.



9.5.2 MongoDB ingestion types

The following table describes the MongoDB ingestion types with additional details, like **Mechanism**, **Topology**, whether or not it's **Zero Touch Production**, and whether or not it **Supports Incrementals**.

- Zero Touch Production (Y) = Interaction with source server is not required by Delphix Continuous Data Engine.
- Zero Touch Production (N) = Interaction with source server is needed by Delphix Continuous Data Engine. Delphix Continuous Data Engine expects connection to source using credentials etc.

dSource Type	Mechanism	Topology	Zero Touch Production	Description	Support Incrementals
`seed`	New ReplicaSet Instance	ReplicaSet	N/A	New empty MongoDB instance created by the Delphix Continuous Data Engine. This is for development of new applications without ingesting data from any source.	N/A
`extendedcluster`	Add member to existing replica set	ReplicaSet	N	Add member to existing source replica set. Instant and near real time incremental data capture in the Delphix Continuous Data Engine.	Y

`onlinemongodump`	Automated `mongodump` backups	ReplicaSet	N	Delphix Continuous Data Engine connects to source from staging environments, executes mongodump utility and captures source data. Delphix Continuous Data Engine captures all databases from the MongoDB instance.	N
`offlinemongodump`	Manual `mongodump` backups	ReplicaSet	Y	Delphix Continuous Data Engine leverages existing mongodumps made available to staging host.-	N
`shardedsource`	MongoDB Ops Manager Backups	Sharded	Y	Delphix Continuous Data Engine leverages existing MongoDB ops manager backups of sharded source as backup files presented to staging host.	N
`shardedsource`	Cluster-to-cluster Sync	Sharded	N	Delphix Continuous Data Engine leverages MongoSync utility to synchronize non-production sharded cluster with existing sharded cluster.	Y
`nonshardedsource`	MongoDB Ops Manager Backups	ReplicaSet	Y	Delphix Continuous Data Engine leverages existing MongoDB ops manager backup of non sharded source. Backup files need to be available on the staging host file system.	N

`stagingpush`	User created MongoDB instance	ReplicaSet	Y/N	Delphix Continuous Data Engine provides a filesystem on the staging host that can be used by you for deploying MongoDB instances. You are responsible for creating working MongoDB instance on staging host either manually or using any tools like MongoDB ops manager after the filesystem creation.	Y/N

9.5.3 MongoDB requirements and prerequisites

This section contains the following pages pertaining to MongoDB requirements and prerequisites in Delphix Continuous Data.

- MongoDB support matrix (see page 1235)
- Requirements for MongoDB source databases (see page 1238)
- Requirements for MongoDB staging/target environments (see page 1239)

9.5.3.1 MongoDB support matrix

Refer to the below support matrix to ensure compatibility between platform, database, connector and Delphix Continuous Data engine version.

9.5.3.1.1 OS and Database compatibility

Supported OS version	**Supported DBMS versions**			
	mongoDB 4.2.x	mongoDB 4.4.x	mongoDB 5.0.x	mongoDB 6.0.x
RHEL 6.x	Y	Y	Y	Y
RHEL 7.x	Y	Y	Y	Y
RHEL 8.x	Y	Y	Y	Y
RHEL 9.x	N	N	N	Y

CentOS 6.x	Y	Y	Y	Y
CentOS 7.x	Y	Y	Y	Y
CentOS 8.x	Y	Y	Y	Y
CentOS 9.x	N	N	N	Y
Windows	N	N	N	N

9.5.3.1.2 Engine and Connector compatibility

Delphix Engine versions	**Supported connector versions**									
	Mongopy 1.0.0	Mongopy 1.0.1	Mongopy 1.0.2	Mongopy 1.1.0	Mongopy 1.2.0	Mongopy 1.2.1	Mongopy 1.3.0	Mongopy 1.3.1	Mongopy 1.3.2	Mongopy 1.4.0
6.0.7.x - 6.0.11.x	Y	Y	N	N	N	N	N	N	N	N
6.0.12.x - 6.0.17.x	Y	Y	Y	Y	N	N	N	N	N	N
7.x - 21.x	N	N	N	N	Y	Y	Y	Y	Y	Y

9.5.3.1.3 Database version and connector version compatibility matrix

Mongo DB version	Mongo py 1.0.0	Mongo py 1.0.1	Mongo py 1.0.2	Mongo py 1.1.0	Mongo py 1.2.0	Mongo py 1.2.1	Mongo py 1.3.0	Mongo py 1.3.1	Mongo py 1.3.2
4.x - 5.x	Y	Y	Y	Y	Y	Y	Y	Y	Y
6.x	N	N	N	N	Y	Y	Y	Y	Y

9.5.3.1.4 Database version and MongoSync compatibility

MongoDB version	MongoSync v1.0.0
4.x - 5.x	N
6.x	Y

9.5.3.1.5 MongoDB shell and connector version compatibility

Mongo DB Shell	Mongo py 1.0.0	Mongo py 1.0.1	Mongo py 1.0.2	Mongo py 1.1.0	Mongo py 1.2.0	Mongo py 1.2.1	Mongo py 1.3.0	Mongo py 1.3.1	Mongo py 1.3.2
mongo	Y	Y	Y	Y	Y	N	N	N	N
mongo sh	N	N	N	N	N	Y	Y	Y	Y

9.5.3.1.6 Atlas supported data ingestion types

MongoDB Atlas supported ingestion types	MongoDB versions

	4.x	5.x	6.x
Seed	N	N	N
OfflineMongoDump	Y	Y	Y
OnlineMongoDump	Y	Y	Y
ShardedSource (Ops Manager backups)	Y	Y	Y
ShardedSource (cluster-to-cluster sync)	Y	Y	Y
ExtendedCluster	N	N	N
NonShardedSource	Y	Y	Y
StagingPush	N	N	N

9.5.3.1.6.1 MongoDB version compatibility across Delphix environments

- MongoDB binaries in all environments (staging and target) must be of the same version for all data ingestion types.
- MongoDB binaries in all environments (source, staging and target) must be of the same version for data ingestion. Type: `extendedcluster`, `nonshardedsource`, `shardedsource` and `stagingpush`.
- MongoDB binaries can be of the same or higher version than source in staging environment for `offlinemongodump` and `onlinemongodump`, adhering to MongoDB vendor support matrix.
- MongoDB binaries in staging and target must be of the same version for data ingestion types `offlinemongodump` and `onlinemongodump`.
- Must use `mongosh` (MongoDB shell) v1.2.1 and onwards.

9.5.3.2 Requirements for MongoDB source databases

When preparing MongoDB as a source database for Delphix integration, specific requirements must be met to ensure functionality and secure data management. This section delineates the necessary configurations and user roles for different types of MongoDB data sources, including extended clusters, online mongodumps, and sharded sources.

Each source type demands unique user privileges and roles, such as `clusterAdmin`, `changeOwnPasswordRole`, and `backup`, to facilitate effective database operations and synchronization.

- **dSource type**: `extendedcluster`
  - **Database requirement**: A source DB user with the **cluserAdmin** and **changeOwnPasswordRole** roles.
    - ```
      use admin
      db.createRole({ role: "changeOwnPasswordRole", privileges:
      [ { resource: { db: "", collection: ""}, actions: [
      "changeOwnPassword" ] } ], roles: [] } ) db.createUser({user:
      "<USERNAME>",pwd: "xxxxxx", roles: ["clusterAdmin","changeOwnPassw
      ordRole"]})
      ```
- **dSource type**: `onlinemongodump`
  - **Database requirement**: A source DB user with the **backup** role.
    - ```
      use admin
      db.createUser({user:"<USERNAME>", pwd:"xxxxxx", roles:[{role:"back
      up", db: "admin"}]})
      ```
- **dSource type**: `shardedsource` (cluster-to-cluster sync)
  - **Database requirement**: A source DB user with the **readAnyDatabase**, **clusterMonitor**, and **backup** roles. Refer to User permissions[377] and Limitations[378] for more.
    - ```
      use admin
      db.createUser( { user: "<USERNAME>", pwd: "xxxxx", roles: [ {role:
      "readAnyDatabase", db: "admin"}, {role: "clusterMonitor", db:
      "admin"}, {role: "backup", db: "admin"} ] } )
      ```
- **dSource type**: Larger datasets
  - **Database requirement**: To limit the MongoDB memory utilization at the staging clusters, set the parameter `wiredTigerCacheSizeGB`, with the appropriate value, in the **Set Parameter** section of the **Add dSource** Page.
    - For memory consumption guidelines, refer to the official MongoDB documentation[379].

9.5.3.3 Requirements for MongoDB staging/target environments

The staging and target environments have the same set of requirements for MongoDB.

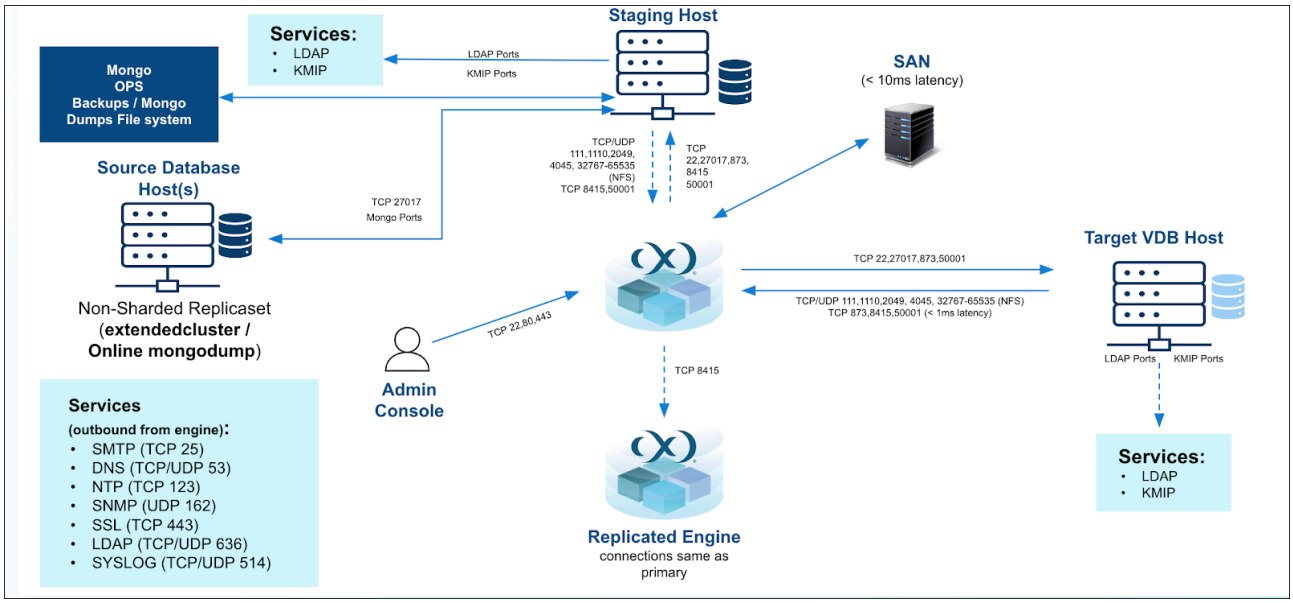
---

377 https://www.mongodb.com/docs/cluster-to-cluster-sync/current/reference/permissions
378 https://www.mongodb.com/docs/cluster-to-cluster-sync/current/reference/limitations/
379 https://www.mongodb.com/docs/manual/reference/configuration-options/#mongodb-setting-storage.wiredTiger.engineConfig.cacheSizeGB
      ```

9.5.3.3.1 Network and connectivity requirements

Before proceeding further, ensure that you meet the general requirements for Delphix Engine network connectivity[380].

For MongoDB plugin specific network and connectivity requirements, refer to the sections below.



9.5.3.3.1.1 SSH connectivity to the target host

1. Target environments are required to have `sshd` running and configured for the Delphix Continuous Data Engine to connect over SSH.
2. The Delphix Continuous Data Engine platform expects to maintain long-running, highly performant SSH connections with the host environment.
3. Once connected to a target environment through SSH, the Delphix Continuous Data Engine initiates a MongoDB client connection (`mongosh`) from the target environment to the source environment.

9.5.3.3.1.2 Creating NFS mounts on target host

Delphix Continuous Data Engine shares its storage space with the target host using NFS mounts.

Protocol	Port Number	Use
TCP/UDP	111	Remote Procedure Call (RPC) port mapper used for NFSv3 mounts.

380 https://cd.delphix.com/docs/latest/network-connectivity-requirements

Protocol	Port Number	Use
TCP	1110	Network Status Monitor (NSM) client from target hosts to the Continuous Data Engine.
TCP	2049	NFS client from target hosts to the Continuous Data Engine (NFSv3 and NFSv4).
TCP	54043	Client mount daemon (NFSv3 only).
TCP	54044	Lock state notification service (NFSv3 only).
TCP	54045	Network Lock Manager (NLM) client from target hosts to Continuous Data Engine (NFSv3 only).
UDP	33434 - 33464	`Traceroute` from the target database server to the Continuous Data Engine (optional).

9.5.3.3.1.3 Connecting to MongoDB instances (source, staging database, VDBs)

Outgoing [From]	Incoming [To]	Protocol	Port Number	Use
Staging host	Source MongoDB instance	TCP	<source-port>	MongoDB source database instance port to connect to, for taking backup as well as setting up streaming replication. The default port for a MongoDB instance is 27017.
End-user hosts	Target MongoDB instance (staging database instance or VDBs).	TCP	<target-port>	Connecting to the created database instance on the target host.

9.5.3.3.1.4 Host environment requirements

Staging/target host requirements	**Explanation**
Host compatibility check • The operating system and architecture of the target host must match the source environment.	It is required that the source and the target environments should have identical operating system and hardware configurations.
MongoDB installation and compatibility check • There must be an installation of MongoDB on the staging/target host that is compatible with the installation of MongoDB on the source host.	The MongoDB binary version on the staging/target must be identical to the source side.
There must be a directory on the staging/target host where the MongoDB connector support files can be installed (i.e. /var/tmp) with the following properties: 1. The directory must be writable by the operating system user mentioned above. 2. The directory must have at least 1.5 GB of available storage.	This path is called the **Toolkit Path** on the **Environment creation** UI page.
During VDB provisioning, there must be a mount point directory (i.e. /mnt) that will be used as the base for mount points – created with the following properties: 1. The mount point directory must be writable by the operating system user mentioned above. 2. The mount point directory should be empty.	N/A

Hostname and IP address must be correctly set in the /etc/hosts file.	The output of `hostname -i` should produce the correct result as the IP address of the server:
	`[root@source ~]# hostname -i`
	`10.110.211.225`

Required utilities and tools • Ensure that either the `netstat` or `ss` utility is installed on the staging/target host.	MongoDB connector checks the status of the running MongoDB instance and validates if ports are available on the staging/target host using the `netstat` or `ss` utility. Check if the `netstat` or `ss` utility is installed with: • `which netstat` • `which ss`
Storage requirement	1. Staging host must have enough storage space to hold the source backup file. 2. Empty folder on staging host to hold Delphix toolkit (approximately 2GB free space).

9.5.3.3.1.5 Sudo privileges for the environment OS user

Delphix Continuous Data Engine connects to the target host (via SSH) to perform the following operations:

1. Delphix Continuous Data Engine dynamically mounts and unmounts directories for linking and provisioning operations. This privilege is required because mount and unmount are typically reserved for superuser.
2. Delphix Continuous Data Engine dynamically creates and removes directories for linking and provisioning operations.

The operations (1) and (2) will require sudo privileges for the OS user.

Configuring sudo access on target environments (as root user)

1. The sudo configuration exists in the **/etc/sudoers** file. To to add the privilege, you must edit this file either directly or by using a utility called `visudo`.
   a. `vi /etc/sudoers` or `visudo`
2. Run the command below to add the privileges.

```
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD: /bin/mount,/bin/umount,/bin/mkdir,/bin/rmdir, /bin/ps
```

> The above examples are for illustrative purposes and the sudo file configuration options are subject to change.

Requiretty settings

Delphix Continuous Data Engine requires that the requiretty setting must be disabled for all Delphix Continuous Data Engine users with sudo privileges.

It is required to specify the NOPASSWD qualifier within the sudo configuration file, as shown in the above command. This ensures that the `sudo` command does not demand the entry of a password, even for the "display permissions" (i.e. `sudo -l`) command.

- Additionally, Delphix OS users should meet the following criteria:
  - Regular OS user with primary group as `mongod`.
  - Execute access on `mongosh` / `mongod` binaries.
  - Must have access to backup files of the source database stored on the staging host.
  - Must have execute access on all files within the MongoDB installation folder. The minimum permission level 775 is recommended.
- `mongosh` and `mongod` binaries must be in the same folder (if required, create softlink).
- Ensure any login banners similar to the one in the image below are disabled on the staging host.

```
[staging_host]$ su - delphix
*******************************************************
*
*              Welcome to the Linux Terminal
*
*-------------------------------------------------------
*
*  You are now logged in as: delphix
*
*  Take caution with your actions and be responsible!
*
*******************************************************
Password:
```

9.5.4 MongoDB installation and upgrade

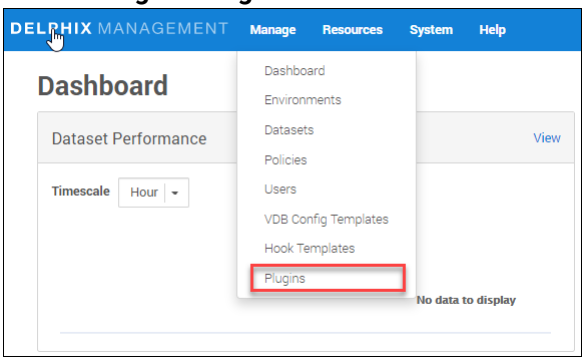
9.5.4.1 Downloading the connector

The connector is provided as a zip file on the Delphix download site. Perform these steps to download the connector:

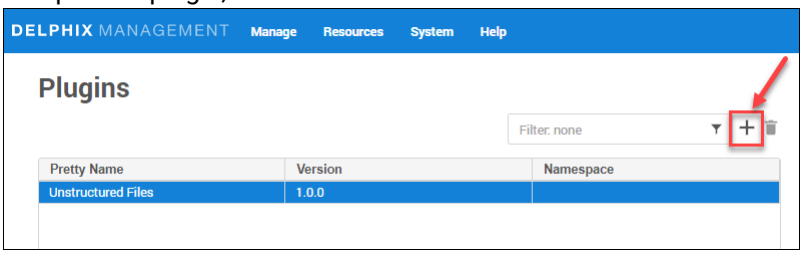
1. Launch a web browser and navigate to the Delphix download[381] portal.
2. Login to the portal using customer support credentials.
3. Navigate to **Delphix Product Releases** -> **Select Connectors** -> **MongoDB_Linux** → **v0.X.X (applicable version number)** → click the **sc_mongo_0.X.X.json** file available to initiate the download.

9.5.4.2 Installing the connector

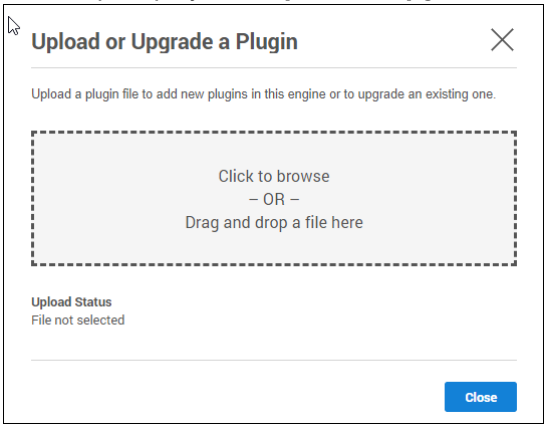
1. Login to the **Delphix Management** application.
2. Click **Manage** -> **Plugins**.



3. To upload a plugin, click the **+** icon.



4. This step displays the **Upload** or **Upgrade a Plugin** dialog. Upload the **sc_mongo_0.X.X.json** file.



For more information, refer to the Continuous Data Engine connector management[382] page.

381 https://download.delphix.com/
382 https://cd.delphix.com/docs/latest/delphix-engine-plugin-management

9.5.4.3 Upgrading the connector

Uploading a higher version connector is the same as installing the MongoDB connector.

Perform the following steps to upgrade a connector version:

1. Upload the higher version connector file at step 4 above.
2. Refresh all corresponding environments.
3. Ensure connector discovery is completed successfully.
4. Take a new snapshot of the staging database and VDB.

> Datasets do not require a disable and re-enabled as part of the upgrade process. Only refresh of environments is required.

To verify the new version of the connector, view the snapshot metadata on the new snapshot.

To upgrade to newer versions of the MongoDB connector, you must follow the upgrade path as given below.

Currently Installed Version	Recommended upgrade path
1.0.x, 1.1.x, 1.2.x, 1.3.x	Upgrade to 1.4.0 using the standard upgrade directions

9.5.4.3.1 Things to consider during upgrade

- `$MONGO_REPO_CFGFILE` file update with connector upgrade
  - After uploading the 1.1.0 MongoDB connector version, specify the MongoDB database tools path in `$MONGO_REPO_CFGFILE`.
  - After uploading the 1.2.0 or later MongoDB connector version, specify all relevant paths for `mongosh` / `mongo`, `mongod`, `mongorestore`, `mongodump`, and `mongosync` in `$MONGO_REPO_CFGFILE`.
  - After uploading the 1.2.1 or later MongoDB connector version, replace the Legacy `mongo` shell with the `mongosh` shell on the staging and target environments, to make benefit of more shell features[383] like MongoDB node.js driver support.
  - Once complete, refresh the environment. Steps can be found in the **Refresh an environment** section in the Environment operations[384] page before executing any other operation.

383 https://www.mongodb.com/docs/mongodb-shell/#the-mdb-shell-versus-the-legacy-mongo-shell
384 https://cd.delphix.com/docs/latest/environment-operations

9.5.4.3.2 Upgrade to MongoDB connector version v1.2.1 or later

If the dSource type is onlinemongodump.

1. Make sure that the **Source MongoDB Host Connect String** contains replica set to PRIMARY as a first node in the connection string.
   a. For example, if **Source MongoDB Host Connect String** is set to `mongohost1.com:28501,mongohost2.com:28502,mongohost3.com:28503`, the `mongohost1.com:28501` must refers to the PRIMARY node of replica set.
2. If the source database is SSL/TLS enabled, modify the **Source MongoDB Host Connect String** using the following rules:
   a. `--ssl` should be replaced with `--tls`
   b. `--sslPEMKeyFile` should be replaced with `--tlsCertificateKeyFile`
   c. `--sslCAFile` should be replaced with `--tlsCAFile`

9.5.5 MongoDB connector operations

To successfully ingest and virtualize a MongoDB instance, you must perform the following connector operations in sequence:

1. Discovering installations[385]
2. Linking data sources[386]
3. Provisioning virtual databases[387]

The respective sections provide detailed information about prerequisites, limitations (if any), and the procedure to be followed for completing each of the operations.

9.5.5.1 Discovering MongoDB installations

The environment discovery and refresh operation enables the MongoDB connector to determine the MongoDB installation details on a host. It is initiated during the environment set up process.

Whenever there is any change (i.e. installing a new database home) to an already set up environment in the Delphix Continuous Data Engine application, you must perform an environment refresh.

9.5.5.1.1 Prerequisite for the discovery operation

When you add an environment to the Delphix Continuous Data Engine, MongoDB connector does an automatic discovery of all MongoDB installed binaries on the server.

385 https://cd.delphix.com/docs/latest/discovering-mongodb-installations
386 https://cd.delphix.com/docs/latest/linking-mongodb-data-sources
387 https://cd.delphix.com/docs/latest/provisioning-mongodb-virtual-databases-vdbs

9.5.5.1.2 Adding an installation

If a new MongoDB installation is added to the environment, you must refresh the environment to view it in the Delphix Continuous Data Engine UI.

9.5.5.2 Adding a MongoDB environment

This page describes the steps for adding a MongoDB staging and target environment.

9.5.5.2.1 Prerequisites

- A MongoDB staging environment must be setup with the necessary requirements as described in MongoDB requirements and prerequisites[388].
- Install MongoDB binaries on staging and target servers.
- Installation of the MongoDB connector is required before the discovery operation.
- The `$MONGO_PATH` variable must be set on the staging and target host using `$HOME/.setDelphixMongoEnv` and `MONGO_REPO_CFGFILE`, which contains the binary path of MongoDB using the below procedure.

9.5.5.2.2 Procedure

1. Login to the staging host as Delphix Continuous Data Engine OS user.
2. Create a file named **.setDelphixMongoEnv** in the home directory of Delphix Continuous Data Engine user.

```
touch $HOME/.setDelphixMongoEnv
```

3. Define the variable pointing to the config file for manual discovery of MongoDB instances. For example:

```
echo "MONGO_REPO_CFGFILE=<Delphix OS User Home directory>/.mongorepos.txt" >
$HOME/.setDelphixMongoEnv
```

4. Define the full path to the MongoDB binary in file represented by `$MONGO_REPO_CFGFILE`. For example:cv

388 https://cd.delphix.com/docs/latest/mongodb-requirements-and-prerequisites

```
source $HOME/.setDelphixMongoEnv
echo "MONGO_PATH=<mongodb installation home directory>/bin/mongod" >
$MONGO_REPO_CFGFILE
```

5. If there are multiple version binaries, specify each in separate lines as shown below:

```
source $HOME/.setDelphixMongoEnv
echo "MONGO_PATH=<mongodb installation home directory>/bin/mongod" >
$MONGO_REPO_CFGFILE
echo "MONGO_PATH=<mongodb installation home directory>/bin/mongod" >>
$MONGO_REPO_CFGFILE
```

6. If the MongoDB database tools are installed at different location, specify the MongoDB tools path in separate lines as shown below:

```
source $HOME/.setDelphixMongoEnv
echo "MONGO_PATH=<mongodb installation home directory>/bin/mongod:<mongodb-database-
tools directory>/bin" > $MONGO_REPO_CFGFILE
echo "MONGO_PATH=<mongodb installation home directory>/mongod:<mongodb-database-tools
directory>/bin" >> $MONGO_REPO_CFGFILE
```

7. In a similar manner, more directories (such as directory containing MongoSync or `mongosh`) corresponding to a repository can be added as shown below:

```
source $HOME/.setDelphixMongoEnv
echo "MONGO_PATH=/u01/mongodb603/bin/mongod:/u01/mongodb603/mongodb-database-tools/
bin:/u01/mongodb603/mongosync/bin:/u01/mongodb603/mongo-shell/bin" >
$MONGO_REPO_CFGFILE
```

8. Login to the **Delphix Management** application.
9. Click **Manage** -> **Environments**.
10. Click the **Actions ...** menu and select **Add Environment**.
11. From the Add Environment dialog window, select **Unix/Linux**.
12. Select **Standalone Host**
13. Click **Next**.
14. Provide the name of the environment.
15. Provide the **Host IP address** or **host FQDN**.
16. Provide the **SSH port**. The default value is 22.
17. Choose the **Login Type** that fits your setup.

- Username and Password – provide the environment username and password
- Username and Public Key – provide the environment username
- Password Vault – select from an existing Enterprise Password Vault

18. Provide an **OS Username** for the environment.
19. If Applicable, provide an **OS Password** for the environment..
20. For Password Login, click **Validate** to test the OS Username and OS Password.
21. Enter **Toolkit Path** (ensure that the toolkit path does not have any spaces).
22. To provide your own Oracle Java Development Kit, select the **Provide my own JDK** checkbox and click **Next**.
    a. On the **Java Development Kit** page, provide an absolute path to your Oracle JDK and click **Next**.
23. Click **Submit**

9.5.5.2.2.1 Points to Remember while creating `$MONGO_REPO_CFGFILE`

1. Each line in the file represents a unique version of the installation. If there are multiple installations in an environment, the file will contain multiple lines.
2. The mandatory binaries for a successful discovery are `mongod` and `mongosh`.
3. Discovery of MongoDump, MongoRestore, and MongoSync are optional based on the dSource type.
4. Multiple paths for each repository can be separated (first priority given to first path provided) for discovering all the desired binaries listed above.
5. The path provided should contain the binary in the immediate directory. For example, for discovering `/u01/mongo603/database-tools/bin/mongorestore`, the path provided should be `/u01/mongo603/database-tools/bin` instead of `/u01/mongo603/database-tools`.

9.5.5.3 Adding a MongoDB source config

An environment contains repositories that are MongoDB installations in the environment. Each environment can have any number of repositories associated with it.

A SourceConfig object defines the configuration of the dSource and is required to create a dSource. You can create any number of SourceConfig objects using a repository, that represents known database instances.

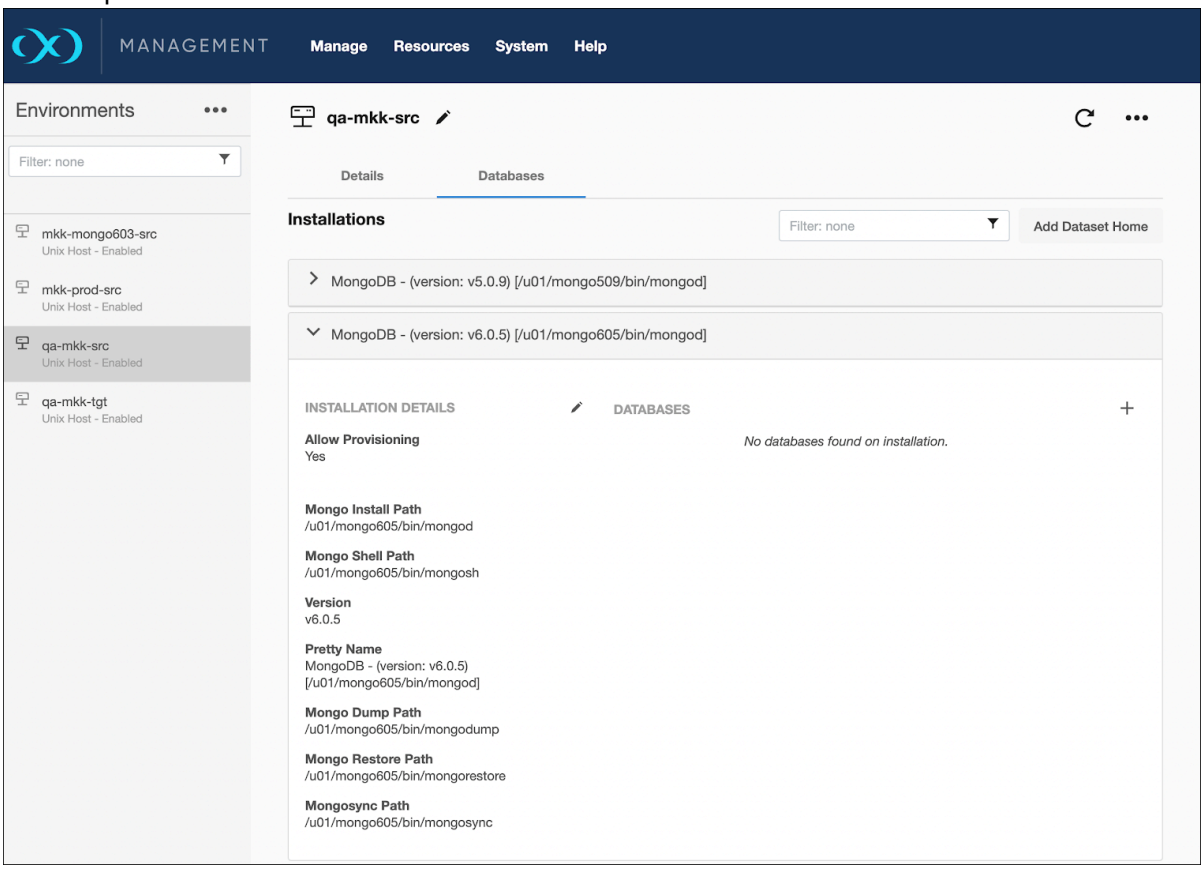
For the MongoDB connector, the Source config must be created manually.

9.5.5.3.1 Procedure

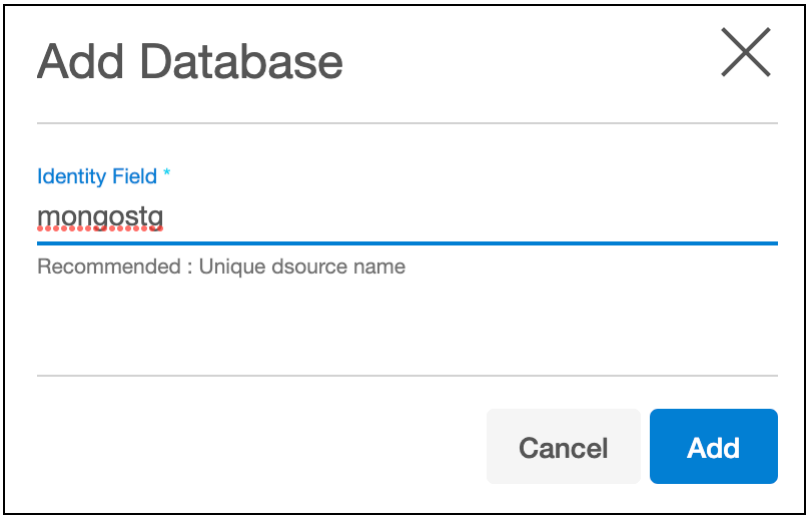
Perform the following steps to add a database.

1. Login to the **Delphix Management** application.

2. Click **Manage** -> **Environments**.
3. Select the repository.
4. Click on plus **+** icon.



5. Add the required details in the **Add database** dialog window.
6. In the **Identity field**, enter a unique name for the staging database.
7. Click **Add**.



9.5.5.4 Linking MongoDB data sources

When linking a dSource from a MongoDB source database, Delphix Continuous Data Engine offers the following ingestion methods.

- seed (see page 1252)
- offlinemongodump (see page 1253)
- onlinemongodump (see page 1256)
- extendedcluster (see page 1258)
- shardedsource (using MongoDB Ops Manager backup files) (see page 1259)
- shardedsource (using cluster-to-cluster sync) (see page 1262)
- nonshardedsource (using MongoDB Ops Manager backup files) (see page 1263)
- stagingpush (see page 1266)

9.5.5.4.1 seed

Perform the following steps to create a dSource with the seed ingestion type.

9.5.5.4.1.1 Prerequisites

The source and staging instances must meet the host requirements as described in Requirements for MongoDB staging/target hosts[389].

9.5.5.4.1.2 Creating a dSource

1. Login to **Delphix Management** application.
2. Click **Manage** -> **Environments.**
3. Click on **Add dSource**.
4. In the **Add dSource** wizard, select the mongo source config created on the staging host.
5. Select the dSource type from the drop-down list in the dSource dialog window.
6. Enter the Mongo-specific parameters for your dSource configuration.
   a. **dSource Type:** This is the dSource type.
   b. **dSource (Sharded/Replica set) details:**
      i. **Staging Replica Set Name:** This is the staging database replica set name. The default value is `dlpx_rs0`.
      ii. **Staging MongoDB Host:** This is the staging host for creating a dSource.
      iii. **Mount Path:** This is the mount path for dSource MongoDB instance data files.
      iv. **bindIP String:** The default value is `0.0.0.0`.
      v. **Staging Port:** This is the staging host port for MongoDB dSource instance.
   c. **dSource tuning details:**

389 https://cd.delphix.com/docs/latest/requirements-for-mongodb-staging-target-environmen

    i. **Oplog Size:** This is the size of `oplogs`.
    ii. **Journal Commit Interval:** This is the journal commit interval, measured in ms.
    iii. **Storage Engine:** The default value is `wiredTiger`.
    iv. **Enable SetParams:** Enable the specified `setParameters`.
    v. **Set Parameter:** These are the parameters that need to be added in the mongod.conf.
    vi. **Enable Auditlog:** Enable the specified audit log parameters.
    vii. **Auditlog Parameters:** These are the parameters that need to be added in the mongod.conf file.
  d. **Enable Authentication:** Enable member and client authentication for the database.
  e. **dSource authentication details:**
    i. **Mongo Admin User:** This is the admin user of the staging database. This user should be created on the fly.
    ii. **MongoDB Password:** This is the admin user password of the staging database.
    iii. **Cluster Auth Mode:** The accepted values are: `keyFile`, `x509`, `sendKeyFile`, `sendX509`.
    iv. **KeyAuth keyfile name:** This is the absolute path of `keyFile`.
  f. **Enable SSL/TLS:** Enable SSL/TLS member authentication.
    i. **SSL/TLS Parameters:** Specify at least the following parameters; `tlsMode`, `tlsCAFile`, `tlsCertificateKeyFile`.
  g. **Enable data encryption:** Enable KMIP or keyFile based encryption.
  h. **Data encryption details:**
    i. **Encryption Method:** KeyFile or KMIP.
    ii. **Local Encryption KeyFile:** This is the `keyFile` path for local encryption.
    iii. **KMIP Parameters:** Provide the KMIP parameter details. For example, `kmipClientCertificateFile`, `kmipPort`, `kmipServerCAFile`, `kmipServerName`.
  i. **Client Certificate for Mongos / Mongo session on Primary Node:** This is the absolute path of client certificate to connect to staging database.
  j. **Client CA Certificate for Mongos / Mongo session on Primary Node:** This is the absolute path of root CA certificate that signed the client certificate.

7. Click **Submit** to create a dSource.

> You must specify the inputs *KeyAuth keyfile name* and *Cluster Auth Mode* when **Enable Authentication** is selected.

9.5.5.4.2 offlinemongodump

Perform the following steps to create a dSource with the offlinemongodump ingestion type.

9.5.5.4.2.1 Prerequisites

- The source and staging instances must meet the host requirements as described in Requirements for MongoDB staging/target hosts[390].
- For creating a dSource type of offlinemongodump, **Backup Metadata File Name** parameter is mandatory. The content of this file should be a single line `datetime` value, specified in format of `MMDDYYYY_HH24MISS`. The content of this file allows the Delphix Continuous Data Engine to validate if there is any new backup uploaded on the staging host that is different from the previous snapshot.
    - Delphix Continuous Data Engine checks this file based on a SnapSync policy, to check whether the new backup is available for ingestion or not. It performs no other checks on backups. If date is updated, Delphix Continuous Data Engine assumes that there are new backup files available in the backup location.
    - Delphix Continuous Data Engine will reingest backup and take a new snapshot. This file must be updated manually or as a part of the backup/copy script. The following command can be added to the backup script that will be executed as soon as the backup is successful or backup copy is successful.

```
echo "`date '+%m%d%Y_%H%M%S'`" > /<PATH>/mongo_bkp_metadata.txt
```

The file name and path must adhere to the organization standard naming convention.

- It supports backups taken using the `mongodump --oplog --gzip -o < bkp_dir >` or `mongodump --gzip -o < bkp_dir >` command. Database dumps for one or more databases or one or more collections are supported.
- For MongoDB Atlas, a backup can be taken using the `mongodump` command line mentioned above.

9.5.5.4.2.2 Creating a dSource

1. Login to **Delphix Management** application.
2. Click **Manage** -> **Environments**.
3. Click on **Add dSource**.
4. In the **Add dSource** dialog window, select the MongoDB source config created on the staging host.
5. Select the dSource type from the drop-down list in the dSource dialog window.
6. Enter the Mongo-specific parameters for your dSource configuration.
    a. **dSource Type:** This is the Type of dSource.
    b. **dSource (Sharded/Replica Set) details:**

390 https://cd.delphix.com/docs/latest/requirements-for-mongodb-staging-target-environmen

   i. **Staging Replicaset Name:** This is the staging Database replica set name. The default value is `dlpx_rs0`.
   ii. **Staging MongoDB Host:** This is the staging host for creating a dSource.
   iii. **Mount Path:** This is the mount Path for dSource MongoDB instance data files.
   iv. **bindIP String:** The default value is `0.0.0.0`.
   v. **Staging Port:** This is the staging host port for MongoDB dSource instance.

c. **dSource tuning details:**
   i. **Oplog Size:** This is the size of `oplogs`.
   ii. **Journal Commit Interval:** This is the journal commit interval, measured in ms.
   iii. **Storage Engine:** The default value is `wiredTiger`.
   iv. **Enable SetParams:** Enable the specified `setParameters`.
   v. **Set Parameter:** These are the parameters that must be added in the mongod.conf.
   vi. **Enable Auditlog:** Enable the specified audit log parameters.
   vii. **Auditlog Parameters:** These are the parameters that must be added in the mongod.conf file.

d. **Mongodump Backup/Tuning details**:
   i. **Backup Metadata File Name:** This is the file that contains backup `datetime` in the `MMDDYYYY_HH24MISS` format.
   ii. **Backup directory for Mongodump**: This is the backup directory for Mongodump.
   iii. **Mongorestore Tuning Parameters:** These are the pairs of parameter name and value for `mongorestore` command line.

e. **Enable Authentication:** Enable member and client authentication for the database.

f. **dSource authentication details:**
   i. **Mongo Admin User:** This is the admin user of the staging database.This user must be created on the fly.
   ii. **MongoDB Password:** This is the admin user password of the staging database.
   iii. **Cluster Auth Mode:** The accepted values are; `None`, `keyFile`, `x509`, `sendKeyFile`, `sendX509`.
   iv. **KeyAuth keyFile name:** This is the absolute path of keyfile.

g. **Enable SSL/TLS:** Enable SSL/TLS member authentication.
   i. **SSL/TLS Parameters:** Specify at least the following parameters; `tlsMode`, `tlsCAFile`, `tlsCertificateKeyFile`.

h. **Enable data encryption:** Enable KMIP or keyfile based encryption.

i. **Data encryption details:**
   i. **Encryption Method:** KeyFile or KMIP.
   ii. **Local Encryption Keyfile:** This is the keyfile path for local encryption.
   iii. **KMIP Parameters:** This is the KMIP parameter details. For example, `kmipClientCertificateFile`, `kmipPort`, `kmipServerCAFile`, `kmipServerName`.

j. **Client Certificate for Mongos/Mongo session on Primary Node:** This is the absolute path of client certificate to connect to staging database.

k. **Client CA Certificate for Mongos/Mongo session on Primary Node:** This is the absolute path of the root CA certificate that signed the client Certificate.

7. Click **Submit** to create a dSource.

> You must specify the inputs `KeyAuth keyFile name` and `Cluster Auth Mode` when **Enable Authentication** is selected.

9.5.5.4.3 onlinemongodump

Perform the following steps to create a dSource with the onlinemongodump ingestion type.

9.5.5.4.3.1 Prerequisites

- The source and staging instances must meet the host requirements as described in Requirements for MongoDB staging/target hosts[391].
- For MongoDB Atlas, you must obtain the MongoDB 3.4 or earlier connection string of Mongo Shell from the Atlas UI to use it as source MongoDB host connect string.

9.5.5.4.3.2 Creating a dSource

1. Login to **Delphix Management** application.
2. Click **Manage** -> **Environments**.
3. Click on **Add dSource**.
4. In the **Add dSource** wizard, select the MongoDB source config created on the staging host.
5. Select the dSource type from the drop-down list in the dSource dialog window.
6. Enter the Mongo-specific parameters for your dSource configuration.
   a. **dSource Type:** This is the dSource type.
   b. **Source database details:**
      i. **Source MongoDB Host Connect String:** These are the details of the source database.
      ii. **MongoDB Source Database User:** This is the source database username.
      iii. **MongoDB Password:** This is the source database password.
   c. **dSource (Sharded/Replica Set) details:**
      i. **Staging Replicaset Name:** This is the staging database replica set name. The default value is `dlpx_rs0`.
      ii. **Staging MongoDB Host:** This is the staging host for creating a dSource.
      iii. **Mount Path:** This is the mount path for dSource MongoDB instance data files.
      iv. **bindIP String:** The default value is `0.0.0.0`.

391 https://cd.delphix.com/docs/latest/requirements-for-mongodb-staging-target-environmen

   v. **Staging Port:** This is the staging host port for MongoDB dSource instance.
d. **dSource tuning details:**
   i. **Oplog Size:** This is the size of `oplogs`.
   ii. **Journal Commit Interval:** This is the journal commit interval, measured in ms.
   iii. **Storage Engine:** The default value is `wiredTiger`.
   iv. **Enable SetParams:** Enable the specified `setParameters`.
   v. **Set Parameter:** These are the parameters that need to be added in the mongod.conf.
   vi. **Enable Auditlog:** Enable the specified audit log parameters.
   vii. **Auditlog Parameters:** These are the parameters that need to be added in the mongod.conf file.
e. **Mongodump Backup/Tuning details:**
   i. **Capture oplogs of replicaset:** Capture `oplogs` while running online `mongodump`.
   ii. **Mongodump Tuning Parameters**: These are the pairs of parameter name and value for `mongodump` command line.
   iii. **Mongorestore Tuning Parameters:** These are the pairs of parameter name and value for `mongorestore` command line.
f. **Enable Authentication** Enable member and client authentication for the database.
g. **dSource authentication details**:
   i. **Mongo Admin User:** This is the admin user of the staging database. This user should be created on the fly.
   ii. **MongoDB Password:** This is the admin user password of the staging database.
   iii. **Cluster Auth Mode:** The accepted values are: `None`, `keyfile`, `x509`, `sendKeyFile`, `sendX509`.
   iv. **KeyAuth keyfile name:** This is the absolute path of keyfile.
h. **Enable SSL/TLS:** Enable SSL/TLS member authentication.
   i. **SSL/TLS Parameters:** Specify at least the following parameters; `tlsMode`, `tlsCAFile`, `tlsCertificateKeyFile`.
i. **Enable data encryption:** Enable KMIP or keyfile based encryption.
j. **Data encryption details:**
   i. **Encryption Method:** keyfile or KMIP.
   ii. **Local Encryption keyfile:** This is the keyfile path for local encryption.
   iii. **KMIP Parameters:** Provide the KMIP parameter details. For example, `kmipClientCertificateFile`, `kmipPort`, `kmipServerCAFile`, `kmipServerName`.
   iv. **Client Certificate for Mongos/Mongo session on Primary Node:** This is the absolute path of client certificate to connect to staging database.

7. Click **Submit** to create a dSource.

> You must specify the inputs `KeyAuth keyfile name` and `Cluster Auth Mode` when **Enable Authentication** is selected.

9.5.5.4.4 extendedcluster

Perform the following steps to create a dSource with the extendedcluster ingestion type.

9.5.5.4.4.1 Prerequisites

The source and staging instances must meet the host requirements as described in Requirements for MongoDB staging/target hosts[392].

9.5.5.4.4.2 Creating a dSource

1. Login to **Delphix Management** application.
2. Click **Manage** -> **Environments**.
3. Click on **Add dSource**.
4. In the **Add dSource** wizard, select the mongo source config created on the staging host.
5. Select the dSource type from the drop-down list in the dSource dialog window.
6. Enter the Mongo-specific parameters for your dSource configuration.
   a. **dSource Type:** This is the dSource type.
   b. **Source database details:**
      i. **Source MongoDB Host Connect String:** These are the details of source database.
      ii. **MongoDB Source Database User:** This is the source database username.
      iii. **MongoDB Password:** This is the source database password.
   c. **dSource (Sharded/Replica set) details:**
      i. **Staging Replica set Name:** This is the staging database replica set name. The default value is `dlpx_rs0`.
      ii. **Staging MongoDB Host:** This is the staging host for creating a dSource.
      iii. **Mount Path:** This is the mount path for dSource MongoDB instance data files.
      iv. **bindIP String:** The default value is `0.0.0.0`.
      v. **Staging Port:** This is the staging host port for MongoDB dSource instance.
   d. **dSource tuning details:**
      i. **Oplog Size:** This is the size of `oplogs`.
      ii. **Journal Commit Interval:** This is the journal commit interval, measured in ms.
      iii. **Storage Engine:** The default value is `wiredTiger`.
      iv. **Enable SetParams:** Enable the specified `setParameters`.
      v. **Set Parameter:** These are the parameters that need to be added in the mongod.conf.
      vi. **Enable Auditlog:** Enable the specified audit log parameters.
      vii. **Auditlog Parameters:** These are the parameters that need to be added in the mongod.conf file.
   e. **Enable Authentication:** Enable member and client authentication for the database.

392 https://cd.delphix.com/docs/latest/requirements-for-mongodb-staging-target-environmen

f. **dSource authentication details:**
   i. **Mongo Admin User:** This is the admin user of the staging database. This user should be created on the fly.
   ii. **MongoDB Password:** This is the admin user password of the staging database.
   iii. **Cluster Auth Mode:** The accepted values are: `None`, `keyFile`, `x509`, `sendKeyFile`, `sendX509`.
   iv. **KeyAuth keyfile name:** This is the absolute path of keyfile.
g. **(Optional) Enable SSL/TLS:** Enable SSL/TLS member authentication.
   i. **SSL/TLS parameters:** Specify at least the following parameters; `tlsMode`, `tlsCAFile`, and `tlsCertificateKeyFile`.

7. Click **Submit** to create a dSource.

> You must specify the inputs `KeyAuth keyfile name` and `Cluster Auth Mode` when **Enable Authentication** is selected.

9.5.5.4.5 shardedsource (using MongoDB Ops Manager backup files)

Perform the following steps to create a dSource with the shardedsource ingestion type.

9.5.5.4.5.1 Prerequisites

- The source and staging instances must meet the host requirements as described in Requirements for MongoDB staging/target hosts[393].
- For creating a dSource type of shardedsource, the **Backup Metadata File Name** parameter is mandatory. The content of this file should be a single line `datetime` value, specified in format of `MMDDYYYY_HH24MISS`. The content of this file allows the Delphix Continuous Data Engine to validate if there is any new backup uploaded on the staging host that is different from the previous snapshot.
  - Delphix Continuous Data Engine checks this file based on a SnapSync policy to check whether the new backup is available for ingestion or not. It performs no other checks on backups. If date is updated, Delphix Continuous Data Engine assumes that there are new backup files available in the backup location.
  - Delphix Continuous Data Engine will reingest backup and take a new snapshot. This file must be updated manually or as a part of the backup/copy script. The following command can be added to the backup script that will be executed as soon as the backup is successful or backup copy is successful.

393 https://cd.delphix.com/docs/latest/requirements-for-mongodb-staging-target-environmen

```
echo "`date '+%m%d%Y_%H%M%S'`" > /<PATH>/mongo_bkp_metadata.txt
```

The file name and path must adhere to the organization standard naming convention.

- For shardedsource, it is expected to have one configserver backup file and one backup file per shard. It is important to keep the name of backup files same for every new backup. If the file names are changed, it needs to be updated in GUI parameters every time a new backup is uploaded. It is recommended to use some standard naming convention for sharded backups.
- The expected backup file format is as follows: `tar.gz` or `tar`

The below list shows the sample names:

- Configsvr_bkp.tar.gz
- myShard_0_bkp.tar.gz
- myShard_1_bkp.tar.gz
- myShard_2_bkp.tar.gz

9.5.5.4.5.2 Creating dSource

1. Login to **Delphix Management** application.
2. Click **Manage** > **Environments.**
3. Click on **Add dSource**.
4. In the **Add dSource** dialog window, select the MongoDB source config created on the staging host.
5. Select the dSource type from the drop-down list in the dSource dialog window.
6. Enter the Mongo-specific parameters for your dSource configuration.
   a. **dSource Type:** This is the Type of dSource.
   b. **dSource (Sharded/Replica set) details:**
      i. **Staging Replicaset Name:** This is the staging Database replica set name. The default value is `dlpx_rs0`.
      ii. **Staging MongoDB Host:** This is the staging host for creating a dSource.
      iii. **Mount Path:** This is the mount Path for dSource MongoDB instance data files.
      iv. **bindIP String:** The default value is `0.0.0.0`.
      v. **Staging Port:** This is the staging host port for MongoDB dSource instance.
   c. **dSource tuning details:**
      i. **Oplog Size:** This is the size of `oplogs`.
      ii. **Journal Commit Interval:** This is the journal commit interval in ms.
      iii. **Storage Engine:** The default value is `wiredTiger`.
      iv. **Enable SetParams:** Enable the specified `setParameters`.
      v. **Set Parameter:** These are the parameters that must be added in the mongod.conf.
      vi. **Enable Auditlog:** Enable the specified audit log parameters.
      vii. **Auditlog Parameters:** These are the parameters that must be added in the mongod.conf file.

d. **Sharded dSource details:**
   i. **Mongos Port Number:** This is the port that must be used for MongoDB's router instance.
e. **OpsManager backup details:**
   i. **Backup Metadata File Name:** This is the file that contains backup `datetime`. The content of this file is in `MMDDYYYY_HH24MISS` format.
   ii. **Config Backup / Replicaset Backup:** This is the absolute path of config backup file name (Sharded).
   iii. **Shard Backup Files:** This is the absolute path of Shard Backup File. Add file for each Shard. One file per shard.
f. **Enable Authentication:** Enable member and client authentication for the database.
g. **dSource authentication details:**
   i. **Mongo Admin User:** This is the admin user of the staging database. This user should be created on the fly.
   ii. **MongoDB Password:** This is the admin user password of the staging database.
   iii. **Cluster Auth Mode:** The accepted values are: `None`, `keyfile`, `x509`, `sendKeyFile`, `sendX509`.
   iv. **KeyAuth keyfile name:** This is the absolute path of keyfile.
   v. **Enable SSL/TLS:** Enable SSL/TLS member authentication.
      1. **SSL/TLS Parameters:** Specify at least the following parameters; `tlsMode`, `tlsCAFile`, `tlsCertificateKeyFile`.
   vi. **Enable data encryption:** Enable KMIP or keyfile based encryption.
   vii. **Data encryption details:**
      1. **Encryption Method:** keyfile or KMIP.
      2. **Local Encryption keyfile:** This is the keyfile path for local encryption.
      3. **KMIP Parameters:** Provide the KMIP parameter details. For example, `kmipClientCertificateFile`, `kmipPort`, `kmipServerCAFile`, `kmipServerName`.
   viii. **Client Certificate for Mongos/Mongo session on Primary Node:** This is the absolute path of client certificate to connect to staging database.
   ix. **Client CA Certificate for Mongos/Mongo session on Primary Node:** This is the absolute path of root CA certificate that signed the client certificate.
h. Click **Submit** to create a dSource.

- If MongoDB Ops Manager backup is taken when the source database is encrypted, you must select the **Source is Encrypted** checkbox and provide corresponding values at `Source Encryption Method`, `KMIP Parameters` or `Local Encryption keyfile`.
- You must specify the inputs `KeyAuth keyfile name` and `Cluster Auth Mode` when **Enable Authentication** is selected.

9.5.5.4.6 shardedsource (using cluster-to-cluster sync)

Perform the following steps to create a dSource with the shardedsource ingestion type.

9.5.5.4.6.1 Prerequisites

- The source and staging instances must meet the host requirements as described in Requirements for MongoDB staging/target hosts[394].
- For MongoDB Atlas, you must obtain the MongoDB 3.4 or earlier connection string of Mongo Shell from Atlas UI to use it as source MongoDB host connect string.

9.5.5.4.6.2 Creating a dSource

1. Login to **Delphix Management** application.
2. Click **Manage** -> **Environments**.
3. Click on **Add dSource**.
4. In the **Add dSource** dialog window, select the MongoDB source config created on the staging host.
5. Select the dSource type from the drop-down list in the dSource dialog window.
6. Enter the Mongo-specific parameters for your dSource configuration.
   a. **dSource Type:** This is the Type of dSource.
   b. **dSource (Sharded/Replica Set) details:**
      i. **Staging Replicaset Name:** This is the staging Database replica set name. The default value is `dlpx_rs0`.
      ii. **Staging MongoDB Host:** This is the staging host for creating a dSource.
      iii. **Mount Path:** This is the mount Path for dSource MongoDB instance data files.
      iv. **bindIP String:** The default value is `0.0.0.0`.
      v. **Staging Port:** This is the staging host port for MongoDB dSource instance.
   c. **dSource tuning details:**
      i. **Oplog Size:** This is the size of `oplogs`.
      ii. **Journal Commit Interval:** This is the journal commit interval, measured in ms.
      iii. **Storage Engine:** The default value is `wiredTiger`.
      iv. **Enable SetParams:** Enable the specified `setParameters`.
      v. **Set Parameter:** These are the parameters that must be added in the mongod.conf.
      vi. **Enable Auditlog:** Enable the specified audit log parameters.
      vii. **Auditlog Parameters:** These are the parameters that must be added in the mongod.conf file.
   d. **Sharded dSource details:**
      i. **Mongos Port Number:** This is the port that must be used for MongoDB's router instance.

394 https://cd.delphix.com/docs/latest/requirements-for-mongodb-staging-target-environmen

ii. **Mongosync Port Number:** This the available port for the `mongosync` utility.
iii. **Enable Authentication:** Enable member and client authentication for the database.
iv. **dSource authentication details:**
    1. **Mongo Admin User:** This is the admin user of the staging database. This user should be created on the fly.
    2. **MongoDB Password:** This is the admin user password of the staging database.
    3. **Cluster Auth Mode:** The accepted values are: `None`, `keyfile`, `x509`, `sendKeyFile`, `sendX509`.
    4. **KeyAuth keyfile name:** This is the absolute path of keyfile.
v. **Enable SSL/TLS:** Enable SSL/TLS member authentication.
    1. **SSL/TLS Parameters:** Specify at least the following parameters; `tlsMode`, `tlsCAFile`, `tlsCertificateKeyFile`. tlsMode must be `allowTLS` or `preferTLS` for shardedsource (cluster-to-cluster sync).
vi. **Enable data encryption:** Enable KMIP or keyfile based encryption.
vii. **Data encryption details:**
    1. **Encryption Method:** keyfile or KMIP.
    2. **Local Encryption KeyFile:** This is the keyfile path for local encryption.
    3. **KMIP Parameters:** Provide the KMIP parameter details. For example, `kmipClientCertificateFile`, `kmipPort`, `kmipServerCAFile`, `kmipServerName`.
viii. **Client Certificate for Mongos/Mongo session on Primary Node:** This is the absolute path of client certificate to connect to staging database.
ix. **Client CA Certificate for Mongos/Mongo session on Primary Node:** This is the absolute path of root CA certificate that signed the client certificate.

e. Click **Submit** to create a dSource.

> You must specify the inputs `KeyAuth keyfile name` and `Cluster Auth Mode` when **Enable Authentication** is selected.

9.5.5.4.7 nonshardedsource (using MongoDB Ops Manager backup files)

Perform the following steps to create a dSource with the nonshardedsource ingestion type.

9.5.5.4.7.1 Prerequisites

- The source and staging instances must meet the host requirements as described in Requirements for MongoDB staging/target hosts[395].

395 https://cd.delphix.com/docs/latest/requirements-for-mongodb-staging-target-environmen

- For dSource type of offlinemongodump, **Backup Metadata File Name** parameter is mandatory. The content of this file should be a single line `datetime` value specified in format of `MMDDYYYY_HH24MISS`. The content of this file allows the Delphix Continuous Data Engine to validate if there is any new backup uploaded on the staging host that is different from the previous snapshot.
    - Delphix Continuous Data Engine checks this file based on a SnapSync policy to check whether the new backup is available for ingestion or not. It performs no other checks on backups. If date is updated, Delphix Continuous Data Engine assumes that there are new backup files available in the backup location.
    - Delphix Continuous Data Engine will reingest backup and take a new snapshot. This file must be updated manually or as a part of the backup/copy script. The following command can be added to the backup script that will be executed as soon as the backup is successful or backup copy is successful.

```
echo "`date '+%m%d%Y_%H%M%S'`" > /<PATH>/mongo_bkp_metadata.txt
```

The file name and path must adhere to the organization standard naming convention.

- For nonshardedsource, one backup file of the replica set is sufficient. It is important to keep the name of backup files same for every new backup. If the file names are changed, it needs to be updated in GUI parameters every time a new backup is uploaded.
- The expected backup file format is as follows: `tar.gz` or `tar` backup files. For example, `myRS_0_bkp.tar.gz`.

9.5.5.4.7.2 Creating a dSource

1. Login to **Delphix Management** application.
2. Click **Manage** -> **Environments**.
3. Click on **Add dSource**.
4. In the **Add dSource** dialog window, select the MongoDB source config created on the staging host.
5. Select the dSource type from the drop-down list in the dSource dialog window.
6. Enter the Mongo-specific parameters for your dSource configuration.
    a. **dSource Type:** This is the Type of dSource.
    b. **dSource (Sharded/Replica Set) details:**
        i. **Staging Replicaset Name:** This is the staging Database replica set name. The default value is `dlpx_rs0`.
        ii. **Staging MongoDB Host:** This is the staging host for creating a dSource.
        iii. **Mount Path:** This is the mount Path for dSource MongoDB instance data files.
        iv. **bindIP String:** The default value is `0.0.0.0`.
        v. **Staging Port:** This is the staging host port for MongoDB dSource instance.
    c. **dSource tuning details:**

i. **Oplog Size:** This is the size of oplogs.
ii. **Journal Commit Interval:** This is the journal commit interval, measured in ms.
iii. **Storage Engine:** The default value is `wiredTiger`.
iv. **Enable SetParams:** Enable the specified `setParameters`.
v. **Set Parameter:** These are the parameters that must be added in the mongod.conf.
vi. **Enable Auditlog:** Enable the specified audit log parameters.
vii. **Auditlog Parameters:** These are the parameters that must be added in the mongod.conf file.

d. **OpsManager backup details:**
   i. **Backup Metadata File Name:** This is the file that contains backup `datetime`. The content of this file is in `MMDDYYYY_HH24MISS` format.
   ii. **Config Backup/Replicaset Backup:** This is the absolute path of config backup file name (Sharded).
e. **Enable Authentication:** Enable member and client authentication for the database.
f. **dSource authentication details:**
   i. **Mongo Admin User:** This is the admin user of the staging database. This user should be created on the fly.
   ii. **MongoDB Password:** This is the admin user password of the staging database.
   iii. **Cluster Auth Mode:** The accepted values are: `None`, `keyfile`, `x509`, `sendKeyFile`, `sendX509`.
   iv. **KeyAuth keyfile name:** This is the absolute path of keyfile.
   v. **Enable SSL/TLS:** Enable SSL/TLS member authentication.
      1. **SSL/TLS Parameters:** Specify at least the following parameters; `tlsMode`, `tlsCAFile`, `tlsCertificateKeyFile`.
   vi. **Enable data encryption:** Enable KMIP or keyfile based encryption.
   vii. **Data encryption details:**
      1. **Encryption Method:** keyfile or KMIP.
      2. **Local Encryption KeyFile:** This is the keyfile path for local encryption.
      3. **KMIP Parameters:** Provide the KMIP parameter details. For example, `kmipClientCertificateFile`, `kmipPort`, `kmipServerCAFile`, `kmipServerName`.
   viii. **Client Certificate for Mongos / Mongo session on Primary Node:** This is the absolute path of client certificate to connect to staging database.
   ix. **Client CA Certificate for Mongos / Mongo session on Primary Node:** This is the absolute path of root CA certificate that signed the client certificate.
g. Click **Submit** to create a dSource.

> If MongoDB Ops Manager backup is taken when the source database is encrypted, you must select the **Source is Encrypted** checkbox and provide corresponding values at `Source Encryption Method`, `KMIP Parameters` or `Local Encryption keyfile`.

> You must specify the inputs `KeyAuth keyfile name` and `Cluster Auth Mode` when **Enable Authentication** is selected.

9.5.5.4.8 stagingpush

Staging Push transfers the responsibility of preparing a staging database to the end user. This ingestion method allows you to manually push the source data into the mount path backed by the Delphix Continuous Data Engine storage. It offers an alternative approach to the Staging pull model where the Delphix Continuous Data Engine takes care of preparing the dSource.

Perform the following steps to create a dSource with the stagingpush ingestion type.

9.5.5.4.8.1 Prerequisites

The source and staging instances must meet the host requirements as described in [Requirements for staging/target hosts](396).

9.5.5.4.8.2 Procedure

1. Login to **Delphix Management** application.
2. Click **Manage** -> **Environments**.
3. Click on **Add dSource**.
4. In the **Add dSource** dialog window, select the MongoDB source config created on the staging host.
5. Select the dSource type from the drop-down list in the dSource dialog window.
6. Enter the Mongo-specific parameters for your dSource configuration.
   a. **dSource Type:** This is the type of dSource.
   b. **Source database details:**
      i. **Source MongoDB Host Connect String:** Connection string of the database to be created on the staging host.
      ii. **MongoDB Source Database User:** Database username of the database to be created.
      iii. **MongoDB Password:** Database password.
   c. **dSource (Sharded/Replica Set) details:**
      i. **Staging Replicaset Name:** This is the staging database replica set name. Mandatory for stagingpush. It should be set to the replica set name which would be created.
      ii. **Staging MongoDB Host:** This is the staging host for creating a dSource.
      iii. **Mount Path:** This is the mount path for dSource MongoDB instance data files.
      iv. **bindIP String:** The default value is `0.0.0.0`.
      v. **Staging Port:** This is the staging host port for MongoDB dSource instance. (Should be similar to the port mentioned in the Source MongoDB Host Connect String)

396 https://cd.delphix.com/docs/latest/requirements-for-mongodb-staging-target-environmen

vi. **Client Certificate for Mongos/Mongo session on Primary Node:** This is the absolute path of client certificate to connect to staging database. Optional.
vii. **Client CA Certificate for Mongos/Mongo session on Primary Node:** This is the absolute path of root CA certificate that signed the client certificate. Optional.

d. Click **Submit** to create a dSource.

9.5.5.5 Taking a snapshot of the staging DB

Perform the following steps to take snapshots of the staging database.

1. Create a staging database on the Delphix Continuous Data Engine mount path by copying files at o/s level or using tools like MongoDB OPS manager. Refer to any of the below methods to create a staging database.
   a. **Create a new MongoDB database instance**
      i. Create a new empty single member MongoDB replica set cluster using MongoDB Ops Manager or by any other tool/mechanism.
      ii. Make sure to use the same Delphix Continuous Data Engine mount path used in dSource creation, appended with **/s0m0**, as the location for data files. Also, use the same port that was provided while creating a dSource.
      iii. You can optionally restore any required data on this instance using `mongorestore`, custom scripts, or MongoDB Ops Manager.
      iv. Update permissions of MongoDB files if any other user is used to create a MongoDB database instance.
   b. Use **one** of the following.

```
login as root
su -m mongod -c 'chmod -R 775 /mnt/provision/stagingpush/s0m0'
```

```
login as mongod
chmod -R 775 /mnt/provision/stagingpush/s0m0
```

2. **Create MongoDB secondary instance (nonsharded only)**
   a. Create a new empty single member MongoDB replica set cluster.
   b. Make sure to use the same Delphix Continuous Data Engine mount path used in dSource creation, appended with **/s0m0**, as the location for data files. Also, use the same port that was provided while creating a dSource.
   c. Modify the source database replica set cluster using MongoDB OPS manager or manually add the newly created single node MongoDB replica set cluster (created above), as a hidden member to the existing source database replica set cluster.
   d. Set `Votes = 0` and `Priority = 0`. This will ensure that this member never contributes in the election and never becomes primary.
   e. Once done, MongoDB source database instance will start replicating data from existing nodes to the newly created member.

2. After the staging database is created successfully with the desired data, take a snapshot.

9.5.5.5.1 Additional notes

- The initial snapshot that is created during the creation of a dSource includes an empty data directory located within the mount point created by the Delphix Continuous Data Engine. Consequently, creating Virtual Databases (VDBs) directly from this snapshot is not possible. Therefore, take a new snapshot manually when the staging database instance is created successfully.
- The Delphix Continuous Data Engine always appends **/s0m0** as a subfolder to the file system provided while creating staging push data ingestion type. For example, if /mnt/provision/stagingpush was provided as the mount point while creating a dSource, provide /mnt/provision/stagingpush**/s0m0** as the data file location while creating the MongoDB Instance manually or via tools.
- While restoring the backup using MongoDB Ops Manager or `mongorestore`, etc. make sure the **/mnt/provision/stagingpush/s0m0/restoreInfo.txt** file exists before taking a snapshot.
- **Replica Set** field must be set in the /mnt/provision/stagingpush/s0m0/restoreInfo.txt file before taking a snapshot, as shown in the example below.

```
#cat restoreInfo.txt
Restore Information
Group Name: <Group name>
Replica Set: <Replica Set Name>
Snapshot timestamp: <as Timestamp at UTC>
Restore timestamp:  <as a BSON Timestamp at UTC>
Last Oplog Applied: <as a BSON Timestamp at UTC>
MongoDB Version: <MongoDB Version>
Storage Engine: <Storage engine type>
Startup Options: {}
Encryption: <Only appears if encryption is enabled on the snapshot>
keyEncryptionKey UUID: <Only appears if encryption is enabled on the snapshot>
Master Key UUID: <Only appears if encryption is enabled on the snapshot>
```

9.5.6 Provisioning MongoDB virtual databases (VDBs)

To provision a virtual database (VDB), you can either use an existing dSource snapshot or use another VDB snapshot for VDB to VDB provisioning. A snapshot is a timeflow created in a dSource or VDB.

9.5.6.1 Prerequisites

- You should either have linked a dSource from a staging instance with instructions from Linking MongoDB data sources[397] or have already created a VDB from which you intend to provision another VDB.

397 https://cd.delphix.com/docs/latest/linking-mongodb-data-sources

- A MongoDB target environment must be setup with the necessary requirements as described in MongoDB requirements and prerequisites[398].
- A MongoDB binary with the same version as the source database must be installed on the target environment.

9.5.6.2 Procedure

1. Login to **Delphix Management** application.
2. Click **Manage** -> **Datasets**.
3. Select the dataset and the snapshot that you want provision from. Click the provision VDB icon to open the **provision VDB** wizard.
4. Select a target environment from the left pane, and an Installation to use from the dropdown list of available MongoDB instances on that environment.
5. Set the **Environment User** to be the Instance Owner.
    a. The picking of instance owners is only possible if you have multiple environment users set on that host.
6. Go to the **Target Configuration** section where you need to enter VDB specific parameters.
    a. **VDB details:**
        i. **Mount Path:** This is the mount path for VDB MongoDB instance data files.
        ii. **bindIP String:** The default value is `0.0.0.0`.
        iii. **Target Port/Target Port Pool for Mongo Instance(s). Specify only the first port for sharded mongo:**
            1. For nonsharded VDB's, this is the port that will be used for MongoDB instances.
            2. For sharded VDB's, this is the starting port.
        iv. Delphix Continuous Data Engine will keep incrementing port by one and use as many as needed based on the configuration selected.
7. **Expand to three member replicaset:** Select this option to expand VDB as a 3-member replicaset. This is supported only for dSource types `nonshardedsource` and `shardedsource`.
    a. **Sharded VDB details:**
        i. **Mongos Port:** This is the port to be used for MongoDB's router instance (This is applicable only for shardedsource dSource type).
    b. **VDB tuning details:**
        i. **Oplog Size:** This is the size of `oplogs`.
        ii. **Journal commit Interval:** This is the journal commit Interval in ms.
        iii. **Enable Auditlog:** Enable auditlog parameters.
        iv. **Auditlog Parameters:** Auditlog parameter details (i.e. `auditDestination`, `auditFilter`)
        v. **Enable SetParams:** Enable `SetParams`.

398 https://cd.delphix.com/docs/latest/mongodb-requirements-and-prerequisites

vi. **SetParams Parameters:** `SetParams` parameter details (i.e. `authenticationMechanisms`, `opensslCipherConfig`)

c. **Enable Authentication:** Enable authentication on VDB.

d. **VDB authentication details:**

i. **MongoDB Admin/Cluster admin User Password:** Update password of Delphix Continuous Data Engine user created during dSource creation. The default value is `delphix`.

ii. **Cluster Auth Mode:** The accepted value is `None`, `keyfile`, `x509`, `sendKeyFile`, `sendX509`.

iii. **keyfile_path:** This is the KeyAuth keyfile name (full path).

e. **Additional Nodes:** Click on the **Add** button to select additional nodes for creating replica set VDB or distributing Shards on different hosts.

f. **Enable SSL/TLS:** Enable SSL/TLS member authentication.

g. **SSL/TLS Parameters:** Specify at least the following parameters; `tlsMode`, `tlsCAFile`, `tlsCertificateKeyFile`.

h. **Client Certificate for Mongos/Mongo Session on Primary Node:** Client SSL/TLS certificate.

i. **Client CA Certificate for Mongos/Mongo Session on Primary Node:** Client CA SSL/TLS certificate.

j. **Data Encryption details (KMIP/keyfile):**

i. **Local Encryption Keyfile:** Full path of keyfile if used for encryption.

ii. **KMIP Parameters:** KMIP parameter details (i.e. `kmipClientCertificateFile`, `kmipPort`, `kmipServerCAFile`, `kmipServerName`)

8. Select a **Target Group** for the VDB.
9. Add a new group, if required.
10. Select a **Snapshot Policy** for the VDB.
11. Click **Next**.
12. Specify any desired hook operations.
13. Click **Next**.
14. Review the **Provisioning Configuration** and Data Management information.
15. Click **Submit**.

When provisioning starts, the progress of the job can be reviewed in the Datasets panel, or in the Job History panel of the Dashboard.

When provisioning completes, a new VDB timeflow will be created. Once the VDB is created successfully, you can review the datasets on the **Manage** -> **Datasets** screen in the left panel.

A MongoDB VDB has now been provisioned and moreover, the VDB will be included in the group you designated and listed in the Databases panel. If you select the VDB in the Databases panel and click the Open icon, you can view its card, which contains information about the database and its Data Management settings.

9.5.6.2.1 Additional notes

1. The following sections are in the form of parameter name and value format. Add as many parameters needed by clicking on **+Add** button.
    - SSL/TLS parameters
    - KMIP parameters
    - Auditlog parameters
    - SetParams parameters

Any parameters specified for the above section need to be compatible with `mongod` / `mongos` command line parameters.

2. Refer to official MongoDB documentation[399] for appropriate names of parameters.
3. If **Expand to three member replica set parameter** is selected, then:
    - Delphix Continuous Data Engine will convert a nonsharded VDB to a three-node replica set. Each additional instance (two instances) will incur storage as the size of the first instance. MongoDB will initiate a complete sync from existing members. This will impact storage, as each new member will receive a full copy of data from the existing host. Two members will use twice the amount of storage.
        - It is recommended to use a single node replica set (if possible) to save storage.
    - Delphix Continuous Data Engine will convert a sharded VDB to a three-node replica set for config server, as well as each shard. Each additional instance will incur storage as the size of the first instance. MongoDB will initiate a complete sync from existing members. This will impact storage as each new member will receive a full copy of data from the existing host. Storage usage will multiply based on the number of shards.
        - It is recommended to use a single node replica set (if possible) to save storage.

9.5.6.3 Rewinding a VDB

Rewinding a VDB rolls it back to a previous point in its timeflow and reprovisions the VDB. The VDB will no longer contain changes after the rewind point.

9.5.6.3.1 Prerequisites

To rewind a VDB, a user must have Delphix Continuous Data Engine admin credentials.

9.5.6.3.2 Procedure

Perform the following steps to rewind a VDB.

---

399 https://www.mongodb.com/docs/v5.0/reference/configuration-file-settings-command-line-options-mapping/#std-label-conf-file-command-line-mapping

1. Login to the **Delphix Management** application.
2. Under **Datasets**, select the VDB you want to rewind.
3. On the **Timeflow** tab, select the rewind point as a snapshot or a point in time.
4. Click **Rewind**.
5. If you want to use login credentials on the target environment other than those associated with the environment user, click **Provide Privileged Credentials**.
6. Click **Yes** to confirm.

9.5.7 MongoDB logging

The MongoDB connector produces three sets of logs during normal runtime. Refer to the list below to understand what they are and how you can access them.

1. **MongoDB plugin logs**
   a. The MongoDB connector is built using the plugin framework. Therefore, it adheres to Delphix Continuous Data Engine's standard connector logging behavior.
2. **MongoDB Instance logs**
   a. MongoDB Instance logs are located at `DSOURCE_MOUNT_PATH/logs` (for dSource).
   b. MongoDB Instance logs are located at `VDB_MOUNT_PATH/logs` (for VDB).
   c. Delphix log names are in the **dlpx.sNmM.port.mongod.log** format, where:
      i. N = 0,1,2,3,... [Shard Number] [Always 0 for any nonsharded implementation]
      ii. M = 0,1 OR 2 [Member Number]
      iii. port = port number of the MongoDB instance.
3. **mongosync logs**
   a. Only for shardedsource cluster-to-cluster Sync.
   b. MongoSync logs are located at `DSOURCE_MOUNT_PATH/.delphix/mongosync/mongosync.log`

> Most of the MongoDB related errors are always hidden in the logfile for the MongoDB instance. Examine logs under `DSOURCE_MOUNT_PATH/logs` and `VDB_MOUNT_PATH/logs` for further investigation.

9.5.8 MongoDB developer's guide

This section includes the following CLI operations.

- MongoDB discovery operations (see page 1273)
- MongoDB link operations (see page 1273)
- MongoDB provision operations (see page 1278)

9.5.8.1 MongoDB discovery operations

To discover an environment via CLI, login to the Delphix Continuous Data Engine using the admin user account.

```
ssh admin@<CONTINUOUS-DATA-FQDN-or-IP>
```

9.5.8.1.1 Discovering the environment

Run the following commands to discover the environment.

```
environment
create
set hostEnvironment.name=demo-stg
set hostParameters.host.address=demo-stg.dcol1.delphix.com
set hostParameters.host.toolkitPath=/var/tmp
set primaryUser.name=delphix_os
set primaryUser.credential.password=delphix
commit
```

9.5.8.2 MongoDB link operations

9.5.8.2.1 dSource creation via CLI

To create a dSource and perform operations via CLI, login to the Delphix Continuous Data Engine using the admin user account.

```
ssh admin@<CONTINUOUS-DATA-FQDN-or-IP>
```

9.5.8.2.2 Set default linking parameters (common for all types of ingestions)

Run the commands below to set the default linking parameters. The below example uses `testscfg` source config.

```
database link
set type=LinkParameters
set linkData.type=AppDataStagedLinkData
set name=testds
set group=Untitled
set linkData.config=testscfg
```

```
set linkData.environmentUser=demo-stg/delphix_os
set linkData.stagingEnvironment=demo-stg
set linkData.stagingEnvironmentUser=demo-stg/delphix_os
set linkData.syncParameters.parameters={\"resync\":true}
```

9.5.8.2.3 Link parameters for the seed ingestion type

Run the commands below to link parameters for the seed ingestion type.

```
set linkData.parameters="{\"dsource_type_definition\":\"seed - [Replicaset(Empty
Instance)]\",\"src_db_details\":[],\"staging_db_details\":[{\"replicaset_name\":
\"myrs0\",\"staging_host\":\"demo-stg.dcol1.delphix.com\",\"mount_path\":\"\/mnt\/
provision\/seed_cli_stg\",\"bind_ip\":\"0.0.0.0\",\"start_portpool\":37701}],
\"staging_tuning_details\":[],\"mongodump_config\":[],\"shard_details\":[],
\"ops_backup_details\":[],\"enable_authentication\":true,\"staging_db_auth\":
[{\"mongo_db_user\":\"dlpxadmin\",\"mongo_db_password\":\"delphix\",
\"cluster_auth_mode\":\"keyFile\",\"keyfile_path\":\"\/home\/delphix\/keyfile\/
keyfile\"}],\"enable_ssl_tls\":false,\"ssl_tls_params\":[],\"enable_ldap\":false,
\"ldap_params\":[],\"source_encrypted\":false,\"encryption_params\":[],
\"client_tls_cert\":\"\",\"client_tls_cacert\":\"\"}"
```

9.5.8.2.4 Link parameters for the offlinemongodump ingestion type

Run the commands below to link parameters for the offlinemongodump ingestion type.

```
set linkData.parameters="
{\"dsource_type_definition\":\"offlinemongodump - [Replicaset(Manual mongodump
backups)]\",\"src_db_details\":[],\"staging_db_details\":[{\"replicaset_name\":
\"myrs0\",\"staging_host\":\"demo-stg.dcol1.delphix.com\",\"mount_path\":\"\/mnt\/
provision\/test_stg\",\"bind_ip\":\"0.0.0.0\",\"start_portpool\":37520}],
\"staging_tuning_details\":[],\"mongodump_config\":[{\"backup_metadata_file\":\"\/
home\/delphix\/509_backup\/offlinedump\/bkp_metadata_file.txt\",\"backupfile\":\"\/
home\/delphix\/509_backup\/offlinedump\/dlpx_cfgrs509\",\"enable_logsync\":false,
\"mongodump_params\":[],\"mongorestore_params\":[]}],\"shard_details\":[],
\"ops_backup_details\":[],\"enable_authentication\":true,\"staging_db_auth\":
[{\"mongo_db_user\":\"dlpxadmin\",\"mongo_db_password\":\"delphix\",
\"cluster_auth_mode\":\"keyFile\",\"keyfile_path\":\"\/home\/delphix\/keyfile\/
keyfile\"}],\"enable_ssl_tls\":false,\"ssl_tls_params\":[],\"enable_ldap\":false,
\"ldap_params\":[],\"source_encrypted\":false,\"encryption_params\":[],
\"client_tls_cert\":\"\",\"client_tls_cacert\":\"\"}"
```

9.5.8.2.5 Link parameters for the onlinemongodump ingestion type

Run the commands below to link parameters for the onlinemongodump ingestion type.

```
set linkData.parameters="
{\"dsource_type_definition\":\"onlinemongodump - [Replicaset(Automated mongodump
backups)]\",\"src_db_details\":[{\"host_conn_string\":\"demo-
src.dcol1.delphix.com:30030\",\"db_user\":\"puneadmin\",\"db_password\":
\"delphix\"}],\"staging_db_details\":[{\"replicaset_name\":\"myrs0\",
\"staging_host\":\"demo-stg.dcol1.delphix.com\",\"mount_path\":\"\/mnt\/provision\/
test_stg\",\"bind_ip\":\"0.0.0.0\",\"start_portpool\":33702}],
\"staging_tuning_details\":[],\"mongodump_config\":[],\"shard_details\":[],
\"ops_backup_details\":[],\"enable_authentication\":true,\"staging_db_auth\":
[{\"mongo_db_user\":\"dlpxadmin\",\"mongo_db_password\":\"delphix\",
\"cluster_auth_mode\":\"keyFile\",\"keyfile_path\":\"\/home\/delphix\/keyfile\/
keyfile\"}],\"enable_ssl_tls\":false,\"ssl_tls_params\":[],\"enable_ldap\":false,
\"ldap_params\":[],\"source_encrypted\":false,\"encryption_params\":[],
\"client_tls_cert\":\"\",\"client_tls_cacert\":\"\"}
```

9.5.8.2.6 Link parameters for the shardedsource (MongoDB Ops Manager backup) ingestion type

Run the commands below to link parameters for the shardedsource ingestion type.

```
set linkData.parameters="
{\"client_tls_cacert\":\"\",\"client_tls_cert\":\"\",\"dsource_type_definition\":
\"shardedsource - [Sharded(Ops Manager Backups)]\",\"enable_authentication\":true,
\"enable_ldap\":false,\"enable_ssl_tls\":false,\"encryption_params\":[],
\"ldap_params\":[],\"mongodump_config\":[],\"ops_backup_details\":[],
\"shard_details\":[],\"source_encrypted\":false,\"src_db_details\":[],
\"ssl_tls_params\":[],\"staging_db_auth\":[{\"mongo_db_user\":\"dlpxadmin\",
\"mongo_db_password\":\"delphix\",\"cluster_auth_mode\":\"keyFile\",\"keyfile_path\":
\"\/home\/delphix\/keyfile\/keyfile\"}],\"staging_db_details\":[{\"bind_ip\":
\"0.0.0.0\",\"mount_path\":\"\/mnt\/provision\/sharded_cli_stg\",\"replicaset_name\":
\"myrs0\",\"staging_host\":\"demo-stg.dcol1.delphix.com\",\"start_portpool\":28000}],
\"staging_tuning_details\":[]\"d_source_type\":\"shardedsource\",
\"src_mongo_host_conn\":\"\",\"src_replicaset_name\":\"\",\"src_db_user\":\"\",
\"src_db_password\":\"\",\"enable_logsync\":false,\"mongo_host\":\"demo-
stg.dcol1.delphix.com\",\"start_portpool\":28501,\"mongos_port\":28511,\"bind_ip\":
\"0.0.0.0\",\"oplog_size\":1024,\"journal_interval\":30,\"storage_engine\":
\"wiredTiger\",\"mount_path\":\"\/mnt\/provision\/sharded_stg\",
\"backup_metadata_file\":\"\/home\/delphix\/509_backup\/sharded_mongo_bkpdate.txt\",
\"mongo_db_user\":\"dlpxadmin\",\"mongo_db_password\":\"delphix\",
\"config_backupfile\":\"\/home\/delphix\/509_backup\/shardedbkp\/config-
configRS.tar.gz\",\"shard_backupfiles\":[{\"backupfile\":\"\/home\/delphix\/
509_backup\/shardedbkp\/myShard_0.tar.gz\"},{\"backupfile\":\"\/home\/delphix\/
509_backup\/shardedbkp\/myShard_1.tar.gz\"},{\"backupfile\":\"\/home\/delphix\/
509_backup\/shardedbkp\/myShard_2.tar.gz\"}],\"enable_ssl_tls\":false,
\"ssl_tls_params\":[],\"enable_authentication\":true,\"keyfile_path\":\"\/home\/
delphix\/509_backup\/shardedbkp\/keyfile\",\"cluster_auth_mode\":\"keyFile\",
\"source_encrypted\":false,\"encryption_method\":\"KeyFile\",\"encryption_keyfile\":
\"\",\"kmip_params\":[],\"enable_ldap\":false,\"ldap_params\":[],
```

```
\"enable_auditlog\":false,\"auditlog_params\":[],\"enable_setparams\":false,
\"setparam_params\":[],\"client_tls_cert\":\"\",\"client_tls_cacert\":\"\"}"
```

9.5.8.2.7 Link parameters for the shardedsource (cluster-to-cluster sync) ingestion type

Run the commands below to link parameters for the shardedsource ingestion type.

```
set linkData.parameters="
{\"dsource_type_definition\":\"shardedsource - [Sharded(Cluster-to-cluster Sync)]\",
\"src_db_details\":[{\"host_conn_string\":\"mkk-punedb01.dcol1.delphix.com:52011\",
\"db_user\":\"puneadmin\",\"db_password\":\"DELPHIX@123\"}],\"staging_db_details\":
[{\"replicaset_name\":\"\",\"staging_host\":\"demo-stg.dcol1.delphix.com\",
\"mount_path\":\"\/mnt\/provision\/sharded_stg\",\"bind_ip\":\"0.0.0.0\",
\"start_portpool\":46701}],\"staging_tuning_details\":[],\"mongodump_config\":[],
\"shard_details\":[{\"mongos_port\":46801,\"mongosync_port\":46899}],
\"ops_backup_details\":[],\"enable_authentication\":true,\"staging_db_auth\":
[{\"mongo_db_user\":\"dlpxadmin\",\"mongo_db_password\":\"delphix\",
\"cluster_auth_mode\":\"keyFile\",\"keyfile_path\":\"\/home\/delphix\/keyfile\/
keyfile\"}],\"enable_ssl_tls\":false,\"ssl_tls_params\":[],\"enable_ldap\":false,
\"ldap_params\":[],\"source_encrypted\":false,\"encryption_params\":[],
\"client_tls_cert\":\"\",\"client_tls_cacert\":\"\"}"
```

9.5.8.2.8 Link parameters for the nonshardedsource (MongoDB Ops Manager backup) ingestion type

Run the commands below to link parameters for the nonshardedsource ingestion type.

```
set linkData.parameters="{\"client_tls_cacert\":\"\",\"client_tls_cert\":\"\",
\"dsource_type_definition\":\"nonshardedsource - [Replicaset(Ops Manager Backups)]\",
\"enable_authentication\":true,\"enable_ldap\":false,\"enable_ssl_tls\":false,
\"encryption_params\":[],\"ldap_params\":[],\"mongodump_config\":[],
\"ops_backup_details\":[],\"shard_details\":[],\"source_encrypted\":false,
\"src_db_details\":[],\"ssl_tls_params\":[],\"staging_db_auth\":[{\"mongo_db_user\":
\"dlpxadmin\",\"mongo_db_password\":\"delphix\",\"cluster_auth_mode\":\"keyFile\",
\"keyfile_path\":\"\/path\/keyfile\"}],\"staging_db_details\":[{\"bind_ip\":
\"0.0.0.0\",\"mount_path\":\"\/mnt\/provision\/test_stg\",\"replicaset_name\":
\"myrs0\",\"staging_host\":\"demo-stg.dcol1.delphix.com\",\"start_portpool\":28000}],
\"staging_tuning_details\":[]\"d_source_type\":\"nonshardedsource\",
\"src_mongo_host_conn\":\"\",\"src_replicaset_name\":\"\",\"src_db_user\":\"\",
\"src_db_password\":\"\",\"enable_logsync\":false,\"mongo_host\":\"demo-
stg.dcol1.delphix.com\",\"start_portpool\":28501,\"mongos_port\":28511,\"bind_ip\":
\"0.0.0.0\",\"oplog_size\":1024,\"journal_interval\":30,\"storage_engine\":
\"wiredTiger\",\"mount_path\":\"\/mnt\/provision\/test_stg\",
\"backup_metadata_file\":\"\/path\/offline_mongo_bkpdate.txt\",\"mongo_db_user\":
\"dlpxadmin\",\"mongo_db_password\":\"delphix\",\"config_backupfile\":\"\/path\/
Replicaset_backup.tar.gz\",\"shard_backupfiles\":[],\"enable_ssl_tls\":false,
\"ssl_tls_params\":[],\"enable_authentication\":true,\"keyfile_path\":\"\/path\/
keyfile\",\"cluster_auth_mode\":\"keyFile\",\"source_encrypted\":false,
```

```
\"encryption_method\":\"KeyFile\",\"encryption_keyfile\":\"\",\"kmip_params\":[],
\"enable_ldap\":false,\"ldap_params\":[],\"enable_auditlog\":false,
\"auditlog_params\":[],\"enable_setparams\":false,\"setparam_params\":[],
\"client_tls_cert\":\"\",\"client_tls_cacert\":\"\"}"
```

9.5.8.2.9 Link parameters for the extendedsource ingestion type

Run the commands below to link parameters for the extendedsource ingestion type.

```
set linkData.parameters="{\"dsource_type_definition\":\"extendedcluster-
[Replicaset(Addmembertoexistingcluster)]\",\"src_db_details\":[{\"host_conn_string\":
\"mkk-punedb01.dcol1.delphix.com:51001\",\"db_user\":\"puneadmin\",\"db_password\":
\"DELPHIX@123\"}],\"staging_db_details\":[{\"replicaset_name\":\"\",\"staging_host\":
\"demo-stg.dcol1.delphix.com\",\"mount_path\":\"\/mnt\/provision\/extended_stg\",
\"bind_ip\":\"0.0.0.0\",\"start_portpool\":34568}],\"staging_tuning_details\":[],
\"mongodump_config\":[],\"shard_details\":[],\"ops_backup_details\":[],
\"enable_authentication\":true,\"staging_db_auth\":[{\"mongo_db_user\":\"dlpxadmin\",
\"mongo_db_password\":\"delphix\",\"cluster_auth_mode\":\"keyFile\",\"keyfile_path\":
\"\/home\/delphix\/keyfile\/keyfile\"}],\"enable_ssl_tls\":false,\"ssl_tls_params\":
[],\"enable_ldap\":false,\"ldap_params\":[],\"source_encrypted\":false,
\"encryption_params\":[],\"client_tls_cert\":\"\",\"client_tls_cacert\":\"\"}"
```

9.5.8.2.10 Link parameters for the stagingpush ingestion type

Run the commands below to link parameters for the stagingpush ingestion type.

```
set linkData.parameters="{\"dsource_type_definition\":\"stagingpush-
[Replicaset(EmptyvFiles)]\",\"src_db_details\":[{\"host_conn_string\":\"demo-
stg.dcol1.delphix.com:51230\",\"db_user\":\"puneadmin\",\"db_password\":
\"DELPHIX@123\"}],\"staging_db_details\":[{\"replicaset_name\":\"\",\"staging_host\":
\"demo-stg.dcol1.delphix.com\",\"mount_path\":\"\/mnt\/provision\/stagingpush_stg\",
\"bind_ip\":\"0.0.0.0\",\"start_portpool\":0}],\"staging_tuning_details\":[],
\"mongodump_config\":[],\"shard_details\":[],\"ops_backup_details\":[],
\"enable_authentication\":false,\"staging_db_auth\":[],\"enable_ssl_tls\":false,
\"ssl_tls_params\":[],\"enable_ldap\":false,\"ldap_params\":[],
\"source_encrypted\":false,\"encryption_params\":[],\"client_tls_cert\":\"\",
\"client_tls_cacert\":\"\"}"
```

9.5.8.2.11 Commit

Run the command below to commit the change.

```
commit;
```

9.5.8.2.12 Disable

Run the command below to disable the dSource.

```
source
select testdb
disable
commit
```

9.5.8.2.13 Enable

Run the command below to enable the dSource.

```
source
select testdb
enable
commit
```

9.5.8.3 MongoDB provision operations

To create a VDB via CLI, you must first create a dSource. For more information on creating a dSource, refer to the MongoDB link operations[400].

To create a VDB and perform operations via CLI, login to the Delphix Continuous Data Engine using the admin user account.

```
ssh admin@<CONTINUOUS-DATA-FQDN-or-IP>
```

9.5.8.3.1 Provision

Run the commands below to provision a VDB. The example below uses **testdb** dSource.

```
database provision
set type=AppDataProvisionParameters
set container.name=testvdb
set container.group=Untitled
set sourceConfig.repository=APPDATA_REPOSITORY-15
set timeflowPointParameters.type=TimeflowPointSnapshot
set timeflowPointParameters.snapshot=APPDATA_SNAPSHOT-253
set sourceConfig.name=testscfgv
```

400 https://cd.delphix.com/docs/latest/mongodb-link-operations

```
set source.allowAutoVDBRestartOnHostReboot=false
set source.parameters="{\"vdb_details\":[{\"mount_path\":\"\/mnt\/provision\/
test_vdb\",\"bind_ip\":\"0.0.0.0\",
\"start_portpool\":47701,\"make_shards_replicaset\":false}],\"vdb_sharded_details\":
[],\"vdb_tuning_details\":[],\"enable_authentication\":true,\"vdb_auth_details\":
[{\"mongo_db_password\":\"delphix\",\"cluster_auth_mode\":\"keyFile\",
\"keyfile_path\":\"\/home\/delphix\/keyfile\/keyfile\"}],\"additional_nodes\":[],
\"enable_ssl_tls\":false,\"ssl_tls_params\":[],\"client_tls_cert\":\"\",
\"client_tls_cacert\":\"\",\"enable_ldap\":false,\"ldap_params\":[],
\"encryption_params\":[]}"

commit
```

9.5.8.3.2 Disable

Run the commands below to disable a dSource. You must select the same sourceConfig name that was given during VDB creation.

```
source
select testscfgv
disable
commit
```

9.5.8.3.3 Enable

Run the command below to enable a dSource.

```
source
select testscfgv
enable
commit
```

9.5.8.3.4 Stop

Run the command below to stop a dSource.

```
source
select testscfgv
stop
commit
```

9.5.8.3.5 Start

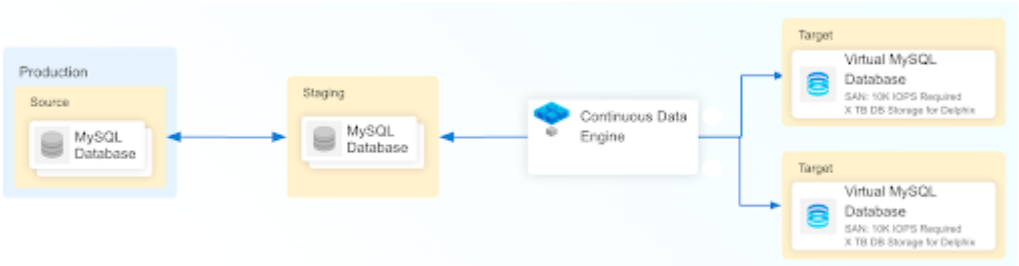
Run the command below to start a dSource.

```
source
select testscfgv
start
commit
```

9.6 MySQL data sources

9.6.1 Overview

The Delphix MySQL virtualization solution consists of the Delphix Continuous Data Engine and the installed MySQL connector. This solution ingests source data using the staging architecture, which copies the production database from the source environment into a staging environment. The MySQL connector helps capture the persistent storage layer of the staging database, allowing the Delphix Continuous Data Engine to create virtualized copies in target environments.



The MySQL connector supports both staging pull and staging push for data preparation inside of the staging database.

The following specific ingestion methods are supported:

1. **Staging Pull:** This ingestion method enables Delphix Continuous Data Engine to execute the necessary operations to prepare and ingest the data contents from the staging environment's database. Staging pull supports the following ingestion methods:
    a. Replication with Delphix-Initiated backups (binlog replication)
    b. Replication with externally-initiated backups provided by the user (binlog replication)

> In the staging pull mechanism, only binlog Replication is supported.

2. **Staging Push:** This ingestion method requires you to customize and execute the necessary operations to prepare the data content in the staging environment's database. Once ready, Delphix Continuous Data Engine ingests the data contents.

9.6.2 Limitations

Delphix Continuous Data Engine and the MySQL connector should not be considered a replacement for the backup and recovery of your source instances and databases. Delphix Continuous Data Engine solution is intended to create non-production database copies.

9.6.3 Support

The MySQL connector is a part of the Select Connector for MySQL license agreement. Purchase of this additional license is required for support. For any questions, bugs, or feature requests, contact us via the Delphix Support[401] or the Delphix Community[402] portals.

9.6.4 MySQL virtualization process

The following material introduces you to the overall process of deploying the MySQL virtualization solution. A sequence of high-level steps is below; each step briefly describes the task and the expected result. Refer to the embedded links for more detailed information.

1. Begin by reviewing the general Delphix Continuous Data Engine architecture diagrams (see page 1014). This will give a good understanding of the architectural strategy and confirm support. As a next step, identify existing or new MySQL source, staging and target environments, the underlying Operating Systems, and MySQL database versions.
2. Review the MySQL Support Matrix (see page 1282). Confirm that your combination of MySQL software, host operating system, Delphix Continuous Data Engine version, and MySQL connector version are supported for the various environments that will integrate into your MySQL deployment. Early validation will reduce unexpected issues later.
3. Deploy and configure Delphix Continuous Data Engine. (see page 556)
4. Install the MySQL connector (see page 1291). The MySQL connector extends Delphix Continuous Data Engine to MySQL datasets. It is recommended to stay current with the latest version of the connector.
5. Create staging and target environments (see page 1288). The staging environment will host a MySQL instance that Delphix Continuous Data Engine will ingest from. The target environment will host one or more MySQL instance(s) in which the virtual database copies will be created.
   a. Consult the source environment (see page 1286) and staging/target environment (see page 1288) requirements documentation for configuration details.
6. Add staging and target environments in the Delphix Continuous Data Engine (see page 1295). Use the **Add Environment** operation in Delphix Continuous Data Engine for each host, which causes Delphix Continuous Data Engine to register the specific host and catalog the MySQL software installation(s) on that host.

401 https://support.delphix.com/
402 https://community.delphix.com/contactus

7. Link the MySQL source database (see page 1301). Configure and create a dSource which allows Delphix Continuous Data Engine to capture a copy of the source data. There are numerous ingestion methods available that can provide a variety of methods for capturing data to be virtualized by Delphix Continuous Data Engine. Therefore, we encourage you to identify the ingestion method that best suits your organization and infrastructure policies.
8. Provision virtual MySQL databases (see page 1316). Administrators and users can provision multiple copies of the source data as virtual databases.

9.6.5 MySQL requirements and prerequisites

This section contains the following pages pertaining to MySQL requirements and prerequisites in Delphix Continuous Data Engine.

- MySQL support matrix (see page 1282)
- Requirements for MySQL source databases (see page 1286)
- Requirements for MySQL staging and target environments (see page 1288)

9.6.5.1 MySQL support matrix

Refer to the following support matrices to ensure compatibility between the environment, database, connector, and Delphix Continuous Data Engine.

Compatible upgrades are indicated by a green **Y**, while incompatible upgrades are indicated by a red **N**.

> According to Delphix Support Policies, when a minor release is certified, all patch releases that fall under that minor release are also certified.

9.6.5.1.1 Environment and database compatibility

The MySQL connector is certified for use in source, staging, and target environments with the following operating systems and database versions.

9.6.5.1.1.1 Operating System and connector

Operating System Version	**Connector version**									
	2.0.26	**2.1.0**	**3.0.0**	**4.0.0**	**4.1.0**	**4.2.0**	**4.3.0**	**4.4.0**	**4.5.0**	**4.6.0**

RHEL 6.9 / 7.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Amazon Linux 2.0	N	N	N	N	N	N	N	N	N	Y

9.6.5.1.1.2 Source environment vs provisioning staging/target environment

Source Environment	**Provisioning Staging/Target Environment**			
	MySQL Community 5.7	**Percona Server 5.7**	**MySQL Community 8.0.x**	**Percona Server 8.0.x**
MySQL Community 5.7	Y	Y	N	N
Percona Server 5.7	Y	Y	N	N
MySQL Community 8.0.x	N	N	Y	Y
Percona Server 8.0.x	N	N	Y	Y

The MySQL binaries must be installed in the staging and target environments. These binary versions must match the version of the MySQL binary on the source environment.

9.6.5.1.1.3 Database and connector

Database Version	**Connector version**									
MySQL Community 5.7.x	**2.0.26**	**2.1.0**	**3.0.0**	**4.0.0**	**4.1.0**	**4.2.0**	**4.3.0**	**4.4.0**	**4.5.0**	**4.6.0**
MySQL Community 5.7.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Percona Server 5.7.x	N	N	Y	Y	Y	Y	Y	Y	Y	Y
MySQL Community 8.0.x	N	N	N	Y	Y	Y	Y	Y	Y	Y
Percona Server 8.0.x	N	N	N	Y	Y	Y	Y	Y	Y	Y

9.6.5.1.2 Connector and Delphix Continuous Data Engine compatibility

The MySQL connector versions are certified to work with the following Delphix Continuous Data Engines.

Delphix Continuous Data Engine Version	Connector Version									
	2.0.26	2.1.0	3.0.0	4.0.0	4.1.0	4.2.0	4.3.0	4.4.0	4.5.0	4.6.0
6.0.4.x	Y	N	N	N	N	N	N	N	N	N
6.0.5.x	Y	N	N	N	N	N	N	N	N	N
6.0.6.x	Y	N	N	N	N	N	N	N	N	N
6.0.7.0 to 6.0.16.x	N	Y	N	N	N	N	N	N	N	N
6.0.16.0 and above	N	Y	Y	Y	Y	Y	Y	Y	Y	Y

For more information, refer to the Delphix End of Support[403] policy to determine whether a connector version is in primary, extended, or legacy support. Contact Delphix customer support with any questions.

9.6.5.1.3 Supported Cloud Platform-as-a-Service (PaaS) database services

The MySQL connector is certified for use with the following cloud PaaS database services and versions.

A Cloud PaaS database service cannot be used as staging or target environments due to strict limitations imposed by the cloud vendors. In these cases, the Cloud PaaS database service can be used only as a source, and self-managed hosts, with a supported operating system and equivalent open-source MySQL database versions, must be used for the staging and target environments. We recommend the Delphix Continuous Data Engine, staging, and target environments be hosted in the same Cloud and region as the PaaS database service.

Cloud database service	Database version
	MySQL Community 8.0.x

403 https://support.delphix.com/Support_Policies_and_Technical_Bulletins/Support_Policies/Software_Product_Line_Policies_(KBA1003)#:~:text=Delphix%20may%20discontinue%20a%20product,18%20months%20of%20Extended%20Support

Amazon Aurora MySQL	Y
Amazon RDS MySQL	Y

Only the single DB ingestion feature is supported for the cloud vendors using MySQL through the mysqldump method with Staging Pull. With Staging Push, other ingestion methods may be possible.

If you wish to combine multiple individual Cloud PaaS databases into a single MySQL instance, it is possible to do so with Staging Push. In this scenario, the individual databases are combined and restored into a single, common staging instance, resulting in one dSource which contains multiple databases and is accessed via one HTTP port.

9.6.5.2 Requirements for MySQL source databases

The requirements for MySQL source databases are only for the staging pull ingestion method. This ingestion method allows Delphix Continuous Data Engine to perform the operations required to prepare the data content in the staging environment using the binary log file position-based replication (binlog replication). The source database requirements are the same for both replication with Delphix-initiated backups and replication with externally-initiated backups provided by the user, and are as follows:

Requirements	Explanation
Connectivity to the source database from the staging database for replication	The connector should be able to connect to the source environment's database from the staging environment and take a backup of the source database(s) using the `mysqldump` utility.

Requirements	Explanation

Source DB user:
A Source DB user can connect to the source database both from the staging host and locally.
A Source DB user must be granted the following permissions in the source database:
SELECT, SHUTDOWN, SUPER, RELOAD, SHOW VIEW, EVENT, TRIGGER, REPLICATION CLIENT, REPLICATION SLAVE
	You are responsible for performing a logical backup of the source database with the `mysqldump` utility and initiating replication between the source and staging databases.
	`mysql -u<user> -p<password> --protocol=TCP --port=<port>`
	`mysql> CREATE USER 'delphix_os'@'<staging_host>' IDENTIFIED BY 'delphix_user_password';`
	`mysql> CREATE USER 'delphix_os'@'localhost' IDENTIFIED BY 'delphix_user_passwd';`
	`mysql> GRANT SELECT, SHUTDOWN, SUPER, RELOAD, SHOW VIEW, EVENT, TRIGGER, REPLICATION CLIENT, REPLICATION SLAVE on *.* to 'delphix_os'@'staging_host';`
	`mysql> GRANT SELECT, SHUTDOWN, SUPER, RELOAD, SHOW VIEW, EVENT, TRIGGER on *.* to 'delphix_os'@'localhost';`
	`'staging_host' = IP address of Staging Host`
	In addition, you have the ability to grant more permissive privileges; however this is not mandatory.
	`mysql> GRANT ALL PRIVILEGES ON *.* TO 'user'@'%';`
	Remember that this is the user that Delphix Continuous Data Engine uses to manage the staging database. As a result, it is recommended that you create a dedicated source db user for Delphix Continuous Data Engine with the privileges described here.

Requirements	Explanation

Binary logging should have been enabled on the source database.	Binary logging on the source database must be enabled in order to set up replication. You can check the status of binary logging as follows:
	`mysql> SHOW VARIABLES LIKE 'log_bin';`
	If binary logging is enabled, the following status should be displayed:
	`+---------------+-------+`
	`| Variable_name | Value |`
	`+---------------+-------+`
	`| log_bin | ON |`
	`+---------------+-------+`
	`1 row in set (0.01 sec)`

The source database must have a non-zero server-id.	The source database must have a non-zero server-id, and the server-id value must be different from the source's server-id value. This is required in order for replication between the source and staging databases to be enabled. Server-id can be checked in `my.cnf` file and MySQL database configuration file. It holds a list of parameters to configure the MySQL database server.

9.6.5.3 Requirements for MySQL staging and target environments

The staging and target environments must meet a set of requirements mentioned below:

Staging/Target environment requirements	Explanation
Host compatibility check The Operating System and architecture of the staging/target host must be compatible with the source environment.	The source and target environments must have identical Operating Systems and hardware configurations.
MySQL installation and compatibility check MySQL must be installed on the staging/target host in a way that is compatible with MySQL installation on the source host.	Two MySQL installations can only be compatible if and only if the staging/target's version number is equal to the source's version number.

Staging/Target environment requirements	Explanation
Network Connectivity Requirements	Delphix Continuous Data Engine, staging and target host must follow the network requirements mentioned in https://cd.delphix.com/docs/latest/network-connectivity-requirements The source database must be accessible from the staging host whenever Replication Ingestion Mechanism is used.

Staging/Target Operating System Environment OS User
An environment user with elevated permissions can execute ps, mount, umount, mkdir, and rmdir commands without entering a password.
The Delphix Continuous Data Engine connects to the target host (via SSH) to perform the following operations:
1. Mount and unmount directories for linking and provisioning operations. This privilege is required because mount and unmount are typically reserved for the superuser.
2. Create and remove directories for linking and provisioning operations.
3. Run all commands requested by the MySQL connector.
	You need to provide the below permissions in the sudoers file for the staging/target or Delphix Continuous Data Engine user for the plugin to work correctly:
	`1. Open the sudoers file using the below:`
	`vi /etc/sudoers`
	`# Or`
	`visudo`
	`2. Below entries need to be created in the sudoers file:`
	`Defaults:delphix_os !requiretty`
	`delphix_os ALL=NOPASSWD: \`
	`/bin/mount, /bin/umount, /bin/mkdir, /bin/rmdir, /bin/ps`
	Delphix Continuous Data Engine requires that the `requiretty` setting be disabled for all Delphix Continuous Data Engine users with sudo privileges.
	It is required to specify the NOPASSWD qualifier within the "sudo" configuration file, as shown here, This ensures that the "sudo" command does not demand the entry of a password, even for the “display permissions” (i.e. `sudo -l`) command.
	Note: Privilege elevation is not supported.

Staging/Target environment requirements	Explanation
Toolkit Path field On the staging/target host, a directory with the following properties must exist: 1. The directory must be writable by the previously mentioned staging/target operating system user. 2. The directory must have at least 1.5 GB of storage space available.	The directory on the staging/target host where the MySQL connector support files are installed (for example, `/opt/toolkit`) is specified in the "Toolkit Path" field on the Environment creation UI page.

Required utilities and tools
Make sure the **netstat** or **ss** utilities are installed on the staging/target host.
	MySQL connector checks the status of the running MySQL instance and the availability of ports on the staging/target host using the `netstat` or `ss` utilities.
	Checking to see if the netstat and ss utilities are installed.
	`which netstat`
	`which ss`

Set Hostname and IP in host file (Optional)
Hostname and IP can be defined in the `/etc/hosts` file.
	The `hostname -i` command should return the correct server IP address.
	For example:
	`[root@source ~]# hostname -i`
	`10.110.252.214`

Staging/Target environment requirements	Explanation
MySQL Configuration File (Optional) Define a `my.cnf` file for further configuration.	• The `my.cnf` file is a MySQL database configuration file. It holds a list of parameters to configure the MySQL database server. • You can place a starter `my.cnf` file into the Delphix Toolkit Directory, and Delphix Continuous Data Engine will use this `my.cnf` file for creating a staging database. • Having a `my.cnf` in the toolkit directory is not mandatory. If this file is not provided, Delphix Continuous Data Engine will create a `my.cnf` file from scratch. This `my.cnf` file will only be used to create staging databases; it will not be inherited by VDBs.

9.6.6 MySQL installation and upgrade

The MySQL connector is managed through Delphix Continuous Data Engine's plugin framework. Plugins are distributed separately from the Delphix Continuous Data Engine. The MySQL plugin must be installed before adding MySQL environments or performing any MySQL virtualization operations.

9.6.6.1 Downloading the connector

Follow the steps below to download the MySQL plugin:

1. Log in to the Delphix Download Portal[404] using your credentials from the Delphix support portal.
2. Find the MySQL folder by navigating to the `Delphix Product Releases / Continuous Data Connectors / MySQL / Linux` path and select the desired version's folder.
3. Download the JSON file by clicking `sc_mysqllinux_<version_number>.json`. For example, Delphix Product Releases / Continuous Data Connectors / MySQL / Linux[405] / v{mysql_connector_version} / sc_mysql_{mysql_connector_version}.json.

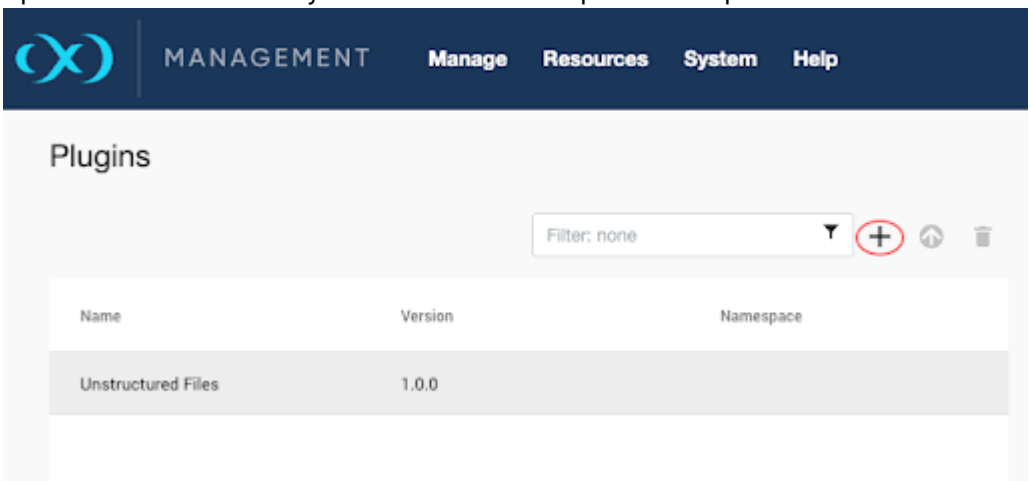
404 https://download.delphix.com/

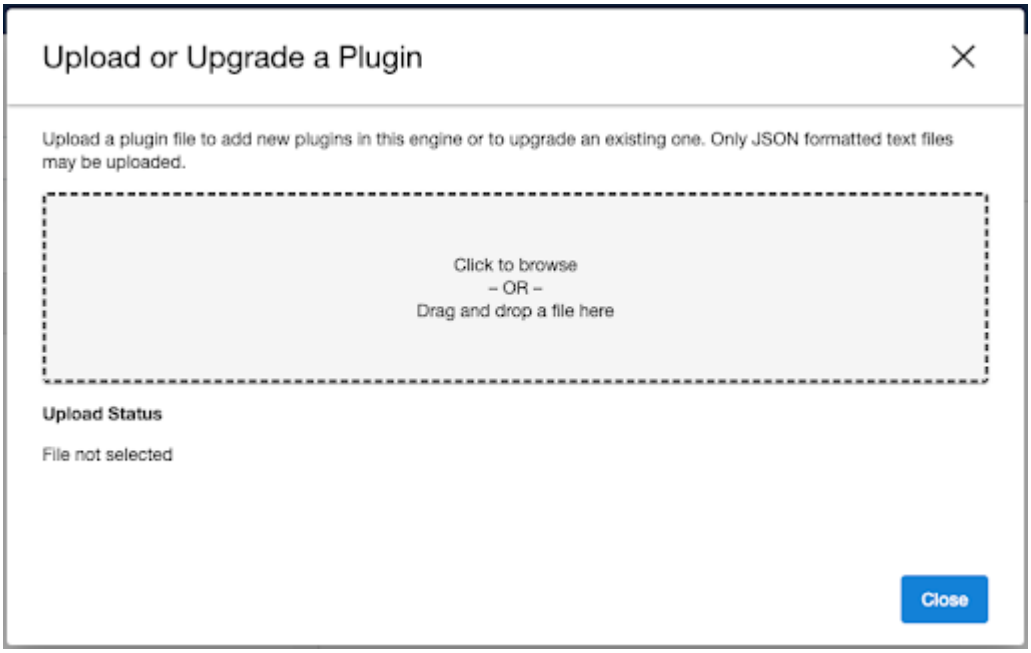
405 https://download.delphix.com/folder/5342/Delphix%20Product%20Releases/Connectors/Continuous%20Data/MySQL/Linux/v4.2.0#

9.6.6.2 Installing the connector

Follow the steps below to upload and install a new connector.

1. Log in to the Delphix Management application.
2. Navigate to the **Manage** > **Plugins** page.
3. To upload a plugin, click the plus icon. This action opens the **Upload or Upgrade a Plugin** dialog window.
4. Upload the JSON file that you downloaded in the previous step.





For more information, refer to the Delphix Continuous Data Engine plugin management (see page 2257) section.

9.6.6.3 Upgrading the connector

The connector must be upgraded to a specific version to accommodate new features and bug fixes that may affect its configuration. If the existing plugin version is older, upgrades may necessitate Delphix Continuous Data Engine upgrades as well.

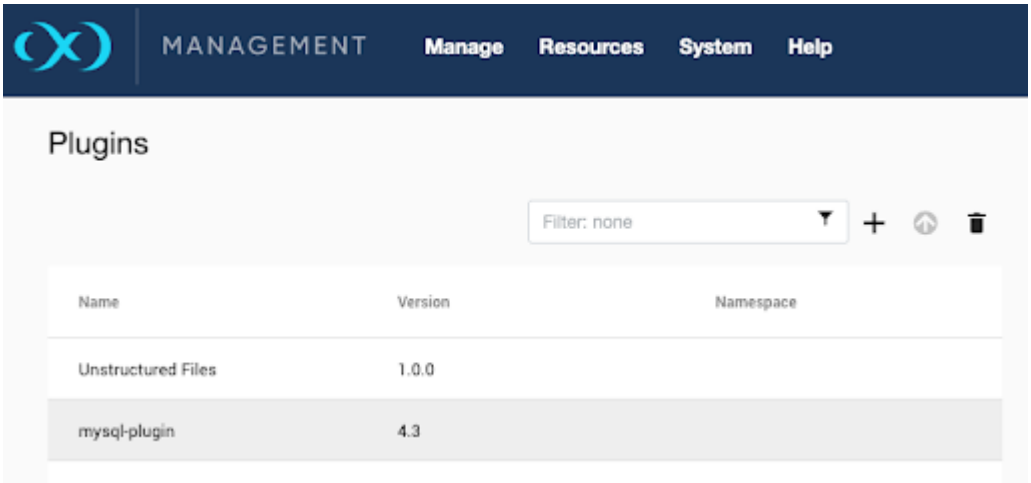
> ⚠ Skipping a recommended version upgrade path will result in configuration issues.

Follow the upgrade path outlined below to upgrade to newer versions of the MySQL connector.

Currently Installed Version	Recommended Upgrade Path
2.0.26	Upgrade to 2.1.0 and follow its upgrade path.
2.1.0	Upgrade to 3.0.0 and follow its upgrade path.
3.0.0	Upgrade directly to 4.0.0 and follow its upgrade path.
4.0.0 4.1.0 4.2.0	Upgrade directly to 4.3.0.
4.3.0	Upgrade directly to 4.4.0.
4.4.0	Upgrade directly to 4.5.0.

Along with the upgrade path, perform the following steps to ensure that your configuration remains compatible with minimal disruption.

1. Upload the most recent version of the plugin. Refer to the *Installing the connector* documentation above.
2. Once the upgrade is complete, navigate to **Manage** > **Plugins** to verify the plugin version.



> If you have installed a version other than those listed above, it is recommended that you upgrade it using the upgrade path outlined above. For example, if you installed version 2.026, then upgrade to 2.1.0 and continue the recommended upgrade path until you reach the most recent version.

3. Using the Delphix Continuous Data Engine UI, refresh all relevant environments associated with this connector.
4. Confirm that the connector discovery process was successful.
5. Create new snapshots of the dSource and VDB.

During connector/plugin upgrades, there is no need to disable and enable datasets.

9.6.7 MySQL connector operations

To successfully discover, ingest (link), and virtualize a MySQL instance, you must perform the following connector operations in sequence:

1. Discovering installations in environments (see page 1294)
2. Linking data sources (see page 1301)
3. Provisioning virtual databases (see page 1316)

The respective sections provide detailed information about prerequisites, limitations (if any), and the procedure to complete each operation.

9.6.7.1 Discovering MySQL installations

Discovery is a process that allows the MySQL connector to gather information about the MySQL installation in an environment. This action is performed in two phases: Dataset Homes (Installations) and Databases (Source Configs). It's important to note, in the case of MySQL, the database concept also includes MySQL's

Instance structure. MySQL performs the Dataset Home discovery automatically. However, Databases (i.e. Instances) must be added manually.

When any changes (such as installing a new database home) are made to an already configured environment in the Delphix Continuous Data Engine application, an environment refresh must be performed.

9.6.7.1.1 Prerequisites

- The MySQL connector must be installed before running discovery.
- MySQL databases and/or binaries should be present on staging/target environments.

9.6.7.1.2 Adding a MySQL environment to Delphix Continuous Data Engine

This topic describes how to add a MySQL staging and target environment.

9.6.7.1.2.1 Prerequisites

A MySQL staging environment must be configured to meet the requirements outlined in Requirements and Prerequisites (see page 1282).

9.6.7.1.2.2 Procedure

Perform the following steps to add an environment to the Delphix Continuous Data Engine:

1. Login to the **Delphix Management** application.
2. Navigate to the **Manage > Environments** page.
3. Click the three dots on the left panel and select **Add Environment**.
4. From the **Add Environment** dialog window, select **Unix/Linux.**
5. Select **Standalone Host**.
6. Click **Next**.
7. Provide the name of the environment.
8. Provide the **Host IP address** or **host FQDN**.
9. Provide the **SSH port**. The default value is 22.
10. Provide an **Environment Username** for the environment.
11. Choose the **Login Type** that fits your setup.

- **Username and Password** – provide the environment username and password.
- **Username and Public Key** – provide the environment username.
- **Password Vault** – select from an existing Enterprise Password Vault.

Using Public Key Authentication
If you want to use public-key authentication for logging into your Unix-based environment, you have two options:

- Use the Delphix Continuous Data Engine's key pair. For more information on how to use the Delphix Continuous Data Engine's key pair, refer to the How to use the Continuous Data Engine's key-pair (see page 1331) in FAQ.
- Provide a key pair for this environment. As an alternative, a key pair specific for this environment can be provided via the API or CLI. For instructions, refer to CLI Cookbook: Setting up SSH key authentication for UNIX environment users (see page 2548).

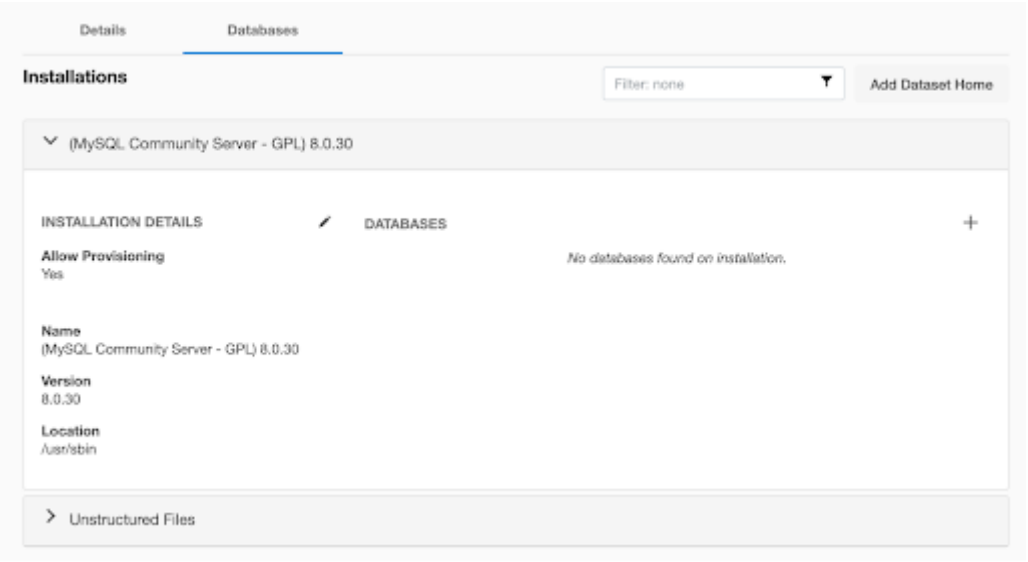
10. Based on your **Login Type**, authenticate the environment by following the directions below:
    a. For **Login Type** 'Username and Password'; enter the password associated with the user in step 11.
    b. For **Login Type** 'Username and Public Key';
        i. Enter the **OS username**.
        ii. Click **View Public Key**.
        iii. Copy the public key that is displayed, and append it to the end of your `~/.ssh/authorized_keys` file. If this file does not exist, it must be created.

Note: The addition of the public key is a one-time requirement per user and per environment. Additionally, you have the option to add public key authentication to an environment user's profile via CLI (For details, refer to the CLI Cookbook: Setting up SSH key authentication for UNIX environment users (see page 2548) topic).

12. Enter **Toolkit Path** (ensure that the toolkit path does not have any spaces).
13. To provide your own Oracle Java Development Kit, select the **Provide my own JDK** checkbox and click **Next**.
    a. On the Java Development Kit page, provide an absolute path to your Oracle JDK and click **Next**.
14. Click **Submit**.

After successfully adding the environment, you should be able to view the newly added environment listed under the **Environments** tab. Discovery should occur automatically to identify and present Dataset Homes. If Dataset Homes do not appear, click the **Refresh** icon.

If you wish to use this environment as a staging environment, then you must add a database (Source Config) to establish the source database.



9.6.7.1.3 Adding a MySQL dataset home

MySQL installations and binaries are represented as Dataset Homes in Delphix Continuous Data Engine. If you added a fresh environment or new database binary installations to an existing environment, then you must run the environment discovery process. This is performed through the **Refresh** action.

For more information, refer to the "Environment Refresh" section on the Environment management (see page 1017) page.

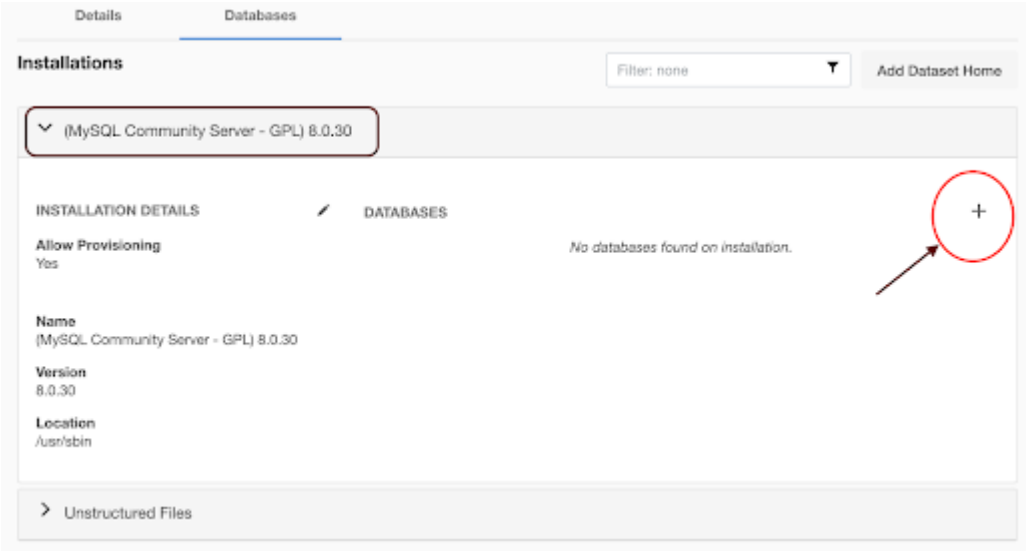
9.6.7.1.4 Add a MySQL database

All environments, as previously stated, contain one or more Dataset Homes, which are MySQL installations in the environment. These are automatically identified. From a Dataset Home, a database (or MySQL instance) must be manually created. This database contains the source configuration (Source Config) which defines the configuration of the dSource and staging database instance. Each environment may have many Dataset Homes which contain many associated databases (Source Configs). These databases are only applicable for staging environments.

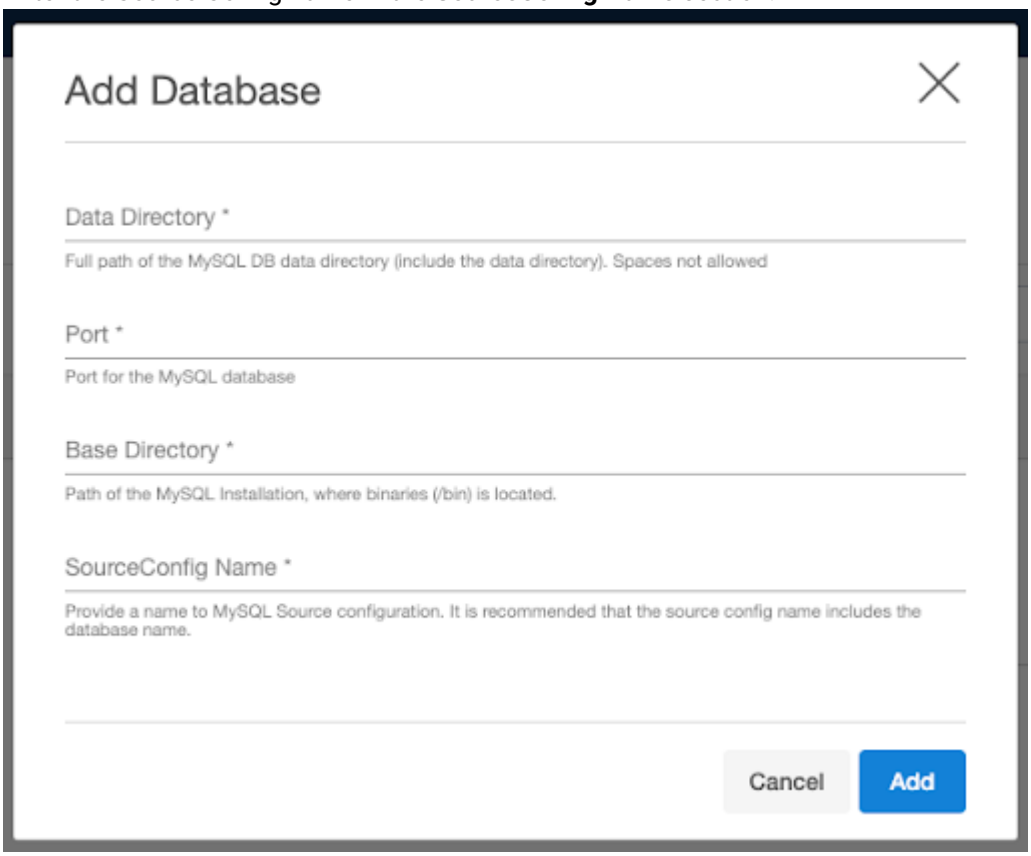
9.6.7.1.4.1 Procedure

Follow these steps to add a database (Source Config), which defines the configuration of the staging database instance, to a staging Environment's Dataset Home:

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Environments**.
4. Select the installation.
5. Click on the plus icon.



6. Add required source configuration details in the **Add Database** section:
   - Enter the source data directory in the section **Data Directory**.
   - Enter the port number on which the MySQL instance is running on a source in the **Port** section.
   - Enter the MySQL base directory on the source host in the **Base Directory** section.
7. Enter the Source Config name in the **SourceConfig Name** section.



Data Directory, Port, and Base Directory can be obtained from `my.cnf` file.

9.6.7.1.5 MySQL environment operations via CLI

The operations explained earlier in this section can also be completed via advanced CLI operations. This operations include:

- Create & Discover
- Update
- Refresh
- Disable
- Enable
- Delete

9.6.7.1.5.1 Login and Authenticate

Before any command, ensure you are logged in to the Delphix Continuous Data Engine using the admin user account.

```
ssh admin@<CONTINUOUS-DATA-FQDN-or-IP>
```

9.6.7.1.5.2 Creating and discovering the environment

Run the following commands to discover the environment.

```
Example -

environment
create
set hostEnvironment.name=rh86-my15-s1
set hostParameters.host.address=rh86-my15-s1.dlpxdc.co
set hostParameters.host.toolkitPath=/opt/toolkit
set primaryUser.name=mysql
set primaryUser.credential.password=mysql
commit
```

9.6.7.1.5.3 Updating the environment

Run the following commands to update the environment.

The name of the environment can be updated from `rh86-my15-s` to `rh86-pg15-s1.dlpxdc.co` using the following command.

```
Example -

environment
select rh86-my15-s1
update
set name=rh86-my15-s1.dlpxdc.co
commit
```

9.6.7.1.5.4 Refreshing the environment

Run the following commands to refresh the environment.

```
Example -

environment
select rh86-my15-s1.dlpxdc.co
refresh
commit
```

9.6.7.1.5.5 Disabling the environment

Run the following commands to disable the environment.

```
Example -

environment
select rh86-my15-s1.dlpxdc.co
disable
commit
```

9.6.7.1.5.6 Enabling the environment

Run the following commands to enable the environment.

```
Example -

environment
select rh86-my15-s1.dlpxdc.co
enable
commit
```

9.6.7.1.5.7 Deleting the environment

Run the following commands to delete the environment.

```
Example -

environment
select rh86-my15-s1.dlpxdc.co
delete
commit
```

9.6.7.2 Linking MySQL data sources

Delphix Continuous Data Engine provides several ingestion methods for linking and ingesting MySQL source databases.

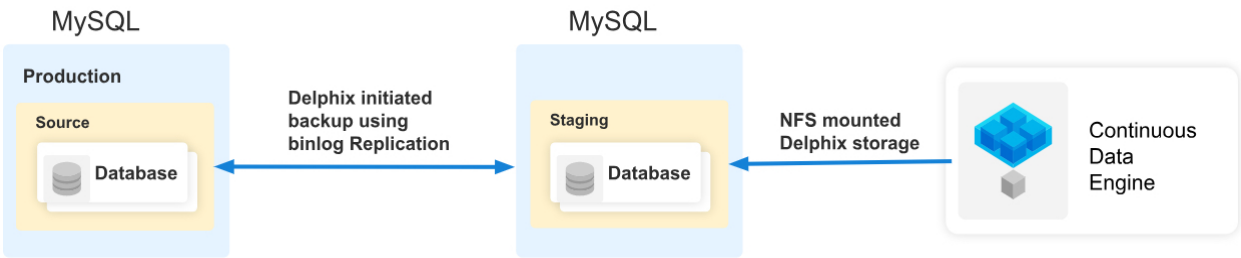
1. Replication with Delphix-Initiated Backups (binlog Replication) (see page 1301)
2. Replication with Externally-Initiated Backups Provided by the User (binlog Replication) (see page 1305)
3. Staging Push for MySQL (see page 1309)

9.6.7.2.1 Replication with Delphix-Initiated backups (binlog replication)

Delphix Continuous Data Engine creates a staging database using an initial backup from the source database in this ingestion method. The connector creates the initial backup by connecting to the source database. Delphix Continuous Data Engine configures MySQL native replication (binlog replication) from the source database to the staging database to ensure that the staging database remains in sync with the source database.

Binlog replication will automatically apply logs (as available from the source) to the staging database (in smaller increments). Then the next snapshot operation will snapshot the current state of the staging database up to the latest binlog that was applied. In Delphix-Initiated backup, a resync operation on the connector can be performed to take another full backup from the source database and restore it to the staging database.

The following diagram demonstrates the ingestion architecture of Delphix Continuous Data Engine's support for MySQL databases with replication between the source and staging databases.



9.6.7.2.1.1 Prerequisites

Before you begin database ingestion and creating a dSource, ensure that you meet the following requirements.

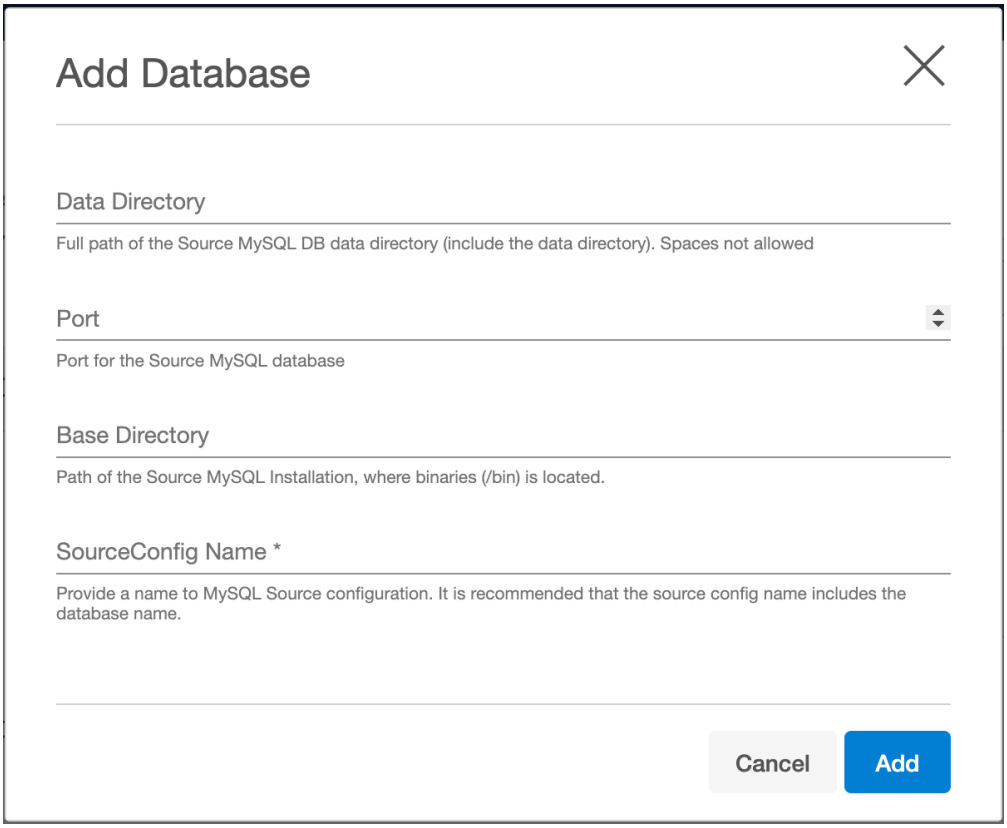
- The environment requirements are met by the source and staging environments as described in Requirements for MySQL source hosts (see page 1286) and Requirements for MySQL staging/target hosts (see page 1288)
- The environment user must have the necessary permissions to access the MySQL commands.
- A Database must be created on the staging environment's MySQL installation.

For more information, refer to Requirements and Prerequisites (see page 1282).

9.6.7.2.1.2 Procedure

Perform the following steps to create a dSource using the Delphix-initiated single database backup ingestion.

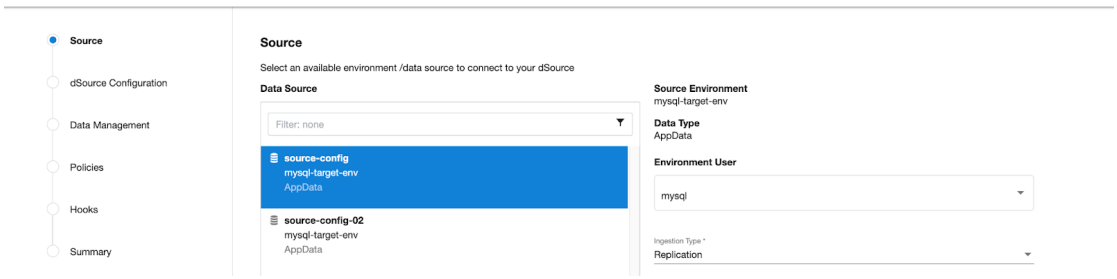
1. Login to the **Delphix Management** application.
2. Navigate to the **Manage > Environments** page and select the staging environment from the list of environments.
3. On the **Databases** tab, choose an installation for your dSource and click the plus icon.
4. In the **Add Database** dialog window, provide the Data Directory, Port, Base Directory of the source database, and SourceConfigName for your source configuration, and then click **Add**. It is necessary to provide values for all the fields for the **Replication** ingestion type.

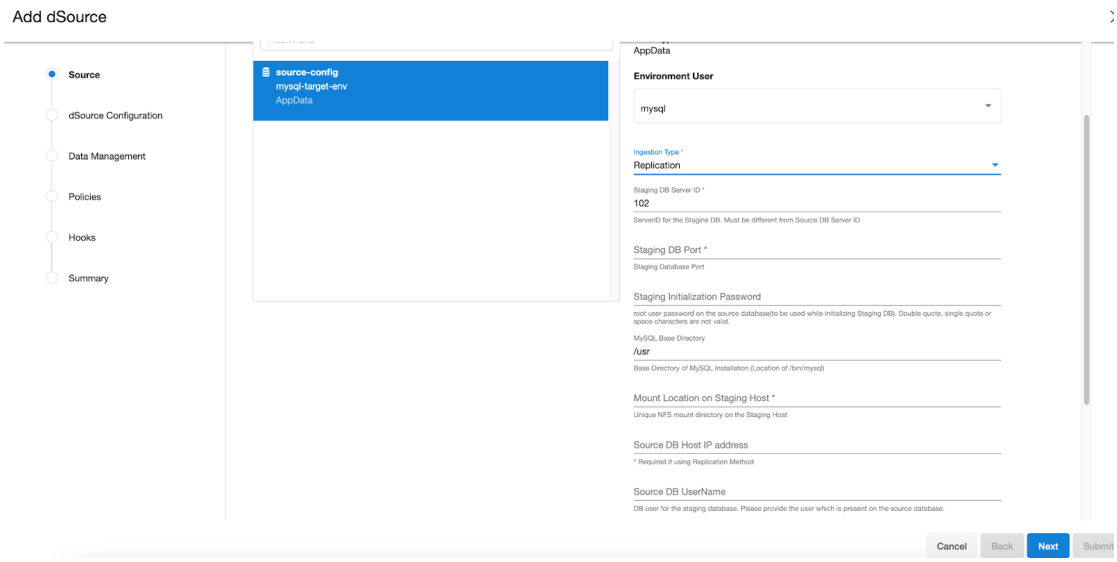


5. Select your source configuration and click the **Add dSource** link located on the right.

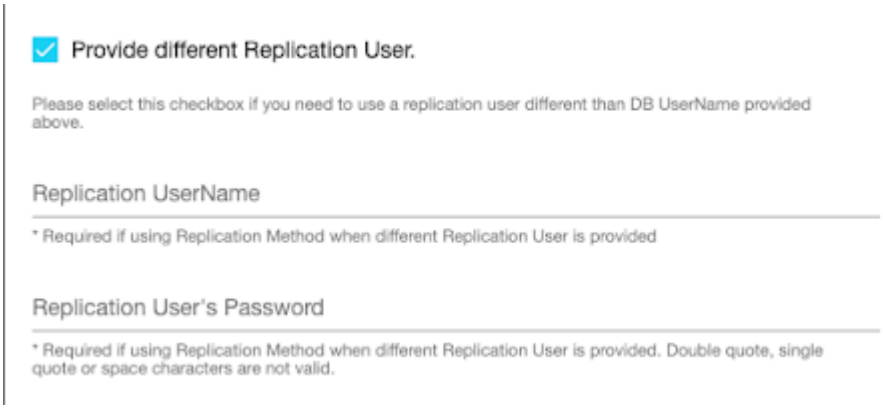
6. On the **Source** tab, the selected MySQL source configuration will appear as selected by default. Do the following:
   a. Select **Replication** from the **Ingestion Type** dropdown, it will show all the fields relevant to the Replication ingestion mechanism.

Add dSource





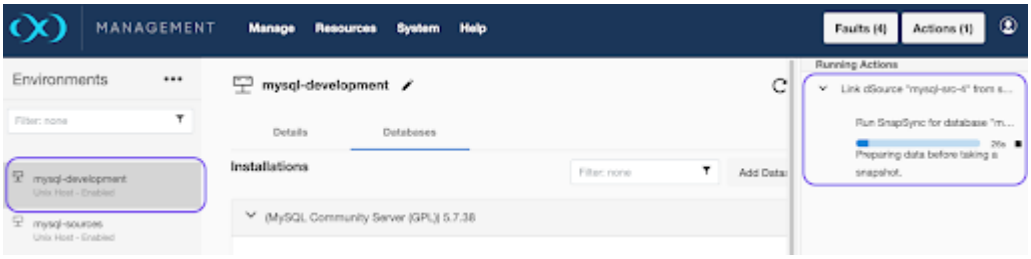
b. **Staging DB Server ID**: This is the Server ID for the dSource database. For Replication Mode, this server-id must be greater than the source database server id.
c. **Staging DB Port:** The port number used by the staging dSource instance. This port should not be used by any other application or MySQL server.
d. **Staging Initialization Password**: This is the Password that the connector uses when it initializes the staging database MySQL instance. This password will be assigned to the 'root' user of the staging database. This password will be used only for the first time the staging database instance is initialized. Note: Once replication starts, this password will be updated with Source DB's root user password.
e. **MySQL Base Directory**: Please provide the location of the MySQL installation directory. This generally has the location of /bin/mysql.
f. **Mount Location on Staging Host:** This is the Delphix mount directory on the staging host. This location should be unique and unoccupied. Additionally, refer to Requirements for MySQL staging/target hosts (see page 1288)to understand user privileges to create mount directories.
g. **Source DB Host IP Address**: IP Address or FQDN of the source host.
h. **Source DB UserName**: Delphix database user that was created on the source database (refer Requirements for MySQL source hosts (see page 1286)). When the initial backup is created or restored, this user is also created on the staging database. If Delphix Continuous Data Engine is initiating the backup generation of the source, Delphix Continuous Data Engine will connect to the source database using the Source DB UserName. This db user will be used by Delphix Continuous Data Engine to manage the dSource and sync operations.
i. **Source Connection Password**: Password for the above source database user.
j. **Full path (including filename) to MySQL Backup**: Do not enter anything in this field for this ingestion method. This field is meant to be filled only in case of Replication with Externally Initiated backup.
k. **Provide different Replication User**: The users can provide different Replication user credentials by selecting this checkbox, as illustrated in the image below. Providing separate replication user credentials is optional. If this checkbox is not selected, the source database user provided above will be used for replication.



l. **Replication UserName**: user with replication privileges to configure replication from source to staging db. If this user is not provided, then the Delphix Continuous Data Engine uses the Source DB user for replication, as mentioned in steps g and h.

m. **Replication User's Password**: The password for the Replication username.

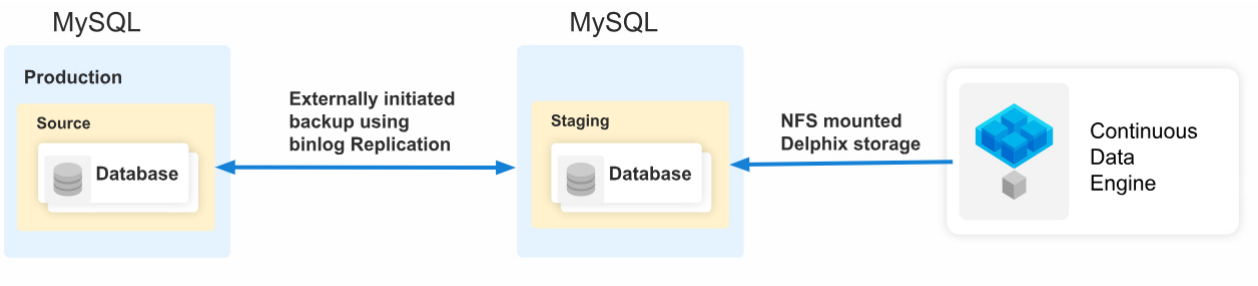
7. On the **dSource Configuration** screen, select the dataset group where the dSource will be placed and click **Next**.
8. On the **Data Management screen**, select the staging environment and the environment user and click **Next**.
9. On the **Policies screen**, select the snapsync and Retention policies for the dSource and click **Next**.
10. On the **Hooks screen**, add any pre-sync and (or) post-sync hooks as required and click **Next**.
11. Review the dSource configuration on the Summary screen and **Submit**. The Linking process has commenced. Wait for the snapshot to be created. From that snapshot, you can now create a new VDB (see page 1316).



9.6.7.2.2 Replication with Externally-Initiated backups provided by the user (binlog replication)

In this method, Delphix Continuous Data Engine creates a staging database using an initial backup from the source database. This initial backup is provided by you. To ensure that the staging database remains in sync with the source database, Delphix Continuous Data Engine sets up a MySQL native replication from the source database to the staging database.

The following diagram demonstrates the architecture of the Delphix Continuous Data Engine support for the MySQL database linking with the "MySQL BinLog Replication with Externally-initiated Backup" between the source and staging database.



9.6.7.2.2.1 Prerequisites

Before you begin ingestion and creating a dSource, ensure that you meet the following requirements.

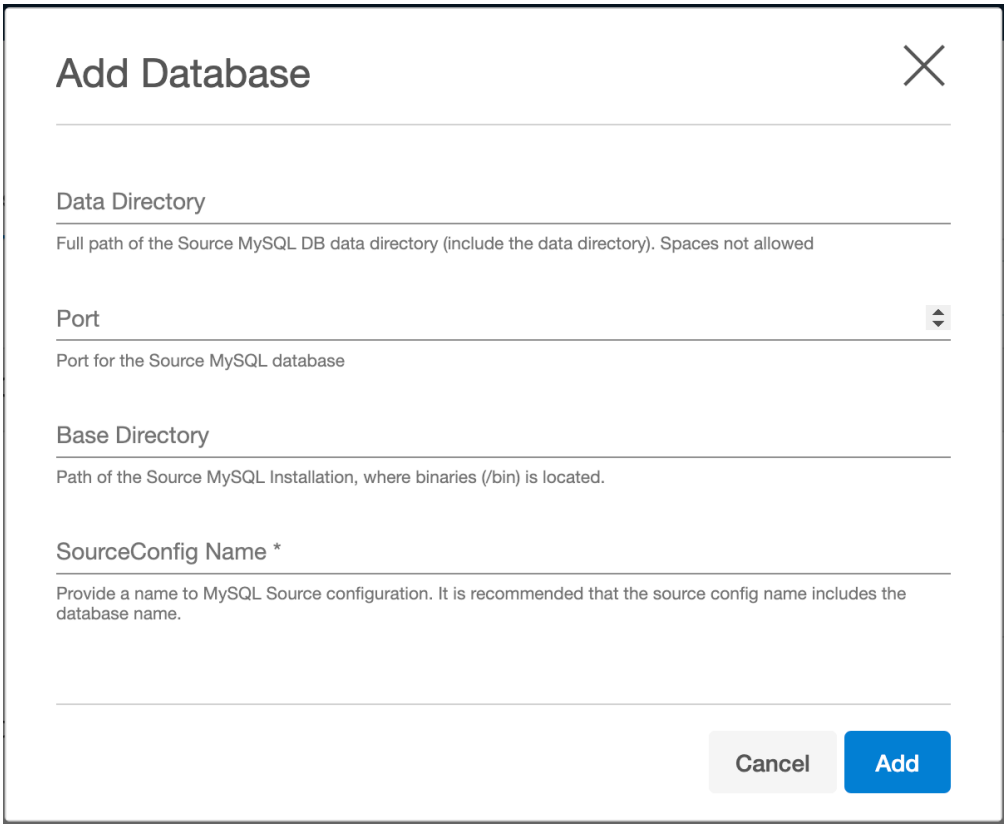
- The environment requirements are met by the source and staging environments as described in Requirements for MySQL source hosts (see page 1286) and Requirements for MySQL staging/target hosts (see page 1288).
- The environment user must have the necessary permissions to access the MySQL commands.
- A database must be created on the staging environment's MySQL installation.

For more information, refer to Requirements and Prerequisites (see page 1282).

9.6.7.2.2.2 Procedure

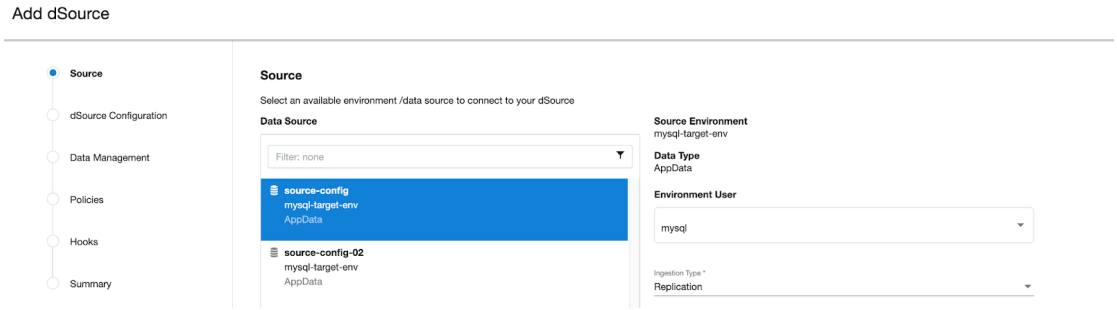
Perform the following steps to create a dSource using the Externally-initiated single database backup ingestion.

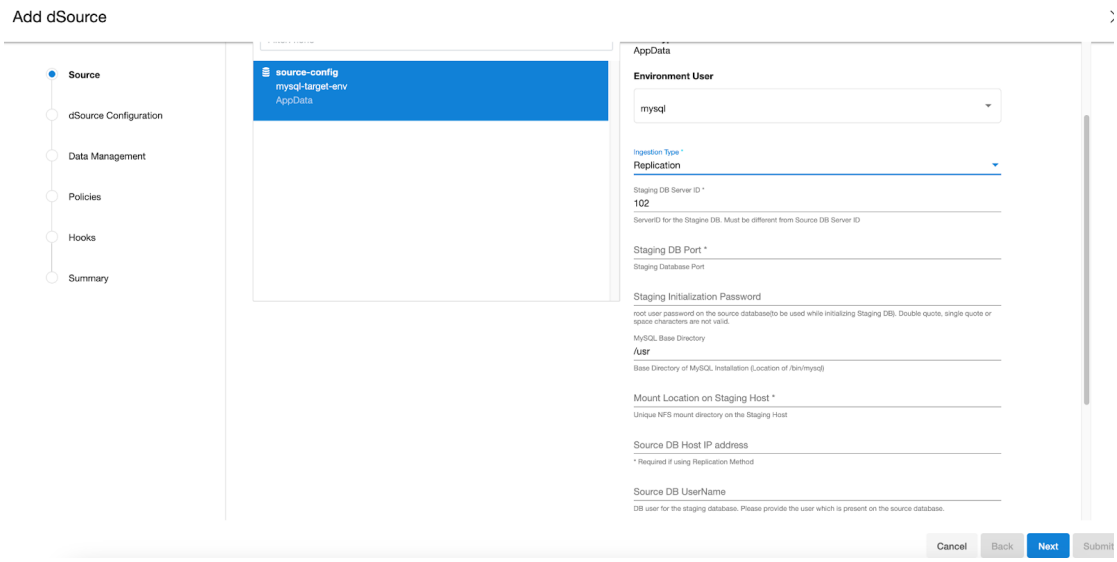
1. Login to the **Delphix Management** application.
2. Navigate to the **Manage > Environments** page and select the Staging environment from the list of environments.
3. On the **Databases** tab, choose an installation for your dSource and click the icon.
4. In the **Add Database** dialog window, provide the Data Directory, Port, Base Directory of the source database, and SourceConfigName for your source configuration, and then click **Add**. It is necessary to provide values for all the fields for the **Replication** ingestion type.



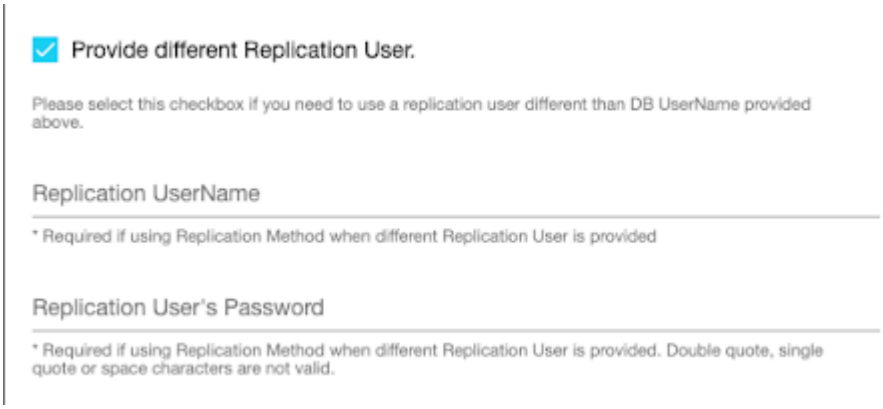
5. Select your source configuration and click the **Add dSource** link located on the right.

6. On the **Source** tab, the selected MySQL source configuration will appear as selected by default. Do the following:
   a. Select **Replication** from the **Ingestion Type** dropdown, it'll show all the fields relevant to the Replication ingestion mechanism.





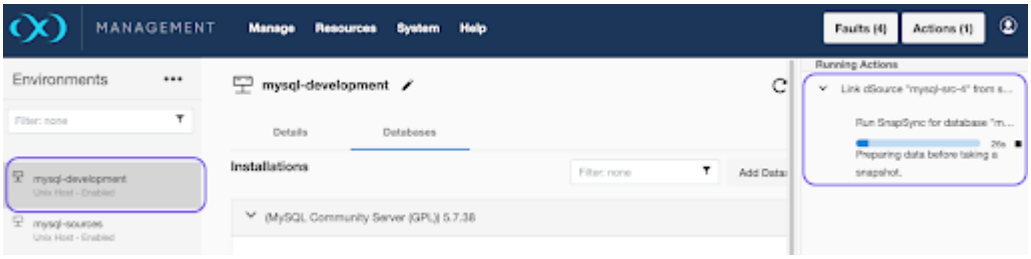
b. **Staging DB Server ID**: This is the Server ID for the dSource database. This server ID must be greater than the source db server ID for Replication Mode to work.
c. **Staging DB Port**: The port number to be used by the staging dSource instance. Any other application or MySQL server should not use this port.
d. **Staging Initialization Password:** This is the Password that the connector uses when it initializes the staging database MySQL instance. This password will be assigned to the 'root' user of the staging database. This password will be used only for the first time the staging database instance is initialized. Note: Once replication starts, this password will be updated with Source DB's root user password.
e. **MySQL Base Directory**: Please provide the location of the MySQL installation directory. This generally has the location of `/bin/mysql`.
f. **Mount Location on Staging Host:** This is the Delphix mount directory on the staging host. This location should be unique and unoccupied. Additionally, refer to Requirements for MySQL staging/target hosts (see page 1288)to understand user privileges to create mount directories.
g. **IP Address of the Source DB Host:** The IP address or FQDN of the source host.
h. **Source DB UserName**: Delphix Continuous Data Engine database user on the source database. Delphix Continuous Data Engine will use this db user to manage the dSource and other sync operations.
i. **Source DB Password**: The password for the source database user.
j. **Full path (including filename) to MySQL Backup**: Full path,including the filename, to the full source database backup to use for dSource creation.
k. **Provide different Replication users**: The users can provide different replication user credentials by selecting this checkbox, as illustrated in the image below. Providing separate replication user credentials is optional. If this checkbox is not selected, the source database user provided above will be used for replication.



l. **Replication UserName**: user with replication privileges to configure replication from source to staging db. If this user is not provided, then the Delphix Continuous Data Engine uses the Source DB user for replication, as mentioned in steps g and h.

m. **Replication User's Password**: The password for the Replication UserName.

7. On the **dSource Configuration screen**, select the dataset group where the dSource will be placed and click **Next**.
8. Select the staging environment and the environment user on the **Data Management screen**, then click **Next**.
9. Select the Snapsync and Retention policies for the dSource and click **Next** on the **Policies screen**.
10. Add any pre-sync and/or post-sync hooks needed on the **Hooks screen**, then click **Next**.
11. On the **Summary** screen, review the dSource configuration and click **Submit**. The linking process has begun. Wait for the snapshot to be created. From that snapshot, you can now create a new VDB (see page 1316).



9.6.7.2.3 Staging push for MySQL

The Staging Push transfers the responsibility of preparing the staging database to the end user. This ingestion method allows you to manually push the source data into the mount path backed by the Delphix Storage. It offers an alternative approach to the Staging Pull model where the Delphix Continuous Data Engine takes care of preparing the dSource.

9.6.7.2.3.1 Prerequisites

- A database must be created on the staging environment MySQL Dataset Home.

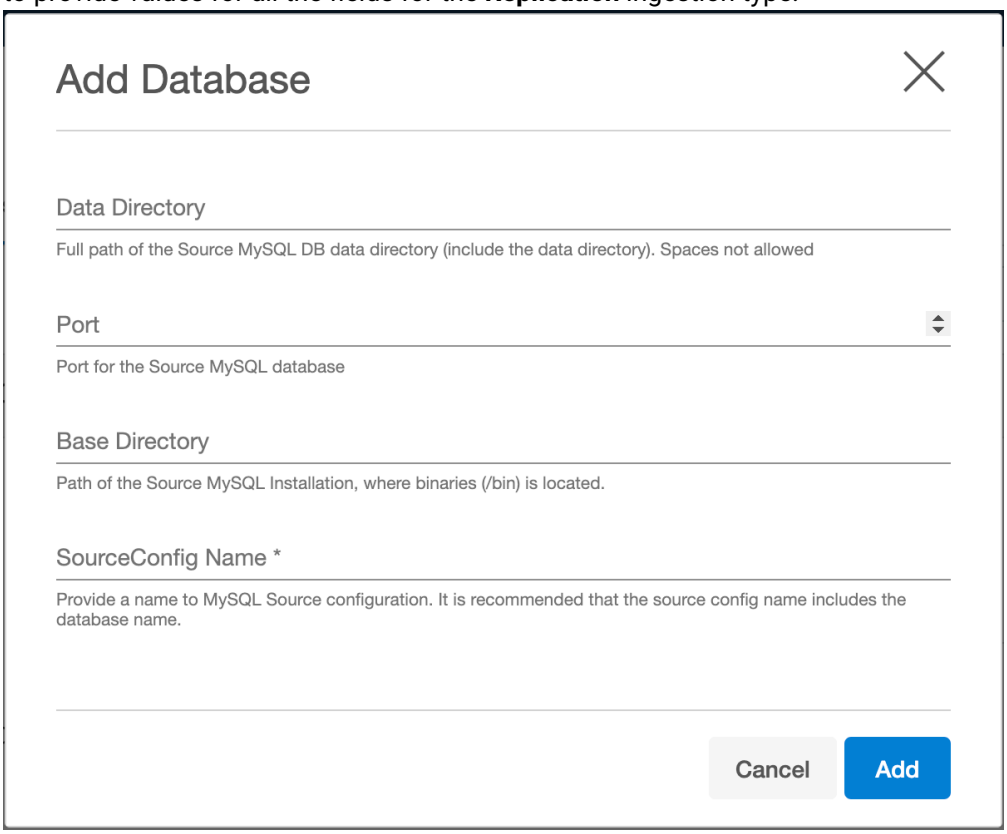
- The environment requirements are met by the source and staging environments as described in the Requirements for MySQL source Hosts (see page 1286) section and Requirements for MySQL staging/target hosts (see page 1288).

For more information, refer to Requirements and Prerequisites (see page 1282).

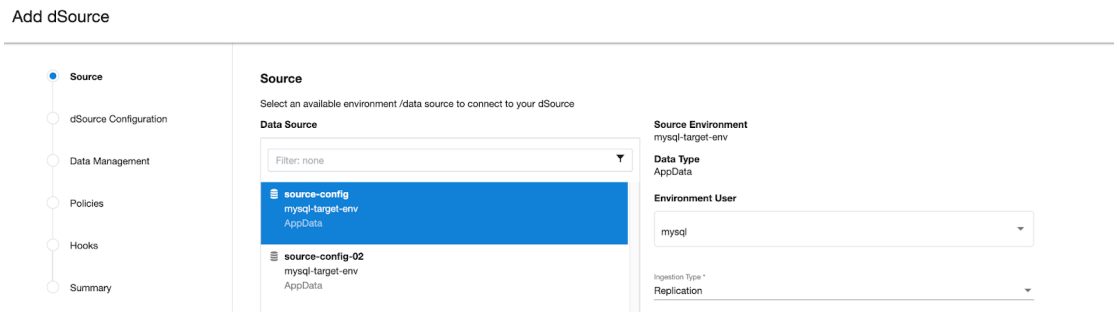
9.6.7.2.3.2 Procedure

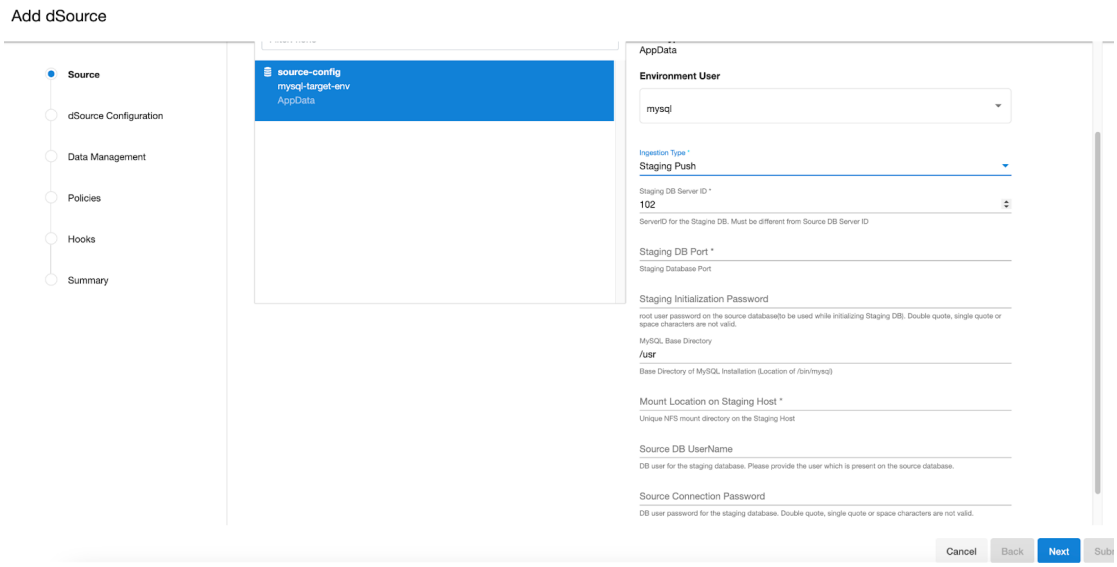
Perform the following steps to create a dSource using the staging push ingestion method.

1. Login to the **Delphix Management** application.
2. Navigate to the **Manage > Environments** page and Select the staging environment from the list of environments.
3. On the **Databases** tab, choose a repository for your dSource and click the icon.
4. In the **Add Database** dialog window, provide the Data Directory, Port, Base Directory of the source database, and SourceConfigName for your source configuration, and then click **Add**. It is necessary to provide values for all the fields for the **Replication** ingestion type.



5. Select your source configuration and click the **Add dSource** link located on the right.
6. On the **Source** tab, do the following:
   a. Select **Staging Push** from the Ingestion Type dropdown. The source screen will show fields/properties only relevant to the staging push.

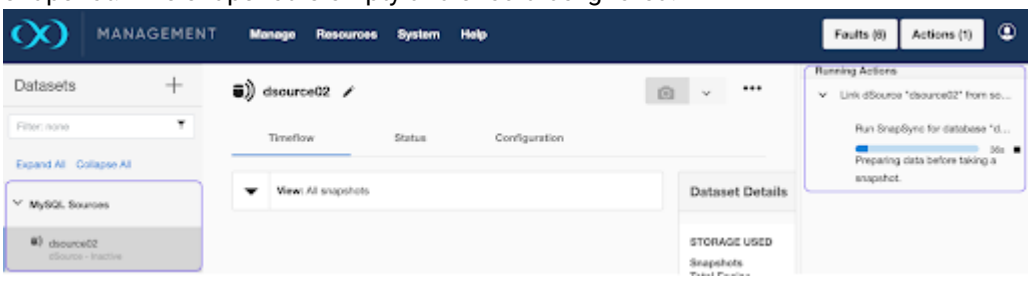




b. **Staging DB Server ID:** This is the Server ID for the dSource database. For Replication Mode, this server id must be greater than the source DB server id.
c. **Staging DB Port**: The port number used by the Staging dSource instance. This port should not be used by any other application or MySQL server.
d. **Staging Initialization Password**: This is the password used by the connector when initializing the staging MySQL database instance. This password will be assigned to the 'root' user.
e. **MySQL Base Directory**: Please provide the location of the MySQL installation directory. This generally has a location of `/bin/mysql`.
f. **Mount Location on Staging Host**: This is the mount directory for Delphix Continuous Data Engine on the staging host. This location should be unique and empty.
g. **Source DB UserName**: Delphix Delphix Continuous Data Engine will be using this database user to manage the staging database and other sync operations. The connector will also attempt to create this user and grant the necessary permissions on the staging database.
h. **Source Connection Password**: Staging database user's password.

7. On the **dSource Configuration screen**, select the dataset group where the dSource will be placed and click **Next**.
8. On the **Data Management** screen, select the staging environment and the environment user and click **Next**.
9. On the **Policies** screen, select the snapsync and retention policies for the dSource and click **Next**.

10. On the **Hooks** screen, add any pre-sync and (or) post-sync hooks as required and click **Next**.
11. Review the dSource configuration on the Summary screen and Submit. The Linking process has commenced. Once the dSource creation is successful, the Timeflow tab should show a new snapshot. This snapshot is empty and should be ignored.



12. Populate the staging database instance with your latest source data through any compatible mechanism. As explained above, in this phase, you are responsible for preparing the data in your staging database instance. The database instance can be populated manually or through automation scripts.
13. Once the staging database instance is in the desired state, navigate to the Delphix Continuous Data Engine UI and take a new snapshot. This second snapshot can then be used to provision virtual copies.

If you wish to populate Delphix Continuous Data Engine with a more recent dataset at a later time, repeat steps 12 and 13 to create a new snapshot.

9.6.7.2.4 MySQL Source sizing implementation for dSource

Source sizing is important to understand the overall storage requirements for your Delphix Continuous Data Engine. As more sources are ingested and their sizes grow, additional hardware might be required to maintain a performant engine. This feature, which supports the Managed Source Data report, will help you monitor the size of each ingested source, which is calculated from the dataset's total, uncompressed, allocated space, and then help you predict storage trends and requirements.

> This value is meant to help size your Delphix Continuous Data Engine, and should not be used when reporting Usage-Based Pricing Measurements to Delphix. Follow the Delphix Pricing Guide[406] for more information.

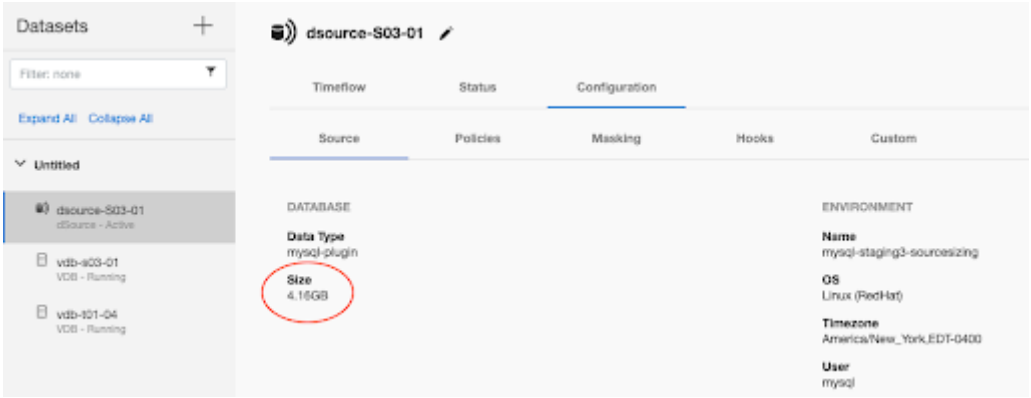
9.6.7.2.4.1 Procedure

The size of the dataset will be calculated using the data usage (du) command, which returns the volume and apparent size of the dataset.

406 https://www.delphix.com/measurement

```
du -sb <mount_point>/data
```

The apparent size is displayed in the Managed Source Data report and dSource configuration, as shown below:



9.6.7.2.5 MySQL link operations via CLI

9.6.7.2.5.1 dSource creation from CLI

To create a dSource via CLI, you must create a source configuration first. For more information on creating a source configuration, refer to Add Source Config using CLI[407].

To create a dSource and perform operations via CLI, log in to the Delphix Continuous Data Engine using the admin user account.

```
ssh admin@<CONTINUOUS-DATA-FQDN-or-IP>
```

9.6.7.2.5.2 Set default linking parameters

> This is common for all types of ingestions.

Run the following commands to set the default linking parameters. The following example uses the `testscfg` source config.

```
Example -
ip-10-110-238-142> database link
ip-10-110-238-142 database link *> set linkData.type=AppDataStagedLinkData
```

407 https://cd.delphix.com/docs/latest/dsource-operations#id-(20.0.0.0)dSourceoperations-SettingupanenvironmentconfigurationusingtheCommandLineInterface(CLI)

```
ip-10-110-238-142 database link *> set name=testds
ip-10-110-238-142 database link *> set group=Untitled
ip-10-110-238-142 database link *> set linkData.config=testscfg
ip-10-110-238-142 database link *> set linkData.environmentUser=mysql-tgt.dlpxdc.co/
mysql
ip-10-110-238-142 database link *> set linkData.stagingEnvironment=mysql-
tgt.dlpxdc.co
ip-10-110-238-142 database link *> set linkData.stagingEnvironmentUser=mysql-
tgt.dlpxdc.co/mysql

ip-10-110-238-142 database link *> set
linkData.syncParameters.parameters={\"resync\":true}
```

9.6.7.2.5.3 Link parameters for replication with Delphix-initiated backup

```
Example -

set linkData.parameters="{\"dSourceType\":\"Replication\",\"backupPath\":\"\",
\"sourceip\":\"ip_address\",\"mountPath\":\"/mnt/mount_path\",\"sourceUser\":
\"delphix_db\",\"sourcePass\":\"password_string\",\"stagingPass\":
\"password_string\",\"stagingPort\":4009,\"serverId\":409,\"stagingBasedir\":\"/
usr\",\"replicationUserFlag\":false}"
```

For **Replication** ingestion mechanism, you can also provide different replication user credentials. In this case, the parameters can be set like this :

```
set linkData.parameters="{\"dSourceType\":\"Replication\",\"backupPath\":\"\",
\"sourceip\":\"ip_address\",\"mountPath\":\"/mnt/mount_path\",\"sourceUser\":
\"delphix_db\",\"sourcePass\":\"password_string\",\"stagingPass\":
\"password_string\",\"stagingPort\":4009,\"serverId\":409,\"stagingBasedir\":\"/
usr\",\"replicationUserFlag\":true, \"replicationUser\":\"replication_username\",
\"replicationPass\":\"replication_password\"}"
```

9.6.7.2.5.4 Link parameters for replication with externally-initiated backup

```
Example -

set linkData.parameters="{\"dSourceType\":\"Replication\",\"backupPath\":\"/var/
backupfile\",\"sourceip\":\"ip_address\",\"mountPath\":\"/mnt/mount_path\",
\"sourceUser\":\"delphix_os\",\"sourcePass\":\"password_string\",\"stagingPass\":
\"password_string\",\"stagingPort\":4005,\"serverId\":405,\"stagingBasedir\":\"/
usr\",\"replicationUserFlag\":false,}"
```

```
ip-10-110-238-142 database link *> set
linkData.syncParameters.parameters={\"resync\":true}
```

9.6.7.2.5.5 Link parameters for staging push

Run the following commands to link parameters for staging push.

```
Example -

set linkData.parameters="{\"dSourceType\":\"Staging Push\",\"mountPath\":\"/mnt/
mount_path\",\"sourceUser\":\"delphix_db\",\"sourcePass\":\"password_string\",
\"stagingPass\":\"password_string\",
\"stagingPort\":4063,\"serverId\":463,\"stagingBasedir\":\"/usr\"}"
```

9.6.7.2.5.6 Commit the change

Run the following command to commit the change.

```
commit;
```

9.6.7.2.5.7 Enable

Run the following command to enable the dSource.

```
Example -

source
select testdb
enable
commit
```

9.6.7.2.5.8 Disable

Run the following command to disable the dSource.

```
Example -

source
select testdb
disable
commit
```

9.6.7.3 Provisioning MySQL virtual databases

The virtual database (VDB) is a Delphix Continuous Data Engine virtualized copy of your source database that runs on your target environment. A VDB is provisioned from a dSource or a VDB snapshot.

9.6.7.3.1 Prerequisites

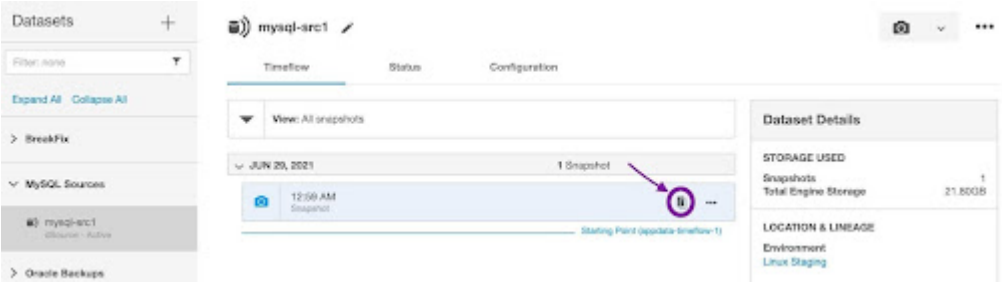
To create a VDB, you must meet the following requirements:

1. Identify an originating dataset and snapshot.
   a. Link a dSource by following the instructions in Linking MySQL data sources (see page 1301).
   b. Select a previously provisioned VDB.
2. Set up a MySQL target environment as described in Requirements for MySQL staging and target environments (see page 2145).
3. A MySQL binary with the same version as the source database must be installed on the target environment.

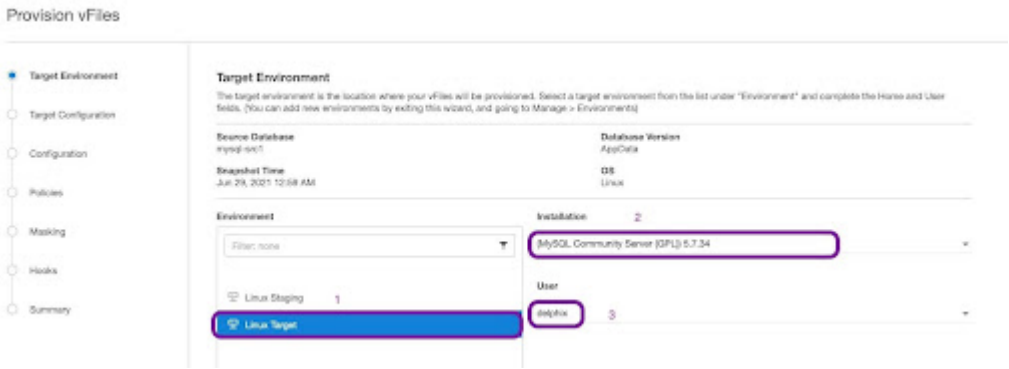
9.6.7.3.2 Provisioning a MySQL VDB

Perform the following steps to provision a MySQL VDB.

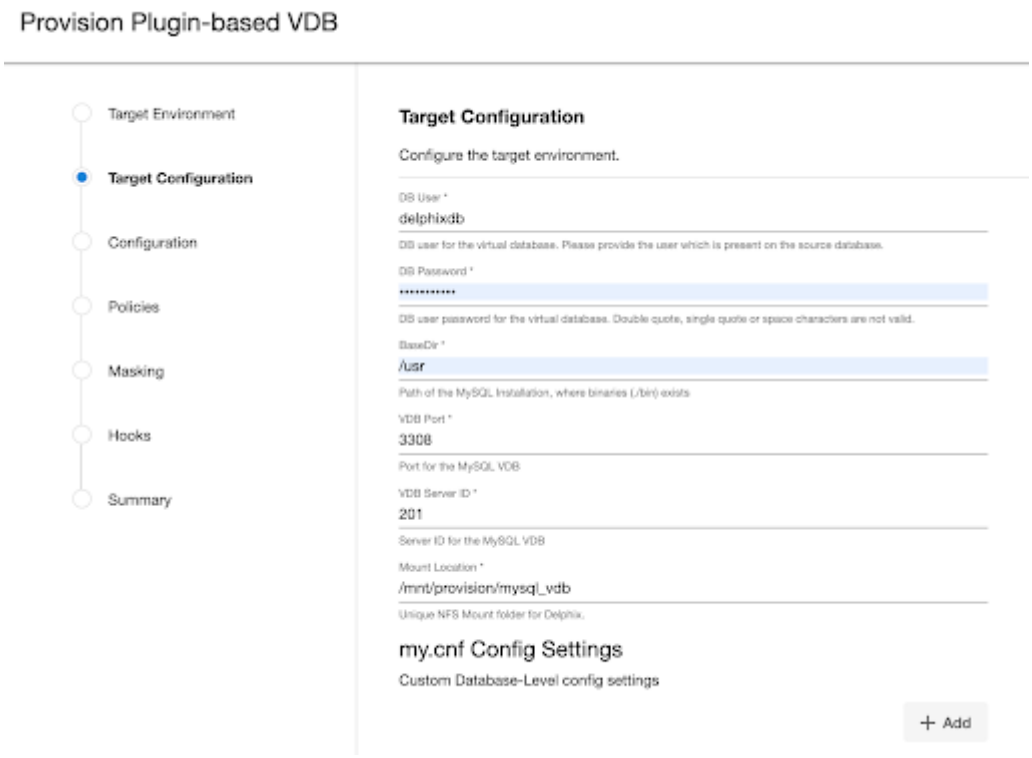
1. Login to the **Delphix Management** application.
2. Navigate to the **Manage > Datasets** page and select a dSource or VDB dataset.
3. Under the Timeflow tab, identify a snapshot, hover over it, and select the **Provision a VDB** button**.**



4. On the **Target Environment** page, perform the following actions:
   - From the **Environment** dropdown list, select a compatible target environment. This will dynamically populate the other properties.
   - From the **Installation** dropdown list, select the specific MySQL instances that you want to use.
     - The VDB target environment's MySQL version should be the same as the dSource environment's MySQL version. Otherwise, you will receive an error message. Refer to the Troubleshooting (see page 1327) section for more information.
   - From the **User** dropdown list, select the environment user that matches the MySQL Instance Owner as configured in the prerequisites.

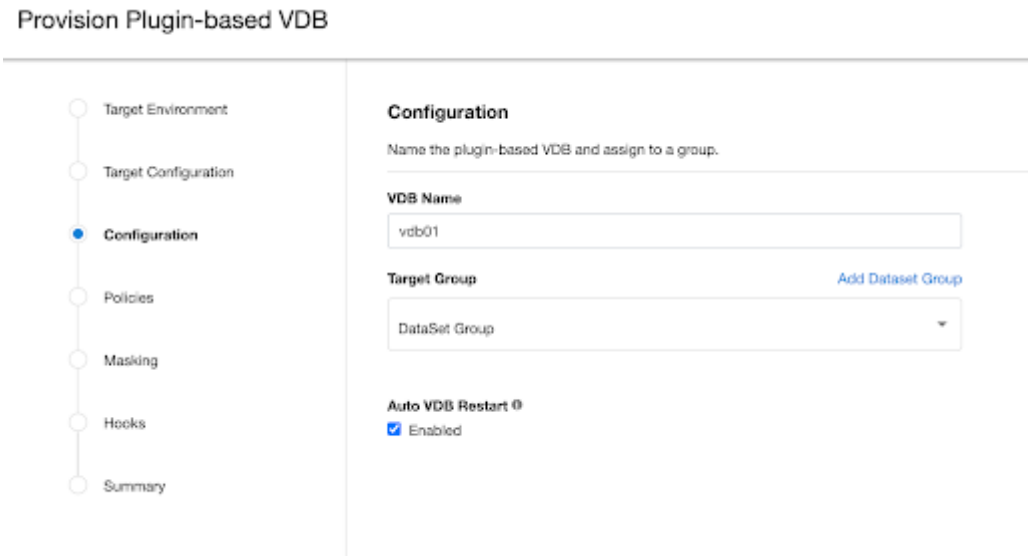


5. On the **Target Configuration** page, fill in the following values:
   - **DB User:** This is the database user for the VDB. Delphix Continuous Data Engine will manage the VDB using this user. You must provide the same user account used by the staging database.
   - **DB Password:** This is the password for the VDB database user.
   - **BaseDir:** This is for the MySQL installation directory.
   - **VDB Port:** Port number to be used by the VDB. This port should not be used by any other application or MySQL server.
   - **VDB Server ID:** MySQL server ID for the VDB. This server-id must be non-zero.
   - **Mount Location:** This is the mount directory for Delphix VDB on the Target host. This location should be unique and empty. On Linux and Unix hosts, this mount path must be the full path and not include symlinks.
   - **(Optional) my.cnf Config Settings:** This allows users to specify MySQL configuration properties while creating a VDB.



MySQL plugin does not allow VDBs to inherit configuration properties from their parent dsource.

6. On the **Configuration** screen, add the VDB name, select the **Target Group**, check the **Auto vFiles Restart** checkbox, and click **Next**.

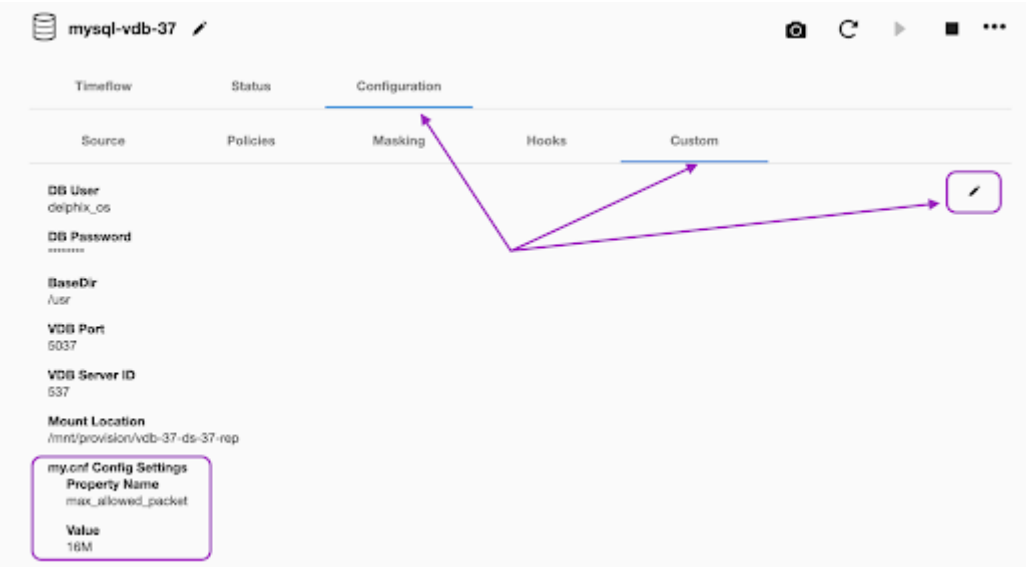


7. (Optional) On the **Policies** page, select a previously defined **VDB Snapshot Policy** or the **Use group policy** checkbox.
8. (Optional) On the **Masking** page, enable the **Mask this VDB** checkbox to mask the database following a successful provision.
    - This feature may be disabled or unavailable if the chosen snapshot is already masked or the Delphix Continuous Compliance Engine is disconnected.
9. (Optional) On the **Hooks** page, select a Hook Point and then click the plus button (**+**) to add a script that you want to run. You have the flexibility to define scripts to run at multiple hook points in the process.
10. In the **Summary** tab, review the configuration profile for your VDB.
11. Click **Submit**.

Once the VDB is created successfully, you can review the datasets on the **Manage** > **Datasets** screen in the left-hand panel. You have now provisioned a MySQL VDB. You can connect a running application and share it with your end users. Connection information can be found in the VDB's Configuration Custom tab and within its target environment's database.

9.6.7.3.2.1 Updating custom configuration properties

MySQL connector UI allows you to add/update/delete custom configuration properties after provisioning a VDB. Go to **Configuration** > **Custom** and click the pencil icon to edit the fields. You can update parameters under **my.cnf Config Settings** section, as shown in the picture.



9.6.7.3.3 MySQL provision operations via CLI

To create a VDB from CLI you must create a dSource first. Instructions on how to create dSource from CLI, refer to MySQL link operation via CLI section (see page 1313).

To create VDB and perform the VDB operations from CLI, you first need to log in to the Delphix Continuous Data Engine with the admin user.

```
ssh admin@<CONTINUOUS-DATA-FQDN-or-IP>
```

9.6.7.3.3.1 Provision

Here we are using dSource `testdb`.

```
Example -

database provision
set type=AppDataProvisionParameters
set container.name=testvdb
set container.group=Untitled
set source.allowAutoVDBRestartOnHostReboot=true
set sourceConfig.repository="(MySQL Community Server (GPL)) 5.7.38"
set timeflowPointParameters.container=testdb
set sourceConfig.name=testscfgv
set source.parameters="{\"vdbUser\":\"root\",\"vdbPass\":\"Password_String\",
\"port\":3310,\"baseDir\":\"/usr\",\"serverId\":202,\"mPath\":\"/mnt/vdb_mount\",
\"config_settings_prov\":[]}"
commit
```

9.6.7.3.3.2 Enable

```
Example -

source
select testscfgv
enable
commit
```

9.6.7.3.3.3 Disable

To disable VDB, select the sourceConfig name given during VDB creation as above.

```
Example -

source
select testscfgv
disable
commit
```

9.6.7.3.3.4 Stop

```
Example -

source
select testscfgv
stop
commit
```

9.6.7.3.3.5 Start

```
Example -

source
select testscfgv
start
commit
```

9.6.7.4 Upgrading dSources and VDBs MySQL database versions

The following document describes the general methodology of upgrading your MySQL database versions across all environments, dSources, and VDBs. The “In-Place” upgrade method is the only recommended MySQL upgrade approach. This process replaces the old MySQL installation on the target environment with the new version, using the same data directories as the previous version.

Perform the MySQL upgrade in the following order:

1. Target environments and their VDBs
2. Staging (Replica) and source environments

9.6.7.4.1 System requirements

Review the following requirements before performing any environment upgrade steps.

- Plan to install compatible versions on the staging and target environments. To know the compatible versions, see the MySQL connector source and target compatibility section in MySQL support matrix (see page 1282).
- Install MySQL Select Connector v4.1.0[408]or greater.

408 https://download.delphix.com/folder

9.6.7.4.2 Prerequisites

You can follow the below prerequisites before performing any activity as per this documentation.

- Review and familiarize with the MySQL upgrade guidelines[409]and the upgrade prerequisites[410].
  - Delphix recommends validating these steps in a separate test environment before following the upgrade steps below.

- The MySQL upgrade from v5.7 to v8 introduced many incompatibility issues. We strongly recommend you install and run the MySQL Shell: Upgrade Checker Utility[411] to identify those problems within all Source, Staging, Target, and VDB databases you plan to upgrade. Upgrading MySQL without fixing these issues can cause database initialization failure which might be unrecoverable and might need a rollback .
- Following a successful MySQL platform upgrade, you will need to take precautions against dSource and VDB corruption. Unlike other database types, MySQL staging and target environments are not backward compatible. We recommend you avoid the following scenarios:
  - Refresh or rollback a VDB to a snapshot with a lower version [ie 5.7] after an upgrade to a higher version [ie 8.0] with incompatible data as defined by the upgrade checker utility.
  - Enable a dSource or VDB when the latest snapshot of the dSource or VDB's MySQL contains incompatible data as defined by the upgrade checker utility.

For the above scenario, follow troubleshooting guidelines in below section to analyze and correct the VDB or dSource.

9.6.7.4.3 How to upgrade MySQL target environments and their VDBs

These directions outline how you can preserve the MySQL target environment and VDBs during an upgrade.

The target environment and VDBs must match the originating dSource to maintain full refresh and provision functionality. Therefore, it should be expected that you may experience a temporary loss of the provision and refresh features. During this time, VDBs can be enabled and running, however, no refreshes should be performed until the following section is followed and the dSource version matches.

409 https://dev.mysql.com/doc/refman/8.0/en/upgrading.html
410 https://dev.mysql.com/doc/refman/8.0/en/upgrade-prerequisites.html
411 https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-utilities-upgrade.html

> Delphix do not support an automated way to upgrade VDBs. In general, Delphix recommends deleting and re-provisioning each VDB from a new snapshot. However, if you wish to preserve certain VDBs, follow the steps below. If you would rather not modify this target environment and its VDBs, you may also create a completely new target environment as well.

9.6.7.4.3.1 Procedure

1. Select the target environment that you plan to upgrade and its set of VDBs.
2. Identify all VDBs that you wish to preserve and **delete** all VDBs that you do not.
3. Run MySQL Utility Checker[412]on the remaining VDBs. This utility will identify incompatibility issues that can occur while upgrading from version 5.7 to 8.0.
4. For each VDB, solve the compatibility issues reported by MySQL Utility Checker.
5. Take a backup of the entire database.
6. **Take a new snapshot** of each VDB in case errors occur.
7. **Remove** any VDB Refresh Policy assigned to the VDB.
8. **Stop** all VDBs.
   - **Caution:** Do use the Disable VDB action as it will remove the VDB from the target environment. Therefore, it cannot be upgraded.
9. By following the official MySQL in-place upgrade guidelines, perform an upgrade of MySQL installation on the Target environment.
10. **Refresh** the target environment on the GUI.
11. **Start** all VDBs.
12. **Take a new snapshot** of each VDB to prevent unsupported rollbacks.
    - a. Delphix recommends that all prior snapshots are deleted as well to eliminate the chance of accidental rollback to a prior version.
13. Validate the upgraded VDBs.

At this point, the target environment should be prepared to provision new VDBs from a matching dSource, and VDBs should be upgraded. Follow the next section to re-enable VDB refreshes.

9.6.7.4.4 How to upgrade MySQL source and staging environments

Follow the steps below to upgrade dSources using the “In-Place” method. Each set of steps is organized based on the ingestion method.

412 https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-utilities-upgrade.html#mysql-shell-utilities-upgrade-run

> If the issues reported by the MySQL Utility checker is not fixed for a dSource and upgrade is performed on the staging environment then the dSource will not start. To resolve this please check issue 2 in the troubleshooting section.

9.6.7.4.4.1 Staging Push

1. Run MySQL Utility Checker[413]on all dSource on the source and staging instances to find all the incompatibility issues that can occur while upgrading from version 5.7 to 8.0.
2. Solve all the compatibility issues reported by MySQL Utility Checker.
3. Take a backup of the entire database.
4. **Create** a new snapshot on an existing dSource.
5. **Disable** the dSource(s) on the staging instance and MySQL database on the source instance
6. By following the official MySQL in-place upgrade guidelines, upgrade the MySQL staging instance.
7. Perform an upgrade of the source MySQL instance as per the official MySQL in-place upgrade guidelines.
8. **Refresh** the staging environments in the GUI.
9. **Enable** the dSource(s) on the staging instance and also start the MySQL database on the source instance.
10. **Take a new snapshot** on an existing dSource.
11. **Validate** the dSource.

9.6.7.4.4.2 Replication with Delphix-Initiated backups (binlog replication)

1. Run MySQL Utility Checker[414]on all dSources on the instance to find all the incompatibility issues that can occur while upgrading from version 5.7 to 8.0.
2. Solve all the compatibility issues reported by MySQL Utility Checker.
3. Take a backup of the entire database.
4. **Create** a new snapshot on an existing dSource.
5. **Disable** the dSource(s) on the staging instance and MySQL database on the source instance
6. By following the official MySQL in-place upgrade guidelines, perform an upgrade of the replica instances. so here it is a staging instance.
7. By following the official MySQL in-place upgrade guidelines, perform an upgrade of the MySQL source instance.

---

413 https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-utilities-upgrade.html#mysql-shell-utilities-upgrade-run
414 https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-utilities-upgrade.html#mysql-shell-utilities-upgrade-run

8. **Refresh** the staging and source environments in the Delphix Continuous Data Engine UI.
9. **Enable** the dSource(s) on the staging instance and also start the MySQL database on the source instance
10. **Take a new snapshot** on an existing dSource.
11. **Validate** the dSource.

9.6.7.4.4.3 Replication with externally-initiated backups provided by the user (binlog replication)

1. Run MySQL Utility Checker[415]on all dSources on the instance to find all the incompatibility issues that can occur while upgrading from version 5.7 to 8.0.
2. Solve all the compatibility issues reported by MySQL Utility Checker.
3. Take a backup of the entire database.
4. **Create** a new snapshot on an existing dSource.
5. **Disable** the dSource(s) on the staging instance and MySQL database on the source instance
6. By following the official MySQL in-place upgrade guidelines, perform an upgrade of the replica instances. So here it is the staging instance.
7. By following the official MySQL in-place upgrade guidelines, perform an upgrade of the MySQL Source instance.
8. **Take a new Backup** of the source database, and place it in the staging environment.
9. **Refresh** the staging and source environments in the GUI.
10. **Edit** the backup file location in the staging dSource configuration on Delphix Continuous Data Engine UI or **replace** the backup file on the staging environment.
11. **Enable** the dSource(s) on the staging instance and also start the MySQL database on the source instance
12. **Take a new snapshot** on an existing dSource.
13. **Validate** the dSource

9.6.7.4.5 Troubleshooting

- **Issue 1:** After starting/enabling or making a new snapshot, VDB/dSource fails to start.
  Solution: This occurs after an upgrade if you refresh or roll back the VDB to an older version of a snapshot or currently pointing to the latest snapshot which contains incompatible data. To resolve this, refresh or roll back to the correct version of the snapshot that was taken immediately after the upgrade.
  The logs in MySQL 8.0 will show that the database initialization failure occurred and because of this, the database was not able to start. An example of such an error is as below. Here there was a table named `catalogs` in MySQL schema of MySQL 5.7 database which is a default table in MySQL 8.0.

415 https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-utilities-upgrade.html#mysql-shell-utilities-upgrade-run

```
023-07-26T15:02:39.477811Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld
(mysqld 8.0.34) starting as process 15891 2023-07-26T15:02:39.513413Z 1
 [System] [MY-011012] [Server] Starting upgrade of data directory. 2023-07-26T1
5:02:39.513453Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has
started. 2023-07-26T15:02:40.265033Z 1 [System] [MY-013577] [InnoDB] InnoDB
initialization has ended. 2023-07-26T15:02:40.270816Z 1 [ERROR] [MY-010781]
[Server] Found ./mysql/catalogs.frm file in mysql schema. DD will create .ibd
file with same name. Please rename table and start upgrade process again. 2023-0
7-26T15:02:40.270838Z 1 [ERROR] [MY-010336] [Server] Found .frm file with same
name as one of the Dictionary Tables. 2023-07-26T15:02:40.271004Z 0 [ERROR]
[MY-010020] [Server] Data Dictionary initialization failed. 2023-07-26T15:02:40.
271028Z 0 [ERROR] [MY-013236] [Server] The designated data directory /temp/
vdbtarget/data/ is unusable. You can remove all files that the server added to
it. 2023-07-26T15:02:40.271039Z 0 [ERROR] [MY-010119] [Server] Aborting 2023-07-
26T15:02:40.819837Z 0 [System] [MY-010910] [Server] /usr/sbin/mysqld: Shutdown
complete (mysqld 8.0.34) MySQL Community Server - GPL. 2023-07-26T15:02:40.8206
69Z 0 [ERROR] [MY-010065] [Server] Failed to shutdown components
infrastructure.
```

- **Issue 2:** dSource is not enabling after an upgrade
  Solution: When the dSource is started after an upgrade below error is shown on the UI of the Delphix Continuous Data Engine.

```
Traceback (most recent call last): File "plugin/platform_server.py", line 66,
in _run_helper response = internal_func(request) File "./_linked.py", line 517,
in _internal_start_staging File "./plugin_runner.py", line 75, in start_staging
File "./pluginops/pluginops.py", line 172, in start_staging File "./libs.py",
line 174, in run_bash File "./libs.py", line 91, in _check_exit_code
dlpx.virtualization.libs.exceptions.PluginScriptError: The script failed with
exit code
```

Here the error depicts that dSource was not able to start. One of the reasons why the dSource won't start after an upgrade is it might have some incompatible data that does not work in MySQL 8.0. Commonly, the issues reported by MySQL Upgrade Checker were ignored and the upgrade was performed. Please follow the below steps if you encounter any error as above.

1. Uninstall MySQL version 8.0
   Note: Please make sure you don't delete your data directory
2. By following the official MySQL guidelines install MySQL version 5.7
3. Enable the dSource and Refresh the environment
4. Connect to the dSource and fix the incompatibilities
5. Take a new snapshot on the dSource
6. Disable the dSource on the GUI
7. Again upgrade the staging to MySQL version 8.0
8. Refresh the staging environment in the GUI.

9. Start the dSource (dSource should be able to start since there is no incompatible data in the snapshot) on the GUI.
10. Take a new snapshot (This snapshot will be of version 8.0).

9.6.8 MySQL troubleshooting

This troubleshooting section describes the common pitfalls and errors around configuring, linking, and provisioning MySQL datasets and provides possible solutions. This section explains the following in detail:

- Connector logs and the path at which these logs are generated. These logs contain information regarding the flow of events, failure regarding any MySQL command failures, trace messages, etc.
- Common MySQL commands. These commands will help you understand how to check replication status, user permissions, etc
- Exit Codes and actions that users can take in case of any error.
- Version compatibility. This guides the user to upgrade from MySQL version 5.7 to 8.

If the user is still not able to resolve the error, please reach out to Delphix Support[416].

9.6.8.1 Logs

There are 2 sets of logs for the MySQL plugin:

1. **Plugin Logs:** Plugin logs are part of the Delphix Continuous Data Engine. Refer to How to Retrieve Logs[417] to find out how to get the Delphix Continuous Data Engine Plugin Logs.
2. **MySQL Shell Operation Logs:** When the plugin performs certain operations (such as Take backup, Link, Provision, etc.), logs are written to log files on the staging or target host under the Delphix Continuous Data Engine Toolkit directory.

Following are the logs to review:

- `delphix_mysql_user_mysql_debug.log`
- `delphix_mysql_user_mysql_info.log`
- `delphix_mysql_user_mysql_error.log`

9.6.8.2 Common MySQL commands

- Connecting to a MySQL instance

```
delphixos> /usr/bin/mysql -u UserName -p Password --protocol=TCP --port=1234
```

- List databases in MySQL

416 https://support.delphix.com/
417 https://developer.delphix.com/References/Logging/#how-to-retrieve-logs

```
mysql> show databases;
```

- Check the replication status in MySQL

```
mysql> show replica status;
```

- Check binary logging

```
mysql> show variables like 'log_bin';
```

- Check user permissions

```
mysql> show grants for 'delphixdb'@'%';
```

9.6.8.3 MySQL Plugin exit codes

The below tables include exit codes that the MySQL connector (plugin) presents if an error occurs.

Exit Code	Description	Possible Reason
1	General Error	Check logs
2	General Error	Check logs
3	Unable to start MySQL	1. Delphix user does not have the necessary permissions on the host 2. Mount location provided is in use 3. Database credentials provided are not accurate 4. Invalid server-id 5. Provided port is not in use
4	Missing config file	A `my.cnf` file was not provided and Delphix was unable to create one.
5	Unable to change `root` password	Delphix was unable to change the password for the root user after creating the staging db. Please check logs.

Exit Code	Description	Possible Reason
6	Unable to restore backup	Delphix was unable to restore the full backup into staging db. Please check logs.
7	Connect failure post restore	Delphix is unable to connect to the staging database after backup is restored. 1. The Source DB username and password provided may be incorrect. 2. The MySQL Database did not restart after the restore. Check logs for more information on the error.
8	Delphix backup failure	Delphix could not take a source database backup. 1. Staging Host may not be able to connect to Source Host 2. Source DB credentials may be incorrect 3. Source DB user may not have the required permissions 4. Databases provided in the list may not be present on the source MySQL instance. Check logs for more information on the error.
9	Customer backup failure	There was an issue with the customer provided backup. 1. Backup location may not exist. 2. The backup file may not exist 3. Backup file may be empty. Check logs for more information on the error.
10	Invalid binary path	`mysql` was not found under the provided installation directory.
11	Connect failure after restart	In Staging Push Mode, the Delphix source user could not connect to the MySQL instance after reboot. Check logs for more information on the error.

Exit Code	Description	Possible Reason
12	Unable to create user	In Staging Push Mode(Staging Push), there was an error while creating the delphix user after staging db was initialized. Check logs for more information on the error.
13	An error occurred while starting replication	Check the provided credentials if they are valid
14	An error occurred while granting the permissions	Check the logs for more information, permission error for performing operation. Try creating dSource again.
15	An error occurred while starting replication	The user has selected to provide a different replication user and password but the provided replication user or password is empty

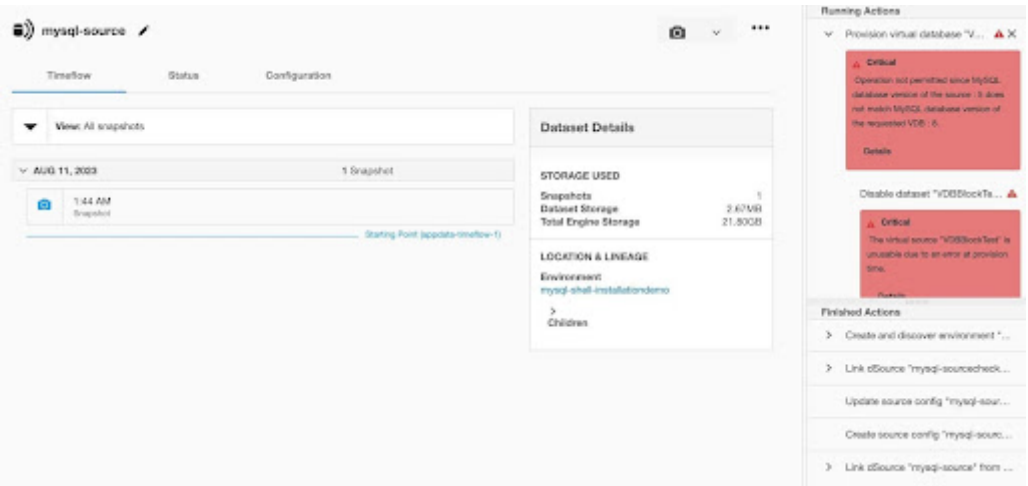
9.6.8.4 Version compatibility

The MySQL connector requires that the staging and target environment's MySQL **major** versions match. This also means that any snapshot's MySQL version must match the installation on the Target environment. This applies to all VDB operations, such as provision, rollback, and refresh. The operation will fail if the two versions are different, such as MySQL v5.7 on Staging and v8 on Target.

MySQL, unlike other databases, does not support automatic upgrades.

Guidance on how to upgrade your staging environments, target environments, and VDBs can be found in the Upgrading dSources and VDBs MySQL database versions (see page 1321) section.

In version 4.1.0 of the MySQL connector, we have introduced guardrails to minimize accidentally broken VDBs when refreshing from a lower versioned dSource. A sample screenshot can be seen below.



9.6.9 MySQL FAQs

9.6.9.1 How to create and configure an environment (host) key pair?

The environment's (host's) key pair enables Delphix Continuous Data Engine to SSH into the host and manage various discovery, linking, and provisioning operations. This authentication mechanism is called Username and Public Key during the environment configuration process. It is an alternative to Username/Password or Password Vault authentication.

Perform the following steps to create and configure an environment key pair during the (a) creation of an Environment or (b) addition of a new Environment User:

1. Under **Login Type**, select **Username and Public Key**.
2. Update the **Username** if desired.
3. The Public Key will either be presented below or select the **View Public Key** button.
   a. The key pair is automatically generated by the Delphix Continuous Data Engine with the Private Key pair saved locally.
4. Copy the full Public Key string that is displayed, and append it to the end of your environment's (host's) `~/.ssh/authorized_keys` file. If this file does not exist on the host, then you must create it.
   a. Run `chmod 600 ~/.ssh/authorized_keys` to allow only the file's owner to read and write to it (make sure the file is owned by you).
   b. Run `chmod 755 ~` to restrict access to your home directory so that no other may write to it.
   c. Run `chmod 700 ~/.ssh` so that others cannot write to it. The `~/.ssh` directory is not writable by group or others. Otherwise, authentication will fail.
5. Once added, select the **Validate** button to ensure it has been configured correctly. A success or error message will appear to direct further actions.
6. To complete the configuration, finish your creation of the Environment or select the **Add** button.

9.6.9.2 Why is the BLOB limited to 1GB in size?

LONGBLOBs can address up to 4 GB of memory using 32-bit binaries. However, this value is ignored as MySQL is instead bounded by the BLOB's smaller 1 GB limit. This is a limitation introduced by the MySQL protocol. Additional information can be found in the MySQL max_allowed_packet[418] documentation.

This question arises when the `mysqldump` command fails if there is more than 1 GB of data in a single LONGBLOB column or field. To resolve this issue, shrink or remove column and field data from your MySQL database that exceeds a size of 1 GB.

418 http://dev.mysql.com/doc/refman/5.1/en/server-system-variables.html#sysvar_max_allowed_packet

9.7 Oracle data sources

This section covers the following topics:

- Delphix architecture with Oracle (see page 1332)
- Transparent data encryption and Delphix (see page 1334)
- Oracle environment management (see page 1343)
- Quick start guide for Oracle on Linux (see page 1343)
- Quick start guide for Oracle on Solaris SPARC (see page 1362)
- Datapatch and Delphix (see page 1381)
- Oracle support and requirements (see page 1384)
- Managing Oracle environments and hosts (see page 1413)
- Linking data sources and syncing data with Oracle (see page 1438)
- Provisioning and managing virtual databases with Oracle (see page 1515)
- Oracle hook operations (see page 1647)

9.7.1 Delphix architecture with Oracle

9.7.1.1 Overview

Various Oracle configurations ranging from Oracle RAC to Oracle multi-tenant can be used with Delphix. This article contains an overview of how Delphix works with Oracle.

There are three key concepts when using Delphix with any data platform:

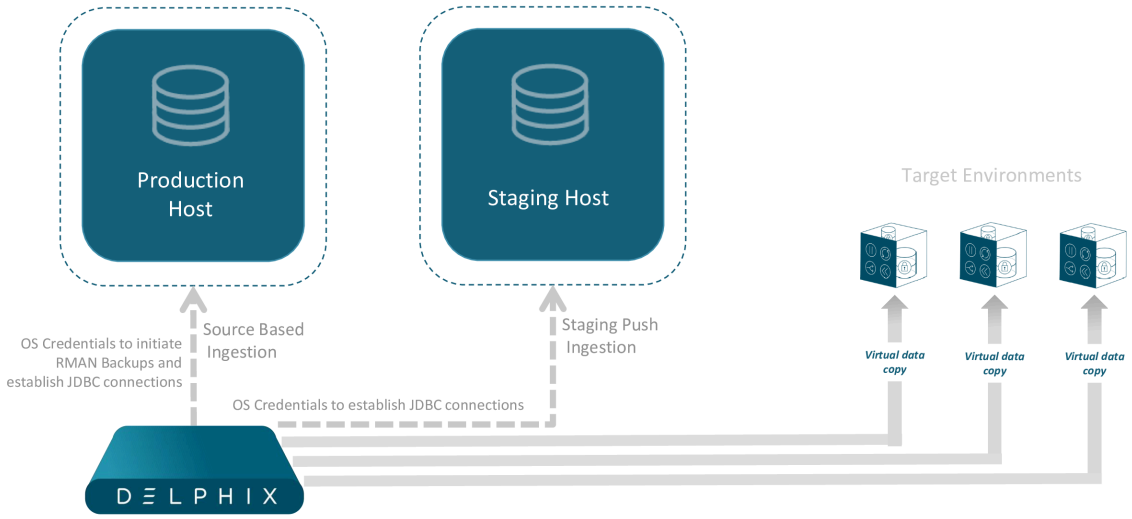
1. Environments: The server and software required to run a data set. For Oracle, this will be an operating system host with Oracle instances running on it.
    a. Source Environment: Source data to be ingested into Delphix. These will be used to create dSources.
    b. Target Environment: Target hosts to provision VDBs. These need Oracle installations that correspond to the versions of the Source environments, per our Oracle matrix (see page 1384).
2. dSources: A database that the Delphix Virtualization Engine uses to create and update virtual copies of your database
3. VDBs: A database provisioned from either a dSource or another VDB which is a copy of the source data. A VDB is created and managed by the Delphix Virtualization Engine.

With these concepts in mind, explore how Delphix connects to Oracle environments and creates Oracle dSources and VDBs.

Delphix is not a replacement for an archived log backup solution and should not be relied on to do so. This can cause issues where Delphix is unable to start a vPDB that was plugged into a physical CDB if Delphix is having to determine and deliver the archived logs. You will need to find a separate archived log backup solution workflow to own and be responsible for physical CDB archived logs.

9.7.1.2 Environment linking and provisioning architecture

As shown in the diagram below, there are two ways by which Delphix Engine can ingest data from an Oracle dSource.



9.7.1.2.1 Source based ingestion

In this method, Delphix Engine begins by ingesting data from your source databases in order to create dSources. Once an environment is added, Delphix Engine will automatically 'discover' databases on it, which are compatible to ingest from, to create new dSources.

When a dSource is to be added, we will leverage RMAN and JDBC to create snapshots of the source by taking a full database backup. The newer snapshots take incremental backups of the database to ingest incremental data between snapshots. SnapSync and LogSync policies can be leveraged for automated snapshots. The result is a TimeFlow with various snapshots from which you can provision a VDB.

9.7.1.2.2 Staging push ingestion

Staging Push introduces the concept of a staging instance for Oracle dSource ingestion. Storage for the staging instance is provided by the Delphix Engine via NFS. The staging database can be populated with production data after which a dSource Snapshot can be taken.

The staging database needs to be active only while taking dSource snapshots. It can be shut down after the snapshot to save system resources on the Staging host. Pre-sync and Post-sync hooks can be used to populate the staging database with production data before taking snapshots.

When you later provision a VDB, you can specify any environment as a target, including the environment that contains the source/staging database. However, we recommend choosing a different target environment for

the best performance. It must have an operating system that is compatible with the one running on the source/staging environment.

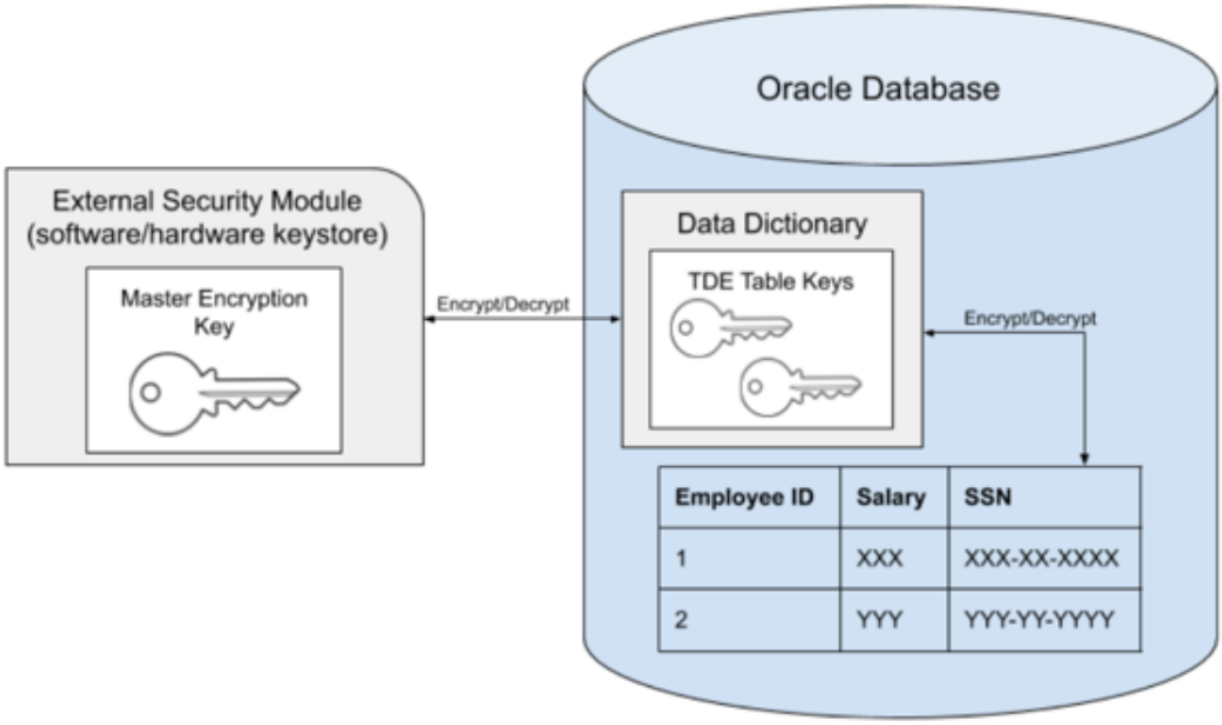
9.7.2 Transparent data encryption and Delphix

9.7.2.1 Overview

The Oracle Transparent Data Encryption (TDE) feature encrypts the sensitive data (database tables and tablespaces) stored on the disk. This prevents misuse of the data if the disks or storage mediums are lost or stolen. The data is transparently decrypted for the authorized users when they access the data.

Data is encrypted with the help of encryption keys which are stored in an external module or file, known as the *wallet* or *keystore* (see page 1336). The keystore is managed by an authorized user and can be either a hardware or software keystore. In order to decrypt the data successfully, the encryption keys must be made available to the database by the authorized user.

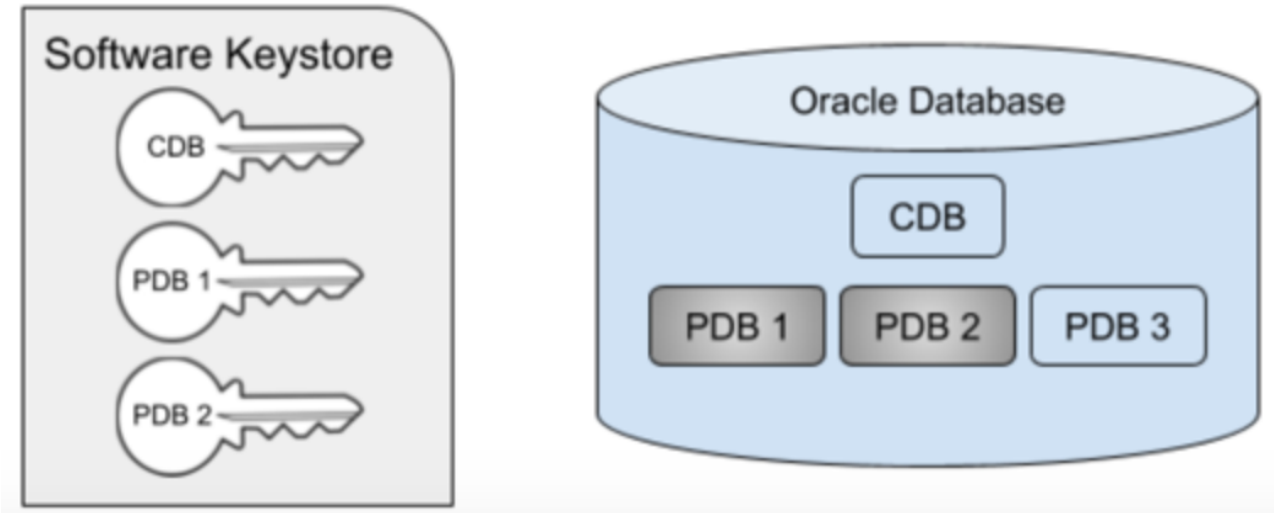
In the Oracle database, the data is organized into tables that are located within a tablespace, which is in turn made up of one or more files on disk. With TDE either individual tables or an entire tablespace can be encrypted. The keys used to encrypt the data are stored within the database itself, in the data dictionary. The keys themselves are further encrypted using the wallet keys. By using 2 layers of encryption in this manner, access to all of the encrypted data can be achieved with just access to the keys in the keystone.



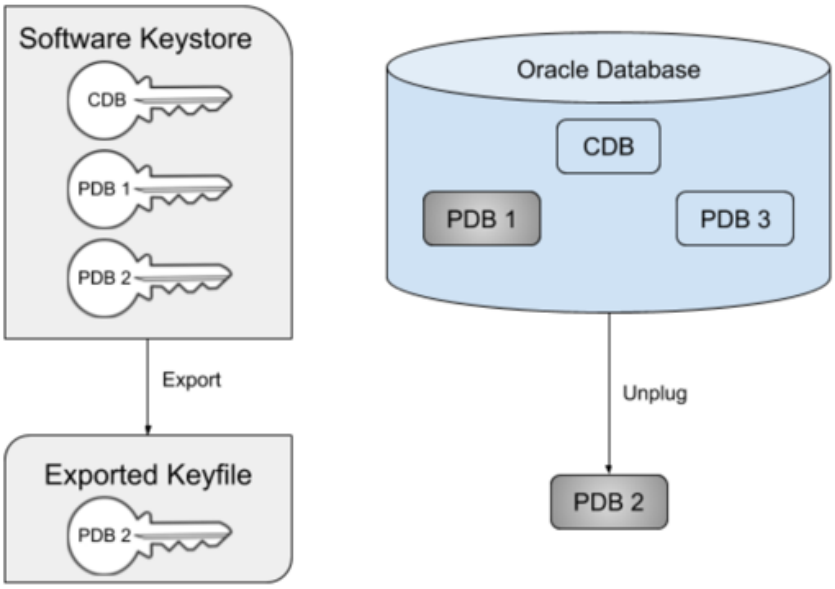
In the remainder of this document, the term *key* will refer to the master key stored in the external keystore, rather than the table keys stored in the data dictionary. In the case of a multi-tenant database, the same principle applies, namely that the keys are stored in the keystore which is located external to the database. However, not all of the PDBs within the container need to be encrypted. The CDB itself must always have a key associated with it, even if the CDB datafiles themselves are not encrypted. This is the case because

Oracle will encrypt both the datafiles belonging to the PDB, as well as the archive logs, which are located in the CDB, as illustrated in the following diagram.

In the diagram below, PDBs 1 and 2 are encrypted, but PDB 3 is not. Note that the keystore contains all of the keys for the entire multitenant database, in this case, keys for the CDB, PDB 1, and PDB 2.



In order to unplug a TDE-enabled PDB, the key(s) for that PDB must be exported from the keystore into a new file, known as an *exported keyfile* (see page 1336). Oracle will prevent the PDB from being unplugged unless the keys have first been exported. The exported keyfile is encrypted with a password (known as the *keyfile secret* (see page 0)) that is specified during the export. The diagram below illustrates the scenario where PDB 2 has been unplugged. Note that the key for PDB 2 still remains in the keystore. This illustrates an important point about Oracle keystores - once added, a key cannot be easily removed from a keystore.



You can plug back a PDB into a new CDB before importing the new keys. This results in the PDB being plugged in in restricted mode with plugin violations. Importing the keys is necessary to resolve the plugin violations and start up the PDB in unrestricted READ WRITE mode.

The design of TDE allows for the keys for a given CDB or PDB to be changed, via the `ADMINISTER KEY MANAGEMENT SET KEY` command. This process is known as *key rotation* and will update the master

encryption key in the keystore while leaving the table keys in the data dictionary. The table keys themselves will be encrypted with the new master encryption key, and any future updates to that data will use the new key. Existing data already written to the datafile or archive log will still be encrypted with the original key, however. For this reason, keys are never removed from an existing wallet, only new keys are added. To decrypt all of the data in a given database, all current and prior keys will be needed.

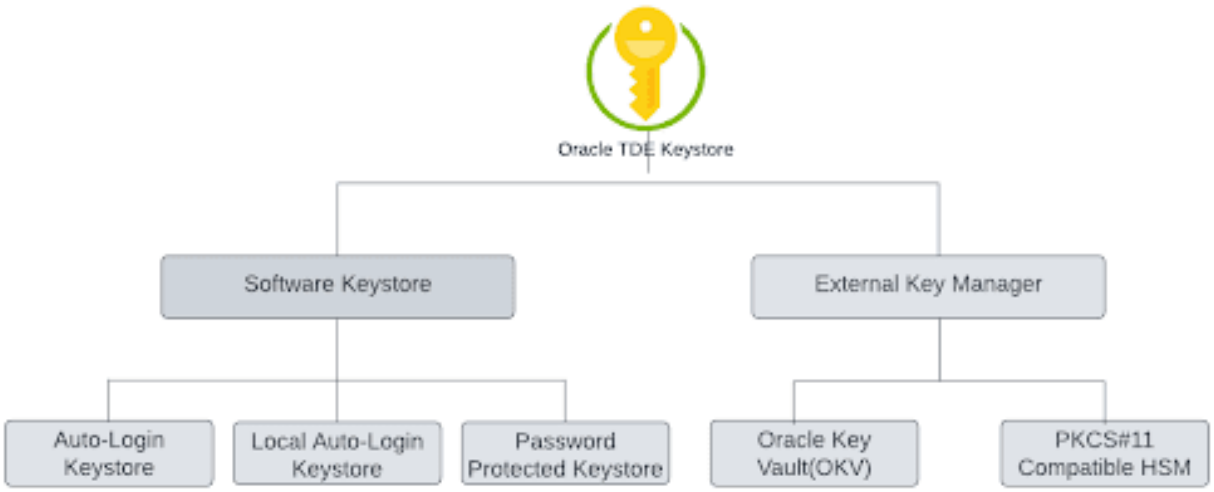
For more information on TDE and how it is configured and enabled, see the Oracle documentation[419].

9.7.2.2 Oracle TDE support for external key managers

Oracle database provides a comprehensive key management framework designed for TDE. This framework facilitates the storage and management of keys and credentials.

The key management capabilities within the Oracle database encompass support for various options, including software keystores, Oracle Key Vault (OKV), and other devices or key management applications compatible with the PKCS#11 standard. This diverse range of key management solutions provides you with the flexibility and choice to tailor your TDE implementations according to your security and operational requirements.

The below image illustrates the types of keystores that the Oracle database supports.



Delphix has introduced compatibility with External Key Managers (EKMs) for Oracle Multitenant databases. This enhancement will allow you to leverage robust key management solutions like OKV and Hardware Security Modules (HSMs) to enhance the security of your sensitive data and use Delphix Continuous Data Engine workflows to create virtual data copies and accelerate project deliveries.

9.7.2.3 Terminology

The following terms are used throughout TDE documentation and are summarized here for clarity. Note that the first occurrence of these terms may be on other documentation pages.

419 https://docs.oracle.com/en/database/oracle/oracle-database/19/asoag/introduction-to-transparent-data-encryption.html#GUID-62AA9447-FDCD-4A4C-B563-32DE04D55952

Term	Definition
Keystore/wallet	File found on the Oracle host which stores the keys used to encrypt and decrypt the internal table keys in a database. Every keystore has a password which is set when it is first created, and must be supplied for operations on it.
Parent keystore	Keystore with the keys used to encrypt the dSource PDB files.
Target keystore	Keystore for the target CDB into which the TDE-enabled vPDB is plugged.
Linked CDB Provision	Provisioning to physical CDBs that are configured to use TDE, and are part of the target environment added in Delphix Continuous Data Engine.
New Virtual CDB (vCDB) Provision	During the provisioning workflow for provisioning a new vPDB to a new vCDB, Delphix Continuous Data Engine will create a vCDB in the target environment and configure TDE.
Existing Virtual CDB (vCDB) Provision	Provisioning to existing vCDBs that are configured to use TDE, and are part of the target environment added in Delphix Continuous Data Engine.
Auxiliary container database (CDB)	Provisioning an Oracle vPDB requires running recovery to bring the snapshotted datafiles into a consistent state. This needs to be done in the context of a container database, which is created on the target system. After recovery is complete, the vPDB is unplugged and plugged into the target container, and the auxiliary container is deleted.
Artifact directory	Directory on the target system (not on Delphix Continuous Data Engine storage) which stores keys needed to support Delphix Continuous Data Engine workflows on TDE-enabled vPDBs. It is located under the keystores root directory.
Exported keyfile	File located on the target Oracle host which contains keys that have been exported from the keystore. It is encrypted with a secret that is specified when it is exported. The exported keyfile itself cannot be used as a keystore, but its contents can be imported into a new keystore.

Term	Definition
Key rotation	Process for changing the master encryption key in the keystore via the `ADMINISTER KEY MANAGEMENT SET KEY` command . This does not remove the original key, rather it adds a new key to the wallet and future data will be encrypted with the new key.
Keyfile secret	Password used to encrypt an exported keyfile.
Keystores root directory	User-specified location on the target system under which all TDE related artifacts such as keystores and exported keyfiles created by Delphix Continuous Data Engine are stored. This includes both the artifact directories used for vPDBs and temporary directories used for auxiliary CDB keystores.
Target Domain	A logical unit in CipherTrust Manager, contains the master encryption keys of target CDB into which the TDE-enabled vPDB is plugged.
Parent Domain	A logical unit in CipherTrust Manager, contains the master encryption keys used to encrypt the dSource PDB files.
TDE External Key Manager Credential	The credentials used to access the master encryption keys of the External Key Manager.
TDE Encryption Secret	A passphrase or key that serves as an additional layer of protection for your exported master encryption key and/or transport secret during `export/import/unplug/plug` operations of vPDB.
OKV Home	Oracle Key Vault Home. The installation directory path of the `okvclient.jar` binary in the Oracle database host.
Target Endpoint	Oracle database, registered and enrolled with OKV, contains the target CDB into which the TDE-enabled vPDB is plugged.
Parent Endpoint	Oracle database, registered and enrolled with OKV, contains keys used to encrypt the dSource PDB files.

9.7.2.4 Requirements

Oracle recommends that the keystore be stored on a separate disk from the datafiles. In accordance with this recommendation, neither keystores nor exported keyfiles are stored on Delphix storage. Rather, they are placed on customer storage. Exported keyfiles generated by Delphix are stored in the *artifact directory* (see page 1336) under the *keystores root directory* (see page 1336), while keystores generated by Delphix are stored in the location specified by `sqlnet.ora` or `WALLET_ROOT` initialization parameter of the target container database. It is the customer's responsibility to maintain these storage locations and ensure they are backed up as needed, just like database files. If the keystore or exported keyfile is lost, the data in the associated vPDB may not be recoverable and the vPDB will cease to operate.

9.7.2.4.1 General requirements

- Migration of a TDE-enabled vPDB from one CDB to another requires manual steps (see page 1613) that must be completed for migrate vPDB to be successful and to support refresh and rewind operations on the migrated vPDB.
- The dSource from which the initial provision is done must be encrypted when it is linked. If a dSource is encrypted after it is linked, a full backup must be taken and the resulting new dSource snapshot should be used for provisioning any TDE enabled vPDBs, otherwise provisioning will fail with `ORA-28311: Oracle encrypted data block not encrypted (file # 1, block # 520)`.
- There is currently no supported transition path from existing TDE-enabled vPDBs using the TDE workaround to the full product solution. The TDE workaround continues to be supported for approved customers.
- Delphix does not support Isolated keystore mode[420]. In isolated mode, a pluggable database (PDB) has its own keystore and the keystore and its TDE master encryption keys can be managed from the PDB only. Only United keystore mode[421] is supported.
- For linked provisions, the target container database should have an autologin keystore configured. For a cluster target, the autologin keystore is shared. For vCDB provisions, Delphix will create an autologin keystore when configuring the vCDB keystore.

> ⚠ Delphix does not support converting an unencrypted vPDB/vCDB into an encrypted one. These will need to be recreated from a new snapshot that is generated after taking a full backup of the corresponding PDB dSource.

420 https://docs.oracle.com/en/database/oracle/oracle-database/19/asoag/managing-keystores-encryption-keys-in-isolated-mode.html

421 https://docs.oracle.com/en/database/oracle/oracle-database/19/asoag/managing-keystores-encryption-keys-in-united-mode.html#GUID-FA569DEC-6FF3-4CA1-86AA-D27F29EDCA3C

9.7.2.4.2 Additional requirements for TDE software keystore based databases

- The Oracle version must be 12cR2 or higher (Oracle 12cR1 is not supported).
- Parent keystores of the source PDB must not be on ASM storage. Target CDB keystores, however, can be on ASM storage.
- Only software keystores on the same host as the database files are supported.

9.7.2.4.3 Additional requirements for TDE OKV-enabled databases

- The Oracle database version must be 18c or higher.
- To link a TDE OKV-enabled dSource, ensure that the OKV home path is set in the source environment added to the Delphix Continuous Data engine. For more information, refer to Adding or Editing the OKV Home (see page 1598).
- The target database endpoint must have at least Read access to the parent database endpoint master encryption keys.
- When provisioning a virtual pluggable database (vPDB) from a TDE OKV-enabled dSource snapshot, set the OKV Home path in the target environment added to the Delphix Continuous Data Engine. For more information, refer to Adding or Editing the OKV Home (see page 1598).
- TDE External Key Manager credential must be set either at the target environment level or at the container database (CDB) level in the target environment added to the Delphix Continuous Data Engine. For more information, refer to Adding or Editing the TDE External Key Manager Credential (see page 1596).

9.7.2.4.4 Additional requirements for TDE HSM-enabled databases

- The Oracle version must be 18c or higher.
- The target domain must have at least read access to the parent domain master encryption keys.
- When provisioning a virtual pluggable database from a TDE HSM-enabled dSource, set the TDE External Key Manager Credential either at the target environment level or at the container database (CDB) level in the target environment added to the Delphix Continuous Data Engine. In case of Thales CipherTrust Manager, the TDE External Key Manager Credential must adhere to the pattern `domain:username:password`. In the case of the default domain, it must be specified as `username:password`. For more information, refer to Adding or Editing the TDE External Key Manager Credential (see page 1596).

> The TDE External Key Manager credential may differ for other HSM providers. Refer to the specific HSM provider documentation for details on how the TDE External Key Manager credential should be specified.

Some or all of these restrictions may be relaxed in future versions of Delphix.

9.7.2.5 Wallet location configuration

Oracle requires that the keystore location be specified to the database so that it can be accessed when reading from or writing to the database files. This location can be specified in 2 ways:

1. The `ENCRYPTION_WALLET_LOCATION` parameter in `sqlnet.ora`.
2. The `wallet_root` initialization parameter is available starting in Oracle 18c, while `ENCRYPTION_WALLET_LOCATION` is available in Oracle 12.2.

> Starting with Oracle Database 23c, the parameter `ENCRYPTION_WALLET_LOCATION` is desupported per Oracle documentation[422].

Delphix supports both configurations for the appropriate releases (i.e. `ENCRYPTION_WALLET_LOCATION` only in 12.2, and both `ENCRYPTION_WALLET_LOCATION` and `wallet_root` in Oracle 18c+). When using `ENCRYPTION_WALLET_LOCATION`, Delphix recommends referencing an environment variable, for example:

9.7.2.5.1 Encryption wallet location for software-based keystore

```
ENCRYPTION_WALLET_LOCATION=
 (SOURCE=
  (METHOD=FILE)
   (METHOD_DATA=
    (DIRECTORY=/u03/app/ora12201/admin/$ORACLE_UNQNAME/wallet/)))
```

As there is only one `sqlnet.ora` file found under `$ORACLE_HOME`, it will be used for all databases that use that home. Specifying an environment variable such as `$ORACLE_UNQNAME` allows a different location for each database in the same `$ORACLE_HOME`. Any environment variable referenced in `sqlnet.ora` must always be set in the environment for the Oracle user. Delphix explicitly sets only `$ORACLE_HOME`, `$ORACLE_SID`, and `$ORACLE_UNQNAME` in the connections which are established by the Delphix engine, so it is recommended that only these variables be referenced in `sqlnet.ora`. For a 12.2 TDE vPDB provision, Delphix creates a unique `sqlnet.ora` file for the use of the auxiliary database during the provision. For provisions to vCDB targets, Delphix will set the `wallet_root` parameter to a user-provided path for versions 18c or higher and will use the path in `sqlnet.ora` for version 12.2.

422 https://docs.oracle.com/en/database/oracle/oracle-database/23/netrf/parameters-for-the-sqlnet.ora.html#GUID-897ABB80-64FE-4F13-9F8C-99361BB4465C

9.7.2.5.2 Encryption wallet location for OKV

- **Using** `sqlnet.ora`

```
ENCRYPTION_WALLET_LOCATION=
(SOURCE=
 (METHOD=OKV)
  (METHOD_DATA=
   (DIRECTORY=/u03/app/wallet/)))
```

- **Using** `WALLET_ROOT` **initialization parameter for Oracle 18c+**

The `WALLET_ROOT` directory must be the parent directory of `OKV_HOME`.

The directory that will be defined as `WALLET_ROOT` must have the following sub-directories (in lowercase) for the database to auto-discover the correct keystore configuration:

- `/okv`: The Oracle Key Vault client software is installed into this directory.
- `/tde`: The TDE wallet (or the auto-open wallet for Oracle Key Vault) goes into this subdirectory.

For provisions to vCDB targets, Delphix Continuous Data Engine will set the `WALLET_ROOT` parameter to a parent directory of the user-specified OKV Home path for versions 18c or higher.

9.7.2.5.3 Encryption wallet location for HSM

- **Using sqlnet.ora**

```
ENCRYPTION_WALLET_LOCATION=
(SOURCE=
 (METHOD=HSM)
  (METHOD_DATA=
   (DIRECTORY=/u03/app/wallet/)))
```

- **Using the "WALLET_ROOT" initialization parameter for databases with Oracle 18c+**

For provisions to the vCDB target, Delphix Continuous Data Engine will set the `WALLET_ROOT` parameter to a user-specified path for Oracle 18c or higher.

9.7.2.6 Required O/S permissions for the Delphix user

The provision itself is executed within the context of the environment user specified during the provision. This user does not have to be the Oracle user, and in fact, often is not. The Delphix user must be a member of the oracle group. During a TDE-enabled vPDB provision, the parent keystore is merged from a user-specified location to a location under the keystores root directory. The Delphix user does this copy via `ADMINISTER KEY MANAGEMENT` command. Since the Oracle user will do this, the Oracle user must be able to also create files in the wallet location.

The privilege requirements are satisfied by ensuring that the parent keystore has group read privileges, and the keystores root directory (owned by the Delphix user) has group write privileges.

> **Applicable only for OKV**
> If TDE is configured using `sqlnet.ora` for a database version of Oracle 18c or higher and provisioning to a vCDB, it is crucial to ensure that the Delphix Continuous Data Engine user has the necessary access to create a directory under `WALLET_ROOT`. This is because Delphix Continuous Data Engine attempts to configure the virtual Container Database (vCDB) using the `WALLET_ROOT` initialization parameter. In the case of Oracle Key Vault, the location of `WALLET_ROOT` is fixed, specifically the parent directory of `OKV_HOME`.

9.7.3 Oracle environment management

In order for the Delphix Engine to ingest data or provision virtual databases, your Oracle environments must comply with Delphix environment configuration standards.

9.7.4 Quick start guide for Oracle on Linux

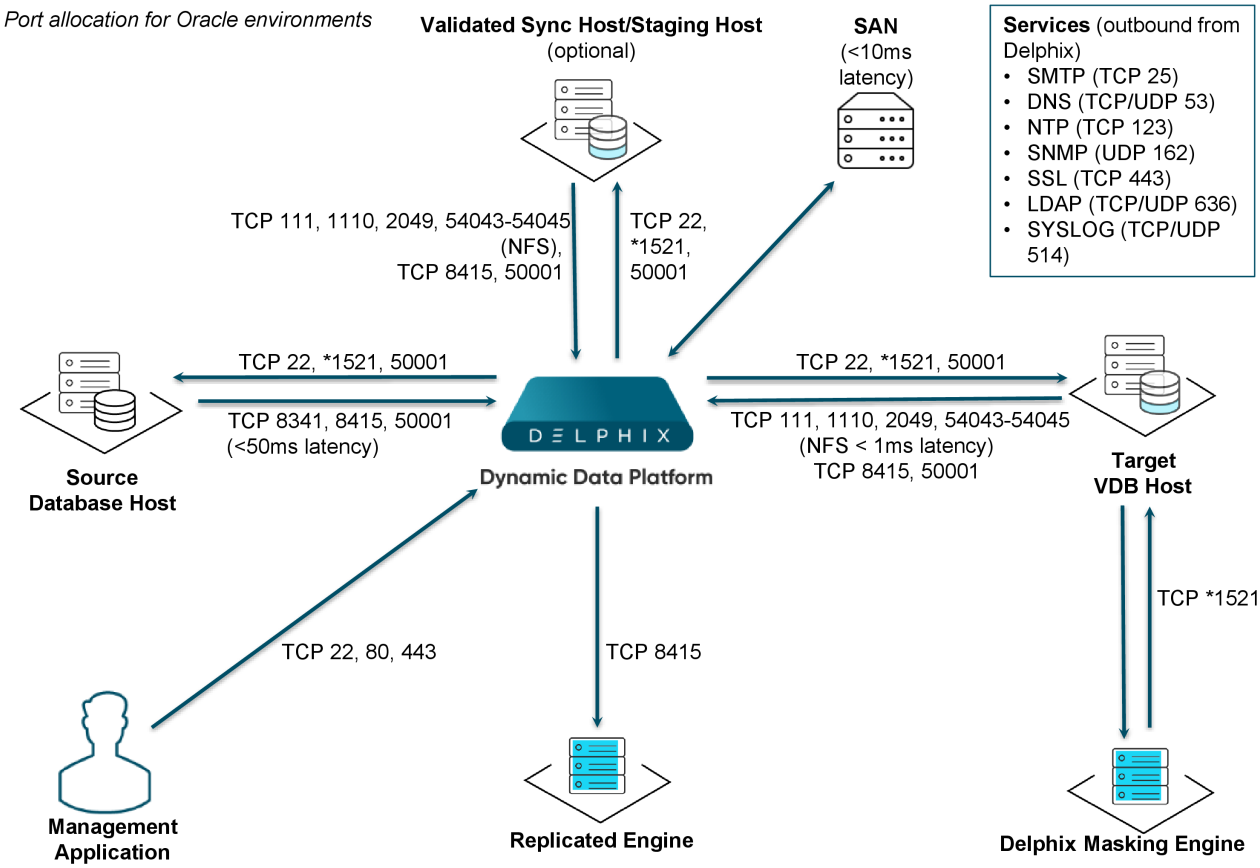
9.7.4.1 Overview

This quick start guide, which is excerpted from the larger User Guide, is intended to provide you with a quick overview of working with Oracle database objects in the Delphix Engine. It does not cover advanced configuration options including Oracle RAC, Linking to Standby, or best practices for performance. It assumes that you are working in a Lab/Dev setting and attempting to quickly test Delphix Engine functionality. It assumes you will use the VMware Hypervisor.

In this guide, we will walk through deploying a Delphix Engine, configuring Oracle Source and Target environments on Linux servers, creating a dSource, and provisioning a VDB.

The following diagram describes the engine topology for Oracle environments. It illustrates the recommended ports to be open from the engine to remote services, to the engine, and to the Source and Target Environments.

For this Quick Start Guide, you can ignore the following components: Validated Sync Host, Replicated Engine, and Delphix Masking Engine.



Note: *Oracle listener typically runs on TCP port 1521. In cases where other ports are used, substitute for 1521 above. Starting with 6.0.7.0 release, this port is required only if database user credentials are provided. The port need not be open if no database user credentials are provided.

9.7.4.2 Deploy OVA on VMware

Use the Delphix-supplied OVA file to install the Delphix Engine. The OVA file is configured with many of the minimum system requirements. The underlying storage for the install is assumed to be redundant SAN storage.

1. Download the OVA file from https://download.delphix.com[423]. You will need a support login from your sales team or a welcome letter.
    a. Navigate to the Delphix Product Releases/ /appliance="" images="">
2. Login using the vSphere client to the vSphere server (or vCenter Server) where you want to install the Delphix Engine.
3. In the vSphere Client, click **File**.
4. Select **Deploy OVA Template**.
5. Browse to the OVA file.
6. Click **Next**.

423 https://download.delphix.com/

7. Select a **hostname** for the Delphix Engine. This hostname will also be used in configuring the Delphix Engine network.
8. Select the **data center** where the Delphix Engine will be located.
9. Select the **cluster** and the **ESX host**.
10. Select one (1) **data store** for the **Delphix OS**. This datastore can be **thin-provisioned** and must have enough free space to accommodate the 127GB comprising the Delphix operating system.
11. Select four (4) or more **data stores** for Database Storage for the Delphix Engine. The Delphix Engine will stripe all of the Database Storage across these VMDKs, so for optimal I/O performance, each VMDK must be equal in size and be configured **Thick Provisioned - Eager Zeroed**. Additionally, these VMDKs should be distributed as evenly as possible across all four SCSI I/O controllers.
12. Select the **virtual network** you want to use. If using multiple physical NICs for link aggregation, you must use vSphere NIC teaming. Do not add multiple virtual NICs to the Delphix Engine itself. The Delphix Engine should use a single virtual network. For more information, see Optimal Network Architecture for the Delphix Engine (see page 719)
13. Click **Finish**. The installation will begin and the Delphix Engine will be created in the location you specified.
14. Once the installation has completed, power on the Delphix Engine and proceed with the initial system configuration as described in Setting Up Network Access to the Delphix Engine (see page 569)

> If your source database is 4 TB, you probably need 4 TB of storage for the Delphix Engine. Add at least 4 data disks of similar size for the Delphix VM. For example: for a source database of 4 TB, create 4 VMDKs of 1 TB each.

> For a full list of requirements and best practice recommendations, see Virtual Machine Requirements for VMware Platform (see page 604)

9.7.4.3 Setup network access to Delphix engine

1. Power on the Delphix Engine and open the Console.
2. Wait for the Delphix Management Service and Delphix Boot Service to come online. This might take up to 10 minutes during the first boot. Wait for the large orange box to turn green.
3. Press any key to access the sysadmin console.
4. Enter `sysadmin@SYSTEM` for the username and `sysadmin` for the password.
5. You will be presented with a description of available network settings and instructions for editing.

```
Delphix Engine Network Setup

To access the system setup through the browser, the system must first be
configured for networking in your
environment. From here, you can configure the primary interface, DNS, hostname,
and default route. When DHCP is
configured, all other properties are derived from DHCP settings.

To see the current settings, run "get." To change a property, run "set =." To
commit your changes,
run "commit." To exit this setup and return to the standard CLI, run "discard."

    defaultRoute    IP address of the gateway for the default route -- for
 example, "1.2.3.4."

    dhcp            Boolean value indicating whether DHCP should be used for
 the primary interface. Setting this value
                    to "true" will cause all other properties (address,
hostname, and DNS) to be derived from the DHCP
                    response

    dnsDomain       DNS Domain -- for example, "delphix.com"

    dnsServers      DNS server(s) as a list of IP addresses -- for example,
"1.2.3.4,5.6.7.8."

    hostname        Canonical system hostname, used in alert and other logs --
for example, "myserver"

    primaryAddress  Static address for the primary interface in CIDR notation
-- for example, "1.2.3.4/22"

Current settings:

    defaultRoute: 192.168.1.1
    dhcp: false
    dnsDomain: example.com
    dnsServers: 192.168.1.1
    hostname: Delphix
    primaryAddress: 192.168.1.100/24
```

6. Configure the `hostname`. If you are using DHCP, you can skip this step.

```
delphix network setup update *> set hostname=<hostname>
```

warning: Use the same `hostname` you entered during the server installation.

7. Configure DNS. If you are using DHCP, you can skip this step.

```
delphix network setup update *> set dnsDomain=<domain>
delphix network setup update *> set dnsServers=<server1-ip>[,<server2-ip>,...]
```

8. Configure either a static or DHCP address.

   DHCP Configuration

```
delphix network setup update *> set dhcp=true
```

   Static Configuration

```
delphix network setup update *> set dhcp=false
delphix network setup update *> set primaryAddress=<address>/<prefix-len>
```

   warning:The static IP address must be specified in CIDR notation (for example, `192.168.1.2/24`)

9. Configure a default gateway. If you are using DHCP, you can skip this step.

```
delphix network setup update *> set defaultRoute=<gateway-ip>
```

10. Commit your changes. Note that you can use the `get` command prior to committing to verify your desired configuration.

```
delphix network setup update *> commit
Successfully committed network settings. Further setup can be done through the
browser at:

        http://<address>

Type "exit" to disconnect, or any other commands to continue using the CLI.
```

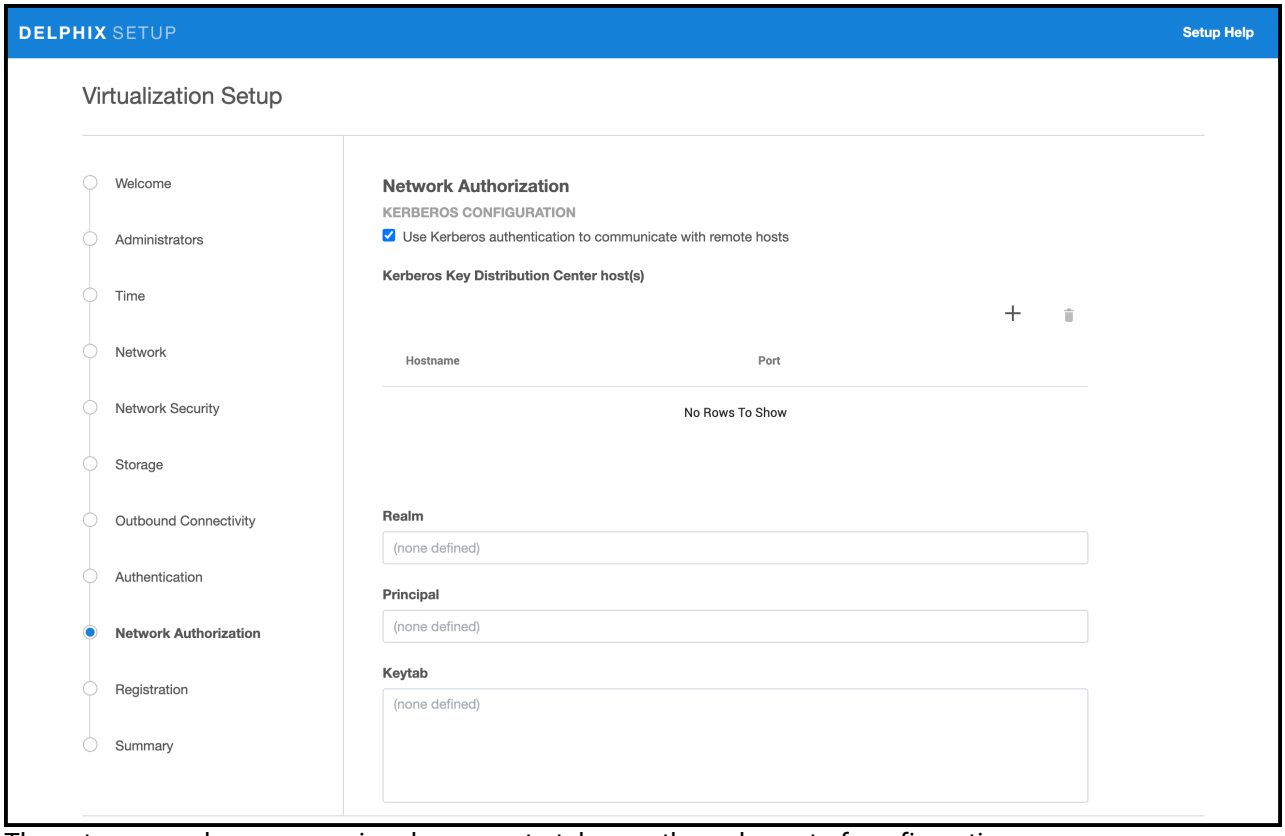
11. Check that you can now access the Delphix Engine through a Web browser by navigating to the displayed IP address, or hostname if using DNS.
12. Exit setup.

```
delphix> exit
```

9.7.4.4 Setting up the Delphix engine

Once you setup the network access for Delphix Engine, navigate to the Delphix Engine URL in your browser for server setup.

The welcome screen below will appear for you to begin your Delphix Engine setup.



The setup procedure uses a wizard process to take you through a set of configuration screens:

- Administrators
- Time
- Network
- Network Security
- Storage
- Outbound Connectivity
- Authentication
- Network Authorization
- Registration
- Summary

1. Connect to the Delphix Engine at `http:///login/index.html#serverSetup`.
   The **Delphix Setup** application will launch when you connect to the server. Enter your **sysadmin** login credentials, which initially defaults to the username **sysadmin**, with the initial default password of **sysadmin**. On first login, you will be prompted to change the initial default password.
2. Click **Next**.

9.7.4.4.1 Administrators

The Delphix Engine supports two types of administrators:

- System administrator (**sysadmin**) - this is the engine system administrator. The sysadmin password is defined here**.**
- Engine administrator (**admin)** - this is typically a DBA who will administer all the data managed by the engine.

On the Administrators tab, you set up the sysadmin password by entering an email address and password. The details for the admin are displayed for reference.

> The default domain user created on Delphix Engines from 5.3.1 is known as **admin** instead of delphix_admin. When engines created before 5.3.1 are upgraded to 5.3.1 or later they will retain their old username 'delphix_admin'. To avoid complications Delphix recommends creating users with an admin role and then Disabling delphix_admin.

9.7.4.4.2 Time

The engine time is used as the baseline for setting policies that coordinate between virtual databases and external applications

Choose your option to setup system time in this section. For a Quick Start, simply set the time and your timezone. You can change this later.

9.7.4.4.3 Network

The initial out-of-the-box network configuration in the OVA file is set to use a single VMXNET3 network adapter.

You have already configured this in the initial configuration. Delphix supports more advanced configurations, but you can enable those later.

9.7.4.4.4 Storage

You should see the data storage VMDKs or RDMs you created during the OVA installation. Click **Next** to configure these for data storage.

9.7.4.4.5 Serviceability

Choose your options to configure serviceability settings.

For a Quick Start, accept the defaults. You can change this later.

9.7.4.4.6 Authentication service

Choose your options to configure authentication services.

For a Quick Start, accept the defaults. You can change this later.

9.7.4.4.7 Registration

If the Delphix Engine has access to the external Internet (either directly or through a web proxy), then you can auto-register the Delphix Engine:

1. Enter your **Support username** and **Support password**.
2. Click **Register**.

If external connectivity is not immediately available, you must perform manual registration.

1. Copy the **Delphix engine registration code** in one of two ways:
   a. Manually highlight the registration code and copy it to clipboard. Or,
   b. Click **Copy registration code to clipboard**.
2. Transfer the Delphix Engine's registration code to a workstation with access to the external network Internet. For example, you could e-mail the registration code to an externally accessible e-mail account.
3. On a machine with access to the external Internet, please use your browser to navigate to the Delphix Registration Portal at http://register.delphix.com[424]
4. Login with your Delphix support credentials (username and password).
5. Paste the **Registration code**.
6. Click **Register**.

> Although your Delphix Engine will work without registration, we strongly recommend that you register each engine as part of the setup. Failing to register the Delphix Engine will impact its supportability and security in future versions.

To regenerate the registration code for a Delphix Engine please refer to, Regenerating the Delphix Engine Registration Code (see page 672). Delphix strongly recommends that you regenerate this code and re-register the engine regularly to maximize the Support Security of the Delphix Engine. Delphix recommends doing this every six months.

9.7.4.4.8 Summary

The final summary tab will enable you to review your configurations for System Time, Network, Storage, Serviceability, and Authentication.

1. Click the **Back** button to go back and to change the configuration for any of these server settings.
2. If you are ready to proceed, then click **Submit**.
3. Click **Yes** to confirm that you want to save the configuration.

---

424 http://register.delphix.com/

4. Click **Setup** to acknowledge the successful configuration.
5. There will be a wait of several minutes as the Delphix Engine completes the configuration.

9.7.4.5 Requirements for Oracle hosts and databases

In order to begin using Oracle environments with Delphix, you will need to configure the source and target hosts with the requirements described on this page.

9.7.4.6 Oracle hosts and databases

On each host with Oracle, there must be an operating system user configured to the required specifications for Delphix, as explained in the table below. This user (i.e. delphix_os) can easily be created by using the createDelphixOSUser.sh script (located at the bottom of the page).

These requirements apply to both source and target environments. However, target environments have additional requirements which are detailed in the 'Target Host Requirements' section below.

9.7.4.7 Source host requirement

Host requirement	Explanation
Profile and privileges should be the same as the Oracle user (e.g. oracle) on the host.	For example, delphix_os should have the same environment variable settings ($PATH, $ORACLE_HOME, etc.), umask, and ulimit settings, as the user oracle.
The Delphix software owner account (e.g. delphix_os) must have the same primary OS group as the Oracle software owner account (e.g. oracle).	Delphix recommends giving the delphix_os user the same primary OS group as the Oracle home owner. This ensures the Delphix engine can fully and automatically discover Oracle homes, databases, and listeners. Often, this is an OS group named oinstall. However, the oinstall group is not always necessary depending on your Oracle configuration. This user requires access to the libobk_proxy.so library in the toolkit for the child processes triggered by RMAN.

Host requirement	Explanation
The delphix_os user must have the Oracle OSDBA group (typically dba) as a primary or secondary OS group.	The OSDBA group allows "OS authentication" when connecting to an Oracle database instance by specifying either username or password (i.e. rman target /), thus eliminating the need to store or retrieve a SYSDBA password. **Oracle 12c** For Oracle 12c and later versions of Oracle, the delphix_os user can also use OSBACKUPDBA as its secondary group. This is typically the BACKUPDBA group on the host.
For secondary group requirements, the Delphix OS user must be a member of the SYSBACKUP OS group (12.1 or higher) or the SYSDBA OS group (11.2 and lower).	This ensures that the Delphix engine is able to take snapshots of source databases using RMAN.
There must be a directory on the source host where the Delphix Engine Toolkit can be installed, for example: /var/opt/delphix/toolkit	The delphix_os user and primary OS group (i.e. oinstall) must own the directory. The directory must have at least permissions -rwxrwx--- (0770) The delphix_os user must have read and execute permissions on each directory in the path leading to the toolkit directory. At least 1.6 GB of storage is needed at the time of setting up the environment and at least 500MB of free space is required to allow refreshes and maintenance of the toolkit, especially during upgrades.
The Delphix Engine must be able to make an SSH connection to the source host (typically port 22).	
Read access to $ORACLE_HOME and all underlying files and directories.	Delphix needs to run locally available Oracle tools such as sqlplus and RMAN. Those executables, as well as various required libraries, reside inside the Oracle home
Bash shell should be installed and available	Delphix requires the ability to run commands using the Bash shell. For the engine to operate with the host, Bash must be available and included in the system path, allowing for the execution of the 'bash' command. Most of the Linux machine bash is available by default and user may not need to do anything.

9.7.4.8 Additional requirements for RAC environments

If the source host is part of a RAC cluster, Delphix will attempt to use all nodes and crsctl for its operations.

RAC environment requirement	Explanation
The delphix_os user and Delphix Toolkit configuration must be the same on each node in the RAC cluster	• delphix_os must have execute permission on crsctl and srvctl on each node in the cluster. • access to olsnodes is needed. • The Delphix Toolkit must be installed in the same directory path on each of the nodes in the source cluster. • All data files, archive logs, and database control file must be located on storage shared by all of the cluster nodes. • Each node in the cluster must be able to access archive logs and the database control file from all other nodes. • At least 1.6 GB of storage is needed at the time of setting up the environment and at least 500MB of free space is required to allow refreshes and maintenance of the toolkit, especially during upgrades.

Masked Provisioning is not supported on Oracle RAC.

9.7.4.9 Auto-discovery requirements

The preferred way to find source databases it to allow Delphix to automatically discover your Oracle Homes and Databases by examining the inventory and oratab files and the listener setup. Successful autodiscovery requires read access to these and related files.

In **most** environments, delphix_os group membership is sufficient to perform auto-discovery.

However, if you have overridden Oracle's group permission structure, you may need to modify privileges to allow auto-discovery.

Auto-discovery requirements	Explanation
The ORATAB file must exist (typically in /etc/oratab or /var/opt/oracle/oratab) and be readable by delphix_os. Read access to: • /etc/orainst.loc or /var/opt/oracle/orainst.loc. • the Oracle inventory file (inventory.xml)	The ORATAB is used to determine the location of the Oracle installations on your host. Note: The Oracle inventory file is identified by the contents of orainst.loc (for example, $INVENTORY_HOME/ContentsXML/inventory.xml).
Permission to run pargs on Solaris hosts and ps on AIX, HP-UX, and Linux hosts, as super-user. This permission is usually granted via sudo authorization of the commands.	Unless you have used a custom TNS_ADMIN setting, elevated access to ps (pargs on Solaris) is not required See the topic Sudo Privilege Requirements for Oracle Environments (see page 1405) for further explanation of this requirement, and Sudo File Configuration Examples for Oracle Environments (see page 1409) for examples of file configurations

9.7.4.10 Source database requirements

For each source database, there are specific configurations required for Delphix to ingest data.

Source database requirement	Explanation
For each Oracle Home, the delphix_os user should have execute permission for the programs in $ORACLE_HOME/bin.	If symlinks are configured, Delphix must be configured with the same $ORACLE_HOME path as was used when starting the instance. Ensure the PermitUserEnvironment configuration parameter = "yes" in the sshd_config file The $ORACLE_HOME/bin/oracle executable must have the SETUID and SETGID flags set. Permissions on the oracle binary must be at least -rwsr-s–x (06751).
Source databases must be in ARCHIVELOG mode to ensure that redo logs are archived.	Archive logs are required to make SnapSyncs consistent and provisionable.
Enable Block Change Tracking (BCT). (Highly Recommended).	Enabling BCT will improve SnapSync operation time. Without BCT, incremental SnapSyncs must scan the entire database.

Source database requirement	Explanation
Enable FORCE LOGGING. (Highly Recommended)	If you do not enable FORCE LOGGING and NOLOGGING operations take place, you will get a Fault from the Delphix Engine. If you must use NOLOGGING to meet specific performance criteria, take a new snapshot of the source database after doing the NOLOGGING operations to bring the dSource up-to-date before provisioning VDBs.

9.7.4.11 Operating system specific requirements

Solaris

On a Solaris host, gtar must be installed. Delphix uses gtar to handle long file names when extracting the toolkit files into the toolkit directory on a Solaris host. The gtar binary should be installed in one of the following directories:

- /bin:/usr
- /bin:/sbin:/usr
- /sbin:/usr/contrib
- /bin:/usr/sfw
- /bin:/opt/sfw
- /bin:/opt/csw/bin

9.7.4.12 Additional target /staging host requirements

This topic describes the user privileges and environment requirements that are required for Oracle target hosts and databases collectively referred to as target environments.

These are in addition to the 'Oracle Hosts and Database Requirements' called out above.

Target/Staging host requirement	Explanation
The Delphix OS user must be a member of the SYSDBA OS group.	This ensures that the Delphix engine is able to create new VDBs on target hosts.
There must be a directory (e.g. "/mnt/provision/" or"/mnt/staging/") that will be used as a container for the NFS mount points that are created when provisioning a VDB.	The delphix_os user and primary OS group (i.e. oinstall or dba) must own the directory. The directory must have at least permissions -rwxrwx--- (0770). There must be no symbolic links in the path of this directory, because NFS can mount into a directory with symlinks in its path, but cannot unmount.

Target/Staging host requirement	Explanation
Permission to run: • mount, umount, mkdir, rmdir as super-user. • pargs on Solaris hosts • ps on Linux, AIX, and HP-UX as super-user. • nfso on AIX hosts as super-user	The following permissions are usually granted via sudo authorization of the commands. See Sudo Privilege Requirements for Oracle Environments (see page 1405) for further explanation of the commands, and Sudo File Configuration Examples for Oracle Environments (see page 1409) for examples of the /etc/sudoers file on different operating systems.
(Optional) Write permission to the $ORACLE_BASE_CONFIG/dbs directory.	(i.e. chmod g+w $ORACLE_BASE_CONFIG/dbs)
An Oracle listener process running on the target host for provisioning a VDB	The listener's version should be equal to or greater than the highest Oracle version that will be used to provision a VDB.
Required packages on target hosts: 1. portmapper / rpcbind 2. status daemon (rpc.statd) 3. NFS lock manager (rpc.lockd/lockmgr)	As the Delphix Engine uses NFSv3 for mounting target host filesystems, the prerequisite packages to support NFSv3 client communication are required for normal operation, and the required services to support NFS client communications (including file locking) must be running. NFSv3 is enabled by default and to enable NFSv4, see NFSv4 Configuration.
If using ASM, the delphix_os user must have the Oracle ASM ownership groups as secondary OS groups (typically asmadmin and asmdba)	This ensures the Delphix engine can fully and automatically discover Oracle homes, databases, and listeners.

Deploy Hostchecker to Validate Delphix Requirements

Delphix has developed a hostchecker script that contains standardized checks for source and target hosts - these checks generally fall into three buckets

- OS and Host permissions/access
- Network Port Checks
- DB-specific functionality

OS and Host permissions/access and network port checks can (and should) be performed prior to Delphix installation to ensure a smooth deployment.

Each DB should have a specifically associated hostchecker - you can find detailed documentation on the DB-specific hostchecker page.

1. Download the appropriate **HostChecker tarball** for your engine from https://download.delphix.com/ Tarballs follow the naming convention "hostchecker_<OS>_<processor>.tar". For example, if you are validating a linux x86 host you should download the hostchecker_linux_x86.tar tarball.

2. Create a working directory and extract the **HostChecker files** from the **HostChecker tarball**.

```
mkdir dlpx-host-checker
cd dlpx-host-checker/
tar -xf hostchecker_linux_x86.tar
```

3. Run the `sh` script contained within:

```
sh hostchecker.sh
```

This will extract the JDK included in the tarball (if necessary) and invoke the HostChecker.

```
ora10205@bbdhcp:/home/ora10205/hostchecker-> sh hostchecker.sh
Extracting the JDK from the tarball jdk-6u45-linux-i586.tar.gz.
```

warning: **Don't Run as Root** Do not run the HostChecker as root; this will cause misleading or incorrect results from many of the checks.

4. Select which **checks** you want to run.

Info: **Run Tests without the Interface**

You can also run checks without spawning the interface. Enter `--help` to get a list of arguments you can pass to the HostChecker.

5. As the checks are made, enter the requested **arguments**.
6. Read the output of the check. The general format is that severity increases as you scroll down the output. First comes informational output, then warnings, then errors. **warning** : Internal Errors If you see a message that starts with `Internal Error`, forward it to Delphix Support immediately. This represents a potential bug in the HostChecker, and not necessarily a problem with your environment.
7. Error or warning messages will explain any possible problems and how to address them. Resolve the issues that the HostChecker describes. Do not be surprised or undo your work if more errors appear the next time you run HostChecker, because the error you just fixed may have been masking other problems.
8. Repeat steps 3 - 7 until all the checks return no errors or warnings.

9.7.4.13 Adding Oracle source and target environments

Follow the steps below to add **both** source and target environments for Oracle.

1. Login to the **Delphix management** application.
2. Click **Manage**.
3. Select **Environments**.
4. Click the **Plus** icon next to **Environments**.

5. In the **Add environment** dialog, select **Unix/Linux**.
6. Select **Standalone host** or **Oracle cluster**, depending on the type of environment you are adding.
7. For standalone Oracle environments enter the **Host IP** address.
8. For Oracle RAC environments, enter the **Node address** and **Cluster home.**
9. Enter an optional **Name** for the environment.
10. Enter the **SSH** port. The default value is **22**.
11. Enter a **Username** for the environment. See Requirements for Oracle Target Hosts and Databases (see page 1392) for more information on the required privileges for the environment user.
12. Select a Login Type: — Username and Password - enter the OS username and password, or — Username and Public Key - enter the OS username. — Password Vault - select from an existing Enterprise Password Vault

    Info : Using Public Key Authentication

    If you want to use public-key encryption for logging into your Unix-based environment:
    a. Select **Public key** for the **Login type.**
    b. Click **View public key**.
    c. Copy the public key that is displayed, and append it to the end of your `~/.ssh/authorized_keys` file. If this file does not exist, you will need to create it.
        i. Run chmod 600 ~/.ssh/authorized_keys to allow only the file's owner to read and write to it (make sure the file is owned by the user).
        ii. Run chmod 755 ~ to restrict access to the user's home directory so no other user may write to it.
        iii. Run chmod 700 ~/.ssh so that others cannot write to it. The ~/.ssh directory cannot be writable by group or other users. Otherwise, authentication will fail.

    The public key needs to be added only once per user and per environment.
13. For **Password login**, click **Verify credentials** to test the username and password.
14. Enter a **Toolkit path**. The toolkit directory stores scripts used for Delphix Engine operations, and should have a persistent working directory rather than a temporary one. The toolkit directory will have a separate subdirectory for each database instance. The toolkit path must have 0770 permissions and at least 345MB of free space.
15. Click **Submit**.

9.7.4.14 Linking an Oracle data source

1. Login to the **Delphix management** application.
2. Navigate to **Manage > Datasets.**
3. Click the plus icon and select **Add dSource**.
4. In the **Add dSource** wizard, select the source database with the correct environment user-specified.

Info: For linking an Oracle staging push dSource, follow the procedure in Staging Push Implementation for Oracle (see page 1440)

5. Enter your login credentials for the source database and click **Verify credentials**. If you are linking a mounted standby, Click **Next**. See the topics under Linking Oracle Physical Standby Databases (see page 1466) for more information about how the Delphix Engine uses non-SYS login credentials.
6. Enter a name and select a group for your dSource. Adding a dSource to a dataset group lets you set Delphix Domain user permissions for that database and its objects, such as snapshots. See the topics under Users and Groups (see page 674) for more information.
7. Select the **Data Management** settings needed. For more information, visit Data Management Settings for Oracle Data Sources (see page 1452)
8. Assign existing policies to the new dSource. New policies can be created and associated later.
9. Enter any scripts that should be run using the **Hooks** page.
10. Review the dSource Configuration and Data Management information, and then click **Submit**.

Once the action to add a dSource has been submitted, the Delphix Engine will initiate a DB_Link job to create the dSource. If the *Load Immediately* option was selected in the data management page a DB_Sync job will also be executed to ingest data from the source, otherwise, this first DB_Sync job will run as per the associated SnapSync policy.

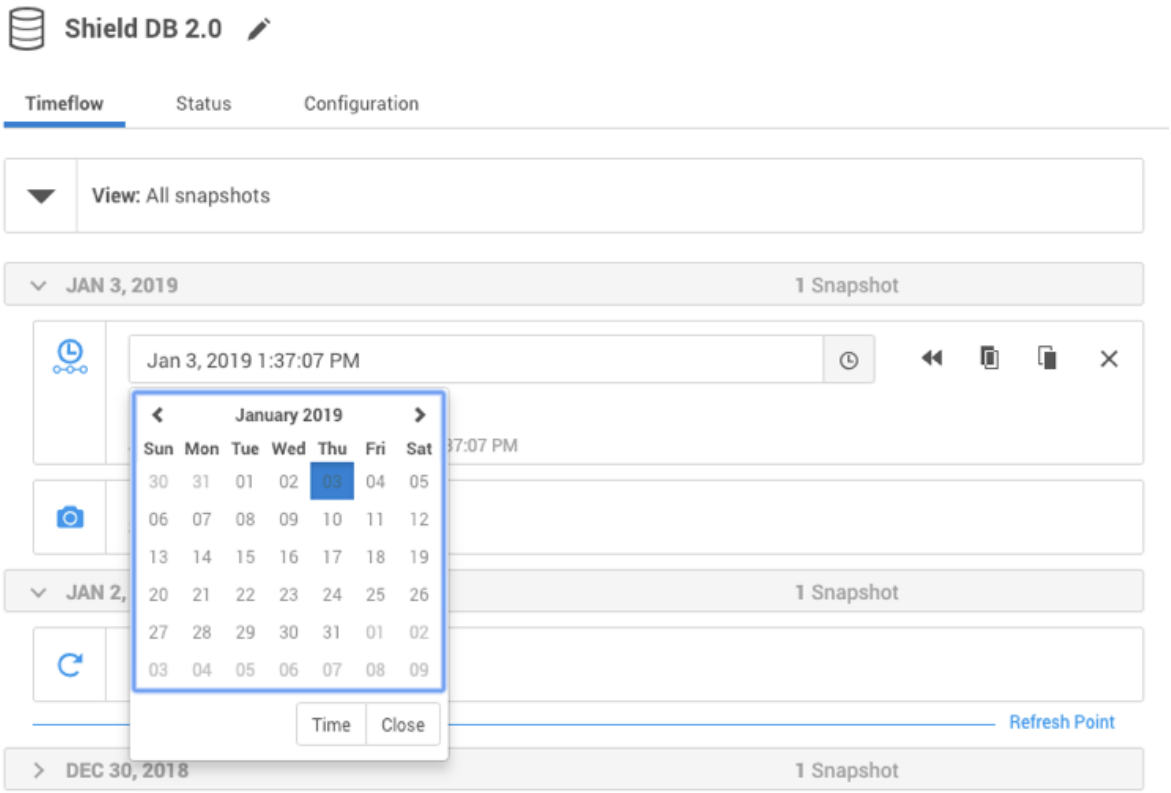
When the jobs have successfully completed, the database icon will change to a dSource icon on the Environments > Databases screen, and the dSource will be added to the list of Datasets under its assigned group.

9.7.4.15 Provisioning an Oracle VDB

1. Login to the **Delphix management** application.
2. Select **Manage > Datasets**.
3. In the **Datasets** panel on the left-hand side, click the **group** containing the dSource or VDB from which you want to provision.
4. Click the **TimeFlow** tab.
5. Select a **snapshot**.

   You can take a snapshot of the dSource from which to provision. To do so, click the **Camera** icon.
6. **Optional:** Select

   to open LogSync timeline.



7. Select

   to provision from a point of time within a snapshot. You can select by date or time.

8. Click

   and the **Provision VDB**wizard will open:

   a. For **Oracle single instance** the fields **Installation home**, **Database unique name**, **SID**, **Database name**, **Mount base**, and **Environment user** will auto-populate with information from the parent.
   b. For **Oracle RAC** the fields **Installation home**, **Database unique name**, **SID**, **Database name**, **Mount base**, **Instance number, Instance name** and **Environment user** will auto-populate with information from the parent.
      Editable Fields in the VDB Provision Wizard
      The following fields are editable:
      Installation Home (need to have an additional compatible target)
      Database Unique Name
      SID
      Database Name
      Mount Base
      Instance Number (RAC Only)
      Instance Name (RAC Only)

9. If you need to add a new target environment for the VDB, click the green **Plus** icon next to the **Filter target** field, and follow the instructions in Oracle Single Instance or RAC Environment (see page 1417)
10. Review the information for **Installation home**, **Database unique name**, **SID**, and **Database name**. Edit as necessary.
11. Review the **Mount base** and **Environment user**. Edit as necessary. The Environment User must have permission to write to the specified Mount Base, as described in Requirements for Oracle Target Hosts and Databases (see page 1392). You may also want to create a new writeable directory in the target environment with the correct permissions and use that as the Mount Base for the VDB.
12. Select **Provide privileged credentials** if you want to use login credentials on the target environment that are different from those associated with the **Environment user**.
13. Click **Advanced** to customize the VDB online log size and log groups, archivelog mode, local_listener parameter (TCP/IPC protocol addresses), additional VDB configuration settings or file mappings, or custom environment variables. For more information, see Customizing Oracle VDB Configuration Settings (see page 1530), Customizing VDB File Mappings (see page 1533), and Customizing Oracle VDB Environment Variables (see page 1530)

    If you are provisioning to a target environment that is running a Linux OS, you will need to compare the `SGA_TARGET` configuration parameter with the shared memory size in `/dev/shm`. The shared memory configured on the target host should match the SGA memory target. You can check the Linux OS shared memory size with the command df -k /dev/shm and the SGA_TARGET configuration parameter by opening the **Advanced** settings and then finding the value for `SGA_TARGET` under **VDB Configuration templates**.
14. Click **Next**.
15. Select a **Target Group** for the VDB.
16. Enable Auto VDB Restart to allow the VDB to be automatically restarted when the target host reboot is detected by Delphix.
17. Click Next.
18. Select a **Snapshot policy** for the VDB.
19. Click **Next**.
20. Enter any operations that should be run at Hooks during the provisioning process. Click **Next**.
21. Click **Submit**.

When provisioning starts, you can review the progress of the job by selecting the VDB and clicking on the **Status** tab, or by selecting **Manage/Dashboards** and viewing the **Job history** panel. Alternatively, you could see this in the **Actions sidebar.** When provisioning is complete, the VDB will be included in the group you designated and listed in the **Datasets** panel. If you select the VDB in the **Datasets** panel and click the **Configuration** tab, you can view information about the database and its Data Management settings.

9.7.4.16 Next steps

Congratulations! You have provisioned your first virtual database!

Now, perform some simple functional tests with your application. You can connect your app to the VDB using standard TNS/JDBC techniques. Delphix has already registered the VDB for you on the target listener.

We suggest the following next steps:

1. Drop a table and use the VDB Rewind feature to test the recovery of your VDB.
2. Take a snapshot of your dSource and refresh your VDB to quickly get fresh production data.
3. Provision a new VDB from your VDB to test the sharing data quickly with other teams.
4. Mask your new VDB to protect sensitive data. Provision new VDBs from that masked VDB to quickly provide safe data to development and QA teams.

9.7.4.17 Script

- createDelphixOSUser.sh [425]

9.7.5 Quick start guide for Oracle on Solaris SPARC

This quick start guide, which is excerpted from the larger User Guide, is intended to provide you with a quick overview of working with Oracle database objects in the Delphix Engine. It does not cover advanced configuration options including Oracle RAC, Linking to Standby, or Best Practices for Performance. It assumes that you are working in a Lab/Dev setting and attempting to quickly test Delphix functionality. It assumes you will use the VMware Hypervisor.

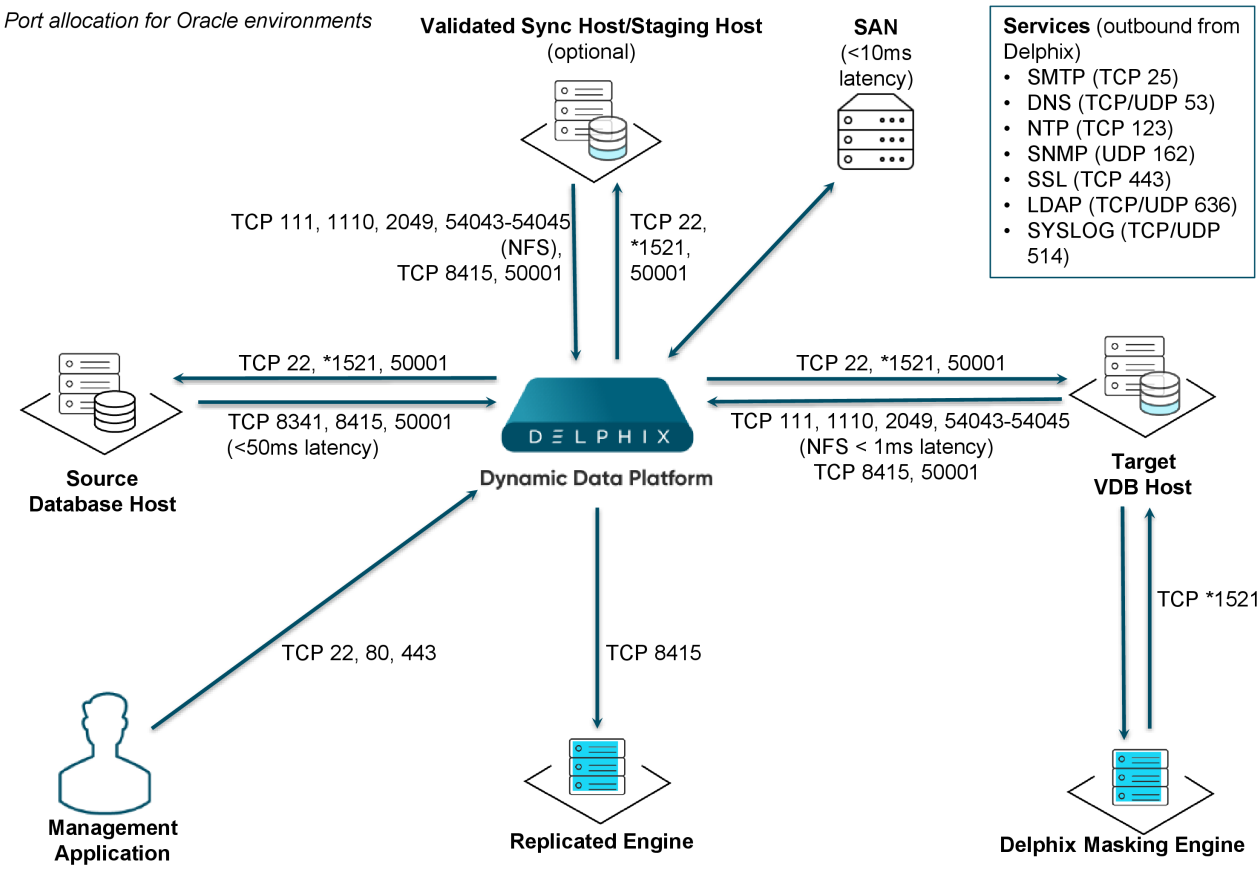
9.7.5.1 Overview

In this guide, we will walk through deploying a Delphix Engine, configuring Oracle Source and Target environments on Solaris SPARC servers, creating a dSource, and provisioning a VDB.

425 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149500419?preview=%2F149500419%2F149500462%2FcreateDelphixOSUser.sh

The following diagram describes the Delphix topology for Oracle environments. It illustrates the recommended ports to be open from Delphix to remote services, to the Delphix Engine, and to the Source and Target Environments.

For this Quick Start Guide, you can ignore the following components: Validated Sync Host, Replicated Engine, and Delphix Masking Engine.



Note: *Oracle listener typically runs on TCP port 1521. In cases where other ports are used, substitute for 1521 above. Starting with 6.0.7.0 release, this port is required only if database user credentials are provided. The port need not be open if no database user credentials are provided.

9.7.5.2 Deploy OVA on VMware

Use the Delphix-supplied OVA file to install the Delphix Engine. The OVA file is configured with many of the minimum system requirements. The underlying storage for the install is assumed to be redundant SAN storage.

1. Download the OVA file from https://download.delphix.com[426]. You will need a support login from your sales team or a welcome letter.
    a. Navigate to the Delphix Product Releases/<Current Version>/Appliance Images page.
2. Login using the vSphere client to the vSphere server (or vCenter Server) where you want to install the Delphix Engine.

426 https://download.delphix.com/

3. In the vSphere Client, click **File**.
4. Select **Deploy OVA template**.
5. Browse to the OVA file.
6. Click **Next**.
7. Select a **hostname** for the Delphix Engine. This hostname will also be used in configuring the Delphix Engine network.
8. Select the **data center** where the Delphix Engine will be located.
9. Select the **cluster** and the **ESX host**.
10. Select one (1) **data store** for the **Delphix OS**. This datastore can be **thin-provisioned** and must have enough free space to accommodate the 127GB comprising the Delphix operating system.
11. Select four (4) or more **data stores** for Database Storage for the Delphix Engine. The Delphix Engine will stripe all of the Database Storage across these VMDKs, so for optimal I/O performance, each VMDK must be equal in size and be configured **Thick Provisioned - eager zeroed**. Additionally, these VMDKs should be distributed as evenly as possible across all four SCSI I/O controllers.
12. Select the **virtual network** you want to use. If using multiple physical NICs for link aggregation, you must use vSphere NIC teaming. Do not add multiple virtual NICs to the Delphix Engine itself. The Delphix Engine should use a single virtual network. For more information, see Optimal Network Architecture for the Delphix Engine (see page 719)
13. Click **Finish**. The installation will begin and the Delphix Engine will be created in the location you specified.
14. Once the installation has completed, power on the Delphix Engine and proceed with the initial system configuration as described in Setting Up Network Access to the Delphix Engine (see page 569)

> If your source database is 4 TB, you probably need 4 TB of storage for the Delphix Engine. Add at least 4 similarly sized data disks for Delphix VM e.g. for 4 TB create 4 VMDKs of 1 TB each.

> For a full list of requirements and best practice recommendations, see Virtual Machine Requirements for VMware Platform (see page 604)

9.7.5.3 Setup network access to Delphix engine

1. Power on the Delphix Engine and open the Console.
2. Wait for the Delphix Management Service and Delphix Boot Service to come online. This might take up to 10 minutes during the first boot. Wait for the large orange box to turn green.
3. Press any key to access the sysadmin console.

4. Enter `sysadmin@SYSTEM` for the username and `sysadmin` for the password.
5. You will be presented with a description of available network settings and instructions for editing.

```
Delphix Engine Network Setup
To access the system setup through the browser, the system must first be
configured for networking in your
environment. From here, you can configure the primary interface, DNS, hostname,
and default route. When DHCP is
configured, all other properties are derived from DHCP settings.

To see the current settings, run "get." To change a property, run "set
<property>=<value>." To commit your changes,
run "commit." To exit this setup and return to the standard CLI, run "discard."

    defaultRoute    IP address of the gateway for the default route -- for
 example, "1.2.3.4."

    dhcp            Boolean value indicating whether DHCP should be used for
 the primary interface. Setting this value
                    to "true" will cause all other properties (address,
hostname, and DNS) to be derived from the DHCP
                    response
    dnsDomain       DNS Domain -- for example, "delphix.com"
    dnsServers      DNS server(s) as a list of IP addresses -- for example,
"1.2.3.4,5.6.7.8."
    hostname        Canonical system hostname, used in alert and other logs --
for example, "myserver"
    primaryAddress  Static address for the primary interface in CIDR notation
-- for example, "1.2.3.4/22"
Current settings:
    defaultRoute: 192.168.1.1
    dhcp: false
    dnsDomain: example.com
    dnsServers: 192.168.1.1
    hostname: Delphix
    primaryAddress: 192.168.1.100/24
```

6. Configure the `hostname`. If you are using DHCP, you can skip this step.

```
delphix network setup update *> set hostname=<hostname>
```

Note: Use the same `hostname` you entered during the server installation

7. Configure DNS. If you are using DHCP, you can skip this step.

```
delphix network setup update *> set dnsDomain=<domain>
delphix network setup update *> set dnsServers=<server1-ip>[,<server2-ip>,...]
```

8. Configure either a static or DHCP address.

   DHCP Configuration

   ```
   delphix network setup update *> set dhcp=true
   ```

   Static Configuration

   ```
   delphix network setup update *> set dhcp=false
   delphix network setup update *> set primaryAddress=<address>/<prefix-len>
   ```

   The static IP address must be specified in CIDR notation (for example, `192.168.1.2/24`)

9. Configure a default gateway. If you are using DHCP, you can skip this step.

   ```
   delphix network setup update *> set defaultRoute=<gateway-ip>
   ```

10. Commit your changes. Note that you can use the `get` command prior to committing to verify your desired configuration.

    ```
    delphix network setup update *> commit
    Successfully committed network settings. Further setup can be done through the
    browser at:
            http://<address>
    Type "exit" to disconnect, or any other commands to continue using the CLI.
    ```

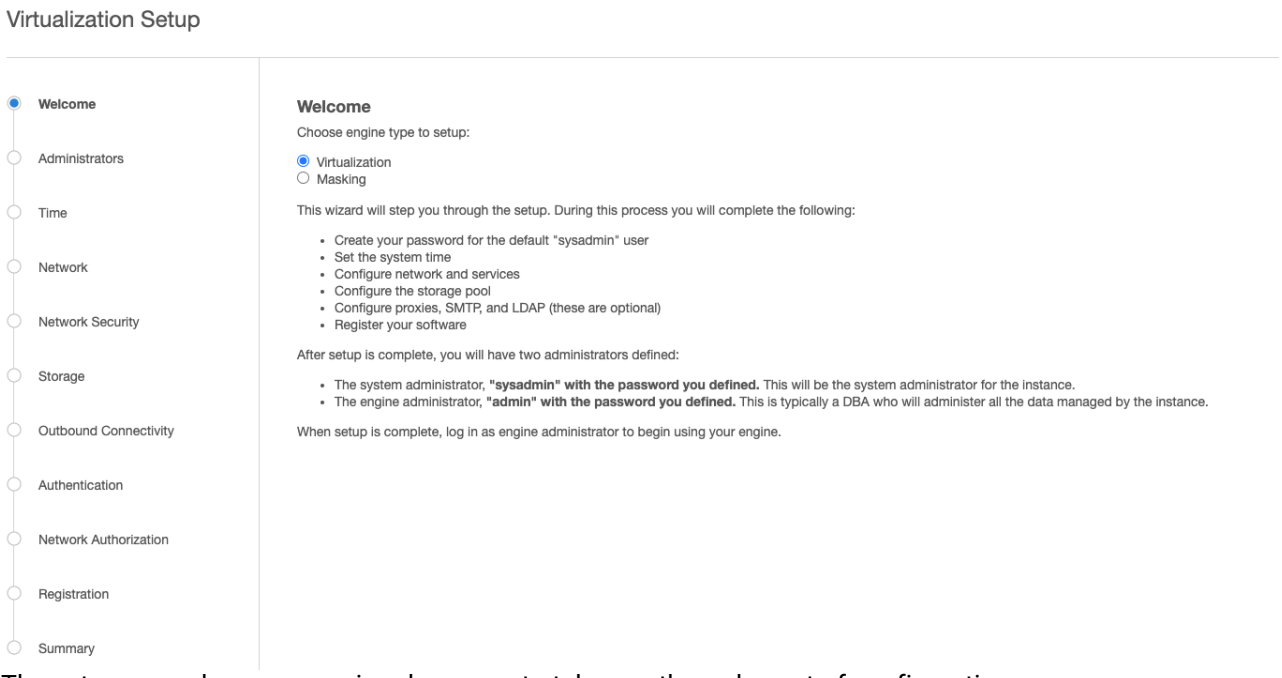
11. Check that you can now access the Delphix Engine through a Web browser by navigating to the displayed IP address, or hostname if using DNS.
12. Exit setup.

    ```
    delphix> exit
    ```

9.7.5.4 Setting up the Delphix engine

Once you setup the network access for Delphix Engine, enter Delphix Engine URL in your browser for server setup.

The welcome screen below will appear for you to begin your Delphix Engine setup.



The setup procedure uses a wizard process to take you through a set of configuration screens:

- Administrators
- Time
- Network
- Network Security
- Storage
- Outbound Connectivity
- Authentication
- Network Authorization
- Registration
- Summary

1. Connect to the Delphix Engine at `http:///login/index.html#serverSetup`. The **Delphix setup** application will launch when you connect to the server. Enter your **sysadmin** login credentials, which initially defaults to the username **sysadmin**, with the initial default password of **sysadmin**. On first login, you will be prompted to change the initial default password.
2. Click **Next**.

9.7.5.5 Administrators

The Delphix Engine supports two types of administrators:

- System Administrator (**sysadmin**) - this is the engine system administrator. The sysadmin password is defined here**.**
- Engine Administrator (**admin)** - this is typically a DBA who will administer all the data managed by the engine.

On the Administrators tab, you set up the sysadmin password by entering an email address and password. The details for the delphix_admin are displayed for reference.

> The default domain user created on Delphix Engines is now **admin** instead of delphix_admin. When engines created before 5.3.1 are upgraded to 5.3.1 or later they will retain their old username 'delphix_admin'. To avoid complications Delphix recommends creating users with an admin role and then Disabling delphix_admin.

9.7.5.6 System time

Choose your option to setup system time in this section.

For a Quick Start, simply set the time and your timezone. You can change this later.

9.7.5.7 Network

The initial out-of-the-box network configuration in the OVA file is set to use a single VMXNET3 network adapter.

You've already configured this in the initial configuration. Delphix supports more advanced configurations, but you can enable those later.

9.7.5.8 Storage

You should see the data storage VMDKs or RDMs you created during the OVA installation. Click Next to configure these for data storage.

9.7.5.9 Outbound connectivity

Choose your options to configure outbound connectivity settings.

For a Quick Start, take the defaults. You can change this later.

9.7.5.10 Authentication

Choose your options to configure authentication services.

For a Quick Start, take the defaults. You can change this later.

Registration

If the Delphix Engine has access to the external Internet (either directly or through a web proxy), then you can auto-register the Delphix Engine:

1. Enter your **Support username** and **Support password**.

2. Click **Register**.

If external connectivity is not immediately available, you must perform manual registration.

1. Copy the **Delphix engine registration code** in one of two ways:
   a. Manually highlight the registration code and copy it to clipboard. Or,
   b. Click **Copy registration Code to Clipboard**.
2. Transfer the Delphix Engine's registration code to a workstation with access to the external network Internet. For example, you could e-mail the registration code to an externally accessible e-mail account.
3. On a machine with access to the external Internet, please use your browser to navigate to the Delphix Registration Portal at http://register.delphix.com[427]
4. Login with your Delphix support credentials (username and password).
5. Paste the **Registration code**.
6. Click **Register**.

> While your Delphix Engine will work without registration, we strongly recommend that you register each Delphix Engine as part of the setup. Failing to register the Delphix Engine will impact its supportability and security in future versions.

To regenerate the registration code for a Delphix Engine please refer to, Regenerating the Delphix Engine Registration Code (see page 672). Delphix strongly recommends that you regenerate this code and re-register the engine regularly to maximize the Support Security of the Delphix Engine. Delphix recommends doing this every six months.

9.7.5.11 Summary

The setup procedure uses a wizard process to take you through a set of configuration screens:

- Administrators
- Time
- Network
- Network Security
- Storage
- Outbound Connectivity
- Authentication
- Network Authorization
- Registration
- Summary

427 http://register.delphix.com/

1. Connect to the Delphix Engine at `http:///login/index.html#serverSetup`. The **Delphix Setup** application will launch when you connect to the server. Enter your **sysadmin** login credentials, which initially defaults to the username **sysadmin**, with the initial default password of **sysadmin**. On first login, you will be prompted to change the initial default password.
2. Click **Next**.

9.7.5.12 Requirements for Oracle hosts and databases

9.7.5.13 Oracle hosts and databases

To begin using Oracle environments in Delphix, reference this article as a provided outline of the configuration requirements for the source and target hosts.

On each host with Oracle, there must be an operating system user configured to the required specifications for Delphix, as explained in the table below. This user (i.e. delphix_os) can easily be created by using the createDelphixOSUser.sh script (located at the bottom of the page).

These requirements apply to both source and target environments. However, target environments have additional requirements which are detailed in the 'Target Host Requirements' section below.

9.7.5.14 Source host requirement

Host requirement	Explanation
Profile and privileges should be the same as the Oracle user (e.g. oracle) on the host.	For example, delphix_os should have the same environment variable settings ($PATH, $ORACLE_HOME, etc.), umask, and ulimit settings, as the user oracle.
The Delphix software owner account (e.g. delphix_os) must have the same primary OS group as the Oracle software owner account (e.g. oracle).	Delphix recommends giving the delphix_os user the same primary OS group as the Oracle home owner. This ensures the Delphix engine can fully and automatically discover Oracle homes, databases, and listeners. Often, this is an OS group named oinstall. However, the oinstall group is not always necessary depending on your Oracle configuration. This user requires access to the libobk_proxy.so library in the toolkit for the child processes triggered by RMAN.

Host requirement	Explanation
The delphix_os user must have the Oracle OSDBA group (typically dba) as a primary or secondary OS group.	The OSDBA group allows "OS authentication" when connecting to an Oracle database instance by specifying neither username or password (i.e. rman target /), thus eliminating the need to store or retrieve an SYSDBA password. **Oracle 12c** For Oracle 12c and later versions of Oracle, the delphix_os user can also use OSBACKUPDBA as its secondary group. This is typically the BACKUPDBA group on the host.
For secondary group requirements, the Delphix OS user must be a member of the SYSBACKUP OS group (12.1 or higher) or the SYSDBA OS group (11.2 and lower).	This ensures that the Delphix engine is able to take snapshots of source databases using RMAN.
There must be a directory on the source host where the Delphix Engine Toolkit can be installed, for example: /var/opt/delphix/toolkit	The delphix_os user and primary OS group (i.e. oinstall) must own the directory. The directory must have at least permissions -rwxr-x-- (0750) The delphix_os user must have -rwxr-x-- (0750) permissions on each directory in the path leading to the toolkit directory. At least 1.6 GB of storage is needed at the time of setting up the environment and at least 500MB of free space is required to allow refreshes and maintenance of the toolkit, especially during upgrades.
The Delphix Engine must be able to make an SSH connection to the source host (typically port 22).	
Read access to $ORACLE_HOME and all underlying files and directories.	Delphix needs to run locally available Oracle tools such as sqlplus and RMAN. Those executables, as well as various required libraries, reside inside the Oracle home

9.7.5.15 Additional requirements for RAC environments

If the source host is part of a RAC cluster, Delphix will attempt to use all nodes and crsctl for its operations.

RAC environment requirement	Explanation
The delphix_os user and Delphix Toolkit configuration must be the same on each node in the RAC cluster	• delphix_os must have a execute permission on crsctl and srvctl on each node in the cluster. • access to olsnodes is needed. • The Delphix Toolkit must be installed in the same directory path on each of the nodes in the source cluster. • All data files, archive logs, and database control file must be located on storage shared by all of the cluster nodes. • Each node in the cluster must be able to access archive logs and the database control file from all other nodes. • At least 1.6 GB of storage is needed at the time of setting up the environment and at least 500MB of free space is required to allow refreshes and maintenance of the toolkit, especially during upgrades.

Masked Provisioning

Masked Provisioning is supported on Oracle RAC only when used with "script-based masking".

Auto-discovery requirements

The preferred way to find source databases it to allow Delphix to automatically discover your Oracle Homes and Databases by examining the inventory and oratab files and the listener setup. Successful autodiscovery requires read access to these and related files.

In **most** environments, delphix_os group membership is sufficient to perform auto-discovery.

However, if you have overridden Oracle's group permission structure, you may need to modify privileges to allow auto-discovery.

Auto-discovery requirements	Explanation
The ORATAB file must exist (typically in /etc/oratab or /var/opt/oracle/oratab) and be readable by delphix_os. Read access to: • /etc/orainst.loc or /var/opt/oracle/orainst.loc. • the Oracle inventory file (inventory.xml)	The ORATAB is used to determine the location of the Oracle installations on your host. Note: The Oracle inventory file is identified by the contents of orainst.loc (for example, $INVENTORY_HOME/ContentsXML/inventory.xml).

Auto-discovery requirements	Explanation
Permission to run pargs on Solaris hosts and ps on AIX, HP-UX, and Linux hosts, as super-user. This permission is usually granted via sudo authorization of the commands.	Unless you have used a custom TNS_ADMIN setting, elevated access to ps (pargs on Solaris) is not required See the topic Sudo Privilege Requirements for Oracle Environments (see page 1405) for further explanation of this requirement, and Sudo File Configuration Examples for Oracle Environments (see page 1409) for examples of file configurations

9.7.5.16 Source database requirements

For each source database, there are specific configurations required for Delphix to ingest data.

Source database requirement	Explanation
For each Oracle Home, the delphix_os user should have execute permission for the programs in $ORACLE_HOME/bin.	If symlinks are configured, Delphix must be configured with the same $ORACLE_HOME path as was used when starting the instance. Ensure the PermitUserEnvironment configuration parameter = "yes" in the sshd_config file The $ORACLE_HOME/bin/oracle executable must have the SETUID and SETGID flags set. Permissions on the oracle binary must be at least -rwsr-s–x (06751).
Source databases must be in ARCHIVELOG mode to ensure that redo logs are archived.	Archive logs are required to make SnapSyncs consistent and provisionable.
Enable Block Change Tracking (BCT). (Highly Recommended).	Enabling BCT will improve SnapSync operation time. Without BCT, incremental SnapSyncs must scan the entire database.
Enable FORCE LOGGING. (Highly Recommended)	If you do not enable FORCE LOGGING and NOLOGGING operations take place, you will get a Fault from the Delphix Engine. If you must use NOLOGGING to meet specific performance criteria, take a new snapshot of the source database after doing the NOLOGGING operations to bring the dSource up-to-date before provisioning VDBs.

Source database requirement	Explanation
If the online redo log files are located on RAW or ASM devices, then LogSync can only operate in Archive Only mode.	See the topics Advanced Data Management Settings for Oracle dSources (see page 2005) and Linking Oracle Physical Standby Databases (see page 1466) for more information.

9.7.5.17 Operating system specific requirements

Solaris

On a Solaris host, gtar must be installed. Delphix uses gtar to handle long file names when extracting the toolkit files into the toolkit directory on a Solaris host. The gtar binary should be installed in one of the following directories:

- / bin:/usr
- / bin:/sbin:/usr
- / sbin:/usr/contrib
- / bin:/usr/sfw
- / bin:/opt/sfw
- / bin:/opt/csw/bin

9.7.5.18 Additional target /staging host requirements

This topic describes the user privileges and environment requirements that are required for Oracle target hosts and databases collectively referred to as target environments.

These are in addition to the 'Oracle Hosts and Database Requirements' called out above.

Target/Staging host requirement	Explanation
The Delphix OS user must be a member of the SYSDBA OS group.	This ensures that the Delphix engine is able to create new VDBs on target hosts.
There must be a directory (e.g. "/mnt/provision/"or "/mnt/staging/") that will be used as a container for the NFS mount points that are created when provisioning a VDB.	The delphix_os user and primary OS group (i.e. oinstall or dba) must own the directory. The directory must have at least permissions -rwxrwx--- (0770). There must be no symbolic links in the path of this directory, because NFS can mount into a directory with symlinks in its path, but cannot unmount.

Target/Staging host requirement	Explanation
Permission to run: • mount, umount, mkdir, rmdir as super-user. • pargs on Solaris hosts • ps on Linux, AIX, and HP-UX as super-user. • nfso on AIX hosts as super-user	The following permissions are usually granted via sudo authorization of the commands. See Sudo Privilege Requirements for Oracle Environments (see page 1405) for further explanation of the commands, and Sudo File Configuration Examples for Oracle Environments (see page 1405) for examples of the /etc/sudoers file on different operating systems.
Optional write permission to the $ORACLE_BASE_CONFIG/dbs directory	If write permission is granted, an instance init file will be written to this directory during provisioning that will simplify manual instance startup. If write permission is not granted, manual instance startup must specify the instance init file. See Manually Starting a VDB (see page 1539)
An Oracle listener process running on the target host for provisioning a VDB	The listener's version should be equal to or greater than the highest Oracle version that will be used to provision a VDB.
Required packages on target hosts: 1. portmapper / rpcbind 2. status daemon (rpc.statd) 3. NFS lock manager (rpc.lockd/ lockmgr)	As the Delphix Engine uses NFSv3 for mounting target host filesystems, the prerequisite packages to support NFSv3 client communication are required for normal operation, and the required services to support NFS client communications (including file locking) must be running. NFSv3 is enabled by default and to enable NFSv4, see NFSv4 Configuration (see page 746)
If using ASM, the delphix_os user must have the Oracle ASM ownership groups as secondary OS groups (typically asmadmin and asmdba)	This ensures the Delphix engine can fully and automatically discover Oracle homes, databases, and listeners.

9.7.5.19 Deploy hostchecker to validate Delphix requirements

Delphix has developed a hostchecker script that contains standardized checks for source and target hosts - these checks generally fall into three buckets

- OS and Host permissions/access
- Network Port Checks
- DB-specific functionality

OS and Host permissions/access and network port checks can (and should) be performed prior to Delphix installation to ensure a smooth deployment.

Each DB should have a specifically associated hostchecker - you can find detailed documentation on the DB-specific hostchecker page.

1. Download the appropriate **HostChecker tarball** for your engine from https://download.delphix.com/ Tarballs follow the naming convention "hostchecker_<OS>_<processor>.tar". For example, if you are validating a linux x86 host you should download the hostchecker_linux_x86.tar tarball.
2. Create a working directory and extract the **HostChecker files** from the **HostChecker tarball**.

   ```
   mkdir dlpx-host-checker
   cd dlpx-host-checker/
   tar -xf hostchecker_linux_x86.tar
   ```

3. Run the `sh` script contained within:

   ```
   sh hostchecker.sh
   ```

   This will extract the JDK included in the tarball (if necessary) and invoke the HostChecker.

   ```
   ora10205@bbdhcp:/home/ora10205/hostchecker-> sh hostchecker.sh
   Extracting the JDK from the tarball jdk-6u45-linux-i586.tar.gz.
   ```

   Warning:
   Don't run as root
   Do not run the HostChecker as root; this will cause misleading or incorrect results from many of the checks.

4. Select which **checks**you want to run.
   Note:
   Run Tests without the Interface

   You can also run checks without spawning the interface. Enter `--help` to get a list of arguments you can pass to the HostChecker.

5. As the checks are made, enter the requested **arguments**.
6. Read the output of the check. The general format is that severity increases as you scroll down the output. First comes informational output, then warnings, then errors.

   Note:
   Internal Errors

   If you see a message that starts with `Internal Error`, forward it to Delphix Support immediately. This represents a potential bug in the HostChecker, and not necessarily a problem with your environment.

7. Error or warning messages will explain any possible problems and how to address them. Resolve the issues that the HostChecker describes. Do not be surprised or undo your work if more errors appear

the next time you run HostChecker, because the error you just fixed may have been masking other problems.

8. Repeat steps 3 - 7 until all the checks return no errors or warnings.

9.7.5.20 Adding Oracle source and target environments

Follow the steps below to add **both** source and target environments for Oracle.

1. Login to the **Delphix management** application.
2. Click **Manage**.
3. Select **Environments**.
4. Click the **Plus** icon next to **Environments**.
5. In the **Add environment** dialog, select **Unix/Linux**.
6. Select **Standalone host** or **Oracle cluster**, depending on the type of environment you are adding.
7. For standalone Oracle environments enter the **Host IP** address.
8. For Oracle RAC environments, enter the **Node address** and **Cluster home.**
9. Enter an optional **Name** for the environment.
10. Enter the **SSH** port. The default value is **22**.
11. Enter a **Username** for the environment. See Requirements for Oracle Target Hosts and Databases (see page 1392) for more information on the required privileges for the environment user.
12. Select a Login Type: — Username and Password - enter the OS username and password, or — Username and Public Key - enter the OS username. — Password Vault - select from an existing Enterprise Password Vault
    Note:
    Using public key authentication

    If you want to use public-key encryption for logging into your Unix-based environment:
    a. Select **Public key** for the **Login type.**
    b. Click **View public key**.
    c. Copy the public key that is displayed, and append it to the end of your `~/.ssh/authorized_keys` file. If this file does not exist, you will need to create it.
        i. Run chmod 600 ~/.ssh/authorized_keys to allow only the file's owner to read and write to it (make sure the file is owned by the user).
        ii. Run chmod 755 ~ to restrict access to the user's home directory so no other user may write to it.
        iii. Run chmod 700 ~/.ssh so that others cannot write to it. The ~/.ssh directory cannot be writable by group or other users. Otherwise, authentication will fail.

    The public key needs to be added only once per user and per environment.
13. For **Password login**, click **Verify credentials** to test the username and password.

14. Enter a **Toolkit path**. The toolkit directory stores scripts used for Delphix Engine operations, and should have a persistent working directory rather than a temporary one. The toolkit directory will have a separate subdirectory for each database instance. The toolkit path must have 0770 permissions and at least 345MB of free space.
15. Click **Submit**.

9.7.5.21 Linking an Oracle data source

1. Login to the **Delphix management** application.
2. Navigate to **Manage > Datasets.**
3. Click the plus icon and select **Add dSource**.
4. In the **Add dSource** wizard, select the source database with the correct environment user-specified.
5. Enter your login credentials for the source database and click **Verify credentials**. If you are linking a mounted standby, Click **Next**. See the topics under Linking Oracle Physical Standby Databases (see page 1466) for more information about how the Delphix Engine uses non-SYS login credentials.
6. Enter a name and select a group for your dSource. Adding a dSource to a dataset group lets you set Delphix Domain user permissions for that database and its objects, such as snapshots. See the topics under Users and Groups (see page 674) for more information.
7. Select the **Data management** settings needed. For more information, visit Data Management Settings for Oracle Data Sources (see page 1452)
8. Assign existing policies to the new dSource. New policies can be created and associated later.
9. Enter any scripts that should be run using the **Hooks** page.
10. Review the dSource Configuration and Data Management information, and then click **Submit**.

Once the action to add a dSource has been submitted, the Delphix Engine will initiate a DB_Link job to create the dSource. If the *Load Immediately* option was selected in the data management page a DB_Sync job will also be executed to ingest data from the source, otherwise, this first DB_Sync job will run as per the associated SnapSync policy.

When the jobs have successfully completed, the database icon will change to a dSource icon on the Environments > Databases screen, and the dSource will be added to the list of Datasets under its assigned group.

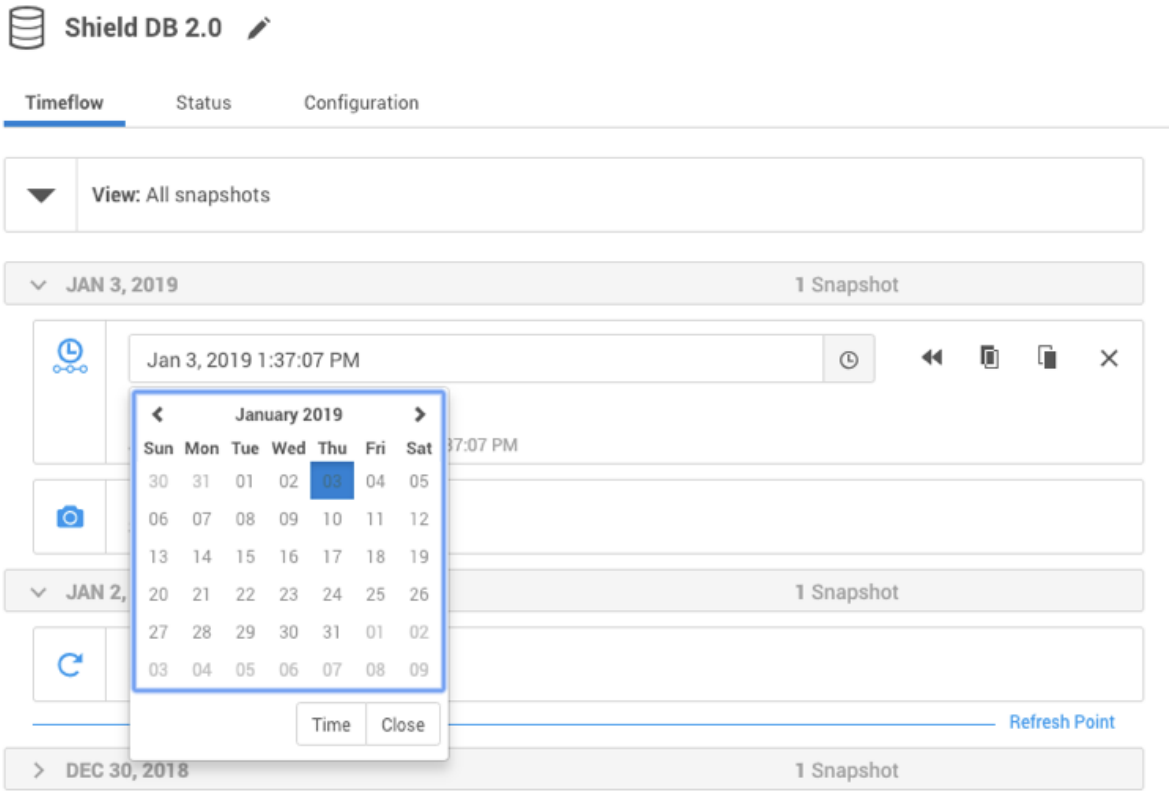
9.7.5.22 Provisioning an Oracle VDB

1. Login to the **Delphix management** application.
2. Select **Manage > Datasets**.
3. In the **Datasets** panel on the left-hand side, click the **group** containing the dSource or VDB from which you want to provision.
4. Click the **TimeFlow** tab.
5. Select a **snapshot**.

Note:
You can take a snapshot of the dSource from which to provision. To do so, click the **Camera** icon.

6. **Optional:** Select

   to open LogSync timeline.

7. Select

   to provision from a point of time within a snapshot. You can select by date or time.

8. Click

   and the **Provision VDB**wizard will open:

   a. For **Oracle single instance** the fields **Installation home**, **Database unique name**, **SID**, **Database name**, **Mount base**, and **Environment user** will auto-populate with information from the parent.
   b. For **Oracle RAC** the fields **Installation home**, **Database unique name**, **SID**, **Database name**, **Mount base**, **Instance number, Instance name** and **Environment user** will auto-populate with information from the parent.
      Note:
      Editable Fields in the VDB Provision Wizard
      The following fields are editable:

Installation Home (need to have an additional compatible target)
Database Unique Name
SID
Database Name
Mount Base
Instance Number (RAC Only)
Instance Name (RAC Only)

9. If you need to add a new target environment for the VDB, click the green **Plus** icon next to the **Filter target** field, and follow the instructions in Adding an Oracle Single Instance or RAC Environment (see page 1417)
10. Review the information for **Installation home**, **Database unique name**, **SID**, and **Database name**. Edit as necessary.
11. Review the **Mount base** and **Environment user**. Edit as necessary. The Environment User must have permission to write to the specified Mount Base, as described in Requirements for Oracle Target Hosts and Databases (see page 1392). You may also want to create a new writeable directory in the target environment with the correct permissions and use that as the Mount Base for the VDB.
12. Select **Provide privileged credentials** if you want to use login credentials on the target environment that are different from those associated with the **Environment user**.
13. Click **Advanced** to customize the VDB online log size and log groups, archivelog mode, local_listener parameter (TCP/IPC protocol addresses), additional VDB configuration settings or file mappings, or custom environment variables. For more information, see Customizing Oracle VDB Configuration Settings (see page 1530), Customizing VDB File Mappings (see page 1533), and Customizing Oracle VDB Environment Variables (see page 1530)
    If you are provisioning to a target environment that is running a Linux OS, you will need to compare the `SGA_TARGET` configuration parameter with the shared memory size in `/dev/shm`. The shared memory configured on the target host should match the SGA memory target. You can check the Linux OS shared memory size with the command df -k /dev/shm and the SGA_TARGET configuration parameter by opening the **Advanced** settings and then finding the value for `SGA_TARGET` under **VDB Configuration templates**.
14. Click **Next**.
15. Select a **Target group** for the VDB.
16. Enable Auto VDB Restart to allow the VDB to be automatically restarted when the target host reboot is detected by Delphix.
17. Click Next.
18. Select a **Snapshot policy** for the VDB.
19. Click **Next**.
20. Enter any operations that should be run at Hooks during the provisioning process. Click **Next**.
21. Click **Submit**.

When provisioning starts, you can review the progress of the job by selecting the VDB and clicking on the **Status** tab, or by selecting **Manage/Dashboards** and viewing the **Job history** panel. Alternatively, you could

see this in the **Actions sidebar.** When provisioning is complete, the VDB will be included in the group you designated and listed in the **Datasets** panel. If you select the VDB in the **Datasets** panel and click the **Configuration** tab, you can view information about the database and its Data Management settings.

9.7.5.23 Next steps

Congratulations! You've provisioned your first virtual database!

You should attempt some simple functional tests with your application. You can connect your app to the VDB using standard TNS/JDBC techniques. Delphix has already registered the VDB for you on the Target listener.

We suggest the following next steps:

- dropping a table and using the VDB Rewind feature to test recovery of your VDB
- Take a snapshot of your dSource and Refresh your VDB to quickly get fresh production data
- Provision a new VDB from your VDB to test sharing data quickly with other teams
- Mask your new VDB to protect sensitive data. Provision new VDBs from that Masked VDB to quickly provide safe data to development and QA teams.

9.7.5.24 Script

- createDelphixOSUser.sh[428]

9.7.6 Datapatch and Delphix

9.7.6.1 Overview

Oracle typically releases a major update yearly, and patch updates quarterly, for their database software. This is in addition to one-off patches for a specific customer (equivalent to a Delphix hotfix). The major releases are identified by the first digit in the release string, i.e. Oracle 18 vs Oracle 19. Patch releases (known as release updates and release update revisions) are identified by the second and third digits, i.e. 18.5.0 vs 18.7.1.

For Oracle 12c, the major version is considered to be 12.2.0.1. Release updates are still produced quarterly for 12c, but they do not change the actual version string.

Moving from one major release to another requires going through the upgrade process, while moving from a patch release requires going through the patch process.

In either the upgrade or patching scenario, following needs to be updated:

- Oracle binaries themselves (i.e. the contents of the Oracle home). Oracle provides a tool OPatch for patching Oracle binaries.
- database files (i.e. the Oracle data dictionary and built in PL/SQL packages). Oracle provides a tool known as Datapatch to accomplish this.

428 https://cdn.document360.io/94013a5a-f4c1-482c-90b9-814b81008535/Images/Documentation/createDelphixOSUser%20(1).sh

After the binary patching has been completed (on all nodes for a cluster), the database files are then patched. Oracle provides a tool known as Datapatch to accomplish this. Datapatch can be invoked either for an entire container and all PDBs within it, or a given set of PDBs. The purpose of Datapatch is to bring the SQL patch level in line with the current binary patch level of the database in which it is run. This may involve applying or rolling back patches in SQL depending on the current state of the SQL and binary patch registries.

Datapatch should be invoked whenever the database has been moved from one Oracle home to another, or when the binaries in the existing Oracle home are updated. This also applies to a PDB. In particular, when a PDB is unplugged from a source container, and then plugged into a target container running in a different Oracle home, the SQL patch registry within the PDB will reflect the source container state, rather than the target container state. Oracle detects this when opening the PDB, and generates a plug in violation and prevents the PDB from being opened in normal mode. After running datapatch on the PDB to bring its SQL state in line with the target container state (which should already be in line with the target binary state) the PDB can be closed and reopened in normal mode, clearing the violation.

For more detailed information, please refer to the Oracle Datapatch User Guide[429].

9.7.6.2 Delphix Implementation of Datapatch Invocation

Delphix supports customers to run Datapatch against their virtual databases when the databases are created, refreshed, rewound, started, or enabled via Delphix GUI or CLI. Delphix does not support running Datapatch for a database if it's not managed by Delphix (e.g. linked CDB, PDB, non-multitenant database) or it's a vCDB with multiple vPDBs, customers can run Datapatch for these databases without Delphix if necessary.

Delphix support for Datapatch has the following requirements:

- Binary patching should be completed (on all nodes for a cluster) on the target environment.
- Datapatch should be applied to the target Linked CDB or existing vCDB before Datapatch is applied to a vPDB in the CDB.

Delphix can invoke Datapatch against a virtual database in the following scenarios:

- Provision an Oracle VDB (Non-multitenant Virtual Database)

  When provisioning a VDB, if the **Invoke Datapatch** option is checked on the **Advanced** page, Delphix invokes Datapatch for the VDB during provision. After the VDB is provisioned, it's patched with the same patch level as that of the target Oracle home.

  If the **Invoke Datapatch** option is checked for the VDB, Datapatch is invoked against the VDB when it's refreshed, rewound, started, or enabled unless the option is unchecked as described in Configuration Settings for Oracle Virtual Databases (see page 1517).

- Provision an Oracle vPDB into a linked CDB or existing vCDB

> Delphix does not apply Datapatch against a linked CDB or existing vCDB, it's your responsibility to ensure that the linked CDB or existing vCDB is fully patched (i.e. the root container's SQL patch level is in line with the binary patch level of the Oracle home) before provisioning a vPDB into the CDB.

429 https://support.oracle.com/knowledge/Oracle%20Database%20Products/2680521_1.html

If the **Invoke Datapatch** option is checked in the **Advanced** page when provisioning an Oracle vPDB into a linked CDB or existing vCDB, after the provision, the provisioned vPDB is patched with the same patch level of the target Oracle home. Delphix only applies Datapatch against the newly provisioned vPDB, any existing PDB/vPDB in the CDB is not impacted.

If the **Invoke Datapatch** option is checked for the vPDB, Datapatch is invoked against the vPDB when it's refreshed, rewound, started, or enabled unless the option is unchecked as described in Configuration Settings for Oracle Virtual Databases (see page 1517).

- Provision an Oracle vPDB into a new vCDB

If the **Invoke Datapatch** option is checked in the **Advanced** page when provisioning an Oracle vPDB into a new vCDB, Delphix patches both the vPDB and vCDB during provision. After the provision, both vPDB and vCDB are patched with the same patch level of the target Oracle home.

If the **Invoke Datapatch** option is checked for the vPDB, Datapatch is invoked against the vPDB and vCDB when they are refreshed, rewound, started, or enabled unless the option is unchecked as described in Configuration Settings for Oracle Virtual Databases (see page 1517).

- Refresh, rewind, start, or enable a VDB or vPDB

If the **Invoke Datapatch** option is checked for the VDB or vPDB (the option can be switched in as described in Configuration Settings for Oracle Virtual Databases (see page 1517)), when the VDB or vPDB is refreshed, rewound, started, or enabled, Delphix applies Datapatch for them.

If a vPDB is the only vPDB in a vCDB, when the vPDB is refreshed, rewound, started, or enabled, Delphix applies Datapatch for both the vPDB and vCDB.

If a vPDB is not the only vPDB in a vCDB, when the vPDB is refreshed, rewound, started, or enabled, Delphix applies Datapatch for the vPDB only. The vCDB and other vPDBs in the vCDB are not impacted.

- Start or enable a vCDB
  When all vPDBs in a vCDB are stopped/disabled, Delphix allows users to start or enable the vCDB before starting/enabling its vPDBs. In this case, if the **Invoke Datapatch** option is checked for the vCDB, Delphix runs Datapatch for the vCDB during its start/enable. Its vPDBs are not impacted. But Delphix applies Datapatch against the vPDBs when they are refreshed, rewound, started, or enabled and the **Invoke Datapatch** option is checked.

For all the scenarios where Datapatch is invoked, Delphix invokes Datapatch in the target environment, that's to say, the source environment is not impacted by Datapatch.

9.7.6.3 When to select the Invoke Datapatch option in Delphix?

Datapatch is designed to be idempotent, thus users can run Datapatch against the same database multiple times even if it has been fully patched. If the database is fully patched, running Datapatch is a no-op, it makes no change to the database, although it takes a couple of minutes for Datapatch to check the patch status of the database. Delphix provides users the flexibility to run or not run Datapatch for a virtual database by switching the **Invoke Datapatch** option for the virtual database as described in Configuration Settings for Oracle Virtual Databases (see page 1517) at any time.

When provisioning a VDB or vPDB, if the source database has a different patch level than the target Oracle home, in order to apply Datapatch, users may check the **Invoke Datapatch** option during provision, so that it's patched after the provision.

If a new patch is applied to the Oracle home where a VDB, vPDB, or vCDB is already running, in order to apply Datapatch, users may check the **Invoke Datapatch** option described in Configuration Settings for Oracle

Virtual Databases (see page 1517) for the VDB, vPDB, or vCDB and refresh, rewind, stop/start, disable/enable it with Delphix.

If a VDB or vPDB is migrated to another environment that has a different patch level than the original environment, to apply Datapatch, users may check the **Invoke Datapatch** option for the VDB or vPDB before enabling the VDB or vPDB in the new environment.

9.7.6.4 Required OS Permissions for the Delphix User

Delphix invokes Datapatch within the context of the environment user associated with the virtual database. This user does not have to be an **Oracle** user, and in fact, often is not. While Datapatch needs to be executed by the **Oracle** user. Thus the environment user is required to set up the necessary sudo configuration for executing Datapatch as the **Oracle** user on the target environment where Datapatch is executed, and Delphix verifies this before invoking Datapatch.

For more information, see Sudo Privilege Requirements for Oracle Environments (see page 1405) and Sudo File Configuration Examples for Oracle Environments (see page 1409).

9.7.7 Oracle support and requirements

This section covers the following topics:

- Oracle matrix (see page 1384)
- Requirements for Oracle hosts and databases (see page 1392)
- Network and connectivity requirements for Oracle environments (see page 1399)
- Sudo privilege requirements for Oracle environments (see page 1405)
- Sudo file configuration examples for Oracle environments (see page 1409)

9.7.7.1 Oracle matrix

Source and Target OS and DBMS Compatibility

- Source and Target must have the same major Oracle version and Edition (Enterprise/Standard). It is recommended to also have the same minor versions and patches.
    - If the minor version (specified with the second digit with Oracle version 18+, or the dated patch level with Oracle version 12.2) or patches differ between the source and target, then a plug-in violation will be received when opening the vPDB on the target, and will need to run datapatch manually or via a hook to resolve it.
- Source and Target operating systems must be compatible. E.g., AIX -> AIX, Solaris SPARC -> Solaris SPARC, x86_64 Linux -> x86_64 Linux.
- In addition, x86_64 Solaris -> x86_64 Linux and x86_64 Linux -> x86_64 Solaris is supported.

9.7.7.1.1 Disclaimers and support notes

Delphix Support Policies specifically list **major** and **minor** release coverage. If a minor release is listed as covered, all patch releases under that minor release are certified. Support applies to corresponding versions of Community Enterprise Operating System (CentOS) / Oracle Linux (OL), 64-bit OS support only.

The Oracle version in the support matrix applies to both Oracle DB and Oracle Grid. Delphix supports the same set of features and functionality for supported non-multitenant database versions. Currently, Delphix does not support the Oracle 12c feature of THREADED_EXECUTION being set to TRUE, because this disables OS authentication.

Summary of Delphix features that are unsupported for the Oracle12c and higher Multi-tenant configuration:

- Existing Virtual Container Database (vCDB) as target for provisioning an additional vPDB for Oracle version 12.1.0.1
- Customize VDB settings/initialization parameters. Includes the following:
    - Customize init.ora database parameter during provisioning
    - Config templates
    - Online redo log size
    - Number of RAC VDB instances
    - Online redo log groups
    - Archive log mode
    - Setting new DBID
    - Customize local listeners
- Virtual to Physical (V2P) Support
- Resumable initial SnapSync
- Validated Sync
- Source continuity for dSource upgraded from Oracle 12c non-multitenant to multitenant database
- Cross-platform provisioning (XPP) to the virtual database
- Oracle LiveSources

Delphix supports systems that are part of an Exadata**,** Exadata Cloud Service (ExaCS), or Exadata Cloud-at-Customer (ExaCC) cluster, provided that the operating system and DBMS version are in the supported list. Oracle Exadata, Exadata Cloud Service (ExaCS), and Exadata Cloud-at-Customer (ExaCC) offer several features that are reliant on the underlying Exadata storage. Delphix supports creating dSources from Exadata source clusters and provisioning VDBs and vPDBs to Exadata target clusters.

The following Exadata-only features will not be available in child VDBs running in Delphix storage:

- Smart Scan - allows SQL processing to be offloaded to Exadata storage cells. Queries that leverage Smart Scan will execute successfully in VDBs but query performance may be impacted.
- Smart Flash Cache - allows frequently used data blocks to be stored in the flash cache of Exadata storage cells to allow for faster access. Queries against objects that are normally cached in Smart Flash Cache will execute successfully in VDBs but query performance may be impacted. As the virtualization engine uses memory to cache frequently used data blocks, allocating extra memory to the engine may help to mitigate this performance impact.
- Hybrid Columnar Compression (HCC) - compresses data and stores it in a columnar format, allowing Exadata storage cells to deliver faster query performance while also saving on storage. Columns

compressed using HCC cannot be used by queries executed in VDBs. Affected columns can be decompressed though care should be taken to evaluate the impact of doing so on VDB refresh times and Delphix storage usage. In Delphix VDBs, queries against partitions or tables using HCC will fail with the following error "*ORA-64307: Exadata Hybrid Columnar Compression is not supported for tablespaces on this storage type*".

Color	Supported?
Y	Yes
N	No
NA	Not Applicable

9.7.7.1.2 Oracle database editions

Delphix supports the following Oracle database editions:

- Enterprise
- Standard

9.7.7.1.3 Red Hat Enterprise Linux (RHEL)

Support for RHEL 8.x requires installing the `libncurses.5` library onto the host. Please refer to KBA 5622[430] for the required steps.

Supported OS version	Supported DBMS version					
	11gR2	12cR1	12cR2	18c	19c	21c
RHEL 5.5	Y	NA	NA	NA	NA	NA
RHEL 5.6	Y	Y	NA	NA	NA	NA

430 https://support.delphix.com/Delphix_Virtualization_Engine/Delphix_Admin/Environment_Operations_Fail_With_%22bash%3A_error_while_loading_shared_libraries%3A_libncurses.so.5%3A_cannot_open_shared_object_file%3A_No_such_file_or_directory%22_(KBA5622)

RHEL 5.7	Y	Y	NA	NA	NA	NA
RHEL 5.8	Y	Y	NA	NA	NA	NA
RHEL 5.9	Y	Y	NA	NA	NA	NA
RHEL 5.10	Y	Y	NA	NA	NA	NA
RHEL 6.0	Y	Y	NA	NA	NA	NA
RHEL 6.1	Y	Y	NA	NA	NA	NA
RHEL 6.2	Y	Y	NA	NA	NA	NA
RHEL 6.3	Y	Y	NA	NA	NA	NA
RHEL 6.4	Y	Y	Y	Y	NA	NA
RHEL 6.5	Y	Y	Y	Y	NA	NA
RHEL 6.6	Y	Y	Y	Y	NA	NA
RHEL 6.7	Y	Y	Y	Y	NA	NA
RHEL 6.8	Y	Y	Y	Y	NA	NA
RHEL 6.9	Y	Y	Y	Y	NA	NA
RHEL 6.10	Y	Y	Y	Y	NA	NA
RHEL 7.0	Y	Y	Y	Y	NA	NA
RHEL 7.1	Y	Y	Y	Y	NA	NA
RHEL 7.2	Y	Y	Y	Y	NA	NA
RHEL 7.3	Y	Y	Y	Y	NA	NA

RHEL 7.4	Y	Y	Y	Y	Y	NA
RHEL 7.5	Y	Y	Y	Y	Y	NA
RHEL 7.6	Y	Y	Y	Y	Y	NA
RHEL 7.7	Y	Y	Y	Y	Y	NA
RHEL 7.8	Y 6.0.2+	Y 6.0.2+	Y 6.0.2+	Y 6.0.2+	Y 6.0.2+	NA
RHEL 7.9	Y 6.0.4+	Y 6.0.4+	Y 6.0.4+	Y 6.0.5+	Y 6.0.4+	NA
RHEL 8.0	N	N	N	NA	Y 6.0.3+	NA
RHEL 8.1	N	Y 6.0.3+	N	NA	Y 6.0.3+	NA
RHEL 8.2	Y 6.0.3+	Y 6.0.3+	Y 6.0.3+	NA	Y 6.0.3+	NA
RHEL 8.3	Y 6.0.7+	Y 6.0.7+	Y 6.0.7+	NA	Y 6.0.7+	Y 6.0.11+
RHEL 8.4	Y 6.0.10+	Y 6.0.10+	Y 6.0.10+	NA	Y 6.0.10+	Y 6.0.11+
RHEL 8.5	Y 6.0.14+	Y 6.0.14+	Y 6.0.14+	NA	Y 6.0.14+	Y 6.0.14+
RHEL 8.6	Y 6.0.15+	Y 6.0.15+	Y 6.0.15+	NA	Y 6.0.15+	Y 6.0.15+
RHEL 8.7	Y 7.0.0+	Y 7.0.0+	Y 7.0.0+	NA	Y 7.0.0+	Y 7.0.0+
RHEL 8.8	Y 12.0.0+	Y 12.0.0+	Y 12.0.0+	NA	Y 12.0.0+	Y 12.0.0+
RHEL 8.9	Y 19.0.0+	Y 19.0.0+	Y 19.0.0+	NA	Y 19.0.0+	Y 19.0.0+
RHEL 9.0	N	N	N	N	Y 20.0.0+	N
RHEL 9.3	N	N	N	N	Y 20.0.0+	N

9.7.7.1.4 SUSE Linux Enterprise Server (SLES)

Support for SLES 15 requires installing the `libncurses.5` library onto the host. Please reference KBA 5622[431] for actionable steps.

Supported OS version	Supported DBMS version					
	11gR2	12cR1	12cR2	18c	19c	21c
SLES 11	Y	NA	NA	NA	NA	NA
SLES 11 SP1	Y	NA	NA	NA	NA	NA
SLES 11 SP 2	Y	Y	NA	NA	NA	NA
SLES 11 SP 3	Y	Y	NA	NA	NA	NA
SLES 11 SP 4	Y	Y	NA	NA	NA	NA
SLES 12	Y	Y	NA	NA	NA	NA
SLES 12 SP 1	Y	Y	Y	Y	NA	NA
SLES 12 SP 2	Y	Y	Y	Y	NA	NA
SLES 12 SP 3	Y	Y	Y	Y	Y	NA

431 https://support.delphix.com/Delphix_Virtualization_Engine/Delphix_Admin/Environment_Operations_Fail_With_%22bash%3A_error_while_loading_shared_libraries%3A_libncurses.so.5%3A_cannot_open_shared_object_file%3A_No_such_file_or_directory%22_(KBA5622)

SLES 12 SP 4	Y	Y	Y	Y	Y	NA
SLES 12 SP 5	NA	Y 6.0.4+	Y 6.0.4+	N	Y 6.0.4+	NA
SLES 15	N	N	Y	Y	Y	NA
SLES 15 SP 1	N	N	N	N	Y	Y 6.0.11+
SLES 15 SP 2	N	N	N	N	Y 6.0.11+	Y 6.0.11+
SLES 15 SP 3	N	N	N	N	Y 6.0.11+	Y 6.0.11+
SLES 15 SP5	N	N	N	N	Y 15.0.0+	Y 15.0.0+

9.7.7.1.5 Solaris SPARC

Supported OS version	Supported DBMS version					
	11gR2	12cR1	12cR2	18c	19c	21c
Solaris 10 U9[1]	Y	NA	NA	NA	NA	NA
Solaris 10 U10[1]	Y	Y	NA	NA	NA	NA
Solaris 10 U11	Y	Y	N	Y	NA	NA
Solaris 11	Y	Y	NA	NA	NA	NA
Solaris 11 U1	Y	Y	NA	NA	NA	NA
Solaris 11 U2	Y	Y	Y	Y	NA	NA

Solaris 11 U3	Y	Y	Y	Y	Y	NA
Solaris 11 U4	Y	N	Y	Y	Y	NA

1 - Solaris 10 U9 and U10 require libc.so.1 version 1.22.7 or newer. This version of libc.so.1 can be found in Solaris 10 SPARC kernel patch 144500, or a newer kernel patch.

9.7.7.1.6 Solaris x86

Supported OS version	Supported DBMS version					
	11gR2	12cR1	12cR2	18c	19c	21c
Solaris 10 U9[1]	Y	NA	NA	NA	NA	NA
Solaris 10 U10[1]	Y	Y	NA	NA	NA	NA
Solaris 10 U11	Y	Y	NA	Y	NA	NA
Solaris 11	Y	Y	NA	NA	NA	NA
Solaris 11 U1	Y	Y	NA	NA	NA	NA
Solaris 11 U2	Y	Y	Y	Y	NA	NA
Solaris 11 U3	N	Y	Y	Y	Y	NA
Solaris 11 U4	N	N	Y	Y	Y	NA

[1] - Solaris 10 U9 and U10 require libc.so.1 version 1.22.7 or newer. This version of libc.so.1 can be found in Solaris 10 X86 kernel patch 144501, or a newer kernel patch.

9.7.7.1.7 Hewlett Packard Unix (HP-UX)

Supported OS version	Supported DBMS version		
	11gR2	12cR1	12cR2
HP-UX 11.31	Y	Y	Y

9.7.7.1.8 Advanced Interactive eXecutive (AIX)

Supported OS version	Supported DBMS version					
	11gR2	12cR1	12cR2	18c	19c	21c
AIX 6.1	Y	Y	N	N	NA	NA
AIX 7.1	Y	Y	Y	Y	Y	NA
AIX 7.2	Y	Y	Y	Y	Y	NA

Required HP-UX patch for Target Servers
PHNE_37851 - resolves a known bug in HP-UX NFS client prior to HP-UX 11.31.

9.7.7.2 Requirements for Oracle hosts and databases

9.7.7.2.1 Oracle hosts and databases

To begin using Oracle environments in Delphix, reference this article as a provided outline of the configuration requirements for the source and target hosts.

On each host with Oracle, there must be an operating system user configured to the required specifications for Delphix, as explained in the table below. This user (i.e. delphix_os) can easily be created by using the createDelphixOSUser.sh script (located at the bottom of the page).

These requirements apply to both source and target environments. However, target environments have additional requirements which are detailed in the 'Target Host Requirements' section below.

9.7.7.2.2 Source host requirement

Host requirement	Explanation
Profile and privileges should be the same as the Oracle user (e.g. oracle) on the host.	For example, delphix_os should have the same environment variable settings ($PATH, $ORACLE_HOME, etc.), umask, and ulimit settings, as the user oracle.
The Delphix software owner account (e.g. delphix_os) must have the same primary OS group as the Oracle software owner account (e.g. oracle).	Delphix recommends giving the delphix_os user the same primary OS group as the Oracle home owner. This ensures the Delphix engine can fully and automatically discover Oracle homes, databases, and listeners. Often, this is an OS group named oinstall. However, the oinstall group is not always necessary depending on your Oracle configuration. This user requires access to the libobk_proxy.so library in the toolkit for the child processes triggered by RMAN.
The delphix_os user must have the Oracle OSDBA group (typically dba) as a primary or secondary OS group.	The OSDBA group allows "OS authentication" when connecting to an Oracle database instance by specifying neither username or password (i.e. rman target /), thus eliminating the need to store or retrieve an SYSDBA password. **Oracle 12c** For Oracle 12c and later versions of Oracle, the delphix_os user can also use OSBACKUPDBA as its secondary group. This is typically the BACKUPDBA group on the host.
For secondary group requirements, the Delphix OS user must be a member of the SYSBACKUP OS group (12.1 or higher) or the SYSDBA OS group (11.2 and lower).	This ensures that the Delphix engine is able to take snapshots of source databases using RMAN.

Host requirement	Explanation
There must be a directory on the source host where the Delphix Engine Toolkit can be installed, for example: /var/opt/delphix/toolkit	The delphix_os user and primary OS group (i.e. oinstall) must own the directory. The directory must have at least permissions -rwxr-x-- (0750) The delphix_os user must have -rwxr-x-- (0750) permissions on each directory in the path leading to the toolkit directory. At least 1.6 GB of storage is needed at the time of setting up the environment and at least 500MB of free space is required to allow refreshes and maintenance of the toolkit, especially during upgrades.
The Delphix Engine must be able to make an SSH connection to the source host (typically port 22).	
Read access to $ORACLE_HOME and all underlying files and directories.	Delphix needs to run locally available Oracle tools such as sqlplus and RMAN. Those executables, as well as various required libraries, reside inside the Oracle home
Bash shell should be installed and available for linux based host.	Delphix requires the ability to run commands using the Bash shell. For the engine to operate with the host, Bash must be available and included in the system path, allowing for the execution of the 'bash' command. Most of the Linux machine bash is available by default and user may not need to do anything.

9.7.7.2.3 Additional requirements for RAC environments

If the source host is part of a RAC cluster, Delphix will attempt to use all nodes and crsctl for its operations.

RAC environment requirement	Explanation
The delphix_os user and Delphix Toolkit configuration must be the same on each node in the RAC cluster	• delphix_os must have a execute permission on crsctl and srvctl on each node in the cluster. • access to olsnodes is needed. • The Delphix Toolkit must be installed in the same directory path on each of the nodes in the source cluster. • All data files, archive logs, and database control file must be located on storage shared by all of the cluster nodes. • Each node in the cluster must be able to access archive logs and the database control file from all other nodes. • At least 1.6 GB of storage is needed at the time of setting up the environment and at least 500MB of free space is required to allow refreshes and maintenance of the toolkit, especially during upgrades.

Masked Provisioning

Masked Provisioning is supported on Oracle RAC only when used with "script-based masking".

Auto-Discovery Requirements

The preferred way to find source databases it to allow Delphix to automatically discover your Oracle Homes and Databases by examining the inventory and oratab files and the listener setup. Successful autodiscovery requires read access to these and related files.

In **most** environments, delphix_os group membership is sufficient to perform auto-discovery.

However, if you have overridden Oracle's group permission structure, you may need to modify privileges to allow auto-discovery.

Auto-discovery requirements	Explanation
The ORATAB file must exist (typically in /etc/oratab or /var/opt/oracle/oratab) and be readable by delphix_os. Read access to: • /etc/orainst.loc or /var/opt/oracle/orainst.loc. • the Oracle inventory file (inventory.xml)	The ORATAB is used to determine the location of the Oracle installations on your host. Note: The Oracle inventory file is identified by the contents of orainst.loc (for example, $INVENTORY_HOME/ContentsXML/inventory.xml).

Auto-discovery requirements	Explanation
Permission to run pargs on Solaris hosts and ps on AIX, HP-UX, and Linux hosts, as super-user. This permission is usually granted via sudo authorization of the commands.	Unless you have used a custom TNS_ADMIN setting, elevated access to ps (pargs on Solaris) is not required See the topic Sudo Privilege Requirements for Oracle Environments (see page 1405) for further explanation of this requirement, and Sudo File Configuration Examples for Oracle Environments (see page 1409) for examples of file configurations

9.7.7.2.4 Source database requirements

For each source database, there are specific configurations required for Delphix to ingest data.

Source database requirement	Explanation
For each Oracle Home, the delphix_os user should have execute permission for the programs in $ORACLE_HOME/bin.	If symlinks are configured, Delphix must be configured with the same $ORACLE_HOME path as was used when starting the instance. Ensure the PermitUserEnvironment configuration parameter = "yes" in the sshd_config file The $ORACLE_HOME/bin/oracle executable must have the SETUID and SETGID flags set. Permissions on the oracle binary must be at least -rwsr-s–x (06751).
Source databases must be in ARCHIVELOG mode to ensure that redo logs are archived.	Archive logs are required to make SnapSyncs consistent and provisionable.
Enable Block Change Tracking (BCT). (Highly Recommended).	Enabling BCT will improve SnapSync operation time. Without BCT, incremental SnapSyncs must scan the entire database.
Enable FORCE LOGGING. (Highly Recommended)	If you do not enable FORCE LOGGING and NOLOGGING operations take place, you will get a Fault from the Delphix Engine. If you must use NOLOGGING to meet specific performance criteria, take a new snapshot of the source database after doing the NOLOGGING operations to bring the dSource up-to-date before provisioning VDBs.

Source database requirement	Explanation
If the online redo log files are located on RAW or ASM devices, then LogSync can only operate in Archive Only mode.	See the topics Advanced Data Management Settings for Oracle dSources (see page 2005) and Linking Oracle Physical Standby Databases (see page 1466) for more information.

9.7.7.2.5 Operating system specific requirements

Solaris

On a Solaris host, gtar must be installed. Delphix uses gtar to handle long file names when extracting the toolkit files into the toolkit directory on a Solaris host. The gtar binary should be installed in one of the following directories:

- / bin:/usr
- / bin:/sbin:/usr
- / sbin:/usr/contrib
- / bin:/usr/sfw
- / bin:/opt/sfw
- / bin:/opt/csw/bin

9.7.7.2.6 Additional target /staging host requirements

This topic describes the user privileges and environment requirements that are required for Oracle target hosts and databases collectively referred to as target environments.

These are in addition to the 'Oracle Hosts and Database Requirements' called out above.

Target/Staging host requirement	Explanation
The Delphix OS user must be a member of the SYSDBA OS group.	This ensures that the Delphix engine is able to create new VDBs on target hosts.
There must be a directory (e.g. "/mnt/provision/" or "/mnt/staging/") that will be used as a container for the NFS mount points that are created when provisioning a VDB or linking a staging push dSource.	The delphix_os user and primary OS group (i.e. oinstall or dba) must own the directory. The directory must have at least permissions -rwxrwx--- (0770). There must be no symbolic links in the path of this directory, because NFS can mount into a directory with symlinks in its path, but cannot unmount.

Target/Staging host requirement	Explanation
Permission to run: • mount, umount, mkdir, rmdir as super-user. • pargs on Solaris hosts • ps on Linux, AIX, and HP-UX as super-user. • nfso on AIX hosts as super-user	The following permissions are usually granted via sudo authorization of the commands. See Sudo Privilege Requirements for Oracle Environments (see page 1405) for further explanation of the commands, and Sudo File Configuration Examples for Oracle Environments (see page 1409) for examples of the /etc/sudoers file on different operating systems.
Optional write permission to the $ORACLE_BASE_CONFIG/dbs directory	If write permission is granted, an instance init file will be written to this directory during provisioning that will simplify manual instance startup. If write permission is not granted, manual instance startup must specify the instance init file. See Manually Starting a VDB (see page 1539)
An Oracle listener process running on the target host for provisioning a VDB	The listener's version should be equal to or greater than the highest Oracle version that will be used to provision a VDB.
Required packages on target hosts: 1. portmapper / rpcbind 2. status daemon (rpc.statd) 3. NFS lock manager (rpc.lockd/lockmgr)	As the Delphix Engine uses NFSv3 for mounting target host filesystems, the prerequisite packages to support NFSv3 client communication are required for normal operation, and the required services to support NFS client communications (including file locking) must be running. NFSv3 is enabled by default and to enable NFSv4, see NFSv4 Configuration (see page 746)
If using ASM, the delphix_os user must have the Oracle ASM ownership groups as secondary OS groups (typically asmadmin and asmdba)	This ensures the Delphix engine can fully and automatically discover Oracle homes, databases, and listeners.
Optional permission to run `$ORACLE_HOME/OPatch/datapatch` as Oracle.	This is required to invoke Datapatch command with any environment user other than the **Oracle** user.

9.7.7.2.7 Scripts

- createDelphixOSUser.sh [432]
- createDelphixDBUser.sh [433]

9.7.7.3 Network and connectivity requirements for Oracle environments

9.7.7.3.1 General port allocation

The Delphix Engine makes use of the following network ports regardless of the type of database platform:

432 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149532270?preview=%2F149532270%2F149532293%2FcreateDelphixOSUser.sh
433 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149532270?preview=%2F149532270%2F149532296%2FcreateDelphixDBUser.sh

9.7.7.3.2 General outbound from the Delphix engine port allocation

Protocol	Port numbers	Use
TCP	25	Connection to a local SMTP server for sending email
TCP/UDP	53	Connections to local DNS servers
UDP	123	Connection to an NTP server
UDP	162	Sending SNMP TRAP messages to an SNMP Manager
TCP	443	HTTPS connections from the Delphix Engine to the Delphix Support upload server
TCP/UDP	636	Secure connections to an LDAP server
TCP	8415	Connections to a Delphix replication target. See Configuring Replication (see page 2381)
TCP	50001	Connections to source and target environments for network performance tests.

9.7.7.3.3 General inbound to the Delphix engine port allocation

Protocol	Port number	Use
TCP	22	SSH connections to the Delphix Engine
TCP	80	HTTP connections to the Delphix GUI
UDP	161	Messages from an SNMP Manager to the Delphix Engine

Protocol	Port number	Use
TCP	443	HTTPS connections to the Delphix Management Application
TCP	8415	Delphix Session Protocol connections from all DSP-based network services including Replication, SnapSync for Oracle, V2P, and the Delphix Connector.
TCP	50001	Connections from source and target environments for network performance tests via the Delphix CLI.

9.7.7.3.4 Firewalls and intrusion detection systems (IDS)

Production databases on source environments (for dSources) are often separated from the non-production environment by firewalls. Firewalls can add milliseconds to the latency between servers. Accordingly, for best performance, there should be no firewalls between the Delphix Engine and the virtual database (VDB) target environments. If the Delphix Engine is separated from a source environment by a firewall, the firewall must be configured to permit network connections between the Delphix Engine and the source environments for the application protocols (ports) listed above.

Intrusion detection systems (IDSs) should also be made permissive to the Delphix Engine deployment. IDSs should be made aware of the anticipated high volumes of data transfer between dSources and the Delphix Engine.

9.7.7.3.5 SSHD configuration

Both source and target Unix environments are required to have `sshd` running and configured such that the Delphix Engine can connect over `ssh`.

The Delphix platform expects to maintain long-running, highly performant `ssh` connections with remote Unix environments. The following `sshd` configuration entries can interfere with these `ssh` connections and are therefore disallowed:

Disallowed `sshd` **configuration entries**
`ClientAliveInterval`
`ClientAliveCountMax`

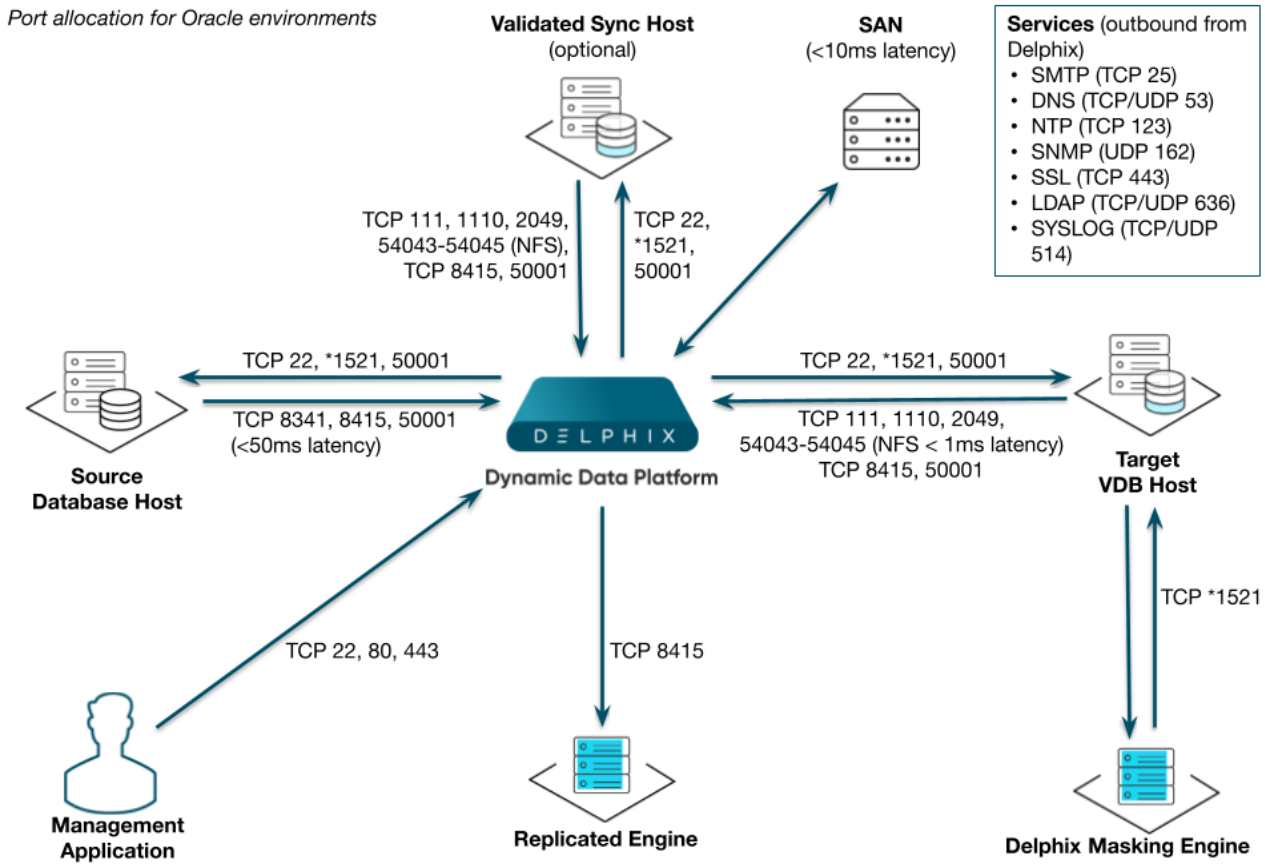
9.7.7.3.6 Network and connectivity requirements for Oracle

- IP connections must exist between the Delphix Engine and the source and target environments.
- For source environments, Delphix Engine uses an **SSH** connection to each source host, an **HTTP** connection from each source environment to Delphix Engine, and a DSP connection to the Delphix Engine. The Delphix Engine uses **SQL*Net*** *connections to the DBMS on the source environment.*
- *For target environments, Delphix uses an **SSH** connection to each target environment and an **NFS** connection to Delphix Engine. Delphix Engine uses **SQL*Net*** connections to the virtual databases on the target environment.
- A TCP Connection is opened between Oracle and the Delphix Connector Process on random high ports (above 1024) on localhost (127.0.0.1).

scp Availability
The **scp** program must be available in the environment in order to add an environment.

9.7.7.3.7 Port allocation for Oracle environments

The following diagram describes the port allocations for Oracle environments. It illustrates the ports that we recommend to be open from Delphix to remote services, to the Delphix Engine, and to the Target Environments.



Note: *Oracle listener typically runs on TCP port 1521. In cases where other ports are used, substitute for 1521 above. Starting with 6.0.7.0 release, this port is required only if database user credentials are provided. The port need not be open if no database user credentials are provided.

The Delphix Engine makes use of the following network ports for Oracle dSources and VDBs:

9.7.7.3.7.1 Outbound from the Delphix engine port allocation

Protocol	Port numbers	Use
TCP	22	SSH connections to source and target environments
TCP	xxx	Connections to the Oracle SQL*Net Listener on the source and target environments (typically port 1521)

9.7.7.3.7.2 Inbound to the Delphix engine port allocation

Protocol	Port number	Use
TCP/UDP	111	Remote Procedure Call (RPC) port mapper used for NFSv3 mounts Note: RPC calls in NFSv3 use additional fixed ports for supporting services (lockd, mountd and statd) seen below.
TCP	1110	NFSv3 Server daemon status and NFS server daemon keep-alive (client info)
TCP/UDP	2049	NFS Server daemon from VDB to the Delphix Engine (used for both NFSv3 and NFSv4)
TCP	8341	Sending data from source to the Delphix Engine (for LogSync)
TCP	8415	SnapSync control and data from source to the Delphix Engine V2P control and data from the target environment to the Delphix Engine.
TCP	54043	NFSv3 mount daemon
TCP	54044	NFSv3 stat daemon (lock state notification service)
TCP	54045	NFSv3 lock daemon/manager
TCP	54046	Connections from Source and Target Environments to the Engine When NFS Encryption is enabled
UDP	33434 - 33464	Traceroute from source and target database servers to the Delphix Engine (optional)

9.7.7.3.8 AppData port requirements

The use of AppData requires the following ports/protocols. Two important notes about these specifications:

1. The next release of the Delphix Engine will significantly augment the port/protocol utilization of AppData. The upcoming-only requirements have been marked with a *.
2. AppData V2P uses RSYNC to export to the target. RSYNC between the target and Delphix Engine is not required for general virtualization usage. The V2P-only requirements have been marked with a ^.

From source to Delphix engine	From Delphix engine to source	From target to Delphix engine	From Delphix engine to target
RSYNC to rsyncd[434] (TCP Port 873)	RSYNC to rsyncd[435] (TCP Port 873)	DSP (Default TCP Port 8415)	DSP (Default TCP Port 8415)
DSP (Default TCP Port 8415)	SSH (TCP Port 22)	NFS	SSH (TCP Port 22)
*NFS	DSP (Default TCP Port 8415)	^RSYNC to rsyncd[436] (TCP Port 873)	^RSYNC to rsyncd[437] (TCP Port 873)

9.7.7.4 Sudo privilege requirements for Oracle environments

This topic describes the rationale behind specific `sudo` privilege requirements for virtualizing Oracle Databases.

434 https://manpages.ubuntu.com/manpages/focal/man5/rsyncd.conf.5.html
435 https://manpages.ubuntu.com/manpages/focal/man5/rsyncd.conf.5.html
436 https://manpages.ubuntu.com/manpages/focal/man5/rsyncd.conf.5.html
437 https://manpages.ubuntu.com/manpages/focal/man5/rsyncd.conf.5.html

Privilege	Sources	Targets	Rationale
`ps | pargs`	Optional, Strongly Recommended	Optional, Strongly Recommended	Delphix auto-discovery uses the `TNS_ADMIN` environment variable of Oracle Listener processes with non-standard configurations to derive their connection parameters. A different user (oracle), rather than **delphix_os** user normally owns the oracle listener. The Delphix Engine needs `sudo` access to `pargs` on the Solaris OS or `ps` on other OSes to determine the environment variables of those Listener processes. This privilege is required for Auto-Discovery with non-default `TNS_ADMIN` locations. It is optional when using a standard `TNS_ADMIN` location, or if you choose to manually configure Oracle Homes, databases and listeners.
`mount/umount`	Not Required	**Required**	The Delphix Engine dynamically mounts and unmounts directories under the provisioning directory during VDB operations. This privilege is required because `mount` and `umount` are typically reserved for superuser.
`nfso` (AIX only)	Not Required	**Required**	The Delphix Engine monitors NFS read and write sizes on an AIX target host. It uses the `nfso` command to query the sizes in order to optimize NFS performance for VDBs running on the target host. Only a superuser can issue the `nfso` command.
`$ORACLE_HOME/OPatch/datapatch`	Not Required	Optional	This is required to invoke Datapatch command with any environment user other than the **Oracle** user.

It is required to specify the NOPASSWD qualifier within the "sudo" configuration file, as shown here: Sudo File Configuration Examples for Oracle Environments. This ensures that the "sudo" command does not demand the entry of a password, even for the "display permissions" (i.e. "sudo -l") command.
Delphix issues "sudo -l" in some scripts to detect if the operating system user has the correct sudo privileges. If it is unable to execute this command, some actions may fail and Delphix will raise an alert suggesting it does not have the correct sudo permissions. Restricting the execution of "sudo -l" by setting “listpw=always” in the “/etc/sudoers” file ***when the Delphix operating system user is configured to use public key authentication*** will cause the Delphix operating system user to be prompted for a password which will fail certain Delphix actions. Use a less restrictive setting for listpw than "always" when the Delphix operating system user is using public-key authentication.

Oracle mount options for RAC	
AIX	`cio,rw,bg,hard,nointr,timeo=600,proto=tcp,nosuid,noac`
HPUX	`rw,bg,hard,rsize=1048576,wsize=1048576,nointr,timeo=600,proto=tcp,nosuid,forcedirectio,noac`
Solaris	`rrw,bg,hard,rsize=1048576,wsize=1048576,nointr,proto=tcp,nosuid,forcedirectio,noac`
For the above platforms, depending on NFS version used, additional options vers=3 or vers=4.x is added (x varies depending on what that platform supports. e.g. vers=4 or vers=4.1)	
Linux (NFSv3)	`-t nfs rw,bg,hard,rsize=1048576,wsize=1048576,nointr,timeo=600,tcp,nosuid,sec=sys,vers=3,actimeo=0`
Linux (NFSv4)	`-t nfs4 rw,bg,hard,rsize=1048576,wsize=1048576,nointr,timeo=600,tcp,nosuid,sec=sys,actimeo=0`

1. For both Single instance and RAC, "port=2049" option is added for all platforms.
2. For AIX, rsize=<value>,wsize=<value> options are added depending on the value returned by "/usr/sbin/nfso -o nfs_max_read_size" and "/usr/sbin/nfso -o nfs_max_write_size" commands.

Oracle unmount options	"-f" is used for all platforms. For Linux, "-lf" is used.

Oracle mount options for single instance	
AIX	`cio,rw,bg,hard,intr,timeo=600,proto=tcp,nosuid`
HPUX	`rw,bg,hard,rsize=1048576,wsize=1048576,nointr,timeo=600,proto=tcp,nosuid,forcedirectio`
Solaris	`rw,bg,hard,rsize=1048576,wsize=1048576,nointr,proto=tcp,nosuid,forcedirectio`
For the above platforms, depending on NFS version used, additional options vers=3 or vers=4.x is added (x varies depending on what that platform supports. e.g. vers=4 or vers=4.1)	
Linux (NFSv3)	`-t nfs rw,bg,hard,rsize=1048576,wsize=1048576,nointr,timeo=600,tcp,nosuid,sec=sys,vers=3`
Linux (NFSv4)	`-t nfs4 rw,bg,hard,rsize=1048576,wsize=1048576,nointr,timeo=600,tcp,nosuid,sec=sys`

9.7.7.5 Sudo file configuration examples for Oracle environments

This topic provides a sample `sudo` file privilege configurations for using the Delphix Engine with various operating systems and the Oracle RDBMS.

> **Requiretty settings**
> Delphix requires that the `requiretty` setting be disabled for all Delphix users with `sudo` privileges.

9.7.7.5.1 Configuring `sudo` access on Solaris SPARC for Oracle source and target environments

Sudo access to `pargs` on the Solaris operating system is required for the detection of listeners with non-standard configurations on both source and target environments. Super-user access level is needed to determine the `TNS_ADMIN` environment variable of the user running the listener (typically **oracle**, the installation owner). From `TNS_ADMIN`, the Delphix OS user **delphix_os** can derive connection parameters.

Example: Solaris `/etc/sudoers` entries for a Delphix Source

```
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD:/usr/bin/pargs
```

On a Solaris target, `sudo` access to `mount` and `umount` is also required.

Example: Solaris `/etc/sudoers` entries for a Delphix Target

```
# Delphix issues sudo -l so we need to allow it via listpw. Never set it to always
when using public key authentication
Defaults      listpw=all

User_Alias DELPHIX_USER=delphix_os

Cmnd_Alias DELPHIX_CMDS= \
/usr/bin/mount, \
/usr/bin/umount, \
/usr/bin/mkdir, \
/usr/bin/rmdir, \
/usr/bin/pargs

DELPHIX_USER ALL=(ALL) NOPASSWD: DELPHIX_CMDS
```

9.7.7.5.2 Configuring `sudo` access on Linux for Oracle source and target environments

Sudo access to `ps` on the Linux operating system is required for the detection of listeners with non-standard configurations on both source and target environments. Super-user access level is needed to determine the `TNS_ADMIN` environment variable of the user running the listener (typically **oracle**, the installation owner). From `TNS_ADMIN`, the Delphix OS user **delphix_os** can derive connection parameters.

Example: Linux `/etc/sudoers` entries for a Delphix Source

```
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD:/bin/ps
```

On a Linux target, sudo access to `mount` and `umount` is also required.

Example: Linux `/etc/sudoers` file for a Delphix Target

```
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD: \
/bin/mount, /bin/umount, /bin/mkdir, /bin/rmdir, /bin/ps
```

9.7.7.5.3 Configuring `sudo` access on AIX for Oracle source and target environments

Sudo access to `ps` on the AIX operating system is required for the detection of listeners with non-standard configurations on both source and target environments. Super-user access level is needed to determine the `TNS_ADMIN` environment variable of the user running the listener (typically **oracle**, the installation owner). From `TNS_ADMIN`, the Delphix OS user **delphix_os** can derive connection parameters.

Example: AIX `/etc/sudoers` entries for a Delphix Source

```
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD:/bin/ps
```

In addition to sudo access to the `mount`, `umount`, and `ps` commands on AIX target hosts, Delphix also requires `sudo` access to `nfso`. This is required on target hosts for the Delphix Engine to monitor the NFS read-write sizes configured on the AIX system. Super-user access level is needed to run the `nfso` command.

Example: AIX `/etc/sudoers` File for a Delphix Target

```
Defaults:delphix_os !requiretty
```

```
delphix_os ALL=NOPASSWD: \
/usr/sbin/mount, \
/usr/sbin/umount, \
/usr/sbin/mkdir, \
/usr/sbin/rmdir, \
/usr/sbin/nfso, \
/usr/bin/ps
```

9.7.7.5.4 Configuring `sudo` access on HP-UX for Oracle source and target environments

No `sudo` privileges are required on source environments running HP-UX. The HP-UX OS does not allow the **delphix_os** user to determine the `TNS_ADMIN` environment variable setting for the **oracle** user. This means that the Delphix Engine cannot auto-discover non-standard listener configurations with non-default `TNS_ADMIN` settings.

On the HP-UX target, `sudo` access to `mount` and `umount` is required as with other operating systems.
Example: HP-UX /etc/sudoers file for a Delphix target

Example: AIX `/etc/sudoers` File for a Delphix target

```
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD:/sbin/mount, /sbin/umount, /sbin/mkdir, /sbin/rmdir
```

Examples of Limiting `sudo` Access for the Delphix OS User

In situations where security requirements prohibit giving the Delphix user root privileges to mount, unmount, make a directory, and remove directory on the global level, it is possible to configure the `sudoers` file to provide these privileges only on specific mount points or from specific Delphix Engines, as shown in these two examples.

> The Delphix Engine tests its ability to run the `mount` command using `sudo` on the target environment by issuing the `sudo mount` command with no arguments. Many of the examples shown in this topic do not allow that. This causes a warning during environment discovery and monitoring but otherwise does not cause a problem. If your VDB operations succeed, it is safe to ignore this warning.
> Similarly, the `ps` or `pargs` command is used for target environment operations such as initial discovery and refresh. The most restrictive sudo setups might not allow the commands `ps (pargs)`. Delphix can still function without these privileges, although auto-discovery may not work.
> However, some users configure the security on the target environments to monitor `sudo` failures and lockout the offending account after some threshold. In those situations, the failure of the sudo commands might cause the **delphix_os** account to become locked. One workaround

for this situation is to increase the threshold for locking out the user account. Another option is to modify `/etc/sudoers` to permit the **delphix_os** user to run `ps (pargs)`, and `mount` command without parameters.

Note that the following examples are for illustrative purposes and the sudo file configuration options are subject to change.

9.7.7.5.4.1 Example 1

This example restricts the **delphix_os** user's use of `sudo` privileges to the directory `/oracle`.

Note that wildcards are allowed for the options on `mount` and `umount` because those commands expect a fixed number of arguments after the options. The option wildcard on the `mount` command also makes it possible to specify the file-system being mounted from the Delphix Engine.

Example `/etc/sudoers` File Configuration on the Target Environment for sudo Privileges on the VDB Mount Directory Only (Linux OS)

```
Defaults:delphix_os !requiretty
delphix_os ALL=(root) NOPASSWD: \
/bin/mount  *        /oracle/*, \
/bin/umount *        /oracle/*, \
/bin/umount          /oracle/*, \
/bin/mkdir -p        /oracle/*, \
/bin/mkdir -p -m 755 /oracle/*, \
/bin/mkdir           /oracle/*, \
/bin/rmdir           /oracle/*, \
/bin/ps
```

Example `/etc/sudoers` File Configuration on the Source Environment to grant Super-User privileges when running PS

```
Defaults:delphix_os !requiretty
delphix_os ALL=(root) NOPASSWD: /bin/ps
```

9.7.7.5.4.2 Example 2

This example restricts the **delphix_os** user's use of `sudo` privileges to the directory / `oracle`, restricts the mount commands to a specific Delphix Engine hostname and IP, and does not allow user-specified options for the `umount` command.

This configuration is more secure, but there is a tradeoff with deployment simplicity. This approach would require a different sudo configuration for targets configured for different Delphix Engines.

A Second Example of Configuring the `/etc/sudoers` File on the Target Environment for Privileges on the VDB Mount Directory Only (Linux OS)

```
Defaults:delphix_os !requiretty
delphix_os ALL=(root) NOPASSWD: \
/bin/mount           <delphix-server-name>* /oracle/*, \
/bin/mount *         <delphix-server-name>* /oracle/*, \
/bin/mount           <delphix-server-ip>*   /oracle/*, \
/bin/mount *         <delphix-server-ip>*   /oracle/*, \
/bin/mount "", \
/bin/umount          /oracle/*, \
/bin/umount *        /oracle/*, \
/bin/mkdir, \
/bin/rmdir, \
/bin/ps
```

9.7.7.5.4.3 Example 3

This example adds the following entries in `/etc/sudoers` to allow the **delphix_os** user to run `datapatch` as the user **oracle** without needing to enter the password and with no additional privileges:

```
Cmnd_Alias ORACLE_CMDS = /u01/app/oracle/product/19.7.0.0/dbhome_1/OPatch/datapatch
Defaults!ORACLE_CMDS env_keep += "ORACLE_HOME ORACLE_SID ORACLE_UNQNAME"
delphix_os ALL=(oracle) NOPASSWD: ORACLE_CMDS
```

It is important to note that the environment variables necessary for Delphix (`ORACLE_HOME` , `ORACLE_SID` and `ORACLE_UNQNAME`) to successfully invoke the `datapatch` command must be preserved using the `env_keep` option, otherwise the command invocation may fail.

9.7.8 Managing Oracle environments and hosts

This section describes the attributes of Oracle-specific environments such as RAC Cluster Users, Listeners, and RAC configurations and covers the following topics:

- Managing environment users (see page 1414)
- Environment attributes for hosts with Oracle (see page 1414)
- Adding an Oracle single instance or RAC environment (see page 1417)
- Adding Oracle standalone and RAC targets in Kerberos (see page 1419)
- Adding a database installation home to an Oracle environment (see page 1423)
- Adding a database to an Oracle environment (see page 1424)
- Adding a listener to an Oracle environment (see page 1425)
- Enabling linking and provisioning for Oracle databases (see page 1427)

- Linking an Oracle RAC source database using a standalone host environment (see page 1427)
- Listener and JDBC verification (see page 1429)
- Providing your own Oracle Java (see page 1025)
- Changing the hostname or IP address for Oracle source and target environments (see page 1433)
- How to change ORACLE_HOME (see page 1435)
- CLI support for partial-full backup in Oracle (see page 1436)
- Managing cluster nodes of an Oracle RAC environment (see page 1436)

9.7.8.1 Managing environment users

9.7.8.1.1 Procedure

1. Login to the **Delphix management** application.
2. Click **Manage**.
3. Select **Environments**.
4. Click on the existing **environment name** you want to modify.
5. In the Details tab under **Basic information**, click the **Plus** icon next to Environment Users to add a user.
6. Enter the **Username** and **Password** for the OS user in that environment.
7. Select **Add** to save the new user.
8. To change the primary user for this environment, click the **Pencil** icon next to **Environment users**. Only the primary user will be used for environment discovery.
9. To delete a user, click the **Trash** icon next to their username.

9.7.8.2 Environment attributes for hosts with Oracle

This topic describes the attributes of Oracle-specific environments such as RAC Cluster Users, Listeners, and RAC configurations. You can manage these settings by the procedure described below.

1. Navigate to Manage > Environments.
2. In the Environments panel, click the name of an environment to view its attributes.
3. Next to Attributes, click the Pencil icon to edit an attribute.
    - For Oracle RAC environments select your node then click the Pencil icon.

9.7.8.2.1 Common environment attributes

Attribute	Description
Environment users	The users for that environment. These are the users who have permission to ssh into an environment or access the environment through the Delphix Connector. For more information on the environment user requirements, see the Requirements topics for specific data platforms.
Host address	The IP address of the environment host.
SSH Port	The SSH port number used for the environment.
Login Type	Select from: • Username and Password • Username and Public Key • Password Vault
Toolkit Path	The plugin path for the environment. Delphix uses the Toolkit path (Toolkit is a synonym for Plugin) for logs, mount points, and keeping lib files.
Notes	Any other information you want to add about the environment.

9.7.8.2.2 Oracle environment attributes

Attribute	Description
Environment Name (RAC)	The Environment Name field under Attributes is used to provide the name of the environment host in the case of cluster environments. This field defaults to the IP address of the host unless you specify another name.

Attribute	Description
Cluster Name (RAC)	The name of the RAC cluster as retrieved during auto-discovery from the Oracle Clusterware Repository (OCR).
Cluster Home (RAC)	This is the full pathname of the directory in which the cluster or grid ORACLE_HOME resides, which contains the executables for the Oracle Clusterware. Besides containing the software for the Oracle CRS (Cluster Ready Services), this directory also contains the binaries for Oracle ASM (Automatic Storage Management).
Version	The version of the RAC cluster as retrieved during auto-discovery from the Oracle Clusterware Repository (OCR).
Cluster User (RAC)	The OS user account with read-execute permission to access the ORACLE_HOME for the cluster/grid, as well as the ORACLE_HOME for the database.
NFS Encryption	Enabling encryption for NFS traffic of environment.
Host Address (RAC)	Public IP hostname or IP address for the specific node (or host) within the RAC cluster.
NFS Address	In the event that the network link to which “public hostname” of the RAC nodes reside is to be used only for command-and-control traffic (i.e. SSH, JDBC, etc), and not for network-attached storage, then this field is where the comma-separated list of IP addresses for the alternate network dedicated to NFS for network-attached storage.
Virtual IP (RAC)	Oracle RAC requires setting up virtual IPs to manage failover. This field informs Delphix which IP addresses to use when a failure is detected. Click the + to add another virtual IP domain and IP address. For more details view Overview of Virtual IP Addresses[438]

438 https://docs.oracle.com/database/121/RACAD/GUID-6C72F02D-BB43-4C56-9F46-244C8D6BB670.htm#RACAD8950

Attribute	Description
Listeners	The listener is used to connect incoming client requests to the database. See Adding a Listener to an Oracle Environment (see page 1425) for more information.
Remote Listener	An Oracle TNS name resolves to an address or address list of Oracle Net remote listeners. This is the Oracle initialization parameter REMOTE_LISTENER, which must be the same on all RAC cluster instances. Click the + to add a remote listener.
SCAN	Single Client Access Name is used to allow clients to access cluster databases.
SCAN Listener	Listener used with SCAN to establish client connections to the database instances in the RAC cluster.

9.7.8.3 Adding an Oracle single instance or RAC environment

This topic describes how to add a new Oracle or Oracle RAC environment.

9.7.8.3.1 Prerequisites

- See the topics Requirements for Oracle Hosts and Databases (see page 1392)
- There can be one Oracle unique database name (DB_UNIQUE_NAME) per Delphix Engine. For example, if you provision a VDB with a database unique name "ABC" and later try to add an environment that has a source database that also has a database unique name of "ABC", errors will occur.

9.7.8.3.2 Procedure

1. Login to the **Delphix management** application.
2. Click **Manage**.
3. Select **Environments**.
4. Click the **Actions (...)** menu next to **Environments** and select **Add environment.**
5. In the **Host and Server** tab, select **Unix/Linux**.
6. Select **Standalone host** or **Oracle cluster**, depending on the type of environment you are adding.

7. Click **Next**.
8. For standalone Oracle environments: Enter the **Environment name** Enter **Host IP address**
9. For cluster Oracle RAC environments: Enter the **Environment name,** Enter **Cluster home, and** Enter **Node address**
10. For NFS Addresses (Optional): Enter one or more comma-separated **IP Address/Hostname**

    In the case of the Oracle RAC environment, ensure that the NFS Address list includes IP Addresses from all the cluster nodes. NFS address given at the time of environment creation is applied to all the discovered RAC nodes. If some nodes in the cluster need a different IP Address list, users can edit the specific host and update the NFS Address after the environment has been created.

> If specified, Delphix Engine only allows NFS requests (mount, etc) originated from IP Addresses specified for the host.

11. If server authentication for remote host communication or engine to host throughput tests is desired, make sure the appropriate config is set. For more details refer to Configuring Network Security Settings (see page 966). You will need to create a JKS or PKCS#12 keystore on the remote host with the full CA chain of the DSP key in the keystore. By default, the key will just the signed by the Delphix CA, but you can replace the DSP key if you wish. Refer to KeyStore Settings (see page 987) for more details.
12. If client authentication for remote host communication or engine to host throughput tests is also desired, make sure the appropriate config is set. For more details refer to Configuring Network Security Settings (see page 966) You will need to create another JKS or PKCS#12 keystore on the remote host with the desired key pair. Make sure the created keystore has permissions such that it is readable by all environment users. Then, add the full CA chain of the remote host's key pair to the TrustStore on the engine. For more details, refer to TrustStore Settings (see page 989)
13. Enter the SSH port. The default value is 22.
14. Select a Login Type:
    - Username and Password - enter the OS username and password, or
    - Username and Public Key - enter the OS username, or
    - Password Vault - select from an existing Enterprise Password Vault

> **Using Public Key Authentication**
>
> If you want to use public-key authentication for logging into your Unix-based environment, there are two options: use the engine's key pair or provide a key pair for this environment.
>
> To use the engine's key pair:
>
> - Select **Public key** for the **Login type.**

- Click **View public key**.
- Copy the public key that is displayed, and append it to the end of your `~/.ssh/authorized_keys` file. If this file does not exist, you will need to create it.
  - Run chmod 600 ~/.ssh/authorized_keys to allow only the file's owner to read and write to it (make sure the file is owned by the user).
  - Run chmod 755 ~ to restrict access to the user's home directory so no other user may write to it.
  - Run chmod 700 ~/.ssh so that others cannot write to it. The ~/.ssh directory cannot be writable by group or other users. Otherwise, authentication will fail.

As an alternative, you can provide a key pair specific for this environment via the API or CLI. See Option 2 in this CLI Cookbook article (see page 1417) for instructions.

15. For **Password login**, click **Verify credentials** to test the username and password.
16. Enter **Toolkit path**. The toolkit directory stores scripts used for Delphix Engine operations, and should have a persistent working directory rather than a temporary one. The toolkit directory will have a separate subdirectory for each database instance and must have 0770 permissions. At least 1.6 GB of storage is needed at the time of setting up the environment and at least 500MB of free space is required to allow refreshes and maintenance of the toolkit, especially during upgrades.
17. To provide your own Oracle Java select the **Provide my own JDK** checkbox and click **Next**.
18. In the Java Development Kit, tab enter the absolute path to your Oracle JDK and click **Next**. For more information, see OpenJDK on Delphix Engine (see page 1025). Adding an Oracle single instance or RAC environment Procedure
19. Click **Submit**.

9.7.8.3.3 Post-requisites

After you create the environment, you can view information about it:

1. Click **Manage**.
2. Select **Environments**.
3. Select the environment name.

9.7.8.4 Adding Oracle standalone and RAC targets in Kerberos

This topic describes how to add an Oracle standalone and RAC targets in Kerberos.

9.7.8.4.1 Prerequisites

- Configure Kerberos credentials through the System Setup interface.

- Kerberized Oracle host operations are only supported in Delphix Engine release 5.2.3.0 and later.
- You must define only one principle in the Kerberos configuration and this principle must be used for all Kerberized host environments.

9.7.8.4.2 Procedure

When adding a new environment or editing the configuration of an existing environment, it is a general Kerberos authentication requirement that you configure the host address using a fully qualified domain name (FQDN). The login type of Kerberos authentication is available after you apply the Kerberos configuration in the System Setup interface.

In the example below, the Kerberos principal krbuser has previously been configured using System Setup, so this is automatically populated when the radio button is selected.

Perform the following steps to add an Oracle standalone and RAC targets in Kerberos.

1. Login to the Delphix Management application.
2. Click **Manage**.
3. Select **Environments**.
4. Click the **Actions (...)** menu next to **Environments** and select **Add environment**.
5. In the **Host and server** tab, select **Unix/Linux** and click **Next**.
6. In the **Environment settings** tab, do the following:
    a. Enter an environment name and the host address.
    b. Under **Login Type**, select the **Kerberos authentication**radio button.

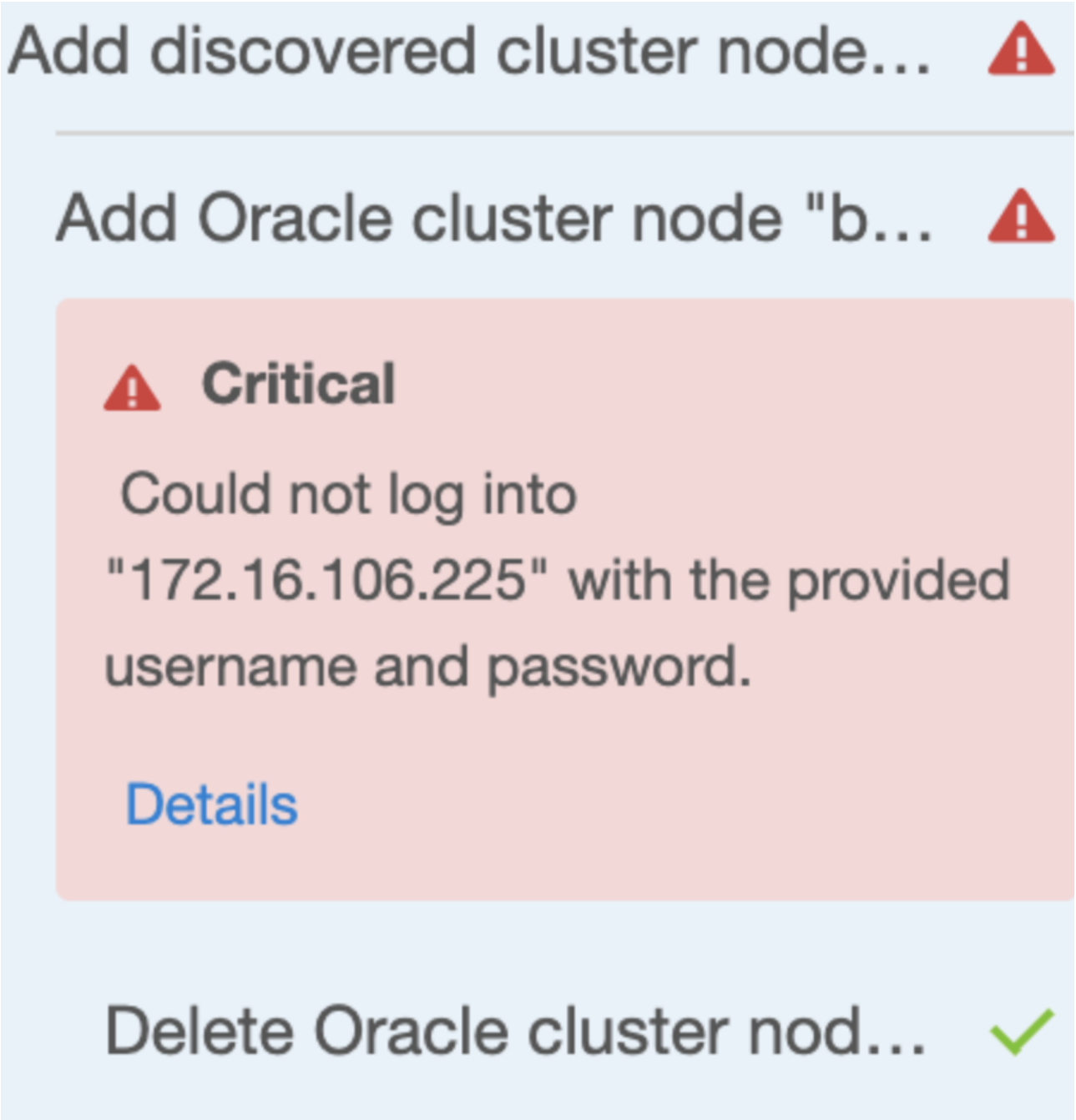
> The Kerberos principal **krbuser** has previously been configured using the System Setup interface, so this is automatically populated when the **Kerberos authentication** radio button is selected.

c. Click Next.

d. The **Summary** tab enables you to review your configurations. Review the configuration and click **Submit**.

9.7.8.4.3 Configuring Oracle RAC using the GUI

As the discovery logic attempts to add other nodes via IP address (based on cluster interrogation results which return the additional node IP addresses), automatic discovery of additional nodes will initially fail, causing Kerberos authentication to fail with "Could not log into <IP_address> with the provided username and password". It is therefore also expected to see one or more failed actions under the Discovery job indicating "Add discovered cluster node" and "Add Oracle cluster node <node name>" were unsuccessful.

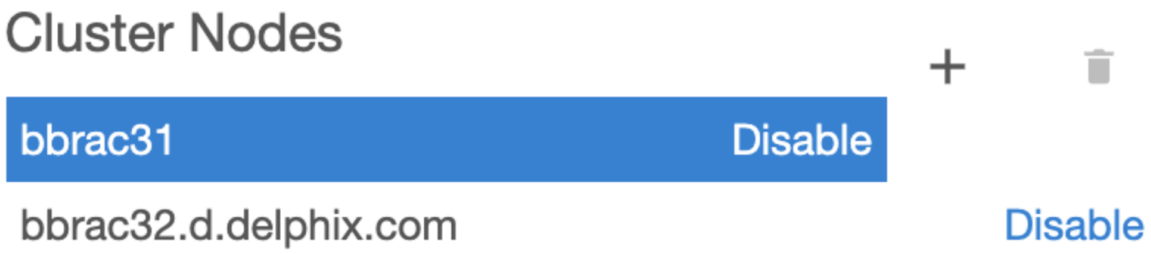


As such, all nodes beyond the first must be added manually. The cluster can be added initially using the GUI or CLI as desired to discover the first node, configured via FQDN. Each additional node is added manually via CLI or GUI.

To add all the additional nodes via GUI, perform the following steps:

1. Login to the Delphix Management application.
2. Click **Manage**.
3. Select **Environments**.

4. Select the cluster node to view the basic information of the respective node.
5. In the **Cluster node** widget, click the plus icon to add a node. In this example, **bbrac31** was discovered normally, and **bbrac32.d.delphix.com** was manually added after the Environment Discovery process was completed.



6. Click the new node in the Cluster Nodes list.
7. Click the plus icon next to the **Virtual IPs** and add the virtual IP name and address for the node.
8. Click the plus icon next to the **Listeners** and add the name and protocol address of a listener on the node. Protocol address should be in the following format. This example uses bbrac15-vip.delphix.com as the host virtual IP address.

```
(ADDRESS=(PROTOCOL=tcp)(HOST=bbrac15-vip.delphix.com)(PORT=1521))
```

9.7.8.4.4 Adding an Oracle RAC cluster using the CLI

You can use the CLI to create the cluster, specifying one of the nodes for discovery. Make sure to set the credential type to KerberosCredential. This procedure uses the following node names bbrac14, bbrac15, and bbrac16 for example purposes. "bbrac14" will be the primary node used for discovery. Make sure that you use an FQDN for the host address (it is a must Kerberos requirement for authentication):

```
delphix.engine> /environment create
delphix.engine environment create *> set type=OracleClusterCreateParameters
delphix.engine environment create *> set cluster.name=bbrac1416
delphix.engine environment create *> set cluster.crsClusterHome=/u01/app/11.2.0/grid
delphix.engine environment create *> set
primaryUser.credential.type=KerberosCredential
delphix.engine environment create *> edit nodes
delphix.engine environment create nodes *> add
delphix.engine environment create nodes 0 *> set name=bbrac14.delphix.com
delphix.engine environment create nodes 0 *> set
hostParameters.host.address=bbrac14.delphix.com
delphix.engine environment create nodes 0 *> set hostParameters.host.toolkitPath=/
work
delphix.engine environment create nodes 0 *> commit
```

Now, use the above procedure described in the **Configuring Oracle RAC using the GUI** section to manually add the node.

9.7.8.5 Adding a database installation home to an Oracle environment

When you add an environment with the Delphix Admin application, all database installation homes on it are automatically discovered. However, if a database installation home is not automatically discovered, you can add it manually to the environment.

9.7.8.5.1 Procedure

1. Login to the **Delphix management** application.
2. Click **Manage**.
3. Select **Environments**.
4. Select an Environment.
5. Click the **Databases** tab**.**
6. Click the **Add dataset home** button.
7. Enter the **Installation home**.
8. Enter the **Version** of the Installation Home.
9. Enter the **Oracle base** of the Installation Home.
10. Enter the **Bits** of the Oracle Home.
11. When finished, click **Add**.

9.7.8.5.2 Troubleshooting

If the environment user has oinstall permissions, Delphix will be able to discover the **Version**, **Oracle Base**, and **Bits**. If any of these fields are found to be different than those provided by the user, a fault will be raised on the repository.

9.7.8.5.2.1 Oracle version

The version can be found in the comps.xml file in $ORACLE_HOME/inventory/ContentsXML/comps.xml. Example of Oracle Home with version 12.1.0.2:

```
<COMP NAME="oracle.server" VER="12.1.0.2.0" BUILD_NUMBER="0" REP_VER="0.0.0.0.0"
 RELEASE="Production" INV_LOC="Components/oracle.server/12.1.0.2.0/1/" LANGS="en"
 XML_INV_LOC="Components21/oracle.server/12.1.0.2.0/" ACT_INST_VER="12.1.0.2.0"
 DEINST_VER="11.2.0.0.0" INSTALL_TIME="2016.Apr.14 12:42:09 PDT" INST_LOC="/u01/app/
oracle/product/12.1.0/dbhome_1/oracle.server">
```

9.7.8.5.2.2 Oracle base

This can be found as a property in $ORACLE_HOME/inventory/ContentsXML/oraclehomeproperties.xml. Example of Oracle Home with Oracle Base "/u03/app/ora11202":

```
<PROPERTY NAME="ORACLE_BASE" VAL="/u03/app/ora11202"/>
```

9.7.8.5.2.3 Bits

This can be found by running

```
file $ORACLE_HOME/bin/rman
```

The output will indicate if the Oracle Home is 32 bit or 64 bit.

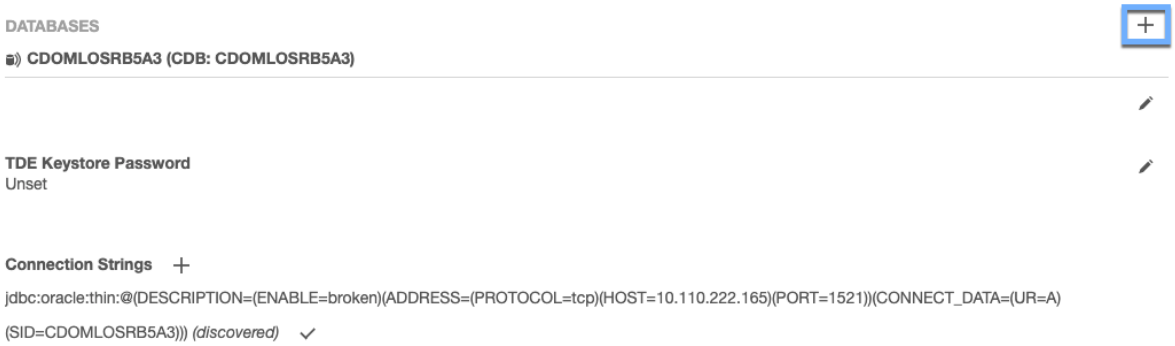
9.7.8.6 Adding a database to an Oracle environment

9.7.8.6.1 Prerequisites

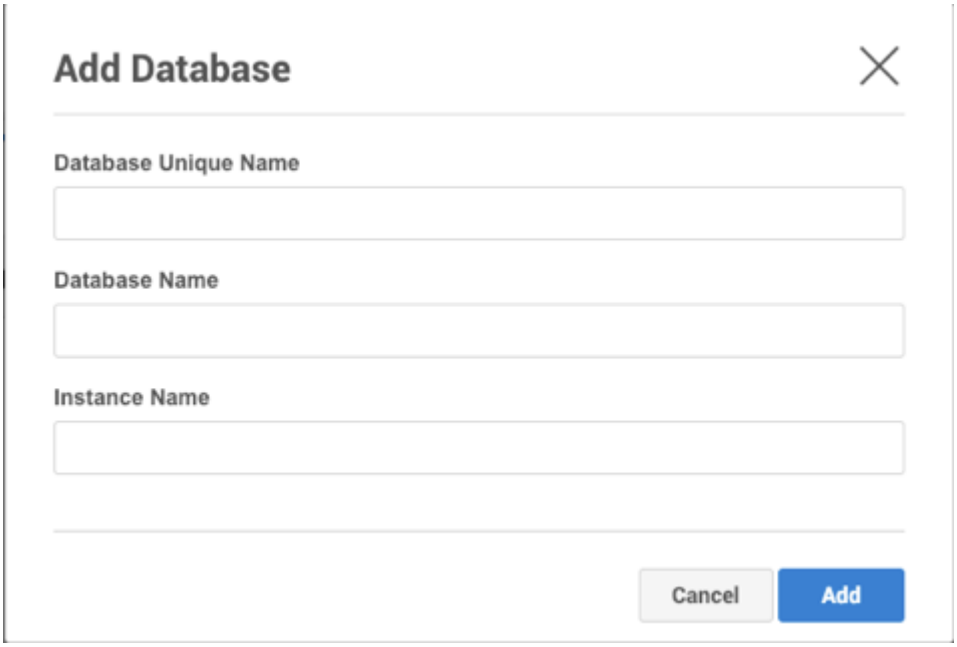
- Make sure your source database meets the requirements described in Requirements for Oracle Hosts and Databases (see page 1392)
- Before adding a database, the installation home of the database must exist in the environment. If the installation home does not exist in the environment, follow the steps in Adding a Database Installation Home to an Oracle Environment (see page 1423)

9.7.8.6.2 Procedure

1. Login to the **Delphix management** application.
2. Click **Manage**.
3. Select **Environments**.
4. Select an Environment.
5. Click the **Databases** tab.
6. Choose the installation home where the database is installed.
7. Click the **Plus** icon.



8. In the **Add Database** dialog box, enter the **Database Unique Name, Database Name** and **Instance Name**.



9. When finished, click **Add**.

9.7.8.7 Adding a listener to an Oracle environment

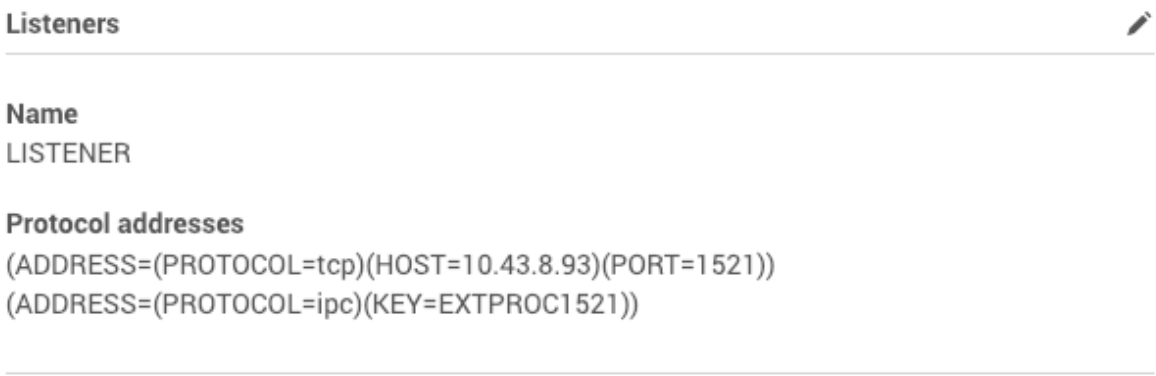
This topic describes how to add listeners for an Oracle environment.

When you add an environment with the Delphix Management application, all listeners that are running on it are automatically discovered. However, if a listener is not automatically discovered, you can add it manually to the environment.

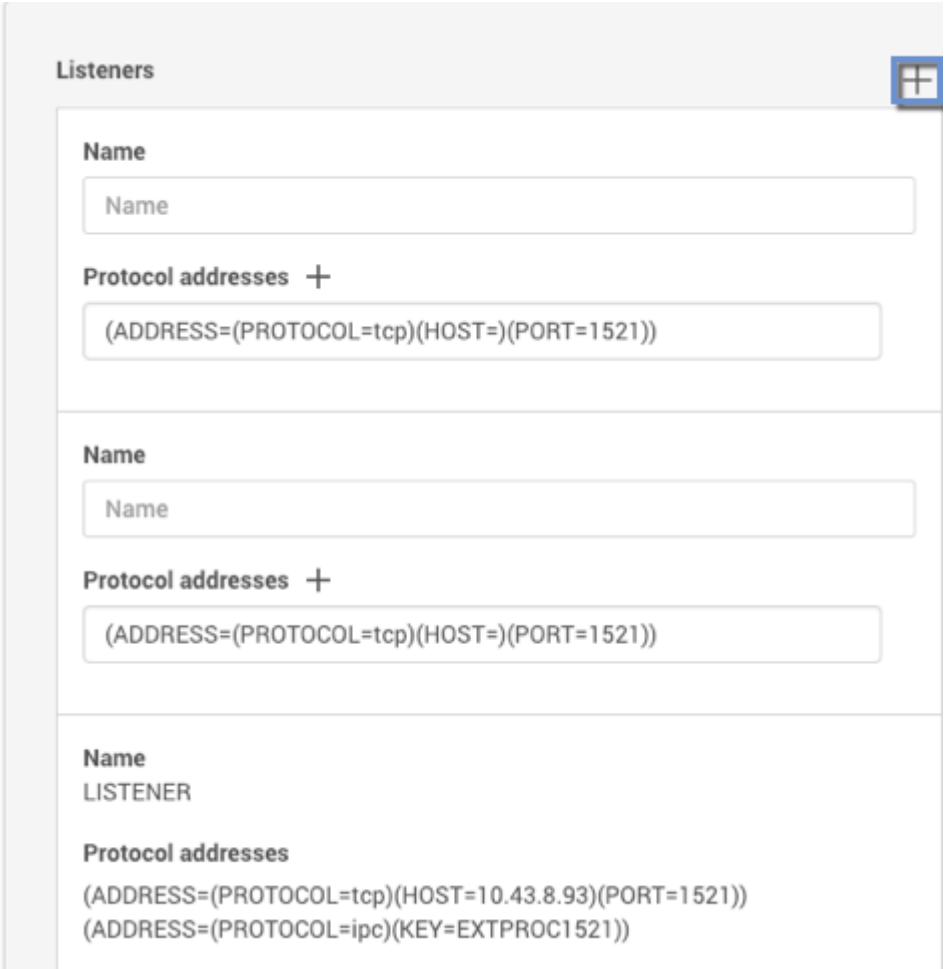
9.7.8.7.1 Procedure

1. Log into the **Delphix management** application.
2. Select **Manage > Environments**.

3. Click on the name of an environment to view its basic information.
4. In the Details tab next to Listeners, click the **Pencil** icon to edit the list of listeners.



5. In the Listeners panel click the **Plus** icon.



6. Enter a **Name** for the new listener, and a **Protocol address** of the listener. The Delphix Engine currently supports TCP and IPC protocol addresses. An example TCP protocol address is

(ADDRESS=(PROTOCOL=TCP)(HOST=10.43.17.92)(PORT=1521)) and an example IPC protocol address is (ADDRESS=(PROTOCOL=IPC)(KEY=DELPHIX))

7. Click the **Plus** icon next to **Protocol addresses** to enter additional protocol addresses.
8. Click the **Check** icon to save your changes.

9.7.8.8 Enabling linking and provisioning for Oracle databases

This topic describes how to enable and disable provisioning, and linking for databases on an environment.

Before you can use a database as a dSource, you must first make sure that you have defined an environment and that linking is allowed on it in the Delphix Environment. Similarly, before you can provision a virtual database (VDB) to a target database, you must make sure that you have allowed provisioning to the host or cluster to which the Delphix Environment is attached.

9.7.8.8.1 Procedure

1. Login to the **Delphix management** application.
2. Click **Manage**.
3. Select **Environments**.
4. Click **Databases**.
   a. **For provisioning**: On the installation, you wish to work on select Edit for the **Installation details** and toggle the **Allow provisioning** checkbox and click save.
   b. **For linking**: Under in the database section of the installation which contains the DB you wish to link click edit for **Allow linking** and toggle the checkbox then click save.

9.7.8.9 Linking an Oracle RAC source database using a standalone host environment

The standard way to link an Oracle RAC database as a dSource in the Delphix workflow is to provide the name or IP of one of the cluster nodes and then discovery will find the other cluster members, listeners, and databases. This is done by adding an 'Oracle Cluster' type Environment in the Add Environment wizard. The cluster nodes will be added to the Environment using the hostname/IP returned by the `'olsnodes'` command and in most cases, this will be sufficient. There may, however, be cases when this is undesirable, such as wanting to run snapsync/logsync traffic over a second network. Using an 'Oracle Cluster' type Environment currently stores a single IP for each cluster node and this IP will be the one associated with the node name returned by the `'olsnodes'` command. Using a Standalone Environment is one way to work around this situation by defining the Environment based on the IP on the network snapsync/logsync should run over. The Delphix Engine should have a NIC connected to this network also. When both Delphix Engine and source servers have NICs on this second network there is no need for static routes to be configured on the Delphix Engine or the source server.

1. Ensure that OS and DB users for Delphix have been configured and meet requirements (see page 1384). Create the OS user on each node using the exact same settings. Always use the hostchecker to verify

the source and target environments are ready for use with Delphix. Even though a single node will be used for linking, it may be desirable to unlink and relink the source to a different node in the future, so make sure the Delphix OS account is configured identically on all nodes.

2. Create a toolkit directory in the same location on each node and ensure permissions are set to 0770.
3. Identify the local IP address on the second network of the node which will be used for linking. Use the `'oifcfg'` and `'ifconfig'` commands to find this information. Then verify by pinging the Delphix Engine using it's IP on the second network. Don't worry about the output from `'oifcfg'` with respect to `PUBLIC,PRIVATE or UNKNOWN` values for the interfaces. These are not actual CRS usage, but rather based on the IP address range (RFC1918) - in the image they all show as `PRIVATE`.

```
[alex:syrinx_3:]/home/oracle => $CRS_HOME/bin/oifcfg iflist -p -n
eth0  10.0.0.0  PRIVATE  255.255.0.0
eth1  192.168.1.0  PRIVATE  255.255.255.0
eth1  169.254.0.0  UNKNOWN  255.255.0.0
eth2  172.16.0.0  PRIVATE  255.240.0.0
[alex:syrinx_3:]/home/oracle => ifconfig eth2
eth2      Link encap:Ethernet  HWaddr 08:00:27:6A:A9:2C
          inet addr:172.16.16.100  Bcast:172.31.255.255  Mask:255.240.0.0
          inet6 addr: fe80::a00:27ff:fe6a:a92c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:47584 errors:0 dropped:0 overruns:0 frame:0
          TX packets:132277 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5026787 (4.7 MiB)  TX bytes:168273350 (160.4 MiB)

[alex:syrinx_3:]/home/oracle => ping 172.16.16.85
PING 172.16.16.85 (172.16.16.85) 56(84) bytes of data.
64 bytes from 172.16.16.85: icmp_seq=1 ttl=255 time=0.301 ms
64 bytes from 172.16.16.85: icmp_seq=2 ttl=255 time=0.534 ms
64 bytes from 172.16.16.85: icmp_seq=3 ttl=255 time=0.251 ms
64 bytes from 172.16.16.85: icmp_seq=4 ttl=255 time=0.483 ms
64 bytes from 172.16.16.85: icmp_seq=5 ttl=255 time=0.358 ms
^C
--- 172.16.16.85 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4372ms
rtt min/avg/max/mdev = 0.251/0.385/0.534/0.108 ms
[alex:syrinx_3:]/home/oracle => _
```

4. Add a 'Standalone Host' type environment using the chosen cluster node's second network IP.
5. When discovery is finished, the Oracle homes on the cluster node should be discovered along with the listeners that run on that node. Once that is done add the database that will eventually be a dSource to by providing the db_unique_name, db_name, and the instance name that runs on the node that was added.

6. Then add a jdbc connect string using the node VIP and a service name.
7. Now click on the add dSource link. Follow the steps in Adding a dSource (see page 711) making sure to set the Snapshot controlfile location in RMAN to a location that's visible to all cluster nodes, or snapsync will fail. An ASM diskgroup is a suitable location. Even if the Snapshot controlfile is not located in a shared location, it is still possible to link the source - the first snapsync will fail with a fault that relates to the Snapshot controlfile location, but subsequent Snapsync operations should succeed.

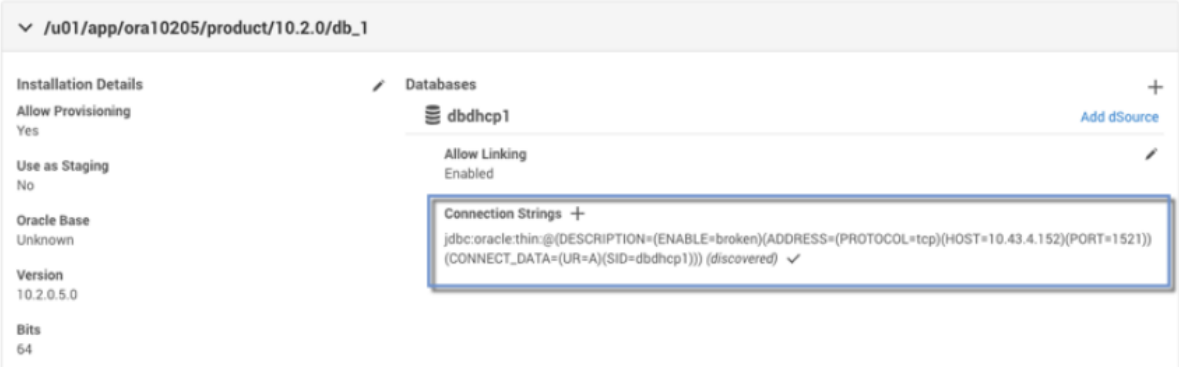
   JDBC connections will go over the public network and snapsync / logsync / environment monitor traffic will go over the second network.

9.7.8.10 Listener and JDBC verification

As with any database, a VDB needs to have a listener in order for external connections to be made. The Delphix Engine also uses a JDBC connection string in order to connect to a source or target database. If the engine does not have the proper listener or JDBC connect string defined, then connection errors can result.

9.7.8.10.1 Verifying the listener configuration

1. Verify that a listener is running on the source or target environment that you are investigating.
2. Login to the Delphix Management application.
3. Click **Manage**.
4. Select **Environments**.
5. Click on the **environment in** which you are troubleshooting.
6. On the **Configuration** tab, locate the **Listeners** section. Verify that the listener for the source or target system is listed there.
7. If a listener is not listed, it may be due to insufficient privileges on the part of the environment user that the engine is using. Verify that the proper sudo permissions have been granted for the user, or adjust them as necessary.

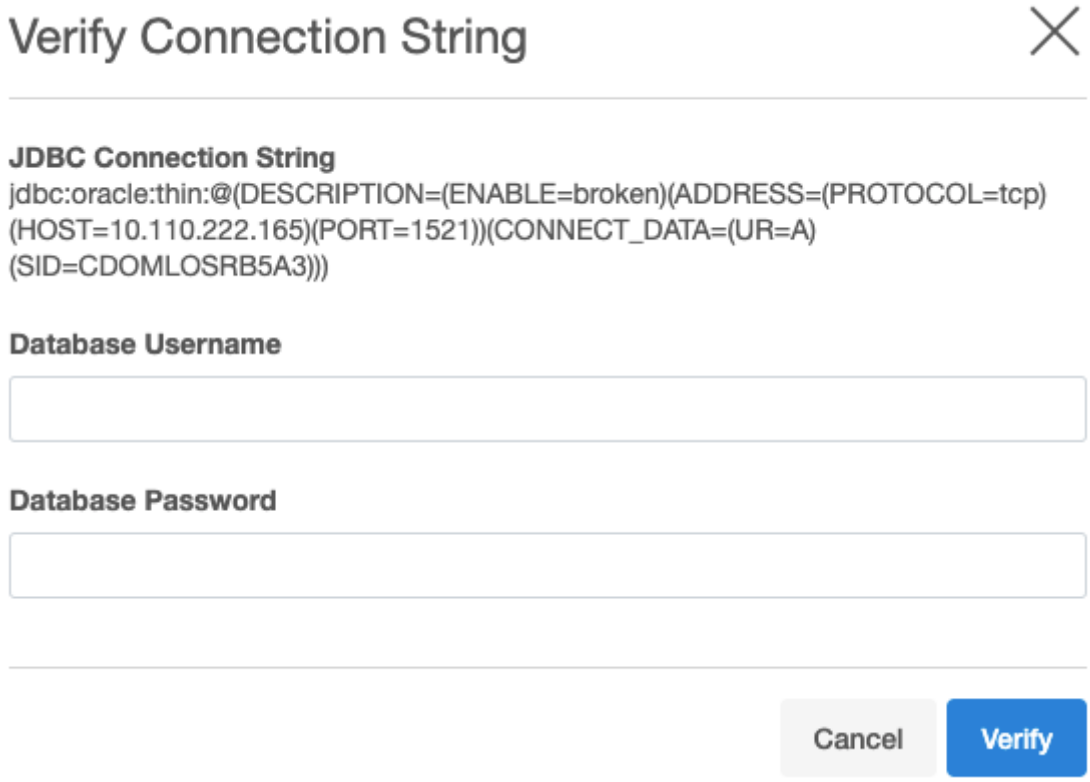


8. If you need to add another listener, you can do so manually.
   a. Click the plus (+) sign.
   b. Enter the appropriate values using the above image as a reference.

9.7.8.10.2 Verifying the JDBC connection string

Each source database and VDB has a connection string defined. If any parameters have changed, you may need to adjust these connection strings.

1. Verify that a listener is running on the source or target environment you are investigating.
2. Login to the Delphix Management application.
3. Click **Manage**.
4. Select **Environments**.
5. Click the environment in which you are troubleshooting.
6. Select the **Databases** tab.
7. The source database or VDB which you are investigating will display the JDBC connection string being used for the given database.
8. To verify that the connection string works, click the **checkmark** to the right of the connection string. You will then see the username and password text boxes.
9. Enter the Oracle username and password used by the Delphix Engine.

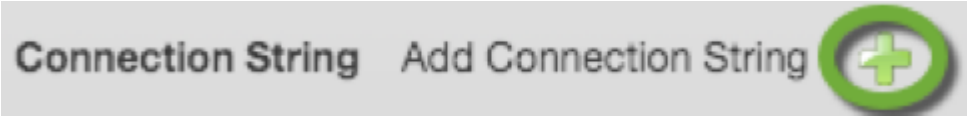


If no errors are encountered and the username/password dialogs simply go away, then a successful connection was made to the database.
If errors are encountered, you must investigate them on a case-by-case basis, just as you would do with any connection errors to an Oracle database.

9.7.8.10.3 Adding JDBC connection string

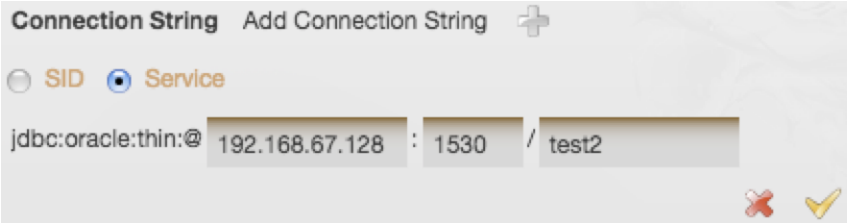
If you need to define an additional JDBC connection string for any reason, take the following steps:

1. Follow the same steps as listed in Verifying the JDBC Connection String (see page 1431) to reach the JDBC connection string definition(s).
2. Click the plus (+) sign to the right of **Connection strings** to define an additional connection string.

   Add Connection String Icon

   In the screenshot below, the user is adding a connection string for the “test2” service on port 1530 instead of the default 1521.

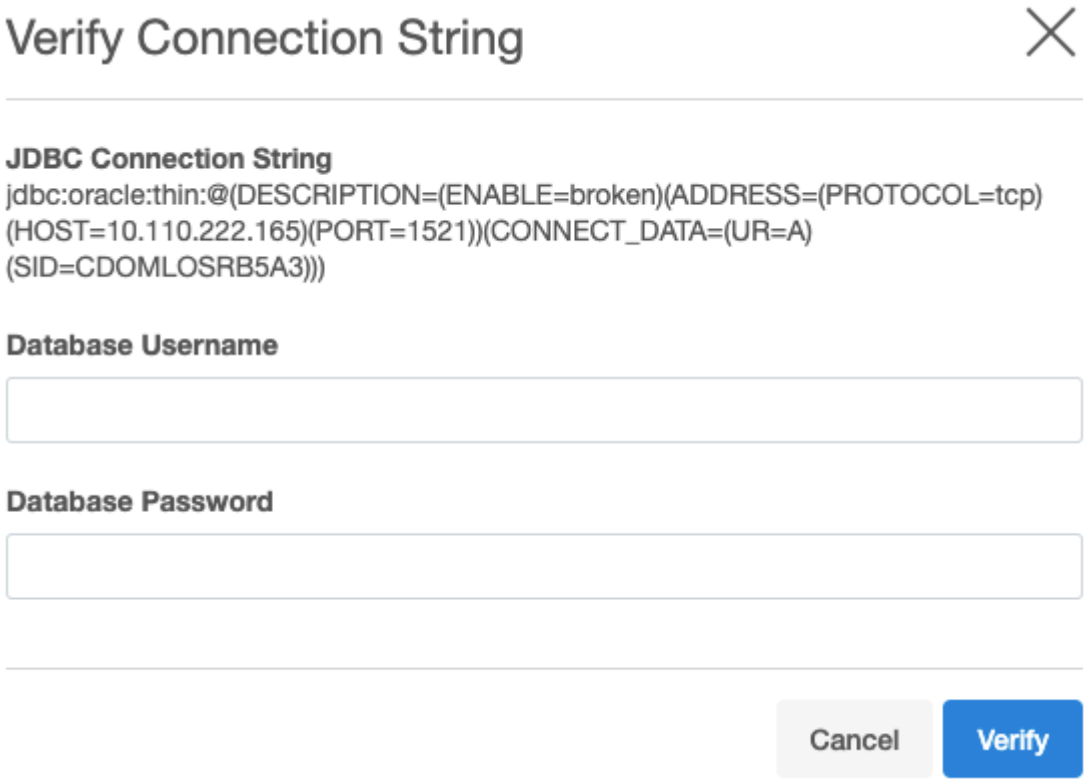
   

3. Click the **Add** to save the changes.
4. Follow the remaining steps in Verifying the JDBC Connection String (see page 1431) to validate your newly added connection string.

9.7.8.10.4 Verifying the JDBC connection string

Each source database and VDB has a connection string defined. If any parameters have changed, you may need to adjust these connection strings.

1. Verify that a listener is running on the source or target environment you are investigating.
2. Login to the Delphix Management application.
3. Click **Manage**.
4. Select **Environments**.
5. Click the environment in which you are troubleshooting.
6. Select the **Databases** tab.
7. The source database or VDB which you are investigating will display the JDBC connection string being used for the given database.
8. To verify that the connection string works, click the **checkmark** to the right of the connection string. You will then see the username and password text boxes.
9. Enter the Oracle **username** and **password** used by the Delphix Engine.



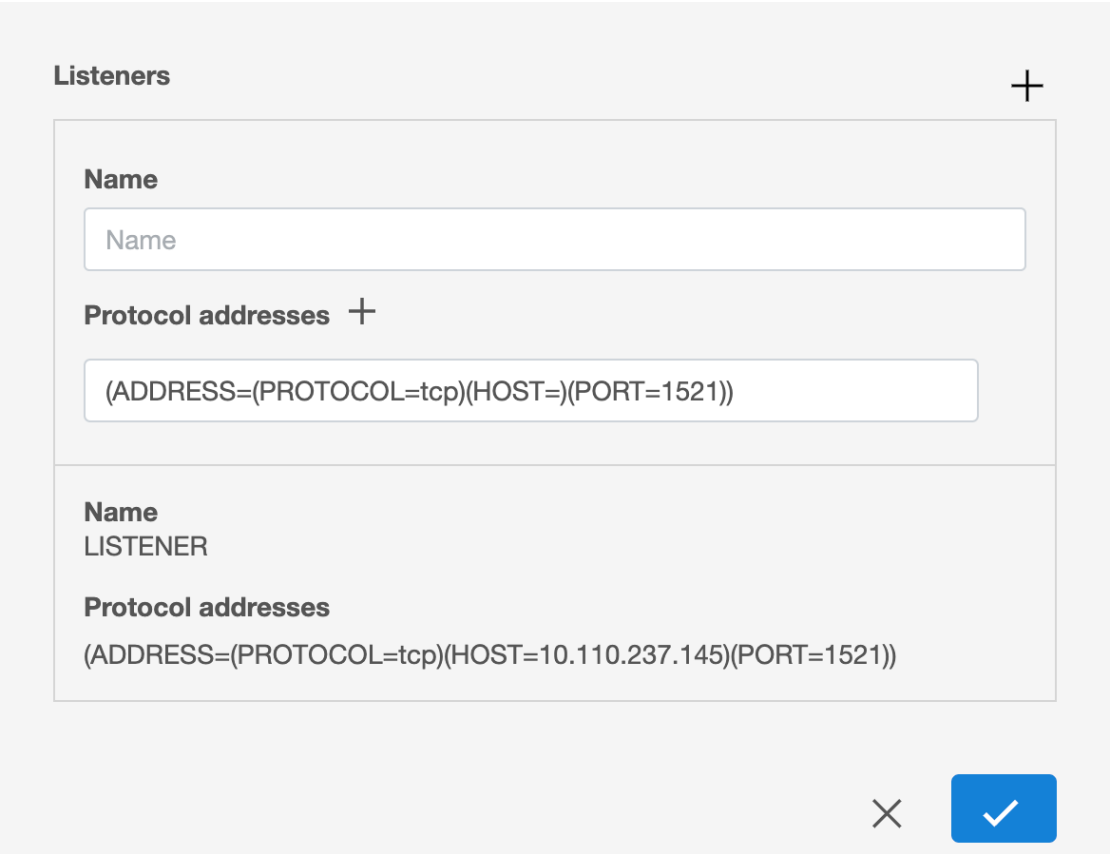
Database Username and Password

If no errors are encountered and the username/password dialogs simply go away, then a successful connection was made to the database.
If errors are encountered, you must investigate them on a case-by-case basis, just as you would do with any connection errors to an Oracle database.

9.7.8.10.5 Verifying the listener configuration

1. Verify that a listener is running on the source or target environment that you are investigating.
2. Login to the Delphix Management application.
3. Click **Manage**.
4. Select **Environments**.
5. Click on the **environment in** which you are troubleshooting.
6. On the **Configuration** tab, locate the **Listeners** section. Verify that the listener for the source or target system is listed there.
7. If a listener is not listed, it may be due to insufficient privileges on the part of the environment user that the engine is using. Verify that the proper sudo permissions have been granted for the user, or adjust them as necessary.
8. If you need to add another listener, you can do so manually.
    a. Click the **Pencil** icon.

b. Click the **Plus** icon



c. Enter the appropriate values using the above image as a reference.

9.7.8.11 Changing the hostname or IP address for Oracle source and target environments

9.7.8.11.1 Procedure

9.7.8.11.1.1 For source environments

1. Disable the dSource as described in Enabling and Disabling Oracle dSources (see page 1041)
2. If the **Host Address** field contains an IP address, edit the IP address.
3. If the **Host Address** field contains a hostname, update your Domain Name Server to associate the new IP address to the hostname. The Delphix Engine will automatically detect the change within a few minutes.
4. In the **Environments** screen of the Delphix Engine, refresh the host.
5. Enable the dSource.

9.7.8.11.1.2 For VDB target environments

1. Disable the VDB as described in Enabling and Disabling Oracle VDBs[439]
2. If the **Host address** field contains an IP address, edit the IP address.
3. If the **Host address** field contains a hostname, update your Domain Name Server to associate the new IP address to the hostname. The Delphix Engine will automatically detect the change within a few minutes.
4. In the **Environments** screen of the Delphix Engine, refresh the host.
5. Enable the VDB.

9.7.8.11.1.3 For the Delphix engine

1. To stop running your VDB select the red **Stop** button located on the VDB **Configuration** tab.
2. Disable all dSources as described in Enabling and Disabling Oracle dSources (see page 1041)
3. You can use either the command-line interface or the Delphix Setup application to change the IP address of the Delphix Engine.
   a. To use the command-line interface, press **F2** and follow the instructions described in Setting Up Network Access to the Delphix Engine (see page 569)
   b. To use the Delphix Setup application, go to **Delphix management > Engine setup** in the Delphix Management application, or click **Server setup** in the Delphix Engine login screen.
      i. In the **Network** panel, click **Modify**.
      ii. Under **DNS services**, enter the new IP address.
      iii. Click **OK**.
4. Refresh all Environments by clicking the **Refresh** symbol on the **Environments** screen.
5. Enable all dSources as described in Enabling and Disabling Oracle dSources (see page 1041)
6. To start all VDBs, click the **Start** button located on the VDB **Configuration** tab.

Using custom init.ora or spfile.ora files

If you are are using custom `init.ora` or `sfpfile.ora` files with your Oracle VDBs, you should use the Oracle command-line interface (`sqlplus/srvctl`) to shut down any active VDBs and copy the parameter files to a backup location. Complete the steps above, then replace the files and re-start the VDB from the Oracle command line to restore your custom settings. See Customizing VDB File Mappings (see page 1533) for more information about customizing init.or and other configuration files.

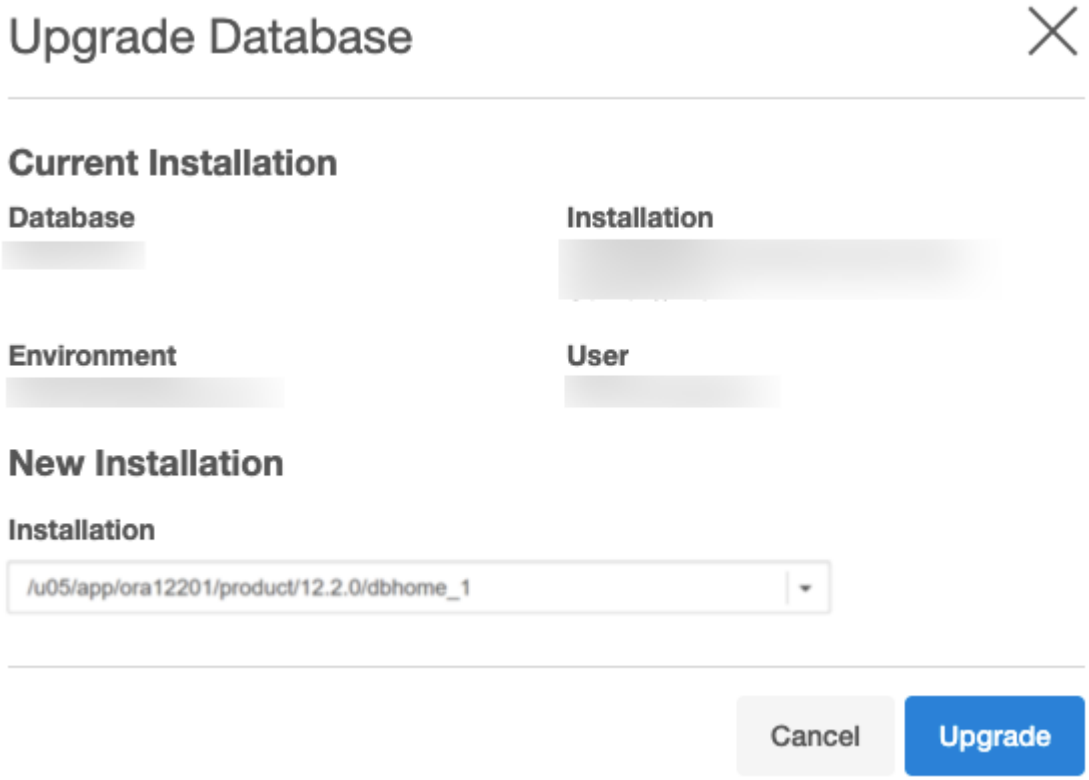
439 https://delphixdocs.atlassian.net/wiki/spaces/DCT/pages/1114600

9.7.8.12 How to change ORACLE_HOME

9.7.8.12.1 Prerequisites

- know the new ORACLE_HOME user and password
- know the VDB you want to change
- have delphix_admin privileges to the VDB

9.7.8.12.2 Procedure

1. Login to the **Delphix management** application.
2. Click **Manage**.
3. Select **Datasets**.
4. Select your VDB.
5. Select the **Configuration tab.**
6. In the upper right-hand corner of the **Source** sub-tab, click the **Actions menu (...)** and select **Upgrade**.
7. Select the new **Installation** from the drop-down menu.
8. 

Click **Upgrade**.

9.7.8.13 CLI support for partial-full backup in Oracle

9.7.8.13.1 Overview

Delphix Engine version 6.0.8.0 introduces an option to perform partial full snapsync via the CLI. A full backup of specified datafiles will be performed, along with an incremental backup of the remaining files.

9.7.8.13.2 Prerequisites

1. Requested datafiles must exist in the database for which SnapSync is being run.
2. In case of an MT, Snapsync must be performance for the PDB, the datafiles can belong to either the PDB or its CDB.
3. Requested datafiles must be online.
4. Partial full SnapSync can not be requested for an initial SnapSync.
5. Partial full SnapSync can not be run if there is a pending SnapSync.

9.7.8.13.3 Command

```
ip-00-000-000-000 database> select CDOMLOSR1CEBPDB3
ip-00-000-000-000 database 'CDOMLOSR1CEBPDB3'> sync
ip-00-000-000-000 database 'CDOMLOSR1CEBPDB3' sync *> set filesForFullBackup=1,2
ip-00-000-000-000 database 'CDOMLOSR1CEBPDB3' sync *> commit
```

9.7.8.14 Managing cluster nodes of an Oracle RAC environment

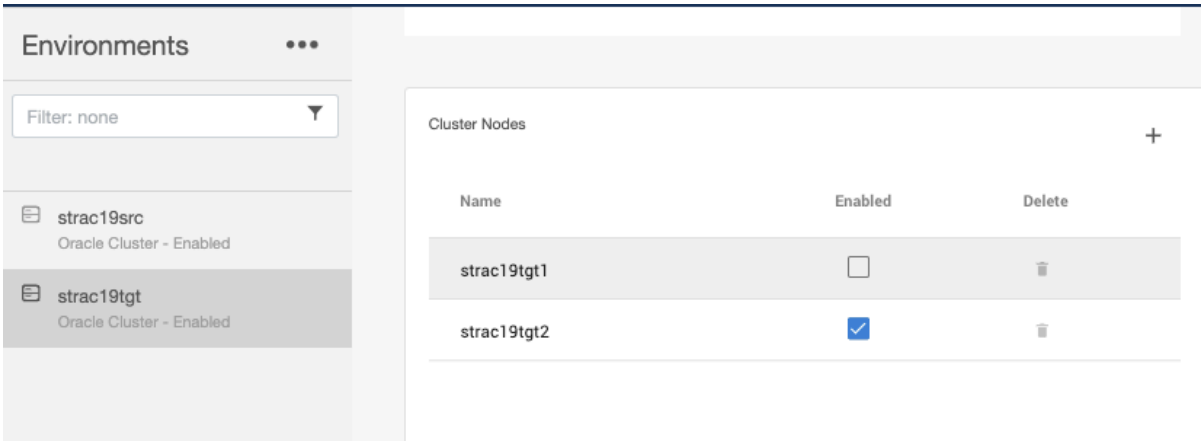
This topic describes how to manage cluster nodes of an Oracle RAC environment from the **Delphix Management** application.

9.7.8.14.1 Enabling a cluster node

9.7.8.14.1.1 Procedure

1. Login to the **Delphix Management** application.
2. Click **Manage**.

3. Select **Environments**.
4. Select your environment.
5. Click the **Details** tab.
6. In the **Cluster Nodes** section, for the desired node, check the check-box in the **Enabled** column.



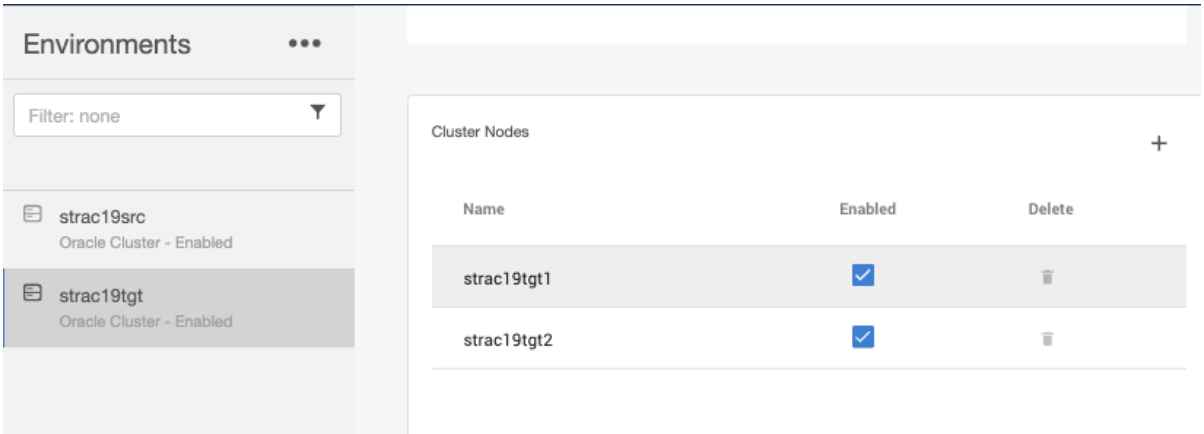
9.7.8.14.2 Disabling a cluster node

9.7.8.14.2.1 Prerequisite

All the virtual database instances on the cluster node must be stopped from the **Delphix Management** application. Refer to the steps documented here (see page 0) for stopping the cluster instances.

9.7.8.14.2.2 Procedure

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Environments**.
4. Select your environment.
5. Click the **Details** tab.
6. In the **Cluster Nodes** section, for the desired node, uncheck the check-box in the **Enabled** column.



9.7.9 Linking data sources and syncing data with Oracle

Linking a data source will create a dSource object on the engine and allow Delphix to ingest data from this source. The dSource is an object that the Continuous Data Engine uses to create and update virtual copies of your database. As a virtualized representation of your source data, it cannot be managed, manipulated, or examined by database tools.

Once a dSource has been linked, you can begin to perform dSource operations on it such as enable, detach, delete, and more.

This section covers the following topics:

- Linking data sources with Oracle (see page 1438)
- Linking an Oracle pluggable database (see page 1439)
- Staging push implementation for Oracle (see page 1440)
- Data management settings for Oracle data sources (see page 1452)
- Prepare and upgrade a non-MT Oracle dSource to MT (see page 1456)
- Repository templates for Oracle databases (see page 1459)
- Using the purgeLogs operation (see page 1460)
- Advanced linking (see page 1463)

9.7.9.1 Linking data sources with Oracle

Linking a dSource will ingest data from the source and create a dSource object on the engine. The dSource is an object that the Continuous Data Engine uses to create and update virtual copies of your database. As a virtualized representation of your source data, it cannot be managed, manipulated, or examined by database tools.

Once a dSource has been linked, you can begin to perform dSource operations on it such as enable, detach, delete, and more.

For an overview of all dSource related actions, please visit Provisioning and Managing Virtual Databases (see page 1438)

1. Login to the **Delphix management** application.
2. Navigate to **Manage > Datasets.**

3. Click the plus icon and select **Add dSource**.
4. In the **Add dSource** wizard, select the source database with the correct environment user-specified.
5. Enter your login credentials for the source database and click **Verify credentials**. If you are linking a mounted standby, Click **Next**. See the topics under Linking Oracle Physical Standby Databases (see page 1466) for more information about how the Delphix Engine uses non-SYS login credentials.
6. Enter a name and select a group for your dSource. Adding a dSource to a dataset group lets you set Delphix Domain user permissions for that database and its objects, such as snapshots. See the topics under Users and Groups (see page 674) for more information.
7. Select the **Data management** settings needed. For more information, visit Data Management Settings for Oracle Data Sources (see page 1452)
8. Assign existing policies to the new dSource. New policies can be created and associated later.
9. Enter any scripts that should be run using the **Hooks** page.
10. Review the dSource Configuration and Data Management information, and then click **Submit**.

Once the action to add a dSource has been submitted, the Delphix Engine will initiate a DB_Link job to create the dSource. If the *Load Immediately* option was selected in the data management page a DB_Sync job will also be executed to ingest data from the source, otherwise, this first DB_Sync job will run as per the associated SnapSync policy.

When the jobs have successfully completed, the database icon will change to a dSource icon on the Environments > Databases screen, and the dSource will be added to the list of Datasets under its assigned group.

9.7.9.2 Linking an Oracle pluggable database

This topic describes how to link an Oracle 12c pluggable database to the Delphix Engine to create a dSource.

9.7.9.2.1 Prerequisites

- Make sure the Delphix Engine has already discovered the multitenant container database and its pluggable databases. If the container database does not exist in the environment, follow the steps in Adding a Database to an Oracle Environment (see page 1424). If the pluggable database you want to link does not exist in the environment, refer to step 4 in the procedure below.
- You should have Block Change Tracking (BCT) enabled for the container database, as described in Requirements for Oracle Hosts and Databases (see page 1392).
- The container database should be in ARCHIVELOG mode and the NOLOGGING option should be disabled, as described in Requirements for Oracle Hosts and Databases (see page 1392).

9.7.9.2.2 Procedure

1. Log into the **Delphix management** application.
2. Select **Manage > Datasets.**
3. Click **the Plus** icon and select **Add dSource.**

4. In the **Add dSource** wizard, select the source pluggable database. If the container database is shown but the pluggable database is not, select the **container database**, enter its **database credentials**, and click **Verify credentials**. The Delphix Engine will discover and list all pluggable databases in the container database. Select the **pluggable database** from the list. Alternatively, you can use the **Environment management** screen to discover pluggable databases in any of the discovered container databases.
5. Enter your **login credentials** for the source database and click **Verify credentials**.
6. Click **Next**.
7. Select a **Database group** for the dSource.
8. Click **Next**.
9. Select an **Initial load** option. By default, the initial load takes place upon completion of the linking process. Alternatively, you can set the initial load to take place according to the SnapSync policy. For example, you can set the initial load to take place when the source database is not in use, or after a set of operations have taken place.
10. Select a **SnapSync** policy. See Data Management Settings for Oracle Data Sources (see page 1452) for more information.
11. Click **Next**.
12. Review the **dSource configuration** and **Data management** information.
13. Click **Submit**. The Delphix Engine will initiate two jobs, **DB_Link** and **DB_Sync**, to create the dSource. You can monitor these jobs by clicking **Active Jobs** in the top menu bar, or by selecting **System > Event Viewer**. When the jobs have completed successfully, the database icon will change to a dSource icon on the **Environments > Datasets** screen, and the dSource will be added to the list of **Datasets** under its assigned group. Link/Sync of the Multitenant Container Database The **DB_Link** job will also link the pluggable database's multitenant container database if it has not been linked yet.

9.7.9.3 Staging push implementation for Oracle

This section covers the following topics:

- Oracle staging push-overview (see page 1441)
- Linking an Oracle staging push dSource (see page 1441)
- Populating staging database with production data (see page 1447)
- Taking Oracle staging push dSource snapshots (see page 1448)
- Staging Push LogSync (see page 1449)
- Migrating to another repository (see page 1451)

9.7.9.3.1 Oracle staging push-overview

9.7.9.3.1.1 Introduction

The staging push feature allows you to ingest production data into the Delphix Engine without the need for the Delphix-initiated backups. It uses a staging database instance that can be populated with production data, by restoring your existing backups or by setting it up as a physical standby, using storage provided by the Delphix Engine. Delphix Engine then uses the database files on this storage to take dSource snapshots. Thus, eliminating the need to access the production environment. Since the restore process is not controlled by Delphix Engine, any backup vendor can be used to restore production database backups on the staging database.

9.7.9.3.1.2 Workflow

Ingesting data from Staging Push dSources consists of the following steps:

- Linking an Oracle Staging Push dSource (see page 1441)
- Populating staging database with production data (see page 1447)
- Taking a snapshot (see page 1448)

The staging database can be populated further with incremental data from the production database to take incremental snapshots.

- Staging Database must be in the MOUNT, READ ONLY or READ ONLY WITH APPLY mode for taking a snapshot.
- Disable or enable operations are allowed only at the CDB level. Disabling a CDB dSource will also disable all the PDB dSources.
- Detach or attach operations are not allowed on the staging push dSources.
- Delete operation is allowed for both CDB and PDB dSources. However, since PDBs cannot be unplugged/dropped from an Oracle database in the MOUNT state, you will be responsible for unplugging/dropping the PDB from the CDB. Deleting a CDB dSource will also delete all the PDB dSources.
- A Non Multi Tenant Staging Push dSource can not be used to ingest a multi-tenant database. For a multi-tenant database, separate Staging Push dSources must be created for CDB and individual PDBs.

9.7.9.3.2 Linking an Oracle staging push dSource

This section covers the following topics:

- Linking an Oracle non multi tenant staging push dSource (see page 1442)
- Linking an Oracle multi tenant staging push dSource (see page 1444)

- While linking, the archive, datafile, external, script and temp directories are mounted from the Delphix Engine to the mount base directory provided in the linking parameters. The location of these directories is as follows:
  - For Non Multi Tenant Databases:
    - *MOUNT_BASE/<Database_Unique_Name>/*
  - For Multi Tenant Databases:
    - For CDB and PDB$SEED: *MOUNT_BASE/<CDB_Unique_Name>/*
    - For PDB: *MOUNT_BASE/<PDB_Name>-<CDB_Unique_Name>/*
- Apart from the `script` directory which contains scripts that are used by Delphix Engine for various database operations, the other directories can be used for the following purposes:
  - `datafile` : All the data files along with the control file should be placed in this directory.
  - `external` : Any files external to the database but needed by the database can be placed in this directory. The content of this directory will be available with the VDB provisioned from the dSource.
  - `archive` : Database archive log files can be placed in this directory. These files are used for point-in-time provisioning of VDBs from the dSource.
  - `temp` : This directory can be used for temporary storage requirements on the staging host. The directory should be cleanup up after its use.
- No snapshot is taken on linking an Oracle Staging Push dSource. To take a snapshot, a sync operation should be initiated after populating the staging database with production data.

9.7.9.3.2.1 Linking an Oracle non multi tenant staging push dSource

You can perform the following steps to link a Non Multi Tenant dSource using the staging push mechanism.

1. Login to the **Delphix management** application.
2. Go to **Manage > Datasets**.
3. Click the plus (+) icon and select **Add dSource**.
4. On the **Preparation** tab, click **Next**.
5. From the **dSource type** tab, select the **Oracle staging push** as dSource type and click **Next**.
6. On the **dSource configuration** step, provide the following dSource configurations:
   a. **dSource name** - This name will be shown on the Delphix Engine interface.
   b. **Target group** - Select a target group from the drop-down list.

c. **Database type** - Select the database type as **Non Multi Tenant**.
d. **Staging environment** - Select an environment from the drop-down list. The staging database will be hosted in this environment.
e. **User** - Select an environment user from the drop-down list.
f. **Repository** - Select a repository (Oracle installation) from the drop-down list.
g. **Database name:** Provide the Database Name for the staging dSource.
Info : Consider the following while providing the database name:
    - This is the name of the staging database that is created on the staging host after linking.
    - The Database Name should be the same as the source database's database name. The source database is the database whose data will be populated on this staging database.
    - Database Name is case-sensitive.
    - You can run either of the two queries on the source database to get the database name:
        - `show parameter db_name;`
        - `select name from v$database;`
h. **Database unique name** - Provide a database unique name for the staging database. It can be different from the database unique name of the source database.
i. **SID** - Provide the SID for the staging database.
j. **Mount base** - Provide a mount path. This is the location on the staging host at which the Delphix Engine will mount its storage for the staging database. To know about the permissions that the mount path directory must-have, see the Additional Target/Staging Host Requirements section of this (see page 1392) article.
k. **Custom environment variables** - Configure the custom environment variables (see page 1530), similar to VDBs, if needed.
l. Optionally, you can configure **Auto staging push restart**, **Configure staging database parameters**, **Physical standby**, and **Validate snapshot by Opening database in read only mode**, as per requirement.
    i. **Auto staging push restart** indicates whether this staging database should be automatically restarted when the staging host reboot is detected.
        1. For non-standby database, only storage is mounted on the staging host.
        2. For a standby database, after mounting the storage the database is also started in MOUNT mode.
    ii. **Configure staging database parameters** to override Oracle database configuration parameters.
    iii. **Physical standby** indicates that the staging database will be configured as a physical standby.
    Info: The staging database is not configured as a Physical Standby by Delphix Engine. The setup should be done outside Delphix Engine before initiating the Snapshot operation.

iv. **Validate Snapshot by Opening Database in Read Only Mode** indicates whether the staging database snapshots will be validated by opening the database in read-only mode.
Info: All these options are disabled by default.

m. Click **Next**.

7. From the **Staging database parameters** tab, select a template and click **Next**.
8. From the **Data Management** tab, select **LogSync** checkbox to **enable** the logsync for Staging Push dSource. For more information on **LogSync**, see Staging Push LogSync (see page 1449), click **Next.**
9. From the **Policies** tab, select any policy for the new dSource. **SnapSync Policy** is used as a default policy for taking snapshots. For more information on SnapSync policy, see Policies for Scheduled Jobs (see page 1068) Click **Next**.
10. From the **Hooks** tab, enter any script that should be run before the snapshot operation (pre-sync), after snapshot but before post-sync hook (pre-log-sync) or after the snapshot operation (post-sync).
11. From the **Summary** tab, review the staging Push dSource configuration profile and click **Submit**. When all the above steps are executed on the staging host, a new database is instantiated with the above provided Database Name, Database Unique Name, and SID, and the database files are expected to be placed at the location: `MOUNT_BASE/<Database_Unique_Name>/datafile`. Once linking is done, you can see this Staging Push dSource under the **Datasets**.

9.7.9.3.2.2 Linking an Oracle multi tenant staging push dSource

Oracle Multi Tenant Staging Push dSource can be linked with the following steps:

1. Link a CDB staging push dSource (see page 1444)
2. Link a PDB staging push dSource (see page 1446)

- A CDB Staging Push dSource must be created before creating a PDB Staging Push dSource.
- A CDB dSource ingests data for the CDB and PDB$SEED databases.
- Except for PDB$SEED, a PDB dSource must be created for each PDB associated with the CDB.

Linking a CDB staging push dSource

You can perform the following steps to link a CDB dSource using the staging push mechanism.

1. Login to the **Delphix management** application.
2. Go to **Manage > Datasets**.
3. Click the plus (+) icon and select **Add dSource**.
4. On the **Preparation** tab, click **Next**.

5. From the **dSource Type** tab, select the **Oracle staging push** as dSource type and click **Next**.
6. On the **dSource configuration** step, provide the following dSource configurations:
   a. **dSource name** - This name will be shown on the Delphix Engine interface.
   b. **Target group** - Select a target group from the drop-down list.
   c. **Database type** - Select the database type as CDB.
   d. **Staging environment** - Select an environment from the drop-down list. The staging database will be hosted in this environment.
   e. **User** - Select an environment user from the drop-down list.
   f. **Repository** - Select a repository (Oracle installation) from the drop-down list.
   g. **CDB name:** Provide the Database Name for the staging CDB dSource.
   h. **CDB unique name:** Provide CDB unique name for the staging database. It can be different from the database unique name of the source CDB database.
      Info: Consider the following while providing the CDB name:
      - This is the name of the staging container database that is created on the staging host after linking.
      - It should be the same as the source CDB's Database Name
      - It is case-sensitive
      - You can run either of the two queries on the source database to get the CDB database name:
        - `show parameter db_name;`
        - `select name from v$database;`
   i. **SID** - Provide the SID for the staging database.
   j. **Mount base** - Provide a mount path. This is the location on the staging host at which the Delphix Engine will mount its storage for the staging database. To know about the permissions that the mount path directory must have, see Additional Target/Staging Host Requirements section of this (see page 1392) article.
   k. **Custom environment variables** - Configure the custom environment variables (see page 1530), similar to VDBs, if needed.
   l. Optionally, you can configure **Auto staging push restart**, **Configure staging database parameters**, **Physical standby**, and **Validate snapshot by opening Database in read only mode**, as per requirement.
      i. **Auto Staging Push Restart** indicates whether this staging database should be automatically restarted when the staging host reboot is detected.
         1. For a non-standby database, only storage is mounted on the staging host.
         2. For a standby database, after mounting the storage the database is also started in MOUNT mode.
      ii. **Configure Staging Database Parameters** to override Oracle database configuration parameters.
      iii. **Physical Standby** indicates that the staging database will be configured as a physical standby.
      Info: The staging database is not configured as a Physical Standby by Delphix Engine.

The setup should be done outside Delphix Engine before initiating the snapshot operation.

iv. **Validate Snapshot by Opening Database in Read Only Mode** indicates whether the staging database snapshot will be validated by opening the database in read-only mode.
Info: All these options are disabled by default

m. Click **Next**.

7. From the **Staging Database Parameters** tab, select a template and click **Next**.
8. From the **Data Management** tab, select the **LogSync** checkbox to **enable** the logsync for Staging Push dSource. For more information on **LogSync**, see Staging Push LogSync (see page 1449), click **Next.**
9. From the **Policies** tab, select any policy for the new dSource. **SnapSync Policy** is used as a default policy for taking snapshots. For more information on SnapSync policy, see Policies for Scheduled Jobs (see page 1068) Click **Next**.
10. From the **Hooks** tab, enter any script that should be run before the snapshot operation (pre-sync), after snapshot but before post-sync hook (pre-log-sync) or after the snapshot operation (post-sync).
11. From the **Summary** tab, review the Staging Push dSource configuration profile and click **Submit**.

When all the above steps are executed on the staging host, a new database is instantiated with the above provided Database Name, Database Unique Name, and SID, and the database files are expected to be placed at the location: `MOUNT_BASE/<CDB_Unique_Name>/datafile`. Once linking is done, you can see this Staging Push CDB dSource under the **Datasets**.

Linking a PDB staging push dSource

You can perform the following steps to link a PDB dSource using the staging push mechanism.

1. Login to the **Delphix Management** application.
2. Go to **Manage > Datasets**.
3. Click the plus (+) icon and select **Add dSource**.
4. On the **Preparation** tab, click **Next**.
5. From the **dSource Type** tab, select the **Oracle Staging Push** as dSource type and click **Next**.
6. On the **dSource Configuration** step, provide the following dSource configurations:
   a. **dSource Name** - This name will be shown on the Delphix Engine interface.
   b. **Target Group** - Select a target group from the drop-down list.
   c. **Database Type** - Select the database type as PDB.
   d. **Staging Environment** - Select an environment from the drop-down list. The staging database will be hosted in this environment.
   e. **Repository** - Select a repository (Oracle installation) from the drop-down list.
   f. **Container Database**: Select the Staging Push Container database to which the PDB will be plugged.
   g. **PDB Name:** Provide the PDB name for the staging PDB dSource.
   h. **Custom Environment Variables** - Configure the custom environment variables (see page 1530), similar to VDBs, if needed.

i. Optionally, you can configure **Validate Snapshot by Opening Database in Read Only Mode**, as per requirement. This indicates whether the staging database snapshot will be validated by opening the database in read-only mode.
j. Click **Next**.

7. From the **Staging Database Parameters** tab, select a template and click **Next**.
8. From the **Policies** tab, select any policy for the new dSource. **SnapSync Policy** is used as a default policy for taking snapshots. For more information on SnapSync policy, see Policies for Scheduled Jobs (see page 1068) Click **Next**.
9. From the **Hooks** tab, enter any script that should be run before the snapshot operation (pre-sync), after snapshot but before post-sync hook (pre-log-sync) or after the snapshot operation (post-sync).
10. From the **Summary** tab, review the Staging Push dSource configuration profile and click **Submit**.

Upon submission, a new pluggable database will be created and plugged into the given Container Database on the staging host. The data files for this pluggable database are expected to be placed at the location: `MOUNT_BASE/<PDB_Name>-<CDB_Unique_Name>/datafile`. Once linking is done, you can see this Staging Push PDB dSource under the **Datasets**.

9.7.9.3.3 Populating staging database with production data

You can update the staging database with the latest data from the source database by either restoring source database backups over it or, by setting it up as a physical standby.

9.7.9.3.3.1 Pre-requisites

> While populating the data on the staging database, make sure that all the data files go to the below Delphix Engine mounted location:
> For Non Multi Tenant Databases:
> - *MOUNT_BASE/<Database_Unique_Name>/datafile*
>
> For Multi Tenant Databases:
> - For CDB and PDB$SEED: *MOUNT_BASE/<CDB_Unique_Name>/datafile*
> - For PDB: *MOUNT_BASE/<PDB_Name>-<CDB_Unique_Name>/datafile*

9.7.9.3.3.2 Restoring and recovering the source database

- For Non Multi Tenant database, after linking the dSource, restore your existing source database backups onto the mount location.
- For Multi Tenant databases, after linking the CDB and PDB dSources:
  - Restore CDB and PDB$SEED backups to the CDB mount location.
  - Restore PDB backup to the PDB mount location.

9.7.9.3.3.3 Partial Ingestion of production data

The production database backup can be partially restored on the staging database by skipping the undesired tablespaces during restore and recovery process.

9.7.9.3.3.4 Automating the data population

You can automate the data population process on the staging database in the following ways:

- Create automation scripts and schedule/execute them outside of Delphix Engine.
- Configure pre/post-sync hooks to restore a backup or start/stop managed recovery process.

9.7.9.3.4 Taking Oracle staging push dSource snapshots

9.7.9.3.4.1 Taking a Snapshot

On initiating a sync (see page 1511), Delphix Engine checks the staging server for the database of the given name and if a match is found, it then takes a snapshot of its data files. Before and after taking the snapshot, Delphix Engine runs a few validations to check the readiness and consistency of the database for the snapshot.

Below are the checks performed by Delphix Engine during snapshot operation:

- The staging database is in the `MOUNT`, `READ ONLY` or `READ ONLY WITH APPLY` mode
- All data files are present on the Delphix Engine mounted storage
- For a non-standby staging database, checkpoint SCN of all online datafiles is the same
- The database must be recovered and no data files must be in `FUZZY` state
- Checkpoint SCN must have advanced since the last snapshot was taken
- Checkpoint SCN has not advanced during the snapshot operation

> CDB dSource snapshots cannot be taken manually. A PDB dSource snapshot operation takes a CDB dSource snapshot first before taking a PDB snapshot.

9.7.9.3.4.2 Taking an incremental snapshot

The staging database can be populated further by:

- Restoring incremental backups
- Through ongoing managed recovery in case of Physical Standby databases

After the staging database is populated with incremental changes, a snapshot operation will take an incremental snapshot of the dSource.

9.7.9.3.5 Staging Push LogSync

9.7.9.3.5.1 Introduction

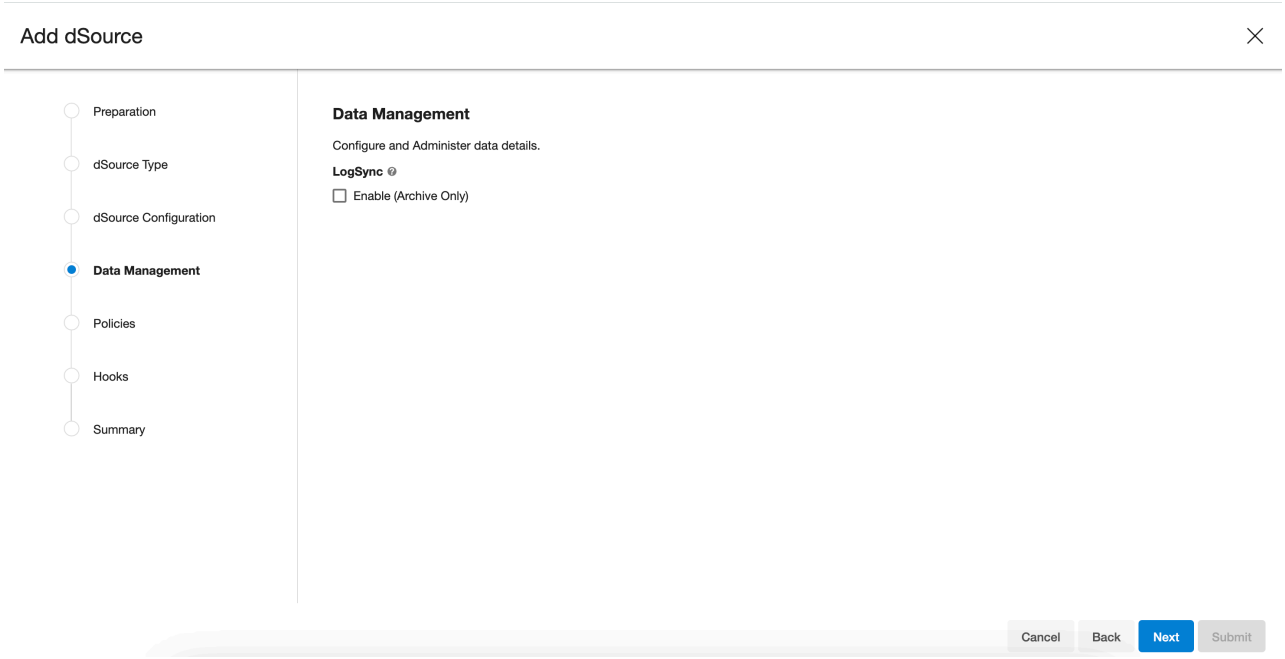
Staging Push LogSync processes the archive logs available at the mounted **archive** file system on the staging host. Once the archived logs are processed, log ranges will appear on the snapshots which can be used for the point-in-time or SCN based provisioning from the Staging Push dSource snapshots.
The archived logs should be placed in the mounted **archive** file system by configuring the archive destination (e.g. “log_archive_dest_1”) to the mounted dSource **archive** directory on the staging host.

- For a non-standby database - Restore the archived logs on the mounted **archive** directory.
- For a standby database - Configure the archive destination to the mounted ‘archive’ directory for the archived logs to be generated in this directory.

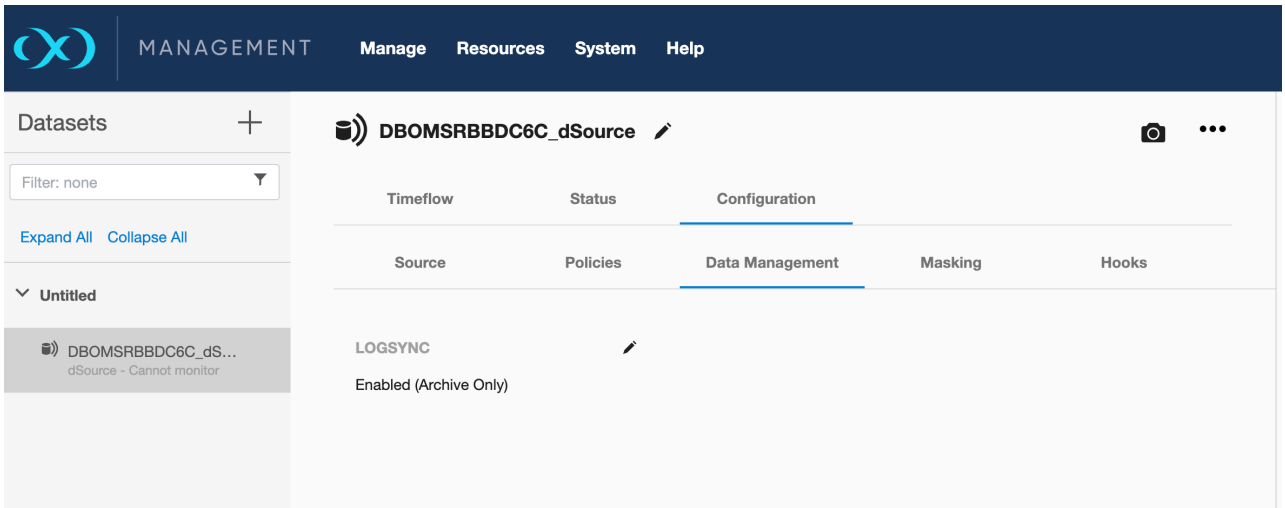
> The archive filesystem is mounted to the following directories on the staging host:
> For Non Multi Tenant Databases:
> - *MOUNT_BASE/<Database_Unique_Name>/archive*
>
> For Multi Tenant Databases:
> - *MOUNT_BASE/<CDB_Unique_Name>/archive*

9.7.9.3.5.2 Enable/Disable LogSync

You can enable LogSync from the **Data Management** screen by selecting the **LogSync** checkbox during the Staging Push dSource creation.



Once dSource is created, LogSync can be enabled/disabled from the **Data Management** screen under the **Configuration** tab by clicking on the pencil icon next to **LogSync** and then selecting/unselecting the **LogSync** checkbox.



9.7.9.3.5.3 Workflow

Staging Push LogSync operation runs every 5 minutes and consists of the following steps:

- Process the archived logs only if at least one staging push dSource snapshot exists.
- Read and process all the new archived log files available in the mounted **archive** file system after the last LogSync.
- Delete invalid archive log files. Online redo logs and SRLs are not processed and will be deleted.

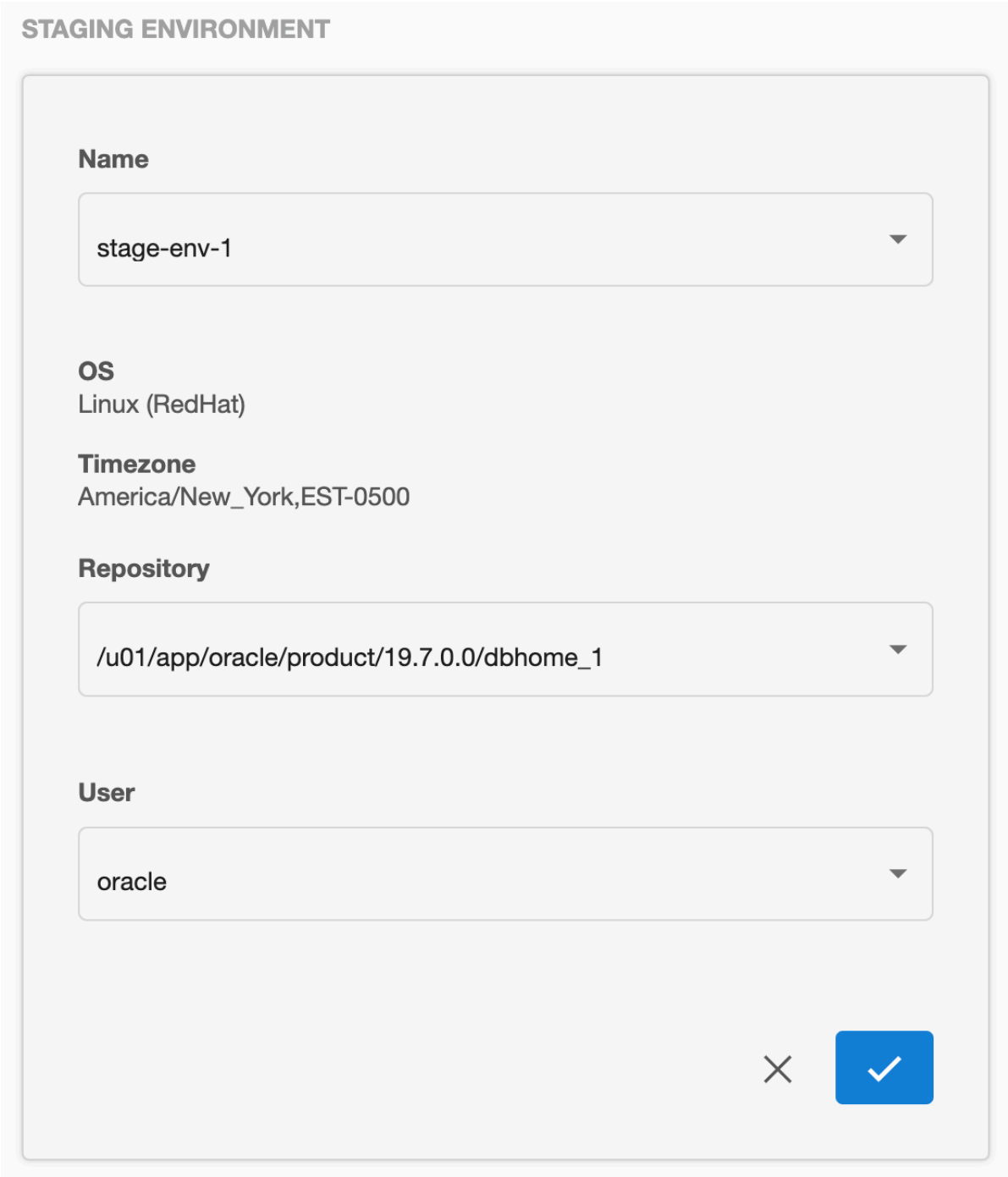
9.7.9.3.5.4 PreLogSync Hook

PreLogSync hook can be used to switch the logs on the primary database in case of a standby staging push setup. It runs after the snapshot is successfully taken but before the post-sync hook.

9.7.9.3.6 Migrating to another repository

The staging push database can be moved to another repository in the same or a different environment using the following steps:

1. Disable the staging push dSource.
2. Under **Datasets**, go to dSource **Configuration** tab.
3. Click on the pencil icon next to **Staging Environment** to get into edit mode.
4. Update the Environment **Name** (if migrating to another environment), **Repository** and **User**.



5. Enable the dSource.

9.7.9.4 Data management settings for Oracle data sources

Each dSource has its own data management settings, which can be configured during the linking workflow as well as in the configuration page for that dSource.

You can configure data management settings to improve overall performance and match the needs of your specific server and data environment.

The following settings are available for Oracle data sources:

Setting	Explanation
Initial Load	By default, the initial load takes place upon completion of the dSource linking wizard. Alternatively, you can set the initial load to take place according to a SnapSync policy. This might be desirable if you want the initial load to take place when the source database is under a lower load.
External Data Directory	An optional directory on the database server containing extra files associated with the database that are not included in normal SnapSync snapshots, for example, external table data files. Nothing in this directory will be modified, but the files will be loaded and provisioned along with the database.
Compression	Enable compression of backup data sent over the network. Compression is enabled by default.
Bandwidth Limit	Select the network bandwidth limit in units of MB/s between the source and the Delphix Engine. The default is 0, indicating that no bandwidth limit is enforced.
Number of Connections	Select the number of TCP connections to use between the Source and the Delphix engine. Multiple connections may improve network throughput especially over long distances and highly congested networks. The default is 1.
Encrypted Linking	Turn on encryption between the source and the Delphix Engine for SnapSync and LogSync communication. Enabling encrypted linking leads to higher CPU utilization and higher throughput. Encrypted Linking is disabled by default.

Setting	Explanation
Data Load Channels	The channel settings determine the number of channels and data files per backup set. While these settings can be increased, you should consider potential adverse effects on the performance of database operations on the Source server • Number of Channels - set the number of RMAN channels used during SnapSync. The default is 2. • Files per Channel - maximum number of data files in a backup set. The default is 5. The product of files-per-channel and number of channels determines the maximum number of data files concurrently backed up by RMAN.
Logical Block Checking	Enable logical block validation by RMAN to detect specific types of corruption during a backup. When enabled, RMAN will test data and index blocks that pass physical corruption checks for logical corruption. This option typically adds some CPU overhead as mentioned in the Oracle documentation[440]. Examples of logical corruption detected are corruption of a row piece or index entry. Logical Block Checking is disabled by default.
Level Backup	Level backups should only be used to workaround Oracle bug 10146187 on physical standby sources. Switching from SCN to LEVEL mode will force a Level 0 backup. See Linking Oracle Physical Standby Databases (see page 1466) for more information. Level Backups are disabled by default.
Skip Available Space Check	Skips checks for available space during the initial load in the Delphix Engine. The initial load will fail if the database needs more space than what is available in the Delphix Engine. This setting is only recommended after consulting with Delphix Support.
Double SnapSync	Double SnapSync, or DoubleSync, takes two snapshots sequentially in order to improve the performance of subsequent operations, such as VDB provisioning and refresh. Performs the initial load without retrieving any archived logs, thereby creating a non-provisionable snapshot. At the completion of the initial load a second SnapSync that will retrieve logs will start. For more information, view the KB article[441].

440 https://docs.oracle.com/en/database/oracle/oracle-database/19/rcmrf/BACKUP.html#GUID-73642FF2-43C5-48B2-9969-99001C52EB50
441 https://support.delphix.com/Delphix_Virtualization_Engine/Oracle/KBA1706_What_is_DoubleSync%3F

Setting	Explanation
LogSync Policy Settings	LogSync Policy Settings are available after the dSource has been added. On the Configuration page, you can select the following options: • Enabled - LogSync fetches log files from the source database, enabling the ability to provision a VDB from a specific point in time or, a database change number (SCN in the case of Oracle databases). LogSync must be enabled for this provisioning functionality to work. • Archive Only, Archive and Online Redo - these settings determine whether LogSync fetches logs from archive storage in the source database file system, or both the file system and online redo logs. Online Redo mode is not supported for physical standby or RAC databases. It is also not supported if the online redo logs are raw devices or Oracle Automatic Storage Management devices. If LogSync detects any of the restricted cases it will automatically enter into Archive Only mode, regardless of the mode that was chosen. Additionally, Online Redo mode is not supported for mounted but not open primary databases. LogSync policy settings for Oracle pluggable databases must be set at their corresponding container databases.
Validated Sync	Oracle validated sync is disabled by default. When enabled, validated sync is performed immediately after every subsequent SnapSync. See Enabling Validated Sync for Oracle (see page 1464) for more information. Validated sync for Oracle pluggable databases is not supported in this release

9.7.9.4.1 Snapshot parameters

There are also parameters you can specify when creating a snapshot of an Oracle data source. These parameters are explained in the table below.

Parameter	Explanation
Force Full Backup	The Delphix Engine will by default perform an incremental backup, if for some reason a full backup is required (most often to resolve issues with datafile corruption), this option should be used.

Parameter	Explanation
Double Sync	Double SnapSync, or DoubleSync, takes two snapshots sequentially in order to improve the performance of subsequent operations, such as VDB provisioning and refresh. Performs the initial load without retrieving any archived logs, thereby creating a non-provisionable snapshot. At the completion of the initial load a second SnapSync that will retrieve logs will start. For more information, view the Using the Double Sync option for Oracle SnapSync (see page 1501)
Do Not Resume	During the initial SnapSync, if a failure is encountered, the Delphix Engine can resume the SnapSync at a later date, this option will cause the engine to not resume, but rather to start the initial SnapSync over again.

9.7.9.5 Prepare and upgrade a Non-MT Oracle dSource to MT

9.7.9.5.1 Overview

This article describes how Delphix works with production databases that are converted from non-multitenant to multitenant, and the workflow for this operation. Once a dSource is converted to a multitenant PDB, it will be able to share storage blocks with its non-multitenant predecessor, and Delphix will only store the incremental changes to the database. This operation is known as STConvertedToPDBAttach and is only available via CLI.

9.7.9.5.2 Environment requirements

1. Delphix Engine 6.0.10.0 or later.
2. Source host with a 11g or 12c non-multitenant source database.
3. PDB that is created as a result of converting the above non-multitenant database to multitenant. This PDB can reside on a different host.
4. Source host with a CDB that hosts the above PDB.

9.7.9.5.3 Workflow steps

At the end of the steps below, there should be 2 containers:

- The detached non-MT container which contains all the non-MT snapshots.
- A new container representing the converted PDB and all its subsequent snapshots with the container name as the PDB name. By default the new MT container will have the name of the PDB, or a parameter can be set by the customer to specify its name during attach.

Use these steps to perform this functionality.

1. Link the non-multitenant Oracle 11g or 12c source database as a dSource within Delphix. Please note, 11g sources Patch 31220011 included in the July 2020 bundle is needed and XML must be installed to the DB.
2. Before performing the conversion to multitenant, detach the dSource and shut it down. Visit the Detach/Re-Attach Oracle dSource (see page 1505) article for this step and step 7.
3. Convert the non-multitenant database to a PDB following procedures provided in the Oracle documentation. Delphix recommends that the new PDB have the same name as the original non-multitenant database. The new CDB name should be different from the original non-multitenant database.
4. Discover the CDB in Delphix Engine if not already present.
5. Refresh the environment to make sure the new PDB is discovered.
6. If the PDB name has the same name as the non-multitenant then the old non-MT container can be renamed or a parameter to give a different name to the PDB database should be used.
7. Attach converted PDB to non-multitenant dSource using CLI (attach-converted-pdb sample script).

Converted PDB rules BEFORE SnapSync

- Attachment of a converted PDB does not perform automatic SnapSync, even if linkNow=true is used. This is to allow detachment of a converted PDB without negative effects on a non-MT container.
- If for any reason it is necessary to re-convert the PDB (after re-opening to the non-MT database to perform changes), then the converted PDB should be removed instead of detached so that the STConvertedToPDBAttach can be done again to save space.
- Once the converted PDB is removed without performing SnapSync on it, it is possible to attach the non-MT container to its non-MT database.

Converted PDB Detach vs. delete BEFORE SnapSync

- Detaching a converted PDB will make it possible to attach the non-MT container to its non-MT database.
- However, if the non-MT database is reconverted and regular attach is used to attach the PDB, there will be no space de-duplication since the detach operation reverted the changes as it was before the STConvertedToPDBAttach operation.
- Detach of a converted PDB and subsequent attach should only be performed as long as the PDB has not been re-created/re-plugged to obtain space de-duplication.
- In general a converted PDB should be removed if no SnapSync is taken.

Converted PDB rules AFTER SnapSync

- Once a SnapSync is taken of a converted PDB, the non-MT container cannot be re-attached to any other container (neither to a non-MT database or to a converted pdb).
- This means that once SnapSync is taken, the non-MT container cannot be used to attach to a converted PDB anymore.
- Even if the converted PDB is removed after taking a SnapSync, attaching to the non-MT container will not be possible as the non-MT container is marked as already converted.

9.7.9.5.4 AttachSource/data properties

These are all the possible properties that can be configured during the attach of a converted PDB. Of these, only newPdbContainer and force have special meaning for a converted PDB. The rest are 'regular' attachment options.

```
machine.name database 'DBOMSR' attachSource attachData *> ls
Properties
    type: OracleSTConvertedToPDBAttachData (*)
    backupLevelEnabled: (unset)
    bandwidthLimit: (unset)
    checkLogical: (unset)
    compressedLinkingEnabled: (unset)
    config: (required)
    encryptedLinkingEnabled: (unset)
    environmentUser: (required)
    externalFilePath: (unset)
    filesPerSet: (unset)
    force: (unset)
    linkNow: (unset)
    newPdbContainerName: (unset)
    numberOfConnections: (unset)
    operations: (unset)
    oracleFallbackCredentials: (unset)
    oracleFallbackUser: (unset)
    rmanChannels: (unset)
```

attach-converted-pdb sample script

```
cd /database
select mydatabase
attachSource
edit attachData
set type=OracleSTConvertedToPDBAttachData
set newPdbContainerName=<name for the container instead of PDB name>
set config=<converted PDB name>
edit oracleFallbackCredentials
set type=PasswordCredential
set password=<database login password>
back
```

```
set oracleFallbackUser=<database login username>
set environmentUser=<environment user>
commit
```

9.7.9.6 Repository templates for Oracle databases

The primary use case and motivation for repository templates are to provide the Delphix administrator with control over the Oracle database parameters used during the staging phase of the VDB provisioning process. It is useful to be able to control these configuration parameters when the physical capabilities of the staging machine, such as CPU count and memory, are smaller than the physical capabilities of the machines hosting the source database repository.

The repository template is a relationship between three entities:

- A database repository – It is the Oracle home to which the vPDB will be provisioned on the target host.
- A database container – It is the source PDB.
- A VDB configuration template – A list of database configuration parameter names and values that you can save on the Delphix Engine to use at a later time. The VDB configuration template for the temporary CDB must contain at least the following database parameters:
    - **compatible**: Required for both Non-Multitenant and Multitenant sources. Must be set with the same value as in the dSource 'compatible' parameter.
    - **enable_pluggable_database**: Required for Multitenant sources only. Must be set to TRUE.

During the staging process, if you do not specify a repository template, then by default, the Delphix Engine will use the configuration parameters taken from the source database to configure the staged database. These parameters may not be appropriate, because the machine used for staging may be physically inferior to the machine hosting the source database.

Instead, the Delphix administrator can create a VDB configuration template, which would be appropriate for the physical machine hosting staging repository. (See VDB Config Templates (see page 1517)) Then the admin can create a repository template entry that will bind together the VDB configuration template, database repository, and database container. This instructs the Delphix Engine to use configuration parameters from the VDB configuration template whenever the database container is staged on the database repository specified, instead of the parameters on the source database.

Currently, repository template relations can only be created via the command-line interface (CLI) in the repository > template.

Provisioning a VPDB using a Repository Template

Perform the following procedure to provision a VPDB using a repository template.

1. Create a VDB configuration template using the Delphix GUI called aux_cdb_params with the following parameters set as a bare minimum.

```
compatible = 12.1.0.2
enable_pluggable_database = true
sga_target = 1551892480
```

The parameters required may differ given a specific environment's requirements from this but this should allow the startup of the auxiliary database.

2. Create a repository template called aux_cdb_templ using the CLI.

```
delphix repository template> create
delphix repository template create *> set name=aux_cdb_templ
delphix repository template create *> set container=PDB12C1
delphix repository template create *> set repository=OELC9/'/u01/app/oracle/
12.1'
delphix repository template create *> set template=aux_cdb_params
delphix repository template create *> commit
```

```
delphix repository template 'pb12c1rt'> ls
Properties
    type: SourceRepositoryTemplate
    name: aux_cdb_templ
    container: PDB12C1
    reference: REPOSITORY_TEMPLATE_REF-4
    repository: OELC9/'/u01/app/oracle/12.1'
    template: aux_cdb_params
```

In the above example:

a. **name** is what you want the template name to be.
b. the **container** is the source PDB.
c. the **repository** is the target environment/ORACLE_HOME that the CDB target is running from.
d. the **template** is the name of the database template created in the GUI.

3. Provision a virtual pluggable database (VPDB) via the Delphix GUI using the source PDB defined as the container in the repository template and provision the VPDB to the target environment defined by the repository (target environment/Oracle Home combination) in the repository template.

9.7.9.7 Using the PurgeLogs operation

Retention policy does not delete the archived logs which are

- Older than the log retention period(can be referred as expired logs) but are required to keep the snapshots provisionable
- Generated after the most recent snapshot
- Required to make retention-proof bookmark points provisionable

Under certain conditions, it may become necessary to delete more archived logs which cannot be deleted by retention policy. The purgeLogs operation can be used to perform this task.

PurgeLogs operation is capable of deleting the archived logs which are older than the log retention period but were not deleted by retention policy to keep the snapshots provisionable. If required, it can also delete the logs which are generated after the most recent snapshot. This operation, however, will always retain the archive logs required to keep the retention proof bookmark points or snapshots marked with ‘keep forever’ (pinned snapshots) provisionable.

PurgeLogs operation is a CLI only operation. For a multitenant dSource, this operation should be performed for CDB only as logs for CDB and all associated PDBs are managed by the CDB itself.

9.7.9.7.1 Example scenario

The retention policy does not delete the logs which are generated after the most recent snapshot. If no new snapshots are being taken within a reasonable period for any reason, the archive logs will continue to accumulate in the Delphix engine. This can lead to high memory and disk usage and can also make the engine unstable. This can also prevents new snapshots from being taken. By clearing these logs, the purgeLogs operation can help to free up the storage space in the Delphix engine.

> The purgeLogs operation does not function with VDBs or any non-Oracle data platforms at this time.

9.7.9.7.2 Strategies for archive logs deletion

Strategy 1

- For non-multitenant dSources, it will delete all the expired logs associated with snapshots as required until the target space to reclaim is achieved.
- For multitenant dSources, it will delete all the expired CDB logs associated with the CDB and PDB snapshots as required until the target space to reclaim is achieved.

It deletes the oldest snapshot logs and moves to the newest one in order for each Timeflow starting from the current Timeflow first. The logs required for retention proof bookmarks and pinned snapshots are not deleted.

Strategy 2

- For non-multitenant dSources, it will delete the logs which are newer than the latest snapshot in the current Timeflow to free up space which retention cannot free.
- For multitenant dSources, it will delete the logs which are newer than the latest PDB snapshot in the current Timeflow to free up space which retention cannot free. Here, the latest PDB snapshot indicates the most recent snapshot among all PDBs in CDB.

Deletion is done in the order from the most recent logs to the oldest one because of the possibility of Timeflow repair at a later time. The deletion is done based on the actual compressed file size of the log sequence on the Delphix Engine. If there is a retention proof bookmark, this operation preserves the logs required to provision from the retention proof bookmark This operation also spares the online in-flight log sequence since chunks of this sequence will keep coming in (this only applies for logsync mode set to Archive + Online Redo).

PurgeLogs operation takes three arguments :

storageSpaceToReclaim - The amount of space user wants to free up (eg; 20K, 5M, 1G etc). This is mandatory and non-zero. Actual space reclaimed is dependent on whether there are enough logs to be deleted. By default, **Strategy 2** is always in effect. The user has the ability to run **strategy 1** if their target space reclaim is not met, the user can run **strategy 1** using the **deleteSnapshotLogs** argument.

deleteSnapshotLogs - This is set to false by default. If set to true, then logs are deleted based on **strategy 1**. It is possible that no more logs are available for deletion as per strategy 1 and the target storage space to reclaim is not yet achieved. In such a scenario, logs would be deleted as per strategy 2 until the target is achieved or there are no more logs to delete. If this is set to false, then deletion is done based on **strategy 2** only.

dryRun - This is a boolean argument and set to true by default. If it is true, then the operation does not actually delete the logs. This argument can be used to see which snapshots and parts of the Timeflow will be affected by the operation. This operation returns the time flow point beyond which logs might be deleted and the set of non MT/PDB snapshots affected by this operation (and will not be provisionable afterwards). When the operation is run in non-dryrun mode, we compute the time range of the snapshot and provide the endpoint as the end of the truncated Timeflow. In dryrun mode, we return the endpoint of the first log which survived as the Timeflow point beyond which it has been truncated.

- Use this operation with `setopt format=json`, otherwise, the list of snapshots may be truncated if there are more than 3.
- **Strategy 1** is an optional step and to be used in the event **Strategy 2** does not clear enough space desired by the user. Please note that any snapshots affected by this step enabled can no longer be provisioned from. The user will be required to restore archivelogs to re-enable provisioning of the affected snapshot. To repair this function please use the TimeFlow Repair Tool.
- VDB transaction logs can be manually removed from the **archive** filesystem mounted on the target for each VDB.

This example shows how to free 50GB of space from container "example_container" using strategies 1 and 2:

1. Log into the CLI.
2. Go to the container using `/database`.
3. Set the output format into json using this statement: `setopt format=json`.
4. Select the Oracle non-multitenant/CDB container using `select example_container`.
5. Select operation `purgeLogs`.
6. Set the target space to reclaim to 50GB using `set storageSpaceToReclaim=50G`.
7. ***(Optional Step)*** `deleteSnapshotLogs=true`. **Warning:** this step should be used only in the event that Strategy 2 does not free enough space. This will remove the ability to provision from affected snapshots or require the restoration of archivelogs to enable provisioning. Use `set dryRun=true` to view the snapshots which will be affected without deleting the logs. Affected snapshots will be listed under the section `affectedSnapshots`.
8. Use `set dryRun=false` to enable the operation.
9. `Commit` operation.

9.7.9.7.2.1 Example usage

```
delphixengine> /database
delphixengine database> select example_container <---- Non-MT/CDB Container ---->
delphixengine database "example"> purgeLogs
delphixengine database "example" purgeLogs *> set storageSpaceToReclaim=50G
delphixengine database "example" purgeLogs *> set deleteSnapshotLogs=true
delphixengine database "example" purgeLogs *> set dryRun=false
delphixengine database "example" purgeLogs *> commit
    type: PurgeLogsResult
    affectedSnapshots:
        0:
            type: OracleSnapshot
            name: '@2014-08-05T23:29:57.576Z'
            consistency: INCONSISTENT
            container: dbdhcp3
            creationTime: 2014-08-05T23:29:57.576Z
            firstChangePoint:
                type: OracleTimeflowPoint
                location: 2825963
                timeflow: dbdhcp3/default
                timestamp: 2014-08-05T23:31:15.000Z
            latestChangePoint:
                type: OracleTimeflowPoint
                location: 2826079
                timeflow: dbdhcp3/default
                timestamp: 2014-08-05T23:31:15.000Z
            missingNonLoggedData: false
            namespace: (unset)
            reference: ORACLE_SNAPSHOT-75
            retention: 0
            runtime: (unset)
            timeflow: dbdhcp3/default
            timezone: America/New_York,EDT-0400
            version: 11.2.0.2.0
    truncatePoint:
        type: OracleTimeflowPoint
        location: 3176794
        timeflow: dbdhcp3/default
        timestamp: 2014-08-07T19:58:55.000Z
```

This output shows that the affected snapshot is '@2014-08-05T23:29:57.576Z' and the snapshot will not be provision-able. The truncate point indicates the last SCN/timestamp that can be provisioned.

9.7.9.8 Advanced linking

This section covers the following topics:

- Enabling validated sync for Oracle (see page 1464)

- Linking Oracle physical standby databases (see page 1466)
- Specifying external data directories for Oracle dSources and VDBs (see page 1474)
- Linking to Oracle dSources with RMAN compression or encryption enabled (see page 1477)
- Upgrading dSources after an Oracle upgrade (see page 1477)
- Oracle source continuity (see page 1479)
- Oracle liveSources (see page 1481)
- Oracle liveSource user workflows (see page 1483)
- Oracle RAC dSource node addition or deletion (see page 1490)
- Using the Double Sync option for Oracle SnapSync (see page 1501)
- Linking dSources from an encrypted Oracle database (see page 1505)
- Detaching and re-attaching Oracle dSources (see page 1505)
- Provisioning from a replicated Oracle dSource (see page 1509)
- Detaching and re-attaching a PDB dSource from a data guard site (see page 1510)
- Working with Oracle snapshots (see page 1511)
- Moving the PDB to a new CDB (see page 1513)

9.7.9.8.1 Enabling validated sync for Oracle

This topic describes the validated sync process for Oracle databases using both the Delphix Management application and Command Line Interface (CLI).

Traditional Oracle dSource snapshots require some recovery during provisioning. By configuring validated sync for Oracle, the Delphix Engine selects a compatible Oracle installation and applies the recovery necessary to provision a snapshot immediately after each SnapSync. Snapshots that have been through this validated sync process step do not require recovery during provisioning.

> The Delphix Engine may be unable to perform validated sync on a physical standby database in Real-Time Apply mode. This is because the standby may apply changes before copying the logs that contain those changes. Without the logs necessary to perform recovery, validated sync cannot be executed. However, you can still provision the snapshot when the archive logs become available on the standby.

9.7.9.8.1.1 Prerequisite: designating a staging host

In order to validate an Oracle dSource snapshot after a sync, the Delphix Engine requires a host with an Oracle installation that is compatible with the dSource. This machine is known as the **staging** host. You must explicitly designate which machines you want the Delphix Engine to use as staging hosts. All machines that have been marked as staging hosts are added to a pool. During sync validation, the Delphix Engine will select a compatible host from the pool, export the requisite archived redo logs and data files, and execute Oracle media recovery on the host. Follow these steps to designate a staging host.

1. Log into the Delphix Management application.
2. Click **Manage**.

3. Select **Environments**.
4. In the **Environments** panel, select the environment you want to designate as staging.
5. Select the **Databases** tab.
6. Scroll down to the installations you want to designate as staging and edit the **Installation Details** by clicking on the **pencil** icon.
7. Select the **Use as staging** checkbox.
8. Select the green checkmark to confirm your change.

To configure validated sync for multiple dSources with different Oracle versions, you must designate a compatible staging source for each. If multiple compatible staging sites exist, the Delphix Engine will select one at random.

The validated sync process will consume some resources on the staging host when snapshots are taken. Designating a performance-critical host as a staging host is not recommended.

The default OS user for the staging host must have access to the Oracle installation that will be used to perform recovery during validated sync.

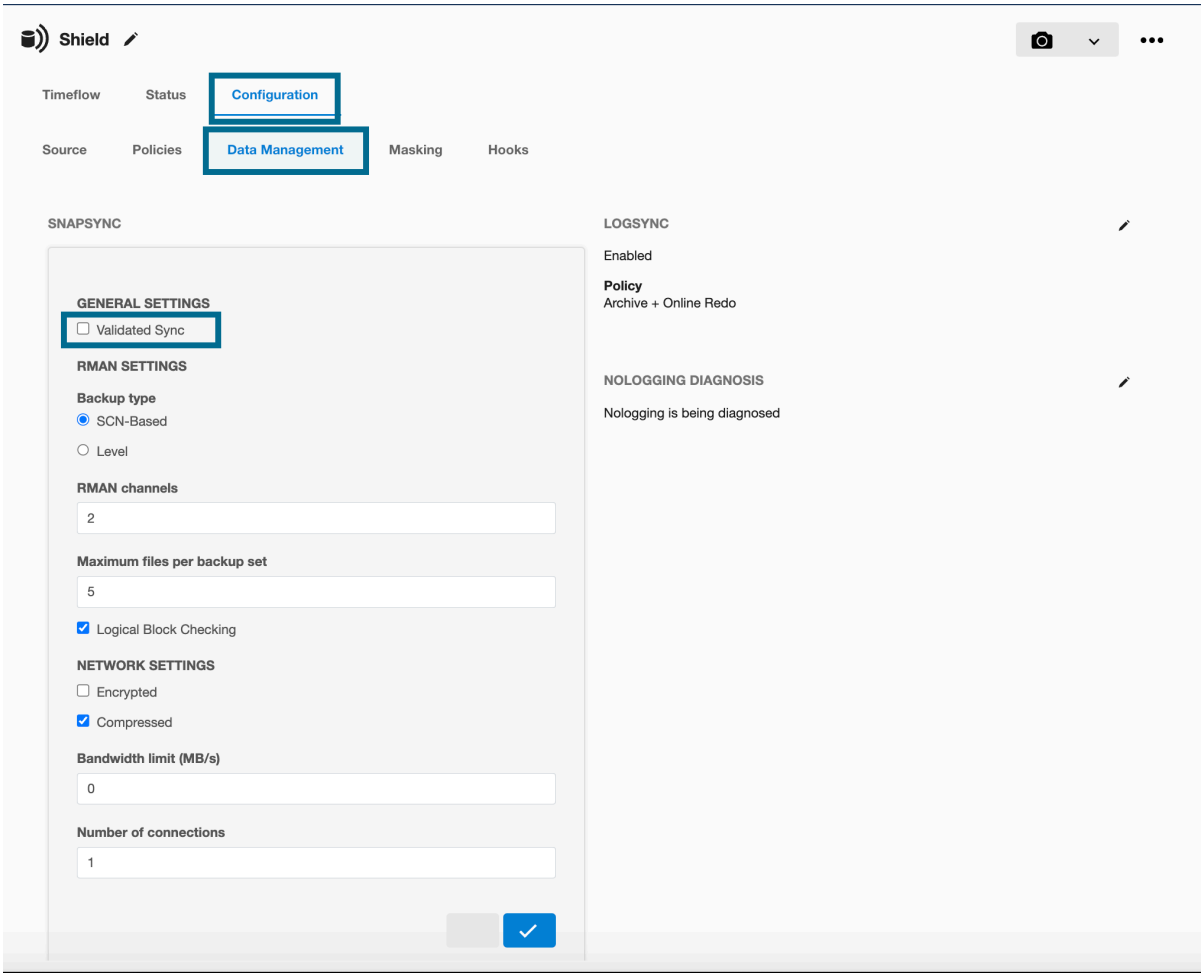
Oracle pluggable database
Validated sync for Oracle pluggable databases is not supported in this release.

9.7.9.8.1.2 Enabling validated sync

Oracle validated sync can be enabled at link time or on any existing dSource. When adding the dSource (at link time), in the Data Management tab, select Show Advanced, and select the Enable checkbox for Validated Sync.

For an existing dSource:

1. Log into the Delphix Management application using **Delphix Admin** credentials.
2. Click **Manage**.
3. Select **Datasets**.
4. In the **Datasets** panel, select the dSource for which you want to enable sync validation.
5. Select the **Configuration** tab and then select the **Data Management** sub-tab.
6. Select the pencil icon located next to **Snapsync**.
7. Select the **Validated Sync** checkbox.



8. Select the checkmark to confirm your change.

9.7.9.8.2 Linking Oracle physical standby databases

This topic describes special considerations for linking Oracle physical standby databases.

The Delphix Engine supports linking both physical and logical standby databases. In previous versions of the Delphix Engine, limitations were placed upon support for Oracle RAC physical standby databases in Real-Time Apply mode. In version 3.0 of the Delphix Engine, these restrictions were lifted.

> **Using block change tracking (BCT) on a physical standby database**
> In general, Delphix recommends enabling Block Change Tracking (BCT) on a primary or standby source database. See Physical Standby Database Support Matrix[442] below for restrictions on enabling BCT on a standby database.

442 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/linking-oracle-physical-standby-databases

BCT is available from Oracle release 11.1.0.6 onward for physical standby databases **only** if they are licensed for the Active Data Guard option.

- Release 11.1.0.6 is unstable for the BCT on physical standby feature
- Release 11.1.0.7 requires a patch for Oracle bugs 7613481, 9068088
- Release 11.2.0.2 requires patches for Oracle bugs 10170431, 12312133
- Release 11.2.0.3 requires patches for Oracle bugs 12312133, 16052165

Patches required
Enabling BCT on a physical standby database without these patches is **not recommended** because of serious performance and stability issues.

BCT on a primary database has been stable since Oracle version 10.2.0.5. In order to make use of BCT (>11.2.0.4), The Physical Standby Database must be in a "Managed Recovery Mode", i.e. achieved using "ALTER DATABASE RECOVER **MANAGED** STANDBY DATABASE".

9.7.9.8.2.1 Physical standby database support matrix

Oracle Version	Apply Mode	Notes
11.x and 12.x in **Level Backup** mode; 12.x in **SCN Backup** mode	Archive Apply mode	No special restrictions.
	Real-Time Apply mode (RTA)	LogSync must be enabled for all database versions if RTA is enabled
11.x in **SCN Backup** mode	Archive Apply mode	If the Physical Standby Database is at version 11.2.0.4 or above, no special actions are required. Due to Oracle bug 10146187, **Redo Apply** must be stopped and the database opened in **read-only** mode during SnapSync. See the section *Stopping and Restarting Redo Apply* below for more information.

Oracle Version	Apply Mode	Notes
	Real-Time Apply mode	If the Physical Physical Standby Database is at version 11.2.0.4 or above, no special actions are required. LogSync must be enabled. Due to Oracle bug 10146187, **Redo Apply** must be stopped and the database opened in **read-only** mode during SnapSync. See the section *Stopping and Restarting Redo Apply* below for more information. In addition, to avoid Oracle Bug 13075226, which results in a hang during the restart of Redo Apply, Delphix requires disabling using BCT on the standby database. The hang occurs when BCT is enabled on a standby database that uses SCN backup mode. **Patch required** SnapSync will fail If running Oracle 11.2 before 11.2.0.4 when using SCN backups and real-time apply mode, Use level based backups instead. If the Oracle installation has already been patched for Oracle bug 13075226, or once the patch is applied, use the CLI to update the repository for this installation so that applied Patches include Oracle bug number 13075226. If the repository does not indicate that Oracle bug 13075226 for the repository has been addressed, SnapSync will not be possible when using SCN backups and real-time apply. See *Updating Repository for Oracle applied patches with the Command Line Interface* below for details on how to update the repository.
18.0.0.0 - 19.3.0.0	Real-Time Apply mode	Oracle patch 29056767 installed, Failure to install this patch may result in SnapSync failure with error.

9.7.9.8.2.2 Level backup mode for SnapSync

By default, the Delphix Engine's SnapSync feature uses **SCN Backup** mode and is designed to not interfere with other backups that may already be in use. However, in cases where RMAN is not being used outside of the Delphix Engine, the Delphix Engine can use the **Level Backup** mode that improves SnapSync behavior on Oracle 11g physical standby databases. In this mode, **redo apply** does not have to be stopped during

SnapSync. See Advanced Data Management Settings for Oracle dSources[443] for more information about SnapSync settings.

Requirements for using level backup mode

Customer not backing up their physical standby with RMAN:

- Set CONTROL_FILE_RECORD_KEEP_TIME to 365

OR all of the following:

- Physical standby database running Oracle 11.2.0.2 or later version
- All RMAN backups must use tags
- RMAN CROSSCHECK commands must specify tags
- RMAN DELETE commands must specify tags
- RMAN DUPLICATE commands must specify tags
- Set CONTROL_FILE_RECORD_KEEP_TIME to 365

> Failure to meet all of these requirements will cause external RMAN backups to be incomplete or result in corrupt SnapSync snapshots. Switching from SCN to LEVEL mode will force a new LEVEL 0 backup.

9.7.9.8.2.3 Stopping and restarting redo apply

Oracle bug 10146187 requires stopping of redo apply before an SCN-based incremental backup can be issued. These scripts can be used as pre- and post-scripts during the dSource linking process to stop and restart **Redo Apply**.

- SnapSync pre-script: disableStandby.sh.templa… [444]

443 https://cd.delphix.com/docs/latest/advanced-linking
444 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149524884?preview=%2F149524884%2F149524909%2FdisableStandby.sh.template

- SnapSync post-script: enableStandby.sh.template[445]

These scripts must be modified for local use, particularly in regard to whether the physical standby database operates in MOUNTED or OPEN mode.

> These scripts are only required if your Physical Standby database is on a version of Oracle 11g that does not have the fix for Oracle bug 10146187 (consult Oracle documentation for details).
>
> Failure to properly customize these scripts could violate your Oracle license terms by running redo apply on an open database, which requires an Oracle Active Data Guard license.

9.7.9.8.2.4 Linking and provisioning a Mounted Standby

> Warning:
> When you link a standby database in the mounted mode and have not been opened read-only, the data files for temporary tablespaces will be present in `v$tempfile` but will not actually be created yet, and therefore will indicate 0 file size. As a result, any VDB provisioned from a snapshot taken in this state will end up with the same tablespaces and datafiles created, but with a default file size of 52428800 bytes.

For databases that are in the **mounted** state, the Delphix database user account must be **SYS** (having the **SYSDBA** role), **SYSBACKUP** (having the **SYSBACKUP** role) or **SYSDG** (having the **SYSDG** role).

445 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149524884?preview=%2F149524884%2F149524897%2FenableStandby.sh.template

> **SYSBACKUP** and **SYSDG** roles are only available in Oracle 12.1 and later releases.

However, for an **open** standby (Active Data Guard) database, only a regular database user account is required.

Connecting to a **mounted** standby with a **SYS** user account requires that the mounted standby be configured with a password file. Delphix does not capture the password file during SnapSync, and for this reason, cannot provision or sync validate a database with a SYS user. A secondary, regular database user account can be specified through either the **Delphix Management** application or CLI. This database user will then be used to connect to the database during provisioning and validated sync. Note that the **SYS** user is still required to perform snapshots of the source database.

In the **Delphix Management** application, the **non-SYS user** can be specified from within the **Add dSource** wizard, or on the back of the Oracle dSource after linking.

9.7.9.8.2.5 Configuring a Standby PDB in Mount Mode

> This procedure is optional and only applies if SCM is disabled.

To configure a standby PDB in the mount mode, you must also provide a non-SYS user for both the CDB and the PDB. The PDB non-SYS user can only be added via the CLI. You must perform a fresh SnapSync after adding the non-SYS user.

Pre-requisite

In order for Delphix Engine to connect, you must configure a static listener configuration for the PDB. You can configure a static listener by adding a configuration into `listener.ora` and restarting the listener.

```
SID_LIST_LISTENER=
(SID_DESC=
(GLOBAL_DBNAME=CDOMLOSR4F71PDB1)
(SID_NAME=stby18c)
(ORACLE_HOME=/u01/app/oracle/product/18.0.0.0/dbhome_1)
)
)
```

In the above example configuration, `GLOBAL_DBNAME` is the PDB name and `SID_NAME` is the SID of the CDB.

Procedure

Run the following commands to configure a PDB and CDB in the mount mode.

1. Update PDB non-SYS user.

```
# Update PDB nonsys user
delphix> /sourceconfig
delphix sourceconfig> select RH74PDB04
delphix sourceconfig 'RH74PDB04'> update
delphix sourceconfig 'RH74PDB04' update *> set nonSysUser=delphix
delphix sourceconfig 'RH74PDB04' update *> set
nonSysCredentials.type=PasswordCredential
delphix sourceconfig 'RH74PDB04' update *> set
nonSysCredentials.password=delphix
delphix sourceconfig 'RH74PDB04' update *> commit;
```

2. Update CDB non-SYS user.

```
# Update CDB nonsys user
delphix> /sourceconfig
delphix sourceconfig> select rh74cdb2
delphix sourceconfig 'rh74cdb2'> update
delphix sourceconfig 'rh74cdb2' update *> set nonSysUser=delphix
delphix sourceconfig 'rh74cdb2' update *> set
nonSysCredentials.type=PasswordCredential
delphix sourceconfig 'rh74cdb2' update *> set
nonSysCredentials.password=delphix
delphix sourceconfig 'rh74cdb2' update *> commit;
delphix sourceconfig 'rh74cdb2'>
```

3. Perform sync of the PDB.

```
# Perform sync of PDB
delphix> /database
delphix database> select RH74PDB04
delphix database 'RH74PDB04'> sync
```

9.7.9.8.2.6 Setting the Non-Sys User on the Oracle dSource

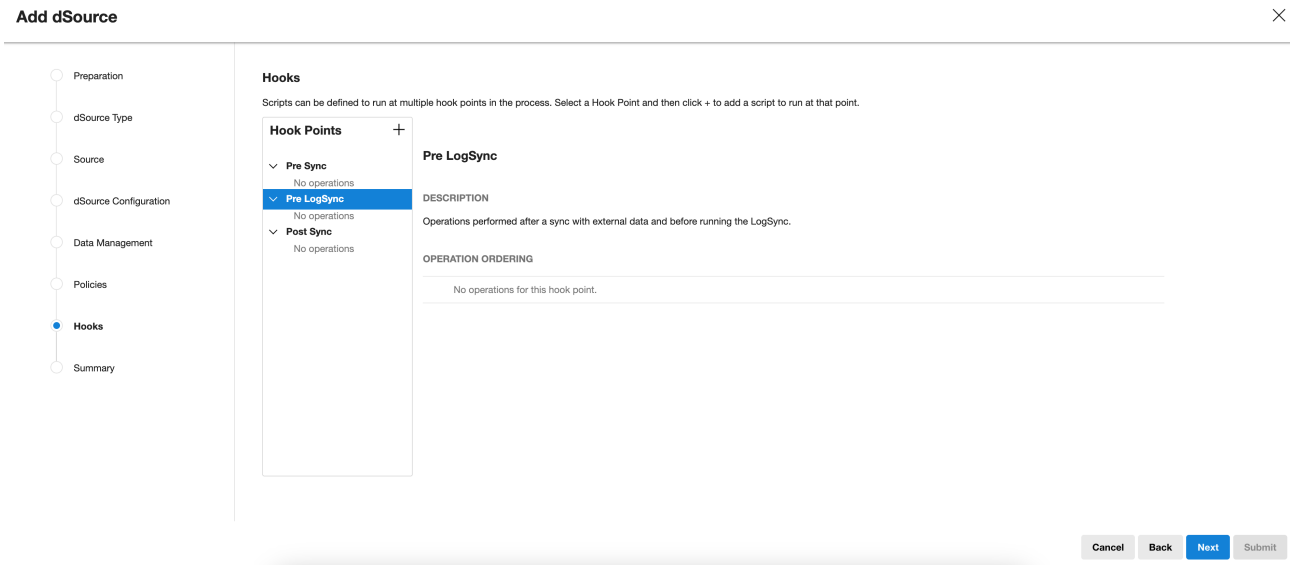
1. Create the delphix_db user in the primary database.
2. Log into the Delphix Management application.
3. From the **Manage** menu, select **Datasets.**
4. From the Configuration tab select the Oracle dSource for which you want to add a **non-SYS** user.
5. Click the dSource's icon to open the dSource information pane.
6. Click the Edit button next to **Non-SYS User**.
7. Enter a non-SYS user and credentials that exist on the standby.

8. Click the **Accept** button to save this user and associated credentials.

The non-SYS user will be used to connect to all VDBs provisioned from snapshots of this dSource that are created after the non-Sys user has been set.

PreLogSync Hook

If the datafiles in the snapshot are inconsistent, latest archive logs are needed to recover them during provisioning. The logs are fetched by the LogSync operation which immediately follows the SnapSync operation. Since Standby Redo Logs (SRL) are not processed by LogSync operation, until the existing SRL at the time of the SnapSync operation is archived, the snapshot is not marked as provisionable.
A PreLogSync hook can be used to archive the redo log file in the primary database which will archive the SRL on the standby database. The PreLogSync hook runs after the SnapSync operation but before the LogSync operation.



9.7.9.8.2.7 Updating repository for applied patches with the command line interface

1. Select the repository of the database

```
delphix> repository select '/opt/app/oracle/product/11.2.0.2/db_1'
```

2. Execute the `update` command.

```
delphix repository ''/opt/app/oracle/product/11.2.0.2/db_1''> update
```

3. Set `appliedPatches` to list current patches applied to the repository.

```
delphix repository ''/opt/app/oracle/product/11.2.0.2/db_1''update *> set
appliedPatches=13075226
```

4. `Commit` the operation.

```
delphix repository ''/opt/app/oracle/product/11.2.0.2/db_1''update *> commit
```

9.7.9.8.2.8 Setting the Non-Sys User with the Command Line Interface

1. Select the `source config` of the mounted standby.

```
delphix> sourceconfig select pomme
```

2. Execute the `update` command.

```
delphix sourceconfig "pomme"> update
```

3. Set the `nonSysUser` and `nonSysCredentials` to a non-SYS user that exists on standby.

```
delphix sourceconfig "pomme" update *> set nonSysUser=<non-sys-username>
delphix sourceconfig "pomme" update *> set
nonSysCredentials.type=PasswordCredential
delphix sourceconfig "pomme" update *> set nonSysCredentials.password=<non-sys-
password>
```

4. Commit the operation.

```
delphix sourceconfig "pomme" update *> commit
```

9.7.9.8.3 Specifying external data directories for Oracle dSources and VDBs

This topic describes the process for including external data files with dSource snapshots and VDBs.

In the following places, you can specify the directory for any external data files that should be included with dSource snapshots:

- During the dSource linking process click on the **Advanced** section of the **Data Management** screen
- After you have created the dSource go to the **Configuration** tab

9.7.9.8.3.1 External file import for the Delphix engine and VDBs

The Delphix Engine will not fetch external tables or external data types such as BFILE. Instead, in order to link external data files to the source database and make it available to the Delphix Engine, you must create a directory in the file system and the database. Any data files in the directory you specify will be applied, recursively, to the dSource.

External data will be provisioned to each VDB that is created from this dSource. You will need to update the external file/data type definition to point to the new location after creating VDBs. Provisioning a VDB with external data creates a directory named **external** in the VDB mount point location.

9.7.9.8.3.2 Configuring the rsync command location for an environment

Files from the external data directory are fetched using the rsync command (via rsyncd[446]) installed in the source environment. In order to SnapSync a dSource with an external data directory, rsync must be installed in the source environment. If rsync is installed in a non-standard location, the path to the rsync command can be configured in the **Environment Details** for the source environment on the **Environment Management** screen.

9.7.9.8.3.3 Example of attaching and redirecting external data files for Oracle databases

This example uses two environments:

1. **172.16.200.446** as the source environment **dinosaur** as the source database
2. **172.16.200.447** as the target environment **vdino** as the target database

Linking a dSource

1. Create an external data directory and an external data file, and attach the directory to the source database.
   a. Log into **172.16.200.446** as the environment user.
   b. Create a physical directory on the source environment. `$ mkdir /work/extdata`
   c. Create a directory in Oracle.

```
$ sqlplus / as sysdba
  SQL> create or replace directory extdata as '/work/extdata';
```

   d. Create a text file `/work/extdata/exttab.dat.`

```
$ cat > /work/extdata/exttab.dat
    1, aaa
    2, bbb
    3, ccc
    ^C
```

   e. Create an external table `exttab`.

```
$ sqlplus / as sysdba
    SQL> create table exttab (id number, text varchar2(10))
```

446 https://manpages.ubuntu.com/manpages/focal/man5/rsyncd.conf.5.html

```
      2  organization external (default directory extdata location('extt
ab.dat'));
```

f. Query the table.

```
SQL> select * from exttab;
        ID TEXT
---------- ----------
         1  aaa
         2  bbb
         3  ccc
```

2. During the process of linking the dSource to the **Dinosaur** database, or in the dSource's **Configuration** tab after creating the link, enter `/work/extdata` in the **External Data Directory** field.

Provisioning a VDB

1. Provision **vdino** from **Dinosaur**.
2. Modify the directory **extdata** in **vdino**
   a. Log into the target environment **172.16.200.447**
   b. Set `SID` to `vdino`

```
$ export ORACLE_SID=vdino
```

   c. A query to `exttab` will fail.

```
$ sqlplus / as sysdba
    SQL> select * from exttab
select * from exttab
*
ERROR at line 1:
ORA-29913: error in executing ODCIEXTTABLEOPEN callout
ORA-29400: data cartridge error
KUP-04063: unable to open log file EXTTAB_23394.log
OS error No such file or directory
ORA-06512: at "SYS.ORACLE_LOADER", line 19
```

3. Modify directory to the new location.

```
SQL> create or replace directory extdata as '/mnt/provision/vdino/external';
```

4. Query `exttab` again.

```
SQL> select * from exttab;
        ID TEXT
---------- ----------
         1  aaa
         2  bbb
         3  ccc
```

9.7.9.8.4 Linking to Oracle dSources with RMAN compression or encryption enabled

This topic describes the behavior of the Delphix Engine when linking to a dSource with RMAN compression or encryption enabled.

In earlier versions of the Delphix Engine, the dSource linking process would fail if RMAN compression or encryption was enabled. In order for the linking process to complete, the administrator was required to ensure that compression was not enabled for device type **SBT_TAPE**, and that encryption was also not enabled.

With the Delphix Engine, linking a dSource succeeds if compression or encryption is enabled, but the RMAN backup that creates the dSource will not be compressed or encrypted. This is true in the case where the administrator has enabled compression for tape, and in the case where the administrator is using OSB and has enabled encryption for tape.

You can check the RMAN compression and encryption settings with the commands `show device type` and `show encryption for database`, respectively.

If compression is enabled on a source environment, then it may need to be enabled for the target environment. Compression should be enabled for target environments if all segments in a source environment utilize **NOCOMPRESS** or **COMPRESS="BASIC"**.

9.7.9.8.5 Upgrading dSources after an Oracle upgrade

This topic describes how to upgrade dSources after an Oracle database upgrade.

9.7.9.8.5.1 Prerequisites

Do not suspend LogSync on the Delphix Engine during an Oracle upgrade of the source environment. LogSync will detect the Oracle version change. It will then notify you to refresh the host and use the Update icon on the configuration tab for all the associated dSources and VDBs (see below). Follow all Oracle instructions and documentation.

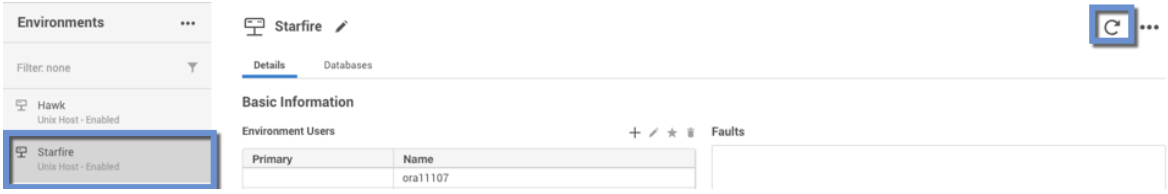
> ⚠ In order to upgrade your Oracle dSource from Non-Multi-Tenant (Non-MT) to Multi-Tenant(MT) database, please refer to Prepare and Upgrade a Non-MT Oracle dSource to MT (see page 1477) section.

There are 2 ways to apply a PSU (Patch Set Update)/Oracle upgrade:

- Apply to existing ORACLE_HOME. Best if on Delphix v4.1.x or higher.
- Create a new ORACLE_HOME (or clone an existing one) and then apply the PSU to the new ORACLE_HOME

9.7.9.8.5.2 Applying to an existing ORACLE_HOME

1. Following Oracle documentation, patch the ORACLE_HOME and the database.
2. Click the refresh icon on the top right of the selected environment.



3. On the Refresh Environment pop-up window, click **Refresh**.
4. Under the **configuration** tab, go to the PDB and verify that the **Repository** and/or **Version** has been updated.

9.7.9.8.5.3 Creating a new ORACLE_HOME

1. Refresh the environment from the Delphix Management application.
2. In the **Databases** tab, verify that the new ORACLE_HOME appears as an ORACLE Installation.
3. Following Oracle documentation, patch the production database, etc.
4. Click **Manage**.
5. Select **Datasets**.
6. Expand the group(s) containing all the non-multitenant and multitenant dSources.
7. Select **dSource**. For upgrading a PDB or multitenant dSource, select the container CDB **dSource**.
8. Click the **Configuration** tab.
9. From the **Actions** menu (...), select **Upgrade** to switch the ORACLE_INSTALLATION to the new one.
10. Under the **configuration** tab, go to the PDB and verify that the **Repository** and/or **Version** has been updated.

Updating the Oracle user after an upgrade
There may be cases when you upgrade the Oracle home and the Oracle User (the OS user who owns the Oracle binary) is a different user than the previous Oracle User. You will then need to update the Oracle User for each environment, and then re-connect each dSource and VDB to the

upgraded Oracle home using the new Oracle User.
The new Oracle User must be in the same OS group (for example, dba or oinstall) as the previous one.

1. Log into the Delphix Management application.
2. Click **Manage**.
3. Select **Environments**.
4. Select the **environment** where you want to add the user.
5. Next to **Environment Users**, click the **Pencil** icon to edit the new user.
6. Follow the procedure to upgrade the dSources and VDBs described in this topic.

9.7.9.8.5.4 Post-requisites

After upgrading the dSource to a new major release of Oracle (11.2.0.4 to 12.1 for example), you must re-run the [447] script.

9.7.9.8.6 Oracle source continuity

9.7.9.8.6.1 Overview

In earlier versions of the Delphix Engine, when an Oracle database underwent a resetlogs operation, you were required to re-link the Oracle source. This meant that you had to completely back up the Oracle database and store it again on the Delphix Engine. If any virtual databases (VDBs) were provisioned from the dSource and needed to be saved, you had to rename and save the old dSource, resulting in a possible doubling of storage space consumed on the Delphix Engine. The old VDBs could not be refreshed to the relinked dSource.

447 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149458890?preview=%2F149458890%2F149458903%2FcreateDelphixDBUser.sh

Beginning with Delphix Engine version 4.1.1.0/4.0.6.0, the Oracle database no longer requires you to re-link sources after a resetlogs operation. The Delphix Engine will detect this condition, automatically take a new full backup, and create a new TimeFlow for the next SnapSync of the source. Benefits of the Oracle Source Continuity feature include:

- Lower storage costs and easier administration.
    - Only the changed blocks of the new SnapSync backup will be stored on the Delphix Engine. Because of the way the Delphix Engine handles duplicate blocks, the full backup likely has a storage requirement similar to an incremental backup.
- Existing VDBs provisioned from previous snapshots for the source will remain.
    - You can use and refresh those VDBs to the new snapshot

The improved user workflow replaces the old user workflow, which directed users to troubleshoot when SnapSync would fail. Begin Oracle Source Continuity in the following way:

1. The database undergoes a resetlogs operation.
2. If LogSync is enabled, it generates a fault and stops.
3. Start SnapSync. The SnapSync does a full restore of the database to a new TimeFlow, clears the fault, and restarts LogSync. If you created VDBs prior to the resetlogs operation, they will still exist after the SnapSync; you can refresh them from the new snapshot.

9.7.9.8.6.2 Creating a new timeFlow

When LogSync detects the restlogs operation, it will still stop and generate a fault. LogSync must stop because a new timeline has been created on the database. This usually happens because the database has been rewound to a past point. The transaction logs being generated on the new timeline are out of sync and conflict with logs from the old timeline. The data files are also out of sync with the data files on the Delphix Engine. You must create a corresponding new Timeflow on the Delphix Engine to store the new logs and new versions of the data files. This requires taking a new backup of the database.

Once LogSync detects the reset operation and throws the fault, no more changes will be retrieved from the database until you start a new SnapSync. This SnapSync will take a full backup, clear the fault, and restart LogSync. Only the new snapshot and Timeflow will be visible in the dSources view. Previous snapshots and Timeflow will still exist and be visible through the command-line (CLI) Capacity screen.

The following CLI output shows that the old and new Timeflow and snapshots are still available. The name of the original Timeflow for the database is "default." The name of the new Timeflow that was created during the SnapSync is "CLONE@2015-01-15T17:07:20."

```
delphix> /TimeFlow list display=name,container
```

```
NAME                          CONTAINER
```

```
'CLONE@2015-01-15T17:07:20'  dbdhcp1
```

```
default                       dbdhcp
```

```
delphix> /snapshot list display=name,container,TimeFlow
```

```
NAME                          CONTAINER  TimeFlow
```

```
'@2015-01-16T00:50:08.784Z' dbdhcp1 default
```

```
'@2015-01-16T00:52:13.685Z' dbdhcp1 default
```

```
'@2015-01-16T00:53:46.873Z' dbdhcp1 default
```

```
'@2015-01-16T00:55:18.079Z' dbdhcp1 default
```

```
'@2015-01-16T01:08:02.411Z' dbdhcp1 'CLONE@2015-01-15T17:07:20'
```

The old snapshots and Timeflow will still be subject to logfile and snapshot retention policies. You can also delete the snapshots manually. In addition, you can use the CLI to provision from the old TimeFlow.

9.7.9.8.7 Oracle liveSources

9.7.9.8.7.1 Oracle liveSources overview

Prior to Delphix Engine version 4.2, users ran reports against virtual databases (VDBs) that they created with the Delphix Engine. Although this workflow helped them offload the reporting load from production, the data in the VDBs was not updated asynchronously. If users wanted newer data, they had to stop their reporting applications, refresh their VDBs, and resume. In the current release, you can run reports against data that is constantly being updated. There is one live data feed for each source database that is linked as a dSource on the Delphix Engine. You can point your reporting applications to this live feed. Additionally, you will continue to have all existing Delphix functionality from the dSource, such as creating read/write VDBs.

9.7.9.8.7.2 Understanding Oracle liveSources

Oracle LiveSources leverage native Oracle Active Data Guard technology to keep a standby database up-to-date with changes happening on the source. The standby database is kept open for reads while it applies changes from the source. You can now connect to this standby database for real-time reporting needs. Using Delphix in conjunction with Active Data Guard gives you the ability to get both live up-to-date data and historical points in time from which you can provision virtual databases.

9.7.9.8.7.3 Understanding how to use Oracle liveSources

Oracle liveSources provide a read-only live data stream from Delphix

You can convert an Oracle dSource to a LiveSource, which is a real-time read-only feed of the linked source. You can access the LiveSource using a JDBC string. Internally, a LiveSource is a standby database instance tracking the Linked Source in real-time managed mode and opened in read-only mode.

Understanding Oracle liveSources with data age and threshold

One of the important utilities of a LiveSource is that it provides a real-time feed of the linked source. In some instances, due to slow networks or other reasons, the LiveSource might fall behind the linked source that it is tracking. When adding a LiveSource, you can specify a data age threshold. If the LiveSource falls behind the linked source by more than the data age threshold, the Delphix Engine will generate a fault and inform you.

The LiveSource information is displayed on the Status tab. Delphix continuously monitors the standby instance and notifies you of any abnormalities.

- Live Data Status - Indicates that the LiveSource standby database instance is actively receiving data from the source database
- Data Age - Displays the data age of the LiveSource

You can change the Data Age Threshold at any time by updating the threshold value located on the **Configuration > Data Management** tab.

9.7.9.8.7.4 Oracle liveSources quickly sync with consistent snapshots

Taking snapshots of a LiveSource is instantaneous since the standby database for the LiveSource is constantly receiving data from the source database and recovering it. Taking snapshots occurs instantaneously by taking a filesystem level snapshot of the data on the Delphix Engine without requiring an RMAN backup of the source database. All LiveSource snapshots are consistent; as a result, provisioning from LiveSource snapshots is fast, because no database recovery needs to happen.

9.7.9.8.7.5 Oracle liveSources use resync and apply

Resync is a way to refresh the LiveSource to the current point in the linked source. The following situations require a Resync to be performed:

- There are unresolvable gaps in the log sequence – for example, logs from the source database deleted before the primary database could ship them over to the LiveSource standby.
- The source database was taken through a point in time recovery/flashback, resulting in a changed incarnation.
- The source database contains non-logged changes. In this case, a Resync is needed only if you are interested in moving the non-logged data over to the LiveSource.
- The LiveSource is significantly behind the source database due to network communication issues or a large number of writes.

LiveSource Resync is a two-step operation of:

- **Start LiveSource Resync** – Start Resync performs an incremental backup of the source database to transfer the latest changes to the Delphix Engine. This operation does not affect the availability of the LiveSource
- **Apply LiveSource Resync –** Applying the Resync data will perform one more incremental backups from the source database to ensure up-to-date data, and recreate the LiveSource instance while preserving all the configurations. This operation requires downtime for the LiveSource.
- **Discard LiveSource Resync –** If the prepared resync data is no longer needed or resync has become obsolete (for example, another controlled change has been done on the source database), you can discard the current resync data with Discard LiveSource Resync. The next Resync will refetch data from the source database.

9.7.9.8.7.6 Pre-requisites: configuration and installation of staging environments to host a standby database

Oracle active data guard required

The LiveSource feature requires an Active Data Guard license. Delphix uses Active Data Guard to replicate changes from the source database that it creates on the staging environment.

Network requirements

LiveSource requires a Data Guard connection between the source and the standby database which utilizes TNS listeners associated with the databases.

Database requirements

LiveSource requires Enterprise Edition of Oracle Database.

LiveSource supports:	LiveSource does not support:
Oracle support matrix	
Oracle 11g and non-multi tenant Oracle 12c, 18c, and 19c	Oracle Standard Edition on the source and staging environments
Physical and standby source databases	LiveSources running on a RAC

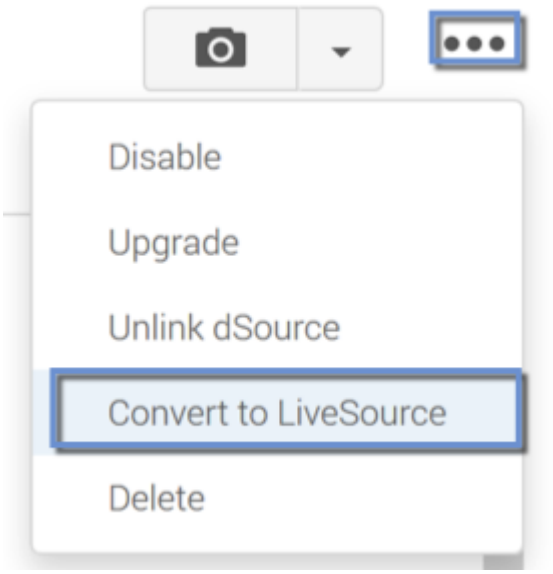
9.7.9.8.8 Oracle liveSource user workflows

Please use the following documentation as a guide to identify and act on common Oracle liveSource User Workflows. The following table of contents includes steps for how to convert a dSource into a liveSource, a provision from a liveSource, sync a liveSource, convert a liveSource back to a dSource, and many other data procedures.

9.7.9.8.8.1 Converting to liveSource from a dSource

To get a live feed to the source database data through the Delphix Engine, you must first link the database to the Delphix Engine to create a dSource. You can then convert the dSource into a liveSource by following the steps outlined below:

1. In the left-hand panel, click the **dSource**.
2. From the Actions menu (...) select **Convert to liveSource**, as highlighted below. This launches the **Convert to liveSource** wizard.



Convert to liveSource database wizard

1. In the Overview tab select **Next**.
2. Enter a **DB Unique Name** for the liveSource.
3. Enter a **Database SID** for the liveSource.
4. Click **Next**.

> The liveSource database name must be the same as the database name of the primary database; therefore, this value is read-only.

Convert to liveSource, environment tab

Select the environment on which the liveSource will be created:

1. Select an **Environment User** for the liveSource instance.
2. Enter the **Mount Point** for the liveSource instance.
3. Select **Listeners** as needed. If you enable **Auto Select Listeners**, the Delphix Engine will pick the first available listener from the environment.
4. Click **Next**.

Convert to liveSource, configuration template tab

1. Select **VDB configuration templates** for the liveSource.
2. Enter additional **DB configuration parameters** for the liveSource.
3. Click **Next**.

Convert to liveSource, data management tab

1. Enter the **data age warning threshold** for the liveSource. If the data in liveSource lags behind the source database by more than this threshold, the Delphix Engine will raise a fault and notify you.
2. Click **Next**.

Convert to liveSource, hooks tab

1. Enter the **operations** to be performed on initial conversion. These operations are performed after the Delphix Engine has created the standby database for the liveSource.
2. Click **Next**.

> These operations will also be performed when resyncing a liveSource.

Convert to liveSource, summary tab

1. Review the configuration summary.
2. Click **Convert** to begin the conversion.

9.7.9.8.8.2 Setting up log transport between a dSource or primary database and a liveSource or standby database

After adding a liveSource instance, you must configure the log transport between the dSource or primary database and the liveSource or standby database. For details on configuring a standby database, refer to the Oracle Data Guard Concepts and Administration guide.

At source/primary database

1. Configure the `LOG_ARCHIVE_CONFIG` parameter to enable the sending of redo logs to remote destinations and the receipt of remote redo logs (the liveSource instance). For example: `alter system set log_archive_config='DG_CONFIG=(sourcedb,liveSource)' scope=both;`

2. Configure the `LOG_ARCHIVE_DEST_n` parameter to point the redo logs to the liveSource instance. For example: `alter system set log_archive_dest_2='SERVICE="(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=liveSource.dcenter.delphix.com)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=liveSource)(SERVER=DEDICATED)))" ASYNC VALID_FOR=(ONLINE_LOGFILE,PRIMARY_ROLE) DB_UNIQUE_NAME=liveSource' scope=both;`
3. Create a **passwd file** for the liveSource into the target site.
4. Configure the corresponding `LOG_ARCHIVE_DEST_STATE_n` parameter to identify whether the log transport is enabled. For example: `alter system set log_archive_dest_state_2='ENABLE' scope=both;`
5. Configure the STANDBY_FILE_MANAGEMENT parameter to enable automatic standby file management. For example: `alter system set standby_file_management='AUTO' scope=both;`

RAC instance as standalone environment

For RAC instances that are discovered as Standalone type environments, please apply the following steps instead of the steps above.

1. `alter system set log_archive_config='DG_CONFIG=(sourcedb,liveSource)' sid='' scope=both;`
2. `alter system set log_archive_dest_2='SERVICE=liveSource ASYNC VALID_FOR=(ONLINE_LOGFILE,PRIMARY_ROLE) DB_UNIQUE_NAME=liveSource' sid='' scope=both;`
3. Set up **tnsnames.ora** in both source and target sites.
4. Make sure to have the same password file in all RAC instances and target site.
5. `alter system set log_archive_dest_state_2='ENABLE' sid='' scope=both;`
6. `alter system set standby_file_management='AUTO' sid='' scope=both;`

> The Delphix DB user must be updated to include grants for select on v$managed_standby and v$dataguard_stats in order to access MRP information and DataGuard statistics:

```
grant select on v_$managed_standby to <delphix DB username>;
create synonym <delphix DB username>.v$managed_standby for v_$managed_standby;
```

```
grant select on v_$dataguard_stats to <delphix DB username>;
create synonym <delphix DB username>.v$dataguard_stats for v_$dataguard_stats;
```

At the staging environment where the liveSource standby database environment is running

1. Configure the `FAL_SERVER` parameter to point to the primary database for proper fetch archive log function. For example:

   `ALTER system SET fal_server='service="(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=sourcedb.dcenter.delphix.com)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=sourcedb)(SERVER=DEDICATED)))"';`
2. If not already created, configure a password for Data Guard.

9.7.9.8.8.3 Removing a liveSource

1. In the **Datasets** panel, click the **liveSource**.
2. Click the **Configuration** tab.
3. From the Actions menu (...) select **Convert to dSource.**
4. Click **Convert to dSource** .

9.7.9.8.8.4 Taking a snapshot on a liveSource

To take a snapshot of a liveSource:

1. In the **Datasets** panel, select the **liveSource**.
2. In the upper right-hand corner, click the **camera** icon.
   liveSource snapshots are instantaneous, Quick Provision snapshots. They do not require an RMAN backup of the source database

9.7.9.8.8.5 Provisioning from a liveSource TimeFlow

Provisioning from a liveSource Timeflow is the same process as provisioning from a snapshot for dSource Timeflow. The only difference is that you will select a liveSource and a liveSource snapshot.

9.7.9.8.8.6 Enabling, disabling, and detaching a liveSource

A liveSource is **enabled** the same way as a regular dSource.

1. Login to the Delphix Management application as **admin** or another user with administrative privileges.
2. Click **Manage**.
3. Select **Datasets**.

4. Click the **liveSource** you want to enable.
5. From the Actions menu (...) select **Enable.**
6. Click **Enable** to confirm.

> When you enable the liveSource, the Delphix Engine will recreate the standby database on the staging environment.

A liveSource is **disabled** the same way as a regular dSource. Disabling a liveSource will stop further operations on the Delphix Engine related to the liveSource.

1. Login to the Delphix Management application as **admin** or another user with administrative privileges.
2. Click **Manage**.
3. Select **Datasets**.
4. Select the **liveSource** you want to disable.
5. In the upper right-hand corner, from the **Actions** menu (...) select **Disable.**
6. In the Disable dialog select **Disable**.

When you are ready to enable the liveSource again, from the Actions menu (...) select **Enable**, and the liveSource will continue to function as it did previously.

> Disabling a liveSource shuts down the standby database that Delphix manages on the staging environment.

You can detach a liveSource in the same way as detaching a regular dSource (see page 1505). Detaching a liveSource will implicitly convert the liveSource into a regular dSource. After a dSource is re-attached, you can convert it back to a liveSource.

9.7.9.8.8.7 Resyncing a liveSource and applying the resync

Resync is a way to refresh the liveSource to the current point in the linked source. Resync is a multi-phase operation comprised of the following:

9.7.9.8.8.8 Perform resync

1. Click **Manage**.
2. Select **Datasets**.
3. Select your liveSource.

4. From the Actions menu (...) select **Start liveSource Resync**. The liveSource can stay up while the Resync is in progress.

9.7.9.8.8.9 Discarding resync data

Prerequisites

- Resync is started and ready to apply

After Resync has finished, you can choose to not apply but rather discard the data that was brought over from the source database as part of Resync.

Procedure

To discard the data:

1. Click **Manage**.
2. Select **Datasets**.
3. Select your liveSource.
4. From the Actions menu (...) select **Discard liveSource Resync.**

9.7.9.8.8.10 Applying resync data

Prerequisites

- Resync started and ready to apply

Procedure

1. Click **Manage**.
2. Select **Datasets**.
3. Select your liveSource..
4. From the Actions menu (...) select **Apply liveSource Resync.**
5. If the apply resync data process fails, first investigate and resolve the cause of failure, such as a full disk. Then follow the procedure to start resync.

9.7.9.8.8.11 Migrating a liveSource

1. Click **Manage**.
2. Select **Datasets**.

3. Select your liveSource.
4. From the Actions menu (...) select **Disable**.
5. From the Actions menu (...) select **Migrate**.
6. Update the environment, user, and repository. From the **Configuration** tab select the **Source** sub-tab and click the Pencil (edit) icon next to the Database.
7. Enable the **dSource**.

After the liveSource is migrated to a different staging environment, you must ensure that the log transport between the source database and the liveSource instance on the new staging environment is set up correctly.

9.7.9.8.9 Oracle RAC dSource node addition or deletion

9.7.9.8.9.1 Managing Oracle RAC node changes in the Delphix Continuous Data Engine

The addition and removal of cluster nodes from an Oracle cluster must be considered where Delphix Continuous Data Engine has dSources attached to RAC databases running from the same cluster environment or RAC VDB's provisioned out to those cluster nodes. Each scenario needs to be considered in its own right as dSources and VDBs require specific changes be made to accommodate the removal of a cluster node.

The changes required by Delphix Continuous Data Engine and made within Oracle that need to be considered when the Oracle cluster node configuration alters and the process for removing or adding an Oracle cluster node is best demonstrated through an example.

The oracle cluster is comprised of 2 cluster nodes as seen from RAC nodes.

```
[oracle@rac1338dd541 ~]$ su - grid
Password:
Last login: Thu Nov 30 17:15:36 AEDT 2023 on pts/1

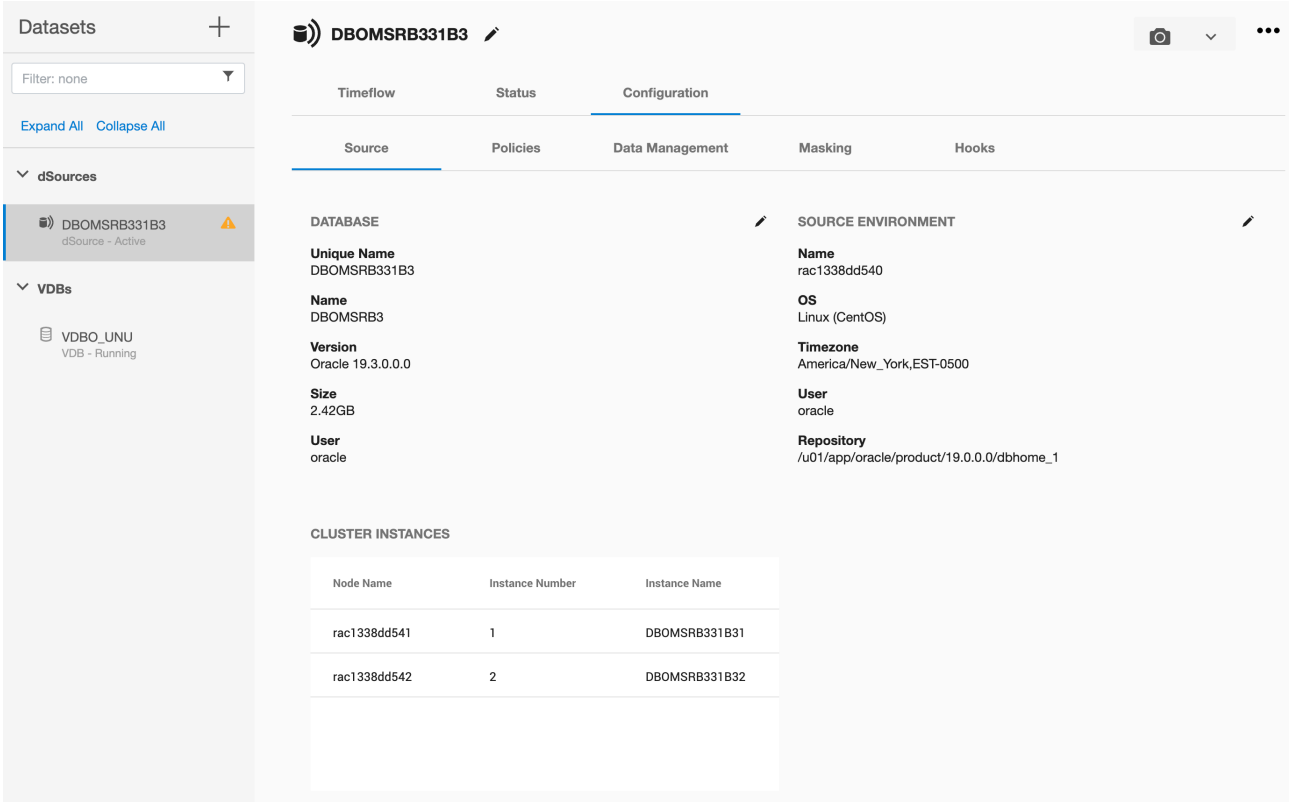
[grid@rac1338dd541 ~]$ . oraenv
ORACLE_SID = [grid] ? +ASM1
The Oracle base has been set to /u01/app/grid

[grid@rac1338dd541 ~]$  olsnodes -s -t
rac1338dd541 Active  Unpinned
rac1338dd542 Active  Unpinned
```

Node 2 in the cluster rac1338dd542 is the node targeted for removal from the Oracle cluster.

9.7.9.8.9.2 Delphix Continuous Data Engine dSources and removing Oracle cluster nodes

In this example, Delphix Continuous Data Engine has 2 Nodes, Oracle RAC database is ingested as a dSource. The sample database name is DBOMSRB3 comprised of 2 RAC instances DBOMSRB331B31 and DBOMSRB331B32. The instance impacted by the removal of node rac1338dd542 is DBOMSRB331B32. The dSource configuration appears as follows:



The cluster node, its associated listeners and instances must be deleted from the cluster through Oracle RAC RDBMS and Clusterware software using the processes described in the Oracle documentation.

This begins with the removal of the RAC instance using the process detailed in the Oracle RAC documentation. This example is operating against an Oracle 19.0 RAC and Cluster and the documentation link for this specific release is the following:

9.7.9.8.9.3 *Removing RAC instances and homes*

https://docs.oracle.com/database/121/RACAD/GUID-CE995C7B-B420-4A4E-B4EA-E7A972277588.htm#RACAD7903

Removal of the RAC instance is typically executed through the database configuration assistant dbca and using the UI provided through dbca the instance DBOMSRB331B32 is removed from node rac1338dd542. After the instance has been deleted using dbca, Oracles srvctl utility will show one instance only (DBOMSRB331B31) running from node rac1338dd541.

```
[oracle@rac1338dd541 ~]$ srvctl config database -d DBOMSRB331B3
Database unique name: DBOMSRB331B3
```

```
Database name: DBOMSRB3
Oracle home: /u01/app/oracle/product/19.0.0.0/dbhome_1
Oracle user: oracle
Spfile: +DATA/DBOMSRB331B3/PARAMETERFILE/spfile.304.1150593529
Password file: +DATA/DBOMSRB331B3/PASSWORD/pwddbomsrb331b3.292.1150592865
Domain:
Start options: open
Stop options: immediate
Database role: PRIMARY
Management policy: AUTOMATIC
Server pools:
Disk Groups: REDO,DATA
Mount point paths:
Services:
Type: RAC
Start concurrency:
Stop concurrency:
OSDBA group: dba
OSOPER group: oper
Database instances: DBOMSRB331B31
Configured nodes: rac1338dd541
CSS critical: no
CPU count: 0
Memory target: 0
Maximum memory: 0
Default network number for database services:
Database is administrator managed
```

Following the removal of the instance, the inventory on the node to be removed is updated to reflect the removal of the Oracle Home that was running this dSource instance on the node. This too must be performed according to the Oracle documentation. Oracles utility runInstaller is used for Oracle inventory maintenance. This must be performed using the operating system owner of the RDBMS Home (in this case oracle)

With the RAC instance and RAC Home removed the node itself can be removed using the process detailed in Oracles Clusterware Administration documentation,

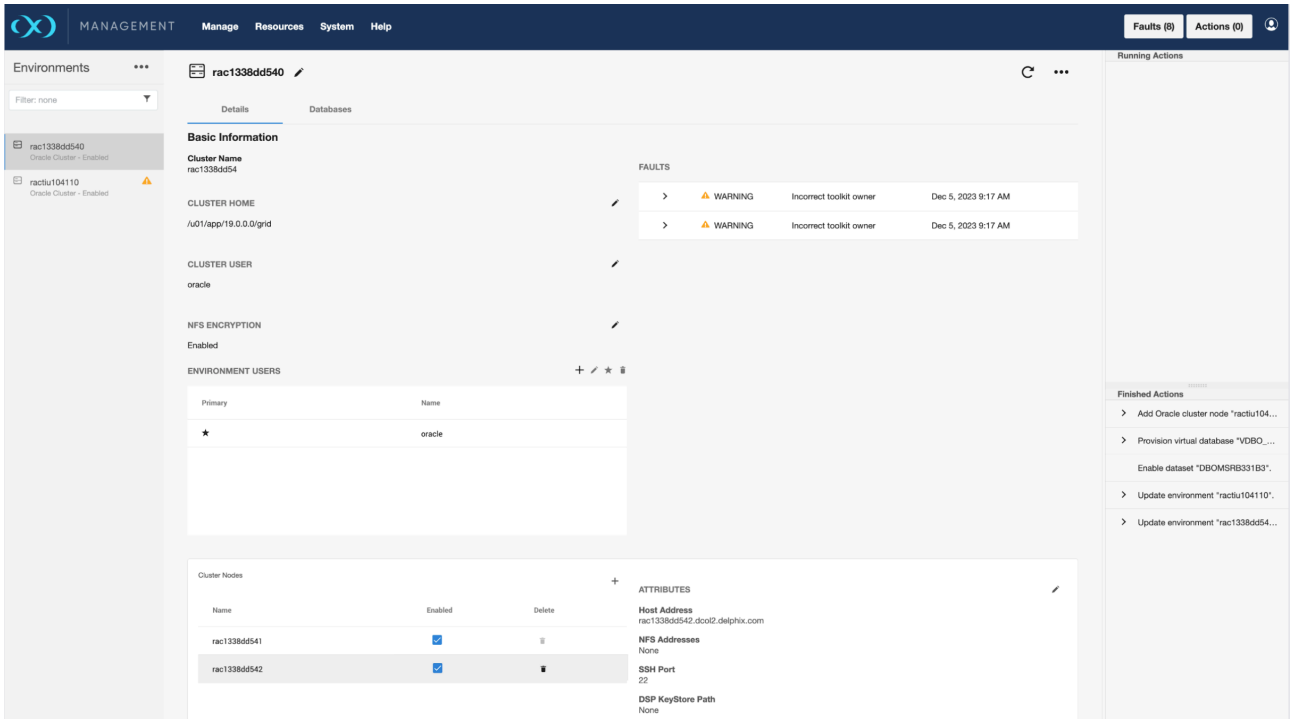
9.7.9.8.9.4 *Removing Oracle cluster nodes*

https://docs.oracle.com/database/121/CWADD/GUID-8ADA9667-EC27-4EF9-9F34-C8F65A757F2A.htm#CWADD90992

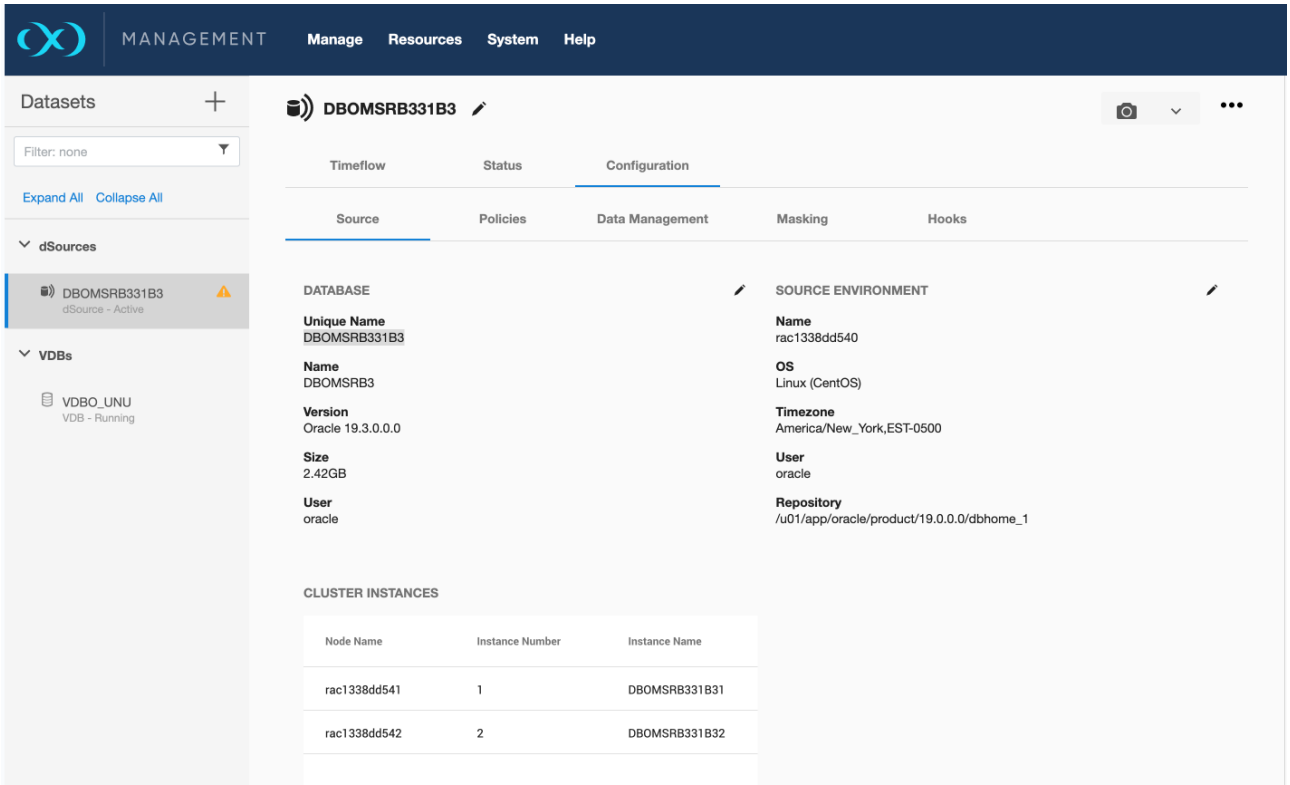
This must be performed as the Oracle Clusterware home operating system owner (in this case grid).

Removal of the node from within Oracles RDBMS and Clusterware software does not alter the fact that Delphix still sees the Oracle Cluster and RAC configuration as it was the last time an environment refresh was performed against the cluster.

The Environment in Delphix shows both cluster nodes are still present along with any node listeners that may have been discovered running on that node.



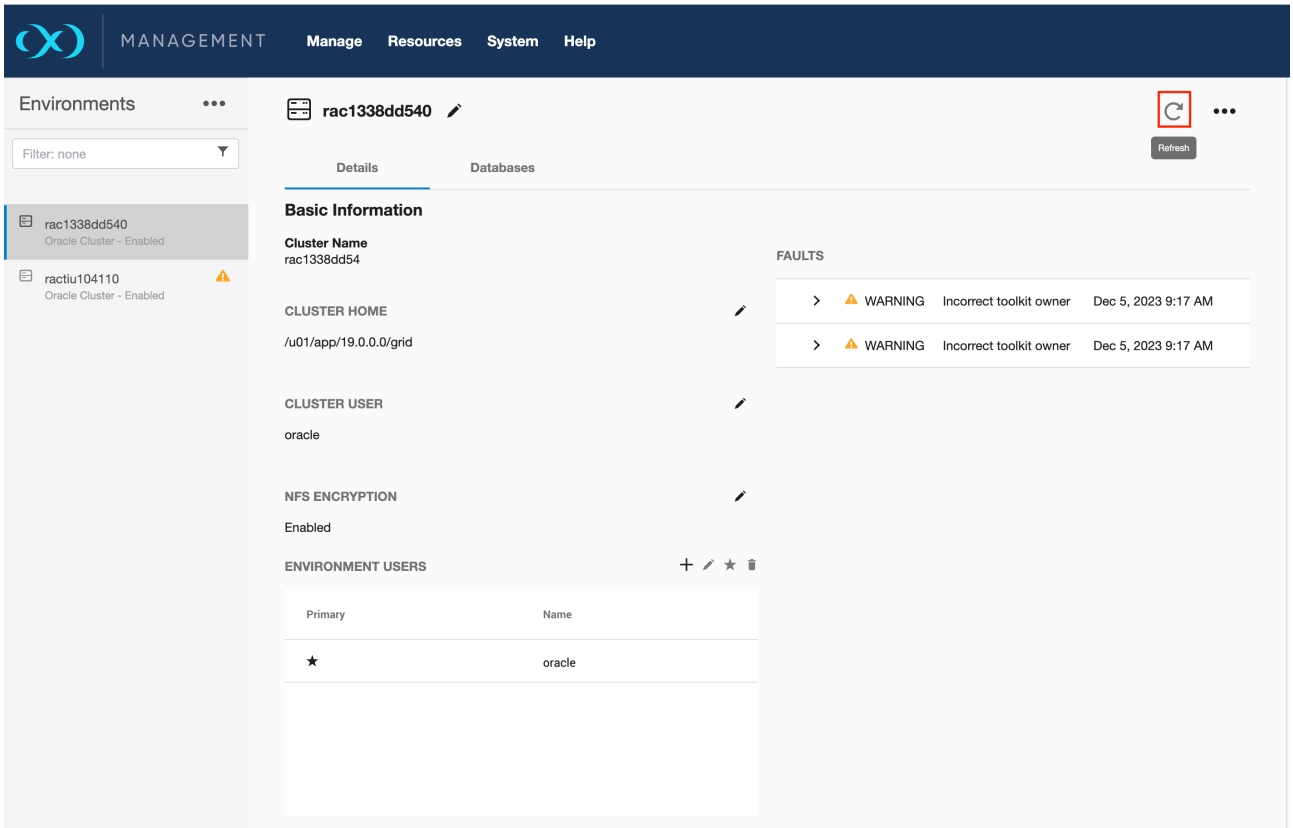
The dSource shows the node and both instances still present this needs to be cleaned up from a Delphix perspective.



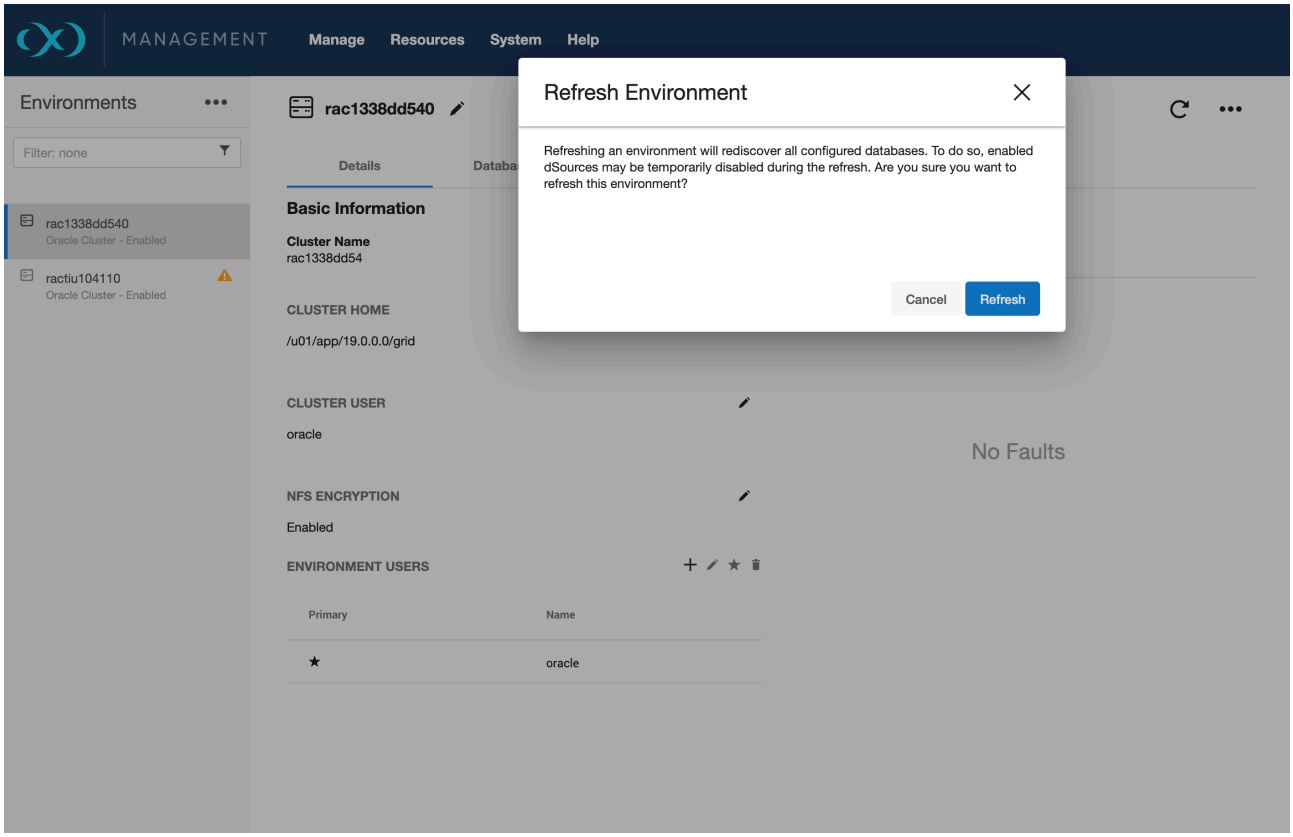
9.7.9.8.9.5 Aligning Delphix Continuous Data Engine with the new Oracle cluster configuration

In order for Delphix Continuous Data Engine to pick up on the latest changes to the environment (rac1338dd540) in this case, the removal of an entire node from the Oracle Cluster and an environment refresh must be performed.

This is performed through the environments management panel in the Delphix Management application and the refresh icon present on the top right-hand side of the panel (highlighted in red)



Clicking on the icon presents a prompt indicating that an environment refresh is about to be initiated.

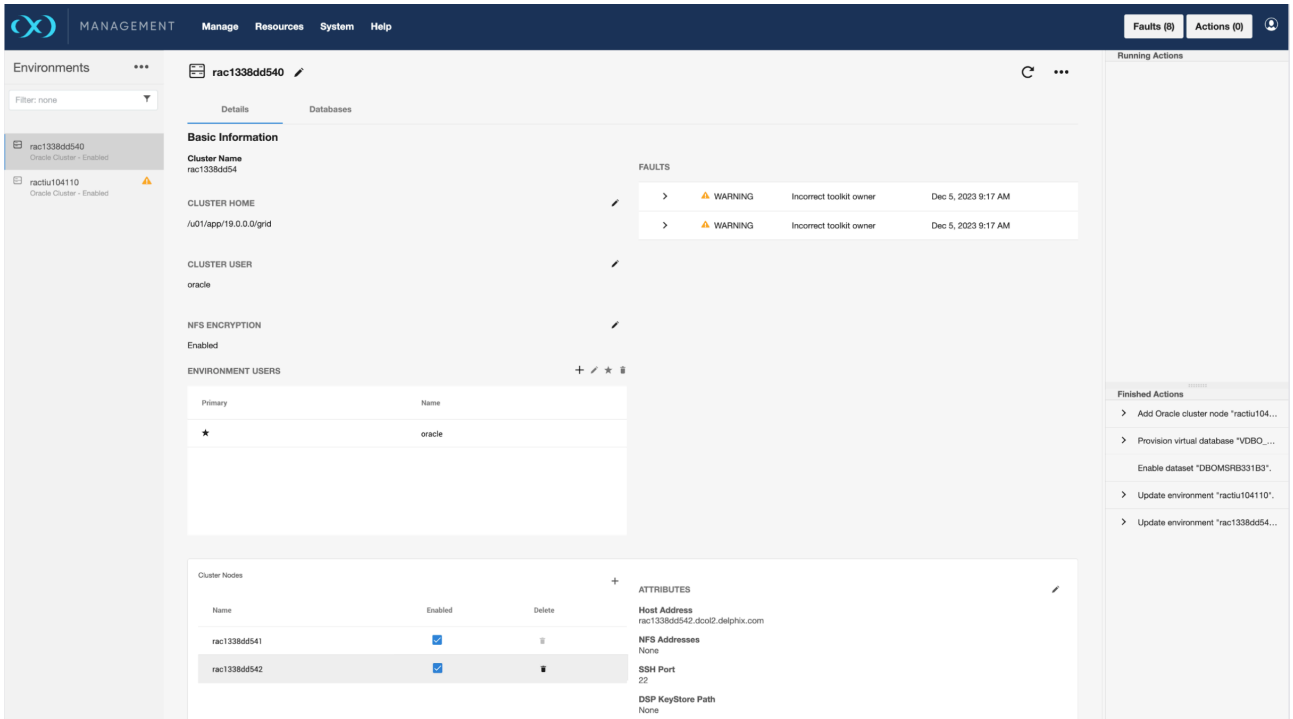


Click on refresh to kick off the environment refresh action.

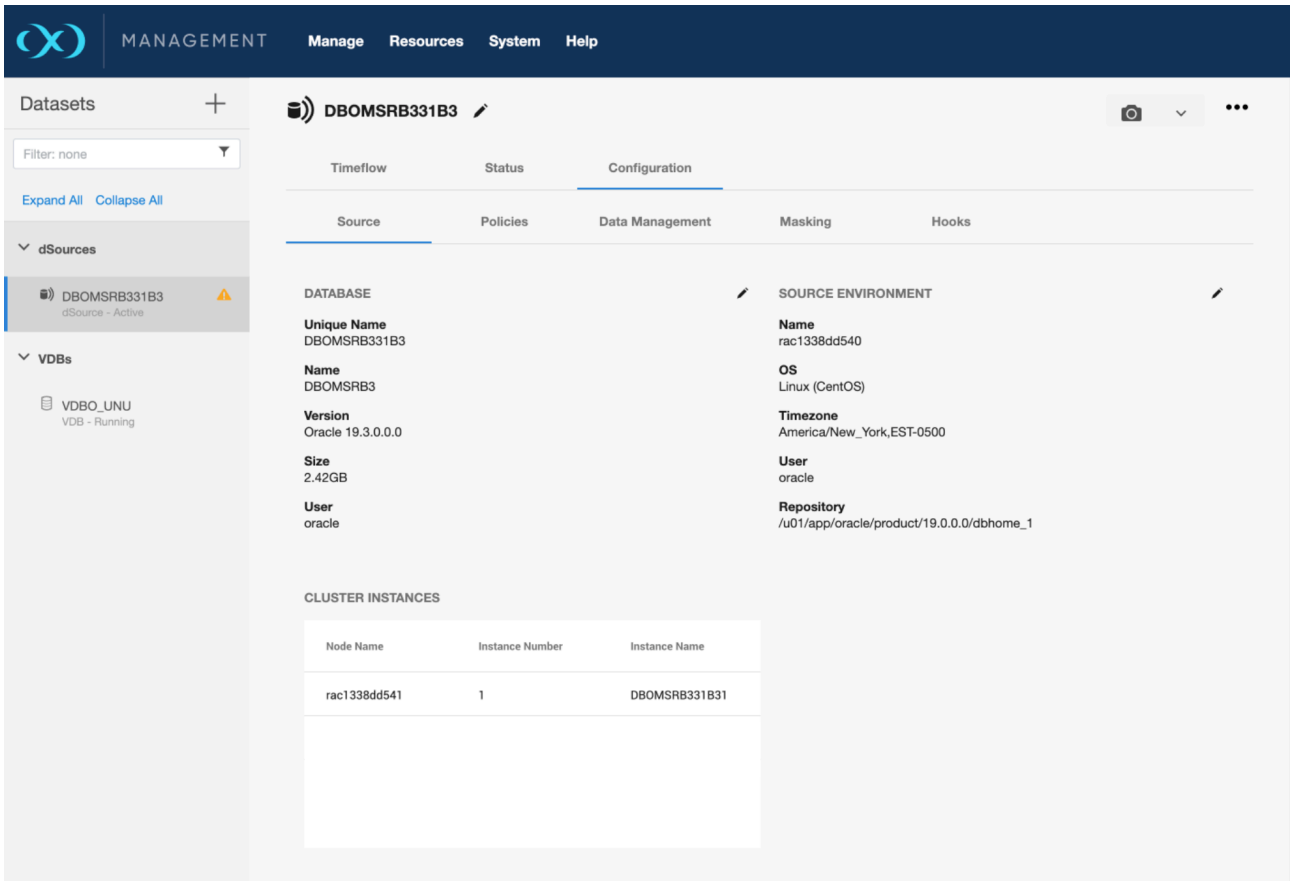
The environment refresh will force Delphix to re-examine the Oracle Cluster.

This will result in the environment and the dSource being updated based on the cluster's current node, instance and listener configuration where only one node rac1338dd541 is present.

After the node removal, the cluster environment in Delphix will look as follows and only rac1338dd541 will appear in the cluster node list.

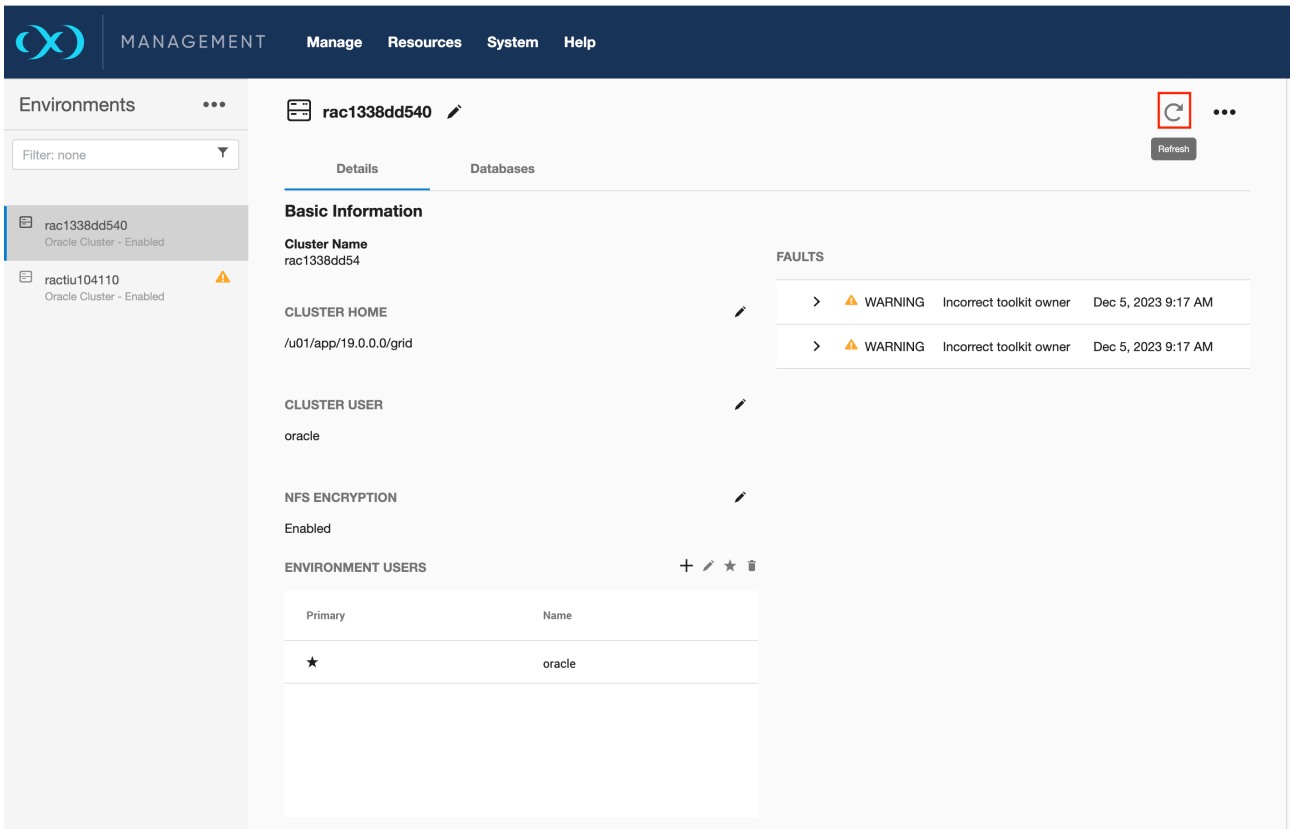


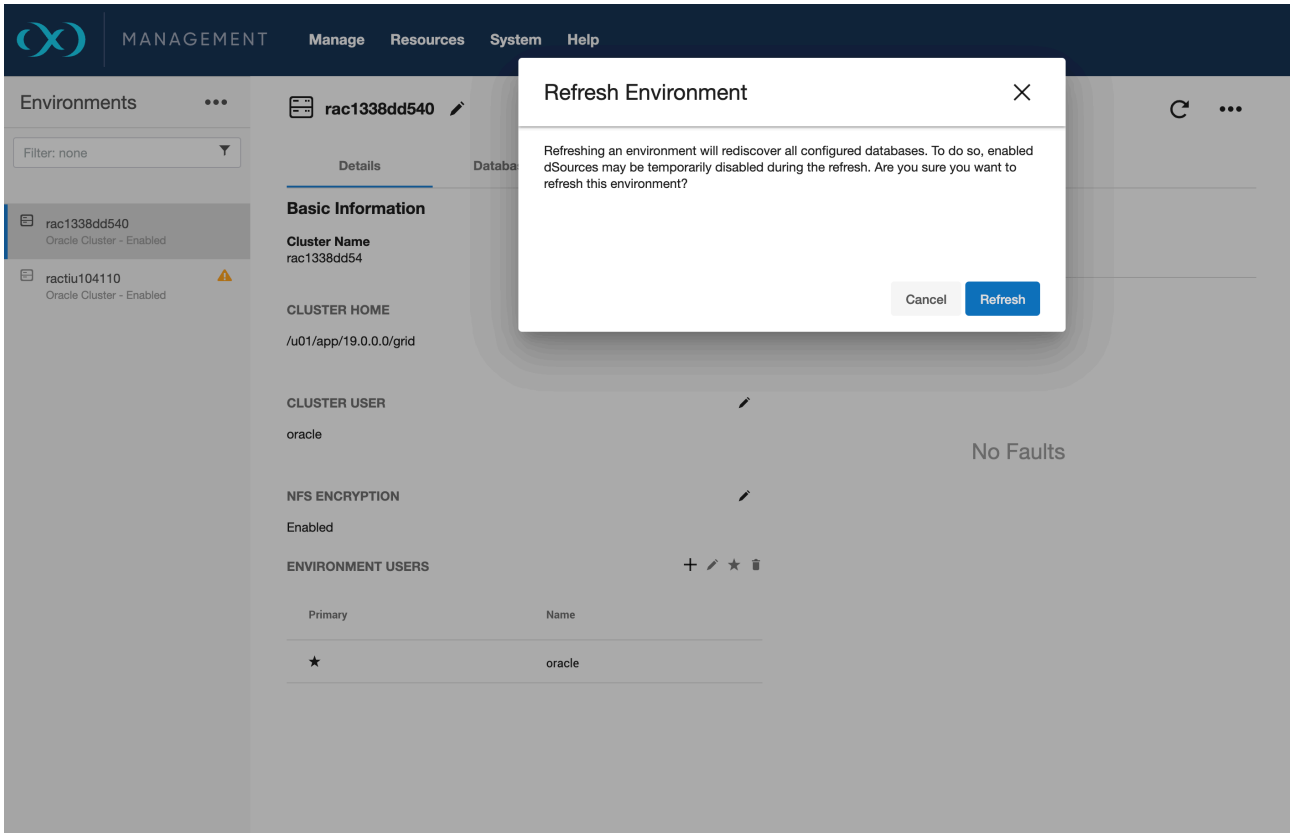
The dSource will now comprise only one instance DBOMSRB331B31 is running from node rac1338dd541.



9.7.9.8.9.6 Adding Oracle cluster nodes

You can add a node through the **Environments** management panel in the Delphix Management application and the **Refresh** icon present on the top right-hand side of the panel. Clicking on the icon presents a prompt indicating that an environment refresh is about to be initiated.



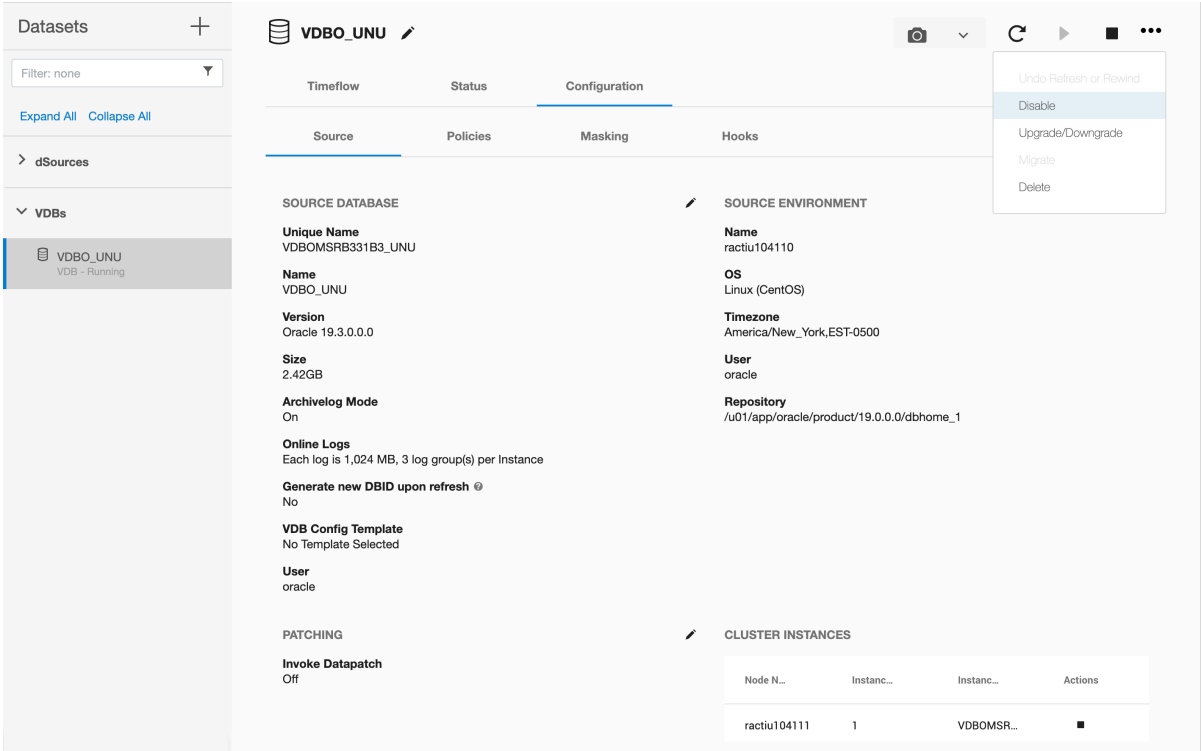


Click on the **Refresh** option to kick off the environment refresh action. The environment refresh will force Delphix to re-examine the Oracle Cluster. This will result in the environment and the dSource being updated based on the cluster's current node, instance, and listener configuration.

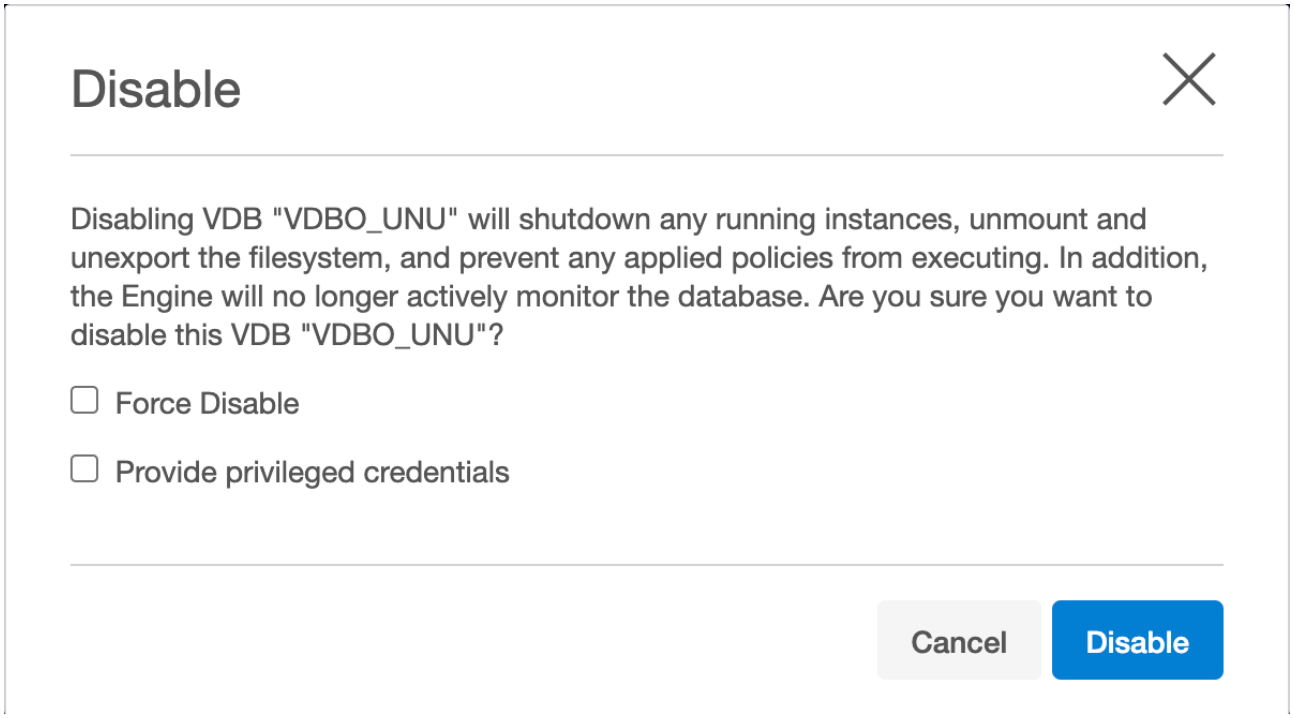
9.7.9.8.9.7 Delphix configuration steps to assign a VDB to a new cluster instance

1. Login to the Delphix Management application.
2. In the Oracle RDBMS, add the required number of nodes using the process detailed in the Oracle RAC documentation[448]
3. In the Delphix Management Engine, select the VDB you want to edit.
4. From the right-side pane select the **Configuration** tab.
5. Refresh the VDB to view new nodes.
6. In the upper right-hand corner, from the **Actions** menu (...) select **Disable.**

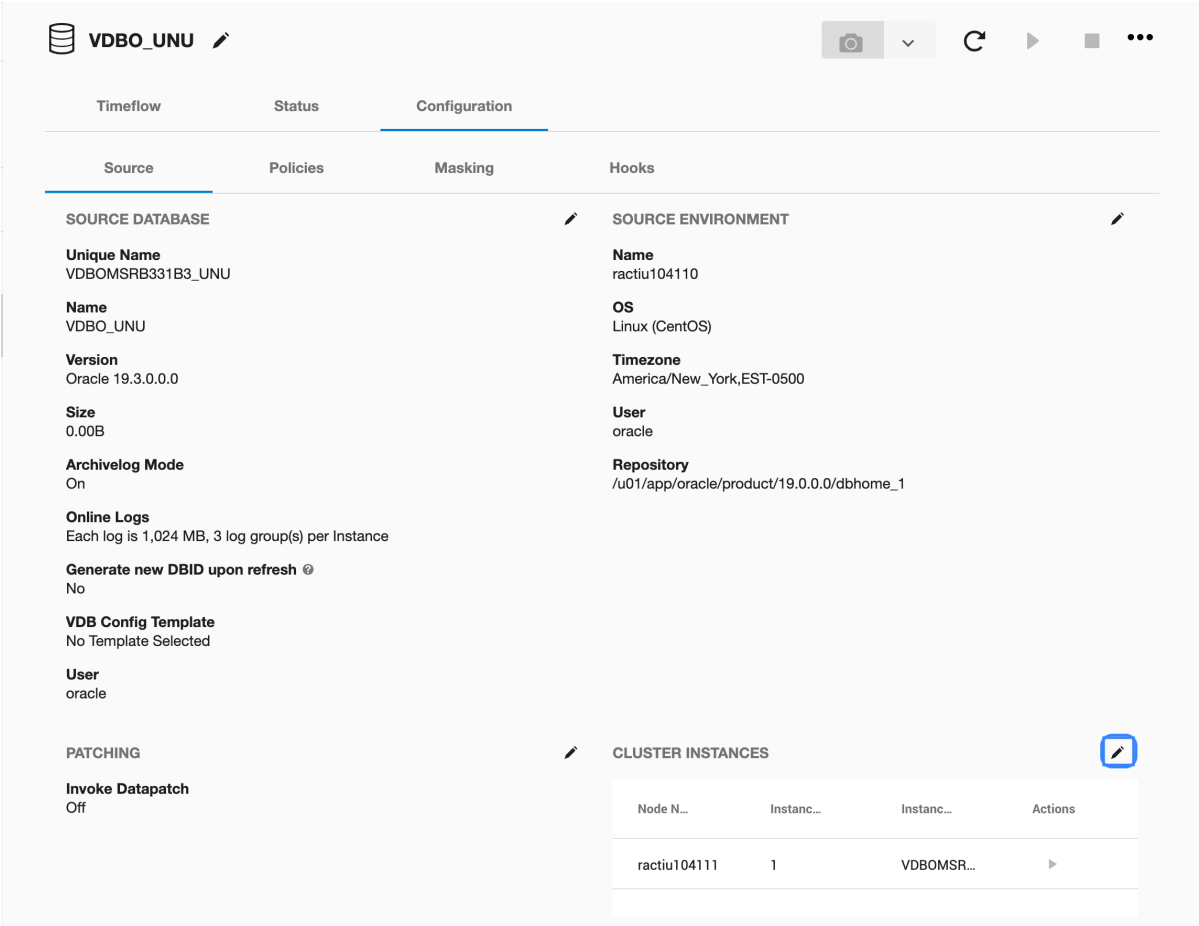
448 https://docs.oracle.com/cd/E11882_01/rac.112/e41960/adddelunix.htm#RACAD7348



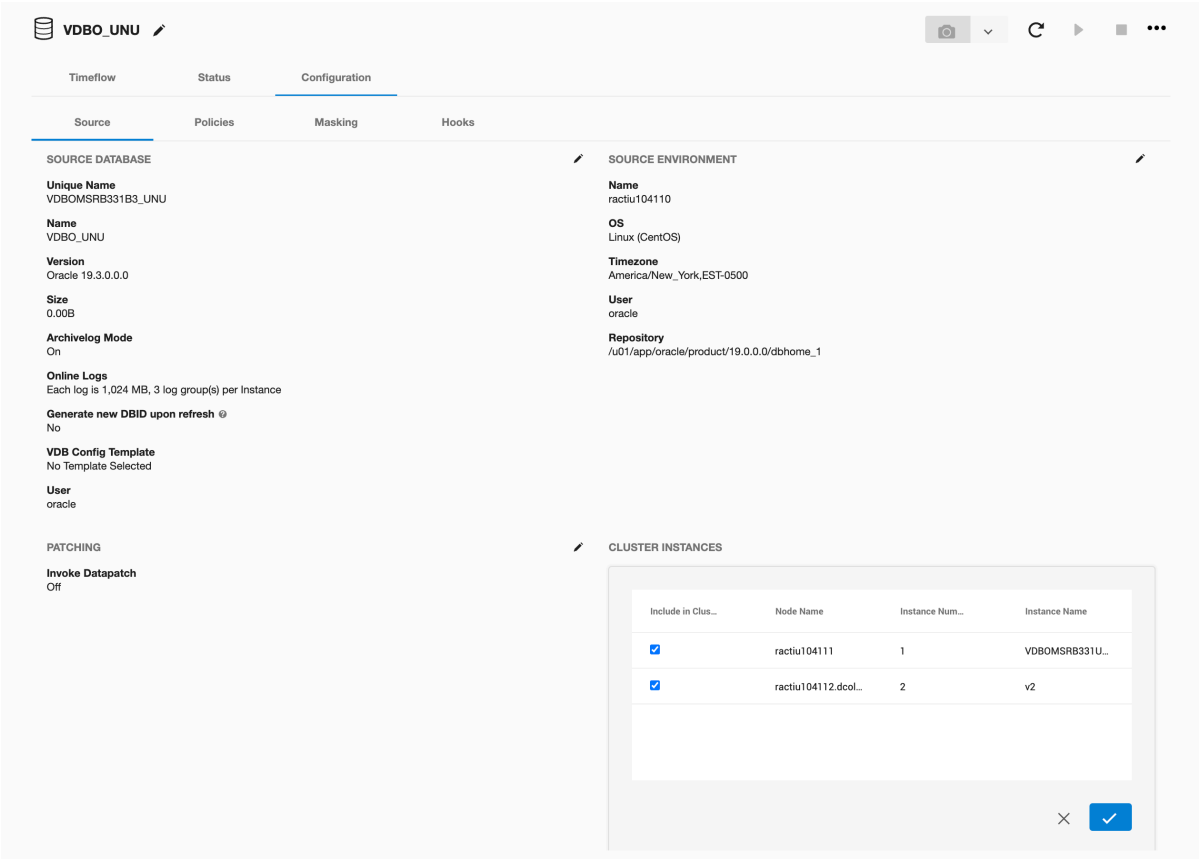
7. In the **Disable** dialog select **Disable.**



8. Click the **pencil** icon next to **Cluster Instances** to be able to add new nodes.



9. In the **Include in Cluster** column, check or uncheck the check-box to the left to add or remove a node for your RAC VDB.



10. Complete editing and then confirm using the **tick** icon.
11. From the Action menu (...), select **Enable.**

9.7.9.8.10 Using the Double Sync option for Oracle SnapSync

It is possible to request during Linking that a `doubleSync` is performed if `linkNow` is set to true. *(This is available in the GUI for linking starting in 5.0, and for manual Snapshots in 5.3.4.0, see notes below for a CLI example).*

When the *doubleSync* option is selected, two SnapSyncs in a row are run, the first one will not even bother to get the archived logs to make the first Snapshot consistent. This means that LogSync will not run during the first SnapSync, or if LogSync is not enabled, the archived logs after the backup of the data will not be retrieved. Once the first SnapSync finishes a second SnapSync is started, starting LogSync if enabled, or taking an archived log backup after the data files have been backed up.

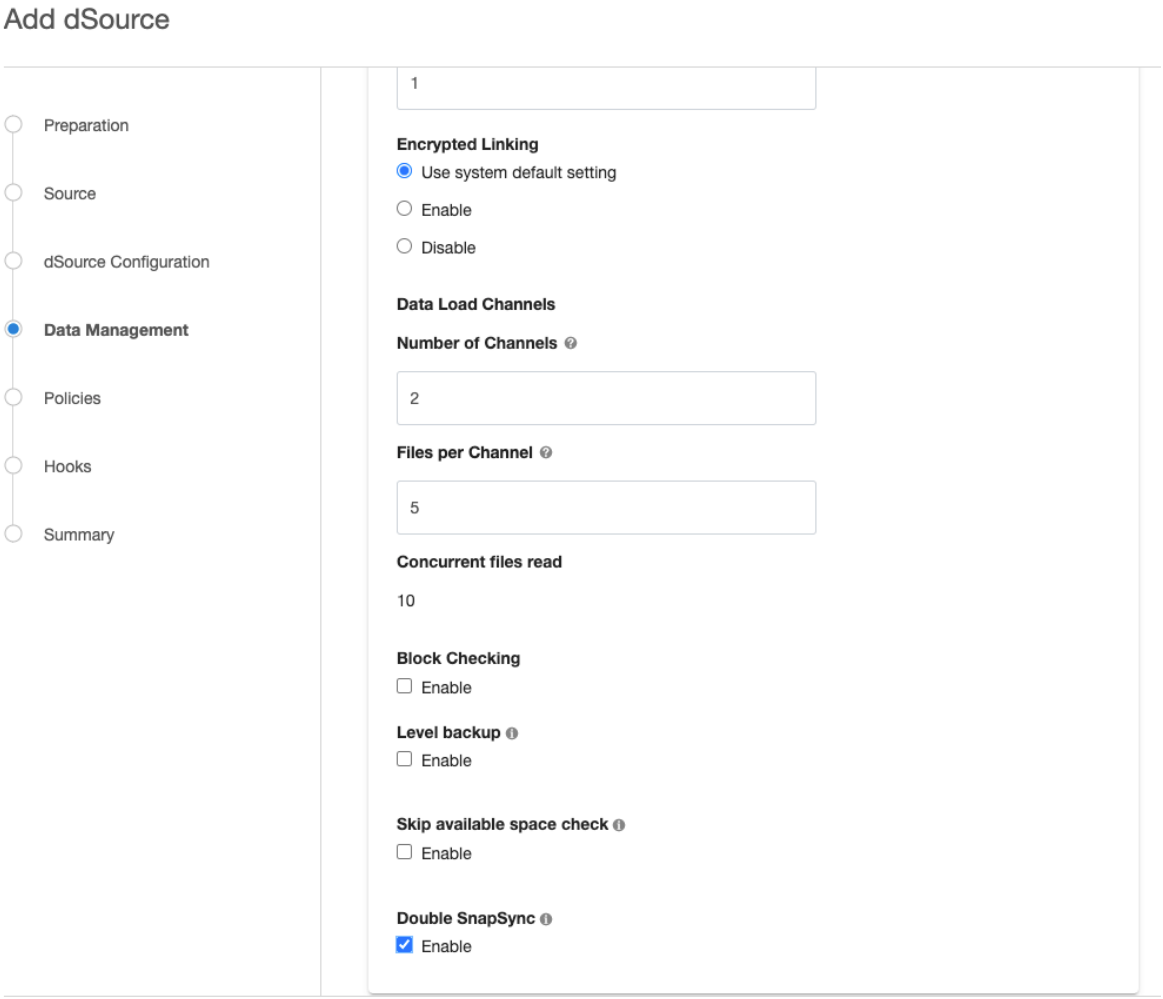
The doubleSync option is also available for manually initiated SnapSync, just like `forceFullBackup` optio n. As a `forceFullBackup` is similar to an initial load, `doubleSync` makes a great companion to `forceFullBackup`, although it is not needed to use both at the same time.

When two Double Sync snapshots are created, the older snapshot will not be provisionable. Therefore, you should not attempt time flow repair operation as the logs required to make it provisionable, were not retrieved by design. Use the new snapshot for provisioning or refreshing operation.

Note that the Double Sync option is not available for Oracle multitenant databases.

9.7.9.8.10.1 Linking with the Double Sync option via the GUI

1. Login to the **Delphix Management** application.
2. Navigate to the Environment with a Data Source you want to link. Or, from the Datasets page, click the plus icon and select **Add dSource**.
3. In the **Add dSource** wizard, select the source database with the correct environment user-specified.
4. Enter your login credentials for the source database and click **Verify Credentials**. If you are linking a mounted standby, see the topics under Linking Oracle Physical Standby Databases (see page 1466) for more information about how the Delphix Engine uses non-SYS login credentials. Click **Next**.
5. Enter a name for your dSource.
6. Select a **Database Group** for the dSource. Adding a dSource to a database group lets you set Delphix Domain user permissions for that database and its objects, such as snapshots. See the topics under Users and Groups (see page 674) for more information.
7. In the **Data Management** page select **Show Advanced** and then select **Enable Double SnapSync**. For more information, visit Data Management Settings for Oracle Data Sources (see page 1452)



8. Assign existing policies to the new dSource. New policies can be created and associated later.
9. Enter any scripts that should be run using the Hooks page.
10. Review the dSource Configuration and Data Management information, and then click Submit.

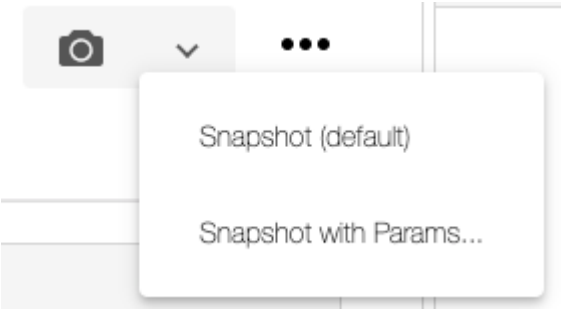
9.7.9.8.10.2 Linking with the Double Sync option via the CLI

```
ssh delphix_admin@your engine
delphix> database
delphix database> link
delphix database link*> edit source.operations
delphix database link*> edit postSync
delphix database link*> add
delphix database link*> set command=""
delphix database link*> back; back
delphix database link*> edit preSync
delphix database link*> add
```

```
delphix database link*> set command=""
delphix database link*> back; back; back
delphix database link*> set source.config=XXXX
delphix database link*> set container.name=XXXX
delphix database link*> set container.group=XXXX
delphix database link*> set container.sourcingPolicy.logsyncEnabled=true
delphix database link*> set container.sourcingPolicy.logsyncMode=ARCHIVE_REDO_MODE
delphix database link*> set linkNow=true
delphix database link*> set doubleSync=true
delphix database link*> set dbUser=XXXX
delphix database link*> set environmentUser=XXXX
delphix database link*> set dbCredentials.password=XXXX
delphix database link*> commit
```

9.7.9.8.10.3 Syncing with the Double Sync option via the GUI

1. Login to the **Delphix Management** application.
2. Navigate to the Datasets page, select the Dataset you want to SnapSync.
3. Click on the arrow next to the camera button.
4. Select **Snapshot with Params.**



5. Select **Double Sync** (and if so desired **Force Full Backup**) from the available parameters.

6. Select **Perform Snapshot.**

9.7.9.8.10.4 Syncing with the Double Sync option via CLI

```
ssh delphix_admin@yourengine
delphix> database
delphix database> select XXXX
delphix database 'XXXX'> sync
delphix database 'XXXX' sync *> set doubleSync=true
delphix database 'XXXX' sync *> commit
```

9.7.9.8.11 Linking dSources from an encrypted Oracle database

This topic describes the behavior of the Delphix Engine when linking to a dSource based on an encrypted Oracle database.

> This topic does not apply to vPDBs.

Beginning with version 10gR2, Oracle supports the encryption of permanent tablespaces using Transparent Data Encryption (TDE). You can link dSources from databases using TDE by following the basic procedure described in Linking an Oracle Data Source (see page 1438). However, in order to provision a VDB from a dSource that is linked to an encrypted database, you must copy wallet files from the physical database in the source environment to the target environment. See Provisioning a VDB from an Encrypted Oracle Database (see page 1584) for more information.

9.7.9.8.12 Detaching and re-attaching Oracle dSources

9.7.9.8.12.1 Overview

Each dSource contains metadata that associates it with the source database, as well as the data it has ingested from the source database in the form of snapshots up to that point. It is possible to detach, or unlink, a dSource from its source database. This breaks the association with the source database without affecting the data within the Delphix Engine. Detached dSources and their source databases have these properties:

- A detached dSources can still be used to provision a virtual database (VDB).
- You can re-link the source database as a different dSource.

9.7.9.8.12.2 Prerequisites

Before detaching an Oracle dSource, you must capture the following RMAN configuration:

- Level or SCN based backups

- Data load channel settings: number of channels and files per channel

A dSource can only be attached to a new data source once it has been unlinked.The above RMAN configuration will be removed once the dSource is unlinked.

When attaching an Oracle dSource to a new data source, the new data source must be the same logical database satisfying the following constraints:

- Same dbid
- Same dbname
- Same creation time
- Same resetlogs SCN
- Same resetlogs time
- Same redo stream, where a log must exist with
    - Same sequence
    - Same thread
    - Same end SCN

For Oracle dSources, this procedure can be used to initially link from a standby server that is faster or less disruptive, unlink the dSource, and then attach it to the production server for subsequent incremental SnapSync operations. When you perform the attach operation, you will need the source config name of an unlinked database.

> It is possible to link a Single Instance Oracle database, unlink from it, and link to a RAC cluster and vice versa as long as the instances are for the same database. For example, initially link a Single Instance standby database of a RAC primary database, unlink the dSource, and then link the RAC primary database itself or another RAC standby database to the same dSource.

9.7.9.8.12.3 Detaching a dSource

1. Login to the **Delphix Management** application as a user with **OWNER** privileges on the dSource, group, or domain.
2. Click **Manage**.
3. Select **Databases**.
4. Select the **database** you want to unlink or delete.
5. From the **Actions** menu (...) select **Unlink.** A warning message will appear.
6. Click **Unlink** to confirm.

To detach a dSource via CLI, see Detaching and Attaching an Oracle dSource via CLI (see page 1505)

> **Rebuilding source databases and using VDBs**

In situations where you want to rebuild a source database but retain the existing dSource, you will need to detach the original dSource and create a new one from the rebuilt data source.

1. Detach the dSource as described in the procedure on this page.
2. You cannot attach a dSource with the same name as a dSource that is already attached. If you intend to give the new dSource the same name as the original one, rename the detached dSource.
    a. At the top of the **Configuration** tab, next to the dSource's name, click the **Edit** (pencil) icon.
    b. After renaming the dSource, click the green **checkmark**.
3. Create the new dSource from the rebuilt database.

You will now be able to provision VDBs from both the detached dSource and the newly created one, but the detached dSource will only represent the state of the source database prior to being detached.

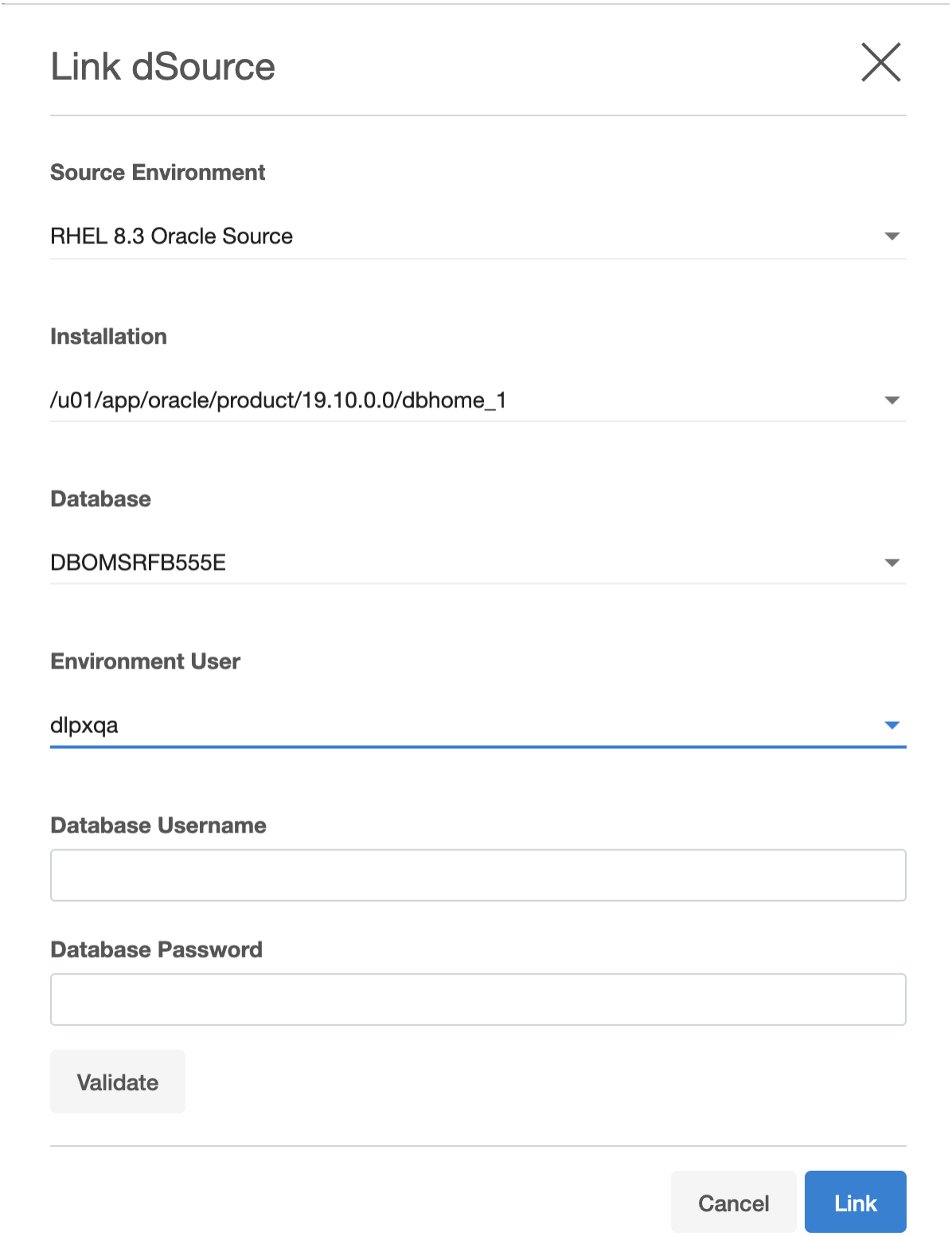
9.7.9.8.12.4 Attaching a previously detached dSource

You can only re-attach databases that represent the same physical database.

Via GUI

To attach a dSource back into the Delphix Engine via GUI, from the **Delphix Management** application:

1. Select **Manage > Datasets**.
2. Select your dSource.
3. From the **Actions** menu (...) select **Link dSource**.
4. In the Link dSource dialog, fill in the information corresponding to the new dSource as well as its new environment.


Link dSource
Source Environment
RHEL 8.3 Oracle Source
Installation
/u01/app/oracle/product/19.10.0.0/dbhome_1
Database
DBOMSRFB555E
Environment User
dlpxqa
Database Username
Database Password
Validate
Cancel
Link

5. Click **Link**.

Via CLI

To attach a dSource back into the Delphix Engine via CLI:

1. Login to the **Delphix CLI** as delphix_admin or a user with OWNER privileges on the dSource, group, or domain.
2. Select the dSource by name using `database select`.
3. Run the `attachSource` command.
4. Set the source config you want to attach to, using `set attachData.config=`. Source configs are named by their database unique name.
5. Set any other source configuration operations as you would for a normal link operation.
6. Run the `commit` command.

Backup mode for sttaching Oracle dSources
For Oracle dSources, the SnapSync backup option should be set to **SCN Backup** mode. **Level Backup** mode is based on information stored in the database control file. If the control file of the newly attached database does not contain information about the previous backups, an initial backup will be created. In addition, Block Change Tracking will not be in sync, and the next SnapSync will need to read the entire database to determine which blocks have changed. See Advanced Data Management Settings for Oracle dSources (see page 2005) for more information about Backup Mode.

9.7.9.8.13 Provisioning from a replicated Oracle dSource

This topic describes how to provision from a replicated dSource or virtual database (VDB).

The process for provisioning from replicated objects is the same as the typical VDB provisioning process, except that first you need to select the replica namespace containing the replicated object.

9.7.9.8.13.1 Prerequisites

- You must have a dSource or VDB that has been replicated from one Delphix Engine to another, as described in Replication Overview (see page 2368)
- The Delphix Engine containing the replicated dSource or VDB must have a compatible target environment that it can use to provision a VDB from the replicated dSource or VDB.

9.7.9.8.13.2 Procedure

1. On the Delphix Engine containing the replicated dSource or VDB, login to the **Delphix Management** application.
2. In the top menu bar, click **Manage**.
3. Select **Datasets**.
4. From the list of replica namespaces, select the **replica namespace** that contains the dSource or VDB from which you want to provision.
5. The provisioning process is now identical to the process for provisioning standard objects.

9.7.9.8.13.3 Post-requisites

Once the provisioning job has started, the user interface will automatically display the new VDB in the live system.

9.7.9.8.14 Detaching and re-attaching a PDB dSource from a data guard site

If you want to move the PDB dSource from one Data Guard site to another, then you must detach the PDB from the current environment and attach it to the new environment.

Perform the following procedure to attach or detach a PDB dSource from a Data Guard site.

1. Detach PDB from CDB (a).
   a. Log into the **Delphix Management** application.
   b. Select **Manage > Datasets.**
   c. Select dSource.
   d. From the **Actions** menu (...), select **Unlink dSource**.
   e. Click **Unlink** to confirm.
2. If you are going to attach the PDB dSource from a different Data Guard site (Primary or Standby site), perform a fresh refresh of the existing environment to discover that host environment into Delphix. From the **Delphix Management** application:
   a. Go to **Manage > Environments**.
   b. If it is already a discovered environment, then from the **Actions(...)** menu, select **Refresh All**. Otherwise, select **Add Environment** to add the environment manually. For more information on adding an environment, see Adding an Oracle Single Instance or RAC Environment (see page 1417). The new database configuration will appear in the Delphix Engine.

> Do not add the discovered PDB as a dSource.

3. Detach the existing PDB dSource.

- Select the dSource.
- From the **Actions** menu (...), select **Unlink dSource**.
- Click **Unlink** to confirm.
- Using the Delphix CLI, detach the current CDB datasets associated with the same PDB dSource.

```
dlpx6010 database> select cdb19c1
dlpx6010 database 'cdb19c1'> detachSource
dlpx6010 database 'cdb19c1' detachSource *> ls
Properties
    type: DetachSourceParameters
    source: (required)
dlpx6010 database 'cdb19c1' detachSource *> set source=cdb19c1
dlpx6010 database 'cdb19c1' detachSource *> commit
    Dispatched job JOB-16
    DB_DETACH_SOURCE job started for "SugoPronto/cdb19c1".
    DB_DETACH_SOURCE job for "SugoPronto/cdb19c1" completed successfully.
```

- Using the Delphix CLI, attach the new CDB configuration from the new environment/Physical Standby site.

```
dlpx6010 *> database
dlpx6010 database *> select cdb19c1
dlpx6010 database 'cdb19c1'*> attachSource
dlpx6010 database 'cdb19c1' attachSource *> set
attachData.type=OracleAttachData
dlpx6010 database 'cdb19c1' attachSource *> set attachData.config=cdb19c1stb
dlpx6010 database 'cdb19c1' attachSource *> set
attachData.environmentUser=OEL7SITDE2/delphix
dlpx6010 database 'cdb19c1' attachSource *> commit;
    cdb19c1
    Dispatched job JOB-18
    DB_ATTACH_SOURCE job started for "SugoPronto/cdb19c1".
    Starting validation of attach parameters for database "cdb19c1".
    Validation finished for database "cdb19c1".
    Obtaining information from source database "SugoPronto/cdb19c1".
    The dSource "cdb19c1stb" was successfully linked from source database
"SugoPronto/cdb19c1".
    DB_ATTACH_SOURCE job for "SugoPronto/cdb19c1" completed successfully.
```

- Using the Delphix Engine user interface, attach the PDB.
  a. Go to **Manage > Datasets**.
  b. Select the PDB from the datasets list.
  c. From the **Actions(...)** menu, select **Link dSource**. The Link dSource page appears.
  d. Perform the linking of the PDB from the new environment and click **Next**.

9.7.9.8.15 Working with Oracle snapshots

This section lists the steps to take a snapshot and delete the same.

Taking a snapshot creates a new snapshot entry in the Oracle dSource's Timeflow. You can use either **snapshot (Default)** or **snapshot with Parameters** option for taking the snapshot.

snapshot (Default)

Perform the following steps to take a snapshot:

1. Login to the **Delphix Management** application.
2. Click **Manage** and select **Datasets** from the dropdown list.
3. Select the dSource you want to snapshot.
4. Click the **Camera** icon. Alternatively, click the arrow next to the Camera icon and select **snapshot (default)**.
5. From the snapshot dialog box, select **Yes**.
6. Navigate to the **Timeflow** tab and click **View: All snapshots** to verify the snapshot you just created.
7. To delete the snapshot, select the snapshot you just created, and from the Actions menu (...), select **Delete snapshot**.
8. From the **Delete snapshot** dialog box, select **Delete**.
9. Navigate to the **Timeflow** tab and click **View: All snapshots** to verify the snapshot you just deleted.

snapshot with parameters

Perform the following steps to take a snapshot:

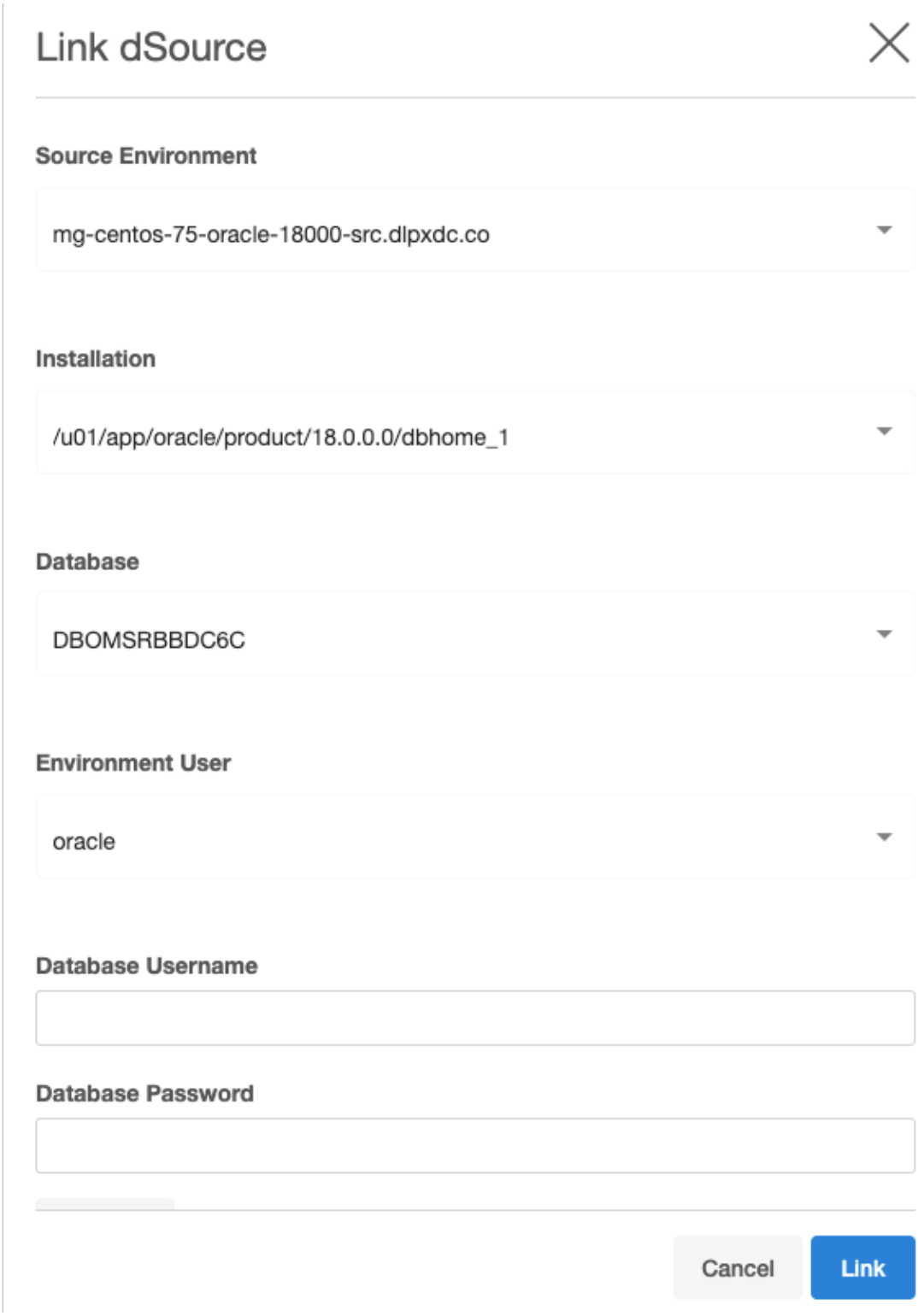
1. Login to the **Delphix Management** application.
2. Click **Manage** and select **Datasets** from the dropdown list.
3. Select the dSource you want to snapshot.
4. Click the arrow next to the Camera icon and select the **snapshot with Params...** option.
5. From the snapshot dialog box, select one of the following:
    a. **Force full backup** - If you select this option, then the Delphix Engine will perform an incremental backup by default. You must select this option only when a full backup is required. Full and Incremental backups consume the same space on the Delphix Engine.
    b. **Double sync** - Selecting this option will perform a SnapSync operation as normal. After the first SnapSync is successful, the Engine will immediately perform a second SnapSync without waiting for the Log Files required for the first SnapSync to be made consistent. This is most useful when performing the initial SnapSync (or when "Force Full Backup" is selected) on a very large database that would lead to a large number of archive logs being required to make the SnapSync consistent. Provisioning from a SnapSync that requires excessive recovery is typically time-consuming.
    c. **Do not resume** - If a failure is encountered during the initial SnapSync, the Delphix Engine can resume the SnapSync at a later date. This option will cause the engine to not resume, but rather to start the initial SnapSync over again.
6. Navigate to the **Timeflow** tab and click **View: All snapshots** to verify the snapshot you just created.
7. To delete the snapshot, select the snapshot you just created, and from the Actions menu (...), select **Delete snapshot**.

8. From the **Delete snapshot** dialog box, select **Delete**.
9. Navigate to the **Timeflow** tab and click **View: All snapshots** to verify the snapshot you just deleted.

9.7.9.8.16 Moving the PDB to a new CDB

Perform the following procedure to move the dSource PDB to a new CDB.

1. Follow Oracle steps to unplug your dSource PDB and plug it into your chosen CDB (Refer to the Oracle webpage on Plugging an Unplugged Pluggable Database). After you have plugged your PDB dSource into the new CDB, discover the host environment that has the new CDB into Delphix, if it is not already discovered. Log into the **Delphix Management** application and do the following:
    a. Select **Manage > Environments**.
    b. Select your new CDB environment, then discover the new CDB where your PDB dSource is.
2. If you have already discovered the CDB, then:
    a. Select **Manage > Environments**.
    b. Select the dSource Environment where the PDB was plugged.
    c. **Select the Refresh** icon.
3. When you are ready to attach the PDB dSource, log into the **Delphix Management**application, and do the following:
    a. Select **Manage > Environments**.
    b. Select your original PDB dSource Environment.
    c. **Select the Refresh** icon.
4. To Attach the PDB dSource back into the Delphix Engine, do the following:
    a. Select **Manage > Datasets**.
    b. Select your PDB dSource.
    c. From the **Actions** menu (...), select **Link dSource**.
    d. In the Link dSource dialog, fill in the information corresponding to the new PDB dSource as well as its new environment:


Link dSource
Source Environment
mg-centos-75-oracle-18000-src.dlpxdc.co
Installation
/u01/app/oracle/product/18.0.0.0/dbhome_1
Database
DBOMSRBBDC6C
Environment User
oracle
Database Username
Database Password
Cancel
Link

e. Click **link**.

9.7.10 Provisioning and managing virtual databases with Oracle

Each VDB has its own data management settings, found during the provisioning workflow as well as in the configuration page for that VDB. When you create a VDB, the Delphix Engine copies configuration settings from the dSource and uses them to create the VDB.

This section covers the following topics:

- Overview of provisioning Oracle virtual databases (see page 1515)
- Configuration settings for Oracle virtual databases (see page 1517)
- Provisioning an Oracle VDB (see page 1541)
- Provisioning an Oracle virtual pluggable database (vPDB) (see page 1546)
- Provisioning a TDE-enabled vPDB (see page 1549)
- Provisioning a vPDB from a non-multitenant source (see page 1576)
- Managing multiple virtual PDBs in a virtual CDB (see page 1586)
- Managing RAC virtual databases with only a subset of configured RAC nodes available (see page 1585)
- Provisioning a VDB from an encrypted Oracle database (see page 1584)
- Refreshing and rewinding a TDE-enabled vPDB (see page 1589)
- Timeflows for RAC provisioning of VDBs (see page 1602)
- Upgrading an Oracle VDB, linked CDB, or a vCDB (see page 1606)
- Customizing RAC instances after provision (see page 1609)
- Migrating an Oracle VDB (see page 1611)
- Migrating a vPDB (see page 1612)
- Migrating a TDE-enabled vPDB (see page 1613)
- Migrating a vPDB (from a linked CDB) to a higher Oracle version (linked CDB) (see page 1622)
- V2P with an Oracle VDB (see page 1624)
- Provisioning from a replicated Oracle VDB (see page 1641)
- Provisioning a TDE-enabled vPDB to a cluster target[449]
- Managing cluster instances of Oracle RAC virtual databases (see page 1645)

9.7.10.1 Overview of provisioning Oracle virtual databases

Virtual databases are a key data management concept for Delphix. In order to create or provision a virtual database, you will need a linked dSource from a source host and a compatible target environment, as described in the overview for Managing Environments and Hosts (see page 1017) and Requirements for Oracle Hosts and Databases (see page 1392)

> A Solaris x86 source host is compatible with a Linux x86 target host and vice versa.

449 https://delphixdocs.atlassian.net/wiki/pages/createpage.action?fromPageId=149490261&linkCreation=true&spaceKey=CD&title=Provisioning+a+TDE-enabled+vPDB+to+a+cluster+target

From a dSource, you can select a snapshot or point in time to create a VDB. Oracle VDBs each have their own configuration settings as described in Configuration Settings for Oracle Virtual Databases (see page 1517)

9.7.10.1.1 Procedure

1. In the Datasets panel on the left-hand side, click the group containing the dSource or VDB from which you want to provision.
2. From the **Timeflow** tab, select a snapshot or point in time to provision from.
3. Click to open the **Provision VDB wizard**, and select a compatible Target Environment for the new Oracle VDB
   - Select Provide Privileged Credentials if you want to use login credentials on the target environment that are different from those associated with the Environment User
4. In the **Advanced** section, you may customize the VDB's configuration settings, listeners, file mappings, patching, or custom environment variables. For more information, see Configuration Settings for Oracle Virtual Databases (see page 1517).
5. Select a **Snapshot Policy** for the VDB.
6. Enable **Masked Provisioning** by selecting an option on the Masking page.
7. Enter any operations that should be run in the Hooks page.
8. Review the VDB Configuration and Summary, and then click **Submit**.

When provisioning starts, you can review the progress of the job by selecting the VDB and clicking on the Status tab, or by selecting System and viewing the Jobs page.

Alternatively, you could see this in the Actions Sidebar. When provisioning is complete, the VDB will be included in the group you designated and listed in the Datasets panel.

9.7.10.1.2 Provisioning by snapshot or logSync

When provisioning by snapshot, you can provision to the start of any particular snapshot, either by time or SCN.

Snapshot Options

Snapshot options	Description
Provision by Time	You can provision to the start of any snapshot by selecting that snapshot card from the TimeFlow view or by entering a value in the time entry fields below the snapshot cards. The values you enter will snap to the beginning of the nearest snapshot.

Snapshot options	Description
Provision by SCN	To open the SCN entry field select the Actions (...) menu for your selected snapshot and select View by SCN. Here, you can type or paste in the SCN to which you want to provision. After entering a value, it will "snap" to the start of the closest appropriate snapshot.

For more granularity, you can use the logSync options to provision to any point in time, or to any SCN, within a particular snapshot.

logSync Options

logSync options	Description
Provision by Time	Select Open logSync control to view the time range within that snapshot. Select the clock icon and then select Time to the point to the time from which you want to provision. You can also enter a date and time directly.
Provision by SCN	Select View SCN to view the range of SCNs within that snapshot. You can also type or paste in the specific SCN to which you want to provision. Note that if the SCN does not exist, you will see an error when you provision.

9.7.10.2 Configuration settings for Oracle virtual databases

Each VDB has its own data management settings, found during the provisioning workflow as well as in the configuration page for that VDB. When you create a VDB, the Delphix Engine copies configuration settings from the dSource and uses them to create the VDB.

The following settings are available for Oracle VDBs:

VDB setting	Explanation
Mount base	The directory to which Delphix will provide mounted storage. This is where the contents of this VDB will be located.
VDB configure parameters	These are customizable parameters of the VDB that you can set either in the provided table or in the text field. This concept is described further below.

VDB setting	Explanation
Open database after provision	Open the VDB immediately after the provision completes. Enabling this option allows you to set the four parameters below: • Online Log Size (MB) • Number of Online Log Groups • Enable Archivelog Mode • Generate new DBID for VDB This setting is recommended to tune the Oracle database settings to match your requirements. Each of these four helps Delphix manage the Oracle database optimally. When provisioning an Oracle VDB with the "Open Database After Provision" option unchecked, the subsequent snapshot job fails because the database is not open. The UI will show that the provision failed, but this is not the case. A VDB provisioned this way will exist with the stop/start enable/disable function, but will need to be manually started on every refresh.
Online log size (MB)	Also known as the online redo log, this specifies the size of the VDB's redo log. We recommend increasing this value to 1024 MB for improved performance.
Number of online log groups	The number of online log groups to be assigned to this VDB. The number of groups will depend on the configuration of your VDB. For more information, please visit Creating redo log groups and members[450]
Enable archivelog mode	When you enable this mode, redo logs will be archived instead of overwritten. This creates a backup of all transactions that have occurred in the database so that you can recover to any point in time. These archive logs are useful for recovering a database or updating a standby database.
Generate new DBID for VDB	Create a new Oracle DBID value for this VDB. **Note: You can also toggle this option after VDB is created.** See Generate a new DBID for Oracle VDBs (see page 1527) and Toggle new DBID generation upon refresh options for Oracle VDBs (see page 2625)
Listeners	Select an Oracle listener from the list provided to use with this VDB.

450 https://docs.oracle.com/cd/B28359_01/server.111/b28310/onlineredo003.htm#ADMIN11319

VDB setting	Explanation
Auto VDB restart	Enabling this option will automatically restart this VDB whenever its target host is rebooted. Auto VDB Restart can help you recover from host downtime automatically since the Delphix Engine will restart VDBs once they become available.
Configure File mapping	This option allows you to leverage Oracle's File Mapping feature with a VDB. Enabling this will add a step to the provisioning wizard to create file mappings.
Invoke Datapatch	Enabling this option will automatically invoke Datapatch against this VDB whenever it's provisioned, refreshed, rewound, started or enabled.

Using Oracle Multitenant offers different configuration settings, as described in the table below:

Multitenant setting	Explanation
vCDB database parameters	vCDB Database Parameters are Oracle configuration settings you can apply during the provisioning process. You can specify name-value pairs to apply these settings to the vCDB if you are creating one.
Auto vCDB restart	Enabling this setting will automatically restart this vCDB whenever the target host is rebooted. This will cause all vPDBs in the vCDB to restart as well.
Invoke Datapatch	Enabling this option will automatically invoke Datapatch against this vCDB whenever it's started or enabled.

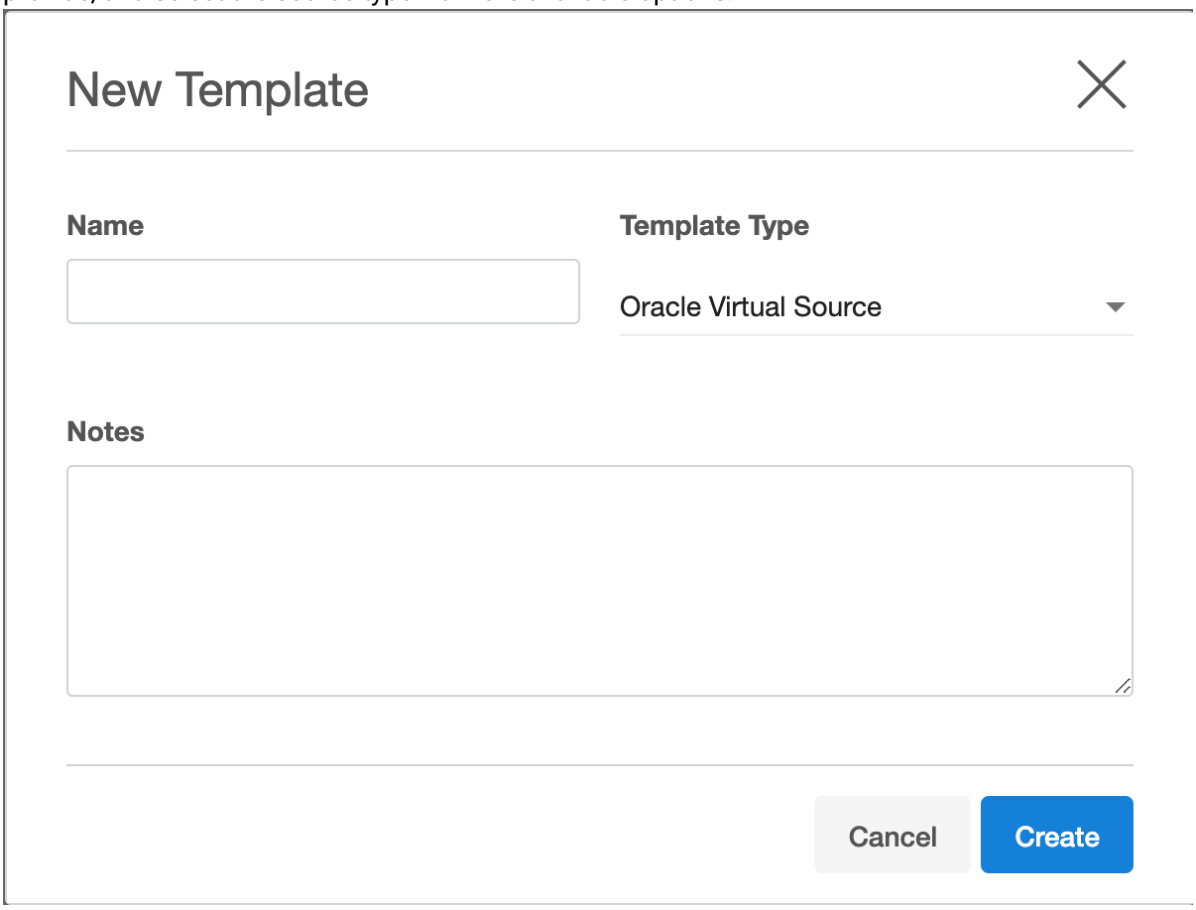
9.7.10.2.1 VDB config templates

A VDB config template is a list of database configuration parameter names and values that you can save on the Delphix Engine to use at a later time.

9.7.10.2.1.1 Creating a VDB config template Via GUI

1. Log into the **Delphix Management** application as an engine administrator.
2. Click **Manage**.
3. Select **VDB Config Templates**.
4. Click the icon next to the **VDB Config Temp...** and select **New Template** to create a new template.

5. In the New Template dialog, enter the name for the new template, parameters that you want to provide, and select the source type from the available options.



6. Click **Create**.

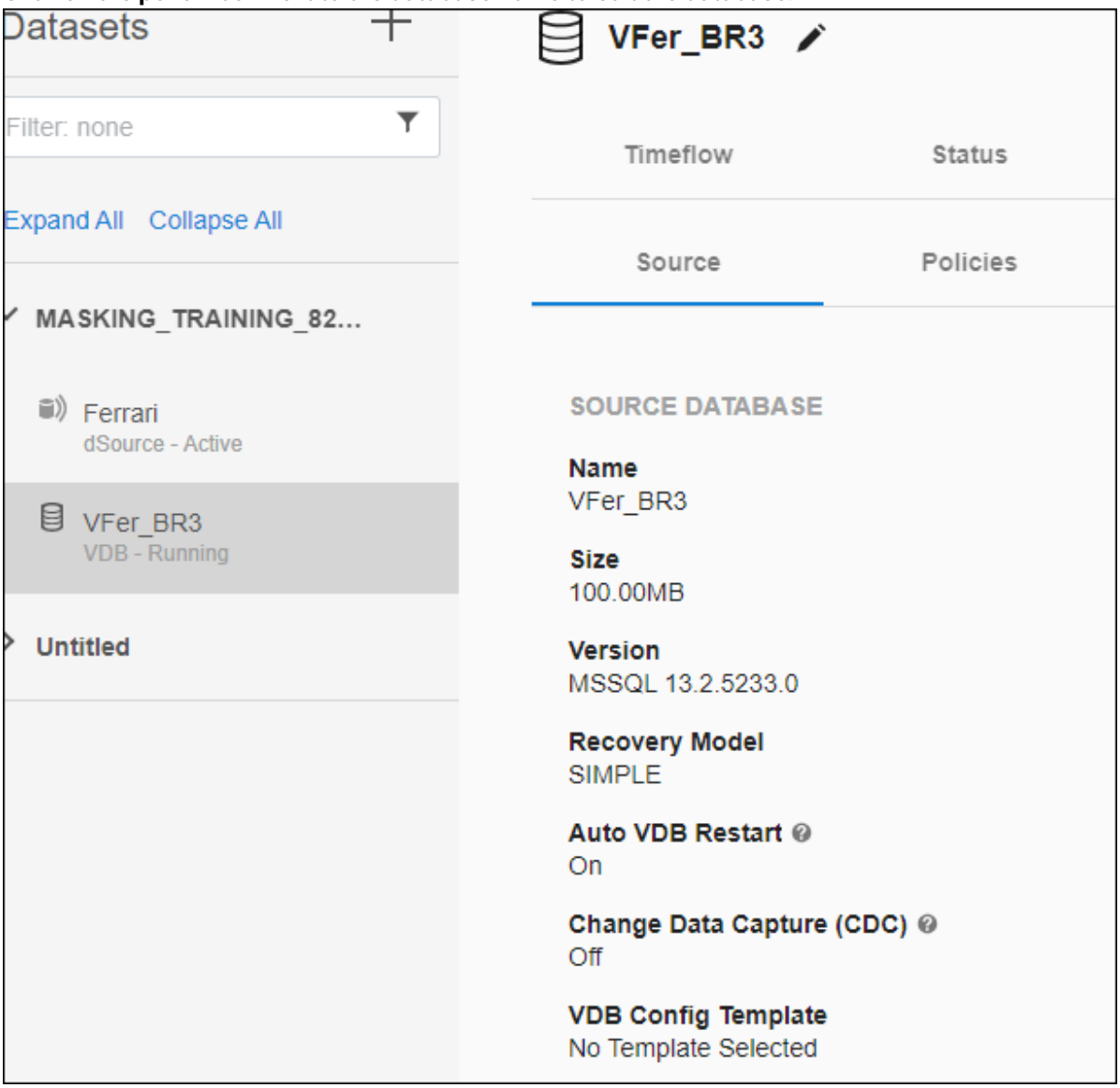
9.7.10.2.1.2 Updating a VDB config template via GUI

1. Log into the **Delphix Management** application as an engine administrator.
2. Click **Manage**.
3. Select **VDB Config Templates**.
4. Select the template from the left-side pane that you need to update.
5. Click on the **pencil** icons to edit an existing VDB template.
6. Click the button to save the changes or click the button to discard the changes that you made.

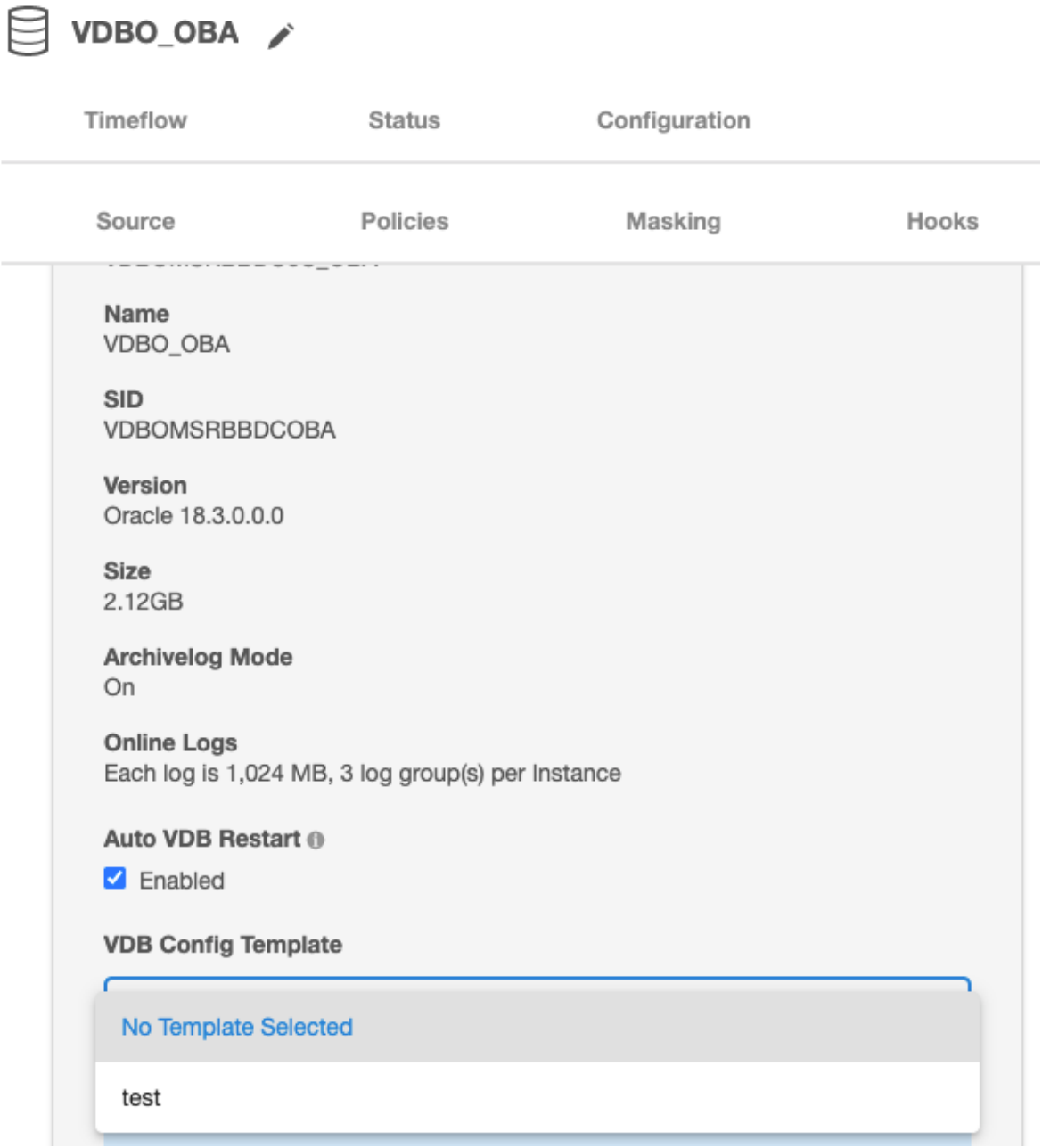
9.7.10.2.1.3 Apply/Configure a VDB config template to a VDB Via GUI

1. Login to the **Delphix Management** application as an engine administrator.

2. Click **Manage** > **Datasets.**
3. Select the VDB you want to edit.
4. From the right-side pane select the **Configuration** tab.
5. Click on the **pencil** icon next to the database name to edit the database.



6. To apply an already existing VDB template, select from the available template in the drop-down under VDB Config Template. Similarly, you can configure the VDB for an already associated template from the available templates.



7. Click the button to save the changes or click the button to discard the changes that you made.

You can apply a VDB Config Template to a VDB during the provisioning process, which copies the values from the template. Any subsequent changes to the template will be reflected in the VDB when that VDB is refreshed. During provisioning, you can specify configuration parameters directly or copy them from a VDB Configuration Template. Once set, the Delphix Engine will use these parameters whenever the VDB is refreshed, even if you change the original template. It is important to know, however, that some configuration parameters cannot be customized. In addition, some configuration parameters are stripped out during the provisioning process but are customizable. The list of restricted and customizable parameters can be found below.

9.7.10.2.1.4 Required parameters

The VDB configuration template must contain at least the following database parameters:

- **compatible** : Required for both Non-Multitenant and Multitenant sources. Must be set with the same value as in the dSource 'compatible' parameter.
- **enable_pluggable_database** : Required for Multitenant sources only. Must be set to TRUE.

9.7.10.2.1.5 Customizable VDB configure parameters

The default value for these parameters is cleared during the provisioning process. They are removed from the VDB configuration file unless you set values for them through a VDB Config Template.

- audit_file_dest
- audit_sys_operations
- audit_trail
- background_dump_dest
- core_dump_dest
- db_domain
- diagnostic_dest
- dispatchers
- fast_start_mttr_target
- log_archive_dest_n
    - n can be values between 1 - 31
- log_archive_dest_state_n
    - n can be values between 1 - 31
- remote_listener
- user_dump_dest

9.7.10.2.1.6 Restricted parameters

These parameters are restricted for use by the Delphix Engine. Attempting to customize these parameters through the use of a VDB Config Template will cause an error during the provisioning process. In addition, these parameters should not be added to, changed in or removed from the virtual database after provision as that will prevent snapshot of the virtual database.

- active_instance_count
- cluster_database
- cluster_database_instances
- cluster_interconnects
- control_files
- db_block_size
- db_create_file_dest
- db_create_online_log_dest_n
    - n can be values between 1 - 5
- db_file_name_convert

- db_name
- db_recovery_file_dest
- db_recovery_file_dest_size
- db_unique_name
- dg_broker_config_file1
- dg_broker_config_file2
- dg_broker_start
- fal_client
- fal_server
- instance_name
- instance_number
- local_listener
- log_archive_config
- log_archive_dest
- log_archive_duplex_dest
- log_file_name_convert
- spfile
- standby_archive_dest
- standby_file_management
- tde_configuration
- thread
- undo_tablespace **
- wallet_root
- __db_cache_size
- __java_pool_size
- __large_pool_size
- __oracle_base
- __pga_aggregate
- __sga_target
- __shared_io_pool_size
- __shared_pool_size
- __streams_pool_size

** Note that although this parameter is restricted for use through a VDB Config Template by the Delphix Engine, it is allowed to be modified after a virtual database has been provisioned. Any other restricted parameter that is added to, changed in or removed from the virtual database after provision will result in a failure to perform snapshots of the virtual database.

9.7.10.2.2 Automatic VDB restart on target server after reboot

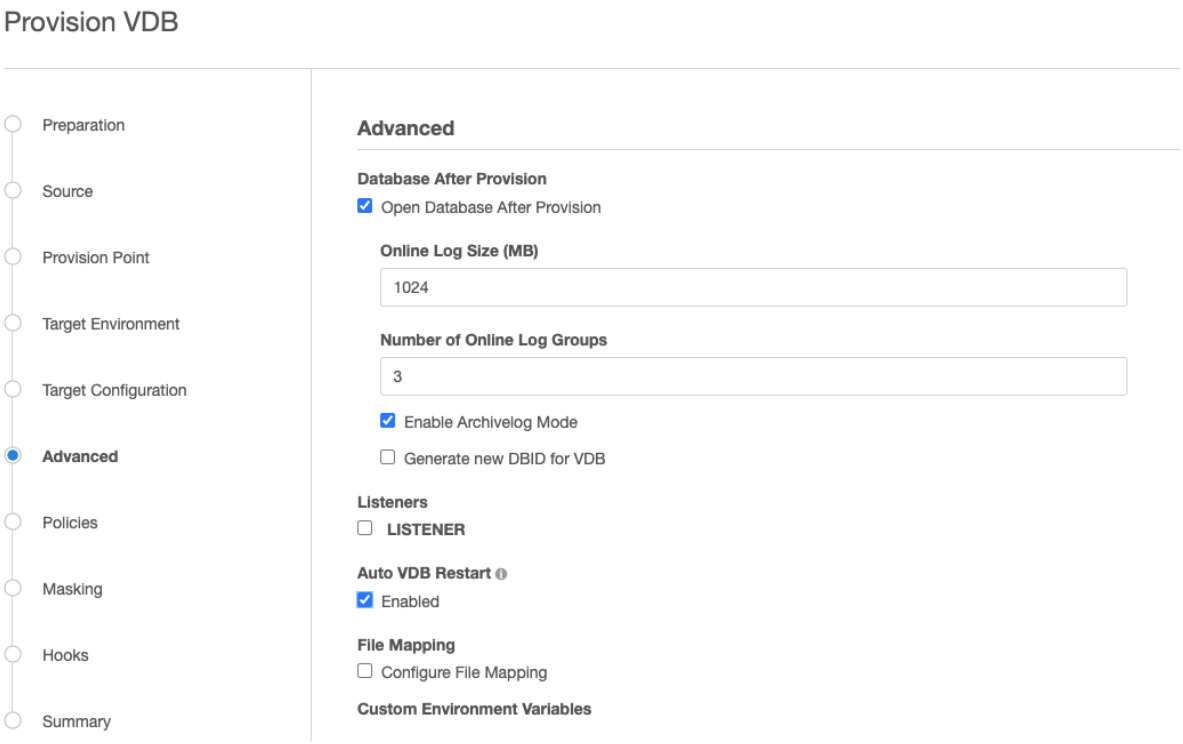
The Continuous Data Engine now automatically detects whether a target server has been rebooted, and proactively restarts any VDB on that server that was previously up and running. This happens independent of the VDB dataset; it is done as if a target server was restarted and a start command was issued from the

Delphix Engine. This feature is compatible with Delphix Self-Service (Jet Stream) ordering dependencies and is limited to non-clustered VDBs.

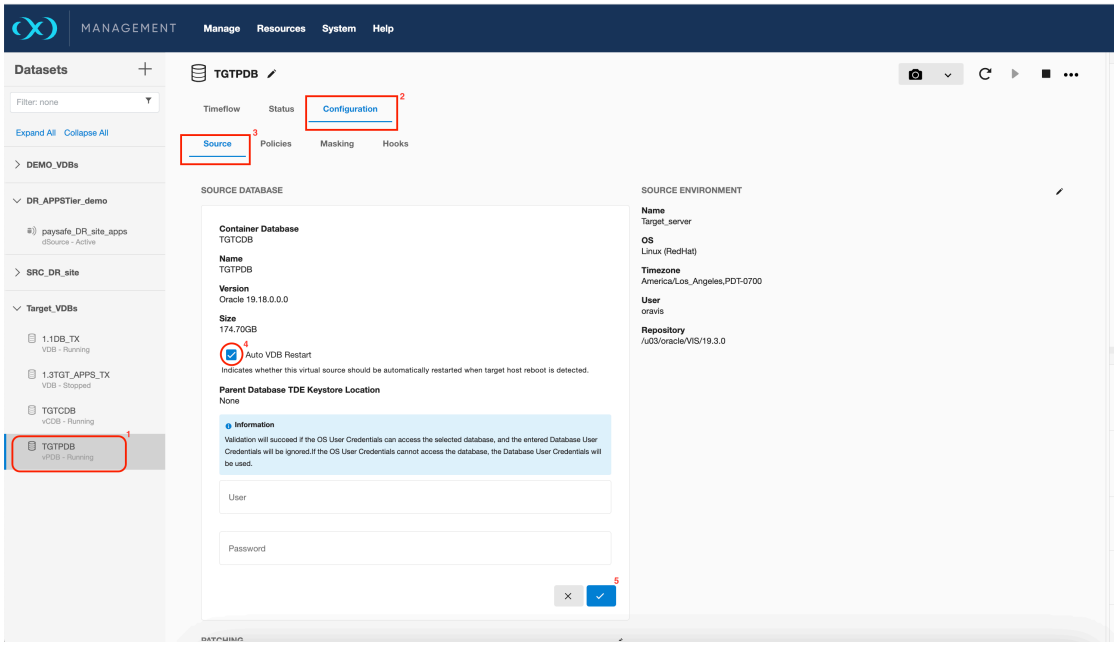
> This feature is not supported for the following data source types:
> - Oracle RAC VDBs
> - Oracle 12.1 and older vPDBs that are provisioned into a non-virtual CDB.
> - MSSQL cluster VDBs.
>
> Automatic VDB Restart is supported for following Oracle data sources types:
> - Non-Multi-Tenant (non-MT) VDBs
> - Oracle 12c and later, vPDBs that are provisioned into a virtual CDB.
> - For Oracle 12.1.0.1, users can choose to enable or disable automatic restart at the virtual CDB level. There is no individual vPDB automatic restart setting.
> - For Oracle 12.1.0.2 and later versions, users can choose to enable or disable automatic restart for a virtual CDB or a virtual PDB.
> - Oracle 12.2 and later vPDBs that are provisioned into a non-virtual CDB. Users can choose to enable or disable individual vPDB's automatic restart.
>
> Please note, the scheduler checks every one minute if any VDB needs to be restarted. The Delphix Engine waits for five minutes after it detects a host has restarted before trying to restart VDBs. The engine will keep trying to restart the VDB for 30 minutes before giving up.

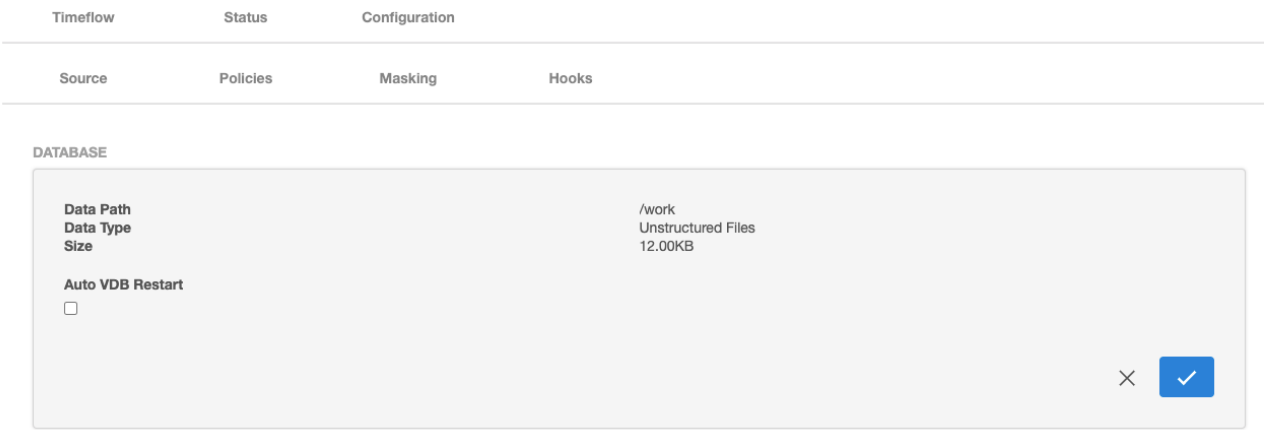
To enable automatic restart, complete the following steps:

1. When provisioning a new VDB in the VDB Provisioning wizard, check the **Auto VDB Restart** box.



2. Once the VDB has been provisioned, you will be able to turn **Automatic VDB Restart** on.
    a. In the **Datasets** panel, select the **VDB**.
    b. Select the **Configuration** tab.
    c. Select **Source** sub-tab.
    d. Select Database edit.





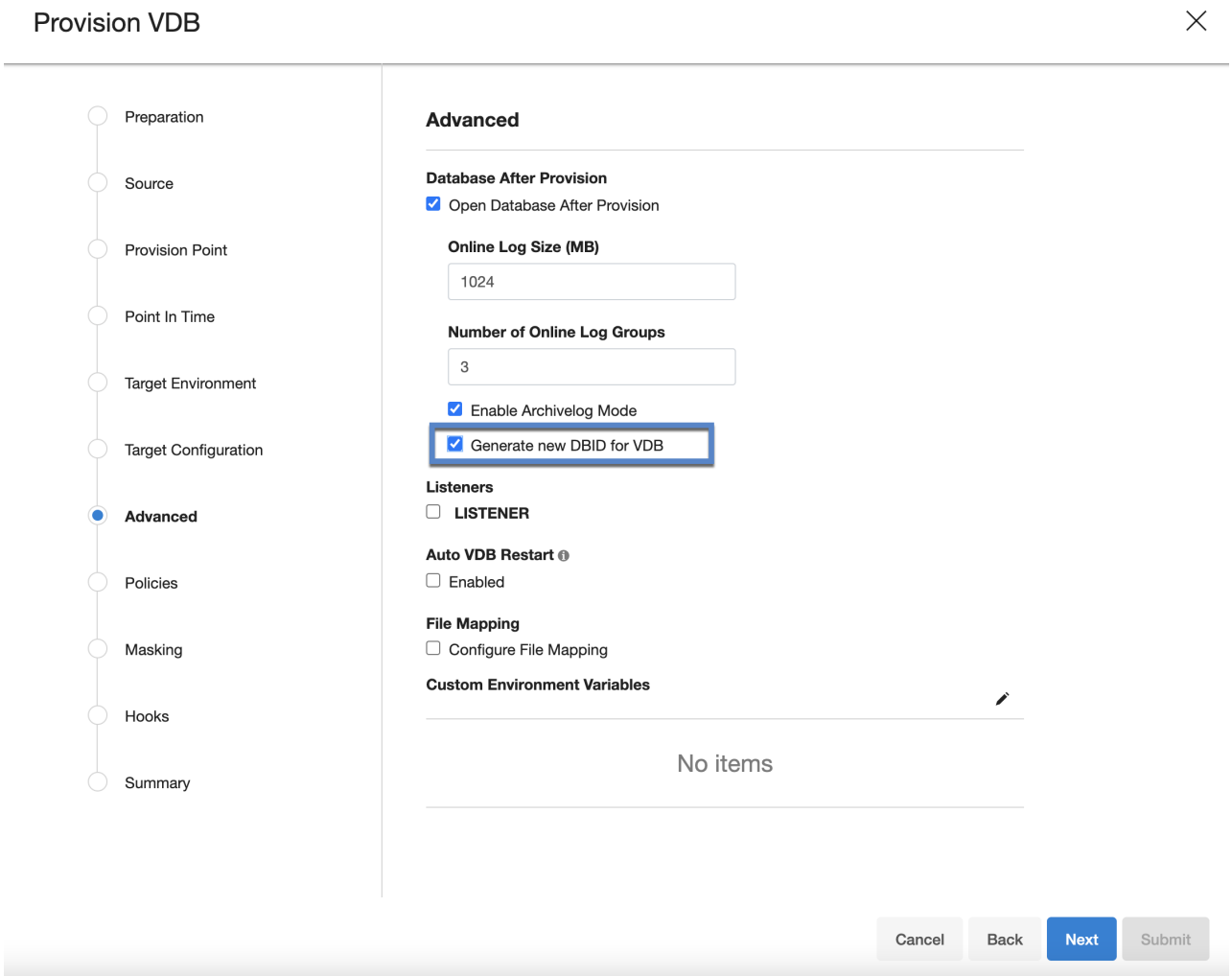
9.7.10.2.3 Generate a new DBID for Oracle VDBs

Delphix helps to generate a new Oracle internal database identifier (DBID) for VDBs. The DBID is an internal, unique identifier for an Oracle database.

> This feature is not supported for the following data source types:
> - Multitenant VDB
> - Oracle Live Source

9.7.10.2.3.1 Enable new DBID generation while provisioning a VDB

To enable generating a new DBID while provisioning a new VDB in the VDB Provisioning wizard, check the **Generate new DBID for VDB** option. This will create a new DBID for the provisioned VDB and subsequent refresh action performed to this VDB will get a different and new DBID.

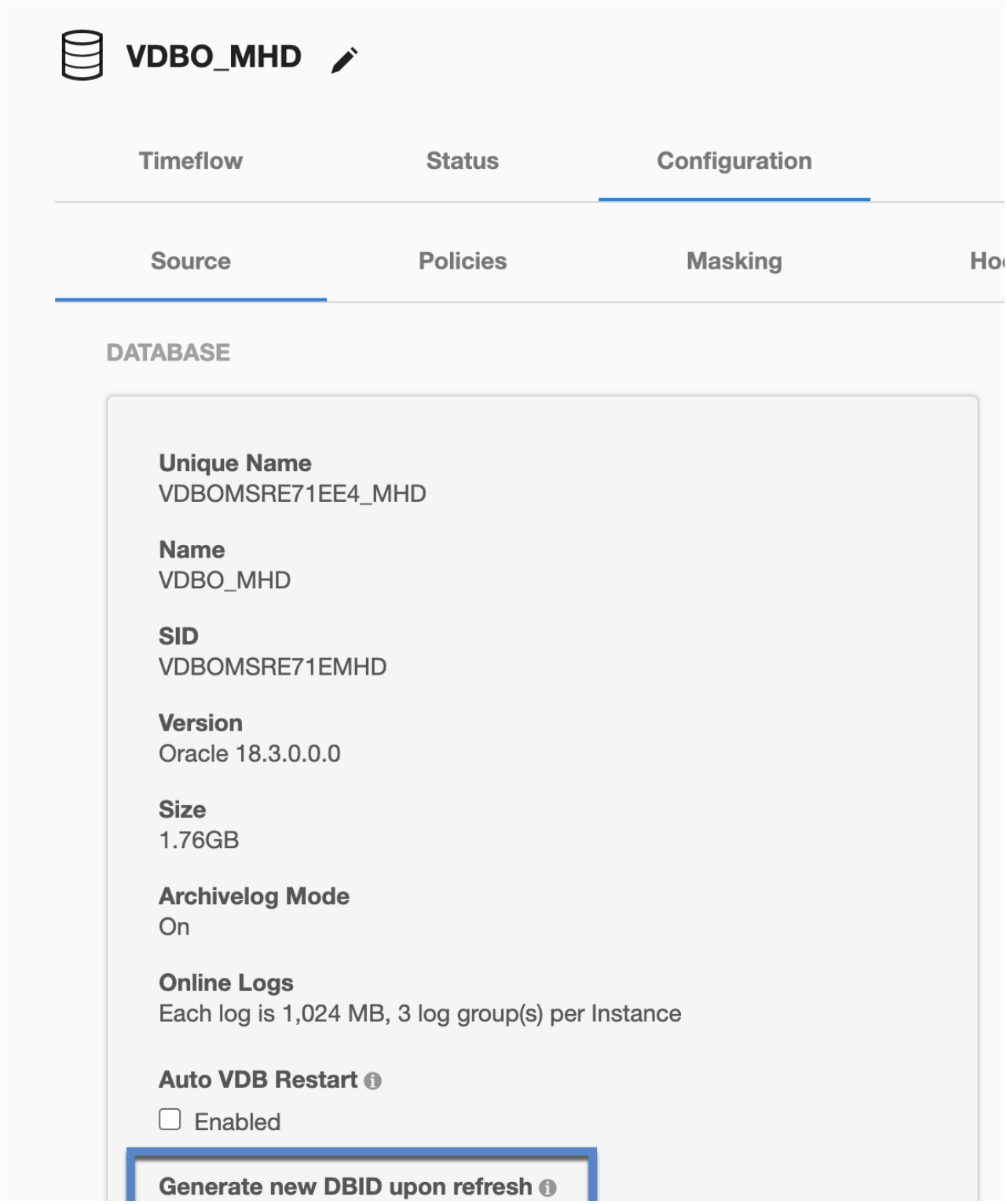


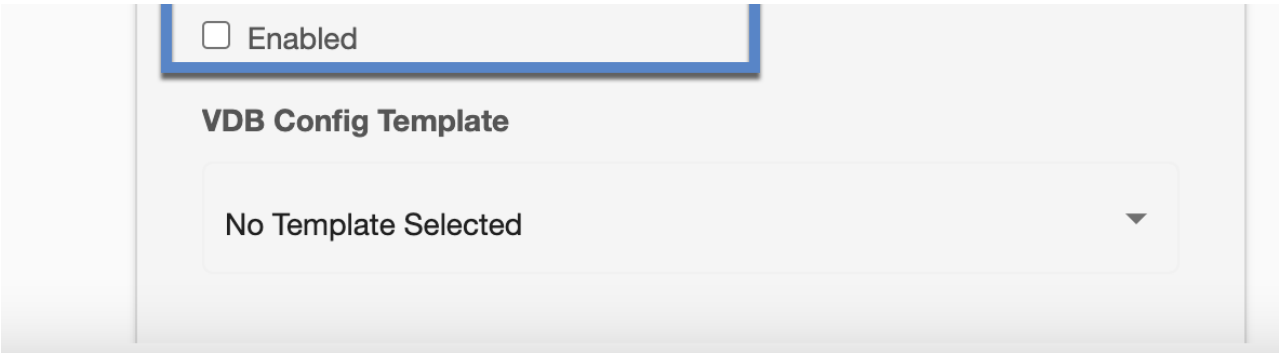
9.7.10.2.3.2 Enable new DBID generation for an already provisioned VDB

If originally the VDB had been provisioned without specifying the **generate new DBID for VDB** option, then you can enable it so that new refreshes will create VDBs with a new DBID (Note the current VDB or previous VDBs will still have the same DBID as the original database). To accomplish this follow the steps below.

1. In the **Datasets** panel, select the **VDB**.
2. Select the **Configuration** tab.
3. Select **Source** sub-tab.
4. Select Database and click the **pencil** icon to edit.
5. Check the **Generate new DBID upon refresh** option.
6. Refresh the VDB to see the changes.

> If the **Generate new DBID upon refresh** option is enabled, then the Delphix engine will generate a new DBID upon refresh.





9.7.10.2.3.3 Disable new DBID generation for an already provisioned VDB

You can stop generating the new DBIDs for VDBs that were enabled for this at the time of provisioning or after provisioning. To accomplish this follow the steps below.

1. In the **Datasets** panel, select the **VDB**.
2. Select the **Configuration** tab.
3. Select **Source** sub-tab.
4. Select Database and click the **pencil** icon to edit.
5. Uncheck the **Generate new DBID upon refresh** option.
6. This will stop generating a new DBID for subsequent refreshes.
7. Refresh the VDB to reflect the changes.

> If the **Generate new DBID upon refresh** option is disabled, then the Delphix engine will use the same DBID of the parent upon refresh.

9.7.10.2.4 Customizing Oracle VDB environment variables

9.7.10.2.4.1 Overview

This topic describes how to customize the set of environment variables sourced prior to administering an Oracle virtual database (VDB). Certain Oracle database parameters are sensitive to the environment variables present when you start or administer the database. For this reason, the Delphix Engine allows you to dictate custom environment variables that will be set prior to any administrative action, such as provision, start, stop, rollback, or refresh.

You can specify environment variables by two different means:

- **Name-value pair** – A literal variable name and value to be set
- **Environment file** – An environment file to be sourced

Environment variables for Oracle RAC databases might vary in value between cluster nodes. Therefore, environment variable specifications for an Oracle RAC database must specify the cluster node to which they apply.

9.7.10.2.4.2 Setting custom environment variables

Prerequisites

If you are adding any environment variables that are environment files, these files must be accessible on the target environment.

Procedure

1. You can configure custom environment variables in the **Provision Wizard**.
    a. On the **Target Environment** tab, click **Advanced**. or
    b. You can also configure these variables on the **Configuration tab** when the VDB is disabled.
2. Click the **Plus** icon to add an environment variable.
3. Choose a **format** for the environment variable.
    a. Name-Value Pair
        i. Enter a **Name** to identify the variable.
        ii. Enter the variable's **Value**.
        iii. For Oracle RAC databases, you must also specify the cluster node to which this environment variable applies.
    b. Environment File
        i. Enter an absolute path to an environment file on the target environment. This path can be followed by parameters. Paths and parameters are separated by spaces.
        Escaping Spaces
        To specify literal spaces, escape them with a backslash ("hello\ world" -> "hello world"). To specify literal backslashes, escape them with a backslash ("foo\" -> "foo"). Any other character preceded by a backslash will retain both the backslash and the original character ("\b" -> "\b"). Escaping is done in order from left to right ("part1\ part2" -> "part1" "part2" will be two parameters).
        ii. For Oracle RAC databases, you must also specify the cluster node to which this environment variable applies.
4. Save the custom environment variables by completing provisioning, or clicking the **Confirm**. These environment variables will take effect when you start the Oracle VDB.

9.7.10.2.4.3 Environment variable denylist

The Delphix Engine denylists the following environment variables; they cannot be set by the user.

- `ORACLE_SID`
- `ORACLE_BASE`

- `ORACLE_HOME`
- `CRS_HOME`
- `ORACLE_UNQNAME`
- `ORAENV_ASK`
- `LOGON_STR`
- `DLPX_SHELL`
- `SQLPLUS_PLSQL_MODIFIERS`
- `SQLPLUS_DML_MODIFIERS`
- `SQLPLUS_DDL_MODIFIERS`

If a Name-Value pair has any one of these prohibited environment variables as the name, an error will be raised.

If an environment file sets one of these variables, the Delphix Engine will override this value when the Oracle VDB is started.

The following environment variables will be set before invoking the user-specified script, and thus can be accessed within the script.

- `ORACLE_SID`
- `ORACLE_BASE`
- `ORACLE_HOME`
- `CRS_HOME`
- `ORACLE_UNQNAME`

9.7.10.2.4.4 User-Input sanitation for environment variables

For security purposes, user-input provided through the custom environment variables feature retains its literal value when interpreted, including ', ", and undefinedORACLE_HOME

- `undefinedORACLE_SID`
- `undefinedORACLE_UNQNAME`
- `undefinedPATH`

9.7.10.2.4.5 Caveats

- Environment variables are sourced on provision, start, stop, rollback, and refresh. Custom environment variables are not applicable to V2P.
- Custom environment variables do not propagate to child VDBs and must be set again on provision.
- Custom environment variables do not persist after migration. On migration of a VDB with custom environment variables, an alert will be raised that the custom environment variables have been removed from the VDB. In order to view the alert, go to **System** > **Event Viewer**.
- Custom environment variables are *not* available within scripts executed by VDB Hook Operations.

9.7.10.2.5 Customizing VDB file mappings

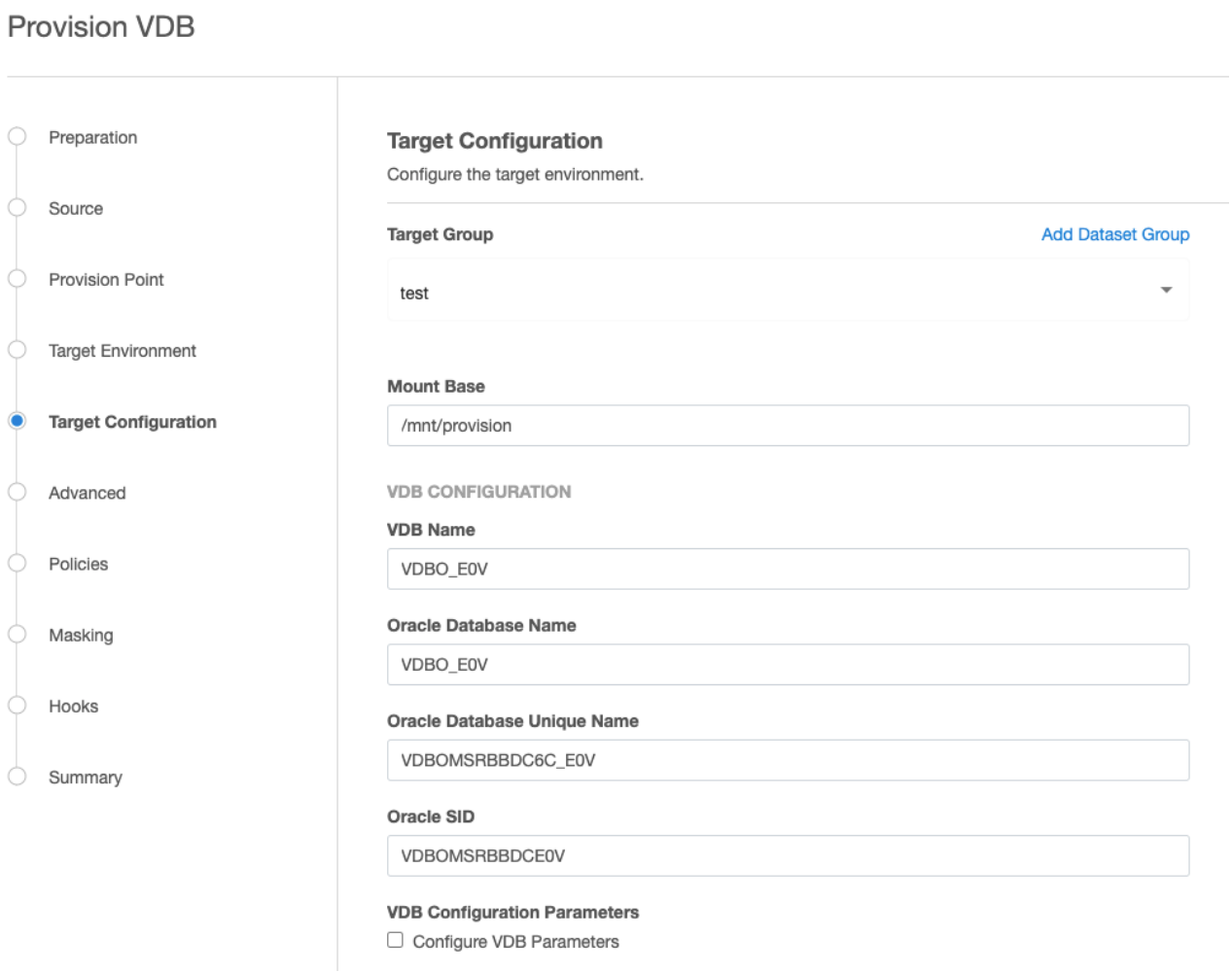
This topic describes how to customize file path mappings when provisioning a virtual database (VDB).

In the VDB provisioning process, it may be necessary to create mappings between files and directories that exist on the source, and files or file directories that exist on the target. An example of this is creating a copy in the target environment of a wallet file for an encrypted database that exists in the source environment.

During VDB provisioning, the archive, datafile, external, script and temp directories are mounted from the Delphix Engine to the unique="">>

You can specify the Mount Base and the Database Unique Name on the Provision VDB card at provision time.
But the archive, datafile, external, script and temp mount points are fixed.

- The default Mount Base is set to /mnt/provision
- The default Database Unique Name starts with a capital V and includes the dSource name



Mount Base: /mnt/provision

Database Unique Name: Vdbhcp3_35F

Mounts: /mnt/provision/Vdbhcp3_35F/archive

/mnt/provision/Vdbhcp3_35F/datafile

/mnt/provision/Vdbhcp3_35F/external

/mnt/provision/Vdbhcp3_35F/script

/mnt/provision/Vdbhcp3_35F/temp

File Mappings affect everything that follows the mount point directories above. For datafiles, the datafile location from the dSource is used. You can determine the directory structure by querying the Source database:

```
SQL> select name from v$datafile;
```

```
NAME
```

```
--------------------------------------------------------------------------------
```

```
/datafile/dbdhcp3/oradata/dbdhcp3/system01.dbf
```

```
/datafile/dbdhcp3/oradata/dbdhcp3/sysaux01.dbf
```

```
/datafile/dbdhcp3/oradata/dbdhcp3/undotbs01.dbf
```

```
/datafile/dbdhcp3/oradata/dbdhcp3/users01.dbf
```

```
/u03/app/ora11202/product/11.2.0/dbhome_1/dbs/dbv_R2V4.dbf
```

Users have control over the datafiles that were returned by the select name from v$datafile and also the controlfile location.

Archive log files go directly into /mnt/provision/Vdbhcp3_35F/archive
Tempfiles go directly into /mnt/provision/Vdbhcp3_35F/temp

By default VDB File Mapping would append the above directory structure to the /mnt/provision/Vdbhcp3_35F/datafile directory:

```
/mnt/provision/Vdbhcp3_35F/datafile/datafile/dbdhcp3/oradata/dbdhcp3/system01.dbf
/mnt/provision/Vdbhcp3_35F/datafile/datafile/dbdhcp3/oradata/dbdhcp3/sysaux01.dbf
/mnt/provision/Vdbhcp3_35F/datafile/datafile/dbdhcp3/oradata/dbdhcp3/undotbs01.dbf
/mnt/provision/Vdbhcp3_35F/datafile/datafile/dbdhcp3/oradata/dbdhcp3/users01.dbf
```

```
/mnt/provision/Vdbhcp3_35F/datafile/u03/app/ora11202/product/11.2.0/dbhome_1/dbs/
dbv_R2V4.dbf
```

Pattern matching example

You can use pattern matching rules to create full path names for data files and control files.

Pattern matching rules have the form **source-regex-expression-KEY : target-replacement-VALUE**. You can use multiple rules, which are applied successively. Multiple rules with the same source key are allowed.

9.7.10.2.5.1 File mapping options

Example 1

For this example, the ultimate goal is to create a VDB that has the following datafile File Mappings:

```
/u03/devvdb3/datafile/sys/system01.dbf
```

```
/u03/devvdb3/datafile/sys/sysaux01.dbf
```

```
/u03/devvdb3/datafile/ctrl/control01.ctl
```

```
/u03/devvdb3/datafile/undo/undotbs01.dbf
```

```
/u03/devvdb3/datafile/data/users01.dbf
```

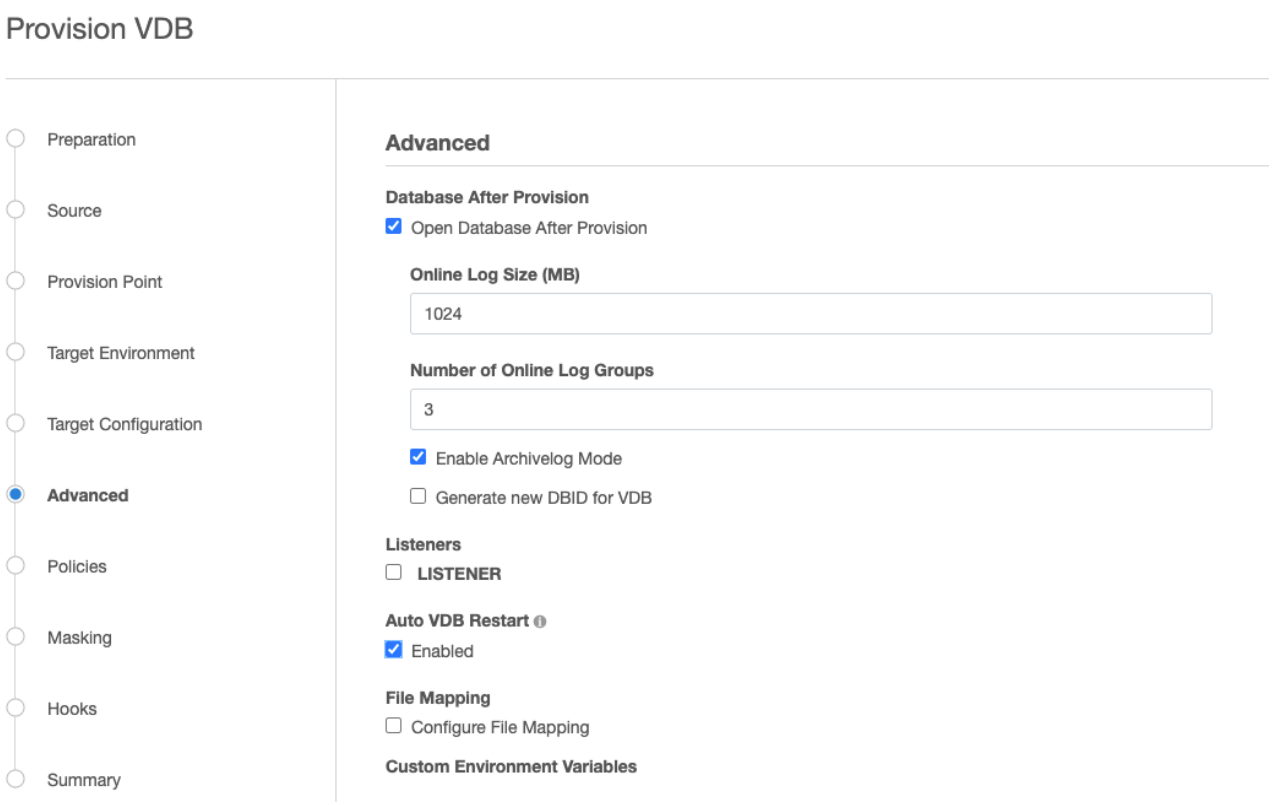
```
/u03/devvdb3/datafile/data/dbv_R2V4.dbf
```

You can change the default behavior in a few different ways.

1. Modify the Mount Base.
2. Modify the Database Unique Name.
3. Modify the Mount Base and the Database Unique Name.
4. Modify the name of the individual subdirectories within the Mount Base/Database Unique Name/datafile directory.

Numbers 1-3 can be changed by simply modifying the Mount Base and/or Database Unique Name values when provisioning a VDB:

The datafile locations can be changed in the **Advanced** tab of the **Provision VDB** wizard and scroll down to **File Mapping**:



When you are changing File Mapping, remember that you only need to refer to the part of the path after / Mount Base/Unique Database Name/datafile.

Start by changing all the datafile locations from /u03/devvdb3/datafile/datafile/dbdhcp3/oradata/dbdhcp3

to

/u03/devvdb3/datafile/data by selecting the **+** sign and setting **File Mapping Source File Match** and the **Replacement** and then select **Validate** to see result.

The result will show the new directory structure for all the datafiles and control file relative to the " /Mount Base/Unique Database Name/datafile" directory, but will only show the directories after the datafile directory.

```
Source File Match                  Replacement         Result
/datafile/dbdhcp3/oradata/dbdhcp3  /data              /data/users.dbf
```

Selecting Validate will show the result of the above mapping, for all the datafiles.

The File Mappings build upon one another, so the first mapping moves almost all the datafiles and the controlfile to /u03//devvdb3/datafile/data.

/u03/app/ora11202/product/11.2.0/dbhome_1/dbs/dbv_R2V4.dbf was not remapped, as it did not contain the directory structure defined in in Source File Match (/datafile/dbdhcp3/oradata/dbdhcp3)

If you wanted the remaining datafiles and controlfile to remain in /data, it would only require the two File Mappings.

The remaining datafiles and controlfile can be relocated by adding to the mapping (select "+" between each Source File Match/Replacement pair).

```
/datafile/dbdhcp3/oradata/dbdhcp3/system01.dbf              /sys/system01.dbf      /
sys/system01.dbf
```

```
/datafile/dbdhcp3/oradata/dbdhcp3/sysaux01.dbf              /sys/sysaux01.dbf      /
sys/sysaux01.dbf
```

```
/datafile/dbdhcp3/oradata/dbdhcp3/undotbs01.dbf             /undo/undotbs01.dbf    /
undo/undotbs01.dbf
```

```
/datafile/dbdhcp3/oradata/dbdhcp3/control01.ctl             /ctrl/control01.ctl    /
ctrl/control01.ctl
Select Validate between each new entry in order to verify that datafiles are being
mapped as expected.
```

Once all the files are located the way you want them, select **Next** to continue the provision process.

The **Summary** page will show the modifications to **Mount Base** and **Unique Database Name** and will show that **Customized File Mapping** was defined.

After provisioning completes, you can login to the Target server and verify that the datafiles were mapped correctly:

```
SQL> select name from v$datafile;
```

```
NAME
```

```
--------------------------------------------------------------------------------
```

```
/u03/devvdb3/datafile/sys/system01.dbf
```

```
/u03/devvdb3/datafile/sys/sysaux01.dbf
```

```
/u03/devvdb3/datafile/undo/undotbs01.dbf
```

```
/u03/devvdb3/datafile/data/users01.dbf
```

```
/u03/devvdb3/datafile/data/dbv_R2V4.dbf
```

Example 2

In this example, several rules are applied to the source file path **/app/oracle/oradata/system01.dbf**.

1. Applying the rule **ora:foo** results in: **/app/foocle/foodata/system01.dbf**
2. Applying the rule **foo:bar** results in: **/app/barcle/bardata/system01.dbf**
3. Applying the rule **ora:no** results in an error, because **ora** is no longer found in the pathname.
4. Applying the rule **bar:oranew** results in: **/app/oranewcle/oranewdata/system01.dbf**
5. Applying the rule **ora:yes** results in **/app/yesnewcle/yesnewdata/system01.dbf**

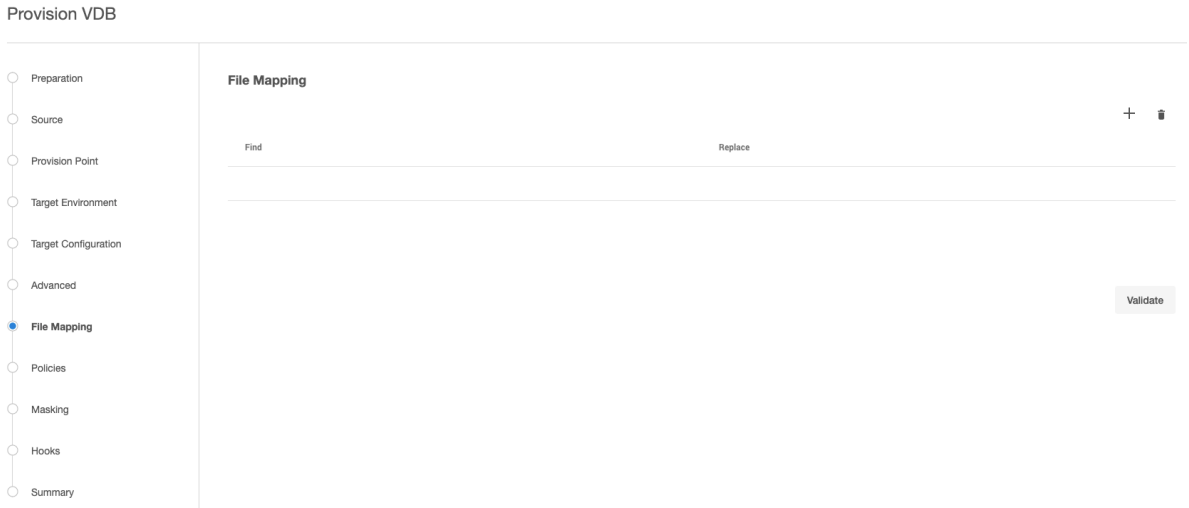
During the pattern matching process, two errors can be generated.

1. **No match for specified mapping rules** This is the result when no rules match a source file
2. **Invalid regex pattern specified for path mapping** This is the result of an invalid regex rule mapping

This topic on the java.regex.util class[451], hosted on docs.oracle.com[452], shows the regular expression syntax and constructs recognized by the Delphix Engine pattern-matching operations.

9.7.10.2.5.2 Applying VDB file mappings during the provisioning process

1. In the **Target Configuration** tab of the **Provision VDB** wizard, click **Advanced**.
2. Select **Configure File Mapping**.
3. Click **Next**.
4. Click the **Plus** icon to add a mapping rule.



451 http://docs.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html
452 http://docs.oracle.com/

5. Enter the mapping rule.
6. Click **Validate** to see the results of applying the rule. If not matches are found, you will see an error message.
7. Click **Next** to continue with the provisioning process.

9.7.10.2.6 Manually starting a VDB

When starting an Oracle VDB instance, an initialization file `$ORACLE_BASE_CONFIG/dbs` is used and by default, this directory does not have group write permissions. Previously to Delphix 6.0.10.0, this limitation forced target hosts to either use the instance owner for provisioning or modify the group permissions on this directory. Both options can be seen as a security risk and increase deployment complexity.

With Delphix Engine 6.0.10.0 onwards, a target host no longer requires write permission to the `$ORACLE_BASE_CONFIG/dbs` directory. Delphix Engine will only copy an initialization file *init<>* to this directory if write permissions exist.

All instance startup attempts from Delphix operations will specify an initialization file to use from the Delphix filesystem rather than the default Oracle location. If a VDB needs to be manually started and Delphix was unable to copy the initialization then it must be specified in the startup command. The instance init file is available in the VDB script directory.

You can use the following syntax to manually start the instance:

```
SQL> startup pfile='/mnt/provision/VDBOMSR66E005_61O/script/VDBOMSR66E061O/
initVDBOMSR66E061O.ora';
ORACLE instance started.
```

If there is no write permission in the `$ORACLE_BASE_CONFIG/dbs` directory then manually starting the instance will fail:

```
SQL> startup
ORA-01078: failure in processing system parameters
LRM-00109: could not open parameter file '/u01/app/oracle/product/11.2.0.4/dbhome_1/
dbs/initVDBOMSR66E061O.ora'
```

> When starting a VDB manually, be sure to use the Delphix's initialization Parameter File (pfile) which in turn points to the Delphix's Server Parameter File (spfile).
> Failure to do so will prevent taking a snapshot of the VDB/vCDB until the database is started with the Delphix's Server Parameter File.

To manually start a VDB, complete the following steps:

1. Ensure that the VDB is enabled and NFS filesystems are mounted. This is required only when the NFS filesystems are offline. Follow the below steps via CLI.

```
demo source> select VDBOMSR66E005_61O
demo source 'VDBOMSR66E005_61O'> enable
demo source 'VDBOMSR66E005_61O' enable *> set attemptStart=false
demo source 'VDBOMSR66E005_61O' enable *> commit
    Dispatched job JOB-47
    SOURCE_ENABLE job started for "VDBOMSR66E005_61O".
    Enabling dataset "VDBOMSR66E005_61O".
    Exporting storage containers from the Delphix Engine.
    Mounting datasets.
    Dataset "VDBOMSR66E005_61O" enabled.
    SOURCE_ENABLE job for "VDBOMSR66E005_61O" completed successfully.
```

2. Login to the target host as a Delphix OS user.
3. Locate the VDB scripts directory. This is located at `/<toolkit_directory>/Delphix_<engineuuid>_<delphix_osuser_id>_<host|cluster>/databases/oracle/<vdb_uniq_name>/<vdb_instance_name>/.`

```
cd /work/Delphix_8e501f827dee_a7a9072fc4fe_2_host/databases/oracle/
VDBOMSR66E005_61O/VDBOMSR66E061O
```

4. Run "`setup-oraenv.sh`". This configures the Oracle environment variables such as ORACLE_SID and ORACLE_HOME.
5. Startup the instance by specifying the path to the instance init file in the scripts directory.

```
[oracle@mwrh74-ora11204-tgt VDBOMSR66E061O]$ sqlplus "/as sysdba"

SQL*Plus: Release 11.2.0.4.0 Production on Thu Jul 22 02:40:47 2021

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Connected to an idle instance.

SQL> startup pfile='/work/Delphix_8e501f827dee_a7a9072fc4fe_2_host/databases/
oracle/VDBOMSR66E005_61O/VDBOMSR66E061O/initVDBOMSR66E061O.ora';
ORACLE instance started.

Total System Global Area 1068937216 bytes
Fixed Size          2260088 bytes
Variable Size         616563592 bytes
Database Buffers      444596224 bytes
Redo Buffers            5517312 bytes
Database mounted.
Database opened.
SQL>
```

If Delphix Engine has permission to write to `$ORACLE_BASE_CONFIG/dbs` during provisioning, the instance can be started up with only the "startup" command.

9.7.10.3 Provisioning an Oracle VDB

This topic describes how to provision a virtual database (VDB) from a dSource or another VDB.

Masked Provisioning is supported on Oracle RAC only when used with "script-based masking".

9.7.10.3.1 Prerequisites

- You must have already done one of the following:
  - linked a dSource from a source database, as described in Linking an Oracle Data Source (see page 1438) or
  - created a VDB from which you want to provision another VDB
- You will need to have the correct OS User privileges on the target environment, as described in Requirements for Oracle Target Hosts and Databases (see page 1392)
- If you want to use customized database configuration settings, first create a VDB Config Template as described in Customizing Oracle VDB Configuration Settings (see page 1517)
- If you are creating a VDB from a dSource linked to an encrypted database, make sure you have copied the wallet file to the target environment, as described in Provisioning a VDB from an Encrypted Oracle Database.
- If you want to invoke Datapatch for the VDB, you will need to grant the environment user used for this provision correct sudo privileges on the target environment, as described in Sudo Privilege Requirements for Oracle Environments (see page 1405).
- If you are provisioning a VDB in an Oracle RAC environment, you will need to ensure that all the enabled nodes of the target environment are also accessible from the Delphix engine.
  You must disable any RAC node (see page 0), that cannot be accessed, from the **Delphix Management** application. Before disabling such a RAC node, make sure that :
  - All virtual source instances on that node are stopped (from Delphix), if the RAC node is still running or
  - The RAC node is shut down.

VDB configuration templates

It is recommended that you always create a VDB Configuration Template prior to provisioning a Virtual Database. This will allow you to customize your parameters for the initial provisioning, and ensure that subsequent changes are reflected on the VDB when refresh and rewind operations are run. See Customizing Oracle VDB Configuration Settings (see page 1517) for more information.

Once a VDB has been provisioned, archive logs, flashback logs, redo logs, and datafiles that are not known to be part of the database will be deleted when a SnapSync is performed on the VDB. As a result, make sure that no files are added to the `datafile` file system, as this is used by Delphix to store datafiles. Use the `external` filesystem for other files that are needed to be included in the snapshot that are not datafiles.

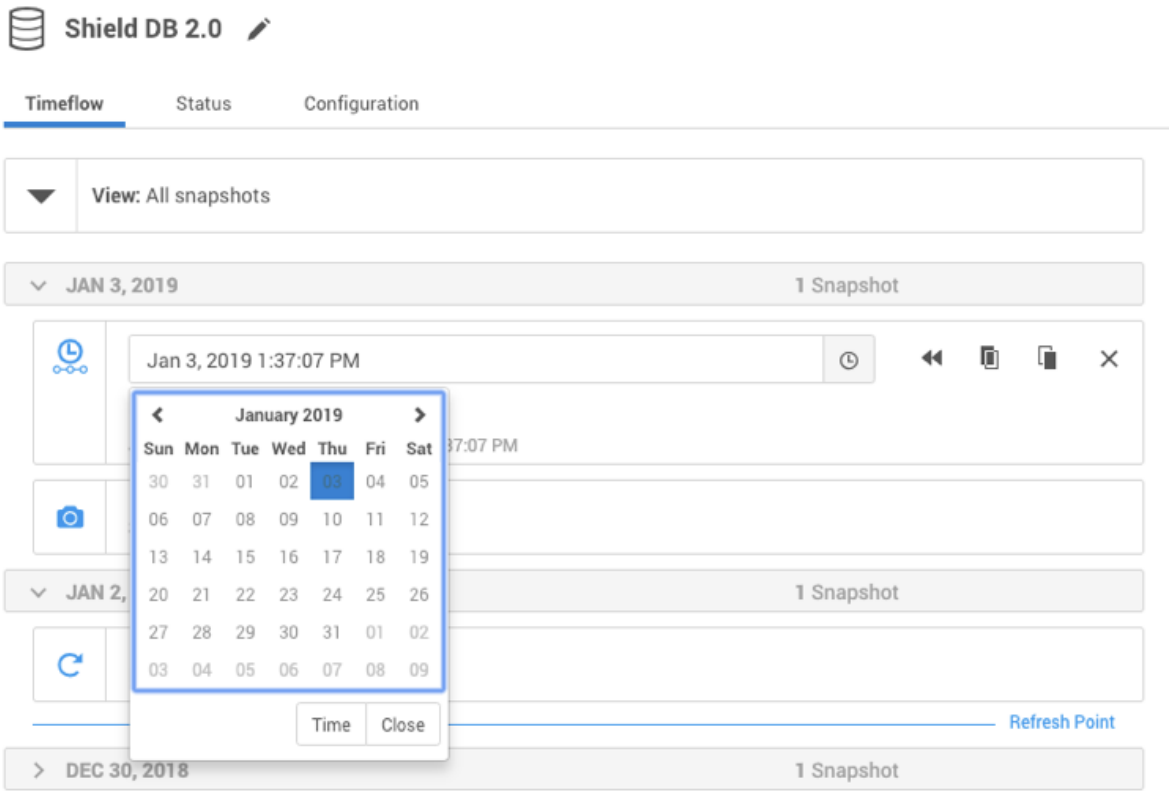
For example, creating a second control file on the `datafile` filesystem is prohibited. Such a file will be removed after SnapSync, rendering the VDB unusable. Similarly, block change tracking must not be enabled on a VDB.

9.7.10.3.2 Procedure

1. Login to the **Delphix Management** application.
2. Select **Manage > Datasets**.
3. In the **Datasets** panel on the left-hand side, click the **group** containing the dSource or VDB from which you want to provision.
4. Click the **TimeFlow** tab.
5. Select a **snapshot**.
   For more information on provisioning options.

   Info : You can take a snapshot of the dSource from which to provision. To do so, click the **Camera** icon.
6. **Optional:** Select

   to open LogSync timeline.



7. Select

   to provision from a point of time within a snapshot. You can select by date or time.

8. Click

   and the **Provision VDB**wizard will open:

- For **Oracle Single Instance** the fields **Installation Home**, **Database Unique Name**, **SID**, **Database Name**, **Mount Base**, and **Environment User** will auto-populate with information from the parent.
- For **Oracle RAC** the fields **Installation Home**, **Database Unique Name**, **SID**, **Database Name**, **Mount Base**, **Instance Number, Instance Name** and **Environment User** will auto-populate with information from the parent.

Editable Fields in the VDB Provision Wizard
The following fields are editable:
Installation Home (need to have an additional compatible target)
Database Unique Name
SID
Database Name

Mount Base
Instance Number (RAC Only)
Instance Name (RAC Only)

9. If you need to add a new target environment for the VDB, click the green **Plus** icon next to the **Filter Target** field, and follow the instructions in Adding an Oracle Single Instance or RAC Environment (see page 1424)
10. Review the information for **Installation Home**, **Database Unique Name**, **SID**, and **Database Name**. Edit as necessary.
11. Review the **Mount Base** and **Environment User**. Edit as necessary. The Environment User must have permission to write to the specified Mount Base, as described in Requirements for Oracle Target Hosts and Databases (see page 1392). You may also want to create a new writeable directory in the target environment with the correct permissions and use that as the Mount Base for the VDB.
    a. Linux and Unix hosts, this mount path must be the full path and not include symlinks.
12. Select **Provide Privileged Credentials** if you want to use login credentials on the target environment that are different from those associated with the **Environment User**.
13. Click **Advanced** to customize the VDB online log size and log groups, archivelog mode, local_listener parameter (TCP/IPC protocol addresses), additional VDB configuration settings or file mappings, patching, or custom environment variables.If you are provisioning to a target environment that is running a Linux OS, you will need to compare the `SGA_TARGET` configuration parameter with the shared memory size in `/dev/shm`. The shared memory configured on the target host should match the SGA memory target. You can check the Linux OS shared memory size with the command df -k /dev/shm and the SGA_TARGET configuration parameter by opening the **Advanced** settings and then finding the value for `SGA_TARGET` under **VDB Configuration Templates**.
14. Click **Next**.
15. Select a **Target Group** for the VDB.
16. Enable Auto VDB Restart to allow the VDB to be automatically restarted when the target host reboot is detected by Delphix.

The **Auto VDB Restart** feature is currently not available for Clusters or LiveSource.

17. Click Next.
18. Select a **Snapshot Policy** for the VDB.
19. Click **Next**.
20. Enter any operations that should be run at Hooks during the provisioning process.
21. Click **Next**.

22. Click **Submit**.

When provisioning starts, you can review the progress of the job by selecting the VDB and clicking on the **Status** tab, or by selecting **Manage/Dashboards** and viewing the **Job History** panel. Alternatively, you could see this in the **Actions Sidebar.** When provisioning is complete, the VDB will be included in the group you designated and listed in the **Datasets** panel. If you select the VDB in the **Datasets** panel and click the **Configuration** tab, you can view information about the database and its Data Management settings.

> If the provisioning fails due to a failure in a hook operation, review the output of the script and correct the error(s). Then you can either refresh the new VDB or delete it and provision it again. Note that you cannot enable or start the VDB after a failed provisioning.

9.7.10.3.3 Provisioning by snapshot or LogSync

Refer to Overview of Provisioning Oracle Virtual Databases (see page 1515)

9.7.10.3.4 Adding or removing RAC VDB cluster node after a VDB is provisioned

9.7.10.3.5 Prerequisites

- Make sure that all the enabled RAC nodes of the target environment are also accessible from the Delphix engine.
  You must disable any RAC node (see page 0), that cannot be accessed, from the **Delphix Management** application. Before disabling such a RAC node, make sure that :
  - All virtual source instances on that node are stopped (from Delphix), if the RAC node is still running or
  - The RAC node is shut down.
- Disable the VDB before you edit the instance configuration.

9.7.10.3.6 Procedure

After provisioning:

1. Click the **group** containing the VDB.
2. Click the **VDB**.
3. Disable the VDB in order to make changes to instance configuration.
4. Under **Configuration > Source** click the **edit** button and edit the following:
   a. **Instance Number** for each corresponding instance
   b. **Instance Name**
   c. Check or uncheck the cluster nodes you want for this RAC VDB

5. Click the check button to save changes.
6. Enable the VDB to apply the instance configuration changes.

9.7.10.4 Provisioning an Oracle virtual pluggable database (vPDB)

In the Oracle multitenant architecture, there are two main database types: container databases (CDBs) and pluggable databases (PDBs).

> Delphix does not support CDBs containing an Oracle Application Container or Oracle Application PDB.
> An application container is an optional, user-created CDB component that stores data and metadata for one or more application back ends. An application PDB is a PDB that resides in an application container.
> Refer to the Oracle documentation[453] for more details.

The process of creating virtual pluggable databases is similar to creating non-multitenant virtual databases, with a few additional steps necessary in the multitenant architecture.

> A Solaris x86 source host is compatible with a Linux x86 target host and vice versa.

Delphix supports provisioning Oracle Virtual Pluggable Databases (vPDBs) in two configurations:

1. **Linked container databases (Linked CDBs)**: Physical CDBs that have been previously provided by the Oracle DBA on the target environment to which Delphix may provision vPDBs. Physical CDBs must be configured and set up specifically for use by Delphix.
2. **Virtual container databases (vCDBs)**: vCDBs are created by Delphix during the provision workflow for vPDBs. Once created for Oracle versions 12.1.0.2 and later, it may be used to provision additional vPDBs.

9.7.10.4.1 Prerequisites for provisioning a vPDB to a linked CDB

There must be a target environment that has an Oracle installation compatible with the Oracle installation of the source CDB and the source PDB. The following database requirements are needed to provision the vPDB to a linked CDB. For more information, see Requirements for Oracle Hosts and Databases (see page 1392)

- Recommend autodiscovery so that the linked CDB can be found. Otherwise, the linked CDB must be manually discovered before provisioning.
- Linked CDB must be running

453 https://docs.oracle.com/en/database/oracle/oracle-database/21/multi/application-containers2.html

- Linked CDB must be in ARCHIVELOG mode
- Linked CDB should be using Block Change Tracking (BCT)
- LogSync must be enabled for the Linked CDB.

9.7.10.4.2 Provisioning a vPDB to a linked CDB

For provisioning a vPDB into a Linked CDB, the source CDB and the target CDB must meet the following compatibility requirements:

- The value of the COMPATIBLE parameter in the source CDB must be less than or equal to the value of the COMPATIBLE parameter in the target CDB
- They must have the same endianness
- They must have compatible character sets and national character sets

9.7.10.4.2.1 Setting up auxillary CDB parameters

During a vPDB provisioning into a Linked CDB or a vCDB, the Delphix Engine creates a temporary CDB instance (or Auxiliary CDB) on the target environment to recover the vPDB to a consistent state. This temporary CDB will be automatically deleted (in case of provisioning to a linked CDB or to an existing vCDB) or converted to a vCDB (in case of provisioning to a new vCDB) after the vPDB is provisioned successfully. By default, this temporary CDB is configured to have the same init parameters as the source database. To manage the configuration of the temporary CDB init parameters, you must set up Repository Templates for Oracle Databases (see page 1459) prior to provisioning.

9.7.10.4.3 Provisioning a vPDB to a new or an existing vCDB

When provisioning a vPDB into a new vCDB:

- Delphix will provision a virtual CDB (vCDB) from the source CDB to host the vPDB you are about to create.
- There is no need to set up a repository template, as this provision workflow allows a user to directly specify a VDB configuration template, or set individual database parameters.

9.7.10.4.3.1 Procedure

1. In the **Datasets panel**, select an Oracle PDB dSource or a previously provisioned vPDB.
2. From the **Timeflow** tab, select a snapshot or point in time to provision from.
3. Once the Provision wizard is open, you can either provision with a:
    - Target Linked CDB: Select an existing container database as the provision target CDB from the **Container Database** drop-down menu of CDBs on that environment.
    - Existing vCDB: Select an existing vCDB as the provision target CDB from the **Container Database** drop-down menu of CDBs on that environment. (Supported only for Oracle versions 12.1.0.2 and later.)
    - New vCDB: Select the **Create a New Container Database** checkbox. This will create a new vCDB object in that environment with this new vPDB plugged into it.

4. Click **Next** to advance the left-hand pane to the **Target Configuration** tab, and edit as necessary.
5. The Environment User must have permission to write to the specified Mount Base, as described in Requirements for Oracle Environments and Data (see page 1392). You can also reuse the Delphix toolkit directory, which already exists as the Mount Base, or create a new writable directory in the target environment with the correct permissions and use that as the Mount Base.
    a. Linux and Unix hosts, this mount path must be the full path and not include symlinks.
6. Enter the vPDB Name, Target Group for the vPDB you are about to provision.
7. If you selected to create a new target vCDB, configure the vCDB:
    - Enter the vCDB Name, Database Unique Name, and Database Name for the vCDB you are about to provision.
    - Select the Configure vCDB Parameters checkbox if you want to use a VDB Configuration Template. See Customizing Oracle VDB Configuration Settings (see page 1517)
8. Click **Next** to advance the left-hand pane to the Advanced tab.
9. The available options are vCDB listeners, Auto vCDB Restart, Auto vPDB Restart, File Mapping, Patching, and custom environment variables. For more information, see Customizing VDB File Mappings (see page 1533) and Customizing Oracle VDB Environment Variables (see page 1530)
10. Click **Next** to advance the left-hand pane to the Policies tab.
11. Select the VDB Snapshot policy to be applied to the vPDB. Select a Retention Policy for the vCDB, if you are provisioning a vCDB.
12. Click **Next** to advance the left-hand pane to the Masking tab. Select the Mask this vPDB checkbox if you want to mask, and select the masking job to be applied.
13. Click **Next** to advance the left-hand pane to the Hooks tab, and create any hooks if necessary. For more information, see Hook Scripts for Automation and Customization (see page 1056)
14. Review the provisioning summary. Click **Submit** to proceed with provisioning the vPDB.

> ⓘ If the provisioning fails due to a failure in a hook operation, review the output of the script and correct the error(s). Then you can either refresh the new vPDB or delete it and provision it again. Note that you cannot enable or start the vPDB after a failed provisioning.

9.7.10.5 Provisioning a TDE-enabled vPDB

9.7.10.5.1 Overview

Refer to the Transparent Data Encryption and Delphix[454] article for an overview of TDE and the Continuous Data implementation.

Delphix supports provisioning a vPDB to three different types of a TDE-enabled target container:

- Software keystore based target container database.
- OKV-enabled target container database.
- HSM-enabled target container database.

The following pages will describe in detail the procedure for each of these.

9.7.10.5.2 Prerequisites for provisioning a TDE-enabled vPDB to a Linked CDB

The same prerequisites that apply for provisioning a regular vPDB to a Linked CDB, also apply to a TDE software keystore based, OKV-enabled or HSM-enabled vPDB. Specifically, there must be a target environment that has an Oracle installation compatible with the Oracle installation of the source CDB and the source PDB. The following database requirements are needed to provision the vPDB to a linked CDB. For more information, refer to the Requirements for Oracle Hosts and Databases (see page 1392) section.

Autodiscovery is recommended so that the linked CDB can be found. Otherwise, the linked CDB must be manually discovered before provisioning.

- Linked CDB must be running
- Linked CDB must be in ARCHIVELOG mode
- Linked CDB should be using Block Change Tracking (BCT)
- LogSync must be enabled for the Linked CDB.

9.7.10.5.3 Compatibility requirements for provisioning a TDE-enabled vPDB to a Linked CDB

For provisioning a vPDB into a Linked CDB, the source CDB and the target CDB must meet the following compatibility requirements:

- The value of the COMPATIBLE parameter in the source CDB must be less than or equal to the value of the COMPATIBLE parameter in the target CDB.
- They must have the same endianness.
- They must have compatible character sets and national character sets.
- If TDE is configured using `sqlnet.ora`, the encryption wallet location should be configured on the target host in the standard location as described in the Oracle documentation. If the `TNS_ADMIN` environment variable is being used to specify the directory location where the `sqlnet.ora` is located, `TNS_ADMIN` should point to the default location as indicated above.

454 https://cd.delphix.com/docs/18.0.0.0/transparent-data-encryption-and-delphix

9.7.10.5.3.1 Setting up auxiliary CDB parameters

During a TDE software keystore based, OKV-enabled or HSM-enabled vPDB provisioning into a Linked CDB or a vCDB, the Delphix Continuous Data Engine creates an auxiliary CDB instance (or temporary CDB used during vPDB provisioning) on the target environment to recover the vPDB to a consistent state. This auxiliary CDB will be automatically deleted (in case of provisioning to a linked CDB or to an existing vCDB) or converted to a vCDB (in case of provisioning to a new vCDB) after the vPDB is provisioned successfully. By default, this auxiliary CDB is configured to have the same init parameters as the source database. To manage the configuration of the auxiliary CDB init parameters, you must set up Repository Templates for Oracle Databases (see page 1459) before provisioning.

For more details on additional TDE-related processing that is performed in the Auxiliary CDB, refer to the A closer look at the TDE provisioning (see page 1571) page.

9.7.10.5.4 Provisioning a TDE software keystore based vPDB

Provisioning a TDE software keystore based Virtual Pluggable Database (vPDB) to a TDE software keystore based target container requires specifying a few TDE provisioning parameters using the GUI or CLI, in addition to the vPDB parameters (such as the vPDB name and target container) and the snapshot to provision from. A TDE software keystore based vPDB can be provisioned to either a Linked CDB or a vCDB. It is important to note that the Continuous Data Engine does not support provisioning a TDE software keystore based vPDB from a source snapshot of a dSource or virtual database that is not encrypted at the time of linking.

To initiate the provision, Continuous Data needs the following pieces of information, all of which can be specified in the GUI or CLI. The seven parameters are listed below with some correlated specifics.

9.7.10.5.4.1 1. Parent Database TDE Keystore Location

- **Description**: The path to a keystore on the target host that contains the keys used to encrypt the source database data files.
  - **Required**.
- **CLI parameter**: `source.parentTdeKeystorePath`
- **Notes**
  - The keystore must be accessible on the target host(s) in the specified path.
  - The wallet `ewallet.p12` must exist in the specified path and must be readable by the Oracle user. This is normally a copy of the source database wallet.
  - If the parent database TDE keystore has moved to a different location on the target host after provisioning the vPDB, this path must be updated via the GUI or CLI, as described in Updating the Parent Database TDE Keystore Location (see page 1599), otherwise a subsequent refresh operation on that vPDB will fail.
  - If the master encryption keys of the source database have changed after the vPDB has been provisioned, you must ensure that a copy of the new encryption keys are present in the parent database TDE keystore location, usually by copying the updated parent database TDE keystore wallet from the source database to the parent database TDE keystore location on the target host.

> **Warning**
> Parent Database TDE Keystore may be located on local storage, NFS or ACFS, but cannot be located on an ASM filesystem.

9.7.10.5.4.2 2. Parent Database TDE Keystore Password

- **Description**: Password for the parent database TDE keystore.
  - **Required**.
- **CLI parameter**: `source.parentTdeKeystorePassword`
- **Notes**
  - This parameter must be updated if the password is changed.

9.7.10.5.4.3 3. TDE Secret for exported keys

- **Description**: Secret for the exported keys.
  - **Required**.
- **CLI parameter**: `source.tdeExportedKeyFileSecret`
- **Notes**
  - Oracle requires a password to be set when exporting keys to a keyfile from a keystore. The secret is an alphanumeric string that protects the keys in the file.
  - This parameter represents a new user-specified secret that is used by Continuous Data when exporting keys, and does not need to match any existing keystore password.
  - Once a vPDB is provisioned using this secret, it cannot be changed for the lifetime of the vPDB.
  - This secret is used by Continuous Data during provisioning and subsequent vPDB operations that require exporting the keys to a keyfile.

> **Warning**
> Make sure the TDE Secret for Exported Keys is stored in a secure location for your records. It is only known to you. In the rare event that keys need to be manually extracted from an exported keyfile, this password will be required. Delphix Support cannot assist with manually exporting keys without this password, therefore it should be known or recorded within your organization.

9.7.10.5.4.4 4. Target Keystore Password

- **Description**: Password for the target Linked CDB or existing vCDB keystore.
  - **Required** for **linked** CDB or **existing** vCDB targets.

    - **Not applicable** to **new** vCDB targets.
- **CLI parameter**: `sourceconfig.tdeKeystorePassword`
- **Notes**
    - This parameter must be updated via the GUI or CLI in the Environments page, as described in Adding or Editing the Target keystore Password (see page 1600).
    - This is required when provisioning to an existing Linked CDB or existing vCDB, and must match the password used to open the Linked CDB or existing vCDB keystore.

9.7.10.5.4.5 5. TDE Keystores Root

- **Description**: Path to a directory on the target host under which all Continuous Data related TDE artifacts will be created.
    - **Required** for **cluster** targets.
    - **Optional** for **single instance** targets.
- **CLI parameters**: `host.oracleParameters.tdeKeystoresRootPath`
- **Notes**
    - This includes keystores used by the auxiliary CDB during provisioning and the artifact directories for TDE-enabled vPDBs.
    - This parameter must be updated via the GUI or CLI in the **Environments** page, as described in Adding or Editing the TDE Keystores Root (see page 1601).
    - This is an arbitrary path, which does not need to be referenced by `sqlnet.ora` or `wallet_root`.
    - When provisioning to a single instance target, this will default to `<toolkit path>/tde`. When provisioning to a cluster target, this path must be on shared storage and available to all cluster hosts.
    - The Environment User must have permission to write to this path.

9.7.10.5.4.6 6. Target vCDB TDE Keystore Location

- **Description**: Path of the location on the target host at which Continuous Data will create the keystore during new vCDB provisions.
    - **Required** for **new vCDB** targets.
    - **Not applicable** to **linked CDB** or **existing vCDB** targets.
- **CLI parameters**: `source.targetVcdbTdeKeystorePath`
- **Notes**
    - This path refers to a location on the target host.
    - This path must either not exist or can be an existing empty directory.
    - The Environment User must have permission to write to this location.
    - For Oracle 12.2, the path must match what is specified by `sqlnet.ora`. For higher versions, Continuous Data will set the `wallet_root` parameter to the provided location.

- This parameter must be updated if the keystore location is changed or else future Continuous Data operations may fail.
- When provisioning to a cluster target, this path must be on shared storage and available to all cluster hosts.

9.7.10.5.4.7 7. Target vCDB TDE Keystore Password

- **Description**: Password for the new vCDB keystore.
  - **Required** for **new vCDB** targets.
  - **Not applicable** to **linked CDB** or **existing vCDB** targets.
- **CLI parameters**: `virtualCdb.sourceConfig.tdeKeystorePassword`
- **Notes**
  - This password is created during provisioning. It does not need to match any existing keystore password.
  - If this password is changed, it must be updated via the GUI or CLI in the **Environments** page, as described in Adding or Editing the Target Keystore Password (see page 1600).

9.7.10.5.4.8 Provisioning a TDE-enabled vPDB

When provisioning a TDE-enabled vPDB into a Linked CDB or an existing vCDB:

- Continuous Data requires that the Linked CDB or vCDB already be present on the target host with all the encryption keys set up correctly.

When provisioning a TDE-enabled vPDB into a new vCDB:

- Continuous Data will provision a vCDB from the source CDB to host the vPDB you are about to create. The newly-created vCDB will be configured with TDE enabled.
- There is no need to set up a repository template, as this provision workflow allows a user to directly specify a VDB configuration template, or set individual database parameters.

Procedure

1. If you're provisioning to a Linked CDB for the first time, add the Target Keystore Password following the steps described in Adding or Editing the Target Keystore Password (see page 1600).

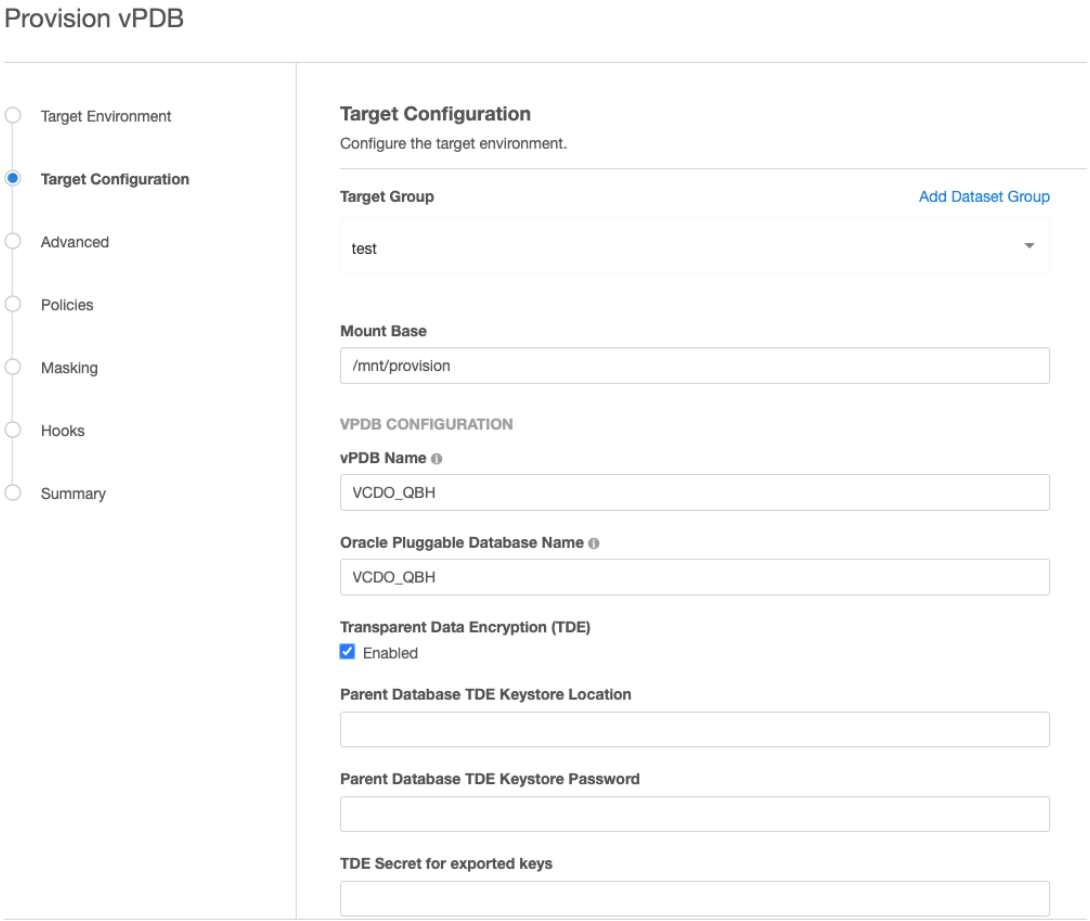
> To validate that the parent database TDE keystore password is correct, refer to the Validating the Keystore Password (see page 0) section.

2. Ensure that the parent database TDE keystore location on the target host contains a keystore that includes the encryption keys from the source database and is readable by the Oracle User used for provisioning.

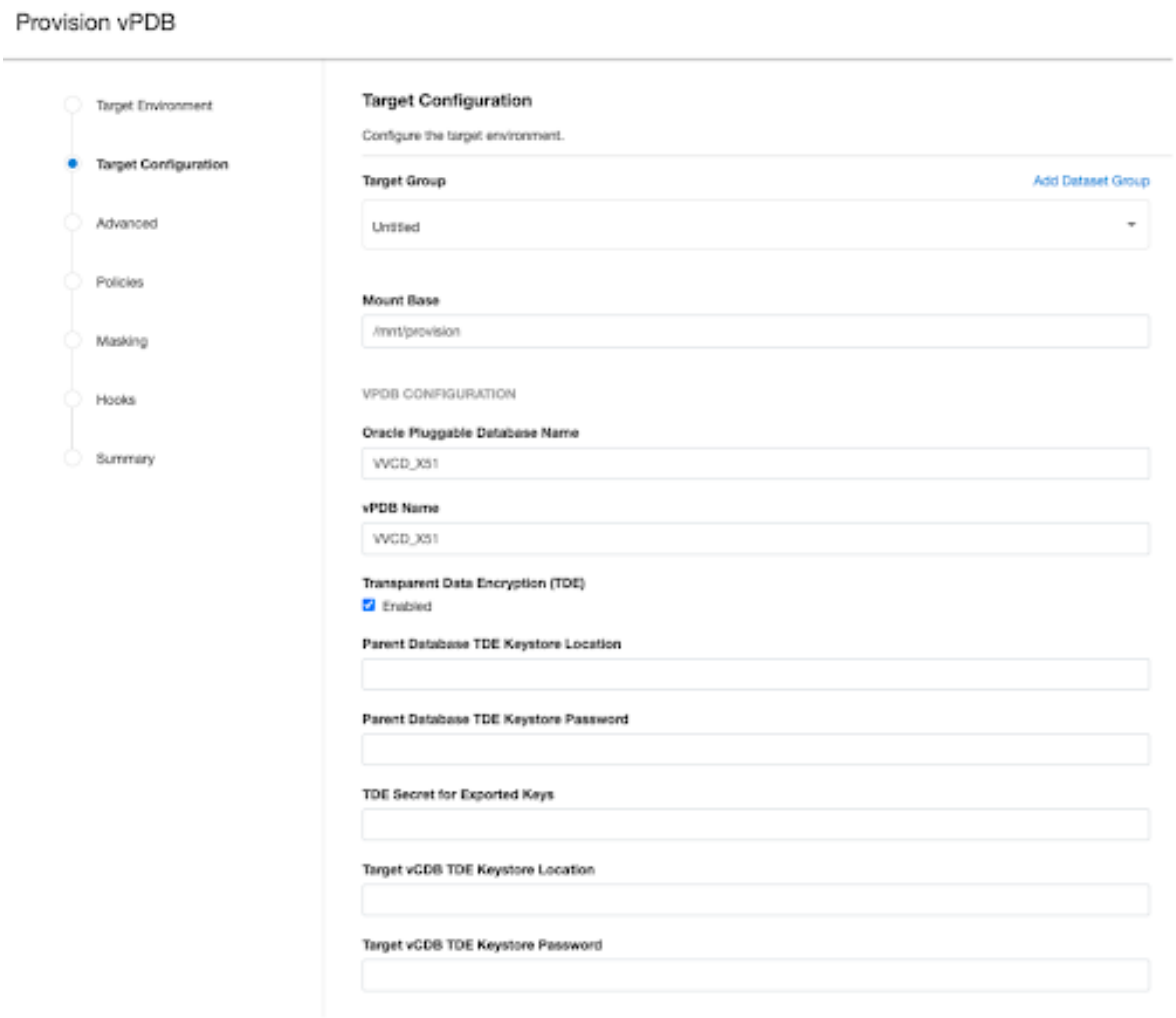
3. If provisioning to a cluster target, ensure that the keystores root directory path is set correctly following the steps described in Adding or Editing the TDE Keystores Root (see page 1601).
4. In the **Datasets panel**, select an Oracle TDE-enabled PDB dSource or a previously provisioned TDE-enabled vPDB.
5. From the **Timeflow** tab, select a snapshot or point in time to provision from.
6. Once the Provision wizard is open, you can either provision with a:
   a. Target Linked CDB: Select an existing container database as the provision target CDB from the **Container Database** drop-down menu of CDBs on that environment.
   b. Existing vCDB: Select an existing vCDB as the provision target CDB from the **Container Database** drop-down menu of CDBs on that environment. (Supported only for Oracle versions 12.1.0.2 and later.)
   c. New vCDB: Select the **Create a New Container Database** checkbox. This will create a new vCDB object in that environment with this new vPDB plugged into it.
7. Click **Next** to advance the left-hand pane to the **Target Configuration** tab, and edit as necessary.
8. Enter the target Group for the vPDB you are about to provision.
9. The Environment User must have permission to write to the specified Mount Base, as described in Requirements for Oracle Environments and Data[455].
   You can also reuse the Delphix toolkit directory, which already exists as the Mount Base, or create a new writable directory in the target environment with the correct permissions and use that as the Mount Base.
   a. Linux and Unix hosts, this mount path must be the full path and not include symlinks.
10. Enter the vPDB Name and the Oracle Pluggable Database Name.
11. Click on the "**Transparent Data Encryption (TDE) Enabled"** checkbox. The following three fields need to be specified during the vPDB provision.
    a. **Parent Database TDE Keystore Location** - Specify the path to a keystore that contains the keys used to encrypt the source database datafiles. As noted earlier, the wallet `ewallet.p12` must exist in the specified path and must be readable by the Oracle user.

       Warning: Parent Database TDE keystore may be located on local storage, NFS or ACFS, but **cannot** be located on an ASM filesystem.
    b. **Parent Database TDE Keystore Password** - Specify the password for the parent database TDE keystore.
    c. **TDE Secret for exported keys** - Specify the password for the exported keys. **Warning:** Make sure the TDE Secret for Exported Keys is stored in a secure location for your records. It is only known to you. In the rare event that keys need to be manually extracted from an exported keyfile, this password will be required. Delphix Support cannot assist with manually exporting keys without this password, therefore it should be known or recorded within your organization.

455 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/16423380



12. If the target CDB is a vCDB, two additional necessary fields need to be specified - “Target vCDB TDE Keystore Location” and “Target vCDB TDE Keystore Password”. Please note that the Environment User must have permission to write to the “Target vCDB TDE Keystore Location” otherwise the provisioning will fail. Refer to the Provisioning parameters for a TDE-enabled vPDB (see page 0) section for important information on these two fields.



13. If you selected to create a new target vCDB, configure the vCDB:
    a. Enter the vCDB Name, Database Unique Name, and Database Name for the vCDB you are about to provision.
    b. Select the Configure vCDB Parameters checkbox if you want to use a VDB Configuration Template. See Customizing Oracle VDB Configuration Settings. (see page 1517)
14. Click **Next** to advance the left-hand pane to the Advanced tab.
15. The available options are vCDB Listeners, Auto vCDB Restart, Auto vPDB Restart, File Mapping, Patching and custom environment variables. For more information, see Customizing VDB File Mappings (see page 1533) and Customizing Oracle VDB Environment Variables. (see page 1530)
16. Click **Next** to advance the left-hand pane to the Policies tab.
17. Select the VDB Snapshot policy to be applied to the vPDB.
    Select a Retention Policy for the vCDB, if you are provisioning a vCDB.
18. Click **Next** to advance the left-hand pane to the Masking tab.
    Select the Mask this vPDB checkbox if you want to mask, and select the masking job to be applied.

19. Click **Next** to advance the left-hand pane to the Hooks tab, and create any hooks if necessary. For more information, see Hook Scripts for Automation and Customization.[456]
20. Review the provisioning summary. Confirm all the fields are correct. Click **Submit** to proceed with provisioning the vPDB.

9.7.10.5.4.9 TDE Keystores Root and Artifact Directory

The artifact directory on the target host stores the exported keyfiles used during the workflows for TDE-enabled vPDBs. It is located under the keystores root, in the directory `oracle_tde_keystores`. Each TDE-enabled vPDB will have its own directory within the `oracle_tde_keystores` directory, identified by the vPDB name, group name, and a unique identifier, separated by an underscore. If the keystores root directory is not specified, then it defaults to the toolkit directory path.

For example, if the keystores root directory is `/work` (or keystores root is not specified, and the toolkit directory is `/toolkit`), the artifact directory for the vPDB `tde_vpdb` in the group `Encrypted` could be

`/toolkit/oracle_tde_keystores/tde_vpdb_Encrypted_ce7a47e6-8860-4398-bab0-cf0233fc5e3c`

Within the artifact directory, there is a subdirectory `exported_keys` which contains within it the exported keyfiles for each timeflow associated with that vPDB. Each time an export is performed, a new exported keyfile is generated with a timestamp. The contents of the artifact directory may change in future releases, but the path to the artifact directory and the naming convention is not anticipated to change.

As the default keystores root directory is at the same level as the toolkit directory, it will not be overwritten if a host is refreshed through the Continuous Data Engine and the toolkit updated. It is the customer's responsibility to backup the keystores root directory and ensure that the contents are not lost, as a disk failure could prevent a TDE-enabled vPDB from being accessed. The Continuous Data Engine never keeps a copy of the keystores or exported keyfiles on Continuous Data storage. Thus it is recommended that the keystores root directory be on a disk which is regularly backed up.

The artifact directory is not removed when a TDE-enabled vPDB is deleted; the customer can remove it after confirming that the vPDB has been removed (including from any replicated Delphix Engines).

> The Continuous Data Engine does not keep a copy of the keystores or exported keyfiles on Delphix storage and thus a disk failure could prevent a TDE-enabled vPDB from being accessed. Therefore, Delphix highly recommends that the TDE keystores root directory be backed up at regular intervals.

456 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/6849297/Hook+scripts+for+automation+and+customization

9.7.10.5.4.10 Other operations on a TDE-enabled vPDB

Once a TDE-enabled vPDB is provisioned, it can be used the same as a non-TDE-enabled vPDB within Continuous Data, with the exception of migrate (see page 1613) which is separately covered below. There are however a few caveats or behavioral differences as follows:

- Refreshing a TDE-enabled vPDB (see page 1589) will use the parent keystore for the recovery. If the parent PDB's master keys are changed, the user will need to update the parent keystore with the new keys, for example by re-copying the parent PDB's `ewallet.p12` file to the parent keystore location on the target host. Similarly, if the location or password to the parent keystore has changed then they should be updated before the refresh.
- Rewinding a TDE-enabled vPDB (see page 1589) will use the target keystore for the recovery. If the vPDB is rekeyed after it is provisioned, then the rekey will update the target keystore, so it does not need to be updated in Continuous Data.
- For a single vPDB in a vCDB, if the Target vCDB TDE keystore location is changed, the new path must be updated in the Continuous Data engine before refresh or rewind.
- Disabling a TDE-enabled vPDB will result in the keys being exported to an exported keyfile in the artifact directory, to be used for a subsequent enable. Refresh and rewind operations will first disable the existing vPDB, so those will also result in a new exported keyfile in the artifact directory.
- Provisioning a second-generation vPDB (vvPDB) from a TDE-enabled vPDB is done in the same manner as a first-generation vPDB, by specifying the TDE parameters during provision. The parent database TDE keystore location for the vPDB can be specified as the parent database TDE keystore location for the vvPDB.

If a vPDB is moved to a different host (either through the migrate workflow or an enable after a failover, then the artifact directory will need to be copied to the new target host. See Migrating a TDE-enabled vPDB (see page 1613) for details on the manual steps needed for migration.

9.7.10.5.5 Provisioning a TDE OKV-enabled vPDB

Provisioning a vPDB to a TDE OKV-enabled target container requires specifying several TDE provisioning parameters either using GUI or CLI. Additionally, vPDB parameters, such as the vPDB name, target container, and the snapshot to provision from must be provided. A vPDB can be provisioned to either a TDE OKV-enabled Linked CDB or a vCDB.

It is important to note that the Delphix Continuous Data Engine does not support provisioning a vPDB from a source snapshot of a dSource or virtual database that is not encrypted or encrypted using a software wallet or Hardware Security Module at the time of linking.

9.7.10.5.5.1 Prerequisites for provisioning a vPDB into a TDE OKV-enabled CDB

Before initiating the provision, Delphix Continuous Data Engine needs the following:

1. **OKV Home path**
    - **Description**: The installation path of the Oracle Key Vault client library, particularly the `okvclient.jar` file, on the target database node will be automatically discovered if it's set for the environment user. (Required)

- **CLI parameter**: `host.oracleHostParameters.tdeOkvHomePath`
- **Notes**
  - This parameter must be updated via the GUI or CLI in the Environments page as described in Adding or Editing the OKV Home (see page 1598).
  - In an Oracle Real Application Clusters (RAC) environment, you need to update this attribute for each Oracle RAC node. This is necessary because each RAC node is enrolled and provisioned as an endpoint, and a separate okvclient.jar will be installed on each RAC node.

2. **TDE External Key Manager Credential**
   - **Description**: The password for the endpoint provided during the installation of the Oracle Key Vault client library (i.e., `okvclient.jar`) on the target database node.
     - **Required** for linked **CDB** or existing **vCDB** targets.
     - **Not applicable** to **new** vCDB targets.
   - **CLI parameter**: `host.oracleHostParameters.tdeExternalKeyManagerCredential` OR `sourceconfig.tdeKeystorePassword`
   - TDE External Key Manager Credential provided at the database level takes **precedence over** that provided at the host level.
   - **Notes**
     - This parameter must be updated via the GUI or CLI in the Environments page as described in Adding or Editing TDE External Key Manager Credential (see page 1596).
     - This is required when provisioning to an existing Linked CDB or existing vCDB, and must match the password used to open the Linked CDB or existing vCDB keystore.
     - This parameter must be updated via GUI or CLI whenever the endpoint password for the target database is rotated.
     - In an Oracle Real Application Clusters (RAC) environment, you need to update this attribute for each Oracle RAC node. This is necessary because each RAC node is enrolled and provisioned as an endpoint, and a separate okvclient.jar will be installed on each RAC node.

3. **Granting target endpoint access to a parent dSource master encryption keys**

   To provision a vPDB, access to the master encryption key of the parent dSource is required. In the case of Oracle Key Vault, ensure that the target endpoint has at least **Read Only** access to the master encryption key of the parent dSource.

   If the same OKV instance is used for both the parent and target endpoints, Oracle Key Vault virtual wallets can be employed. These wallets control the security object access mechanism among users, groups, and endpoints. For instructions on setting up this access, refer to the Oracle Key Vault documentation[457].

457 https://docs.oracle.com/en/database/oracle/key-vault/21.4/okvag/okv_manage_endpoints.html#GUID-44D05C48-1536-4B35-BD26-D6A19E5A406B

However, if different OKV instances are used for the parent and target endpoints, the master encryption key of the parent dSource must be uploaded into the target endpoint. For instructions on downloading and uploading the master encryption keys, refer to the Oracle Key Vault Utility documentation[458].

9.7.10.5.5.2 Provisioning parameters for a TDE OKV-enabled vPDB

To initiate the provision, Delphix Continuous Data Engine needs the following parameters(all of which can be specified in the GUI or CLI).

1. **TDE Encryption Secret**
   - **Description**: Encryption Secret for the pluggable database while executing unplug operation. (Required)
   - **CLI parameter**: `source.tdeExportedKeyFileSecret`
   - **Notes**
     - Oracle requires a transport secret to be set when executing the unplug operation on a TDE-enabled vPDB.
     - This parameter represents a new user-specified secret that is used by Continuous Data when unplugging vPDB, and does not need to match any existing keystore password.
     - Once a vPDB is provisioned using this secret, it cannot be changed for the lifetime of the vPDB.
     - This secret is used by Continuous Data during provisioning and subsequent vPDB operations that require unplugging or plugging of vPDB.
2. **TDE Keystores Root**
   - **Description**: Path to a directory on the target host under which all Continuous Data related TDE artifacts will be created.
     - **Required** for **cluster** targets.
     - **Optional** for **single instance** targets.
   - **CLI parameter**: `host.oracleParameters.tdeKeystoresRootPath`
   - **Notes**
     - This includes keystores used by the auxiliary CDB during provisioning and the artifact directories for TDE-enabled vPDBs.
     - This parameter must be updated via the GUI or CLI in the Environments page, as described in Adding or Editing the TDE Keystores Root (see page 1601).
     - This is an arbitrary path, which does not need to be referenced by sqlnet.ora or wallet_root.

458 https://docs.oracle.com/en/database/oracle/key-vault/21.7/okvag/okvutil_endpoint_utility_reference.html#GUID-A3E85D53-86C6-4B17-86E9-037AB61AFD40

    - When provisioning to a single instance target, this will default to <toolkit path>/tde. When provisioning to a cluster target, this path must be on **shared storage** and available to all cluster hosts.
    - The Delphix User must have permission to write to this path.

3. **Target vCDB TDE External Key Manager Credential**
    - **Description**: The Target endpoint password for the new vCDB keystore.
        - **Required** for **new vCDB** targets.
        - **Not applicable** to **linked CDB** or **existing vCDB** targets.
    - **CLI parameter**: `virtualCdb.sourceConfig.tdeKeystorePassword`
    - **Notes**
        - This is the same password that was provided during the installation of the Oracle Key Vault client library (`okvclient.jar`) on the target database endpoint.
        - If this password is changed, it must be updated via the GUI or CLI on the Environments page, as described in Adding or Editing TDE External Key Manager Credential (see page 1596).

9.7.10.5.5.3 Provisioning a TDE OKV-enabled vPDB

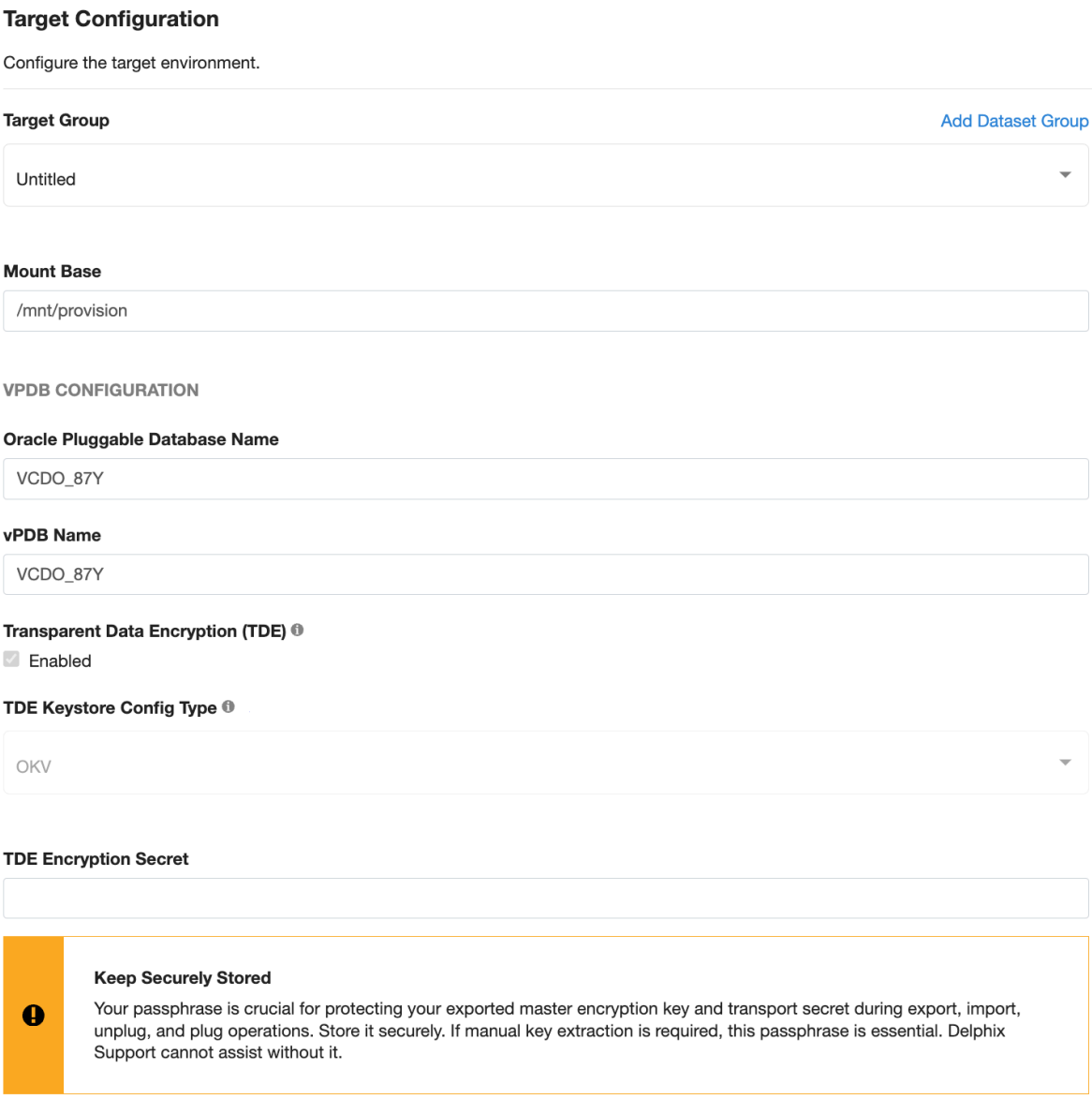
1. If you are provisioning to a linked CDB or a new vCDB for the first time, add the OKV Home by following the steps listed in Adding or Editing the OKV Home (see page 1598). This is a one-time activity. Once you have updated the OKV Home, you can provision any number of vPDBs in this environment. If you are provisioning to a RAC cluster, make sure to update it for each database node.
2. If you're provisioning to a Linked CDB for the first time, add the TDE External Key Manager Credential for the target by following the steps listed in Adding or Editing TDE External Key Manager Credential (see page 1596). If you are provisioning to a RAC cluster, make sure to update it for each database node. To validate the correctness of the TDE External Key Manager Credential for the target endpoint, refer to the Oracle Key Vault okvutil utility documentation[459].
3. If you are provisioning to a Linked CDB or vCDB for the first time, ensure that you grant at least **read access** to the parent dSource endpoint master encryption key for the target endpoint.
4. If provisioning to a RAC cluster target, ensure that the keystores root directory path is set correctly following the steps described in Adding or Editing the TDE Keystores Root (see page 1601).
5. In the **Datasets panel**, select an Oracle TDE OKV-enabled PDB dSource or a previously provisioned TDE OKV-enabled vPDB.
6. From the **Timeflow** tab, select a snapshot or point in time to provision from.
7. Once the Provision wizard is open, you can either provision with a:

459 https://docs.oracle.com/en/database/oracle/key-vault/21.7/okvag/okvutil_endpoint_utility_reference.html#GUID-C7A842A4-99B7-42C5-89CF-B7C775B24142

   a. Target Linked CDB: Select an existing container database as the provision target CDB from the **Container Database** drop-down menu of CDBs on that environment.
   b. Existing vCDB: Select an existing vCDB as the provision target CDB from the **Container Database** drop-down menu of CDBs on that environment.
   c. New vCDB: Select the **Create a New Container Database** checkbox. This will create a new vCDB object in that environment with this new vPDB plugged into it.

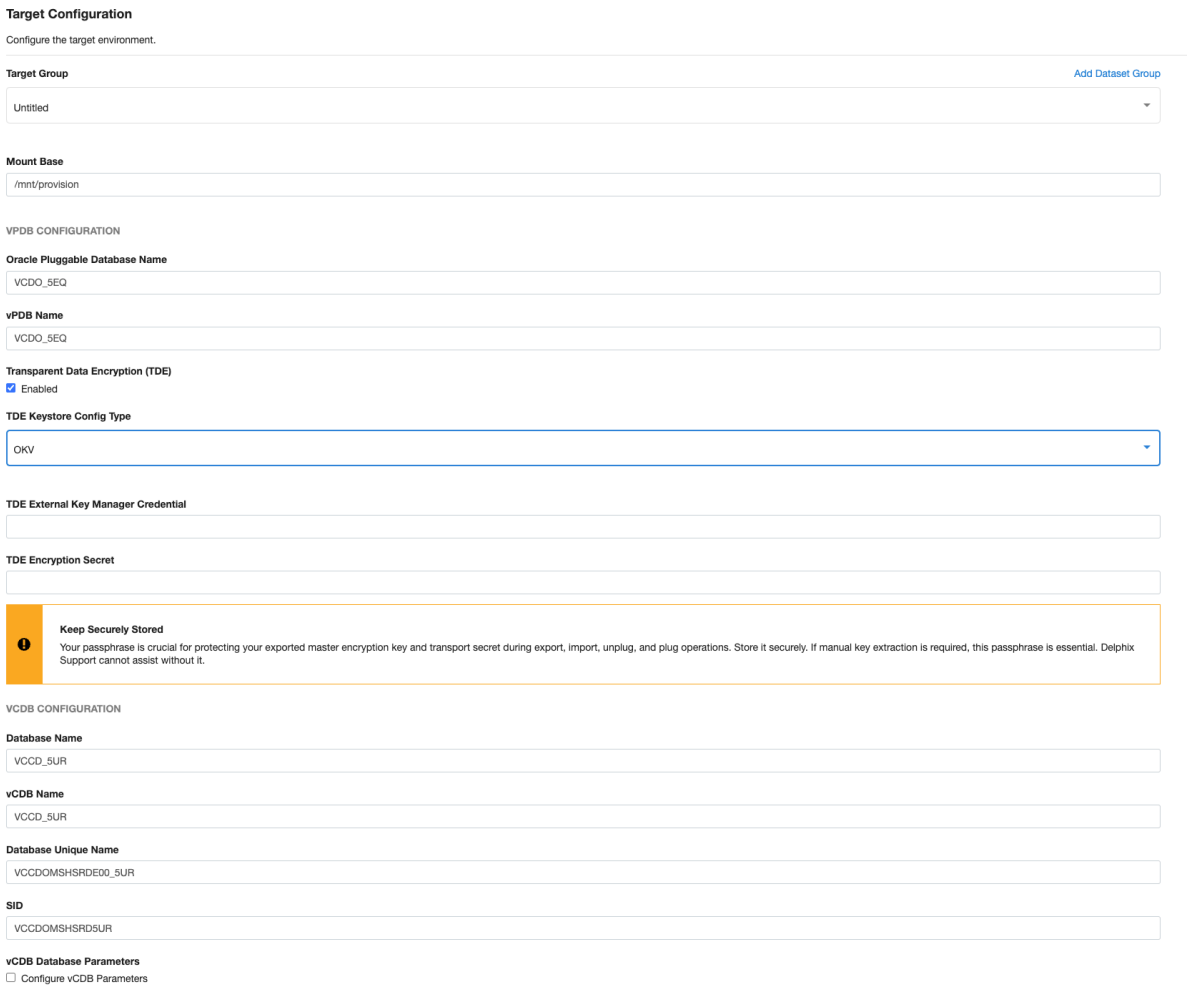
8. Click **Next** to advance the left-hand pane to the **Target Configuration** tab, and edit as necessary.
9. Enter the target Group for the vPDB you are about to provision.
10. The Environment User must have permission to write to the specified Mount Base, as described in Requirements for Oracle Environments and Data[460].
    You can also reuse the Delphix Continuous Data Engine toolkit directory, which already exists as the Mount Base, or create a new writable directory in the target environment with the correct permissions and use that as the Mount Base.
    a. Linux and Unix hosts, this mount path must be the full path and not include symlinks.
11. Enter the vPDB Name and the Oracle Pluggable Database Name.
12. When provisioning to a Linked CDB or existing vCDB, the '**Transparent Data Encryption (TDE) Enabled**' checkbox is automatically **checked**, and the '**TDE Keystore Config Type**' dropdown is populated with '**OKV**'.
13. **TDE Encryption Secret** - Specify the passphrase which is required during unplug/plug operation of the vPDB. **Warning:** Make sure the TDE Encryption Secret is stored in a secure location for your records. It is only known to you. In the rare event that vPDB needs to be manually plugged from an unplugged vPDB, this passphrase will be required. Delphix Support cannot assist with manually plugging vPDB without this passphrase, therefore it should be known or recorded within your organization.

460 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/16423380



14. When provisioning to a new vCDB, click on the "**Transparent Data Encryption (TDE) Enabled"** checkbox and select **'OKV'** from the "**TDE Keystore Config Type**" dropdown. Two additional necessary fields need to be specified - "**TDE External Key Manager Credential**" and "**TDE Encryption Secret**".
15. **TDE External Key Manager Credential** - Specify the password for the target endpoint provided during the installation of the Oracle Key Vault client library (i.e., `okvclient.jar`) on the target database node.
16. **TDE Encryption Secret** - Specify the passphrase which is required during unplug/plug operation of the vPDB. **Warning:** Make sure the TDE Encryption Secret is stored in a secure location for your records. It is only known to you. In the rare event that vPDB needs to be manually plugged from an unplugged

vPDB, this passphrase will be required. Delphix Support cannot assist with manually plugging vPDB without this passphrase, therefore it should be known or recorded within your organization.



17. If you selected to create a new target vCDB, configure the vCDB:
    a. Enter the vCDB Name, Database Unique Name, and Database Name for the vCDB you are about to provision.
    b. Select the Configure vCDB Parameters checkbox if you want to use a VDB Configuration Template. See Customizing Oracle VDB Configuration Settings. (see page 1517)
18. Click **Next** to advance the left-hand pane to the Advanced tab.
19. The available options are vCDB Listeners, Auto vCDB Restart, Auto vPDB Restart, File Mapping, Patching and custom environment variables. For more information, see Customizing VDB File Mappings (see page 1533) and Customizing Oracle VDB Environment Variables. (see page 1530)
20. Click **Next** to advance the left-hand pane to the Policies tab.
21. Select the VDB Snapshot policy to be applied to the vPDB.
    Select a Retention Policy for the vCDB, if you are provisioning a vCDB.

22. Click **Next** to advance the left-hand pane to the Masking tab.
    Select the Mask this vPDB checkbox if you want to mask, and select the masking job to be applied.
23. Click **Next** to advance the left-hand pane to the Hooks tab, and create any hooks if necessary. For more information, see Hook Scripts for Automation and Customization. (see page 1056)
24. Review the provisioning summary. Confirm all the fields are correct. Click **Submit** to proceed with provisioning the vPDB.

9.7.10.5.6 Provisioning a TDE HSM-enabled vPDB

Provisioning a TDE HSM-enabled Virtual Pluggable Database (vPDB) to a TDE HSM-enabled target container requires specifying several TDE provisioning parameters using either the GUI or CLI. Additionally, vPDB parameters, such as the vPDB name, target container, and the snapshot to provision from, must be provided. A TDE HSM-enabled vPDB can be provisioned to either a Linked CDB or a vCDB.

It is important to note that the Delphix Continuous Data Engine does not support provisioning a TDE HSM-enabled vPDB from a source snapshot of a dSource or virtual database that is not encrypted or encrypted using a software wallet or OKV at the time of linking.

9.7.10.5.6.1 Prerequisites for provisioning a TDE HSM-enabled vPDB

Before initiating the provision, Delphix Continuous Data Engine needs the following pieces of information:

1. **TDE External Key Manager Credential**
    - **Description**: The credential of the external key manager which is managing master encryption keys of target databases.
        - **Required** for linked **CDB** or existing **vCDB** targets**.**
        - **Not applicable** to **new** vCDB targets**.**
    - **CLI parameter**: `host.oracleHostParameters.tdeExternalKeyManagerCredential` OR `sourceconfig.tdeKeystorePassword`
        - TDE External Key Manager Credential provided at the database level takes **precedence over** that provided at the host level.
    - **Notes**
        - This parameter must be updated via the GUI or CLI in the Environments page as specified in Adding or Editing TDE External Key Manager Credential (see page 1596).
        - This is required when provisioning to an existing Linked CDB or existing vCDB, and must match the password used to open the Linked CDB or existing vCDB keystore.
        - This parameter must be updated via GUI or CLI whenever the endpoint password for the target database is rotated.
        - In the context of **CipherTrust Manager,** the external key manager credential format is specified as follows: **"domain:username:password"** For the default domain, the format simplifies to **"username:password"**.

2. **Granting target domain access to a parent dSource master encryption keys**

   To provision a vPDB, access to the master encryption key of the parent dSource is required. In the case of Thales CipherTrust Manager, ensure that the target domain has at least **Read Only** access to the master encryption key of the parent dSource domain.

   If the same CipherTrust domain or subdomain is used for both the parent and target endpoints, we don't need to set up the access. Same domain security objects are automatically accessible.

   However, if different CipherTrust domains or subdomains are used for the parent and target endpoints, the master encryption key of the parent dSource must be accessible into the target domain or subdomain. For instructions on setting up access, refer to the CipherTrust Manager Documentation[461].

9.7.10.5.6.2 Provisioning parameters for a TDE HSM-enabled vPDB

1. **TDE Encryption Secret**
   - **Description**: Encryption Secret for the pluggable database while executing unplug operation. (Required)
   - **CLI parameter**: `source.tdeExportedKeyFileSecret`
   - **Notes**
     - Oracle requires a transport secret to be set when executing the unplug operation on a TDE-enabled vPDB.
     - This parameter represents a new user-specified secret that is used by Continuous Data when unplugging vPDB, and does not need to match any existing keystore password.
     - Once a vPDB is provisioned using this secret, it cannot be changed for the lifetime of the vPDB.
     - This secret is used by the Delphix Continuous Data Engine during provisioning and subsequent vPDB operations that require unplugging or plugging of vPDB.
2. **TDE Keystores Root**
   - **Description**: Path to a directory on the target host under which all Continuous Data related TDE artifacts will be created.
     - **Required** for **cluster** targets.
     - **Optional** for **single instance** targets.
   - **CLI parameter**: `host.oracleParameters.tdeKeystoresRootPath`
   - **Notes**
     - This includes keystores used by the auxiliary CDB during provisioning and the artifact directories for TDE-enabled vPDBs.
     - This parameter must be updated via the GUI or CLI in the Environments page, as specified in Adding or Editing the TDE Keystores Root (see page 1601).

461 https://www.thalesdocs.com/ctp/cm/2.14/admin/cm_admin/abac-permissions/domains/index.html

    - This is an arbitrary path, which does not need to be referenced by `sqlnet.ora` or `WALLET_ROOT`.
    - When provisioning to a single instance target, this will default to `<toolkit path>/tde`. When provisioning to a cluster target, this path must be on **shared storage** and available to all cluster hosts.
    - The Delphix Continuous Data Engine User must have permission to write to this path.
3. **Target vCDB TDE External Key Manager Credential**
    - **Description**: The credential of the external key manager which is managing master encryption keys of target databases.
        - **Required** for **new vCDB** targets.
        - **Not applicable** to **linked CDB** or **existing vCDB** targets.
    - **CLI parameter**: `virtualCdb.sourceConfig.tdeKeystorePassword`
    - **Notes**
        - If this password is changed, it must be updated via the GUI or CLI in the Environments page, as described in Adding or Editing TDE External Key Manager Credential (see page 1596).
4. **Target vCDB Autologin Wallet Location**
    - **Description**: Path of the location on the target host at which Continuous Data will create the auto-login keystore during new vCDB provisions.
        - **Required** for **new vCDB** targets.
        - **Not applicable** to **linked CDB** or **existing vCDB** targets.
    - **CLI parameter**: `source.targetVcdbTdeKeystorePath`
    - **Notes**
        - This path refers to a location on the target host.
        - This path must either be an existing auto-login location or can be an existing empty directory.
        - The Environment User must have permission to write to this location.
        - Delphix Continuous Data Engine will set the `WALLET_ROOT` parameter to the provided location.
        - This parameter must be updated if the keystore location is changed or else future Delphix Continuous Data Engine operations may fail.
        - When provisioning to a cluster target, this path must be on shared storage and available to all cluster hosts.

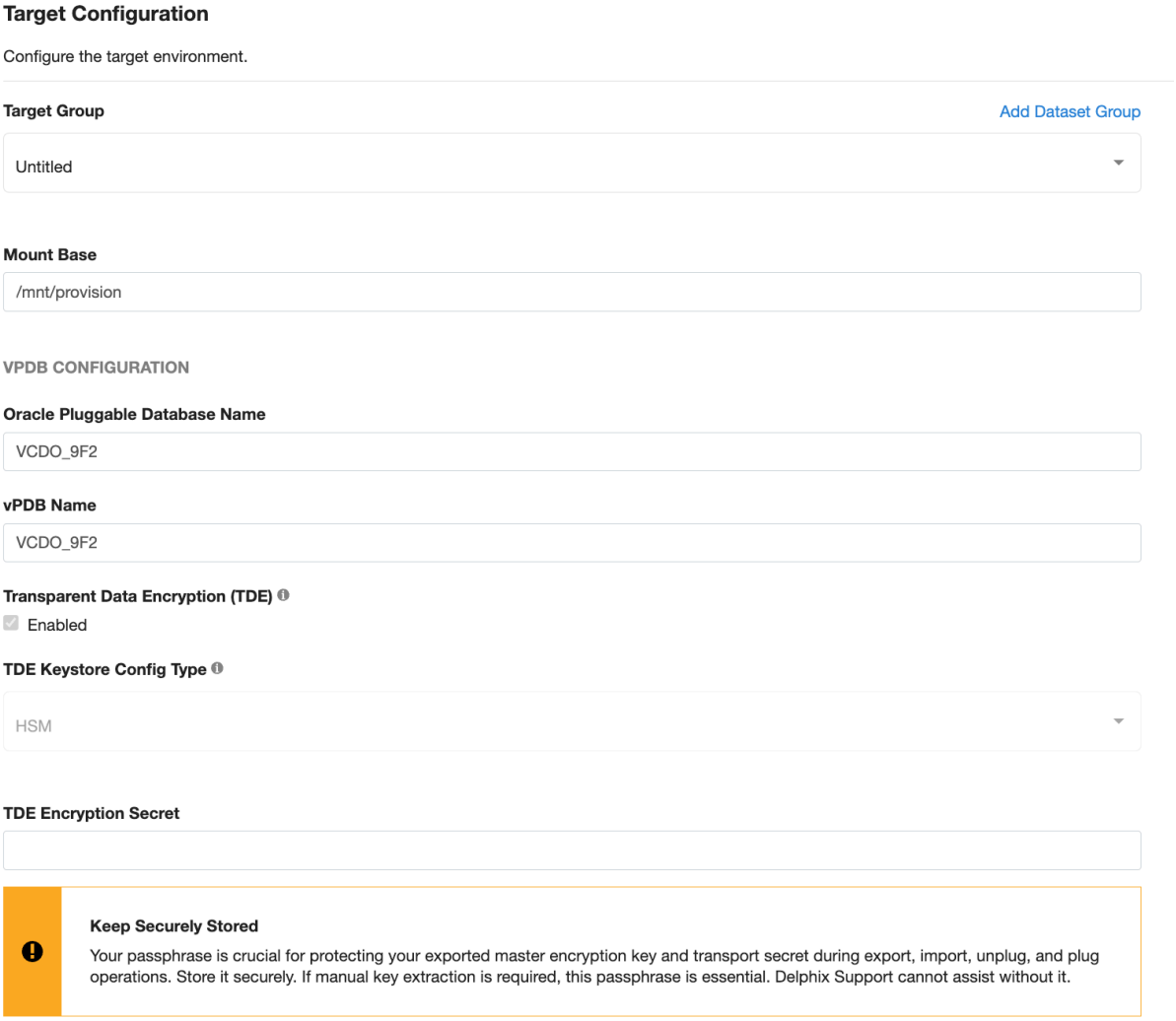
9.7.10.5.6.3 Provisioning a TDE HSM-enabled vPDB

1. If you're provisioning to a Linked CDB for the first time, add the TDE External Key Manager Credential for the target by following the steps listed in Adding or Editing TDE External Key Manager Credential (see page 1596). If you are provisioning to a RAC cluster, make sure to update it for each database node.
2. If you are provisioning to a Linked CDB or vCDB for the first time, ensure that you grant **read access** to the parent dSource master encryption key for the target. Please refer to the 'Granting Target Domain Access to a Parent dSource master encryption keys' section.
3. If provisioning to a RAC cluster target, ensure that the keystores root directory path is set correctly following the steps listed in Adding or Editing the TDE Keystores Root (see page 1601).
4. In the Datasets panel, select an Oracle TDE HSM-enabled PDB dSource or a previously provisioned TDE HSM-enabled vPDB.
5. From the **Timeflow** tab, select a snapshot or point in time to provision from.
6. Once the Provision wizard is open, you can either provision with a:
   a. Target Linked CDB: Select an existing container database as the provision target CDB from the **Container Database** drop-down menu of CDBs on that environment.
   b. Existing vCDB: Select an existing vCDB as the provision target CDB from the **Container Database** drop-down menu of CDBs on that environment.
   c. New vCDB: Select the **Create a New Container Database** checkbox. This will create a new vCDB object in that environment with this new vPDB plugged into it.
7. Click **Next** to advance the left-hand pane to the **Target Configuration** tab, and edit as necessary.
8. Enter the target Group for the vPDB you are about to provision.
9. The Environment User must have permission to write to the specified Mount Base, as described in Requirements for Oracle Environments and Data[462].
   You can also reuse the Delphix Continuous Data Engine toolkit directory, which already exists as the mount base, or create a new writable directory in the target environment with the correct permissions and use that as the mount base.
   a. Linux and Unix hosts, this mount path must be the full path and not include symlinks.
10. Enter the vPDB Name and the Oracle Pluggable Database Name.
11. When provisioning to a Linked CDB or existing vCDB, the '**Transparent Data Encryption (TDE) Enabled**' checkbox is automatically **checked**, and the '**TDE Keystore Config Type**' dropdown is populated with '**HSM**'.
12. **TDE Encryption Secret** - Specify the passphrase that is required during unplug/plug operation of the vPDB.

    Warning: Make sure the TDE Encryption Secret is stored in a secure location for your records. It is only known to you. In the rare event that vPDB needs to be manually plugged from an unplugged vPDB, this passphrase will be required. Delphix Support cannot assist with manually plugging vPDB

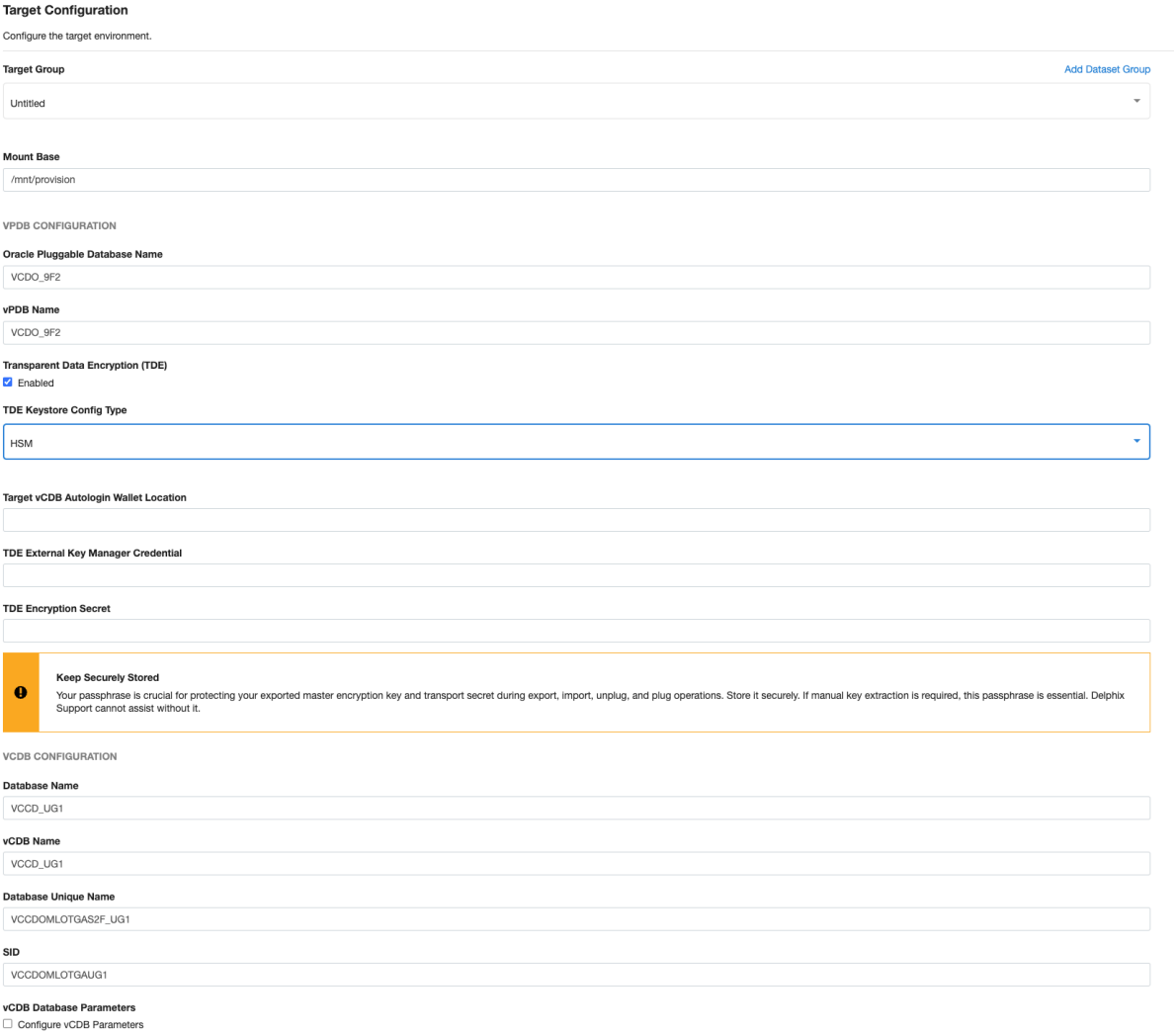
462 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/16423380

without this passphrase, therefore it should be known or recorded within your organization.



13. When provisioning to a new vCDB, click on the **"Transparent Data Encryption (TDE) Enabled"** checkbox and select **'HSM'** from the **"TDE Keystore Config Type"** dropdown. Three additional necessary fields need to be specified - **"Target vCDB Autologin Wallet Location"**, **"TDE External Key Manager Credential"** and **"TDE Encryption Secret"**.
14. **Target vCDB Autologin Wallet Location -** Specify the location on the target host at which Continuous Data will create the auto-login keystore during new vCDB provisions. Refer to the "Provisioning parameters for a TDE with Hardware Security Module (HSM)-enabled vPDB" section for information on this field.
15. **TDE External Key Manager Credential** - Specify the credential of the external key manager which is managing master encryption keys of target databases. Refer to the "Provisioning parameters for a TDE with Hardware Security Module (HSM)-enabled vPDB" section for information on this field.
16. **TDE Encryption Secret** - Specify the passphrase which is required during unplug/plug operation of the vPDB.

Warning: Make sure the TDE Encryption Secret is stored in a secure location for your records. It is only known to you. In the rare event that vPDB needs to be manually plugged from an unplugged vPDB, this passphrase will be required. Delphix Support cannot assist with manually plugging vPDB without this passphrase, therefore it should be known or recorded within your organization.



17. If you selected to create a new target vCDB, configure the vCDB:
    a. Enter the vCDB Name, Database Unique Name, and Database Name for the vCDB you are about to provision.
    b. Select the Configure vCDB Parameters checkbox if you want to use a VDB Configuration Template. See Customizing Oracle VDB Configuration Settings. (see page 1517)
18. Click **Next** to advance the left-hand pane to the Advanced tab.
19. The available options are vCDB Listeners, Auto vCDB Restart, Auto vPDB Restart, File Mapping, Patching and custom environment variables. For more information, see Customizing VDB File Mappings (see page 1533) and Customizing Oracle VDB Environment Variables. (see page 1530)
20. Click **Next** to advance the left-hand pane to the Policies tab.

21. Select the VDB Snapshot policy to be applied to the vPDB.
    Select a Retention Policy for the vCDB, if you are provisioning a vCDB.
22. Click **Next** to advance the left-hand pane to the Masking tab.
    Select the Mask this vPDB checkbox if you want to mask, and select the masking job to be applied.
23. Click **Next** to advance the left-hand pane to the Hooks tab, and create any hooks if necessary. For more information, see Hook Scripts for Automation and Customization. (see page 1056)
24. Review the provisioning summary. Confirm all the fields are correct. Click **Submit** to proceed with provisioning the vPDB.

9.7.10.6 A closer look at TDE provisioning

> The following is intended to provide a deeper understanding of the Delphix TDE implementation, and is not required for a general understanding of the general provisioning workflows.

9.7.10.6.1 Delphix Provisioning workflow with and without TDE

Provisioning a Virtual Pluggable Database (vPDB) first involves using the GUI or CLI to specify the vPDB parameters (such as the vPDB name and target container) along with the snapshot to provision from. Once the provision job is started with these parameters, the Delphix Engine does the following:

1. Mounts the snapshot files on the target host.
2. Creates and opens (in mount mode) the auxiliary container database on the target host, using the snapshot files. The auxiliary container database will have both the CDB and PDB data files from the dSource.
3. Completes recovery to bring the auxiliary container database into a consistent state.
4. Finalizes the state of the auxiliary database and unplugs the vPDB datafiles.
5. Plugs the vPDB into the target database, and opens it in read-write mode for general use.

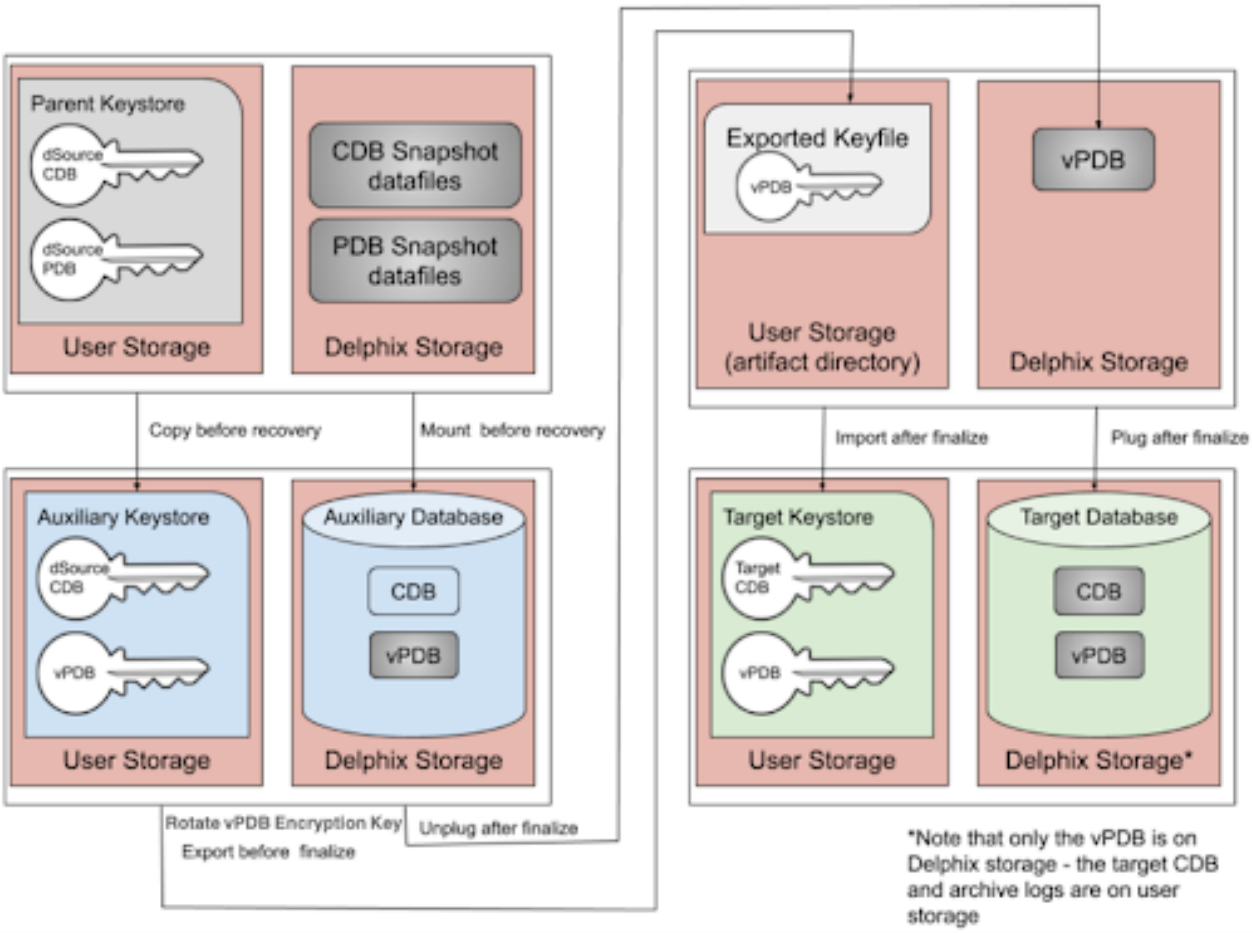
If the dSource is TDE-enabled, then Delphix will need to perform additional operations to complete the provision of a TDE-enabled vPDB to a TDE-enabled target container (indicated in red below):

1. Mounts the snapshot files on the target host.
2. Creates and opens (in mount mode) the auxiliary container database on the target host, using the snapshot files. The auxiliary container database will have both the CDB and PDB data files from the dSource.
3. Creates a keystore for the auxiliary container database with the necessary keys to apply encrypted archived log files.
4. Completes recovery to bring the auxiliary container database into a consistent state.

5. Rotates the vPDB and auxiliary CDB master encryption keys by generating new keys that are unique to the vPDB / auxiliary CDB and not associated with the source PDB or CDB.
6. Exports only the newly generated keys to an exported keyfile to enable unplug.
7. Finalizes the state of the auxiliary database and unplugs the vPDB datafiles.
8. Imports the keys from the exported keyfile into the target keystore.
9. When provisioning to a new vCDB target, converts the auxiliary CDB into the final vCDB and creates the vCDB keystore from the auxiliary CDB keystore.
10. Plugs the vPDB into the target database, and opens it in read-write mode for general use.

> If a plugin violation is detected after plugging the vPDB into the target database, the vPDB provision (and enable) operation will throw a critical error during the SnapSync operation, but it will succeed nonetheless. The vPDB will remain in a running state. You must manually resolve the plugin violations outside of Delphix, close and reopen the vPDB to make the changes effective, and then disable/enable the vPDB within Delphix.

The following diagram illustrates the provisioning steps.



At each stage of the provisioning process, the keys and exported keyfiles are always on user storage. The exported keyfile is located in the artifact directory, while the auxiliary and target keystores are in the auxiliary keystores directory. Both the artifact directory and auxiliary keystores directory are subdirectories of the TDE keystores root directory, which is either user specified, or if not specified defaults to the toolkit root directory. Similar to non-TDE-enabled vPDBs, the final vPDB (and vCDB, if applicable) is on Delphix storage while the target Linked CDB and its archive logs remain on user storage.

9.7.10.6.2 Validating the keystore password

To validate that the password for a given keystore is correct, Oracle provides the `mkstore` command-line utility. Navigate to the keystore folder on the target host where `ewallet.p12` exists and run the command `mkstore -wrl . list` as the Oracle user. If the password is incorrect, you will see output similar to the following:

```
$ mkstore -wrl . -list
Oracle Secret Store Tool Release 19.0.0.0.0 - Production
Version 19.4.0.0.0
Copyright (c) 2004, 2021, Oracle and/or its affiliates. All rights reserved.

Enter wallet password:   xxxxxxxx
```

```
oracle.security.crypto.core.CipherException: Invalid padding string (or incorrect
password)
```

If the password is correct, you will see output similar to the following:

```
$ mkstore -wrl . -list
Oracle Secret Store Tool Release 19.0.0.0.0 - Production
Version 19.4.0.0.0
Copyright (c) 2004, 2021, Oracle and/or its affiliates. All rights reserved.

Enter wallet password: xxxxxxxx
Oracle Secret Store entries:
ORACLE.SECURITY.DB.ENCRYPTION.AQtDPGje509PvxYeQuG7fmYAAAAAAAAAAAAAAAAAAAAAAAAAAAA
ORACLE.SECURITY.DB.ENCRYPTION.AVXDkpbWnE9rv8fzTPiHTXcAAAAAAAAAAAAAAAAAAAAAAAAAAAA
ORACLE.SECURITY.DB.ENCRYPTION.AWS1QwdQHk/jv0i5eJ3sb10AAAAAAAAAAAAAAAAAAAAAAAAAAAA
ORACLE.SECURITY.DB.ENCRYPTION.AZcpc0/QrU/Nv/Q54WRWRSMAAAAAAAAAAAAAAAAAAAAAAAAAAAA
ORACLE.SECURITY.DB.ENCRYPTION.MASTERKEY
ORACLE.SECURITY.DB.ENCRYPTION.MASTERKEY.C204B25A34B93EB7E055000000000001
ORACLE.SECURITY.DB.ENCRYPTION.MASTERKEY.C204B30EA4A03FABE055000000000001
ORACLE.SECURITY.DB.ENCRYPTION.MASTERKEY.C204B407D46A3FFDE055000000000001
ORACLE.SECURITY.ID.ENCRYPTION.
ORACLE.SECURITY.KB.ENCRYPTION.
ORACLE.SECURITY.KM.ENCRYPTION.AQtDPGje509PvxYeQuG7fmYAAAAAAAAAAAAAAAAAAAAAAAAAAAA
ORACLE.SECURITY.KM.ENCRYPTION.AVXDkpbWnE9rv8fzTPiHTXcAAAAAAAAAAAAAAAAAAAAAAAAAAAA
ORACLE.SECURITY.KM.ENCRYPTION.AWS1QwdQHk/jv0i5eJ3sb10AAAAAAAAAAAAAAAAAAAAAAAAAAAA
ORACLE.SECURITY.KM.ENCRYPTION.AZcpc0/QrU/Nv/Q54WRWRSMAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

9.7.10.6.3 Example Illustrating Generation of New Encryption Keys for vPDBs

This example illustrates the generation of new encryption keys as a result of provisioning a TDE-enabled vPDB. Consider a vPDB `tde_vpdb` that is provisioned from a dSource `CDOMSHSR52CAPDB2` on the VM `tde-source19`, which is an Oracle database running version 19.11.0. Connecting to this database, we can query `v$encryption_keys` to determine the current keys in use by each PDB:

```
SQL> show pdbs
CON_ID CON_NAME                OPEN MODE  RESTRICTED
------ ---------------------- ---------- ----------
     2 PDB$SEED                READ ONLY  NO
     3 CDOMSHSR52CAPDB1        READ WRITE NO
     4 CDOMSHSR52CAPDB2        READ WRITE NO
     5 CDOMSHSR52CAPDB3        READ WRITE NO
SQL> select con_id, key_id from v$encryption_keys order by con_id;
CON_ID KEY_ID
------ ----------------------------------------------------
     1 Ac9MY5kQwU8GvwlYMXImXmMAAAAAAAAAAAAAAAAAAAAAAAAAAAA
     3 AedrXL3aUk9zv+9t7J8ZsVYAAAAAAAAAAAAAAAAAAAAAAAAAAAA
     4 AdDdKibLKU9mv6PDAIvVvH0AAAAAAAAAAAAAAAAAAAAAAAAAAAA
     5 AWdc3ZRaP09Pvw4+2FmLwHAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

The `v$encryption_keys` output for this environment shows that there are 3 PDBs within this CDB, all of which are TDE-enabled. In particular, the PDB used for the dSource has a `con_id` of 4, and an encryption `key_id` starting with **AdDdKibL**.

The vPDB `tde_vpdb` is provisioned to the CDB `CDOMSHTG93CF` on the VM `tde-target19`. Connecting to this database, we can again query `v$encryption_keys` to determine the keys in use by each PDB:

```
SQL> show pdbs
CON_ID CON_NAME         OPEN MODE  RESTRICTED
------ ---------------- ---------- ----------
     2 PDB$SEED         READ ONLY  NO
     3 CDOMSHTG93CFPDB1 READ WRITE NO
     4 CDOMSHTG93CFPDB2 READ WRITE NO
     5 CDOMSHTG93CFPDB3 READ WRITE NO
     6 TDE_VPDB         READ WRITE NO
SQL> select con_id, key_id from v$encryption_keys order by con_id;
CON_ID KEY_ID
------ ----------------------------------------------------
     1 AZTc9eKqlk98v8GkQ8/AmaAAAAAAAAAAAAAAAAAAAAAAAAAAAA
     6 AZ3DQws5pE9LvxYDP14hDHwAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

The key which was originally present in the wallet on the dSource - **AdDdKibL** - is not present in the target keystore. Instead, Delphix has generated a new key - **AZ3DQws5** for `con_id` of 6, which corresponds to the `con_id` of the vPDB. This key is unique to each vPDB and is not associated with the source PDB. This happens as a result of rotating the vPDB encryption keys as described in Delphix Provisioning workflow with and without TDE (see page 0).

There are several things to note about the behavior of Oracle and the `v$encryption_keys` table:

1. Keys are never deleted from existing keystores by Oracle, only new keys are added. Therefore, if we were to disable the vPDB, which will unmount and unplug it from the CDB, `v$encryption_keys` will still show the key as present, with its original `con_id`, even though it has been unplugged:

```
SQL> show pdbs
CON_ID CON_NAME         OPEN MODE  RESTRICTED
------ ---------------- ---------- ----------
     2 PDB$SEED          READ ONLY  NO
     3 CDOMSHTG93CFPDB1  READ WRITE NO
     4 CDOMSHTG93CFPDB2  READ WRITE NO
     5 CDOMSHTG93CFPDB3  READ WRITE NO
SQL> select con_id, key_id from v$encryption_keys order by con_id;
CON_ID KEY_ID
------ ----------------------------------------------------
     1 AZTc9eKqlk98v8GkQ8/AmaAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
     6 AZ3DQws5pE9LvxYDP14hDHwAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

2. If the wallet is closed for a particular PDB, `v$encryption_keys` will not show any entries for that PDB. The wallet status can be determined by querying `v$encryption_wallet`.
3. Querying `v$encryption_wallet` while the session is attached to `CDB$ROOT` will return information about the entire CDB, otherwise, only the keys for the current PDB are returned.

9.7.10.7 Provisioning a vPDB from a non-multitenant source

Delphix supports provisioning a virtual pluggable database (vPDB) from a snapshot of a non-multitenant (non-MT) source database (VDB). This feature is only available through the API or command-line interface (CLI). This topic describes how to provision a vPDB from a snapshot of a non-MT VDB (also simply referred below as source VDB) using CLI.

The high-level workflow for the provisioning is as follows:

This feature has the following restrictions:

1. Transparent Data Encryption (TDE) is not supported.
2. The provision point must correspond to a snapshot. Provisioning from a point in time between snapshots is not supported.
3. The target CDB (where the new vPDB will be plugged into) must be a physical CDB. Virtual CDB targets are not supported.
4. For provisioning from a source VDB of Oracle 11g version, the VDB must be upgraded before provisioning (i.e. you must use the Upgrade Option 1 described below). Upgrade option 2 cannot be used in this case.

- Choose the upgrade option to use if the target CDB (where the new vPDB will be plugged into) version is newer than the source VDB. Refer to the prerequisites section below.
- Create the required hook scripts.
- Take source VDB snapshot.

- Provision the vPDB to the target CDB using the VDB snapshot.

9.7.10.7.1 Prerequisites

9.7.10.7.1.1 Environment requirements

Provisioning a vPDB from a non-MT source has the following environment requirements:

- Source host with a non-MT Oracle 11g or newer source database.
- VDB Target host for provisioning a non-MT VDB from the source database.
- CDB target host with a running Oracle target version CDB. The target CDB will be automatically linked if it is not already linked.

9.7.10.7.1.2 Select upgrade option

The target CDB can be a newer Oracle version than the source database (for example, the source is 12.2 and target is 19c). When an upgrade is also required, there are two options for upgrading the database:

- Upgrade Option 1: After source VDB is provisioned and before vPDB is provisioned via CLI. This upgrade is done manually by the user, before initiating the vPDB provisioning API/CLI. This option requires the ability to upgrade to the Oracle target version on the VDB target host.
- Upgrade Option 2: After plugging the newly provisioned vPDB into the target CDB database. This upgrade is performed by Delphix API/CLI using a hook script.
  If the source VDB's Oracle version is 11g, Upgrade Option 1 must be selected. For source VDB's with Oracle versions 12c and above, either option may be selected.

9.7.10.7.1.3 Prepare hook scripts

There are following three scripts used during this procedure and must be created manually before executing the vPDB provisioning command by CLI:

1. **Pre-snapshot Hook on source VDB**: This hook will open the database in "read only" mode and issue a call to the `dbms_pdb.describe` procedure to generate an XML file called delphix_plugin.xml, describing the VDB. The XML file will be used to plug the source VDB data files into the target CDB. The source VDB must be open read only during the subsequent snapshot so that the VDB data files do not require recovery when plugging them into the target CDB. This can be added using Delphix Management Application UI from the Datasets panel by selecting the source VDB and then going to Configuration -> Hooks tab.
2. **Post-snapshot Hook on source VDB**: This hook will return the VDB to "read write" mode. This is an optional script. The source VDB can also remain read only. This can be added using Delphix Management Application UI from the Datasets panel by selecting the source VDB and then going to Configuration -> Hooks tab.
3. **Post-plug Hook for vPDB**: This script will run as a hook present on the Linked CDB target host and will be executed after the newly provisioned vPDB is plugged into the target CDB.
   a. If no upgrade is required or using Upgrade Option 1, then this script will call the Oracle script

`$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql` to convert the VDB into a PDB. b. If using Upgrade Option 2, this script will upgrade the database and then call Oracle script `$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql`.
This script must be created manually and stored in the root folder of the Delphix Toolkit directory of the target CDB host. The name of the vPDB being provisioned/converted will be supplied by Delphix as the first parameter to the script when it invokes the script. The VDB data files will already be plugged into the Linked CDB target at the time the script is invoked.

Refer to the Sample Scripts (see page 1576) section below for the content of scripts.

> Note about the sample scripts provided in this document:
>
> - These scripts are provided as-is, without warranty of any kind or commercial support through Delphix.
> - The scripts may need to be modified depending on the Oracle version or the SQL script package version being used.

9.7.10.7.2 Workflow

1. Link the non-MT source database as a dSource within Delphix.
2. Provision a non-MT Oracle VDB from the dSource onto the VDB target host. This will be referred to as the **Golden VDB**.
3. **If no upgrade is required:** a. Create a Pre-snapshot hook (see page 1576) on the **Golden VDB.** b. (Optional) Create a Post-snapshot hook (see page 1576) on the **Golden VDB**. c. Take a snapshot of the **Golden VDB**. d. Create a PDB conversion script (see page 1576) named `dx-post-plug-hook.sh` in the root of the Delphix toolkit directory of the Linked CDB target host.
   If using Upgrade Option 1: a. Upgrade the **Golden VDB** to the Oracle target version: manually upgrade the database and point it to the new Oracle home. b. Create a Pre-snapshot hook (see page 1576) on the **Golden VDB.** c. (Optional) Create a Post-snapshot hook (see page 1576) on the **Golden VDB**. d. Take a snapshot of the **Golden VDB**. e. Create a PDB conversion script (see page 1576) named `dx-post-plug-hook.sh` in the root of the Delphix toolkit directory of the Linked CDB target host.
   If using Upgrade Option 2: a. Create a Pre-snapshot hook (see page 1576) on the **Golden VDB.** b. (Optional) Create a Post-snapshot hook (see page 1576) on the **Golden VDB**. c. Take a snapshot of the **Golden VDB**. d. Create a PDB Upgrade and Conversion script (see page 1576) named `dx-post-plug-hook.sh` in the root of the Delphix toolkit directory of the Linked CDB target host.
4. Select the **snapshot** on the **Golden VDB** created above and provision a vPDB to the Linked CDB target. The detailed steps for this are documented in the next section.

This step can be executed via the **API / CLI only**, and will not be allowed via the Delphix UI.

9.7.10.7.3 CLI procedure to provision a vPDB from a VDB

1. Log into the Delphix command-line interface using the admin user or a user with admin privileges.

   $ ssh admin@YOUR_ENGINE

2. Move to the database provisioning command line object.

   ```
   delphix> database provision
   ```

3. Set the parameter type to `OracleMultitenantProvisionParameters`.

   ```
   set type=OracleMultitenantProvisionParameters
   ```

4. (Optional) Set the login details for the provision and Delphix OS user who is to perform the provision.

   ```
   delphix database provision *> set username=delphix
   delphix database provision *> set credential.type=PasswordCredential
   delphix database provision *> set credential.password=delphix
   ```

5. Give the dataset a name.

   ```
   delphix database provision *> set container.name=vpdb
   ```

6. Place the new dataset in a Group that appears in the Delphix GUI, in this case, the Targets group.

   ```
   delphix database provision *> set container.group=Targets
   ```

7. Set the destination mount point which Delphix NFS mounts are to be linked to under the virtual PDB. This folder must exist at a file system level on the Linked CDB target host. Do not use single quotes around the mount path.
   a. Linux and Unix hosts, this mount path must be the full path and not include symlinks.

      ```
      delphix database provision *> set source.mountBase="/mnt/provision"
      ```

8. If automatically restarting the vPDB is not required after a reboot of the Linked CDB target host, set this to option to false. False is possibly a better option given the container database would need to be running prior to any attempt to pull up a vPDB.

```
delphix database provision *> set source.allowAutoVDBRestartOnHostReboot=false
```

9. Supply the destination container database name. The container database should already be discovered. This will be where the vPDB will ultimately be placed.

```
delphix database provision *> set sourceConfig.cdbConfig=CDBSTAGE
```

10. Name the vPDB. This is what it will appear as in the destination container database.

```
delphix database provision *> set sourceConfig.databaseName=vpdb
```

11. Supply the source **Golden VDB** details. In this example, the provision will use the latest snapshot available from the **Golden VDB** as the point in time from which to provision the vPDB. A specific snapshot can also be picked, but an arbitrary point in time is not supported.

    delphix database provision *> set timeflowPointParameters.type=TimeflowPointSemantic delphix database provision *> set timeflowPointParameters.container=gold_vdb delphix database provision *> set timeflowPointParameters.location=LATEST_SNAPSHOT

12. Check that all the settings you require are in place using the "ls" command.

```
delphix database provision *> ls
Properties
  type: OracleMultitenantProvisionParameters
  container:
    type: OracleDatabaseContainer
    name: vpdb (*)
    description: (unset)
    diagnoseNoLoggingFaults: true
    group: Targets (*)
    performanceMode: DISABLED
    preProvisioningEnabled: false
    sourcingPolicy: (unset)
  credential:
    type: PasswordCredential (*)
    password: ******** (*)
  masked: (unset)
  maskingJob: (unset)
  source:
    type: OracleVirtualPdbSource (*)
    name: (unset)
    allowAutoVDBRestartOnHostReboot: false (*)
    config: (unset)
    customEnvVars: (unset)
    fileMappingRules: (unset)
    LogCollectionEnabled: false
    mountBase: /mnt/provision (*)
```

```
    operations: (unset)
    parentTdeKeystorePassword: (unset)
    parentTdeKeystorePath: (unset)
    tdeExportedKeyFileSecret: (unset)
  sourceConfig:
    type: OraclePDBConfig
    cdbConfig: CDBSTAGE (*)
    databaseName: vpdb (*)
    environmentUser: (unset)
    linkingEnabled: true
    nonSysCredentials: (unset)
    nonSysUser: (unset)
    repository: (unset)
    services: (unset)
  timeflowPointParameters:
    type: TimeflowPointSemantic
    container: gold_vdb (*)
    location: LATEST_SNAPSHOT (*)
  username: delphix (*)
  VirtualCdb: (unset)

Operations
defaults
```

13. Initiate the provision by committing the operation in the CLI.

```
delphix database provision *> commit
  vpdb
  Dispatched job JOB-333
  DB_PROVISION job started for "Targets/vpdb".
  Starting provision of virtual PDB database "vpdb" converted from a single
tenant database.
  Preparing multitenant container database "CDBSTAGE".
  Creating new TimeFlow.
  Generating recovery scripts.
  Exporting storage.
  Preparing XML manifest file prior to plugin.
  Plugging in Oracle pluggable database.
  Running user-defined post plug hook.
  Opening Oracle pluggable database.
  Setting OMF destination for Oracle pluggable database.
  Creating PDB tempfiles.
  Checking Oracle pluggable database plugin violations.
  DB_PROVISION job for "Targets/vpdb" completed successfully.
```

To refresh the data in the vPDB from production, first, refresh the **Golden VDB** from the dSource, then refresh the vPDB from the new snapshot in the **Golden VDB**.

There are some workflow customizations required for RAC databases:

1. The PDB conversion script must be in the root of the Delphix toolkit directory for all the target CDB RAC instances.
2. The **Golden VDB** Pre-Snapshot hook, as provided below, will not work in a clustered (RAC) environment with more than one active instance because it only shuts down the local instance. `dbms_pdb.describe` will not execute while an instance is open read-write. The workarounds are:
    a. Provision the **Golden VDB** as single-instance, either by provisioning to a non-RAC target or by provisioning to a RAC target with only one active instance. The sample hook will work in this case.
    b. Write a customized pre-snapshot hook that shuts down all instances, restarts only one instance in read-only mode, and runs `dbms_pdb.describe`.
    c. Manually perform the actions of the hook: shutdown the **Golden VDB**, restart one of the instances in read-only mode and then run `dbms_pdb.describe`.

9.7.10.7.4 Sample scripts

9.7.10.7.4.1 Golden VDB pre-snapshot hook

Restarts the source/Golden VDB in read only mode and runs `dbms_pdb.describe` to generate an XML file describing the VDB. The XML file will be used to plug the VDB into the Linked CDB target. The target for the XML file must be $DELPHIX_MOUNT_PATH/$DELPHIX_DATABASE_UNIQUE_NAME/datafile/delphix_plugin.xml.

```
#!/bin/sh

sqlplus "/ AS SYSDBA" <<-EOF
  whenever sqlerror exit 2;
  spool $DELPHIX_MOUNT_PATH/$DELPHIX_DATABASE_UNIQUE_NAME/datafile/presnapshot.out
replace
  shutdown immediate
  startup mount
  alter database open read only;
  exec dbms_pdb.describe(pdb_descr_file=>'$DELPHIX_MOUNT_PATH/
$DELPHIX_DATABASE_UNIQUE_NAME/datafile/delphix_plugin.xml');
  exit;
EOF
```

9.7.10.7.4.2 Golden VDB post-snapshot hook

This is only necessary if the source VDB should not be left in read-only mode after the snapshot.

```
#!/bin/sh

sqlplus "/ AS SYSDBA" <<-EOF
  whenever sqlerror exit 2;
  spool $DELPHIX_MOUNT_PATH/$DELPHIX_DATABASE_UNIQUE_NAME/datafile/postsnapshot.out
replace
  shutdown immediate
  startup
  exit;
EOF
```

9.7.10.7.4.3 PDB conversion script

This script converts the source/Golden VDB datafiles into PDB datafiles. The script should be named `dx-post-plug-hook.sh` and reside in the root of the Delphix toolkit directory of the Linked CDB target host. Delphix will supply the name of the PDB being provisioned/converted as the first parameter.

The VDB datafiles will have already been plugged into the target CDB at the time the script is invoked and the virtual PDB will be in the mounted (not open) state. The PDB conversion script should return with the virtual PDB in either the mounted or open (not restricted) state. Delphix does not enforce a time-out for the script.

```
#!/bin/sh

DELPHIX_PDB_NAME=$1
SCRIPT_DIR="$( cd "$( dirname "$0" )" && pwd )"
CONVERT_LOGFILE=$SCRIPT_DIR/$DELPHIX_PDB_NAME-pdbconvert.log

sqlplus "/ AS SYSDBA" <<-EOF
  whenever sqlerror exit 2;
  spool $CONVERT_LOGFILE replace
  alter session set container=$DELPHIX_PDB_NAME;
  @?/rdbms/admin/noncdb_to_pdb.sql
  exit;
EOF
```

9.7.10.7.4.4 PDB upgrade and conversion script

This script upgrades the newly provisioned vPDB to the target CDB version and then converts the source/Golden VDB datafiles into PDB datafiles. The script should be named `dx-post-plug-hook.sh` and reside in the root of the Delphix toolkit directory of the Linked CDB target host. Delphix will supply the name of the PDB being provisioned/converted as the first parameter. The VDB datafiles will have already been plugged into the target CDB at the time the script is invoked and the virtual PDB will be in the mounted (not

open) state. The PDB conversion script should return with the virtual PDB in either the mounted or open (not restricted) state. Delphix does not enforce a time-out for the script.

```
#!/bin/sh

DELPHIX_PDB_NAME=$1
SCRIPT_DIR="$( cd "$( dirname "$0" )" && pwd )"
UPGRADE_LOGFILE=$SCRIPT_DIR/$DELPHIX_PDB_NAME-dx-post-plug-upgrade.log
UPGRADE_LOGDIR=$SCRIPT_DIR/$DELPHIX_PDB_NAME-upgrade

mkdir $UPGRADE_LOGDIR
cd $ORACLE_HOME/rdbms/admin
switches="-c '$DELPHIX_PDB_NAME' -l $UPGRADE_LOGDIR"
$ORACLE_HOME/perl/bin/perl catctl.pl $switches catupgrd.sql &>> $UPGRADE_LOGFILE

CONVERT_LOGFILE=$SCRIPT_DIR/$DELPHIX_PDB_NAME-pdbconvert.log

sqlplus "/ AS SYSDBA" <<-EOF
  whenever sqlerror exit 2;
  spool $CONVERT_LOGFILE replace
  alter session set container=$DELPHIX_PDB_NAME;
  @?/rdbms/admin/noncdb_to_pdb.sql
  exit;
EOF
```

9.7.10.8 Provisioning a VDB from an encrypted Oracle database

9.7.10.8.1 Overview

This topic describes how to provision a VDB from an encrypted database. The Delphix Engine supports provisioning from a dSource linked to a physical database that has been encrypted with Oracle's Transparent Database Encryption (TDE), which can be used to encrypt columns or tablespaces.

> The Delphix engine supports provisioning from a dSource with an encrypted system tablespace in a non-multitenant configuration. The VDB's wallet on the target environment must include an auto-login wallet. Local auto-login wallets are not supported for this configuration.

Provisioning a VDB from an encrypted dSource requires an auto-open wallet setup in the target environment, because the provisioning process requires the master key to be stored in the wallet file. On the dSource, export the keys and copy the export file (both `ewallet.p12` and `cwallet.sso)` to the VDB server. On the VDB server, create an empty wallet and import the keys into the wallet, then set the key to be used.

When provisioning a VDB from an encrypted dSource, if the target environment has other databases that also use TDE, each database should use a different wallet. This also includes a scenario where the VDB has been provisioned back to the same environment as the encrypted dSource. Please check Oracle documentation on

how to set up different wallet locations for different databases. For example, use `$ORACLE_SID` in the `DIRECTORY` clause of the `ENCRYPTION_WALLET_LOCATION` parameter in `sqlnet.ora`.

```
ENCRYPTION_WALLET_LOCATION=(SOURCE=(METHOD=FILE)(METHOD_DATA=(DIRECTORY=/opt/oracle/
wallets/$ORACLE_SID)))
```

9.7.10.8.2 Procedure

1. Check for any encrypted columns or tablespaces on the source database by using these commands:

```
SELECT t.name name, e.encryptionalg algorithm FROM v$tablespace t,
v$encrypted_tablespaces e
WHERE t.ts# = e.ts# and upper(e.encryptedts) = 'YES';
```

2. Copy wallet files from the source database to the target environment, and then configure the `sqlnet.ora` file on the target to point to the directory where the wallet is located.

```
$ more sqlnet.ora
ENCRYPTION_WALLET_LOCATION=(SOURCE(METHOD=file) (METHOD_DATA=(DIRECTORY=/opt/
oracle/oradata/nf/wallet)))
```

3. If the source database does not use the auto-open wallet, create the auto-open wallet at the target environment.

```
$ orapki wallet create -wallet /opt/oracle/oradata/nf/wallet -auto_login [-pwd
password]
```

4. Proceed with provisioning the VDB as described in Provisioning an Oracle VDB (see page 1541)

9.7.10.9 Managing Oracle RAC virtual databases with only a subset of configured RAC nodes available

Overview

You can refresh, rewind, start, stop, disable or enable Oracle RAC virtual databases even when some of the RAC nodes are not available as a result of being shutdown temporarily.

If only some, but not all of the configured RAC nodes, are not available, ensure that these nodes are disabled in the **Environments** page of the **Delphix Management** application**.** This will enable the management operations on the virtual databases using only the nodes that are enabled and accessible.

> If a node is enabled but cannot be accessed from the Delphix engine, the VDB management operations will fail with an error as '*Enabled nodes "<hostnames of nodes>" are unavailable for the RAC cluster database "your vdb".* '

Before attempting VDB operations with a subset of RAC nodes available

If the nodes are already shutdown, disable them in the **Environments** page of the **Delphix Management** application and refresh the target environment, before attempting any operation on the virtual sources running on these nodes.

If the nodes are planned to be shutdown, follow the steps below before the nodes are shutdown and attempting any operation on the virtual sources running on these nodes:

- Stop the virtual database instance (see page 0) on these nodes from the **Delphix Management** application.

> If a virtual CDB is running on such nodes and it contains more than one Virtual PDB, then after stopping all the vPDB instances on the nodes, stop the vCDB instance (see page 0) on these nodes using Delphix CLI.

- Disable the nodes (see page 0) from the **Delphix Management** application.
- Refresh the target RAC environment.

Before attempting VDB operations after nodes are made available again

When the RAC nodes are started back up and ready to run the virtual database instances again, before attempting any operation on the virtual databases, follow the steps below :

- Enable the nodes (see page 0) from the **Delphix Management** application**.**
- Refresh the target RAC environment.
- Start the virtual source.
  For a vPDB in a virtual CDB, start the virtual CDB (from the **Delphix Management** application) before starting the vPDB.

9.7.10.10 Managing multiple virtual PDBs in a virtual CDB

9.7.10.10.1 Overview

Earlier, only one virtual PDB was supported in a virtual CDB. Starting with Delphix Engine version 8.0.0.0, you can provision multiple vPDBs in a vCDB. This enables you to manage vCDBs and vPDBs independently, when there are more than one vPDB in a vCDB.

This new feature is supported for Oracle versions 12.1.0.2 and later only.

> To support multiple vPDBs in a vCDB, Oracle 12.1.0.2 needs an Oracle patch for bug 18967466. Once the patch or update (containing the patch) is installed,
>
> make sure that the `appliedPatches` property of the Oracle repository is also updated to include the bug number. Refer to Updating repository for applied patches with the Command Line Interface (see page 1466) for instructions.

9.7.10.10.2 Managing vCDBs

You can enable/disable, start/stop or delete a vCDB using **Actions (...)** menu from the **Delphix Management** application.

9.7.10.10.2.1 Prerequisites

- To disable a vCDB, all of its vPDBs must already be disabled using Delphix Management application.
- To stop a vCDB, all of its vPDBs must already be stopped or disabled using Delphix Management application.
- To delete a vCDB, all of its vPDBs must already be deleted using Delphix Management application.

9.7.10.10.2.2 Procedure

1. Login to the **Delphix Management** application.
2. Navigate to **Manage > Datasets.**
3. Select the vCDB you want to manage.
4. Click **Actions (...)** menu
5. Select the desired action for the vCDB.

9.7.10.10.3 Managing vPDBs

You can refresh/rewind(undo refresh), enable/disable, start/stop or delete a vPDB using **Actions (...)** menu from the **Delphix Management** application.

9.7.10.10.3.1 Prerequisites

- To disable a vPDB, its vCDB must be in a running state. You cannot disable a vPDB if its vCDB is in a stopped/disabled state. This is required in the current Delphix Engine version, because in previous versions disable succeeds even if the vCDB is in a stopped/disabled state.
- To enable/start a vPDB, if there are multiple vPDBs in its vCDB, then vCDB must have been enabled/started and open in RW mode already. If the vCDB is not enabled/started, enable/start it using **Delphix Management** application's **Action (...)** menu as mentioned in above.

9.7.10.10.3.2 Procedure

1. Login to the **Delphix Management** application.
2. Navigate to **Manage > Datasets.**
3. Select the vPDB you want to manage.
4. Click **Actions (...)**
5. Select the desired action for the vPDB.

Make sure to follow the below points, once the desired action is selected.

- Refresh/Rewind:
  - Only one vPDB in a vCDB with source-level parent-child relationship: Both the vPDB and its vCDB will be refreshed/rewound.

> The vPDB and its vCDB has a source-level parent-child relationship if the dSource CDB timeflow, from which vCDB timeflow is derived, is the same as the parent CDB timeflow of the dSource PDB timeflow from which vPDB is derived. This was always the case when there was only one vPDB in a vCDB. But with multiple vPDBs support, this may not be always the case, since later vPDB (provisioned to the same vCDB) may be provisioned from a dSource PDB of a different dSource CDB.

Consider the following scenarios:

- Provisioning a vPDB (VPDB1) from a dSource PDB (PDB1 in a source CDB1) to a new vCDB (VCDB1) will create a vPDB-vCDB virtual databases with source-level parent-child relationship. If such a vPDB is refreshed/rewound, then both vPDB and its vCDB will be refreshed/rewound.
- Provisioning a 2nd vPDB (VPDB2) from another dSource PDB (PDB2 in another source container CDB2) to the same vCDB (VCDB1) will create vPDB-vCDB databases (i.e. VPDB2-VCDB1) that does not have such a relationship. So, if such a vPDB is the only vPDB in the vCDB, then only the vPDB will be refreshed/rewound and not its vCDB.
- Only one vPDB in a vCDB with no source-level parent-child relationship: Only the vPDB, on which the action is performed, will be refreshed/rewound. Its vCDB will not be refreshed/rewound. (Refer to the example scenario 2 above.)
- Multiple vPDBs in a vCDB: Only the vPDB, on which the action is performed, will be refreshed/rewound.
- Disable/Stop:
  - If there are multiple vPDBs in a vCDB, then the action will be performed only for the vPDB.
  - If there is only one vPDB in the vCDB, then its vCDB will be also be disabled/stopped after the vPDB is disabled/stopped.

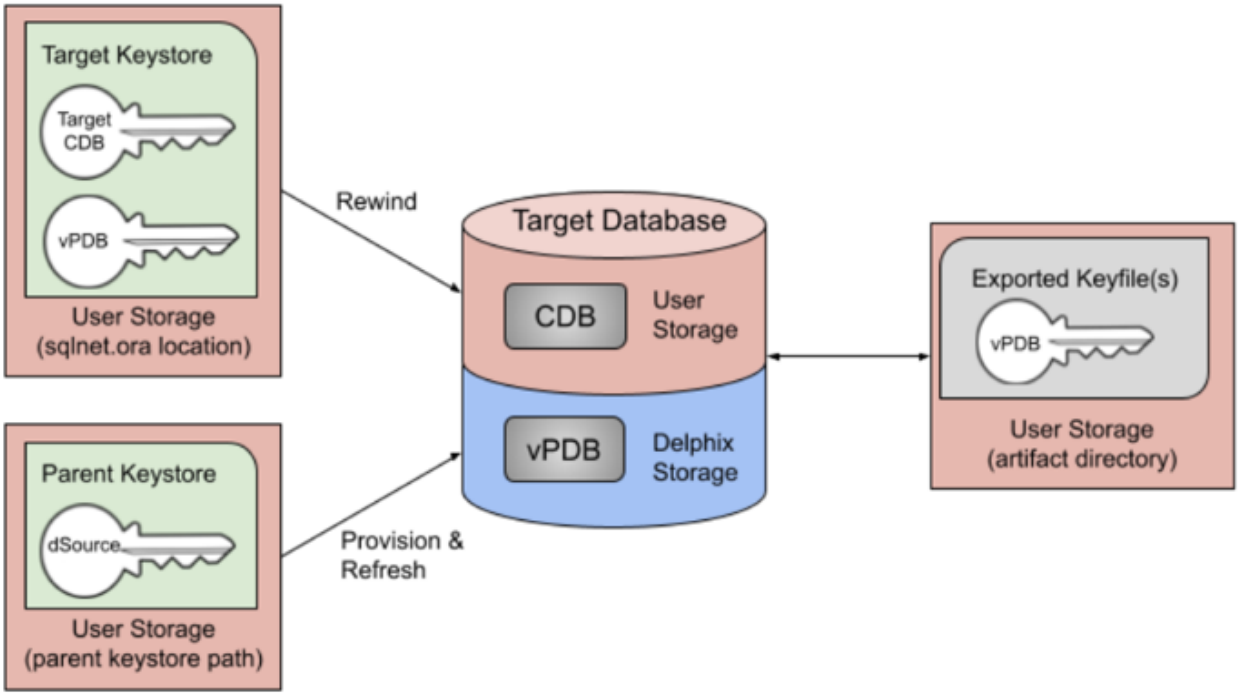
> This differs from the previous Delphix Engine(version above 8.0.0.0) behavior which always disabled/stopped the vCDB as well.

- Enable/start:
  - If there are multiple vPDBs in a vCDB, then the action will enable/start the vPDB only.
  - If there is only one vPDB in the vCDB, then the vCDB will be also be enabled/started if required before enabling/starting the vPDB itself.
- Delete
  - If this is the last vPDB in a vCDB, then the vCDB is also deleted.
  - If you are deleting a vPDB using the force option, make sure to follow the steps below to clean up the vPDB from the target host:
    1. Drop the virtual PDB from the virtual CDB manually after it is deleted from the Delphix Management UI.
    2. Remove the virtual PDB mount point from the target host (residing in the mount base directory) to avoid stale mounts.

> If a physical PDB is added to a virtual CDB or a virtual PDB from a virtual CDB is deleted using the force option, LogSync will fail with the error "`Virtual container database <vCDB Name> has pluggable databases that are not managed by Delphix: <PDB names>.`" and/or SnapSync will fail with the error code "`exception.oracle.vdb.foreign.pdbs.found.in.vcdb`". To resolve the errors, unplug the physical PDBs from the virtual CDB manually or drop the virtual PDBs (that were deleted from Delphix using the force option) from the virtual CDB manually.

9.7.10.11 Refreshing and rewinding a TDE-enabled vPDB

Just like a non-TDE-enabled vPDB, a TDE-enabled vPDB can be refreshed from the dSource or rewound to a previous snapshot or point in time. In each case, no additional manual steps or input from the user is required. The first step of a refresh or rewind operation is to disable the existing vPDB, which will result in a new keyfile exported to the artifact directory. The appropriate snapshot files are then mounted for the auxiliary database so that it can be recovered and brought to a consistent state. Since the vPDB is TDE-enabled, a keystore is needed for the recover operation. For a refresh, the Delphix Engine will use the parent keystore, and for a rewind, the Delphix Engine will use the target keystore, as shown below.



9.7.10.11.1 Overview of Key Rotation

Some customers have strict security compliance standards that mandate that production master keys cannot be shared into non-production zones. Delphix supports the ability to perform automated keystore sanitization of a vPDB. In simpler terms, Delphix allows provisioning a vPDB that has no previous production keys associated with it. A freshly provisioned vPDB will thus contain one and only one newly-set master encryption key that can be imported into the target CDB keystore to resolve TDE-plugin violations at the end of a provision job. Note that the tablespace encryption keys, which are themselves encrypted by the PDB key, are not rotated. In such a scenario, this new encryption key is expected to be the only key imported by the target container database (CDB) at the end of the provision job. It is important to note that Delphix does not re-encrypt the actual data files when the production master key is rotated.

There are two potential places for keys to be rotated in a vPDB environment:

1. dSource: If the dSource keys are rotated and a new snapshot taken with the new key, the customer is responsible for updating the parent keystore before refreshing from the later snapshot encrypted with the new key. The parent keystore would then contain both the new key and the original keys.
2. Target: If the target CDB keys are rotated, the target keystore will be updated. This is why the Delphix Engine uses the target keystore for rewind operations.

In either scenario, the keystore used for recovery will contain the current and all prior keys used to encrypt the datafiles and archive logs, for both the vPDB and CDB used in the auxiliary container.

9.7.10.11.2 vPDB encryption key rotation

During the provisioning process of a TDE-enabled vPDB, Delphix generates a unique encryption key for the vPDB. This unique key is not associated with the parent keystore to ensure that no keys from the parent are imported by the target. During refresh and rewind operations, Delphix reuses that key after recovery has

finished. It is possible to customize the key that is used by updating the `tdeKeyIdentifier` parameter of the source via the CLI.

Key rotation can be performed in the following two ways:

1. *Key rotation performed by Delphix*. This covers the scenario when you want Delphix to manage key rotation for your vPDBs. The following steps need to be performed before subsequent Delphix operations on the vPDB can function normally:
   a. Unset the existing `tdeKeyIdentifier` for the vPDB via Delphix CLI - this is documented in step 5 in Locating and Updating the Value of tdeEncryptionKey (see page 2692).
   b. Refresh/rewind the vPDB via Delphix.

> After the `tdeKeyIdentifier` field is unset, Delphix will generate a new encryption key for the vPDB to be used from that point onward.

2. *Key rotation manually performed outside of Delphix*. This is useful for the scenario when you have a Delphix-managed vPDB but prefer a manual process for rotating the vPDB encryption keys. In this case, the following steps need to be performed before subsequent Delphix operations on the vPDB can function normally:
   a. Set a new master encryption key in the keystore via the `ADMINISTER KEY MANAGEMENT SET KEY` command on the target virtual database.
      i. **NOTE**: If you use the `ADMINISTER KEY MANAGEMENT CREATE KEY` command to create a new encryption key, you must activate this key using the `ADMINISTER KEY MANAGEMENT USE KEY` command otherwise subsequent Delphix operations may fail.
   b. Update `tdeKeyIdentifier` for the vPDB via Delphix CLI - this is documented in step 7 in Locating and Updating the Value of tdeEncryptionKey (see page 2692).

> If a valid `key_id` is entered for a key that is already present in the keystore, that key will be used as the active encryption key for the vPDB at the end of refresh/rewind.

9.7.10.11.3 vCDB encryption key rotation

There are two scenarios to be considered here.

9.7.10.11.3.1 Scenario 1 - the vCDB contains multiple vPDBs.

In this scenario, the vCDB encryption key can only be rotated outside of Delphix.

a. Set a new master encryption key in the keystore via the `ADMINISTER KEY MANAGEMENT SET KEY` command on the target virtual database.

NOTE: If you use the `ADMINISTER KEY MANAGEMENT CREATE KEY` command to create a new encryption key, you must activate this key using the `ADMINISTER KEY MANAGEMENT USE KEY` command otherwise subsequent Delphix operations may fail.

b. Update `tdeKeyIdentifier` for the vCDB via Delphix CLI - this is documented in step 7 in Locating and Updating the Value of tdeEncryptionKey (see page 2692).

9.7.10.11.3.2 Scenario 2 - the vCDB contains only one vPDB.

Key rotation can be performed in the following ways:

Key rotation performed by Delphix

This covers the scenario when you want Delphix to manage key rotation for your vCDB. The following steps need to be performed before subsequent Delphix operations on the vCDB can function normally:

- Unset the existing `tdeKeyIdentifier` for the vCDB via Delphix CLI - this is documented in step 5 in Locating and Updating the Value of tdeEncryptionKey (see page 2692).
- If you want to rotate the vPDB encryption key, you also need to unset the existing `tdeKeyIdentifier` for the vPDB via Delphix CLI.
- Refresh/rewind the vPDB via Delphix.

> After the `tdeKeyIdentifier` field is unset, Delphix will generate a new encryption key for the vCDB to be used from that point onward.

Key rotation manually performed outside of Delphix.

This is useful for the scenario when you prefer a manual process for rotating the vCDB encryption keys. In this case, the following steps need to be performed before subsequent Delphix operations on the vCDB can function normally:

- Set a new master encryption key for the vCDB in the keystore via the `ADMINISTER KEY MANAGEMENT SET KEY` command on the target virtual database.

  NOTE: If you use the `ADMINISTER KEY MANAGEMENT CREATE KEY` command to create a new encryption key, you must activate this key using the `ADMINISTER KEY MANAGEMENT USE KEY` command otherwise subsequent Delphix operations may fail.

- When there is only one vPDB in a vCDB and the vPDB is refreshed, the vCDB is re-created with the encryption key specified by the `tdeKeyIdentifier` of the vCDB source object. If you prefer to use this new vCDB encryption key after the vPDB is refreshed or rewound, you need to update the

`tdeKeyIdentifier` for the vCDB via Delphix CLI - this is documented in step 7 in Locating and Updating the Value of tdeEncryptionKey (see page 2692). This will ensure that the new vCDB encryption key is used by Delphix after the vPDB is refreshed or rewound.

9.7.10.12 Rotating TDE external key manager password

Changing passwords regularly is a common security practice to enhance the overall security of systems and maintain compliance. When OKV or HSM is used with TDE, changing the password of the External Key Manager (EKM) can result in Delphix workflow failures for datasets that rely on the EKM.

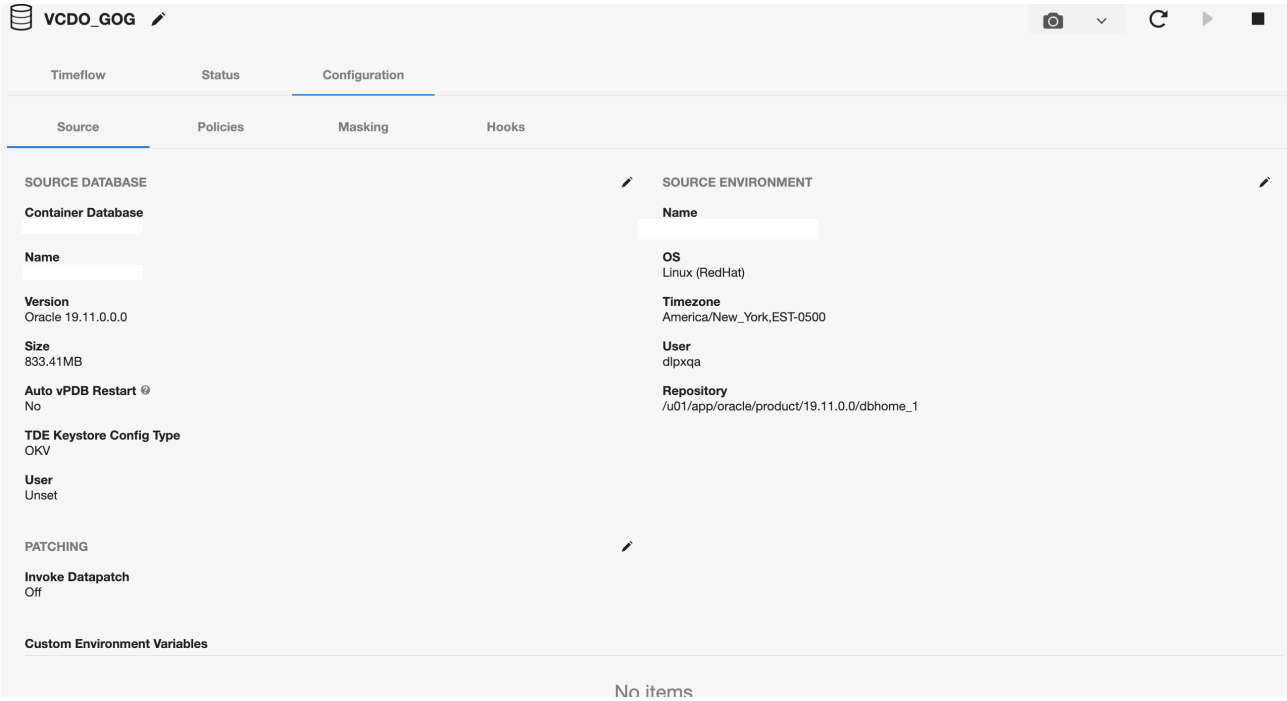
When the password of the EKM is rotated, it is necessary to update the TDE Key Manager Credential in the Delphix engine by following the procedure documented in Adding or Editing the TDE External Key Manager Credential (see page 1596).

Additionally, manual steps must be taken to update the credential in the autologin wallet created or shared with Delphix-managed datasets.

To update the password for **OKV endpoint**, refer to the Oracle OKV documentation[463]. To update the password for **Thales CipherTrust**, refer to the Thales CipherTrust documentation[464].

9.7.10.12.1 Example for updating the TDE external key manager password for OKV endpoint

Consider the following TDE OKV-enabled vPDB:



463 https://docs.oracle.com/en/database/oracle/key-vault/21.7/okvag/okvutil_endpoint_utility_reference.html#GUID-BC3DCC1B-F4A5-4C7C-A9B5-E47A92757866

464 https://thalesdocs.com/ctp/cm/2.14/admin/cm_admin/changing-passwords/index.html

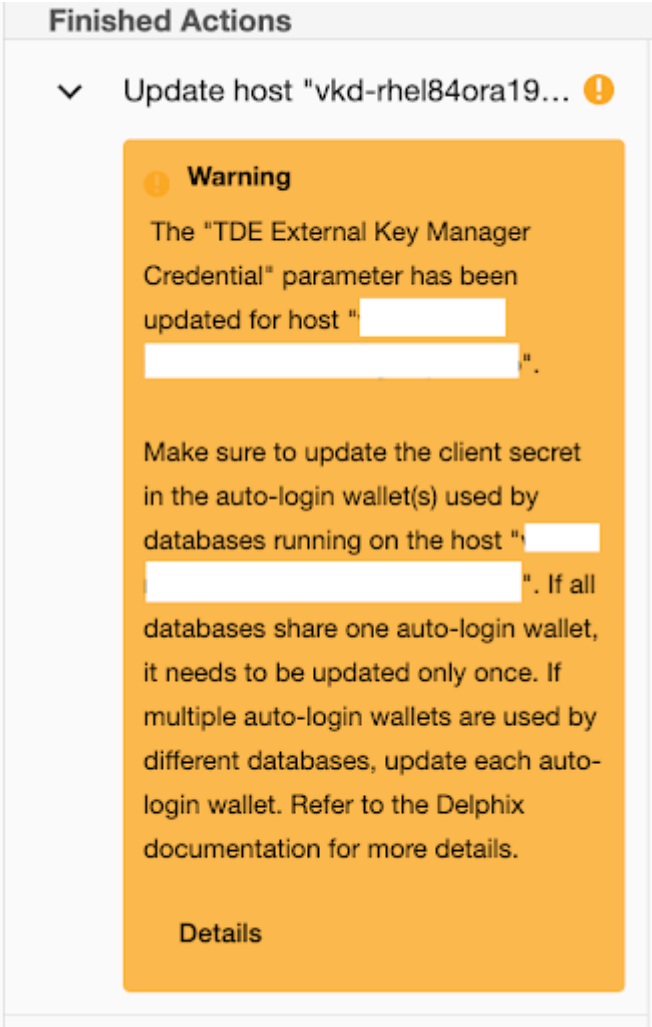
- Log in to the database instance **CDOMSHTGCF36** as a user who has been granted the `ADMINISTER KEY MANAGEMENT` or `SYSKM` privilege.
- Close the external keystore

```
SQL> ADMINISTER KEY MANAGEMENT SET KEYSTORE CLOSE CONTAINER = ALL;
keystore altered.
```

- Change the Oracle Key Vault endpoint password. The following example shows how to use the **okvutil changepwd** command to change the endpoint password. When you are prompted to create the new password, enter a password that is between 8 and 30 characters.

```
[oracle@ip~]$ $OKV_HOME/bin/okvutil changepwd -l $OKV_HOME/ssl -t WALLET
Enter new wallet password:
Confirm new wallet password:
Wallet password changed successfully
```

- Update the TDE Key Manager Credential in Delphix by following the procedure documented in Adding or Editing the TDE External Key Manager Credential (see page 1596). A job warning will be displayed to update the client secret of the autologin wallet.



- The old Oracle Key Vault endpoint password was stored in a [local] auto-open wallet within the `WALLET_ROOT/tde` directory. To update the password, use the following syntax:

```
SQL> ADMINISTER KEY MANAGEMENT UPDATE SECRET 'Test@123' FOR CLIENT
'OKV_PASSWORD' TO AUTO_LOGIN KEYSTORE '/work/oracle/tde';
keystore altered.
```

- The client secret name for Oracle Key Vault is **OKV_PASSWORD**, and in the case of a Hardware Security Module, it will be **HSM_PASSWORD**.
- Open the external keystore

```
SQL> ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN FORCE KEYSTORE IDENTIFIED BY
"Test@123" CONTAINER =ALL;
keystore altered.
```

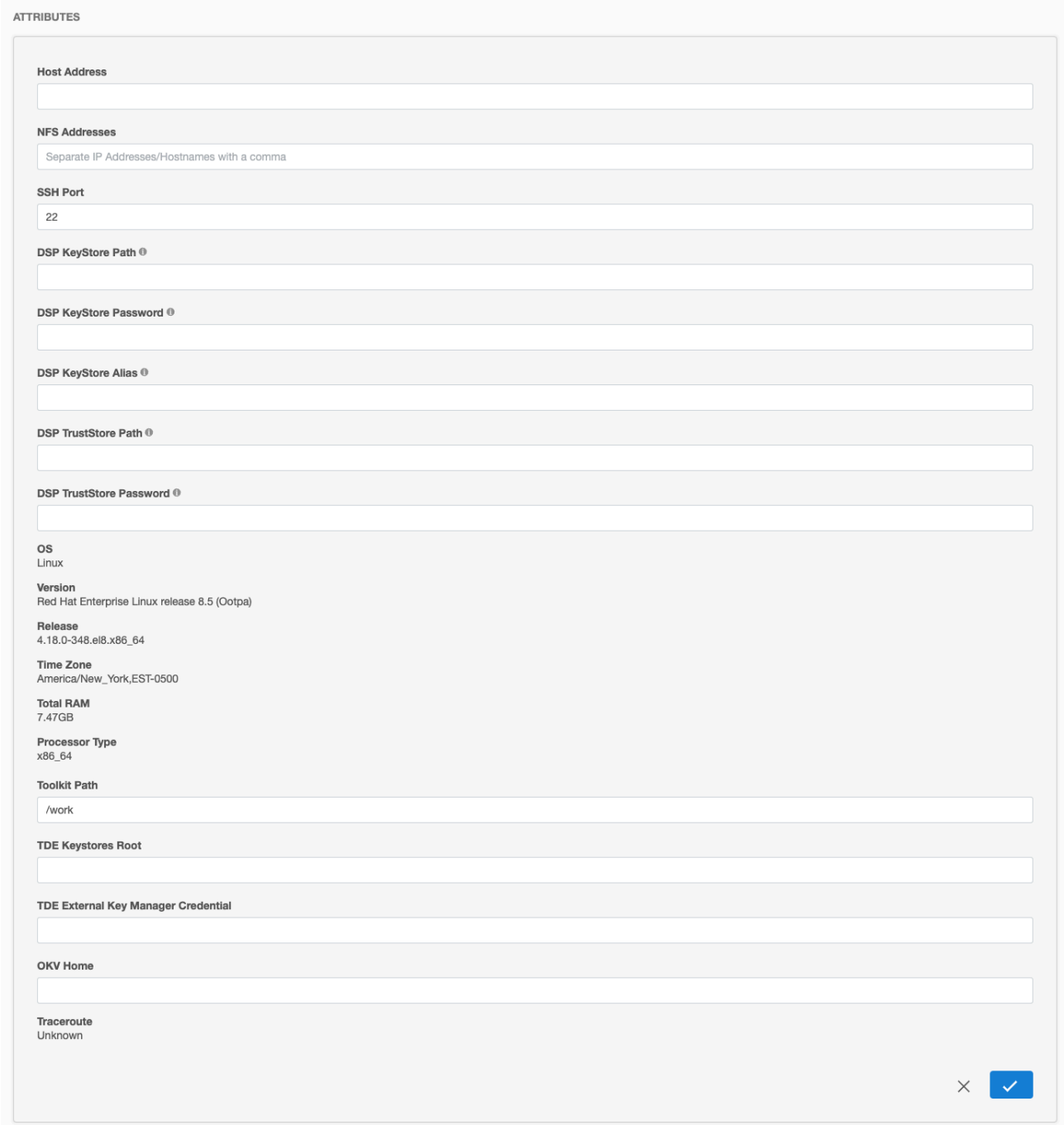
The above steps are also valid when using TDE HSM-enabled vPDB. After executing the aforementioned steps, the Delphix workflows for the vPDB should function correctly.

9.7.10.13 Adding or Editing TDE External Key Manager Credential

The Target **TDE External Key Manager Credential** can be specified either in the GUI Host Detail tab under the Attribute section or in the GUI Databases tab under **Environments**. The value specified at the Database tab under **Environments** will take precedence.

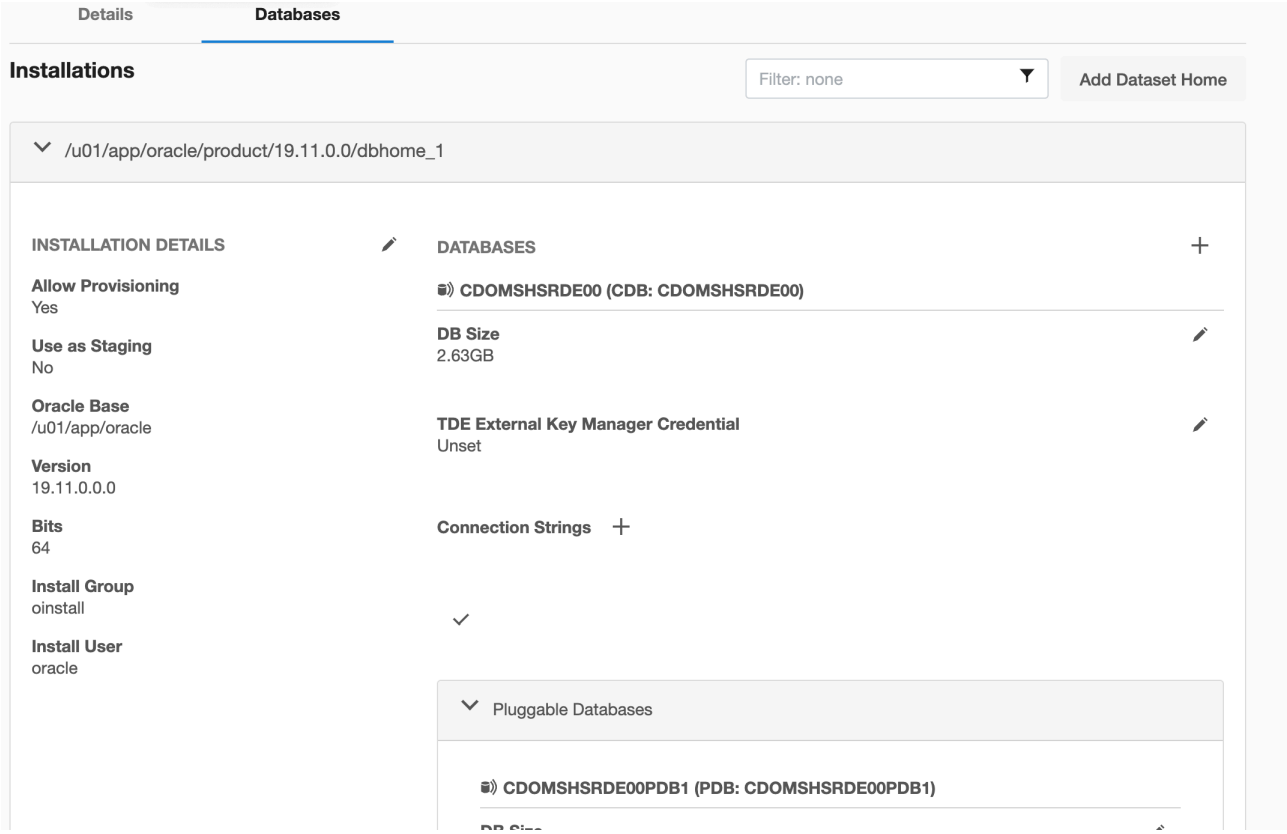
Edit target **TDE External Key Manager Credential** in the GUI in the Details tab under **Environments**:

1. Login to the **Delphix Management** application.
2. Click **Manage** → **Environments**.
3. Click the **Details** tab for your target environment.
4. Next to **Attributes**, click on the pencil icon to set or update attributes, including the TDE External Key Manager Credential.



Edit target TDE External Key Manager Credential in the GUI in the Database tab under Environments:

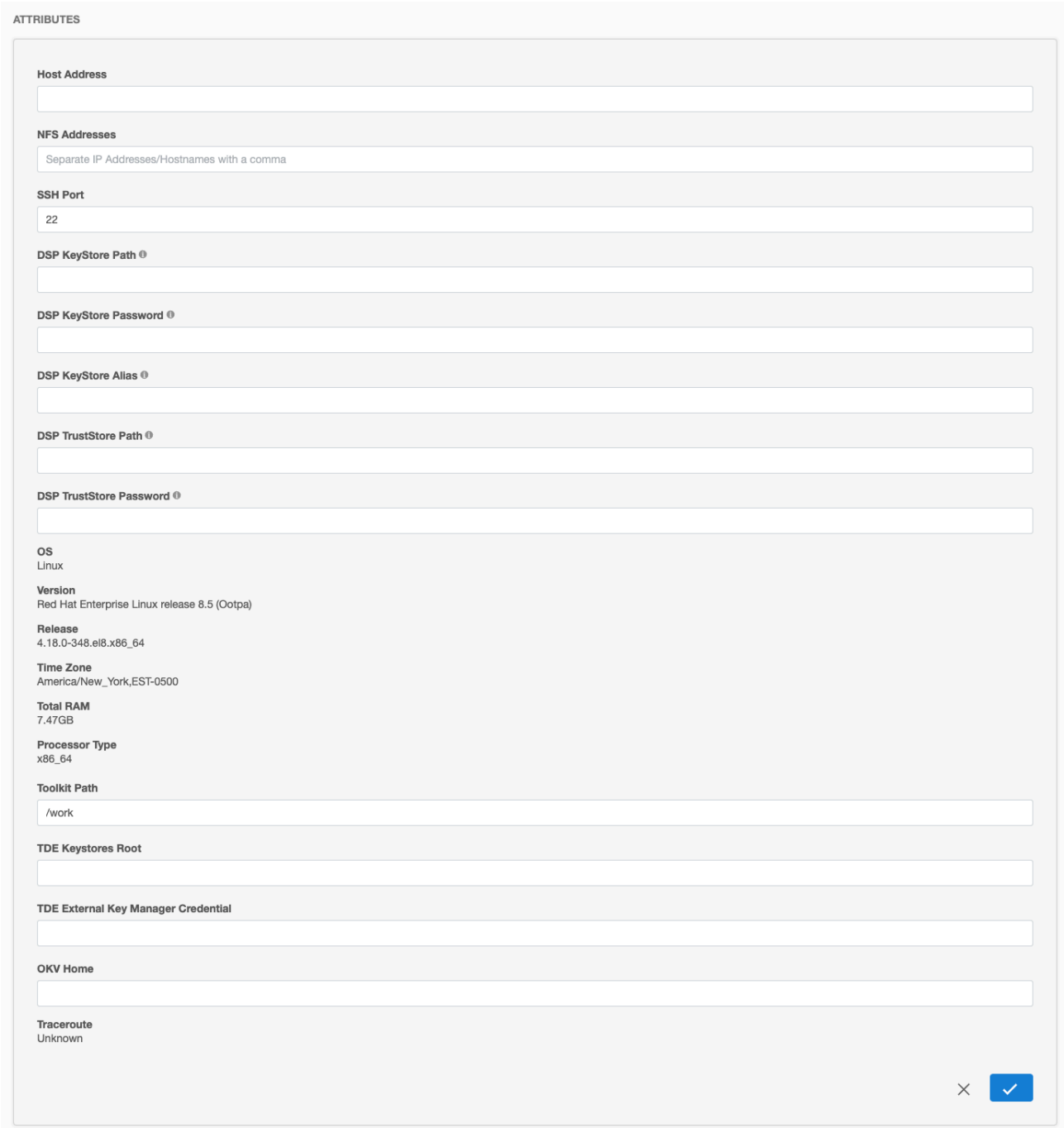
1. Login to the **Delphix Management** application.
2. Click **Manage** → **Environments.**
3. Click the **Databases** tab for your Environment.
4. Next to **TDE External Key Manager Credential** click on the pencil icon to set or update the credential.



9.7.10.14 Adding or Editing the OKV Home

The OKV Home directory path is specified in the GUI in the Details tab under Environments:

1. Login to the **Delphix Management** application.
2. Click **Manage** → **Environments**.
3. Click the **Details** tab for your target environment.
4. Next to **Attributes** click on the pencil icon to set or update attributes, including the OKV Home.

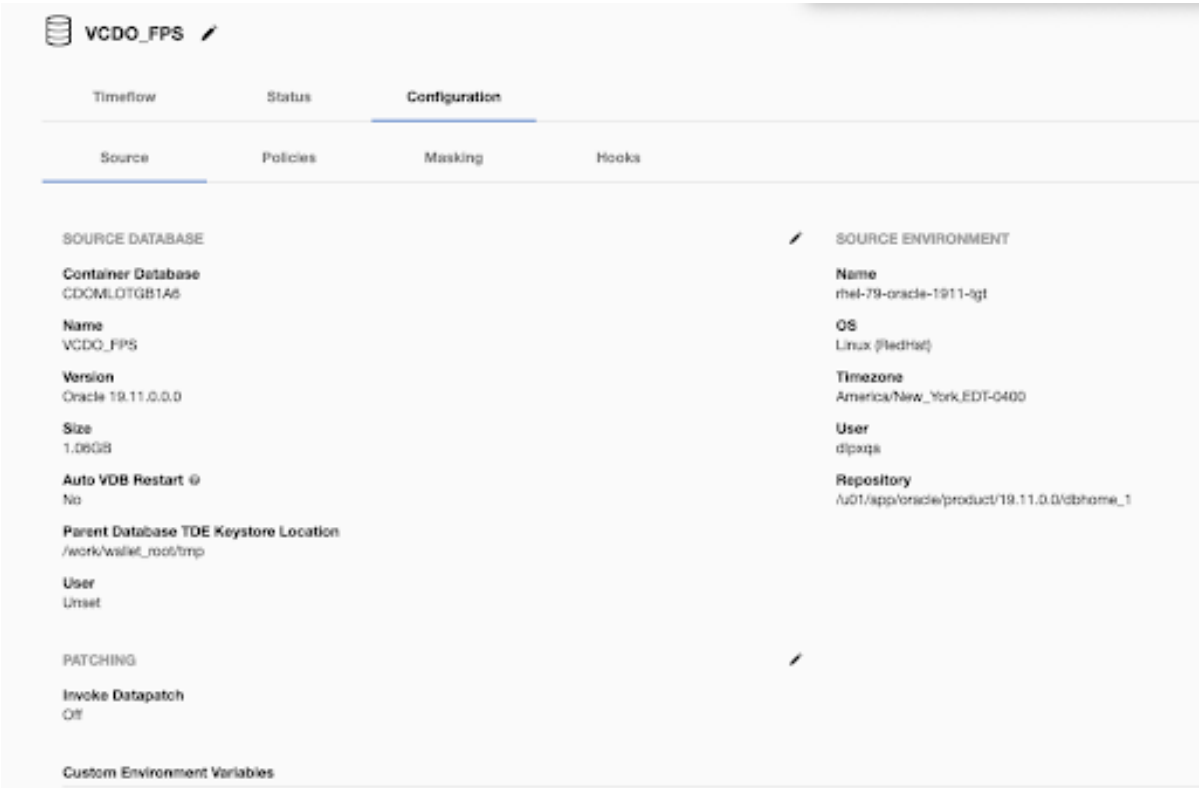


9.7.10.15 Updating the Parent Database TDE Keystore Location

The Parent Database TDE Keystore Location is specified in the GUI in the Configuration tab under the vPDB in the Source tab.

1. Login to the **Delphix Management** application.
2. Click **Manage > Datasets.**

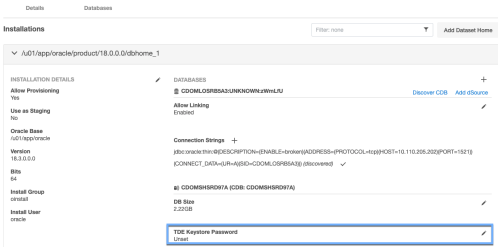
3. Click on the vPDB.
4. Click the **Configuration** tab.
5. Click the **Source** tab.
6. Next to Source Database, click on the pencil icon to update the **Parent Database TDE Keystore Location**.
7. Click on the checkmark to save your changes.



9.7.10.16 Adding or Editing the Target Keystore Password

The target keystore password is specified in the GUI in the Databases tab under Environments:

1. Login to the **Delphix Management** application.
2. Click **Manage > Environments**.
3. Click the **Databases** tab for your Environment.
4. Next to **TDE Keystore Password** click on the pencil icon to set or update the password.

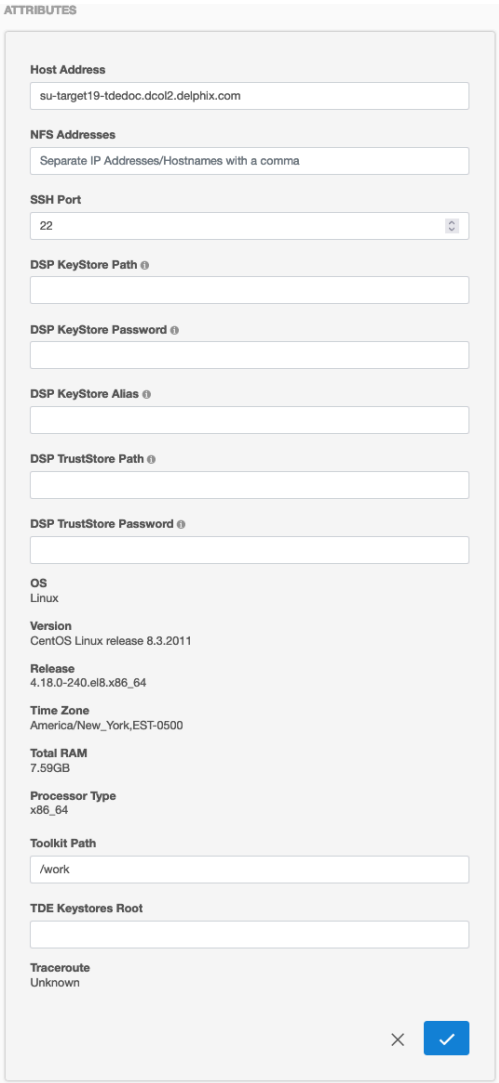


9.7.10.17 Adding or Editing the TDE Keystores Root

The TDE keystores root directory path is specified in the GUI in the Details tab under Environments:

1. Login to the **Delphix Management** application.
2. Click **Manage** → **Environments**.
3. Click the **Details** tab for your target environment.

Next to **Attributes** click on the pencil icon to set or update attributes, including the keystores root.



9.7.10.18 Timeflows for RAC provisioning of VDBs

9.7.10.18.1 Overview

This topic describes special considerations when provisioning by timestamp from a RAC time flow.

Timestamps in Oracle RAC time flows can be imprecise because of time skew among the hosts in a RAC configuration. The time stamps will generally track the host with the fastest clock. For this reason, provisioning by a timestamp may not leave the VDB provisioned at the exact time desired. The provision by SCN should be used if more fine-grained control is required when provisioning.

9.7.10.18.2 Introduction

The Delphix Engine provides the ability to link to an external database by creating a dSource within the Delphix system. Once linked, the Delphix Engine maintains a complete history of the database as part of a Timeflow, limited by the retention policies configured by the administrator. From any time within that

Timeflow, you can provision a virtual database (VDB) from the Delphix Engine. This Timeflow is maintained through the use of SnapSync and LogSync.

The SnapSync operation pulls over the complete data set of the external database during the initial load. Subsequent SnapSync operations pull and store only incremental changes. At the end of each SnapSync operation, the Delphix Engine creates a snapshot that serves as the base point for provisioning operations. In addition, LogSync periodically connects to the host(s) running the source database and pulls over any log files associated with the database. These log files are stored separately from the SnapSync data and are used to provision from points in between SnapSync snapshots. Usually, SnapSync operates against a live database with changes actively being made to it. Hence the data that it pulls over is “fuzzy” and logs must be applied to the data to make it consistent and provisionable. If LogSync is enabled, SnapSync relies on it to copy the logs over. If LogSync is not enabled, SnapSync copies the logs itself. Occasionally, LogSync or SnapSync is not able to retrieve one or more log files from the database. This creates a break in the Timeflow or can prevent a snapshot from being provisioned. To remedy this situation, the Delphix Engine has tools to repair, or patch, a snapshot and the Timeflow.

9.7.10.18.3 Snapshot repair

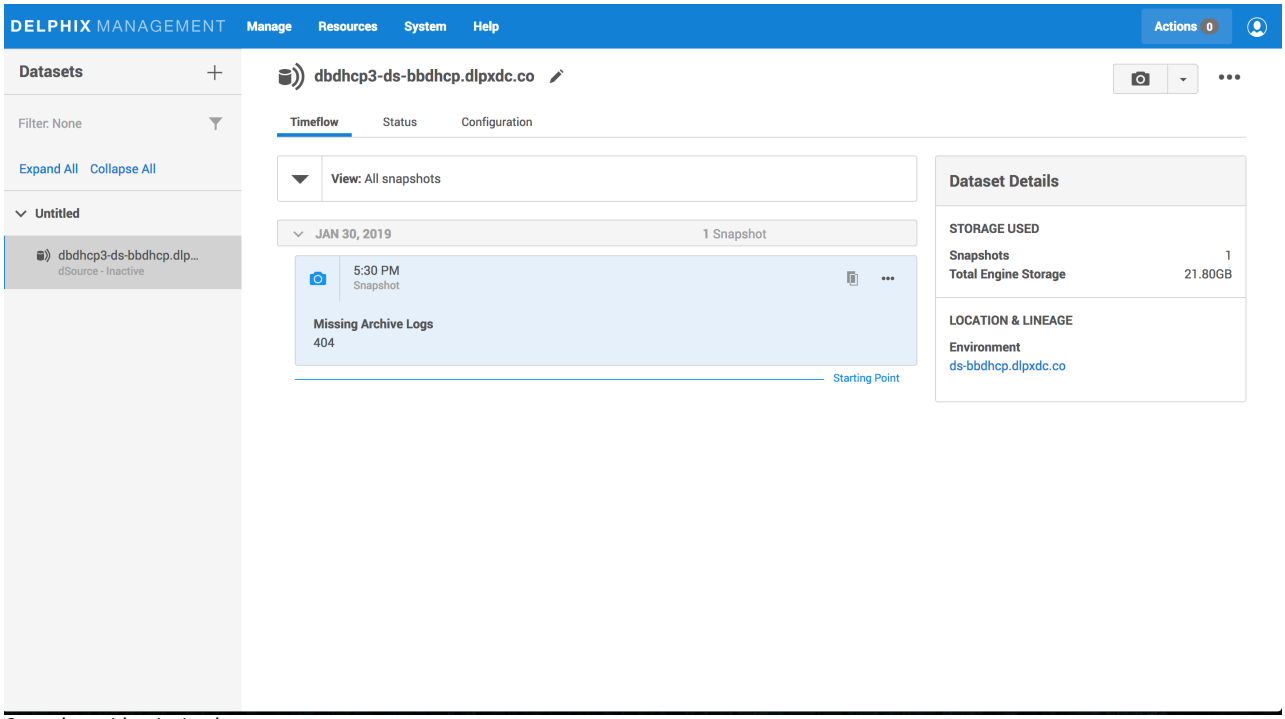
ASM

The steps below do not apply if your archive logs are stored on ASM. If they are stored on ASM, you must move the archived logs to a supported filesystem directory.

When missing log files prevent the Delphix Engine from provisioning a snapshot, you can use the Delphix Management application to identify the missing logs and repair the snapshot. The Delphix Engine will generate a fault whenever missing logs prevent a snapshot from being provisionable. The fault will likely have the title "Cannot provision database from snapshot" and will contain a description of the cause. The common causes are:

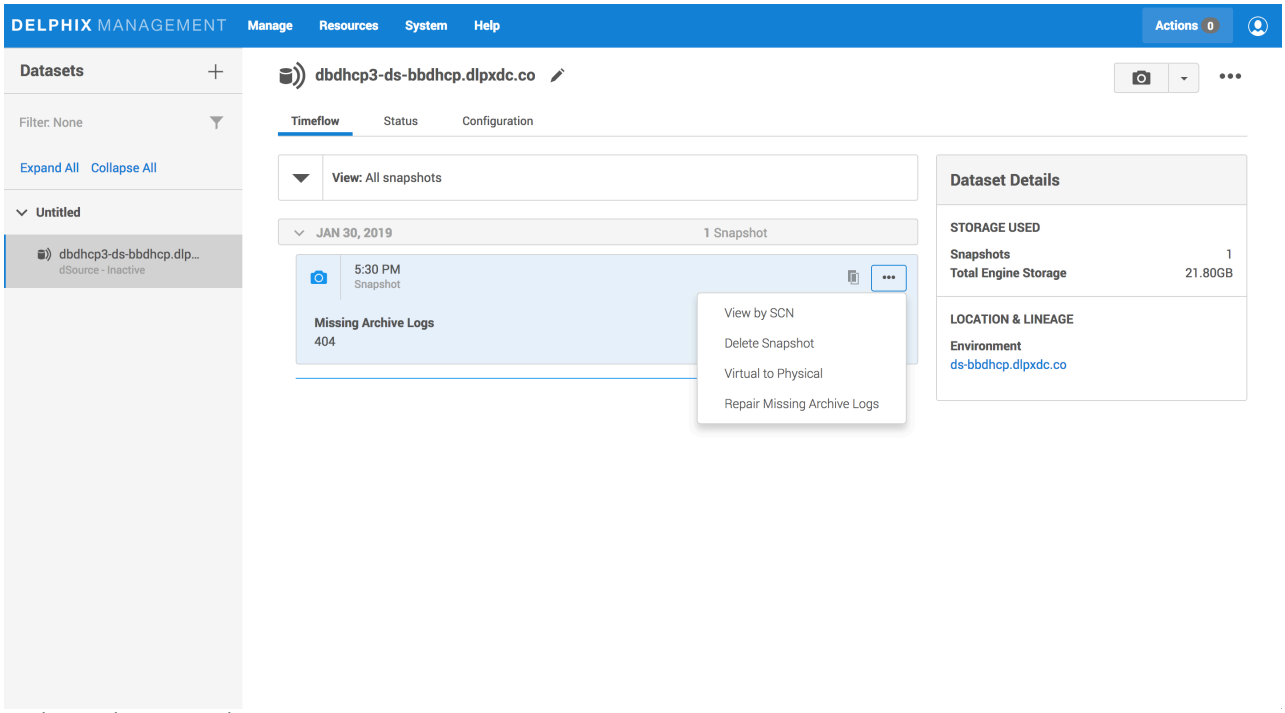
- Logs were deleted/moved/archived from the database before the Delphix Engine could retrieve them. In this case, the archive log retention policy on the source database may be too aggressive. Use the GUI snapshot repair tool to fetch the logs.
- LogSync is still fetching the logs. SnapSync is relying on LogSync to fetch the logs needed to make the snapshot consistent. SnapSync normally will wait up to 15 minutes for LogSync to fetch the logs. If LogSync has not fetched the logs by then, SnapSync will generate a fault and finish. The best course of action, in this case, maybe to wait for LogSync to fetch the logs.
- The source database is a physical standby in real-time apply mode. The changes described in the current online log of the database are needed to make the snapshot consistent. LogSync cannot retrieve the log until it is archived, and SnapSync cannot force the log to be archived because the source database is a physical standby. Force a log switch on the primary database or wait until the log is naturally archived.

Below is a screenshot of a snapshot with missing logs. Clicking on the snapshot causes the list of missing log(s) to appear. In this example, log sequence 404 is missing.

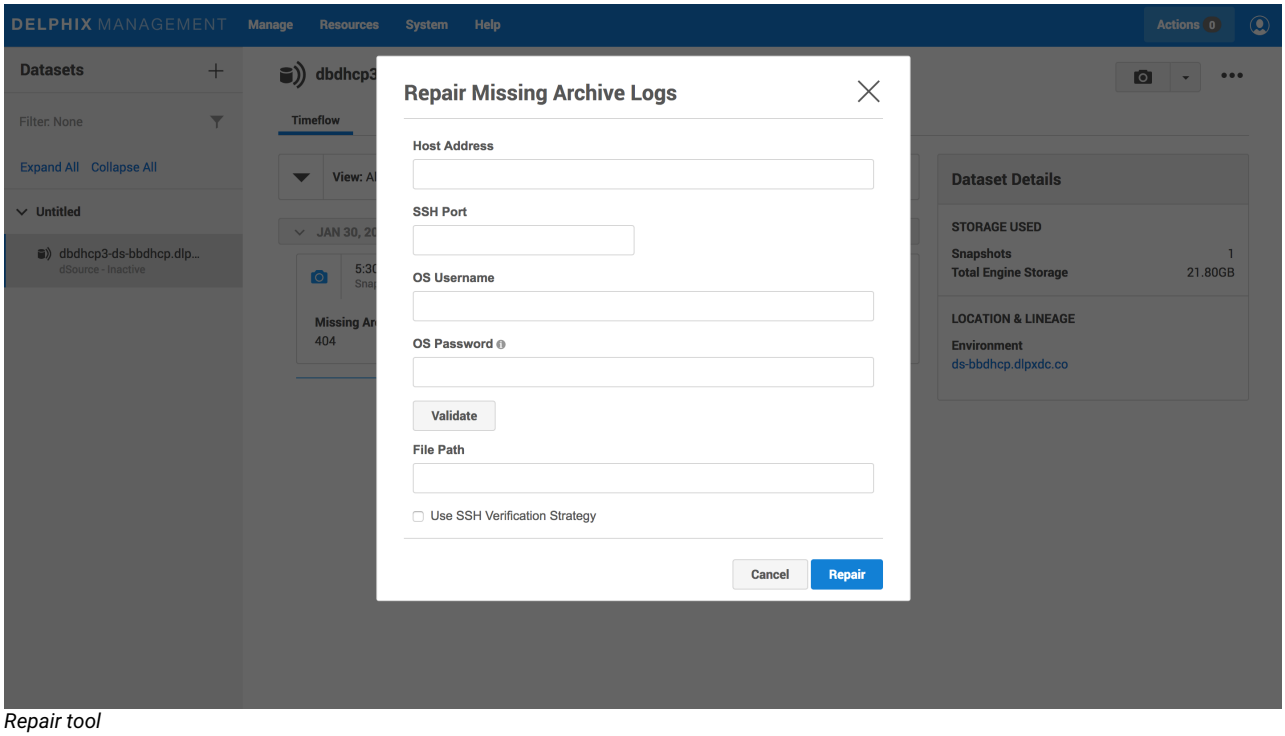


Snapshot with missing logs

If the snapshot can be repaired by fetching the logs from the source database, then you can use the GUI snapshot repair tool to fetch the logs. Hovering over the snapshot exposes more options ("**...**") which can then be clicked to show the **Repair Missing Archive Logs** option. Clicking on the option starts the repair tool.



Selecting the repair tool



Repair tool

To use the snapshot repair tool, as seen above:

1. Enter a **Hostname**. This should be the host from which to retrieve the log(s).
2. Enter a **Username** and **Password**. These should be the credentials for a user who can read the archived log file(s). The user credentials are optional if the host and user credentials have already been added to the Delphix Engine.
3. Enter a **File Path**. This should be the name of the directory containing the missing log(s).

If more than one file is missing, they should all exist in the directory specified by **File Path**. The tool will read every file in the **File Path** directory, so it is best that it only contains the files that are to be retrieved.

9.7.10.18.4 Timeflow patching

When missing log files cause a break in the Timeflow, you can use the command-line interface (CLI) to identify the missing logs and patch the Timeflow. The Delphix Engine will generate a fault whenever there are missing logs on a portion of the Timeflow. The fault will likely have the title “Cannot provision a database from a portion of Timeflow” and will contain a description of the cause. The most common cause is an overly aggressive archive log retention policy on the source database causing a log to be deleted before LogSync can fetch it. Other faults can also be generated by describing the specific errors encountered when fetching the log(s).

You can use the CLI to list the missing logs and patch the Timeflow. The following CLI Cookbook entry demonstrates how to do this: CLI Cookbook: Repairing a Timeflow (see page 2656)

If you delete or move archivelogs from the source database that are not needed for a snapshot, you still may need to repair the TimeFlow to provision using LogSync. In this case, an icon will not be visible on the TimeFlow tabs. This means you cannot repair the TimeFlow in the GUI. However, you can still repair it using the CLI.

9.7.10.19 Upgrading an Oracle VDB, linked CDB, or a vCDB

9.7.10.19.1 Prerequisites

Upgrading a VDB involves Oracle level activity and Delphix GUI/CLI level activity. The Oracle activity, to be performed first, involves upgrading/patching current oracle homes, installing new oracle homes, and managing init.ora/spfile.ora files. The Delphix GUI/CLI activity, to be performed second, involves rediscovering new oracle installations, giving a VDB a new oracle home, and updating the oracle grid home. These are discussed in detail below.

> During an Oracle upgrade, refreshing an Environment will generally discover new Oracle installations, allowing for a virtual database (VDB) or dSource upgrade to be handled through the UI. However, if the Oracle Grid home was changed, due to an Oracle upgrade, the crsClusterHome parameter will need to be updated manually through the command line (CLI).

9.7.10.19.2 Limitations

Currently, it is not possible to convert an existing VDB into a vPDB.

9.7.10.19.3 PSU/Oracle upgrade procedure

Normally a PSU or Oracle upgrade will have both binary changes and some scripts to run on the database side as well.

There are 3 ways to apply a PSU/Oracle upgrade:

1. Apply to existing ORACLE_HOME. You must be on Delphix version 4.1.x or higher to do this.
2. Create a new ORACLE_HOME (could clone the existing one) and then apply the PSU to the new ORACLE_HOME
3. After a dSource is upgraded, use refresh on the Timeflow tab to upgrade the VDB

Follow Oracle documentation and run the appropriate script(s) and/or steps on the databases using those ORACLE_HOMEs. In option B, stop the instance using the old ORACLE_HOME, then restart the instance with the new ORACLE_HOME from the command line as normal.

9.7.10.19.4 Applying to an existing ORACLE_HOME

1. Following Oracle documentation, patch the ORACLE_HOME, then the database for the VDB(s).
2. Refresh the environment the VDBs are on in the Delphix Management application.

9.7.10.19.5 Creating a new ORACLE_HOME

1. Refresh the environment from the Delphix Management application. Verify that the new ORACLE_HOME is picked up and displayed in the **Databases** tab as an ORACLE Installation.
2. Stop the VDB instance (old ORACLE_HOME) using Oracle tools. Do not use the Delphix VDB **stop** operation as the VDB should be stopped outside of Delphix.
3. Export ORACLE_HOME=(newORACLE_HOME). Follow Oracle documentation to patch the database.
4. Copy the init.ora for that VDB in this new $ORACLE_HOME/dbs directory. The delphix_os user will need the write permissions to this directory.
5. If there are any database parameter changes, update the spfile located on the Delphix mount base with those values.
6. Navigate to the **Datasets** view
7. Expand the group(s) containing all non-multitenant and multitenant VDBs.
8. Click the **Configuration** tab.
9. From the Actions menu (...) select **Upgrade** to switch the ORACLE_INSTALLATION to the new one.
10. If the database is a linked CDB or vCDB, go to any child vPDBs and verify that the **Repository** and/or **Version** has been updated under the **configuration** tab.

9.7.10.19.6 Using refresh

1. Refresh the environment from the Delphix Management application.
2. Verify that the new ORACLE_HOME is picked up and displayed in the **Databases** tab of the **Environments** screen as an ORACLE Installation.
3. On the **VDB Configuration tab,** click the **stop** icon to stop the VDB.
4. From the Actions menu (...) select **upgrade** to switch the ORACLE_INSTALLATION to the new upgrade version same as the dSource.
5. Navigate to the **Datasets** view, select the VDB, and then select the **Timeflow** tab.
6. Click the **Refresh** button.
7. Select a new **snapshot** from the dSource that was taken after the dSource was upgraded. (The database version is on the snapshot card.)

Updating the Oracle user after an upgrade
There may be cases when you upgrade the Oracle home and the Oracle User (who owns the binary) is a different user than the previous Oracle User. You will then need to update the Oracle User for each environment, and then re-connect each VDB to the upgraded Oracle home using the new Oracle User.

The new Oracle User must be in the same OS group (for example, dba or oinstall) as the previous one.

1. Login to the Delphix Management application using delphix_admin credentials.
2. Click **Manage**.
3. Select **Environments**.
4. Select the **environment** where you want to add the user.
5. Next to **Environment Users**, click the **Pencil** icon to add the new user.
6. Set the new user as the **default** user.
7. Follow the procedure to upgrade VDBs described in this topic.

9.7.10.19.7 Linked CDB upgrade procedure

For vPDBs in the linked CDBs, there are two ways to upgrade the linked CDB:

A. Perform the Oracle upgrade of the current target CDB.

B. Create an entirely new target CDB of the higher version.

9.7.10.19.8 Performing the Oracle upgrade of the current target CDB

1. Perform the Oracle upgrade of the current target CDB.
2. Login to the Delphix Management application.
3. Navigate to **Manage** > **Datasets** and select the target CDB.
4. From the **Actions** menu (...), select **Upgrade** to switch the ORACLE_INSTALLATION to the new one.
5. Click Upgrade.
6. Under the **Configuration** tab, verify that the **Repository** and/or **Version** has been updated.

9.7.10.19.9 Creating a new target CDB

You can first disable the vPDB, then use the migrate vPDB feature to select a new container database.

9.7.10.19.9.1 Procedure

1. Login to the Delphix Management application.
2. Navigate to **Manage** > **Datasets**.
3. Select the vPDB you want to migrate.
4. From the **Actions** menu (...) select **Disable**.

5. Click **Disable** to confirm.
6. From the **Actions** menu (...) select **Migrate**.
7. Select the new container database for the vPDB, the user for that environment, and the database installation where the container database of the vPDB will reside.
8. Click the **Migrate** option to confirm your selections.
9. Manually copy the exported keys to the target host's toolkit directory.
10. From the **Actions** menu (...) select **Enable**.
11. Click **Enable** to confirm. Your vPDB will re-start in the new environment, and you can continue to work with it as you would any other vPDB.

9.7.10.20 Customizing RAC instances after provision

Provisioning a RAC VDB requires users to select which RAC node to be included for this VDB, as well as the instance number and instance name of each RAC node that runs the RAC VDB. At any time after the provision, there may be a need to downsize or increase the number of instances of the RAC VDB or reassign instance number and instance names to individual VDB RAC node.

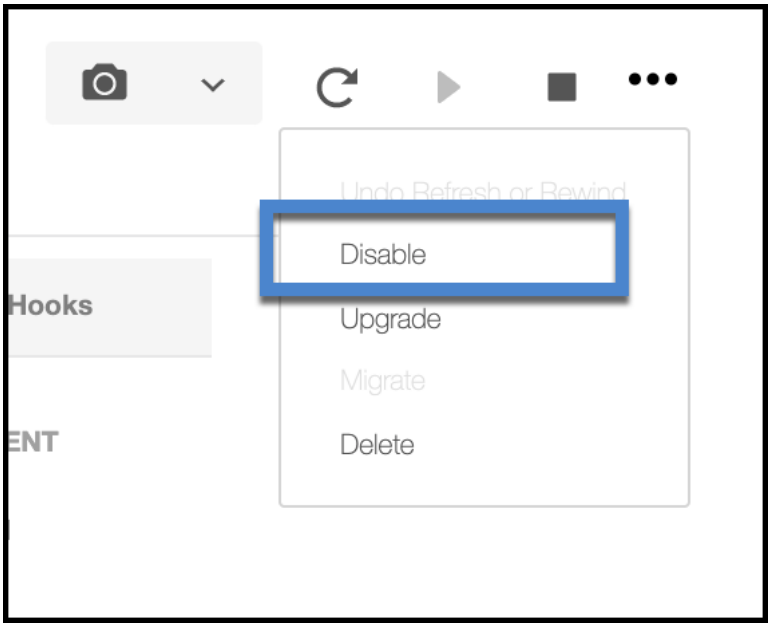
Oracle pluggable database
The current release does not support the customization of RAC vPDB instances after provision.

If you make any changes to the VDB's configuration on the engine either by changing the VDB's configParams using CLI or VDB configuration template that was used to provision it, the Delphix engine will push out these changes when the VDB is next enabled after the RAC instance changes are made.

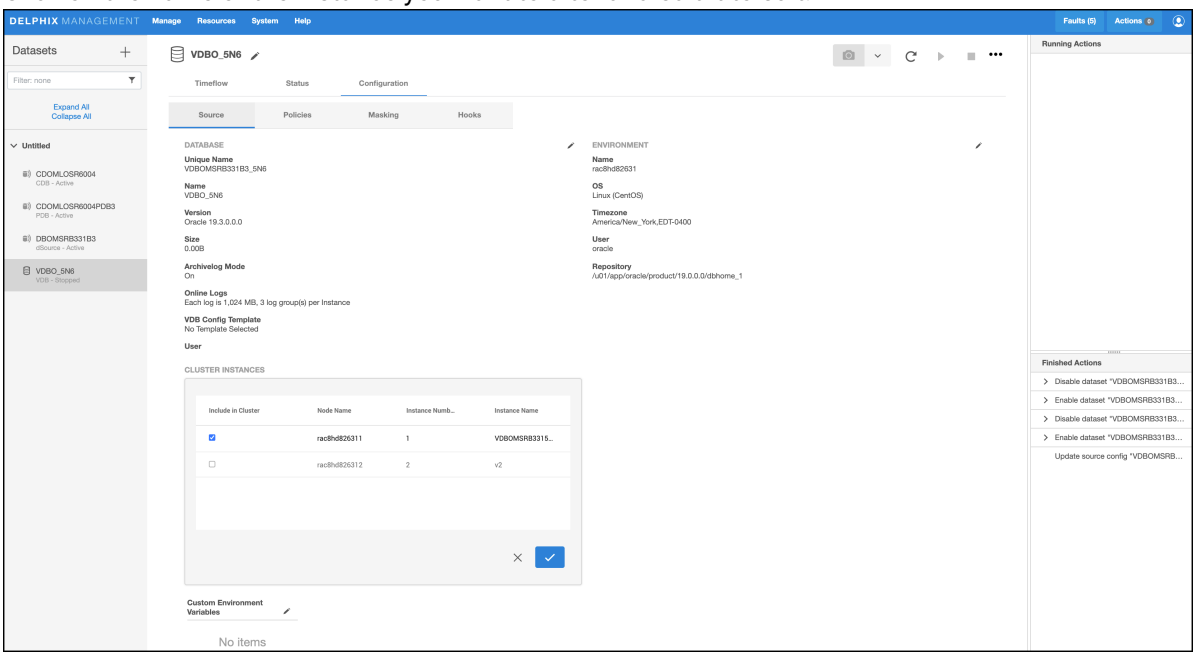
9.7.10.20.1 Procedure

The following steps illustrate how you can customize your RAC VDB instance configuration after provision:

1. Login to your Delphix Management application.
2. Select the VDB you want to edit.
3. From the right-side pane select the **Configuration** tab.
4. From the Action menu (...), select **Disable**.



5. Click on the **pencil** icon next to 'CLUSTER INSTANCES' to be able to edit your instance names.
6. Click on the name of the instance you want to alter and edit it to suit.



7. Check or uncheck the check-box to the left to indicate if you want to add or remove a cluster node for your RAC VDB.
8. Complete editing and then confirm using the tick icon.
9. From the Action menu (...), select **Enable**.

> Disabling a VDB will result in the VDB Instances being shut down and the NFS mounts for that VDB presented from Delphix being dismounted.

9.7.10.21 Migrating an Oracle VDB

This topic describes how to migrate a Virtual Database (VDB) from one target environment to another.

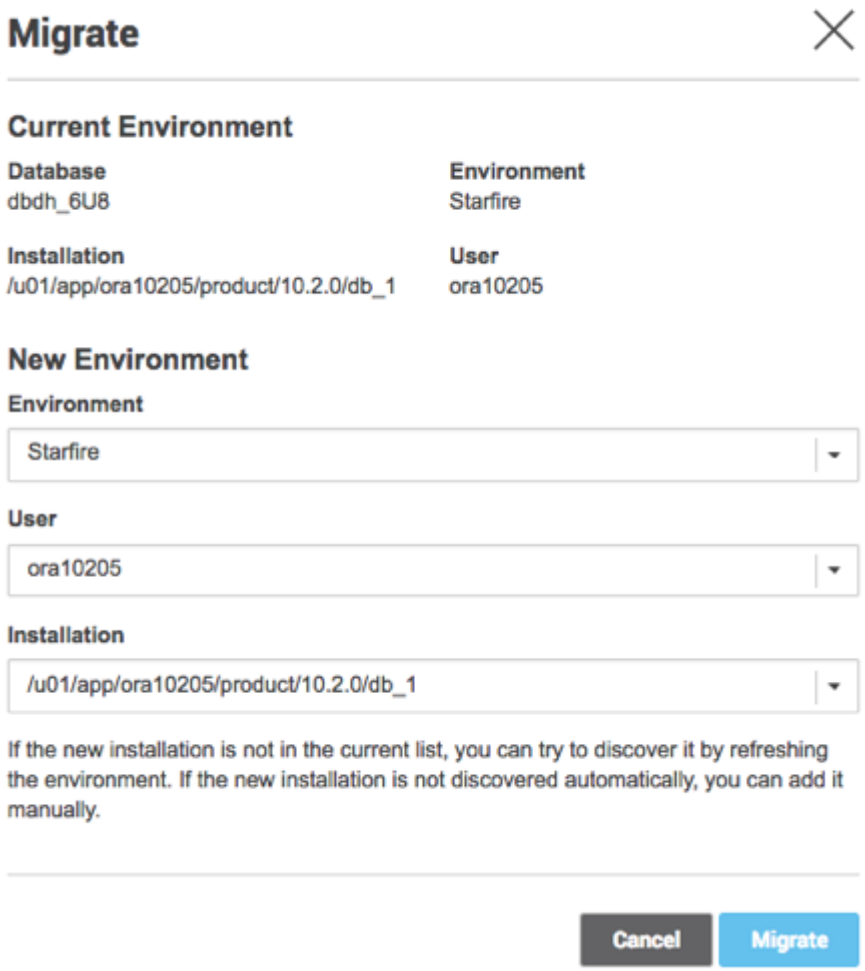
There may be situations in which you want to migrate a virtual database to a new target environment, for example when upgrading the host on which the VDB resides, or as part of a general data center migration. This is easily accomplished by first disabling the database, then using the Migrate VDB feature to select a new target environment.

9.7.10.21.1 Prerequisites

- You should have already set up a new target environment that is compatible with the VDB that you want to migrate.
- You cannot migrate a single instance of Oracle VDB to a RAC environment and vice versa. An additional reconfiguration is needed when converting a single instance to RAC that is only performed during a VDB provision. Instead, you should provision a new VDB.

9.7.10.21.2 Procedure

1. Login to your Delphix Management application.
2. Click **Manage**.
3. Select **Datasets**.
4. Select the **VDB** you want to migrate.
5. From the Actions menu (...) select **Disable.**
6. Click **Disable** to confirm. When the VDB is disabled, its icon will turn grey.
7. From the Actions menu (...) select **Migrate**.
8. Select the new target environment for the VDB, the user for that environment, and the database installation where the VDB will reside.



9. Select **Migrate** to confirm your selections.
10. From the Actions menu (...) select the **Enable**
11. Click **Enable** to confirm.

Within a few minutes, your VDB will re-start in the new environment, and you can continue to work with it as you would any other VDB.

9.7.10.22 Migrating a vPDB

There may be situations in which you want to migrate a virtual pluggable database (vPDB) to a new container database on the same or a different target environment, for example when upgrading the host on which the vPDB resides, or as part of a general data center migration. This is easily accomplished by first disabling the vPDB, then using the Migrate vPDB feature to select a new container database.

9.7.10.22.1 Pre-requisites

- You should already set up and have Delphix discover a container database in the same environment as the vPDB currently is or from an environment to which the vPDB will be migrated to.

- The virtual CDB from/to which vPDB is to be migrated must be already running and open in a read-write mode.

9.7.10.22.2 Procedure

Login to your Delphix Management application.

1. Click **Manage**.
2. Select **Datasets**.
3. Select the **vPDB** you want to migrate.
4. From the Actions menu (...) select **Disable**.
5. Click **Disable** to confirm.
6. From the Actions menu (...) select **Migrate**.
7. Select the new **container database** for the vPDB, the **user** for that environment, and the **database installation** where the container database of the vPDB will reside.
8. Click the **Migrate** to confirm your selections.
9. Manually copy the exported keys to the target hosts toolkit directory. When migrating a TDE enabled vPDB, the disable step results in the vPDB keys being exported to the toolkit. Enabling the vPDB in a different target host will result in plugin violations, therefore when migrating to a new host, the user must manually copy the exported keys to the toolkit directory on the target host.
10. From the Actions menu (...) select **Enable**.
11. Click **Enable** to confirm. Your vPDB will re-start in the new environment, and you can continue to work with it as you would any other vPDB.

9.7.10.23 Migrating a TDE-enabled vPDB

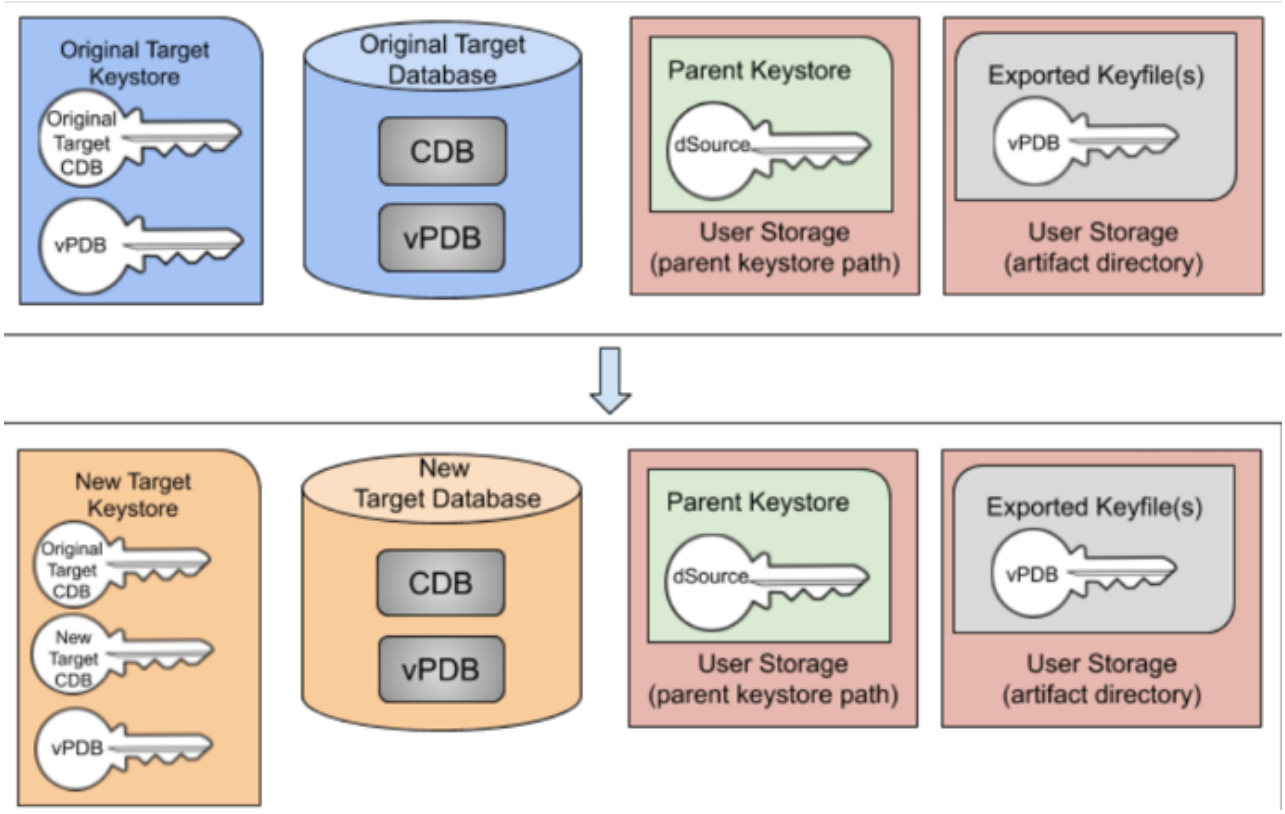
Migrating a TDE-enabled virtual pluggable database (vPDB) from one container database to another container database on the same target host or a different target host involves the following steps:

1. **Disable** the vPDB.
2. **Migrate**.
3. (*Applicable only for software keystore based vPDB*)
   For a software keystore based vPDB, there are additional steps that are required before the enable. The steps are necessary because the keystore and exported keyfiles are not present on Delphix Continuous Data storage. These steps are:
   a. Copy the parent TDE keystore from the original target host to the new target host. If the vPDB is not being migrated to a new target host, then this step is not needed. Similarly, if the new host has the parent keystore in a different location, then the parent keystore path for the vPDB needs to be updated.
   b. Copy the artifact directory for the vPDB from the original target host to the new target host. If the vPDB is not being migrated to a new host, then this step is not needed.

   c. Ensure that the new target container has the TDE keystore password set.
   d. Merge the original target keystore into the new target keystore. This is required to support a rewind operation to a snapshot taken before the migrate.

4. (*Applicable only for OKV-enabled or HSM-enabled vPDB*)
   For a TDE OKV or HSM-enabled vPDB, additional steps are required before enabling. These steps are essential because the exported keyfiles are not present on Delphix Continuous Data Storage:
   a. Provide access to the vPDB master encryption key on the new target endpoint (refer to Oracle Key Vault documentation[465] or CipherTrust Manager documentation[466]).
   b. Copy the artifact directory for the vPDB from the original target host to the new target host.
   c. Ensure that the new target container has the **TDE Key Manager Credential** set.
5. **Enable**.

The diagram below illustrates the scenario of a migrated TDE-enabled vPDB.



5 Migrated TDE-enabled vPDB (software-based key store)

9.7.10.23.1 Example for TDE software keystore based vPDB

Consider the following TDE software keystore based vPDB:

465 https://docs.oracle.com/en/database/oracle/key-vault/21.7/okvag/okv_wallets_and_keys.html#GUID-2B541778-EF4E-4D3A-B251-377FBF4490C7

466 https://www.thalesdocs.com/ctp/cm/2.14/admin/cm_admin/abac-permissions/domains/index.html

This vPDB is currently provisioned to the container database `CDOMLOTG9620` on the host `tde-target18`. Connecting to that database, you can see the keys for the target container and the vPDB:

```
SQL> show pdbs
CON_ID CON_NAME            OPEN MODE  RESTRICTED
------ --------------------- ---------- ----------
     2 PDB$SEED            READ ONLY  NO
     3 CDOMLOTG9620PDB1        READ WRITE NO
     4 CDOMLOTG9620PDB2        READ WRITE NO
     5 CDOMLOTG9620PDB3        READ WRITE NO
     6 TDE_VPDB            READ WRITE YES
SQL> select con_id, key_id from v$encryption_keys;
CON_ID KEY_ID
------ ----------------------------------------------------
     1 ASFwmcfaMk8vv1LvzV0H8BEAAAAAAAAAAAAAAAAAAAAAAAAAAAA
     6 AdDdKibLKU9mv6PDAIvVvH0AAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

The key for the target container starts with **ASFwmcfa** and the vPDB key starts with **AdDdKibL**. There is also an artifact directory for this vPDB found in `<toolkit_directory>/oracle_tde_keystores`:

```
$ ls /toolkit/oracle_tde_keystores/
tde_vpdb_Encrypted_e9d2befb-9849-43c8-85f5-5ee8b760e334
```

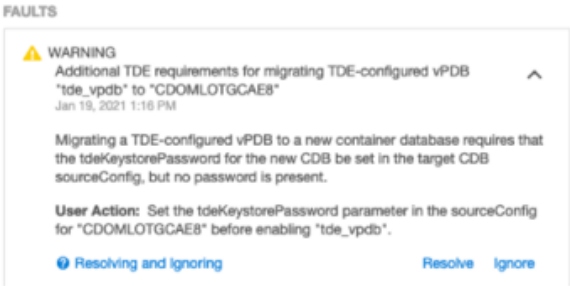
Migrate this vPDB to the container database `CDOMLOTGCAE8` located on the host `tde-target18b`. Currently, that container has the following configuration:

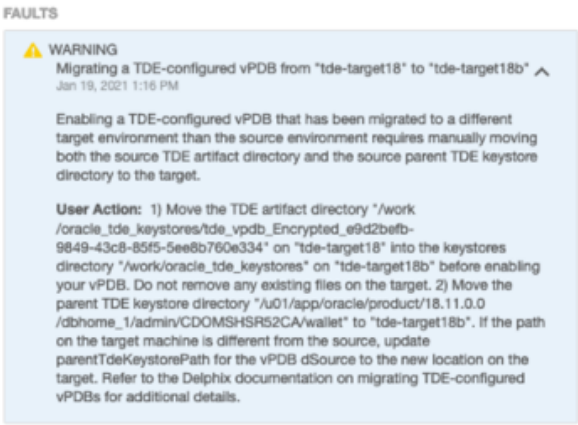
```
SQL> show pdbs
CON_ID CON_NAME            OPEN MODE  RESTRICTED
------ --------------------- ---------- ----------
     2 PDB$SEED            READ ONLY  NO
     3 CDOMLOTGCAE8PDB1        READ WRITE NO
     4 CDOMLOTGCAE8PDB2        READ WRITE NO
     5 CDOMLOTGCAE8PDB3        READ WRITE NO
SQL> select con_id, key_id from v$encryption_keys order by con_id;
CON_ID KEY_ID
------ ----------------------------------------------------
     1 Ad344VSUDk/jv1VDb1QNBHMAAAAAAAAAAAAAAAAAAAAAAAAAAAA
     3 AetaL+IKpE+ov5avXouApwUAAAAAAAAAAAAAAAAAAAAAAAAAAAA
     4 AXZTgaBbxk+6v6x2yHQ4ArgAAAAAAAAAAAAAAAAAAAAAAAAAAAA
     5 AayTgbjkJE/3v82/hBBBAWEAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

The new target container has a CDB key starting with **Ad344VSU**. The migration steps are as follows:

1. Disable the vPDB in the original container on the host `tde-target18a`.

2. Migrate the vPDB to point to the new container on the new target host `tde-target18b`. The UI will report the following warnings during the migrate operation, which indicate the manual steps which need to be done:





3. Set the TDE Keystore Password for the new target container `CDOMLOTGCAE8` on the new target host `tde-target18b`.
4. Copy the parent keystore to the new target host `tde-target18b`. In this case, keep the existing location `/u01/app/oracle/product/18.11.0.0/dbhome_1/admin/CDOMSHSR52CA/wallet` on both hosts, so the configuration does not need to be updated.
5. Copy the TDE artifact directory `/toolkit/oracle_tde_keystores/tde_vpdb_Encrypted_e9d2befb-9849-43c8-85f5-5ee8b760e334` to the new target host `tde-target18b`. If there are no TDE-enabled vPDBs on the new target host, the `oracle_tde_keystores` directory may need to be created first.
   a. On the new target host, check if `/toolkit/oracle_tde_keystores` exists. Create it, if not. While creating the directory, you should make sure the group ownership is set to `oinstall` and the directory's permissions are set to `0770`. For example:

   ```
   mkdir -p /toolkit/oracle_tde_keystores
   chgrp oinstall /toolkit/oracle_tde_keystores
   chmod 0770 /toolkit/oracle_tde_keystores
   ```

   b. Copy the TDE artifact directory from the old target host to the new target host, preserving file permissions. For example, if using `scp` on the new target host:

   ```
   scp -rp <old_target_hostname>:/toolkit/oracle_tde_keystores/
   tde_vpdb_Encrypted_e9d2befb-9849-43c8-85f5-5ee8b760e334 /toolkit/
   oracle_tde_keystores/
   ```

c. If the copied files are now owned by a user that is not the Oracle home owner, add group-write permissions to key export files to prevent key imports from failing with `ORA-46646: file from which keys are to be imported is invalid`. For example:

```
find /toolkit/oracle_tde_keystores/tde_vpdb_Encrypted_e9d2befb-9849-43c8
-85f5-5ee8b760e334 -name "*.keys" -exec chmod g+w {} \;
```

6. Merge the original target keystore into the new target keystore. Oracle provides the `ADMINISTER KEY MANAGEMENT MERGE KEYSTORE` command to facilitate this. Note that you cannot successfully merge into a keystore that is currently in use by the database, so the recommended process is to first copy the original target keystore to the new host, merge the two keystores into a new keystore, replace the existing new target keystore and finally bounce the CDB to start using the merged keystore:
   a. Copy the original keystore to `/tmp/tde-target18a` on `tde-target18b`.
   b. Create a temporary directory `/tmp/merged` on `tde-target18b.`
   c. While connected to the new container on `tde-target18b`, issue the merge keystore command:

```
SQL> administer key management merge keystore '/tmp/tde-target18a'
 identified by
  *** and keystore '/u01/app/oracle/product/18.11.0.0/dbhome_1/admin/
CDOMLOTGCAE8/wallet'
  identified by *** into new keystore '/tmp/merged' identified by ***;
```

   d. Copy the new merged keystore `/tmp/merged/ewallet.p12` into the existing keystore location on `tde-target18b` (backing up the existing keystore first).
   e. If there is an autologin wallet configured for the container, it must be recreated:

```
SQL> administer key management create auto_login keystore from keystore
 '/tmp/merged' identified by ***;
```

   Copy the new merged autologin keystore `/tmp/merged/cwallet.sso` into the existing keystore location on `tde-target18b` (backing up the existing autologin keystore first).
   f. Shutdown and restart the new target container database on the host `tde-target18b`.

7. Confirm that the keystore now contains both the original key and the new key:

```
SQL> select key_id from v$encryption_keys where con_id = 1;
KEY_ID
----------------------------------------------------
ASFwmcfaMk8vv1LvzV0H8BEAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Ad344VSUDk/jv1VDb1QNBHMAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

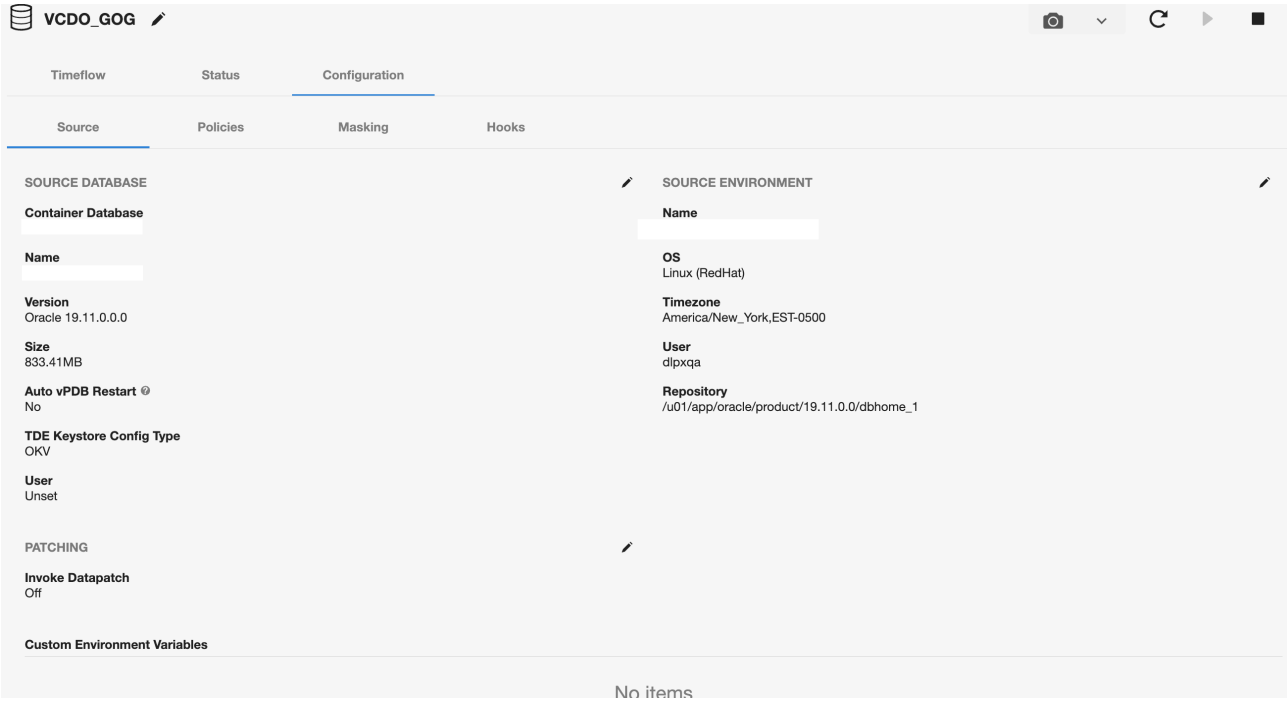
Have the key starting with **ASFwmcfa** from the original container and the key starting with **Ad344VSU** from the new container.

8. Enable the vPDB to complete the migration operation. The vPDB is successfully started in the new container, with its key starting with **AdDdKibL**:

```
SQL> show pdbs
CON_ID CON_NAME            OPEN MODE  RESTRICTED
------ ------------------------------ ---------- ----------
     2 PDB$SEED            READ ONLY  NO
     3 CDOMLOTGCAE8PDB1        READ WRITE NO
     4 CDOMLOTGCAE8PDB2        READ WRITE NO
     5 CDOMLOTGCAE8PDB3        READ WRITE NO
     6 TDE_VPDB            READ WRITE NO
SQL> select con_id, key_id from v$encryption_keys order by con_id;
CON_ID KEY_ID
------ ----------------------------------------------
     1 Ad344VSUDk/jv1VDb1QNBHMAAAAAAAAAAAAAAAAAAAAAAAAAAAA
     1 ASFwmcfaMk8vv1LvzV0H8BEAAAAAAAAAAAAAAAAAAAAAAAAAAAA
     3 AetaL+IKpE+ov5avXouApwUAAAAAAAAAAAAAAAAAAAAAAAAAAAA
     4 AXZTgaBbxk+6v6x2yHQ4ArgAAAAAAAAAAAAAAAAAAAAAAAAAAAA
     5 AayTgbjkJE//v82/hBBBAWEAAAAAAAAAAAAAAAAAAAAAAAAAAAA
     6 AdDdKibLKU9mv6PDAIvVvH0AAAAAAAAAAAAAAAAAAAAAAAAAAAA
6 rows selected.
```

9.7.10.23.2 Example for TDE OKV-enabled vPDB

Consider the following TDE OKV-enabled vPDB:



This vPDB is currently provisioned to the container database `CDOMSHTGCF36` on the host `tde-okv-target19`. Connecting to that database, you can see the keys for the target container and the vPDB:

```
SQL> show pdbs
    CON_ID CON_NAME           OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
     2 PDB$SEED           READ ONLY  NO
     3 VCDO_GOG           READ WRITE NO
SQL> select con_id, key_id from v$encryption_keys;
    CON_ID KEY_ID
---------- ----------------------------------------------------------------
     1 06CB1F66B0B76E4FFFBFA67F9B47C5DC0
     3 069A375EE65D7D4FD9BF4604513679C807
```

The `key_id` for the target container starts with **06CB1F66B0** and the vPDB key starts with **069A375EE6**. There is also an artifact directory for this vPDB found in `<toolkit_directory>/oracle_tde_keystores`:

```
[oracle@ip~]$ ls /work/oracle_tde_keystores/
VCDO_GOG_Untitled_3f9c41d4-c123-42b4-ba36-56ef03ade746
```

Migrate this vPDB to the container database `CDOMLOSR50BD` located on the host `tde-okv-source19`. Currently, that container has the following configuration:

```
SQL> show pdbs
    CON_ID CON_NAME           OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
     2 PDB$SEED           READ ONLY  NO
     3 CDOMLOSR50BDPDB1       READ WRITE NO
     4 CDOMLOSR50BDPDB2       READ WRITE NO
     5 CDOMLOSR50BDPDB3       READ WRITE NO
SQL> select con_id, key_id from v$encryption_keys where con_id=1;
    CON_ID KEY_ID
---------- ----------------------------------
     1 06EC44AF4DD6FA4F49BF1D5C3267D3AEF4
```

The new target container has a CDB key starting with **06EC44AF4**. The migration steps are as follows:

1. Disable the vPDB in the original container on the host `tde-okv-target19`.
2. Migrate the vPDB to point to the new container on the new target host `tde-okv-source19`. The UI will report the following warnings during the migration operation, which indicate the manual steps that need to be done:

FAULTS

⌄	⚠ WARNING	Migrating a vPDB encrypted with OKV or HSM from " " to " "	Jan 12, 2024 1:07 PM

Enabling a vPDB encrypted with OKV or HSM that has been migrated to a different target environment than the source environment requires manually moving the source TDE artifact directory to the target. It also requires making the source CDB$ROOT and vPDB master encryption keys available to the target.

User Action

Move the TDE artifact directory "/work/oracle_tde_keystores/VCDO_GOG_Untitled_3f9c41d4-c123-42b4-ba36-56ef03ade746" on " " into the keystores directory "/work/oracle_tde_keystores" on " " before enabling your vPDB. Do not remove any existing files on the target. Also, grant the target access to the source CDB$ROOT and vPDB master encryption keys. Refer to the Delphix documentation on migrating TDE-configured vPDBs for additional details.

Resolving and Ignoring Resolve Ignore

⚠ WARNING

Date
Jan 12, 2024 1:07 PM

Title
Migrating a vPDB encrypted with OKV or HSM from "
tgt" to "

Target
VCDO_GOG

Details
Enabling a vPDB encrypted with OKV or HSM that has been migrated to a different target environment than the source environment requires manually moving the source TDE artifact directory to the target. It also requires making the source CDB$ROOT and vPDB master encryption keys available to the target.

User Action
Move the TDE artifact directory "/work/oracle_tde_keystores/VCDO_GOG_Untitled_3f9c41d4-c123-42b4-ba36-56ef03ade746" on "
" into the keystores directory "/work/oracle_tde_keystores" on "
" before enabling your vPDB. Do not remove any existing files on the target. Also, grant the target access to the source CDB$ROOT and vPDB master encryption keys. Refer to the Delphix documentation on migrating TDE-configured vPDBs for additional details.

3. Set the **TDE External Key Manager Credential** for the new target container **CDOMLOSR50BD** on the new target host `tde-okv-source19` if it is not already set. Please refer to Adding or Editing the TDE External Key Manager Credential (see page 1596).
4. Grant the target access to the `CDB$ROOT` and vPDB master encryption keys (refer to Oracle Key Vault documentation[467]) to add the members into the target virtual wallet.
5. Copy the TDE artifact directory `/<toolkit_directory>/oracle_tde_keystores/VCDO_GOG_Untitled_3f9c41d4-c123-42b4-ba36-56ef03ade746` to the new target host `tde-okv-source19`. If there are no TDE OKV-enabled vPDBs on the new target host, the `oracle_tde_keystores` directory may need to be created first.
   a. On the new target host, check if `<toolkit_directory>/oracle_tde_keystores` exists. Create it, if not. While creating the directory, you should make sure the group

467 https://docs.oracle.com/en/database/oracle/key-vault/21.7/okvag/okv_wallets_and_keys.html#GUID-2B541778-EF4E-4D3A-B251-377FBF4490C7

ownership is set to `oinstall` and the directory's permissions are set to `0770`. For example:

```
mkdir -p /work/oracle_tde_keystores
chgrp oinstall /work/oracle_tde_keystores
chmod 0770 /work/oracle_tde_keystores
```

b. Copy the TDE artifact directory from the old target host to the new target host, preserving file permissions. For example, if using `scp` on the new target host:

```
scp -rp <old_target_hostname>:/work/oracle_tde_keystores/
VCDO_GOG_Untitled_3f9c41d4-c123-42b4-ba36-56ef03ade746 /work/
oracle_tde_keystores/
```

c. If the copied files are now owned by a user that is not the Oracle home owner, add group-write permissions to key export files to prevent key imports from failing with `ORA-46646: file from which keys are to be imported is invalid`. For example:

```
find /work/oracle_tde_keystores/VCDO_GOG_Untitled_3f9c41d4-c123-42b4-
ba36-56ef03ade746 -name "*.keys" -exec chmod g+w {} \;
```

6. Enable the vPDB to complete the migration operation. The vPDB is successfully started in the new container, with its key starting with **069A375EE6**:

```
SQL> show pdbs
    CON_ID CON_NAME           OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
     2 PDB$SEED           READ ONLY  NO
     3 CDOMLOSR50BDPDB1       READ WRITE NO
     4 CDOMLOSR50BDPDB2       READ WRITE NO
     5 CDOMLOSR50BDPDB3       READ WRITE NO
     6 VCDO_GOG           READ WRITE NO
SQL> select con_id, key_id from v$encryption_keys;
    CON_ID '06'||MKID
---------- ----------------------------------
     1 06EC44AF4DD6FA4F49BF1D5C3267D3AEF4
     6 069A375EE65D7D4FD9BF4604513679C807
```

9.7.10.24 Migrating a vPDB (from a linked CDB) to a higher Oracle version (linked CDB)

Given a vPDB in a linked CDB (CDBold for hereon) the following steps can be used to migrate it to a higher Oracle version linked CDB (CDBnew from hereon):

- It might be necessary to run pre-upgrade Oracle scripts before doing the upgrade, consult Oracle documentation.
- Disable the vPDB, using the GUI or the CLI.
- Using the CLI perform manual migration. If the CDBnew is in the same host as the CDBold, then only cdbConfig needs to be specified, otherwise the repository and the environmentUser also need to be specified:

```
> cd /sourceconfig
sourceconfig> select vpdb
sourceconfig 'vpdb'> update
sourceconfig 'vpdb' update *> set cdbConfig=CDBnew
sourceconfig 'vpdb' update *> set environmentUser=<new oracle user>'
sourceconfig 'vpdb' update *> set repository=''
sourceconfig 'vpdb' update *> commit
```

After enabling the vPDB without starting, the environment of the linked CDB should not be refreshed until the vPDB has been created and upgraded.

- Enable the vPDB without starting it:

```
> cd /source
source> select vpdb
source 'vpdb'> enable
source 'vpdb' enable *> set attemptStart=false
source 'vpdb' enable *> commit
```

- Add the vPDB to CDBnew using sqlplus, which has two options:
  - The vPDB can be plugged as a clone (to get a new DBID/GUID):

```
SQL> create pluggable database vpdb as clone using'/mnt/provision/vpdb-
CDOMLOSRA92E/datafile/delphix_group_writable/VPDB.xml' nocopy tempfile
reuse;
```

  - The vPDB can be plugged without the clone option (to retain the same DBID/GUID):

```
SQL> create pluggable database vpdb using'/mnt/provision/vpdb-
CDOMLOSRA92E/datafile/delphix_group_writable/VPDB.xml' nocopy tempfile
reuse;
```

- Open the vPDB in upgrade mode:

```
SQL> alter pluggable database vpdb open upgrade;
```

- Upgrade the vPDB following the procedures documented by Oracle.
- Take a new snapshot with the new version upgraded and delete the previous snapshot containing the previous version metadata.

All Delphix operations on the upgraded vPDB are possible, which include:

- Start
- Stop
- Enable
- Disable
- Taking a snapshot
- Rewind (to an upgraded snapshot)
- Refresh (requires additional steps)
    - Disable the vPDB
    - Using the CLI to migrate the vPDB back to CDBold:

      ```
      > cd /sourceconfig
      sourceconfig> select vpdb
      sourceconfig 'vpdb'> update
      sourceconfig 'vpdb' update *> set cdbConfig=CDBold
      sourceconfig 'vpdb' update *> set environmentUser=<old oracleuser>'
      sourceconfig 'vpdb' update *> set repository=''
      sourceconfig 'vpdb' update *> commit
      ```

    - Refresh the vPDB
    - Repeat the procedure for migration as described before

9.7.10.25 V2P with an Oracle VDB

This topic describes the procedure for exporting a virtual database (VDB) to a physical one, also known as V2P.

9.7.10.25.1 Procedure

1. Login to the **Delphix Management** application.
2. Click **Manage** and select **Datasets**.
3. Select the dSource or VDB you want to export.
4. Select the snapshot of the dSource or VDB state you want to export.
5. If you want to export the state of the database from a specific point in time, select the LogSync icon and then select the point in time from which you want to create the export.

6. From the actions menu (...) select **Virtual to Physical**.
7. Select the **target environment**. The target environment should have been added to Delphix previously and should meet all target host requirements, see Requirements for Oracle hosts and databases (see page 1392)
8. Enter the Target Directory for the export.
   The target directory you enter must exist in the target environment, and the Delphix operating system user listed under the environment must have permission to write to it. The target directory should be empty.
9. Select whether or not to **Open database after recovery**.
   If you do not select this option, the Oracle database will not undergo open resetlogs, and the database will not be available for read/write access. This can be useful if the files are to be used to restore an existing data file for recovery purposes. You can use the scripts that are created in the target environment to complete the database open process at a later time. For more information, see Manually recovering a database after V2P (see page 1039).
10. Click **Advanced** to customize **data transfer settings**, customize the **target directory layout**, enter any **database configuration** parameters or enter **file mappings** from the source environment to the target. For more information, see Customizing target directory structure for database export (see page 1047), Configuration settings for Oracle virtual databases (see page 1517) and Customizing VDB file mappings (see page 1533). The data transfer settings are described below:
    a. **Compression** – Enable compression of data sent over the network. Default is **Off**.
    b. **Encryption** – Enable encryption of data sent over the network. Default is **Off**.
    c. **Bandwidth Limit** – Select the network bandwidth limit in units of megabytes per second (MB/s) between the Delphix Engine and the target environment. The default is **0**, which means no bandwidth limit is enforced.
    d. **Number of Connections** – Select the number of transmission control protocol (TCP) connections to use between the Delphix Engine and the target environment. Multiple connections may improve network throughput, especially over long-latency and highly-congested networks. The default is **1**.
    e. **Number of Files to Stream Concurrently** – Select the number of files that V2P should stream concurrently from the Delphix Engine to the target environment. The default is **3**.
11. Click **Next**.
12. Select whether you want to have an email sent to you when the export process completes.
13. Click **Submit**.

9.7.10.25.2 Post-requisites

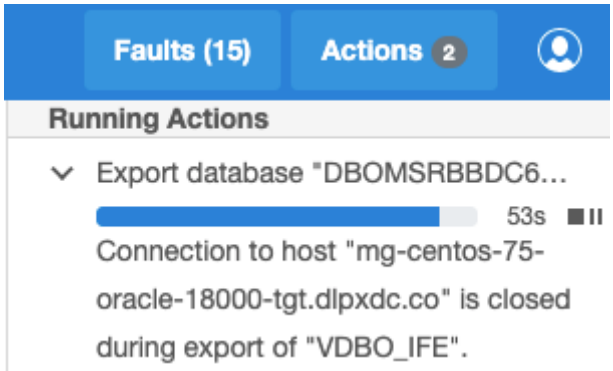
If you did not select **Open Database After Recovery**, follow the instructions in Manually Recovering a Database after V2P (see page 1039) to complete the database open process.

9.7.10.25.3 Resumable V2P

Resumable V2P is a capability that allows you to suspend a V2P operation and then resume it at a later time, without redoing any of the work already completed. For example, any portion of a file that has already been transferred to the target environment is not re-sent. For an entire file that has already been transferred, no part is re-sent.

The image below presents a progress bar, a stop button, and a pause button while a V2P is running. To manually suspend a V2P operation:

1. Click the pause button.

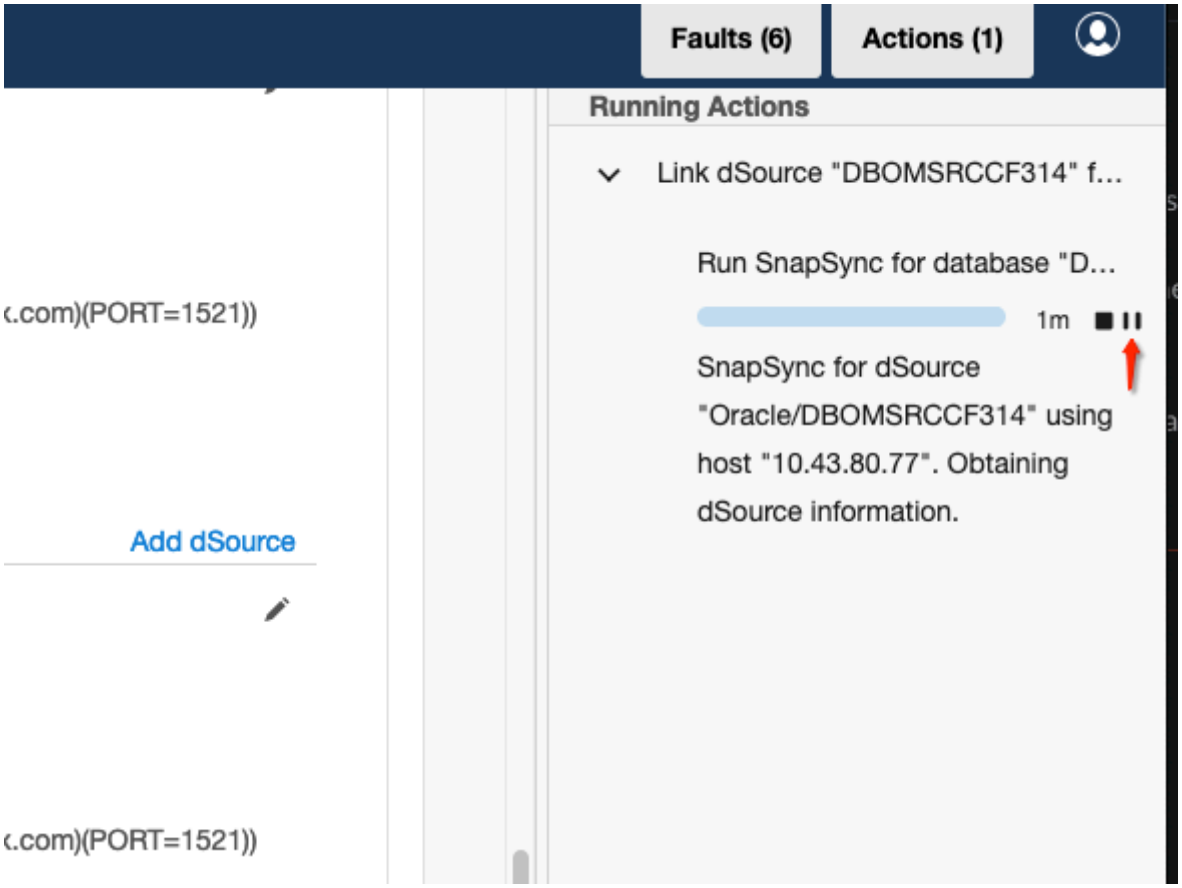


The next image represents a message alert generated after a V2P job has been suspended. To manually resume the job:

1. Click the play button.

The below image indicates that you can also pause and resume the initial snapsync from the **Actions** panel.

1. Click the resume button



9.7.10.25.4 Recoverable errors

Broadly speaking, a "recoverable error" is an error condition caused by a disruption in the environment or on the target host, not errors in the actual V2P operation. Examples of recoverable errors include:

- A timeout due to a network outage
- Running out of disk space on the target environment
- An inability to create directories or files on the target environment

You can often address recoverable errors by taking some action to fix the problem, such as freeing up space on the target environment.

9.7.10.25.5 Auto-suspend

A V2P operation that encounters a recoverable error is auto-suspended: it appears as a suspended job in the user interface (UI), with a message detailing the error condition. Once you have fixed the error, you can simply resume the job. Alternatively, you can cancel the job. Just as when you manually suspend and resume a job, any portion of a file that has been transferred, including possibly the entire file itself, is not re-sent when the job resumes.

9.7.10.25.6 Export an Oracle VDB or a vPDB to a physical ASM or exadata database

9.7.10.25.6.1 Overview

This article describes how to perform an export or an in-place conversion of a virtual database (VDB) or a virtual pluggable database (vPDB) into a physical database stored on Oracle Automatic Storage Management (ASM) disk groups. No intermediate storage is needed; the database files are moved directly from Delphix into the ASM diskgroup(s).

This procedure can be used to export an Oracle VDB, or a vPDB in a Linked CDB, to ASM disk group(s), including disk group(s) residing in an Oracle Exadata machine. The procedure follows Oracle's recommended best practice of using a single disk group for data files. A separate disk group can be specified for redo log files.

> This procedure can be performed using the CLI only and applies to all Oracle RDBMS Versions s upported by Delphix. For CLI commands, refer to the CLI cookbook: export a non-multitenant virtual Oracle database to ASM (see page 2666) or CLI cookbook: export a multitenant virtual pluggable Oracle database to ASM (see page 2668) articles.
>
> Furthermore, it is also fully supported with TDE-enabled virtual databases provisioned through Delphix.

Prerequisites

1. Oracle Managed Files (OMF) must be enabled on the VDB or vPDB before export can be performed. OMF eliminates the need for the DBA to directly manage the operating system files that comprise an Oracle Database. As a result of this OMF requirement, it is expected that all database file names of the exported physical database would change.
2. The following conditions must be met prior to the export operation. Export will fail if any of these conditions are not met:
    a. Sufficient storage space must be available in the target ASM diskgroup for datafiles and the target ASM diskgroup for online logs if specified. The validation of the target ASM diskgroup for online logs is only applicable in the case of VDBs and not vPDBs.
    b. While exporting a vPDB, if a new PDB name is specified:
        i. it must meet all the naming constraints as defined in the Oracle documentation.
        ii. it must be different from the existing PDBs in the target CDB.
    c. While exporting a VDB, if a new database unique name is specified:
        i. it must meet all the naming constraints as defined in the Oracle documentation.
        ii. it must be different from the database unique name of any other database on the same target host.
        iii. it must not be a RAC database.
    d. No offline datafiles or tablespaces must exist in the VDB or vPDB.

   e. The VDB or vPDB on which the export is initiated must be Open.
   f. Export of a VDB in a RAC environment must be performed as the Oracle user, otherwise the export will fail. This is required because Delphix issues srvctl commands to configure the resulting physical database in RAC and these commands can only be run with Oracle user privileges.
   g. No other job must be running on the virtual database or its associated environment.
   h. When running export of a VDB or vPDB in a RAC environment, ensure that the states of all the cluster nodes are displayed as 'Enabled' in the Delphix management GUI. If one of the cluster nodes is disabled, the export operation will fail, although the physical copy has been completed and it is usable, however the VDB or vPDB will need to be force disabled manually.
   i. The VDB or vPDB must have a provisionable snapshot, otherwise operation will fail with the following message: `Cannot find a point in the TimeFlow for the semantic location "LATEST_POINT"`.

3. If you want to export a vPDB in a vCDB, the vPDB must be migrated to a Linked CDB and then enabled, or alternatively, provision a child vPDB from this vPDB snapshot to a linked CDB and then perform the export.

Procedure

1. Delphix takes a new snapshot of the virtual database/pluggable database before starting export. Also, Delphix retains the timeflow and all its snapshots after the virtual source is exported. As such, after export, the virtual source can be easily migrated and rewinded to the previously created snapshots.
2. It is recommended that the export be performed on a newly provisioned VDB or vPDB, although it can still be performed on your existing VDB or vPDB. The reason is it simplifies the post export process to re-enable the existing VDB or vPDB.
3. Please refer to the appropriate CLI cookbook sections for the procedure to export a VDB (see page 2666) or export a vPDB (see page 2668) to a physical ASM or Exadata database.

Performance considerations before running the export

1. When deciding the number of RMAN channels to use, there are tradeoffs between speed and resource consumption on the host.
2. The number of RMAN channels should not be more than the number of datafiles.
3. Similar to selecting the number of RMAN channels to perform backup, if impact to other databases is not a concern, then setting the number of channels should be increased to the point of diminished returns. Otherwise it is a compromise between what the system can handle and how fast we want the export to finish.
4. By default it is set to 8, but this value might be too large for some environments and should be adjusted down appropriately.

Performance tuning considerations for Oracle databases with bigfile tablespaces

As part of the export workflow, Delphix performs the data transfer using RMAN. By default, 8 RMAN channels are created to copy files from Delphix to ASM. The number of RMAN channels can be specified in the `export` CLI. Each RMAN channel takes up one datafile at a time and picks up another as soon as it is done copying one. For databases that contain hundreds or thousands of small data files, this is usually sufficient.

Consider a database which contains 100 data files of sizes 20-40GB each, and 1 large datafile of size 1TB or larger. In such a case, one of the RMAN channels will be doing all the work of transferring the 1TB datafile and the rest of the RMAN channels would be idling until the 1TB of data is copied. To speed up the data transfer for such databases, Delphix now provides a new `rmanFileSectionSizeInGb` API parameter that can be passed to the export CLI. When this parameter is specified, Delphix will pass the `SECTION SIZE` parameter to the RMAN `BACKUP AS COPY` command, and RMAN then creates several chunks of backup copies in which each chunk contains the blocks from one file section. This type of copy is called a multisection copy. The purpose of multisection copies is to enable RMAN channels to copy a single large file in parallel. Thus, when the `rmanFileSectionSizeInGb` parameter is specified in the export CLI, RMAN will divide the work among multiple channels, with each channel copying one file section in a file. For example, if `rmanFileSectionSizeInGb` is set to 64, then the large 1TB datafile transfer will be broken into 16 sections of 64GB each, and all 8 channels will then be utilized for copying the large datafile once the copy of the smaller files has completed. Copying a file in separate sections can thus improve the performance of copies of large data files which in turn could speed up the time taken by the V2ASM export job.

Considerations after successful export of a VDB or vPDB to ASM

After the export is successful, no Delphix operations are allowed on the VDB/vPDB except migrate or a fresh provision of a new child VDB or vPDB. However, there may be situations where you may need to re-enable the VDB or vPDB, please perform a migrate of the VDB/vPDB to a different host and/or CDB and then rewind the VDB/vPDB to it's latest snapshot

1. After successfully performing export of a VDB to replace a linked physical database that is damaged and is unusable with new physical database having the same name, same unique name and same SID as the original damaged database, the new physical database can be linked back to Delphix engine using the following steps:
    a. Delete or Migrate the VDB that was used to create a new physical database to a different environment.
    b. Refresh the environment where the new physical database is created.
    c. Detach the dSource from the original source database.
    d. Force attach the dSource to the newly created physical database.
2. After successfully performing export of a vPDB with the new physical PDB name that is same as the vPDB name, the new physical PDB can be linked back to Delphix engine using the following steps:
    a. Delete or migrate the vPDB to a different CDB.
    b. Refresh the environment where the new physical PDB is created.
    c. Detach the dSource PDB from the original source PDB.
    d. Force attach the dSource to the newly created physical PDB.

9.7.10.25.7 Move an Oracle VDB to a physical ASM or exadata database

Deprecation Notice
Warning: This procedure is deprecated as of 9.0.0.0. Delphix recommends using the new export feature in 9.0.0.0 that is documented in the Export an Oracle VDB or vPDB to a Physical ASM or Exadata Database (see page 1628) article. This procedure will be eventually removed in future versions of the Delphix Continuous Data Engine.

9.7.10.25.7.1 Overview

This article describes how to move a virtual database (VDB) into a physical database stored on Oracle Automatic Storage Management (ASM) disk groups. This is a scripted procedure that is assisted (but not fully automated) by Delphix. No intermediate storage is needed; the database files are moved directly from Delphix into the ASM diskgroup(s).

This procedure can be used to export a VDB to ASM disk group(s), including disk group(s) residing in an Oracle Exadata machine. The `move-to-asm.sh` script follows Oracle's recommended best practice of using a single disk group for data files. A separate disk group can be specified for redo log files. Professional Services can be engaged if customization is needed.

This procedure applies to **all** Oracle RDBMS Versions supported by Delphix; in addition, it is not supported by PDB/Multi-tenant type databases.

9.7.10.25.7.2 Procedure

1. Create an ASM disk group that will contain all the database files. Optionally, create a separate disk group for redo log files.
2. Provision a VDB on the target machine that is running Oracle ASM or Exadata in order to do the export; this VDB will be deleted after the export is complete. This temporary VDB can be provisioned from an existing VDB. You can also provision it from a dSource, which allows you to achieve a full restore of the original source ASM database.
3. Login to the target machine (or a node of the target RAC cluster) where the VDB instance and ASM instance are running. The directory will be in the form:

```
<toolkit_path>/Delphix_<engine_id><user><host|cluster>/databases/oracle/
<db_unique_name>/<database_sid>/
```

4. For a RAC database, be sure to use the directory path containing the SID of the instance running on the node where you have logged in.
5. Ensure that Oracle environment variables ORACLE_HOME,ORACLE_SID, and CRS_HOME (RAC only) are correctly set for the VDB that needs to be moved.
6. Execute the script `move-to-asm.sh` as the **Environment User** who provisioned the single instance VDB. For a RAC VDB, the **Environment User** selected to execute the `move-to-asm.sh` *must* be the Oracle installation owner. This is due to an Oracle restriction that only the installation owner can invoke `srvctl` to add or remove database configurations. Alternatively, enter `move-to-asm.sh` as a **Post-Provision Hook Script** when provisioning the VDB. This will provision and move the VDB into ASM disk groups in a single flow. You must specify the `-noask` option to execute in non-interactive mode. For more information about hook scripts, visit the Hook Scripts for Automation and Customization (see page 1056) article.

Command syntax: `move-to-asm.sh [-noask] [-parallel #] [-dbunique db_unique_name] <data_diskgroup> [<redo_diskgroup>]`

Arguments	Description
`-noask` [optional]	Do not prompt for confirmation before moving the VDB. The default is to prompt.
`-parallel #` [optional]	The number of RMAN channels used to move the VDB to ASM. The default is 8.
`-dbunique <>` [optional]	Unique database name for the resulting physical database. The default is the unique name of the VDB. The `-dbunique` argument only works if a database with that name that you select already exists.
`<>` [required]	Target ASM disk group for data, server parameter and control files.
`<>` [optional]	Target ASM disk group for redo log files. Default is `<>`.

move-to-asm script

The *move-to-asm.sh* script generates several output files. These files are all written to the working directory of the script:

```
init<oracle_sid>_run<process-id>_moveasm.ora
```

The log file for the operation:

```
move-to-asm.sh_<oracle_sid>_run<process-id>.log
```

The init.ora parameter file created for the ASM DB instance:

```
source_init<oracle_sid>.ora
```

The init.ora for the source database (from which the VDB was created).

9.7.10.25.7.3 *Finishing steps*

The final steps to manually execute are shown when the script completes and included in the script output.

1. For Single Instance only: copy generated init<$ORACLESID>.ora parameter file to the default $ORACLE_HOME/dbs/init<$ORACLE_SID>.ora
2. Modify initialization parameters to match the original source database parameters, if necessary.

As a convenience to assist with this step, the source database parameters are restored as source_init<$ORACLE_SID>.ora in the execution directory. Be sure to copy this file to a save place *before* deleting the temporary VDB.

The *move-to-asm.sh* script generates several output files. These files are all written to the working directory of the script and should also be copied to a safe place if needed:

```
move-to-asm.sh_<oracle_sid>_run<process-id>.log
```

The log file for the operation:

```
init<oracle_sid>_run<process-id>_moveasm.ora
```

The init.ora parameter file created for the ASM DB instance:

```
source_init<oracle_sid>.ora
```

The init.ora for the source database (from which the VDB was created).

3. Delete the Delphix VDB that was moved.

4. Start up the physical database that will now run on ASM.
   (A RAC database is automatically started up by the `move-to-asm.sh` script using `srvctl` but may have been stopped when the VDB was deleted, if it has the same name.)

9.7.10.25.7.4 Sample execution

```
$ cd toolkit/Delphix_564dc816_d457_6224_1327_4d43fd1b3d97_oracle_cluster/databases/
oracle/RAC2ASM/RAC2ASM1
$ export ORACLE_HOME=/u01/app/oracle/product/12.2.0.1/dbhome_1
$ export ORACLE_SID=RAC2ASM1
$ sh move-to-asm.sh -noask +DATA +REDO
END_OF_SETUP
============================================================
Virtual-to-ASM script (move-to-asm.sh v5.2) for Delphix
Copyright (c) 2013, 2018 by Delphix. All rights reserved.
============================================================

Moving database RAC2ASM to ASM: started at Wed Feb 19 16:08:43 EST 2020
db_unique_name => RAC2ASM
ORACLE_SID => RAC2ASM1
ORACLE_HOME => /u01/app/oracle/product/12.2.0.1/dbhome_1
Datafile diskgroup => +DATA
Online redo diskgroup => +REDO
RMAN Channels => 8

Create cluster database RAC2ASM
Generate script to move tempfiles to ASM
Generate script to drop old tempfiles
Generate script to drop offline tablespaces
Generate script to make read-only tablespaces read-write
Generate script to make tablespaces read-only again
Make read-only tablespaces read-write
Remove offline tablespaces
Updating server parameter file with ASM locations
Creating temporary initialization parameter file
Move spfile to ASM
Move datafiles to ASM: started at Wed Feb 19 16:09:41 EST 2020
Move datafiles to ASM: completed at Wed Feb 19 16:10:31 EST 2020
Stop cluster database RAC2ASM
Restart cluster database RAC2ASM
Remove old tempfiles
Move tempfiles into ASM
Move Online logs
Restore any read-only tablespaces
Shutdown database
Starting up RAC database RAC2ASM...
Database RAC2ASM moved to ASM: completed at Wed Feb 19 16:12:45 EST 2020

Final Steps to complete the move to ASM:
```

```
Source parameters are restored at /home/oracle/toolkit/
Delphix_564dc816_d457_6224_1327_4d43fd1b3d97_oracle_cluster/databases/oracle/RAC2ASM/
RAC2ASM1/source_initRAC2ASM1.ora
*.audit_file_dest='/u01/app/oracle/admin/DBOMSRD914EA/adump'
*.audit_trail='DB'
*.cluster_database=TRUE
*.compatible='12.2.0'
*.control_files='+DATA/DBOMSRD914EA/CONTROLFILE/current.297.1027776391','+REDO/
DBOMSRD914EA/CONTROLFILE/current.264.1027776391'
*.db_block_size=8192
*.db_create_file_dest='+DATA'
*.db_create_online_log_dest_1='+DATA'
*.db_create_online_log_dest_2='+REDO'
*.db_name='DBOMSRD9'
*.db_recovery_file_dest='+REDO'
*.db_recovery_file_dest_size=6989807616
*.db_unique_name='DBOMSRD914EA'
*.diagnostic_dest='/u01/app/oracle'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=DBOMSRD914EAXDB)'
*.dlpx_recovery_log_archive_format='arch_%t_%s_%r.log'
*.instance_number=1
*.listener_networks=''
*.local_listener='(ADDRESS=(PROTOCOL=TCP)(HOST=10.43.5.243)(PORT=1521))'
*.log_archive_format='%t_%s_%r.dbf'
*.memory_max_target=1073741824
*.memory_target=1073741824
*.nls_language='AMERICAN'
*.nls_territory='AMERICA'
*.open_cursors=300
*.processes=300
*.remote_listener='node1.sample.delphix.com:1521'
*.remote_login_passwordfile='EXCLUSIVE'
*.thread=1
*.undo_tablespace='UNDOTBS1'

Copy this file to a safe location before removing the VDB

1) Delete VDB on Delphix.
2) Modify initialization parameters to match source and restart.
```

9.7.10.25.8 Refresh/Rewind an Oracle virtual database exported to physical ASM database

The "Export an Oracle VDB or a vPDB to a Physical ASM or Exadata Database (see page 1628)" feature introduced in Delphix 9.0.0.0, allowed the export or in-place conversion of a virtual database to ASM, but the refresh or rewind operation was not allowed after the virtual database was exported to ASM.

Starting with the 12.0.0.0 release, Delphix allows a refresh or rewind operation to be performed on a virtual database (VDB or vPDB) after the V2ASM export job completes.

9.7.10.25.8.1 How to allow refresh/rewind on a virtual database after it has been exported to ASM?

A new parameter `allowRefreshRewindPostV2P` under `operationsPostV2P` has been introduced to allow refresh/rewind on the virtual database after it has been exported to ASM.

```
ip-10-110-247-210 database export *> ls
Properties
    type: OracleDBExportParameters (*)
    storageStrategy:
        type: OracleExportASMStorageStrategy (*)
        asmLayout:
            type: OracleASMLayout (*)
            defaultDataDiskgroup: +DATA (*)
            redoDiskgroup: (unset)
    transferStrategy:
        type: OracleExportDBInPlaceTransferStrategy (*)
        dbUniqueName: (unset)
        operationsPostV2P: (unset)
        rmanChannels: 8 (*)
        rmanFileSectionSizeInGb: 0 (*)
        virtualSource: VDBOMSRASF1B4_25I (*)
```

To allow the refresh or rewind operation on the virtual database after the export operation, set the parameter `allowRefreshRewindPostV2P` under `operationsPostV2P` as true.

On CLI, run the following command to set the same.

```
set transferStrategy.operationsPostV2P.allowRefreshRewindPostV2P=true
```

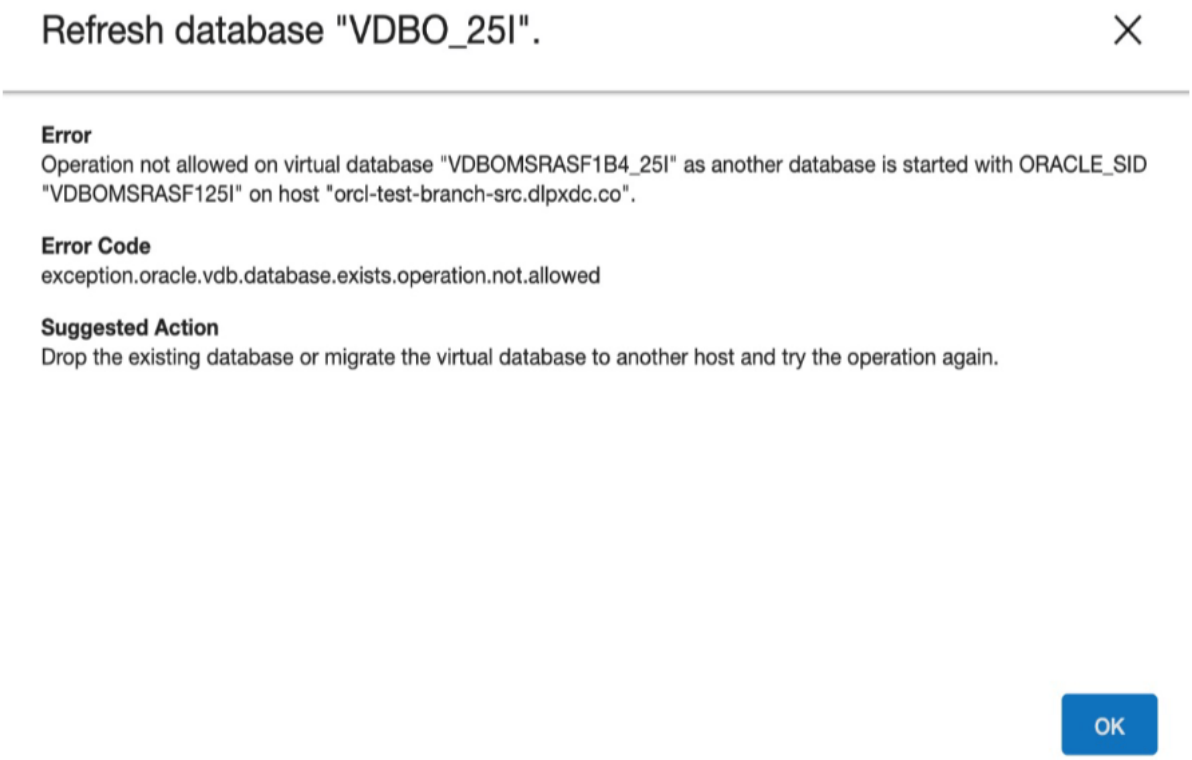
```
ip-10-110-247-210 database export *> ls
Properties
    type: OracleDBExportParameters (*)
    storageStrategy:
        type: OracleExportASMStorageStrategy (*)
        asmLayout:
            type: OracleASMLayout (*)
            defaultDataDiskgroup: +DATA (*)
            redoDiskgroup: (unset)
    transferStrategy:
        type: OracleExportDBInPlaceTransferStrategy (*)
        dbUniqueName: (unset)
        operationsPostV2P:
            type: OracleExportOperationsPostV2P (*)
            allowRefreshRewindPostV2P: true (*)
        rmanChannels: 8 (*)
        rmanFileSectionSizeInGb: 0 (*)
```

```
        virtualSource: VDBOMSRASF1B4_25I (*)
```

After the export job, a physical database will be running on Oracle Host.

9.7.10.25.8.2 What happens if a refresh or rewind operation is attempted on a virtual database after export?

On refresh or rewind operation, the user will get an exception.



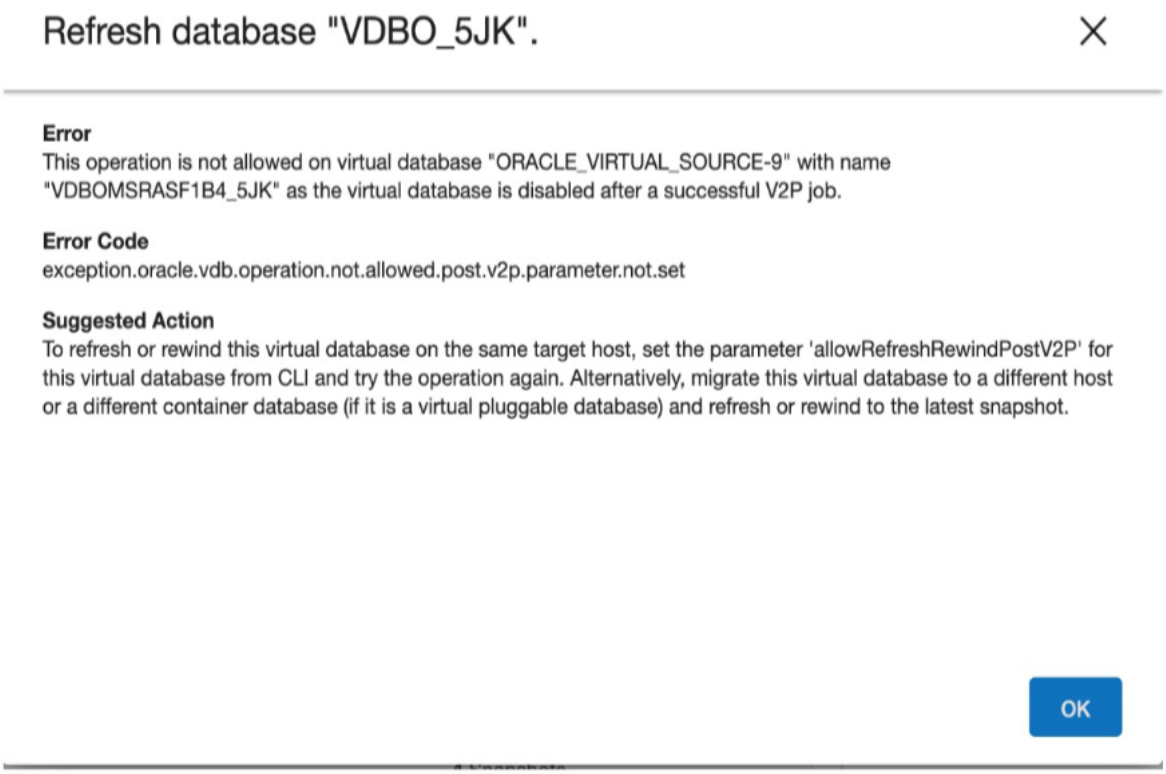
The refresh or rewind operation is not allowed because there is another database running on the Oracle host with the same SID. To go ahead with the refresh or rewind operation, shut down the physical database and then try doing the refresh operation again. The refresh or rewind operation will be successful now.

9.7.10.25.8.3 What if a user forgets to set the operationsPostV2P during export?

```
ip-10-110-247-210 database export *> ls
Properties
    type: OracleDBExportParameters (*)
    storageStrategy:
        type: OracleExportASMStorageStrategy (*)
        asmLayout:
            type: OracleASMLayout (*)
            defaultDataDiskgroup: +DATA (*)
            redoDiskgroup: (unset)
```

```
    transferStrategy:
        type: OracleExportDBInPlaceTransferStrategy (*)
        dbUniqueName: (unset)
        operationsPostV2P: (unset)
        rmanChannels: 8 (*)
        rmanFileSectionSizeInGb: 0 (*)
        virtualSource: VDBOMSRASF1B4_5JK (*)
```

After the export job, if the user tries refresh or rewind operation on a virtual database, an exception will be thrown because the parameter `allowRefreshRewindPostV2P` is not set.



9.7.10.25.8.4 How can this parameter be set from the Delphix Engine CLI using source update API?

After the export operation, the Delphix source update API does not allow updating the parameters for a virtual database that has been disabled after an export operation. The only exception is the `allowRefreshRewindPostV2P` parameter, which can be set as shown below:

```
ip-10-110-247-210> source
ip-10-110-247-210 source> select VDBOMSRASF1B4_5JK
ip-10-110-247-210 source 'VDBOMSRASF1B4_5JK'> update
ip-10-110-247-210 source 'VDBOMSRASF1B4_5JK' update *> set allowRefreshRewindPostV2P=t
rue
ip-10-110-247-210 source 'VDBOMSRASF1B4_5JK' update *> commit;
    Dispatched job JOB-116
```

```
    SOURCE_UPDATE job started for "VDBOMSRASF1B4_5JK".
    SOURCE_UPDATE job for "VDBOMSRASF1B4_5JK" completed successfully.
```

An attempt to update any other virtual database parameter will fail with the following error:

```
ip-10-110-247-210 source 'VDBOMSRASF1B4_5JK' update *> set
allowAutoVDBRestartOnHostReboot=true;
ip-10-110-247-210 source 'VDBOMSRASF1B4_5JK' update *> commit;
    Dispatched job JOB-117
    SOURCE_UPDATE job started for "VDBOMSRASF1B4_5JK".
    This operation is not allowed on source 'ORACLE_VIRTUAL_SOURCE-9' with name
'VDBOMSRASF1B4_5JK' as the source is disabled after a successful V2P job.
  Action: To re-enable, migrate this virtual source to a different host or a
different CDB (if it is a virtual pluggable database) first and then rewind to the
latest snapshot. Delete the virtual source if it is no longer needed. This does not
impact the newly created physical database.
ip-10-110-247-210 source 'VDBOMSRASF1B4_5JK'>
```

After setting the parameter `allowRefreshRewindPostV2P` to true, the refresh or rewind operation will be successful from GUI.

9.7.10.26 V2P with a snapshot or timeflow-point

V2P with a snapshot refers to creating a physical database on the target with the data files from a particular snapshot(or timeflow-point).

This section covers the following topic:

- Export an Oracle non-multitenant or a PDB snapshot to a physical ASM or Exadata database (see page 1639)

9.7.10.26.1 Export an Oracle non-multitenant or a PDB snapshot to a physical ASM or Exadata database

9.7.10.26.1.1 Overview

This article describes how to perform an export on a snapshot or a timeflow-point belonging to a non-multitenant database or a pluggable database to create a physical database stored on Oracle Automatic Storage Management (ASM) disk groups. No intermediate storage is needed; a temporary virtual database or virtual pluggable database is provisioned and the database files are moved directly from Delphix into the ASM diskgroup(s).

This procedure can be used to export snapshots of an Oracle non-multitenant dSource or a VDB. It can also be used to export snapshots of a PDB dSource or a vPDB to a pluggable database in a linked CDB. The physical database is created on ASM disk group(s), including disk group(s) residing in an Oracle Exadata machine. The procedure follows Oracle's recommended best practice of using a single disk group for data files. A separate disk group can be specified for redo log files.

> This procedure can be performed using the CLI only and applies to all Oracle RDBMS Versions supported by Delphix. For CLI commands, refer to the CLI cookbook: export a snapshot or a timeflow point of a non-multitenant Oracle database to ASM (see page 2670) and the CLI cookbook: export a snapshot or a timeflow point of a multitenant pluggable Oracle database to ASM (see page 2668) articles.
> Furthermore, it is also fully supported with TDE-enabled databases.

9.7.10.26.1.2 Prerequisites

1. Oracle Managed Files (OMF) must be enabled on the VDB or vPDB.
2. Sufficient storage space must be available in the target ASM diskgroup for datafiles, tempfiles and the target ASM diskgroup for online logs, if specified.
3. While exporting from a PDB snapshot, the new PDB name:
   a. must meet all the naming constraints as defined in the Oracle documentation.
   b. must be different from the existing PDBs in the target CDB.
4. While exporting a non-multitenant snapshot, the new database unique name:
   a. must meet all the naming constraints as defined in the Oracle documentation.
   b. must be different from the database unique name of any other database on the same host.
   c. must be different from the database unique name of any RAC database which has a node on this host even if the node is down.
5. Offline tablespaces can exist; however, offline datafiles must not exist in an online tablespace.
6. Export of a non-multitenant snapshot(or point-in-time) in a RAC environment must be performed as the Oracle software owner user account, otherwise the export will fail. This is required because Delphix issues `srvctl` commands to configure the resulting physical database in RAC and these commands can only be run with Oracle software owner user account privileges.
7. When running export of a non-multitenant or an multitenant snapshot in a RAC environment, ensure that the states of all the cluster nodes are displayed as `Enabled` in the Delphix management GUI.

Procedure

1. Delphix provisions a temporary virtual database/pluggable database, from the provided snapshot or timeflow point, which would be converted in-place to a physical database.
2. Delphix takes a new snapshot of the temporary virtual database/pluggable database before starting export.
3. The virtual database/pluggable database is converted in-place to a physical database on ASM. The temporary virtual source is then deleted.

4. Please refer to the appropriate CLI cookbook sections for the procedure to export a non-multitenant database snapshot (see page 2670) or export a pluggable database snapshot (see page 2675) to a physical ASM or Exadata database.

Performance considerations before running the export

1. When deciding the number of RMAN channels to use, there are tradeoffs between speed and resource consumption on the host.
2. The number of RMAN channels should not be more than the number of datafiles.
3. Similar to selecting the number of RMAN channels to perform backup, if impact to other databases is not a concern, then setting the number of channels should be increased to the point of diminished returns. Otherwise it is a compromise between what the system can handle and how fast we want the export to finish.
4. By default it is set to 8, but this value might be too large for some environments and should be adjusted down appropriately.

9.7.10.27 Provisioning from a replicated Oracle VDB

This topic describes how to provision from a replicated dSource or virtual database (VDB).

The process for provisioning from replicated objects is the same as the typical VDB provisioning process, except that first you need to select the replica namespace containing the replicated object.

9.7.10.27.1 Prerequisites

- You must have a dSource or VDB that has been replicated from one Delphix Engine to another, as described in Replication Overview (see page 2367)
- The Delphix Engine containing the replicated dSource or VDB must have a compatible target environment that it can use to provision a VDB from the replicated dSource or VDB.

9.7.10.27.2 Procedure

1. On the Delphix Engine containing the replicated dSource or VDB, login to the **Delphix Management** application.
2. In the top menu bar, click **Manage**.
3. Select **Datasets**.
4. From the list of replica namespaces, select the **replica namespace** that contains the dSource or VDB from which you want to provision.
5. The provisioning process is now identical to the process for provisioning standard objects.

9.7.10.27.3 Post-requisites

Once the provisioning job has started, the user interface will automatically display the new VDB in the live system.

9.7.10.28 Provisioning a TDE-enabled vPDB to a RAC cluster target

9.7.10.28.1 Overview

Provisioning a Virtual Pluggable Database (vPDB) to a linked RAC container database first involves using the GUI or CLI to specify the vPDB parameters (such as the vPDB name and target container) along with the snapshot to provision from. Once the provision job is started with these parameters, the Delphix Engine does the following:

1. Chooses an available instance from the target cluster and mounts the snapshot files on that instance.
2. Creates and opens (in mount mode) the auxiliary container database on the target instance, using the snapshot files. The auxiliary container database will have both the CDB and PDB data files from the dSource.
3. Completes recovery to bring the auxiliary container database into a consistent state.
4. Finalizes the state of the auxiliary database and unplugs the vPDB datafiles.
5. When provisioning to a vCDB target, converts the auxiliary CDB into the final vCDB.
6. Plugs the vPDB into the target database, and opens it in read-write mode on the same target instance used for recovery.
7. Spawns start jobs to open the vPDB in read-write mode on the remaining target instances in parallel.

If the dSource is TDE-enabled, then Delphix will need to perform additional operations to complete the provision of a TDE-enabled vPDB to a TDE-enabled cluster target (indicated in red):

1. Chooses an available instance from the target cluster and mounts the snapshot files on that instance.
2. Creates a keystore with the necessary keys to apply encrypted archived log files on the target instance.
3. Creates and opens (in mount mode) the auxiliary container database on the target instance, using the snapshot files. The auxiliary container database will have both the CDB and PDB data files from the dSource.
4. Completes recovery to bring the auxiliary container database into a consistent state.
5. Rotates the vPDB and auxiliary CDB master encryption keys by generating new keys that are unique to the vPDB/auxiliary CDB and not associated with the source PDB or CDB.
6. Exports only the newly generated keys to an exported keyfile to enable unplug.
7. Finalizes the state of the auxiliary database and unplugs the vPDB datafiles.

8. Imports the vPDB key from the exported keyfile into the target keystore.
9. When provisioning to a vCDB target, converts the auxiliary CDB into the final vCDB and creates the vCDB keystore from the auxCDB keystore.
10. Opens the keystore on each node.
11. Plugs the vPDB into the target database, and opens it in read-write mode on the same target instance used for recovery.
12. Spawns start jobs to open the vPDB in read-write mode on the remaining target instances.

All the same information needed for a single instance TDE-enabled vPDB provision is also required for a cluster TDE-enabled vPDB provision, specifically the target keystore password, parent keystore path and password, and encryption secret. The keystores root path is required for a cluster provision.

9.7.10.28.2 Shared storage for keystores

In a cluster database, the database files are on shared storage, which is accessible from all instances in the cluster. If the database is encrypted, then the keystore file itself is also located on the operating system. Oracle recommends that the keystore also be on shared storage, on a different disk from the database files. If the keystore is not on shared storage, then it must be copied to all instances in the cluster after any changes, such as importing a key or generating a new key. Similarly, Delphix recommends that the parent keystore specified for the provision also be on shared storage. If not, then the same file must be copied to all of the instances before the vPDB is first provisioned, and any updates to the parent keystore must also be copied to all of the instances before any vPDB refresh or rewind.

As the autologin wallet is located in the same location as the password-based keystore, it should also be on shared storage. For this reason, `local_autologin` wallets will not work properly, as they will be accessed from multiple nodes in the cluster.

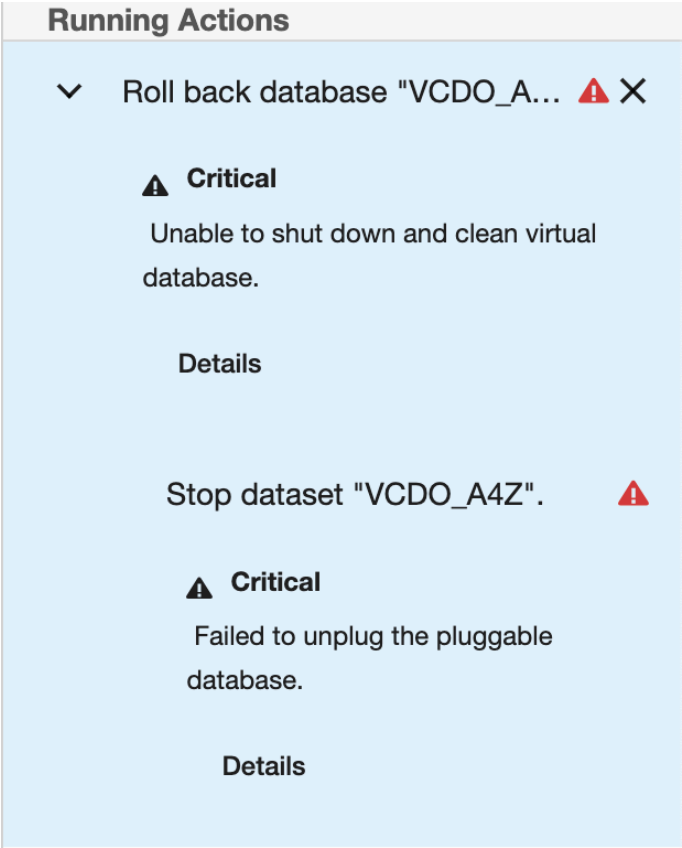
When provisioning to a new vCDB in a cluster target, the path provided for **Target vCDB TDE Keystore location** must be on shared storage and available to all cluster hosts.

9.7.10.28.3 Keystores root path requirements

The artifact directory for a given vPDB is created under the keystores root. As a subsequent operation on the vPDB may choose a different instance than the one used for the initial provision, the artifact directory needs to be accessible from all instances in the cluster. Delphix requires that the keystores root be specified for a TDE-enabled vPDB provision to a cluster target, and furthermore that it be located on shared storage. The engine will validate that this is the case before proceeding with the initial provision.

9.7.10.29 Refreshing or rewinding a broken/unusable virtual PDB

A virtual pluggable database could be broken/unusable if it cannot be operated upon from the Delphix Management such that all Delphix operations involving opening or unplugging of the virtual pluggable database from its container database like Disable, Enable, Refresh and Rewind fail.



A virtual PDB can get into a such state because of (but not limited to):

- Datafile is deleted or gets corrupted
- Datafile is renamed or moved
- Oracle DB instance crashed and archive logs needed to perform recovery are lost/deleted/ unavailable.

9.7.10.29.1 Using force option

A virtual PDB can be brought out of the bad state by Refreshing or Rewinding it with a `force` option. The force option is only available in the CLI. The `force` parameter is part of `OracleRefreshParameters` and `OracleRollbackParameters` for the Refresh and Rewind operations respectively. The default value for the `force` parameter is `false`. For details on how to use the force option, refer to CLI cookbook: Force refresh/rewind a virtual PDB (see page 2699)

9.7.10.29.2 Implications of `force` option

Using the `force` option with Refresh/Rewind has the following implications:

- A forced Refresh/Rewind cannot be subsequently undone using the 'Undo Refresh or Rewind' functionality.A user cannot switch to the timeflow prior to the force Refresh/Rewind using switchTimeflow API
- Timeflow before the force Refresh/Rewind will remain in the broken state, however, any snapshots in that timeflow will continue to remain provisionable.
- In the case of Rewind, Delphix may not be able to identify the 'last good point in timeflow' for the vPDB.

9.7.10.30 Managing cluster instances of Oracle RAC virtual databases

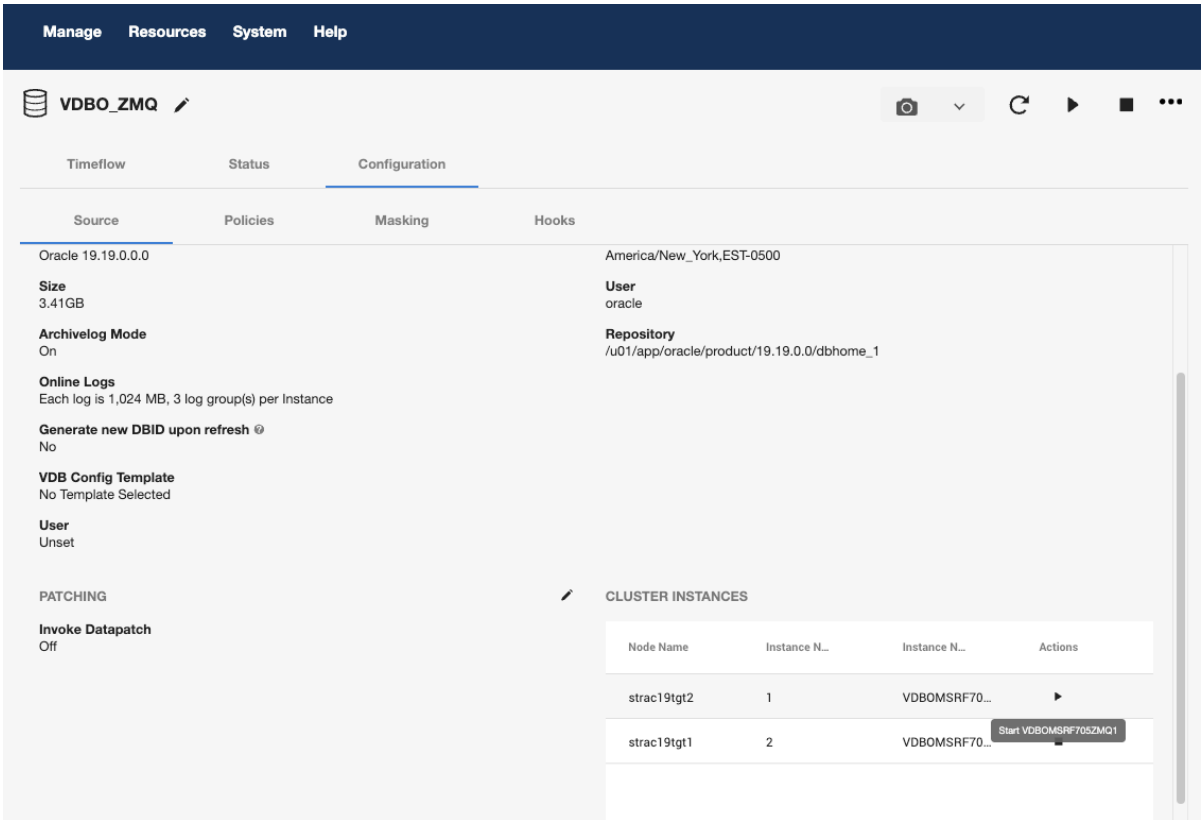
This topic describes how to start and stop cluster instances of the Oracle RAC virtual databases from the **Delphix Management** application.

> The Delphix Management application does not support starting or stopping of a cluster instance of Oracle RAC virtual CDB database. You must use the CLI to start or stop these instances for a virtual CDB (see page 2700).

9.7.10.30.1 Starting a cluster instance

9.7.10.30.1.1 Procedure

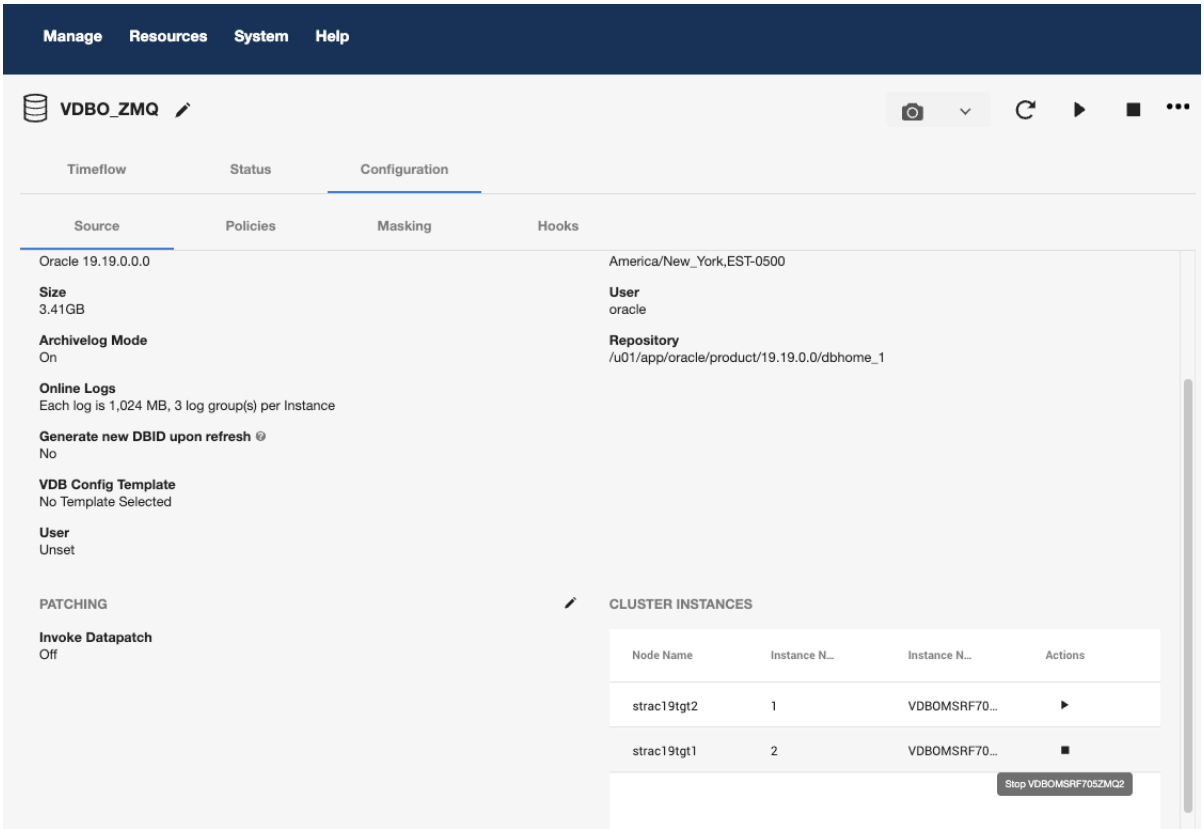
1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Datasets**.
4. Select the virtual database for which you want to start the instance.
5. Click **Configuration** tab for the virtual database.
6. Click **Source** tab of the **Configuration**.
7. In the **CLUSTER INSTANCES** table, for the desired cluster node, click the **Start** icon under the **Actions** column.



9.7.10.30.2 Stopping a cluster instance

9.7.10.30.2.1 Procedure

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Datasets**.
4. Select the virtual database for which you want to start the instance.
5. Click **Configuration** tab for the virtual database.
6. Click **Source** tab of the **Configuration**.
7. In the **CLUSTER INSTANCES** table, for the desired cluster node, click the **Stop** icon under the **Actions** column.



9.7.11 Oracle hook operations

9.7.11.1 Oracle RAC

When linking from, or provisioning to Oracle RAC environments, hook operations will not run once on each node in the cluster. Instead, the Delphix Engine picks a node in the cluster at random and guarantees all operations within any single hook will execute serially on this node.

Note that the Delphix Engine does not guarantee the same node is chosen for the execution of every hook but does guarantee that Pre-/Post- hook pairs (such as Pre-Sync and Post-Sync) will execute on the same node.

9.7.11.2 Shell operations

9.7.11.2.1 RunCommand operation

The RunCommand operation runs a shell command on a Unix environment using whatever binary is available at `/bin/sh`. The environment user runs this shell command from their home directory. The Delphix Engine captures and logs all output from this command. If the script fails, the output is displayed in the Delphix Management application and command-line interface (CLI) to aid in debugging.

If successful, the shell command must exit with an exit code of `0`. All other exit codes will be treated as an operation failure.

9.7.11.2.1.1 Examples of RunCommand operations

You can input the full command contents into the RunCommand operation.

```
remove_dir="$DIRECTORY_TO_REMOVE_ENVIRONMENT_VARIABLE"

if test -d "$remove_dir"; then
    rm -rf "$remove_dir" || exit 1
fi

exit 0
```

If a script already exists on the remote environment and is executable by the environment user, the RunCommand operation can execute this script directly.

```
/opt/app/oracle/product/10.2.0.5/db_1/dbs/myscript.sh "$ARG_ENVIRONMENT_VARIABLE"
"second argument in double quotes" 'third argument in single quotes'
```

9.7.11.2.2 RunBash operation

The RunBash operation runs a Bash command on a Unix environment using a `bash` binary provided by the Delphix Engine, unless it's a Linux environment, in which case it uses the system's native bash binary. The environment user runs this Bash command from their home directory. The Delphix Engine captures and logs all output from this command. If the script fails, the output is displayed in the Delphix Management application and command-line interface (CLI) to aid in debugging.

If successful, the Bash command must exit with an exit code of `0`. All other exit codes will be treated as an operation failure.

9.7.11.2.2.1 Example of RunBash operations

You can input the full command contents into the RunBash operation.

```
remove_dir="$DIRECTORY_TO_REMOVE_ENVIRONMENT_VARIABLE"

# Bashisms are safe here!
if [[ -d "$remove_dir" ]]; then
    rm -rf "$remove_dir" || exit 1
fi

exit 0
```

9.7.11.2.3 Shell operation tips

9.7.11.2.3.1 Using `nohup`

You can use the `nohup` command and process backgrounding from resource in order to "detach" a process from the Delphix Engine. However, if you use `nohup` and process backgrounding, you MUST redirect `stdout` and `stderr.`

Unless you explicitly tell the shell to redirect `stdout` and `stderr` in your command or script, the Delphix Engine will keep its connection to the remote environment open while the process is writing to either `stdout` or `stderr` . Redirection ensures that the Delphix Engine will see no more output and thus not block waiting for the process to finish.

For example, imagine having your `RunCommand` operation background a long-running Python process. Below are the bad and good ways to do this.

Bad Examples
• `nohup python file.py & # no redirection` • `nohup python file.py 2>&1 & # stdout is not redirected` • `nohup python file.py 1>/dev/null & # stderr is not redirected` • `nohup python file.py 2>/dev/null & # stdout is not redirected`

Good Examples
• `nohup python file.py 1>/dev/null 2>&1 & # both stdout and stderr redirected, Delphix Engine will not block`

9.7.11.3 Other operations

9.7.11.3.1 RunExpect operation

The RunExpect operation executes an Expect script on a Unix environment. The Expect utility provides a scripting language that makes it easy to automate interactions with programs which normally can only be used interactively, such as `ssh` . The Delphix Engine includes a platform-independent implementation of a subset of the full Expect functionality.

The script is run on the remote environment as the environment user from their home directory. The Delphix Engine captures and logs all output of the script. If the operation fails, the output is displayed in the Delphix Management application and CLI to aid in debugging.

If successful, the script must exit with an exit code of `0`. All other exit codes will be treated as an operation failure.

9.7.11.3.1.1 Example of a RunExpect operation

Start an `ssh` session while interactively providing the user's password.

```
spawn ssh user@delphix.com
expect {
    -re {Password: } {
        send "${env(PASSWORD_ENVIRONMENT_VARIABLE)}\n"
    }
    timeout {
        puts "Timed out waiting for password prompt."
        exit 1
    }
}
exit 0
```

9.7.11.4 Oracle environment variables

to access the dSource or VDB.

9.7.11.4.1 dSource environment variables

Environment variable	Description
`ORACLE_SID`	The SID of the dSource
`ORACLE_BASE`	The home directory of the Oracle software hosting the dSource
`ORACLE_HOME`	The Oracle Home for the dSource
`CRS_HOME` (only set for RAC dSources)	The home directory for cluster services hosting the dSource

Environment variable	Description
`ORAENV_ASK`	Always set to NO
`DELPHIX_DATABASE_NAME`	The database name reported by Oracle
`DELPHIX_DATABASE_UNIQUE_NAME`	The database unique name reported by Oracle
`DELPHIX_PDB_NAME` (only set for PDBs)	The PDB name reported by Oracle

9.7.11.4.2 VDB environment variables

Environment variable	Description
`ORACLE_SID`	The SID for the VDB
`ORACLE_BASE`	The home directory for the Oracle software hosting the VDB
`ORACLE_HOME`	The Oracle Home for the VDB
`CRS_HOME` (only set for RAC VDBs)	The home directory for cluster services hosting the RAC VDB
`ORAENV_ASK`	Always set to NO
`DELPHIX_DATABASE_NAME`	The database name reported by Oracle
`DELPHIX_DATABASE_UNIQUE_NAME`	The database unique name reported by Oracle
`DELPHIX_PDB_NAME` (only set for PDBs)	The PDB name reported by Oracle
`DELPHIX_MOUNT_PATH`	The root of the NFS mount hosting the VDB data

Environment variable	Description
`MASKING_CONNECTOR_HOST (only set for masked provisioning)`	The host that DMSuite will use for the connector
`MASKING_CONNECTOR_PORT (only set for masked provisioning)`	The port that DMSuite will use for the connector

PATH and LD_LIBRARY_PATH configuration

`PATH` is configured by appending the `bin` directory in the Oracle home for the dSource or VDB. `LD_LIBRARY_PATH` is configured by appending the `lib` directory in the Oracle home for the dSource or VDB.

9.8 Oracle E-Business Suite (EBS) environments and data sources

This section contains the following topics:

- Oracle E-Business Suite and Delphix conceptual overview (see page 1652)
- Oracle EBS virtualization process overview (see page 1656)
- Oracle EBS requirements and prerequisites (see page 1656)
- Host Checker for external connectors (see page 1681)
- Oracle EBS installation and upgrade (see page 1686)
- Oracle EBS connector operations (see page 1689)
- Troubleshooting Oracle EBS (see page 1755)
- Oracle EBS additional information (see page 1756)
- Oracle EBS glossary (see page 1765)
- Oracle EBS FAQs (see page 1766)

9.8.1 Oracle E-Business Suite and Delphix conceptual overview

This topic provides a conceptual overview of the integration between the Delphix Continuous Data Engine and Oracle E-Business Suite (EBS).

9.8.1.1 Why use the Delphix Continuous Data Engine with the Oracle E-Business Suite?

Oracle E-Business Suite (EBS) offers a multitude of enterprise resource planning (ERP), supply chain management (SCM), and customer relationship management (CRM) applications useful for running a business. Typical EBS instances are consumed horizontally across an organization but managed on-site by an IT organization. Most businesses hire consultants or in-house developers to customize these instances to fit business needs.

Production EBS instances are critical to many business processes; EBS is expected to be always available and functioning correctly. An IT organization's EBS team is responsible for maintaining this production instance: this team monitors the instance and manages how software modifications are integrated. Common EBS software modifications include patches provided by Oracle or new application logic provided by consultants or in-house developers.

However, before making a software modification to their production EBS instance, most EBS teams vet the modification. These teams typically maintain a series of non-production copies of their production EBS instance to use for the vetting process. To vet a modification, the EBS team will modify a non-production instance in an attempt to identify bugs in the modification that might lead to incorrect functionality, performance regressions, or downtime. Once a modification has been installed and tested successfully, it is considered vetted and the EBS team makes the same modification to the production instance.

The problem with this process is that maintaining non-production copies of EBS is costly. EBS teams not only need to provision and maintain resources for each copy, they also must update each copy to match production every time production itself is modified. This process takes two hours at a minimum and can take up to a week or more.

The Delphix Continuous Data Engine can alleviate much of the pain around managing non-production copies of Oracle E-Business Suite. You can link a production EBS instance to the Delphix Continuous Data Engine and use the engine to provision virtual copies. These virtual copies lower storage costs associated with non-production. They also drastically reduce the amount of active attention required from the EBS team. When a change is made to the production EBS instance, the Delphix Continuous Data Engine can synchronize its copy; all virtual EBS instances can then be refreshed to match this new version of the production. The process of refreshing a virtual EBS instance is fully automated, so an EBS team no longer needs to invest copious amounts of time refreshing non-production instances manually.

In addition, the Delphix Continuous Data Engine can take snapshots of virtual EBS instances in known healthy states so that you can easily roll back risky modifications while vetting them. This functionality provides modern version control semantics for a process that previously had none.

9.8.1.2 Overview of linking and provisioning Oracle EBS

The Oracle EBS Suite as described here[468] can be virtualized using Delphix Continuous Data Engine. We can virtualize the Application Tier and Database Tier. The Database Tier is made up of two different datasets: the Oracle Database and the Database Technology Stack herewards referred as DBTechstack. The Application Tier (AppsTier) and Database Technology Stack (DBTechstack) is managed by the Oracle EBS connector while the Oracle Database is managed by the Oracle connector. The Delphix Continuous Data Engine can also manage custom extensions and plug-ins.

The process of linking EBS data involves creating multiple dSources:

468 https://docs.oracle.com/cd/E26401_01/doc.122/e22949/T120505T120508.htm

- a dSource for the DBTechstack
- a dSource for the Oracle database used by EBS
- a dSource for the AppsTier

These dSources are collectively referred to as EBS dSources. They are also the source datasets from which you can provision virtual EBS instances.

The process of provisioning a virtual EBS instance involves provisioning from each of these dSources separately to the proper environments in sequence. You can add custom configuration logic to the Delphix Continuous Data Engine for each EBS instance such that the linking, provisioning, and refresh processes are fully automated.

> The connector versions for AppsTier and DB Tech Stack need to be the same, at installation and through the upgrade. Installation and usage of different versions of these connectors are not supported and may encounter operational issues.

9.8.1.3 DBTechStack dataset

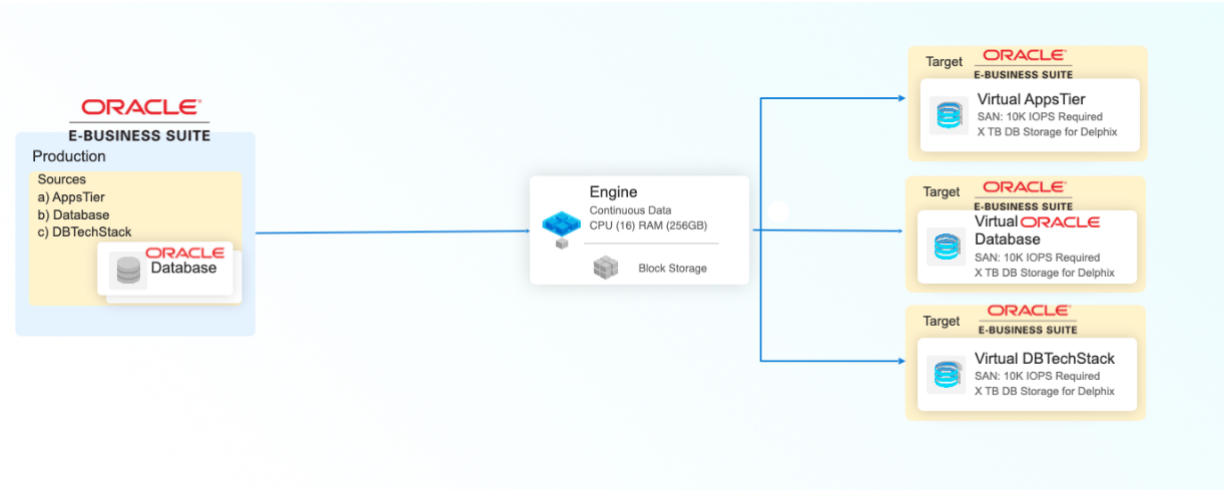
ORACLE_HOME is referred to as DBTechstack. The source DBTechstack(ORACLE_HOME) is linked using the Delphix Continuous Data Engine's EBS connector. The linking process runs pre-clone(adpreclone utility) logic to ensure the readiness of EBS configuration for data capture. When you provision EBS, the Delphix Continuous Data Engine automates post-clone configuration such that a copy of the DBTechstack is available for use on the target dbTier server with no additional effort. You can add this copy of the DBTechstack to the Delphix Continuous Data Engine as an Oracle installation home and use it to host an EBS virtual database (VDB).

9.8.1.4 Database dataset

The database dSource is linked and ingested using the Delphix Continuous Data Engine's native Oracle data source functionality. This dSource contains database data files that EBS is currently using. For more information about managing Oracle databases, refer Oracle data sources (see page 1332).

9.8.1.5 AppsTier dataset

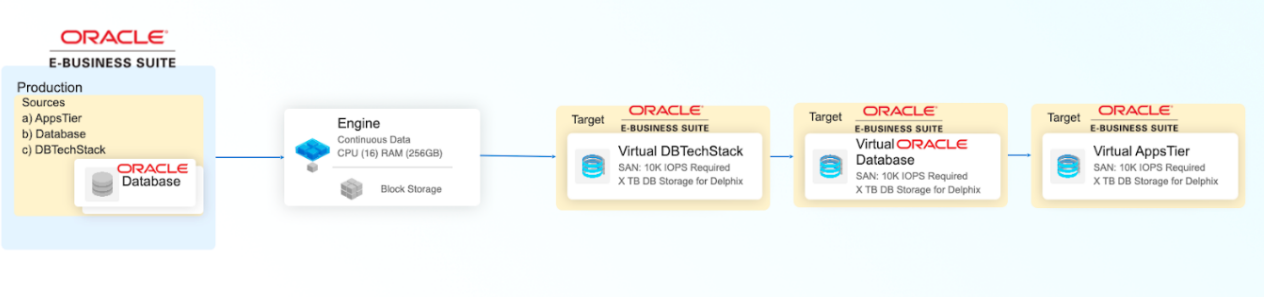
The AppsTier is linked using the Delphix Continuous Data Engine's EBS support: the linking process automatically runs pre-clone logic to ensure EBS configuration is always appropriately staged at the time of data capture. When you provision EBS, the Delphix Continuous Data Engine will automate post-clone configuration such that a copy of the AppsTier is available for use on the target AppsTier server. This virtual copy of the AppsTier will connect to the provisioned EBS virtual database (VDB).



9.8.1.5.1 Sequence of provisioning

Unlike linking of data sources which can be done in any order or parallelly, provisioning the EBS application has to be done in a specific order mentioned below:

Order	Dataset
1	**DBTechstack**
2	**Database**
3	AppsTier



9.8.1.6 How does it work?

The Delphix Continuous Data Engine's linking and provisioning logic follows the Direct Ingestion architecture. More information about this architecture method can be found in the Direct Ingestion section (see page 1014)

The Delphix Engine's linking and provisioning logic follows the Oracle-recommended cloning procedures outlined in the following Oracle Support documents found at http://docs.oracle.com[469]:

469 http://docs.oracle.com/en/

- Cloning Oracle E-Business Suite Release 12.2 with Rapid Clone (Doc ID 1383621.1)
- Cloning Oracle Applications Release 12 with Rapid Clone (Doc ID 406982.1)

9.8.2 Oracle EBS virtualization process overview

The following material introduces you to the overall process of deploying the Oracle EBS virtualization solution. A sequence of high-level steps is presented; each step briefly describes the task and the expected result. Refer to the embedded links for more detailed information.

- Begin by reviewing the general Delphix Continuous Data Engine architecture diagrams (see page 1014). Identify existing or new Oracle EBS source and target Environments, the underlying Operating Systems, and Oracle EBS versions. This will give a good understanding of the architectural strategy and confirm support.
- Reviewing the Oracle EBS Support Matrix (see page 1657). Confirm that your combination of Oracle EBS software, host operating system, Delphix Engine version and Oracle EBS connector version are supported for the various environments which will integrate into your Oracle EBS deployment. Early validation will reduce unexpected issues later.
- Deploy and configure Delphix Continuous Data.
- Installing the Oracle EBS connector (see page 1686). The Oracle EBS connectors extend the Delphix Continuous Data Engine to Oracle EBS datasets. It is recommended to stay current with the latest version of the connector.
- Preparing source and creating target environments (see page 1656). Prepare source and create the target hosts and ensure that they are running and reachable. Follow the host requirements to configure the respective environments. The source environment is the R12.2 E-Business Suite that the Delphix Continuous Data Engine will ingest from. The target environment will host one or more EBS instance(s) in which the virtual database copies will be created.
    a. Consult the source environment (see page 1664) and target environment (see page 1667) requirements documentation for configuration details.
- Discovering source and target environments in the Delphix Continuous Data engine (see page 1690). Use the "Add Environment" operation in Delphix Continuous Data Engine for each host, which causes Delphix to register the specific host and catalog the EBS software installation(s) on that host.
- Linking the source database. (see page 1693) Configure and create a dSource which allows the Delphix Continuous Data Engine to capture a copy of the source data. There are many ingestion methods that can offer a variety of methods to capture the data to be virtualized by Delphix. Therefore, we encourage you to identify the ingestion method that best suits your organization and infrastructure policies.
- Provisioning virtual EBS dataset (see page 1700). (see page 1693) Administrators and users can provision multiple copies of the source data as virtual copies.

9.8.3 Oracle EBS requirements and prerequisites

The section covers the following topics:

- Oracle EBS support matrix (see page 1657)
- Requirements for Oracle EBS source data sources (see page 1664)
- Requirements for Oracle EBS target data sources (see page 1667)

- Sudo privileges for the Oracle EBS environment users (see page 1672)
- Privilege elevation requirements for Oracle EBS (see page 1673)

9.8.3.1 Oracle EBS support matrix

Note: EBS connector version 4.3.0 onwards can handle provision of virtual DBTechstack if AD/TXK codelevel is 13 or later where adpreclone.pl is shipped with version 120.31.12020000.27 or later.

In accordance with Delphix Support Policies, when a minor release is certified, all the patch releases falling under that minor release are also considered certified.

9.8.3.1.1 Oracle EBS connector and Delphix Continuous Data Engine compatibility

Refer to the following support matrices to ensure compatibility between the environment, database, connector, and Delphix Continuous Data Engine.

Compatible upgrades have a green Y, while incompatible upgrades show a red N.
NA means Not applicable.

Engine version	Connector Version for EBS 12.2 with Oracle 19c																	
	1.8.2	1.8.3	1.8.4	2.0.0	2.0.1	3.0.0	3.0.1	3.0.2	3.0.3	4.0.2	4.1.0	4.2.0	4.2.1	4.2.2	4.3.0	4.4.0	4.5.0	4.6.0
6.0.0.0 to 6.0.8.0	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.9.0	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N

6.0.10.0	N	N	N	N	N	Y	Y	N	N	N	N	N	N	N	N	N	N	N
6.0.11.0	N	N	N	N	N	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N
6.0.12.0	N	N	N	N	N	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N
6.0.13.0	N	N	N	N	N	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N
6.0.14.0	N	N	N	N	N	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N
6.0.15.0	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N
6.0.16.0	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N
6.0.17.0	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N
7.0.0.0	Y	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y
8.0.0.0	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
9.0.0.0	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
10.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
11.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
12.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y

13.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
14.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
15.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
16.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
17.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
18.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
19.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
20.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
21.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
22.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y

Engine Version	**Connector Version for EBS 12.2 with Oracle 12c**																	
	1.8.2	**1.8.3**	**1.8.4**	**2.0.0**	**2.0.1**	**3.0.0**	**3.0.1**	**3.0.2**	**3.0.3**	**4.0.2**	**4.1.0**	**4.2.0**	**4.2.1**	**4.2.2**	**4.3.0**	**4.4.0**	**4.5.0**	**4.6.0**
6.0.0.0	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N

6.0.1.0	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N
6.0.2.0	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N
6.0.3.0	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N
6.0.4.0	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N
6.0.5.0	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N
6.0.6.0	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N
6.0.7.0	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N
6.0.8.0	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N
6.0.9.0	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N
6.0.10.0	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N
6.0.11.0	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N
6.0.12.0	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N
6.0.13.0	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N
6.0.14.0	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N
6.0.15.0	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N
6.0.16.0	N	N	N	N	N	N	N	N	N	Y	Y	N	N	N	N	N	N	N
6.0.17.0	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N
7.0.0.0	N	N	N	N	N		N	N	N	N	N	N	Y	Y	Y	Y	Y	Y
8.0.0.0	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y

9.0.0.0	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
10.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
11.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
12.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
13.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
14.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
15.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
16.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
17.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
18.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
19.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
20.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
21.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y
22.0.0.x	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y

9.8.3.1.2 Operating system and Database compatibility

The Oracle EBS connector is certified for use in source, and target environments with the following operating systems and database versions.

Supported OS version	**Supported EBS 12.2 Apps version**	
	Oracle 11g/12c	**Oracle 19c**

RHEL 7.0	Y (AD/TXK codelevel 12 or below)	**NA**
RHEL 7.1	Y (AD/TXK codelevel 12 or below)	**NA**
RHEL 7.2	Y (AD/TXK codelevel 12 or below)	**NA**
RHEL 7.3	Y (AD/TXK codelevel 12 or below)	**NA**
RHEL 7.4	Y (AD/TXK codelevel 12 or below)	**NA**
RHEL 7.5	Y (AD/TXK codelevel 12 or below)	Y (AD/TXK codelevel 12 or below)
RHEL 7.6	Y (AD/TXK codelevel 12 or below)	Y (AD/TXK codelevel 12 or below)
RHEL 7.7	Y (AD/TXK codelevel 12 or below)	Y (AD/TXK codelevel 12 or below)
RHEL 7.8	Y (AD/TXK codelevel 12 or below)	Y (AD/TXK codelevel 12 or below)
RHEL 7.9	Y (AD/TXK codelevel 12 or below)	Y (AD/TXK codelevel 12 or below)
RHEL 8.0	**NA**	Y (AD/TXK codelevel 12 or below)
RHEL 8.1	**NA**	Y (AD/TXK codelevel 12 or below)
RHEL 8.2	**NA**	Y (AD/TXK codelevel 12 or below)

RHEL 8.3	**NA**	Y (AD/TXK codelevel 13 or later)
RHEL 8.4	**NA**	Y (AD/TXK codelevel 13 or later)
RHEL 8.5	**NA**	Y (AD/TXK codelevel 13 or later)
RHEL 8.6	**NA**	Y (AD/TXK codelevel 13 or later)
RHEL 8.7	**NA**	Y (AD/TXK codelevel 13 or later) 11.0.0.0+
RHEL 8.8	**NA**	Y (AD/TXK codelevel 13 or later) 11.0.0.0+
RHEL 8.9	**NA**	Y (AD/TXK codelevel 13 or later) 11.0.0.0+
Solaris 11 U4 (SPARC)	**Y**	Y (AD/TXK codelevel 12 or below) 6.0.17.x+

9.8.3.1.3 Supported topologies

Database topology	Supported versions
Oracle single-instance (SI) DBTechstack and Database	Oracle11gR2 v11.2.0.4 Oracle12cR1 v12.1.0.2 Oracle 19c MT v19.3.0.0 or later

Oracle RAC DBTechstack and Database	Oracle11gR2 v11.2.0.4 Oracle12cR1 v12.1.0.2

9.8.3.2 Requirements for Oracle EBS source data sources

9.8.3.2.1 Requirements for Source DBTechstack

This topic outlines the requirements for the Oracle EBS source.

The Delphix Continuous Data Engine supports linking a variety of versions and configurations of Oracle E-Business Suite R12.2. Below, detailed compatibility notes are outlined.

Note: The Delphix Continuous Data Engine does not provide support for linking an EBS R12.2 instance utilizing custom context variables maintained in the EBS context file.

9.8.3.2.2 Source DBTechstack requirements:

Source environment requirement	Explanation
Database open_mode	The source site must be in READ_WRITE (or) READ ONLY (or) READ ONLY WITH APPLY mode.
ETCC patch level	ETCC patch level (see page 1758) should be the latest one.
Permission to run ps, nfso, netstat and lsof on AIX, solaris, and Linux hosts, as super-user. This permission is usually granted via sudo authorization of the commands.	See the topic Sudo Privilege Requirements for Oracle Environments[470] for further explanation of this requirement, and Sudo File Configuration Examples for Oracle Environments[471] for examples of file configurations

Host requirement	Explanation

470 https://cd.delphix.com/docs/latest/sudo-privilege-requirements-for-oracle-environment
471 https://cd.delphix.com/docs/latest/sudo-file-configuration-examples-for-oracle-enviro

Profile and privileges must be the same as the Oracle user (e.g. oracle) on the host.	For example, delphix_os should have the same environment variable settings ($PATH, $ORACLE_HOME, etc.), umask, and ulimit settings, as the user oracle.
The Delphix software owner account (e.g. delphix_os) must have the same primary OS group as the Oracle software owner account (e.g. oracle).	Delphix recommends giving the delphix_os user the same primary OS group as the Oracle home owner. More info can be found in the Oracle EBS Glossary (see page 1765).
The delphix_os user must have the Oracle OSDBA group (typically dba) as a primary or secondary OS group.	The OSDBA group allows "OS authentication" when connecting to an Oracle database instance by specifying neither username or password (i.e. rman target /), thus eliminating the need to store or retrieve an SYSDBA password. For more info on delphix_os user, check Oracle EBS Glossary (see page 1765)
For secondary group requirements, the Delphix OS user must be a member of the SYSBACKUP OS group (12.1 or higher) or the SYSDBA OS group (11.2 and lower).	This ensures that the Delphix engine is able to take snapshots of source databases using RMAN.
There must be a directory on the source host where the Delphix Engine Toolkit can be installed, for example: /var/opt/delphix/toolkit	The delphix_os user and primary OS group (i.e. oinstall) must own the directory. For more info on delphix_os user, check Oracle EBS Glossary (see page 1765)
The Delphix Engine must be able to make an SSH connection to the source host (typically port 22).	
Read access to $ORACLE_HOME and all underlying files and directories.	Delphix needs to run rsync to ingest the ORACLE_HOME in the Continuous data engine.

For successful linking of DBTechstack from the standby site, Make sure following points are followed:

Source environment requirement	**Explanation**
If Source is physical standby database	Source database must be in READ ONLY/ READ ONLY WITH APPLY mode. Even during subsequent snapsync operations, you have to make sure that the standby database container & its PDB is in READ ONLY / READ ONLY WITH APPLY mode.

Source PDB services status	The source PDB must be in READ ONLY mode. The two default EBS PDB services ebs_<PDBname>, <PDBname>_ebs_patch must be in READY STATE.

The Oracle database is ingested using Continuous Data engine's native functionality, So refer oracle database requirement guidelines as below:

- Requirements for Oracle hosts and databases[472]
- Oracle data source requirements[473]
- Delphix architecture with Oracle[474]

9.8.3.2.3 Requirements for Source AppsTier

9.8.3.2.4 Supported Oracle EBS AppsTier set-ups

AppsTier topology	Support
Single-node AppsTier	Supported
Multi-node AppsTier (SHARED APPL_TOP, NON-SHARED APPL_TOP)	Supported

Source AppsTier requirement	Explanation
Source weblogic Adminserver must be UP & running before taking snapshot	To stage source AppsTier, adpreclone.pl utility is called which has a requirement that adminserver should be up & running.
Permission to run: • ps on Linux, AIX, and solaris as super-user. • nfso on AIX hosts as super-user	The following permissions are usually granted via sudo authorization of the commands. See Sudo Privilege Requirements for Oracle Environments[475] for further explanation of the commands, and Sudo File Configuration Examples for Oracle Environments[476]

472 https://cd.delphix.com/docs/latest/requirements-for-oracle-hosts-and-databases
473 https://cd.delphix.com/docs/latest/requirements
474 https://cd.delphix.com/docs/latest/delphix-architecture-with-oracle
475 https://cd.delphix.com/docs/latest/sudo-privilege-requirements-for-oracle-environment
476 https://cd.delphix.com/docs/latest/sudo-file-configuration-examples-for-oracle-enviro

	for examples of the /etc/sudoers file on different operating systems.

9.8.3.3 Requirements for Oracle EBS target data sources

This topic outlines the requirements for the EBS source databases.

9.8.3.3.1 Requirements for Target DBTechstack environments

Provisioning of Oracle RAC Home is not supported: The Delphix Continuous Data Engine does not support provisioning Oracle RAC DBTechstack for use with Oracle E-Business Suite.

However, you may provision an Oracle standalone DBTechstack from a linked EBS

Follow these requirements to ensure that your environment is configured accordingly:

Target environment requirement	Explanation
Ensure your target EBS R12.2 instance is supported	See Virtual EBS R12.2 instance requirements to ensure you can provision your EBS R12.2 instance.
Ensure your EBS 12.2 environments comply with Oracle's documentation	Your environments must comply with Oracle's requirements for installing EBS. These requirements are outlined on **Oracle E-Business Suite Release Notes, Release 12.2 (Doc ID 1320300.1)** found at https://support.oracle.com. Oracle has released an E-Business Suite Pre-Install RPM (available on ULN and public yum) that includes all required RPMs for both the AppsTier and DBTier of an R12.2 installation. Details can be found in Oracle E-Business Suite Installation and Upgrade Notes Release 12 (12.2) for Linux x86-64 (Doc ID 1330701.1) found at https://support.oracle.com.
Unix requirements	The dbTier must meet the target requirements outlined in Requirements for Unix Environments. These requirements are generic to all target Unix environments you add to the Delphix Continuous Data Engine.

oracle user	The Delphix Continuous Data Engine must have access to an oracle user on the DBTier. This user should be a member of both the EBS dba and oinstall groups. This user will be given proper permissions to manage the DBTechstack and database. The user should own the stage directory for DBTechstack and database files that will be cloned.
oraInst.loc	An oraInst.loc file must exist on the dbTier prior to provisioning. This file will specify where the oraInventory directories live or where they should be created if they do not already exist. The oraInst.loc file is typically located at /etc/oraInst.loc on Linux or /var/opt/oracle/oraInst.loc on Solaris. Ensure that the oraInventory to which this file points is writable by the oracle user. Consult Oracle EBS documentation for more information about where to place this file on your dbTier and what this file should contain.

/etc/hosts	For compatible Oracle Linux, RHEL, AIX and Solaris environments make sure that the /etc/hosts file is formatted as follows:
	`127.0.0.1 localhost.localdomain localhost`
	`[ip_address] [node_name].[domain_name] [node_name]`
	Hostname and its IP address of Database host and AppsTier host(s) (primary and secondary hosts for multi node) should be added in the /etc/hosts file of all the Database, host, AppsTier host(s) (primary and secondary hosts for Multi Node).
	These correct entries are required to avoid adcfgclone utility timeout issue on the prompt for "**Target System Domain Name**".

Permission to run: • ps on Linux, AIX, and solaris as super-user. • nfso on AIX hosts as super-user • rootpre.sh on AIX hosts as super-user(on target DB server)	The following permissions are usually granted via sudo authorization of the commands. See Sudo Privilege Requirements for Oracle Environments[477] for further explanation of the commands, and Sudo File Configuration Examples for Oracle Environments[478] for examples of the /etc/sudoers file on different operating systems.
Kernel shmmax and shmall values	The minimum required kernel.shmmax value on target hosts should be at least half the size of physical memory in bytes. The minimum required kernel.shmall value on target hosts should be greater than or equal to the value of shmmax, in pages. For more information refer to Oracle documentation: https://docs.oracle.com/en/database/oracle/oracle-database/19/ladbi/minimum-parameter-settings-for-installation.html#GUID-CDEB89D1-4D48-41D9-9AC2-6AD9B0E944E3

9.8.3.3.2 Requirements for Target database environments

Follow these requirements outlined in Oracle support and requirements to ensure that your environment is configured accordingly. These requirements are generic to all target Unix environments expected to host a virtual Oracle database.

9.8.3.3.3 Requirements for Target AppsTier environments

Follow these requirements to ensure that your environment is configured accordingly:

The AppsTier must meet the target requirements outlined in Requirements for Unix environments. These requirements are generic to all target Unix environments added to the Delphix Continuous Data Engine.

Target environment requirement	Explanation
`applmgr` user or OS user which will own the application binary file system	The Delphix Continuous Data Engine must have access to an applmgr user on the AppsTier. This user should be a member of the EBS oinstall group. The user should own the stage directory for AppsTier files that will be cloned.

477 https://cd.delphix.com/docs/latest/sudo-privilege-requirements-for-oracle-environment
478 https://cd.delphix.com/docs/latest/sudo-file-configuration-examples-for-oracle-enviro

delphix_os OS user account	In order to separate authentication and perform privileged operations with a non-privileged OS account, first create an OS user account (i.e. "delphix_os") on the EBS DB Tier and AppsTier node to be used as a target. This user is easily created by the createDelphixOSUser.sh (located below on this page) script. The primary OS group of the Delphix Continuous Data Engine software owner account (i.e. delphix_os) must be the same as the EBS Database/AppsTier software owner account (i.e. oravis or applmgr). Primary group = Oracle Install Group (typically oinstall), secondary group = OSDBA Group (typically dba). There are lots of cases where the OS group named dba fills this role, so be sure to check the group membership of the EBS Database and AppsTier software owner account. Please note, the non-privileged OS account must have the **same** group as assigned to EBS Database or AppsTier privileged account (like oravis or applmgr).
utilities	Ensure that "expect", "wget" and "lsof" utilities exist on the remote host.

9.8.3.3.4 Multi node(shared APPL_TOP) support for AppsTier

Following are the requirements in order to deploy a multi-node AppsTier. These requirements should be met ***along with*** the requirements for regular AppsTier target environments.

username and GID	The primary host and the secondary host user name and its UID, Group name and GID must be the same to avoid any permission issue during the provisioning operation. Check the high and low privilege user id and GID in the primary and secondary node and change it accordingly using groupmod with the following commands: • `$ id oravis` • `uid=509(oravis) gid=502(oinstall)` • `$ id delphix_os` • `uid=510(delphix_os) gid=502(oinstall)` • `Change the group id as root using,` • ``` • ``` • `# groupmod -g 502 oinstall` • `# usermod -g oinstall oravis` `# usermod -g oinstall delphix_os`
passwordless ssh	Primary and secondary AppsTier hosts need to be configured with passwordless ssh if the target WLS (see page 1765) password is going to be different from the source WLS password or for configuring WLS password different from source.
Oracle central inventory should be on common shared location between Application servers	While configuring shared APPL_TOP application, instead of storing oraInventory on the local directory of each server, all application servers should point to the same oraInventory directory on shared Delphix mount storage, so that you need to patch only on a single server and the oraInventory gets in-sync for all application servers thus preventing the oraInventory of remaining servers becoming invalid. Details can be found in Sharing The Application Tier File System in Oracle E-Business Suite Release 12 (Doc ID 384248.1) found at https://support.oracle.com. You can also consult Oracle EBS documentation for more information about where to place this file on your AppsTier and what this file should contain.

9.8.3.4 Sudo privileges for the Oracle EBS environment users

An environment user account on the environment must be selected to communicate with both the staging and target environments. As required by the EBS connector, you must grant this environment user the following two key privileges:

Disable requiretty

The Delphix Continuous Data Engine requires the `requiretty` setting to be disabled. This allows the environment user to run sudo commands without a logged-in terminal session (tty).

Add NOPASSWD

By including the NOPASSWD qualifier, you ensure that the "sudo" command does not prompt for a password when executing certain commands such as mount, umount, mkdir, and rmdir.

In the example configurations, the environment username **delphix_os** is used. SImilarly the environment user owning Oracle EBS DatabaseTier is named **oravis** and the user owning Oracle EBS AppsTier is named **applvis.** Make sure to modify the username based on the environment user available in the environment for performing the linking and/or provisioning operations.

9.8.3.4.1 Procedure

Perform the following steps on both the staging and target environments.

1. The sudo configuration is located in the **/etc/sudoers** file. Run the following commands to open the `sudoers' file:

```
vi /etc/sudoers
```

> **Note:** Third party tools like `visudo`, can be helpful in minimizing syntax errors.

2. Next, update the file with the following snippet to grant the required privileges to the `delphix_os` user:

```
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD: /bin/su - oravis -c /bin/mount, /bin/su - oravis -c /
bin/umount
```

Similar to above snippet, add the entries for all the target and source host environments to elevate the privileges of the delphix_os user. The following sudoers entry is only for template purposes. Modify the path in the below sudoers entry with the appropriate binary paths of your environment.

9.8.3.4.2 Creating delphix_os user

The following script can be used to create the delphix_os user.

createDelphixOSUser.sh[479]

9.8.3.5 Privilege elevation requirements for Oracle EBS

The optional Privilege Elevation feature allows a non-privileged environment user (e.g., delphix_os) to gain the permissions of a privileged environment user (e.g., oravis) for operations on the staging or target environments. This configuration can be valuable to maximize an environment's security posture.

9.8.3.5.1 Directions

First, the Privilege Elevation script, dlpx_db_exec, must be created and uploaded onto the Delphix Continuous Data Engine. Then, the non-privileged environment user must be granted the appropriate permissions..

9.8.3.5.2 Upload the privilege elevation script

Perform one of the following methods to upload the privilege elevation script:

a) CURL file upload method

Before running any of the following commands, update <*CONTINUOUS-DATA-FQDN-or-IP*>, <*USERNAME*>, and <*PASSWORD*> placeholder values.

1. Create a session with the Delphix Continuous Data Engine.
    a. Also, update the version values for `major`, `minor`, and `micro` fields.

    The correct values for your Delphix Continuous Data Engine version can be found in the API version information[480].

479 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149466510?preview=%2F149466510%2F149466521%2FcreateDelphixOSUser.sh

480 https://cd.delphix.com/docs/latest/api-version-information

```
curl -i -c cookies.txt -X POST -H "Content-Type:application/json" https://
<CONTINUOUS-DATA-FQDN-or-IP>/resources/json/delphix/session -d '{
    "version":{
        "major":1,
        "minor":11,
        "micro": 5,
        "type":"APIVersion"
    },
    "type":"APISession"
}'
```

2. Login to Delphix Continuous Data Engine as the `*admin*` user.

```
curl -i -c cookies.txt -b cookies.txt -X POST -H "Content-Type:application/
json" https://<CONTINUOUS-DATA-FQDN-or-IP>/resources/json/delphix/login -d '{
    "username":"<USERNAME>",
    "password":"<PASSWORD>",
    "type":"LoginRequest",
    "target":"DOMAIN"
}'
```

3. Copy *DLPX_DB_EXEC* contents to the Delphix Continuous Data Engine.

```
curl -i -b cookies.txt -X POST -H "Content-Type:application/json" https://
<CONTINUOUS-DATA-FQDN-or-IP>/resources/json/delphix/host/privilegeElevation/
profileScript/HOST_PRIVILEGE_ELEVATION_PROFILE_SCRIPT-7 -d '{
    "type": "HostPrivilegeElevationProfileScript",
    "contents": "#\n# Copyright (c) 2018 by Delphix. All rights reserved.
\n#\n#\n# This script allows customization of command execution with an
alternate user\n# account.\nif [[ $1 != -u* ]]; then\n    echo \"Incorrect
command line parameters, -u<optional user account> is required as the first
parameter\"\n    exit 1\nfi\nuser_id=`echo $1 | sed -e \"s\/^-u\/\/\"`\nshift
1\nif [[ $user_id != \"delphix_os\" ]]; then\n    command=$(printf \"%s\"
\"$@\")\n    cd /tmp\n    sudo -E su $user_id -p -c \"$command\"\nelse\n
$@\nfi\n"
}'
```

Note: If the *dlpx_db_exec* script is updated after the environment(s) is added, refresh the environment to propagate the changes.

CLI Method

For steps on creating a Privilege Elevation Profile, refer to [CLI Cookbook: How to create or edit privilege elevation profiles and profile scripts](481).

9.8.3.5.3 Configure sudo privileges to grant Privilege Elevation

With Privilege Elevation, all EBS connector commands are run via the **dlpx_db_exec** script and within this script the commands are run using 'sudo'. In order to execute the connector commands as the high-privileged environment user, sudoers entries are required.

To add additional sudoers entries you need to edit the **/etc/sudoers** file. Run the following commands to open the `sudoers' file:

```
vi /etc/sudoers
# or
visudo
```

> **Note:** Third party tools like `visudo`, can be helpful in minimizing syntax errors.

9.8.3.5.4 Sudoers entry for Linux

Entries required for provisioning via low privileged user (delphix_os) on **target** EBS DB Tier

```
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD: /bin/su - oravis -c
/bin/mount, /bin/su - oravis -c
/bin/umount, /bin/su - oravis -c echo *,
/bin/su - oravis -c export *,
/bin/su - oravis -c test*mkdir*,
/bin/su - oravis -c test*touch*chmod 750*cat*,
/bin/su - oravis -c */EBS_kill/kill_script.sh*,
/bin/su - oravis -c rm -f */test_status.tmp*,
/bin/su - oravis -c cp -f */dlpx_force_autoflush*;
chmod 755 */dlpx_force_autoflush*,
/bin/su - oravis -c rm -f */dlpx_force_autoflush*,
/bin/su - oravis -c */*.env*bin/lsnrctl status*,
/bin/su - oravis -c */*.env*/addlnctl.sh*,
/bin/su - oravis -c rm -f /u01/oracle/VIS/*/.delphix_adpreclone.lck*,
/bin/su - oravis -c rm -rf */appsutil/clone/dbts*,
/bin/su - oravis -c touch /u01/oracle/VIS/*/.delphix_adpreclone.lck*,
/bin/su - oravis -c */bin/runInstaller -silent -detachHome*,
/bin/su - oravis -c export PATH* export PERL5LIB*perl -mdlpx_force_autoflush */
adcfgclone.pl dbtechstack*,
```

481 https://cd.delphix.com/docs/latest/cli-cookbook-how-to-create-or-edit-a-privilege-ele

```
/bin/su - oravis -c export PERL5LIB*perl -mdlpx_force_autoflush */adpreclone.pl
dbtechstack*,
/bin/su - oravis -c cd *; make -f ins_rdbms.mk*ioracle*,
/bin/su - oravis -c cd *; make -f ins_rdbms.mk dnfs_off*,
/bin/su - oravis -c rm -f *bak, /bin/su - oravis -c mv *,
/bin/su - oravis -c */*.env*sqlplus* as sysdba*,
/bin/su - oravis -c */*.env*perl */appsutil/clone/bin/adcfgclone.pl dbconfig*,
/bin/su - oravis -c */*.env*perl */appsutil/scripts/*/adpreclone.pl database*,
/bin/su - oravis -c sed*sqlnet.ora*, /bin/su - oravis -c */*.env*sqlplus apps*,
/bin/su - oravis -c */*.env*; make -f *rdbms/lib/ins_rdbms.mk dnfs_off*,
/bin/su - oravis -c */*.env*; ln -s *, /bin/mount, /bin/umount, /bin/ps,
/bin/mkdir, /bin/su - oravis -c source* -dboraclehome* perl*txkGenCDBTnsAdmin.pl*,
/bin/su - oravis -c */*.env*perl */appsutil/bin/txkCfgUtlfileDir.pl -contextfile*,
/bin/su - oravis -c */*.env* mkdir -p*, /bin/su - oravis -c chmod 775 *dbs*,
/bin/su - oravis -c chmod 6751 */bin/oracle*,
/bin/su - oravis -c cp *dlpx_force_autoflush.pm*,
/bin/su - oravis -c chmod 755 */dlpx_force_autoflush*,
/bin/su - oravis -c umask*touch *source_apps_file.txt,
/bin/su - oravis -c mkdir -p*, /bin/su - oravis -c cp *pairsfile*,
/bin/su - oravis -c *perl -mdlpx_force_autoflush */adclonectx.pl*,
/bin/su - oravis -c *perl -mdlpx_force_autoflush */adcfgclone.pl dbtechstack*,
/bin/su - oravis -c touch */.delphix_adpreclone.lck*,
/bin/su - oravis -c rm -f */.delphix_adpreclone.lck*
```

Entries required for provisioning via low privileged user (delphix_os) on **target** EBS AppsTier

```
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD: /bin/su - applvis -c echo *,
/bin/su - applvis -c rm *.dlpx_run_edition*,
/bin/su - applvis -c rm -f *dlpx_force_autoflush*,
/bin/su - applvis -c cd *echo *dlpx_force_autoflush*,
/bin/su - applvis -c export PATH* export PERL5LIB* cd *perl -mdlpx_force_autoflush ./
adpreclone.pl AppsTier*,
/bin/su - applvis -c */rsync*, /bin/su - applvis -c test*mkdir*, /bin/su - applvis -c
test*touch*chmod 750*cat*,
/bin/su - applvis -c */EBS_kill/kill_script.sh*, /bin/su - applvis -c rm -f */
test_status.tmp*, /bin/su - applvis -c */*.env* sqlplus -s *apps*,
/bin/su - applvis -c */adstrtal.sh*, /bin/su - applvis -c */adstpall.sh*,
/bin/su - applvis -c */*.env* cd *adautocfg.sh*,
/bin/su - applvis -c export PATH* export PERL5LIB* cd *perl -mdlpx_force_autoflush ./
adcfgclone.pl AppsTier*,
/bin/su - applvis -c *perl -mdlpx_force_autoflush */adclonectx.pl addnode
contextfile*pairsfile*outfile*,
/bin/su - applvis -c *adadminsrvctl.sh*, /bin/su - applvis -c *admanagedsrvctl.sh*,
/bin/su - applvis -c *adnodemgrctl.sh*, /bin/su - applvis -c */*.env* cd *perl
-mdlpx_force_autoflush ./
txkUpdateEBSDomain.pl*contextfile*action*updateAdminPassword*,
/bin/su - applvis -c */bin/runInstaller -silent -detachHome*, /bin/su - applvis -c rm
-rf */inst/apps/*, /bin/su - applvis -c rm -rf *FMW_Home*,
/bin/su - applvis -c rm -rf *fs*,
/bin/su - applvis -c find*exec rm -rf *,
/bin/su - applvis -c cp */inst/apps/*appl/admin* */inst/apps/*,
```

```
/bin/su - applvis -c */*EBSapps.env*perl */patch/115/bin/
txkSetAppsConf.pl*contextfile*configoption*oacore*oafm*forms*formsc4ws*,
/bin/su - applvis -c rsync -aH --delete --ignore-errors */EBSapps/ */EBSapps/*,
/bin/su - applvis -c rm */serviceStartfile.tmp*, /bin/su - applvis -c rm -rf */
change_apps_password*,
/bin/su - applvis -c mkdir -p */change_apps_password*, /bin/su - applvis -c */*.env*
run; cd */change_apps_password*/fnd/12.0.0/bin/FNDCPASS*apps*system*SYSTEM APPLSYS*,
/bin/su - applvis -c */*.env* sqlplus *apps*,
/bin/su - applvis -c */*.env* rm -f */updateDSpwd.py*cat */updateDSpwd.py*,
/bin/su - applvis -c */*.env*/adautocfg.sh*,
/bin/su - applvis -c */*.env* rm -f */serverStateAll.py*cat*serverStateAll.py*,
/bin/su - applvis -c */*.env* run; */wlserver_10.3/common/bin/wlst.sh */
updateDSpwd.py*,
/bin/su - applvis -c */*.env*; */wlserver_10.3/common/bin/wlst.sh */
serverStateAll.py*,
/bin/su - applvis -c *lsof*, /bin/su - applvis -c *sed *,
/bin/su - applvis -c */*.env*;*adapcctl.sh status*,
/bin/mount, /bin/umount, /bin/ps, /bin/su - applvis -c find*,
/bin/su - applvis -c cp *dlpx_force_autoflush.pm*,
/bin/su - applvis -c *perl -mdlpx_force_autoflush ./adcfgclone.pl AppsTier*,
/bin/su - applvis -c cat*serverStateAll.py*, /bin/su - applvis -c
cat*updateDSpwd.py*,
/bin/su - applvis -c mkdir -p */pairsdir*,
/bin/su - applvis -c export PERL5LIB* cd *perl -mdlpx_force_autoflush *adpreclone.pl
AppsTier*,
/bin/su - applvis -c mv *scratch_file* *,
/bin/su - applvis -c pmap -r*,
/bin/su - applvis -c */*.env* patch; */wlserver_10.3/common/bin/wlst.sh */
updateDSpwd.py*
```

Entries required for provisioning via low privileged user (delphix_os) on **source** EBS DB Tier.

```
Default:delphix_os !requiretty
delphix_os ALL=NOPASSWD: /bin/su - oravis -c echo *,
/bin/su - oravis -c rm -f */dlpx_force_autoflush*,
/bin/su - oravis -c cp -f */dlpx_force_autoflush*; chmod 755 */dlpx_force_autoflush*,
/bin/su - oravis -c rm -rf */appsutil/clone/dbts*,
/bin/su - oravis -c export PERL5LIB*perl -mdlpx_force_autoflush */adpreclone.pl
dbTier*,
/bin/su - oravis -c */rsync*, /bin/su - oravis -c test*mkdir*,
/bin/su - oravis -c test*touch*chmod 750*cat*,
/bin/su - oravis -c */EBS_kill/kill_script.sh*,
/bin/su - oravis -c rm -f */test_status.tmp*,
/bin/ps, /bin/su - oravis -c cp *dlpx_force_autoflush.pm*,
/bin/su - oravis -c chmod 755 */dlpx_force_autoflush*
```

Entries required for provisioning via low privileged user (delphix_os) on **source** EBS AppsTier.

```
Default:delphix_os !requiretty
delphix_os ALL=NOPASSWD: /bin/su - applvis -c echo *,
```

```
/bin/su - applvis -c rm *.dlpx_run_edition*,
/bin/su - applvis -c rm -f */dlpx_force_autoflush*,
/bin/su - applvis -c cd *echo *dlpx_force_autoflush*,
/bin/su - applvis -c export PATH* export PERL5LIB* cd *perl -mdlpx_force_autoflush ./
adpreclone.pl AppsTier*,
/bin/su - applvis -c */rsync*, /bin/su - applvis -c test*mkdir*,
/bin/su - applvis -c test*touch*chmod 750*cat*,
/bin/su - applvis -c */EBS_kill/kill_script.sh*,
/bin/su - applvis -c rm -f */test_status.tmp*,
/bin/su - applvis -c *.env* sqlplus -s apps*,
/bin/su applvis -c *.env* sqlplus -s apps*,
/bin/su - applvis -c */*.env*echo*adadminsrvctl.sh status -nopromptmsg*,
/bin/ps, /bin/su - applvis -c cp *dlpx_force_autoflush.pm*,
/bin/su - applvis -c export PERL5LIB* cd *perl -mdlpx_force_autoflush */adpreclone.pl
AppsTier*
```

9.8.3.5.5 Sudoers entry for Solaris

Entries required for provisioning via low privileged user (delphix_os) on **source** EBS DB Tier.

```
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD: /usr/bin/su - oravis -c echo *, /usr/bin/su
- oravis -c rm -f */dlpx_force_autoflush*, /usr/bin/su
- oravis -c cp -f */dlpx_force_autoflush*; chmod 755 */dlpx_force_autoflush*, /usr/
bin/su
- oravis -c rm -rf */appsutil/clone/dbts*, /usr/bin/su
- oravis -c export PERL5LIB*perl -mdlpx_force_autoflush */adpreclone.pl dbTier*, /
usr/bin/su
- oravis -c */rsync*, /usr/bin/su - oravis -c test*mkdir*, /usr/bin/su
- oravis -c test*touch*chmod 750*cat*, /usr/bin/su
- oravis -c */EBS_kill/kill_script.sh*, /usr/bin/su
- oravis -c rm -f */test_status.tmp*, /bin/ps, /usr/bin/su
- oravis -c cp *dlpx_force_autoflush.pm*, /usr/bin/su
- oravis -c chmod 755 */dlpx_force_autoflush*
```

Entries required for provisioning via low privileged user (delphix_os) on **source** EBS AppsTier.

```
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD: /usr/bin/su
- oravis -c echo *, /usr/bin/su
- oravis -c rm *.dlpx_run_edition*, /usr/bin/su
- oravis -c rm -f */dlpx_force_autoflush*, /usr/bin/su
- oravis -c cd *echo *dlpx_force_autoflush*, /usr/bin/su
- oravis -c export PATH* export PERL5LIB* cd *perl -mdlpx_force_autoflush ./
adpreclone.pl AppsTier*, /usr/bin/su
- oravis -c */rsync*, /usr/bin/su
- oravis -c test*mkdir*, /usr/bin/su
- oravis -c test*touch*chmod 750*cat*, /usr/bin/su
- oravis -c */EBS_kill/kill_script.sh*, /usr/bin/su
```

```
- oravis -c rm -f */test_status.tmp*, /usr/bin/su
- oravis -c *.env* sqlplus -s apps*, /usr/bin/su oravis -c *.env* sqlplus -s apps*, /
usr/bin/su
- oravis -c */*.env*echo*adadminsrvctl.sh status -nopromptmsg*, /bin/ps, /usr/bin/su
- oravis -c cp *dlpx_force_autoflush.pm*, /usr/bin/su
- oravis -c export PERL5LIB* cd *perl -mdlpx_force_autoflush */adpreclone.pl
AppsTier*
```

Entries required for provisioning via low privileged user (delphix_os) on **target** EBS DB Tier.

```
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD: /usr/bin/su
- oravis -c /usr/sbin/mount, /usr/bin/su
- oravis -c /usr/sbin/umount, /usr/bin/su
- oravis -c echo *, /usr/bin/su
- oravis -c export *, /usr/bin/su
- oravis -c test*mkdir*, /usr/bin/su
- oravis -c test*touch*chmod 750*cat*, /usr/bin/su
- oravis -c */EBS_kill/kill_script.sh*, /usr/bin/su
- oravis -c rm -f */test_status.tmp*, /usr/bin/su
- oravis -c cp -f */dlpx_force_autoflush*; chmod 755 */dlpx_force_autoflush*, /usr/
bin/su
- oravis -c rm -f */dlpx_force_autoflush*, /usr/bin/su
- oravis -c */*.env*bin/lsnrctl status*, /usr/bin/su
- oravis -c */*.env*/addlnctl.sh*, /usr/bin/su
- oravis -c rm -f */.delphix_adpreclone.lck*, /usr/bin/su
- oravis -c rm -rf */appsutil/clone/dbts*, /usr/bin/su
- oravis -c touch */.delphix_adpreclone.lck*, /usr/bin/su
- oravis -c */bin/runInstaller -silent -detachHome*, /usr/bin/su
- oravis -c export PATH* export PERL5LIB*perl -mdlpx_force_autoflush */adcfgclone.pl
dbtechstack*, /usr/bin/su
- oravis -c export PERL5LIB*perl -mdlpx_force_autoflush */adpreclone.pl
dbtechstack*, /usr/bin/su
- oravis -c cd *; make -f ins_rdbms.mk*ioracle*, /usr/bin/su
- oravis -c cd *; make -f ins_rdbms.mk dnfs_off*, /usr/bin/su
- oravis -c rm -f *bak, /usr/bin/su - oravis -c mv *, /usr/bin/su
- oravis -c */*.env*sqlplus* as sysdba*, /usr/bin/su
- oravis -c */*.env*perl */appsutil/clone/bin/adcfgclone.pl dbconfig*, /usr/bin/su
- oravis -c */*.env*perl */appsutil/scripts/*/adpreclone.pl database*, /usr/bin/su
- oravis -c sed*sqlnet.ora*, /usr/bin/su - oravis -c */*.env*sqlplus apps*, /usr/bin/
su
- oravis -c */*.env*; make -f *rdbms/lib/ins_rdbms.mk dnfs_off*, /usr/bin/su
- oravis -c */*.env*; ln -s *, /usr/sbin/mount, /usr/sbin/umount, /usr/bin/ps, /usr/
bin/mkdir, /usr/bin/su
- oravis -c source* -dboraclehome* perl*txkGenCDBTnsAdmin.pl*, /usr/bin/su
- oravis -c */*.env*perl */appsutil/bin/txkCfgUtlfileDir.pl -contextfile*, /usr/bin/
su
- oravis -c */*.env* mkdir -p*, /usr/bin/su
- oravis -c chmod 775 *dbs*, /usr/bin/su
- oravis -c chmod 6751 */bin/oracle*, /usr/bin/su
- oravis -c cp *dlpx_force_autoflush.pm*, /usr/bin/su
- oravis -c chmod 755 */dlpx_force_autoflush*, /usr/bin/su
```

```
- oravis -c umask*touch *source_apps_file.txt, /usr/bin/su
- oravis -c mkdir -p*, /usr/bin/su
- oravis -c cp *pairsfile*, /usr/bin/su
- oravis -c *perl -mdlpx_force_autoflush */adclonectx.pl*, /usr/bin/su
- oravis -c *perl -mdlpx_force_autoflush */adcfgclone.pl dbtechstack*, /usr/bin/su
- oravis -c touch */.delphix_adpreclone.lck*, /usr/bin/su
- oravis -c rm -f */.delphix_adpreclone.lck*, /usr/bin/su
- oravis -c chmod 755 *hooksUtil*,  /usr/bin/su
- oravis -c set -o *hooksUtil*, /usr/bin/su
- oravis -c mkdir *hooksUtil*, /usr/bin/mkdir, /usr/bin/rmdir, /usr/sbin/mount, /usr/
sbin/umount, /usr/bin/pargs, /usr/bin/ps, /usr/bin/netstat
```

Entries required for provisioning via low privileged user (delphix_os) on **target** EBS AppsTier.

```
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD: /usr/bin/su
- oravis -c echo *, /usr/bin/su
- oravis -c rm *.dlpx_run_edition*, /usr/bin/su
- oravis -c rm -f *dlpx_force_autoflush*, /usr/bin/su
- oravis -c cd *echo *dlpx_force_autoflush*, /usr/bin/su
- oravis -c export PATH* export PERL5LIB* cd *perl -mdlpx_force_autoflush ./
adpreclone.pl AppsTier*, /usr/bin/su
- oravis -c */rsync*, /usr/bin/su - oravis -c test*mkdir*, /usr/bin/su
- oravis -c test*touch*chmod 750*cat*, /usr/bin/su
- oravis -c */EBS_kill/kill_script.sh*, /usr/bin/su
- oravis -c rm -f */test_status.tmp*, /usr/bin/su
- oravis -c */*.env* sqlplus -s *apps*, /usr/bin/su
- oravis -c */adstrtal.sh*, /usr/bin/su
- oravis -c */adstpall.sh*, /usr/bin/su
- oravis -c */*.env* cd *adautocfg.sh*, /usr/bin/su
- oravis -c export PATH* export PERL5LIB* cd *perl -mdlpx_force_autoflush ./
adcfgclone.pl AppsTier*, /usr/bin/su
- oravis -c *perl -mdlpx_force_autoflush */adclonectx.pl addnode
contextfile*pairsfile*outfile*, /usr/bin/su
- oravis -c *adadminsrvctl.sh*, /usr/bin/su
- oravis -c *admanagedsrvctl.sh*, /usr/bin/su
- oravis -c *adnodemgrctl.sh*, /usr/bin/su
- oravis -c */*.env* cd *perl -mdlpx_force_autoflush ./
txkUpdateEBSDomain.pl*contextfile*action*updateAdminPassword*, /usr/bin/su
- oravis -c */bin/runInstaller -silent -detachHome*, /usr/bin/su
- oravis -c rm -rf */inst/apps/*, /usr/bin/su
- oravis -c rm -rf *FMW_Home*, /usr/bin/su
- oravis -c rm -rf *fs*, /usr/bin/su
- oravis -c find*exec rm -rf *, /usr/bin/su
- oravis -c cp */inst/apps/*appl/admin* */inst/apps/*, /usr/bin/su
- oravis -c */*EBSapps.env*perl */patch/115/bin/
txkSetAppsConf.pl*contextfile*configoption*oacore*oafm*forms*formsc4ws*, /usr/bin/su
- oravis -c rsync -aH --delete --ignore-errors */EBSapps/ */EBSapps/*, /usr/bin/su
- oravis -c rm */serviceStartfile.tmp*, /usr/bin/su
- oravis -c rm -rf */change_apps_password*, /usr/bin/su
- oravis -c mkdir -p */change_apps_password*, /usr/bin/su
```

```
- oravis -c */*.env* run; cd */change_apps_password*/fnd/12.0.0/bin/
FNDCPASS*apps*system*SYSTEM APPLSYS*, /usr/bin/su
- oravis -c */*.env* sqlplus *apps*, /usr/bin/su - oravis -c */*.env* rm -f */
updateDSpwd.py*cat */updateDSpwd.py*, /usr/bin/su
- oravis -c */*.env*/adautocfg.sh*, /usr/bin/su - oravis -c */*.env* rm -f */
serverStateAll.py*cat*serverStateAll.py*, /usr/bin/su
- oravis -c */*.env* run; */wlserver_10.3/common/bin/wlst.sh */updateDSpwd.py*, /usr/
bin/su
- oravis -c */*.env*; */wlserver_10.3/common/bin/wlst.sh */serverStateAll.py*, /usr/
bin/su
- oravis -c *lsof*, /usr/bin/su - oravis -c *sed *, /usr/bin/su
- oravis -c */*.env*;*adapcctl.sh status*, /usr/sbin/mount, /usr/sbin/umount, /usr/
bin/ps, /usr/bin/su
- oravis -c find*, /usr/bin/su - oravis -c cp *dlpx_force_autoflush.pm*, /usr/bin/su
- oravis -c *perl -mdlpx_force_autoflush ./adcfgclone.pl AppsTier*, /usr/bin/su
- oravis -c cat*serverStateAll.py*, /usr/bin/su
- oravis -c cat*updateDSpwd.py*, /usr/bin/su
- oravis -c mkdir -p */pairsdir*, /usr/bin/su
- oravis -c export PERL5LIB* cd *perl -mdlpx_force_autoflush *adpreclone.pl
AppsTier*, /usr/bin/su
- oravis -c mv *scratch_file* *, /usr/bin/su - oravis -c pmap -r*, /usr/bin/su
- oravis -c */*.env* patch; */wlserver_10.3/common/bin/wlst.sh */updateDSpwd.py*, /
usr/bin/su
- oravis -c */*.env* run; */adcmctl.sh status*, /usr/bin/su
- oravis -c set -o pipefail; cat*tee */.*_pairs.txt*, /usr/bin/su
- oravis -c chmod 755  */.*_pairs.txt*, /usr/bin/su
- oravis -c *perl -mdlpx_force_autoflush */adcfgclone.pl*AppsTier*, /usr/bin/su
- oravis -c /usr/bin/netstat -an *, /usr/bin/mkdir, /usr/bin/rmdir, /usr/sbin/
mount, /usr/sbin/umount, /usr/bin/pargs, /usr/bin/ps, /usr/bin/netstat
```

9.8.4 Host Checker for external connectors

9.8.4.1 Overview

The Host Checker for External Connectors utility performs various checks to ensure Delphix's Oracle EBS Source and Target environments (hosts) are configured correctly. We recommend using this utility to validate a fresh environment or following an environment's database upgrade.

- The scripts provided as part of this automation package are provided as-is and are intended to help you easily check the feasibility of linking and provisioning operations.
- Although the scripts have been tested by Delphix Continuous Data Engineering, they are not officially supported by Delphix. This means that Delphix is not liable for any bugs or defects in this implementation.
- You can create or add your own scripts to support implementing the Host Checker for External Connectors operations using the EBS utilities of your choice.

9.8.4.2 Structure

The Host Checker for External Connectors utility is structured as follows:

- The utility is designed to be easily extensible, with new checks being added as required.
- The utility is designed to be easily customizable, with parameters that can be changed to suit specific requirements.

9.8.4.3 Workflows

The Host Checker for External Connectors utility provides four specific environment workflows:

- AppsTier - Source
- AppsTier - Target
- DbTier - Source
- DBTier - Target

9.8.4.4 Prerequisites

- Linux or Solaris environment

Exclusions For Solaris OS

1. For Solaris, Delphix will not check the sudoers entries. The utility won't ask if you are running privilege elevation.
   If you are running privilege elevation on Solaris, see the Host requirements section on Requirements for Oracle source databases for linking (see page 1664) for required sudoers entries.
2. Kernel `shmmax` and `shmall` values are not being checked for Solaris, see /etc/hosts section on Requirements for Oracle EBS target databases for provisioning (see page 1667) for required values.

9.8.4.5 Instructions

Follow the steps below to run the Host Checker for External Connectors utility. It runs in interactive mode through the command terminal.

1. Download the Host Checker for External Connectors utility from the Delphix Download Portal[482].

482 https://download.delphix.com/folder/5800/Delphix%20Product%20Releases/Continuous%20Data%20Connectors/Host%20Checker%20for%20External%20Connectors

   a. Find the Host Checker for External Connectors utility under the **Utilities** section in the **Delphix Product Releases > Continuous Data Connectors** download page.

2. Manually copy it to the Source or Target environment you intend to validate.
3. Extract the zip file and view its contents.
4. Grant the `main.sh` file execution permissions using the command: `chmod +x main.sh`
5. Run the Host Checker utility using the command: `./main.sh`
6. Follow the terminal prompts to validate your chosen environment.

The following is a sample runtime log from an AppsTier Target Environment:

```
[applvis@e27a171at-app ebs-hostchecker]$ ./main.sh
Select the host type:
1. AppsTier
2. dbtier
Enter your choice (1 or 2): 1
Select the AppsTier host type:
1. source
2. target
Enter your choice (1 or 2): 2
Executing checks for AppsTier target host..
Enter the toolkit directory path: /var/tmp
Enter AppsTier target hostname(fqdn): e27a171at-app.dlpxdc.co
Enter dbtier target hostname(fqdn): e27a171at-db.dlpxdc.co
Do you want to use privilege elevation? [y/n]: y
Enter the 'high' privilege user used for privilege elevation in sudoers entries:
applvis
Enter the 'low' privilege user used for privilege elevation in sudoers entries:
delphix_os
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++
Current CPU utilization: 2 percent
CPU utilization is below threshold i.e. 80.
###Printing server results###
Server_Configuration | Current_Value | Required_Value
--------------------------------------------------------------------------------
-----------------
Architecture | x86_64 Linux |
Kernel version | 4.18.0-240.el8.x86_64 |
Total CPU cores on server | 8 cores | Minimum 2 cores
Total memory on server | 32149 MB | Minimum 8GB
Free memory on server | 31805 MB | Minimum 8GB
kernel.shmmax value set on server | 31814338025095168 bytes | 30340517068 bytes
kernel.shmall value set on server | 8291456 Pages | 8230392 pages
--------------------------------------------------------------------------------
-----------------
For EBS sizing Refer- https://docs.oracle.com/cd/E26401_01/doc.122/e22950/
T422699i4773.htm#ebs_sizing
--------------------------------------------------------------------------------
-----------------
```

```
Memory utilization OK.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++
Checking for expect
found expect utility at - /usr/bin/expect
Checking for ss
found ss utility at - /usr/sbin/ss
Checking for netstat
found netstat at - /usr/bin/netstat
Checking for make
found make at - /usr/bin/make
Checking for ld
found ld at - /usr/bin/ld
Checking for gcc
found gcc at - /usr/bin/gcc
Checking for g++
found g++ at - /usr/bin/g++
Checking for ar
found ar at - /usr/bin/ar
Utility check completed successfully
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++
Available disk space in toolkit directory: 511817 MB
There is enough disk space in the toolkit directory.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++
Ping to e27a171at-app.dlpxdc.co succeeded. IP address: 10.110.237.75
Entry for e27a171at-app.dlpxdc.co with IP address 10.110.237.75 exists in /etc/hosts.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++
Ping to e27a171at-db.dlpxdc.co succeeded. IP address: 10.110.241.161
Entry for e27a171at-db.dlpxdc.co with IP address 10.110.241.161 exists in /etc/hosts.
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++
Checking sudoers file entries. Using : AppsTier_tgt_sudo.sh
Sudoers entry check completed successfully!
Following entries not found in /etc/sudoers:
/bin/su - applvis -c rm -f *updateDSpwd.py,
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++
Central Inventory location is /u01/oracle/VIS12210/oraInventory
Make sure Oracle installation owners you intend to use have the Oracle Inventory
group oinstall as their primary group
Printing members of oinstall group, group id 9999 using /etc/group file
applvis
delphix_os
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++
### Beginning ulimit check ###
### Oracle Software Installation Owner Resource Limit Recommended Ranges for Standard
Installations ###
--------------------------------------------------------------------------------------
----------------------------------------
```

```
Resource Shell Limit | Resource | Soft Limit | Hard Limit
--------------------------------------------------------------------------------
----------------------------------------
Open file descriptors | nofile | at least 1024 | at least 65536
Number of processes available to a single user | nproc | at least 2047 | at least
16384
Size of the stack segment of the process | stack | at least 10240 KB| at least 10240
 KB, and at most 32768 KB
Maximum locked memory limit | memlock | unlimited | unlimited
--------------------------------------------------------------------------------
----------------------------------------
For EBS sizing, refer to:
Checking Resource Limits for Oracle Software Installation Users
https://docs.oracle.com/en/database/oracle/oracle-database/21/ladbi/checking-
resource-limits-for-oracle-software-installation-users.html#GUID-293874BD-8069-470F-
BEBF-
A77C06618D5A:~:text=Table%205%2D1%20Oracle%20Software%20Installation%20Owner%20Resour
ce%20Limit%20Recommended%20Ranges%20for%20Standard%20Installations
--------------------------------------------------------------------------------
----------------------------------------
Current system settings:
--------------------------------------------------------------------------------
----------------------------------------
Resource Shell Limit | Resource | Soft Limit | Hard Limit
--------------------------------------------------------------------------------
----------------------------------------
Open file descriptors | nofile | 4096 | 65536
Number of processes available to a single user | nproc | 2047 | 16384
Size of the stack segment of the process | stack | 10240 | 16384
--------------------------------------------------------------------------------
----------------------------------------
Other ulimit settings |
-----------------------------------------------------------------
Resource Shell | Limit |
-----------------------------------------------------------------
max locked memory | 64 in KB |
max memory size | unlimited in KB|
virtual memory | unlimited in KB|
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++
--------------------------------------------------------------------------------
----------------------------------------
Health checker warnings!
--------------------------------------------------------------------------------
----------------------------------------
Following entries not found in /etc/sudoers:
/bin/su - applvis -c rm -f *updateDSpwd.py,
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
-----------------------------------------------------------------
```

9.8.5 Oracle EBS installation and upgrade

The Oracle EBS connector is managed through the Delphix Continuous Data Engine's connector framework. Connectors are distributed separately from the engine. The connector must be installed before Oracle EBS Environments can be added, and before attempting any Oracle EBS virtualization operations.

9.8.5.1 Downloading the connector

Perform the following steps to download the Oracle EBS connector to your Desktop system.

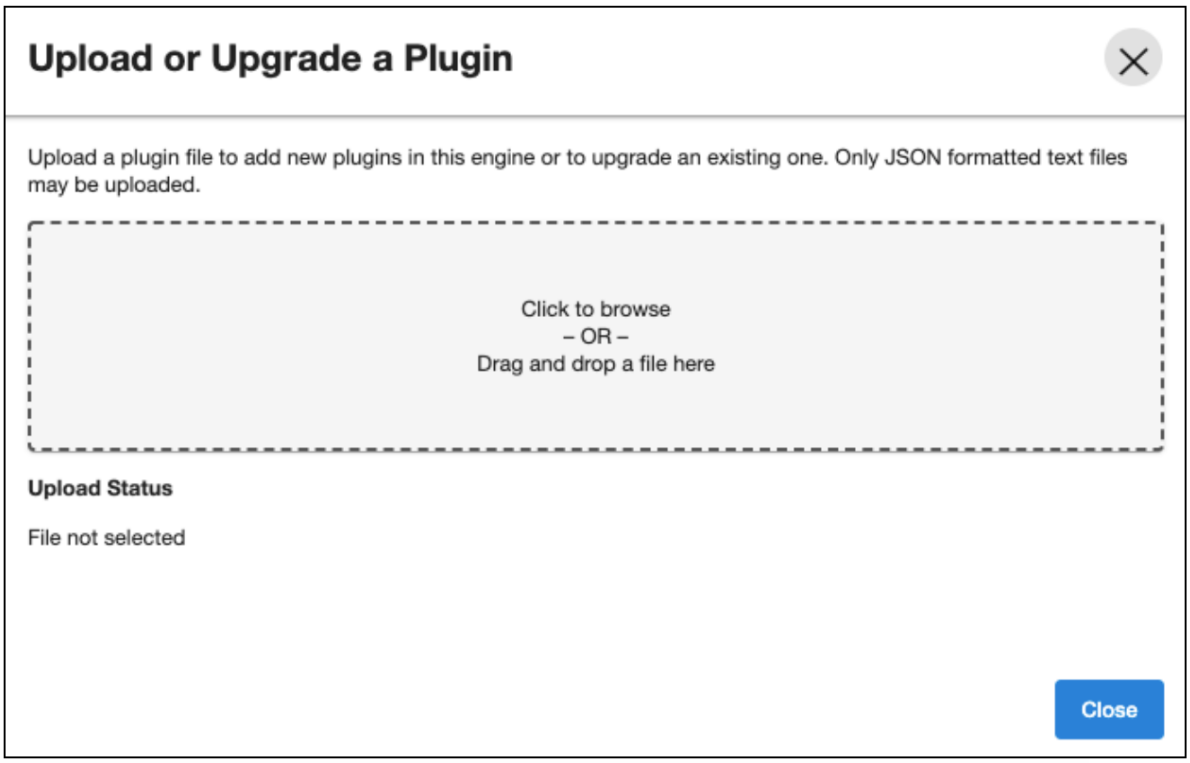
1. Log in to the Delphix Download Portal[483] using your credentials from the Delphix Support Portal.
2. Find the Oracle EBS plugin folder by navigating to the **Delphix Product Releases >Continuous Data Connectors > Oracle E-Business Suite (EBS) > EBS 12.2** path and select the desired version's folder.
3. Download the .zip plugin file by clicking `EBS_Plugin_<version>_for_EBS12.2.zip`.
4. After the download is complete, extract the contents of the downloaded .zip file to obtain the JSON files. You must find two files named as `ebs122app-<version>.json` and `ebs122db-<version>.json`

9.8.5.2 Installing the connector

Perform the following steps to upload and install a new connector.

1. Log in to the Delphix Management application.
2. Navigate to the Manage > Plugins page.
3. To upload a plugin, click the ⊞ icon. This action displays a "Upload or Upgrade a Plugin" dialog window.
4. Browse to the JSON files on your desktop which you downloaded and extracted in the previous step
5. Upload the downloaded JSON (for example, ebs122db-<version>.json) files one by one. Once the first upload is successful, close the dialog box and then start a new upload for the second file.

483 https://download.delphix.com/



For more information, refer to the Delphix Continuous Data Engine connector management[484] section.

9.8.5.3 Upgrading the connector

Users must upgrade their connectors to use new features and apply bug fixes.

Connectors are not upgraded through engine upgrades. However, some connector upgrades are only compatible with a minimum Delphix Continuous Data Engine version. Please ensure your engine is upgraded to a compatible version before upgrading the connector. For full compatibility details, refer to the Oracle EBS support matrix[485].

Perform the following process to upgrade your Oracle EBS connector to a later version.

> ⚠ Warning! Since v4.0.2, connector upgrades have been streamlined. However, skipping a version upgrade step before upgrading to v4.0.2 may lead to configuration issues. If upgrading from a pre-v4.0.2, follow the Upgrade matrix (see page 0) instructions closely.

484 https://cd.delphix.com/docs/latest/delphix-engine-plugin-management
485 https://cd.delphix.com/docs/latest/oracle-e-business-suite-ebs-matrix

9.8.5.3.1 Upgrade process

Perform the following steps when upgrading your Oracle EBS connector.

1. Disable all Oracle EBS dSources and VDBs.(Only required for upgrades of EBS connector version lower than v4.0.2)

> **Note**: Consult the Upgrade matrix (see page 0) instructions for more details.

2. Upload the newer connector version into the Delphix Continuous Data Engine. For more information, refer to the Installing the Oracle EBS connector[486] documentation.
3. Refresh all the associated Oracle EBS environments to trigger the discovery workflow. Verify that the workflow is completed.
4. Enable dSources and VDBs, if disabled, and create new dSource and VDB snapshots.
5. Review the newly created snapshot's metadata for the new plugin version to confirm the upgrade was successful.

9.8.5.3.2 Upgrade matrix

This section lists Oracle EBS connector versions that a user can upgrade from and to.

> Warning! You must adhere to the following upgrade considerations. Ignoring the below upgrade matrix may result in incompatible Oracle EBS data sources.
> - It is mandatory to upgrade to the v4.0.2 version of the Oracle EBS connector and then upgrade to only v4.1.0. of the Oracle EBS connector.
> - After upgrading to v4.1.0, you may upgrade directly to any higher version.

Follow the matrix below to avoid connector corruption:

Your installed version	Required upgrade path	dSource and VDB Downtime Required?

486 https://cd.delphix.com/docs/latest/oracle-ebs-installation-and-upgrade

v1.8.2 v1.8.3 v1.8.4	Upgrade to v2.0.0 or v2.0.1	Yes
v2.0.0 v2.0.1	Upgrade to v3.0.0, 3.0.1, v3.0.2, or v3.0.3	Yes
v3.0.0 v3.0.1 v3.0.2 v3.0.3	Upgrade directly to v4.0.2.	Yes
v4.0.2	Upgrade directly to v4.1.0.	
v4.1.0+ (vSDK)	Upgrade to any more recently released version.	No

9.8.6 Oracle EBS connector operations

To successfully ingest and virtualize a EBS instance, you must perform the following connector operations in sequence:

- Oracle EBS discovery operations (see page 1690)
- Oracle EBS linking data sources (see page 1693)
- Provisioning Oracle EBS VDBs (see page 1700)
- Refreshing and rewinding a TDE-enabled vPDB for Oracle EBS (see page 1745)
- Starting and stopping a virtual EBS instance (see page 1745)
- Rewinding a virtual EBS instance (see page 1746)
- Refreshing a virtual EBS instance (see page 1748)
- Enabling and disabling a virtual EBS instance (see page 1749)
- Deleting a virtual EBS instance (see page 1751)
- Modifying the AppsTier topology (see page 1752)
- Replicating a virtual EBS instance (see page 1752)
- Upgrading a Delphix Engine hosting a virtual EBS instance (see page 1753)
- Migrating an EBS VDB (see page 1753)

The respective sections provide detailed information about prerequisites, limitations (if any), and the procedure to be followed for completing each of the operations.

9.8.6.1 Oracle EBS discovery operations

9.8.6.1.1 Adding an environment

This topic describes the process of adding a Oracle EBS source and target environment.

9.8.6.1.1.1 Prerequisites

Ensure all the essential requirements, as described in Requirements and Prerequisites are met.

9.8.6.1.1.2 Procedure

Perform the following steps to add an environment to the Delphix Continuous Data Engine:

1. Login to the **Delphix Management** application.
2. Navigate to the **Manage > Environments** page.
3. Click the **Actions** ··· menu and select **Add Environment**.
4. From the **Add Environment** dialog window, select **Unix/Linux.**
5. Select **Standalone Host**.
6. Click **Next**.
7. Provide the name of the environment.
8. Provide the **Host IP address** or **host FQDN**.
9. Provide the **SSH port**. The default value is 22.
10. Provide an **Environment Username** for the environment. If you intend to use a low-privileged environment user, ensure that you have fulfilled the Privilege elevation requirements for Oracle EBS (see page 1673).
11. Choose the **Login Type** that fits your setup.

- Username and Password – provide the environment username and password
- Username and Public Key – provide the environment username
- Password Vault – select from an existing Enterprise Password Vault

Note: Using Public Key Authentication
If you want to use public-key authentication for logging into your Unix-based environment, you have two options:

- Use the Delphix Continuous Data Engine's key pair. For more information on how to use the Delphix Continuous Data Engine's key pair, refer to the How to use the Continuous Data Engine's key-pair (see page 1767) section.

- Provide a key pair for this environment. As an alternative, a key pair specific for this environment can be provided via the API or CLI. For instructions, refer to CLI Cookbook: Setting up SSH key authentication for UNIX environment users[487].

12. For Password, enter the password associated with the user in step 11.
13. If you want to use Public Key Encryption for logging into your environment:
    a. Select Public Key for the Login Type
    b. Click View Public Key
    c. Copy the public key that is displayed, and append it to the end of your ~/.ssh/authorized_keys file. If this file does not exist, it must be created.

> **Note:** The addition of the public key is a one-time requirement per user and per environment. Additionally, you have the option to add public key authentication to an environment user's profile via CLI (for details, refer to the CLI Cookbook: Setting up SSH key authentication for UNIX environment users[488] topic).

14. For Password Login, click Verify Credentials to test the username and password.
15. Enter **Toolkit Path** (ensure that the toolkit path does not have any spaces).
16. To provide your own Oracle Java Development Kit, select the **Provide my own JDK** checkbox and click **Next**.
    a. On the Java Development Kit page, provide an absolute path to your Oracle JDK and click **Next**.
17. Click **Submit**.

After successfully adding and discovering the environment, you should be able to view the newly added environment listed under the Environments tab. In case it does not appear, click the refresh icon in your browser.

After the environment has been successfully discovered, if you wish to establish further links, add a source configuration to the previously discovered installation.

Perform the following steps to access the information about any environment you have created:

1. Navigate to **Manage** > **Environments**.
2. Select the environment name.

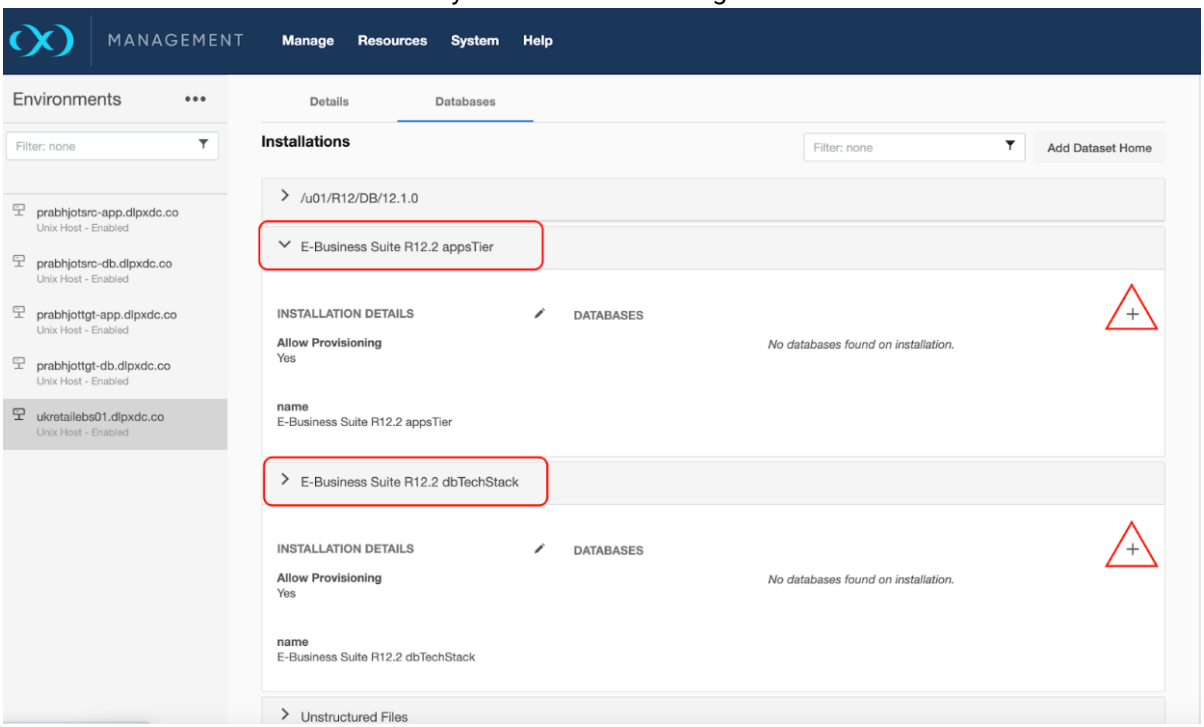
487 https://cd.delphix.com/docs/latest/cli-cookbook-setting-up-ssh-key-authentication-for
488 https://cd.delphix.com/docs/latest/cli-cookbook-setting-up-ssh-key-authentication-for

> **Note:** Discovery of the Oracle database is handled by Continuous Data Engine. To add the EBS DBTechstack and AppsTier installations, you must add “Source Config” for each under the respective installations.

9.8.6.1.2 Adding Source Config

Details of how to add source config for DBTechstack and AppsTier

1. In Delphix Continuous Data Engine, navigate to **Manage** > **Environments**.
2. Select the environment name in which you want to add source config.
3. Click the **Environment Details** tab.
4. If the oracle environment user described in [Requirements for Oracle EBS source databases for linking](#) (see page 1664) is not already added to the Delphix Continuous Data Engine, add the user.
5. Click the **Databases** tab.
6. Click the **Plus** icon within the E-Business **Suite R12.2 DBTechstack** section under installations.
7. **Add Database** dialog box pops to add source config for DBTechstack.
8. Provide a user-defined name to the `` `Database Name` `` field and Oracle base install directory in the `Path` field.
9. Click on the **Add** button to successfully add the source config.



9.8.6.2 Oracle EBS linking data sources

This topic outlines the procedure for linking an EBS R12.2 instance to the Delphix Continuous Data Engine.

Linking an EBS instance is a three step process. All of these steps can be done in parallel:

- Oracle EBS linking the DBTechstack (see page 1693)
- Linking Oracle EBS database (see page 1696)
- Linking the AppsTier (see page 1699)

9.8.6.2.1 Oracle EBS linking the DBTechstack

9.8.6.2.1.1 Prerequisites

Ensure your EBS R12.2 instance is supported. See Requirements for Source DBTechstack (see page 1664) to ensure you can link your EBS R12.2 instance to the Delphix Continuous Data Engine.

Ensure your EBS 12.2 environments comply with Oracle's documentation. Your environments must comply with Oracle's requirements for installing EBS. These requirements are outlined on **Oracle E-Business Suite Release 12 Installation Guidelines (Doc ID 405565.1)** found at https://support.oracle.com

9.8.6.2.1.2 Procedure

Upload the EBS 12.2 DBTechstack connector and refresh the environments during the preparation of the source environment.

Then follow these steps to link DBTechstack

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Environments**.
4. Select the **source dbTier environment** containing the source DBTechstack.

> **Note**: If you are linking from a RAC dbTier, select the environment for a single running node of the RAC cluster.

5. Click the **Environment Details** tab.
6. If the Oracle environment user described in Requirements for Oracle EBS source databases for linking (see page 1664) is not already added to the Delphix Continuous Data Engine, add the user.
7. Click the **Databases** tab.
8. Click the **Plus** icon within the E-Business **Suite R12.2 DBTechstack** section under installations.

9. **Add Database** dialog box pops to add source config for DBTechstack.
10. Provide a user-defined name to the `` `Database Name` `` field and Oracle base install directory in the `Path` field.
11. Click on the **Add** button to successfully add the source config.
12. Re-check the **E-Business Suite R12.2 DBTechstack** section under installations for the `Add dSource` option mapped to the source config that is added in the above step.
13. Click **Manage > Datasets.**
14. Click the **Plus** icon next to Datasets and select **Add dSource**.
15. In the **Add dSource wizard**, select the Linked dSource as the type.
16. Enter the **EBS-specific parameters** for your DBTechstack.
17. These parameter values will be used when adpreclone.pl is run. Ensure that the **DB Tier Context Name uses** the short hostname.
18. Select an Environment user.
19. **Privileged OS Account (Optional)** field should contain a high privileged user when the low privileged user is being used for linking.
20. Provide source oracle home directory name's relative path to oracle base installation directory in **Oracle Home** input field.
21. In the **DB Tier** Context Name field, provide the source context name; To get the DB Tier context name on source side:
    a. For Oracle 12.1.0.2 version,

    ```
    cd $ORACLE_HOME
    ```

    b. source SID_hostname.env file

    ```
    echo $CONTEXT_NAME
    ```

    c. Below is an example, Consider VISDB as the database SID and tavsrc-db as the short hostname.

    ```
    [oravis@tavsrc-db VIS]$ source /u01/oracle/VIS/12.1.0.2/VISDB_tavsrc-
    db.env
    [oravis@tavsrc-db VIS]$ echo $CONTEXT_NAME
    VISDB_tavsrc-db
    ```

    d. For multi-tenant database, CONTEXT_NAME will be (<PDB_NAME>_<short hostname>)
    For RAC, CONTEXT_NAME will be `<INSTANCE_SID>_<HOSTNAME>`
    For example: `VIS2_exaukdb02`

> **Note**: When preparing the source Oracle RAC system for cloning, make sure that the linking is performed against the primary database node.

22. Provide apps schema password in the **Apps Password Credentials** section by selecting the **Username** and **Password** from the dropdown, leaving the username field empty.
23. Exclude the EBS database's data files if they are stored underneath the Oracle base install directory.
24. These data files will be linked with the database instead of with the DBTechstack. Add the "relative path" to the data files to the **Paths to Exclude** list. If the audit files (located under $ORACLE_HOME/rdbms/audit) are large in size, the path of these files could also be excluded so as to avoid DBTechstack timeout issues during provisioning.

> **Note**: Ensure that you add the relative paths and NOT the absolute paths. For example, if you want to exclude the */u01/oracle/VIS/data* path, then add "*data*" only under the **Paths to Exclude** list. Here, */u01/oracle/VIS/data* is the absolute path and "*data*" is the relative path to be added.

25. Click **Next**.
26. Enter a **dSource Name**.
27. Select a **Database Group** for the dSource.
28. Click **Next**. Adding a dSource to a database group enables you to set Delphix Domain user permissions for that dSource's objects, such as snapshots.
29. Select a **SnapSync** policy.
30. Click **Next**.
31. Enter any **custom pre or post sync logic** as Pre-Sync or Post-Sync hook operations. Remember that `adpreclone.pl dbTechStack` is already run prior to every Snapshot of the DBTechstack. The Pre-Sync hook operations will be run prior to running the adpreclone.pl tool.
32. Click **Next**.

Review the **dSource Configuration** and **Data Management** information, and then click **Finish**. The Delphix Continuous Data Engine will initiate two jobs to create the dSource, **DB_Link** and **DB_Sync** . You can monitor these jobs by clicking **Active Jobs** in the top menu bar, or by selecting **System** > **Event Viewer**. When the jobs have completed successfully, the **files** icon will change to a **dSource** icon on the **Environments** > **Databases** screen, and the dSource will be added to the list of **Datasets** under its assigned group.

9.8.6.2.2 Linking Oracle EBS database

9.8.6.2.2.1 Prerequisites

The Oracle EBS Database must meet the source requirements outlined in Oracle support and requirements. These requirements are generic to all Unix environments containing an Oracle database to be linked.

9.8.6.2.2.2 Procedure

Link the Oracle database used by EBS, as outlined in Linking an Oracle data source (see page 1664)

Link the Oracle Pluggable database for 19c MT, as outlined in Provisioning the Oracle pluggable database (see page 1706)

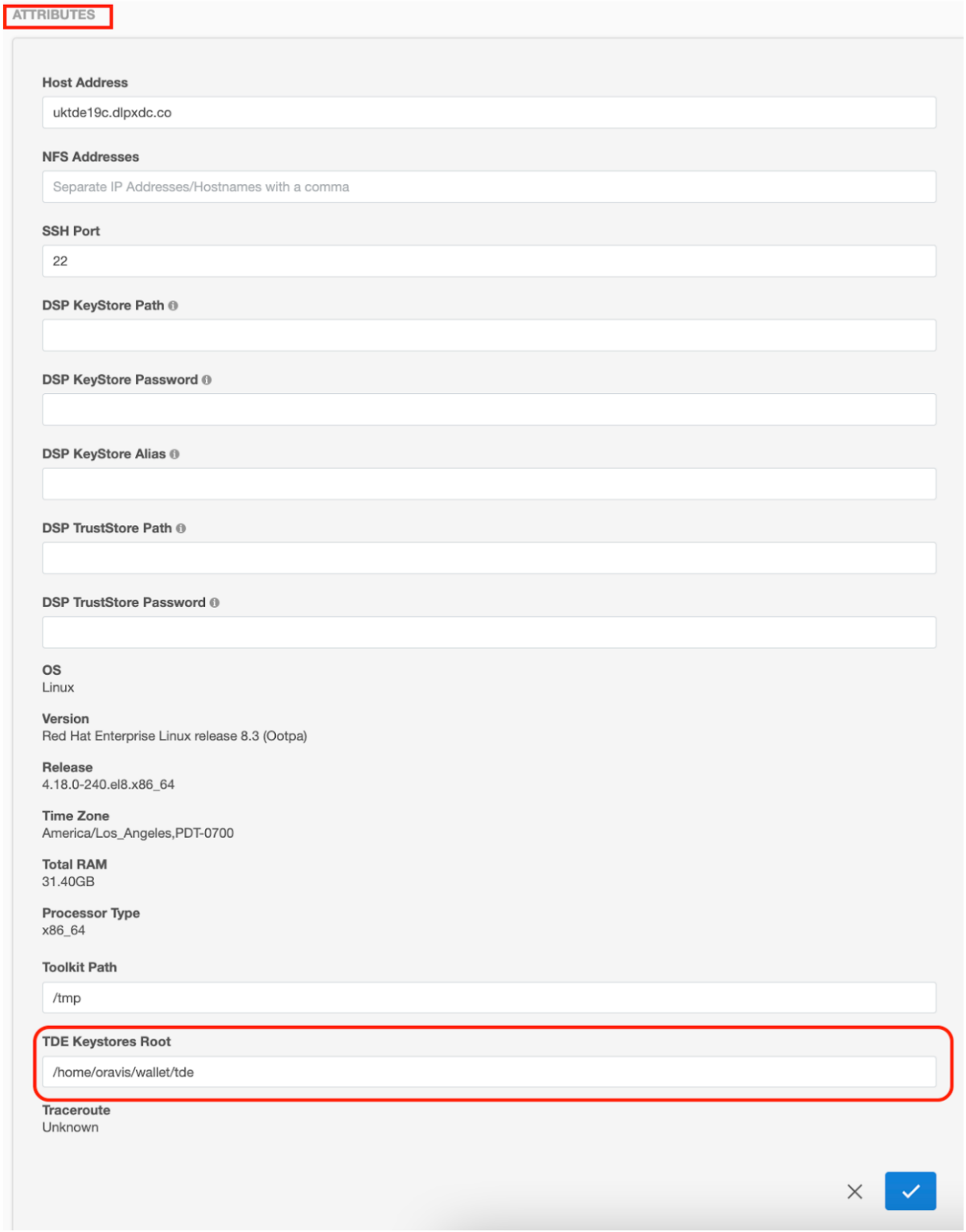
9.8.6.2.2.3 Linking of source TDE enabled PDB

Prerequisites

Adding or editing the TDE keystores root directory path

The TDE Keystores root directory path is specified on the Details tab under Environments. To edit the path, do the following:

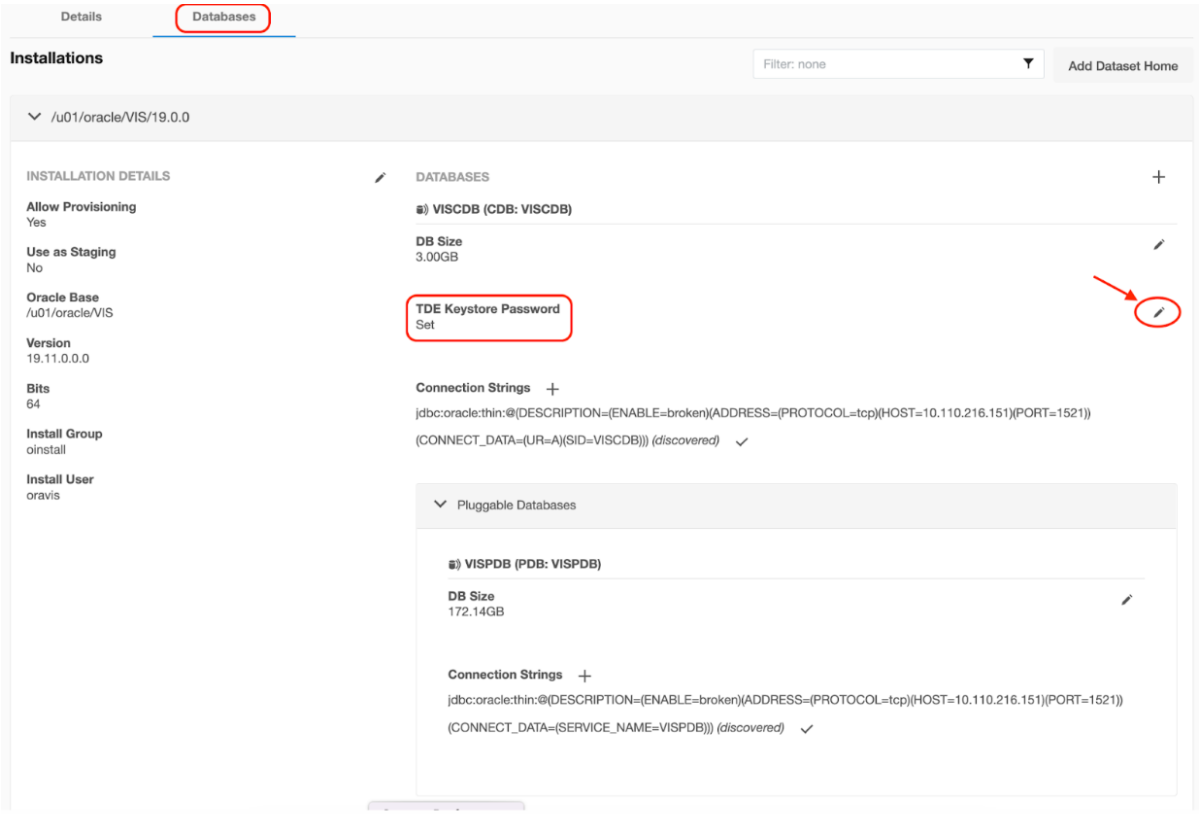
1. Login to the Delphix Management application.
2. Go to Manage > Environments.
3. Click the Details tab of the environment.
4. Click on the edit icon next to ATTRIBUTES to set or update attributes, including the TDE Keystores root directory path.
5. Click the tick mark button to save the configuration.



Adding or editing the target keystore password

The target Keystore password is specified on the Databases tab under Environments. To add or edit the target Keystore password, do the following:

1. Login to the Delphix Management application.
2. Go to **Manage > Environments**.
3. Click the **Databases** tab of the Environment.
4. Click on the edit icon next to **TDE Keystore Password** to set or update the password.
5. Click on **Add source**. The Add dSource wizard displays.
6. Fill in the details and click **Submit**.



9.8.6.2.2.4 Procedure

Linking can be done similarly to an Oracle data source (see page 1438), refer to Linking an Oracle pluggable database & Linking data sources with Oracle[489] for a detailed procedure.

489 https://cd.delphix.com/docs/latest/transparent-data-encryption-and-delphix

9.8.6.2.3 Linking the AppsTier

9.8.6.2.3.1 Prerequisites

The AppsTier must meet the source requirements outlined in Requirements for Source AppsTier (see page 1664). These requirements are generic to all source Unix environments added to the Delphix Continuous Data Engine.

9.8.6.2.3.2 Procedure

Follow these steps to link the AppsTier.

1. Login to the **Delphix Management** application using **Delphix Admin** credentials.
2. Click **Manage**.
3. Select **Environments**.
4. Select the **source AppsTier environment**. Linking from multi-node AppsTier

> **Note**: If you are linking from a multi-node AppsTier, select the environment for the node on which EBS admin WLS services reside.

5. Click the **Environment Details** tab.
6. If the applmgr environment user described in Requirements for Oracle EBS source databases for linking (see page 1664) is not already added to the Delphix Continuous Data Engine, add the user.
   Note - There is no actual requirement that the account be named applmgr on source environment. You can use the account that can run adpreclone.pl successfully and ingest the RUN filesystem directory into Delphix Continuous Data Engine.
7. Click the **Databases** tab.
8. Click the **Plus** icon within the E-Business **Suite R12.2 AppsTier** section under installations.
9. **Add Database** dialog box pops to add *source config* for AppsTier.
10. Provide a user-defined name to the 'Database Name' field and $APPS_BASE directory in the 'Path' field.
    $APPS_BASE path is where the EBSapps.env file is located.
11. Click on the **Add** button to successfully add the source config.
12. Re-check the **E-Business Suite R12.2 AppsTier** section under installations for the **Add dSource** option mapped to the source config that is added in the above step.
13. Click **Manage > Datasets.**
14. Click the **Plus** icon next to Datasets and select **Add dSource**.

15. In the **Add dSource wizard**, select the Linked dSource as the type.
16. In the **Add dSource wizard**, select the **AppsTier files source** you just created.
17. Enter the **EBS-specific parameters** for your AppsTier.
    These parameter values will be used when adpreclone.pl is run.
18. Select an Environment User.
19. **Privileged OS Account (Optional)** field should contain a high privileged user when the low privileged user is being used for linking.
20. Provide the apps schema password in the **Apps Password** section by choosing the **Username** and **Password** from the dropdown, leaving the username field empty.
21. Provide the weblogic password in the **Weblogic AdminServer Password** section by choosing the **Username** and **Password** from the dropdown, leaving the username field empty.
22. With 4.3.0 or later We **do not need** to specify the "relative paths" of files to exclude in the **Paths to Exclude** list.
    Note: Delphix Continuous Data Engine will automatically detect the RUN FS.
23. Click **Next**.
24. Enter a **dSource Name**.
25. Select a **Database Group** for the dSource.
26. Click **Next**. Adding a dSource to a database group enables you to set Delphix Domain user permissions for that dSource's objects, such as snapshots.
27. Click **Next**.
28. Select a **SnapSync** policy.
29. Click **Next**.
30. Enter any **custom pre or post-sync logic** as Pre-Sync or Post-Sync hook operations.
    Remember that “`perl adpreclone.pl AppsTier”` is already run before every SnapSync of the AppsTier.
    The Pre-Sync hook operations will be run before running the adpreclone.pl tool.
31. Click **Next**.

Review the **dSource Configuration** and **Data Management** information, and then click **Finish**.
The Delphix Continuous Data Engine will initiate two jobs to create the dSource, **DB_Link,** and **DB_Sync**. You can monitor these jobs by clicking **Active Jobs** in the top menu bar, or by selecting **System** > **Event Viewer**. When the jobs have completed successfully, the **files** icon will change to a **dSource** icon on the **Environments** > **Databases** screen, and the dSource will be added to the list of **Datasets** under its assigned group

9.8.6.3 Provisioning Oracle EBS VDBs

This sections covers the following topics:

- Provisioning a virtual Oracle EBS R12.2 instance (see page 1701)
- Provisioning the Oracle EBS DBTechstack (see page 1701)

- Provisioning TDE Enabled (19c) Oracle EBS PDB (see page 1706)
- Provisioning the Oracle 19C database (see page 1713)
- Provisioning the Oracle 12C database (see page 1728)
- Provisioning the EBS AppsTier (see page 1738)

9.8.6.3.1 Provisioning a virtual Oracle EBS R12.2 instance

The virtual database (VDB) is a Delphix virtualized copy of your source database that runs on your Target Environment. A VDB is provisioned from a dSource or VDB snapshot.

9.8.6.3.1.1 Provision/refresh rules

Provision/refresh has to be in an order for each dataset:

1. Provision DBTechStack (Oracle Home of the EBS Database)
2. Provision DB Tier(Oracle Database for EBS Application)
3. Provision Apps Tier (Oracle E-Business Application)

If any dataset's provision/refresh has failed, the next dataset provision/refresh should not be started until the Tier beneath that dataset is UP & RUNNING. In simple words, If database provision/refresh has failed then Enduser cannot trigger AppsTier provision/refresh workflow from Delphix UI.

9.8.6.3.2 Provisioning the Oracle EBS DBTechstack

9.8.6.3.2.1 Prerequisites

Meet the following requirements to create a VDB:

1. Identify an originating dataset and snapshot.
   a. Link a dSource by following the instructions in Requirements for Oracle EBS databases for linking (see page 1664).
   b. Select a VDB that has been previously provisioned.
2. Set up an EBS target environment as described in Requirements for target databases (see page 1667).
3. To use privilege elevation, you must meet the Privilege Elevation Requirements (see page 1673).
4. Maintain the integrity in names(CDB name, PDB name, DB hostname, /etc/hosts file format nomenclature). During DBTechstack, vPDB or vDB provisioning, maintain the same lower case and use the same name vPDB during virtual AppsTier provision as well. There is no restriction on using UPPER or LOWER case characters. However, it is recommended to maintain consistency.
5. Snapshot coordination: Changes applied to EBS and picked up only in certain dSource snapshots may make certain combinations of snapshots across the AppsTier and dbTier incompatible. When provisioning, refreshing or rewinding a virtual EBS instance, be sure the points in time you select for each dataset are compatible with each other.

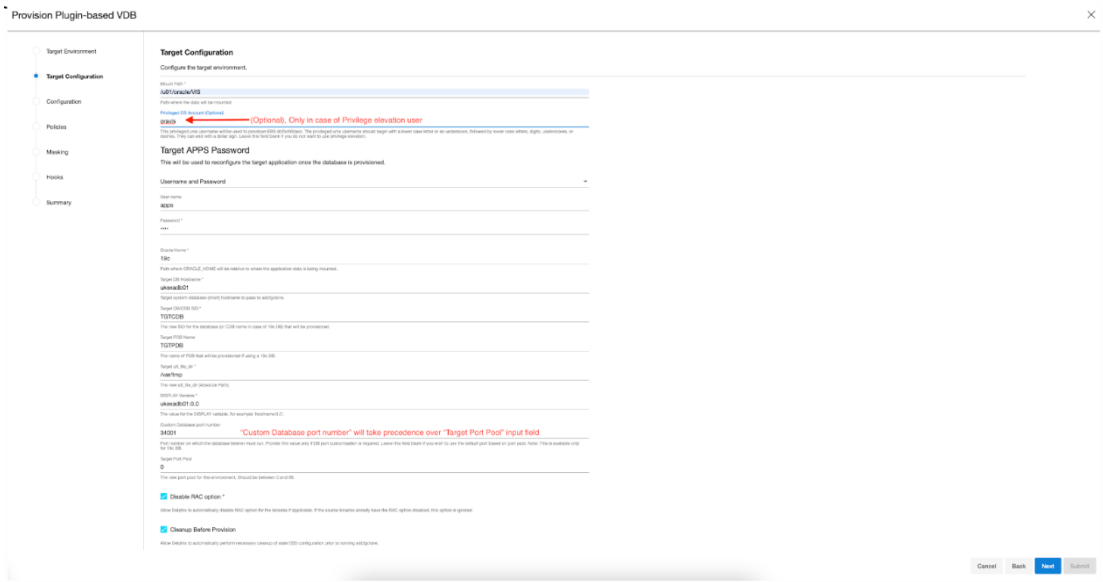
6. The source DB host may have multiple DATA_TOPs but the target DB host will be provisioned with a single DATA_TOP that resides on the target Mount Base path provided during the database provisioning.

9.8.6.3.2.2 Procedure

Perform the following steps to provision a DBTechstack VDB.

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Datasets**.
4. Select the **DBTechstack dSource**.
5. Click the **TimeFlow** tab.
6. Select a dSource **snapshot**.
7. Click **Provision**.
   The **Provision VDB** wizard will open.
8. Select an **Environment.**
   This environment will host the virtual DBTechstack and be used to execute hook operations specified in step 15 in the **section**.
9. Select **E-Business Suite R12.2 DBTechstack** from the **Installation** dropdown.
10. Select an **Environment User**.
11. This user should be the Oracle user-outlined in the Requirements for Oracle EBS target databases for provisioning (see page 1667)
12. Enter a **Mount Path** for the virtual DBTechstack files.
13. Enter the **EBS-specific parameters** for the virtual DBTechstack. A subset of these parameters is discussed in more detail below.
    a. The **Privileged OS Account (Optional)** field should contain the high-privileged user when the low-privileged user is being used for provisioning.
    b. The **Target APPS Password** is the new apps password that is required to configure virtual DBTechstack. This password is encrypted when stored within the Delphix Continuous Data Engine and is available as an environment variable to the *adcfgclone* process.
    c. Provide source Oracle home directory name relative to Oracle base installation directory that was provided during linking of DBTechstack in Oracle Home input field.
    d. Ensure that the **Target DB Hostname** value is the short hostname, not the fully qualified hostname.
    e. The **Target DB/CDB SID** is the new database SID (CDB SID in case of 19c) that is required to configure the virtual DBTechstack.
    f. The **Target PDB Name** field is added for the pluggable database to be configured for the Oracle 19c database. Please leave this field empty when using the Oracle 11g/12c database.
    g. Provide **Target utl_file_dir** , the default value for this field will be “/var/tmp". The ***utl_file_dir*** initialization parameter is deprecated in Oracle Database 12c Release 2 (12.2.0.1).

h. Provide **DISPLAY Variable**, the default value is hostname:0.0
i. [Optional] Provide the **Custom Database Port Number** on which the database listener must run.
   Info: Enter a numeric port value in this field only if you want to use DB port customization. Leaving this field blank automatically uses the default port based on the port pool. The custom database port number is applicable only for the Oracle 19c database.
   i. To run the DBTechstack and listener on a custom DB port, it is recommended to provide **Target Port Pool** value either as default or 0; also the **Target Port Pool** value in DBTechstack is not related to **Run/Patch Edition Port Pool** value in AppsTier.
j. Provide **Target Port Pool**, please provide an available port pool. By default, the value is 1. The range for this value is 0 to 99.
k. Enable the **Disable RAC** option if you want to permit the Delphix Continuous Data Engine to automatically disable the RAC option for the binaries when applicable. The **Disable RAC** option is ticked(enabled) by default. This option is necessary if provisioning from a dSource with RAC dbTier because the binaries are relinked with the rac_on option even after running adcfgclone. If the source binaries already have the RAC option disabled (also the case for SI dbTier), the Delphix Continuous Data Engine ignores this option.
l. Enable the **Cleanup Before Provision Option** if you want to permit the Delphix Continuous Data Engine to automatically clean up stale EBS configuration during a refresh. This option is recommended, but only available if your Oracle Home is patched with Oracle Universal Installer (OUI) version 10.2 or above.
   i. With this option enabled, the Delphix Continuous Data Engine will inspect the target environment's oraInventory prior to refreshing this virtual DBTechstack. If any Oracle Homes are already registered within the specified **Mount Path**, the Delphix Continuous Data Engine will detach them from the inventory prior to running adcfgclone. These homes must be detached prior to running post-clone configuration. If they are not detached, adcfgclone will fail, citing conflicting oraInventory entries as an issue.
   ii. Without this option enabled, Oracle Homes that conflict with the specified **Mount Path** will be reported in an error instead of automatically detached. For refresh to succeed, you must manually detach conflicting Oracle Homes prior to refresh.
m. Provide **Oracle OS User** and **Group**, it’s optional.



14. Click **Next**.
15. Enter a **VDB Name**.
16. Select a **Target Group** for the VDB.
    If necessary, click the green **Plus** icon to add a new group.
17. (Optional)When provisioning a new VDB in the VDB Provisioning wizard, check the Auto VDB Restart box. Reference - Automatich VDB restart on target server after reboot[490].
18. Select a **Snapshot Policy** for the VDB.
    If necessary, click the **Plus** icon to create a new policy.
    To avoid SnapSync conflicts, spread out your SnapSync policies for an EBS instance by one hour or more.
19. Click **Next**.
20. (Optional) Enter any **custom hook operations** that are needed to help correctly manage the virtual DBTechstack files. For more information about these hooks, when they are run and how operations are written, visit the Hook scripts for automation and customization page and section.
    The Configure Clone hook will be run after the adcfgclone.pl tool has both mounted and configured the DBTechstack.
21. Click **Next**.
22. Click **Submit**.

When provisioning starts, you can review the progress of the job in the **Datasets** panel, or in the **Job History** panel of the **Dashboard**. When provisioning is complete, the DBTechstack VDB will be included in the group you designated and listed in the **Datasets** panel. If you select the DBTechstack VDB in the **Datasets** panel and click the **Configuration** tab, you can view information about the virtual files and its Data Management settings.

490 https://cd.delphix.com/docs/latest/automatic-vdb-restart-on-target-server-after-reboo

For tips on monitoring the progress of DBTechstack provisioning, see Monitoring EBS R12.2 DBTechstack Provisioning Progress[491]

9.8.6.3.2.3 Registering the EBS DBTechstack before database provisioning

Register the freshly-provisioned DBTechstack with the Delphix Continuous Data Engine.
It is a mandatory step to register the freshly-provisioned DBTechstack so we can provision the VDB using this ORACLE_HOME only.
There are two ways to discover the new DBTechstack ORACLE_HOME are as below -

First approach:

1. Refresh the dbTier environment.
   Refreshing the environment will discover an EBS database listener and ensure it is available for selection when provisioning the EBS database.
   a. Click the **Refresh** button in the top right-hand corner of the environment card.

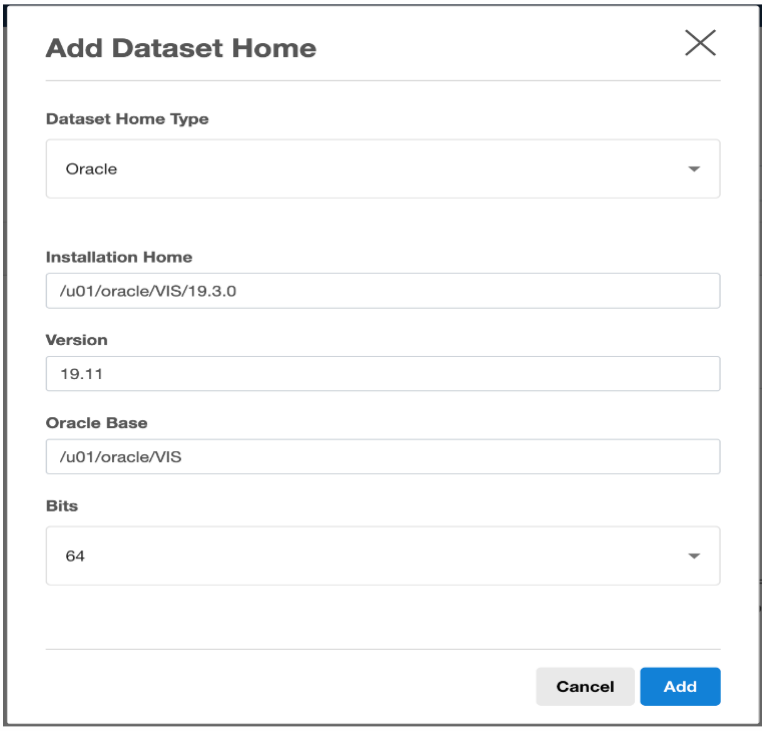
> **Note:** If the DBTechstack is not discovered by the first approach, follow the second approach to discover it manually.

Second approach:

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Environments**.
4. Select the **target dbTier environment**.
5. Click the **Databases** tab.
6. Click the **Add Dataset Home** button.
7. Select **Oracle** as your **Dataset Home Type**.
8. Enter an **Installation Home**.
   This path should be the value of $ORACLE_HOME on your target dbTier; this path will live under the Delphix mount path specified when you provisioned the virtual DBTechstack.
   Commonly, this path looks like /u01/oracle/VIS/11.2.0.
9. Enter the version of your Oracle installation in the **Version** textbox.
10. Enter the Oracle Base directory in the **Oracle Base** field.
11. Click the **add** button to save your dataset home.

491 https://cd.delphix.com/docs/latest/monitoring-ebs-r12-2-dbtechstack-provisioning-prog

If necessary, scroll down the list of dataset homes to view and edit this dataset home.



9.8.6.3.3 Provisioning TDE Enabled (19c) Oracle EBS PDB

Refer Provisioning a TDE-enabled vPDB[492] and
Provisioning a TDE software keystore based vPDB[493] sections.

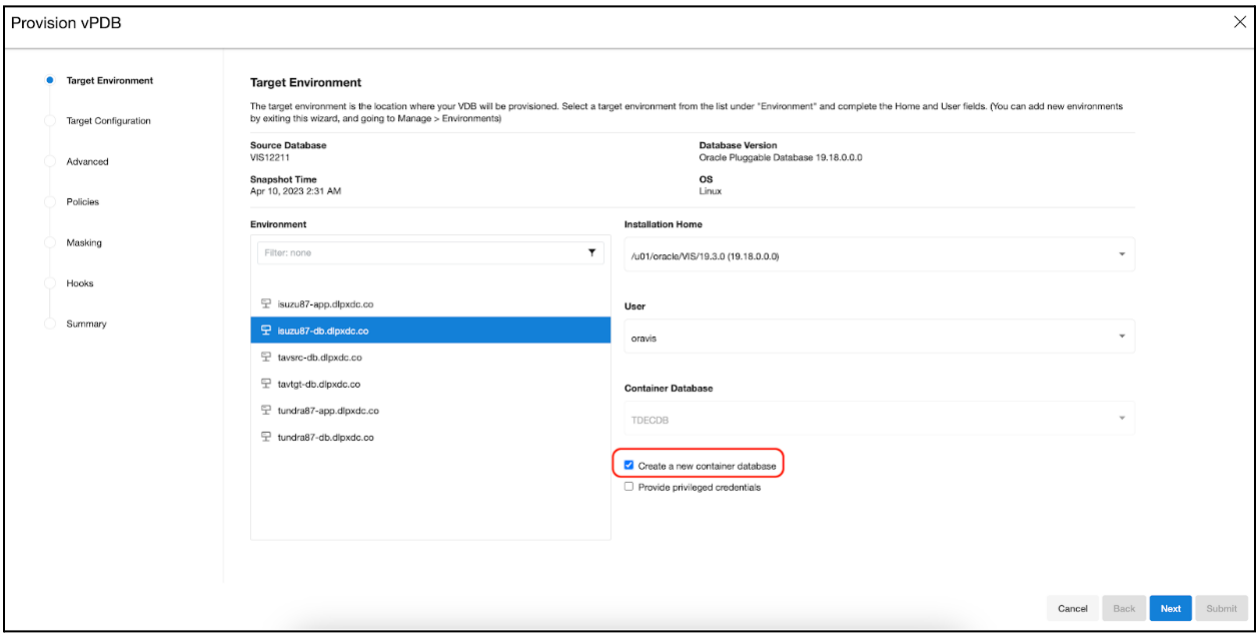
- Copy the source DB keystore (WALLET_ROOT) files cwallet.sso, ewallet.p12 to target server and mention that location in Parent Database TDE Keystore Location.
- On Source CDB, run below query to get WALLET_ROOT(files cwallet.sso, ewallet.p12) location -

```
select wrl_parameter from v$encryption_wallet;
```

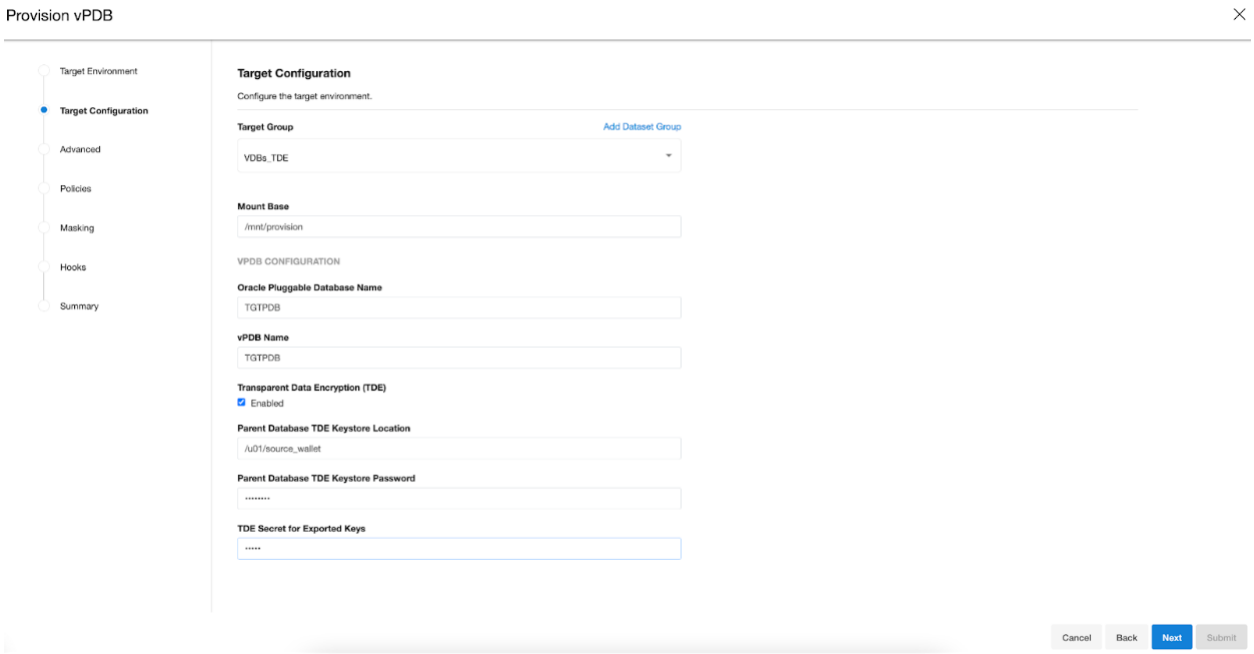
```
WRL_PARAMETER
 ------------------------
 /home/oravis/wallet/tde/
```

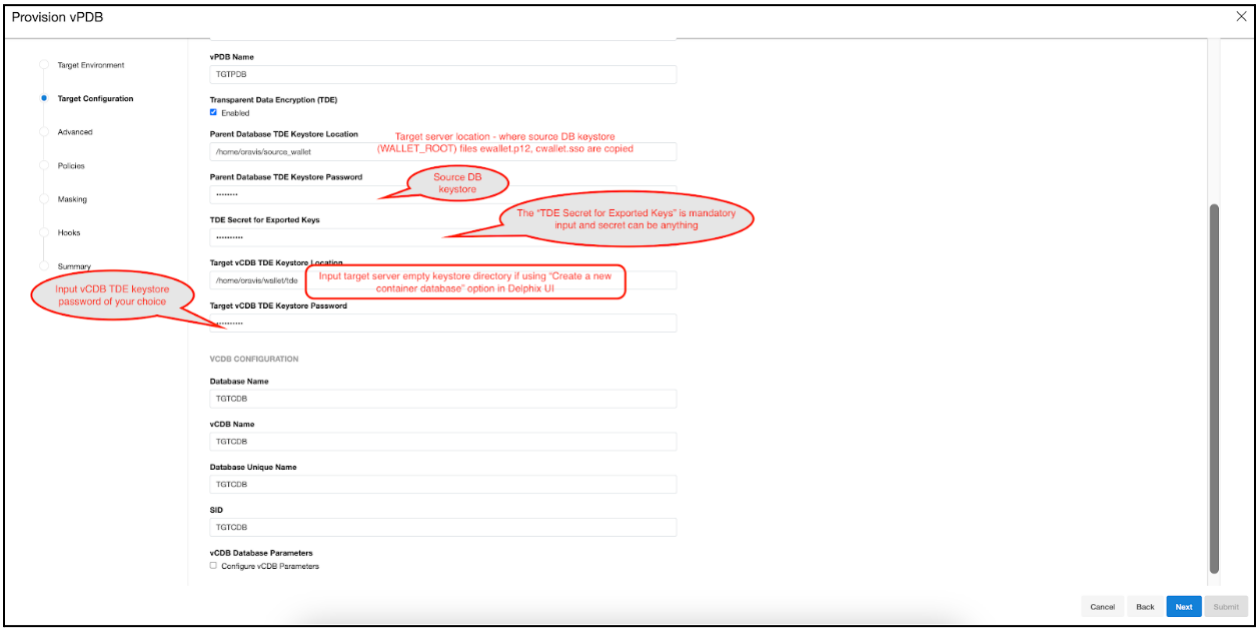
It is mandatory to copy the ewallet.p12 file from the source to the target server for the target container database to use it. No exporting of keys needed, this is done by the Delphix Continuous Data Engine itself using the path and secrets provided in the configuration. On the target server, cwallet.sso, ewallet.p12 files should be owned by the ORACLE_HOME installation owner.

492 https://cd.delphix.com/docs/latest/provisioning-a-tde-enabled-vpdb
493 https://cd.delphix.com/docs/latest/provisioning-a-tde-software-keystore-based-vpdb



The Parent Database TDE keystore Location is the location of the keystore for the dSource, which can be the actual location if it's a provision back to the source, or else it needs to be copied to the target. The parent keystore password is the password for the dSource keystore. You can decide the TDE Secret for Exported Keys, which is a mandatory input and secret can be anything, it is used when exporting and importing the encryption keys.





Once a TDE-enabled vPDB is provisioned, it can be used the same as a non-TDE-enabled vPDB within Delphix, with the exception of migrate. There are few caveats:

- A refresh operation will use the parent keystore for the recovery. If the dSource is rekeyed then the user will need to update the parent keystore with the new keys. Similarly, if the location or password to the parent keystore has changed then they should be updated before the refresh.
- A rewind operation will use the target keystore for the recovery. If the vPDB is rekeyed after it is provisioned, then the rekey will update the target keystore, so it does not need to be updated in Delphix.
- For a single vPDB in a vCDB, if the vCDB keystore location is changed, the new path must be updated in Delphix before refresh or rewind.
- Each disable operation will result in the keys being exported to an exported keyfile in the artifact directory, to be used for a subsequent enable. Refresh and rewind operations will first disable the existing vPDB, so those will also result in a new exported keyfile in the artifact directory.
- Provisioning a second-generation vPDB (vvPDB) from a TDE-enabled vPDB is done in the same manner as a first-generation vPDB, by specifying the TDE parameters during provision. The current keystore for the vPDB can be specified as the parent keystore.

For versions 19.3 or later, follow the below guidelines if you see any of the following error:

1. ORA-01017: invalid username/password logon denied
2. ORA-28040 During cloning: No Matching Authentication Protocol

Occasionally, it is necessary to restart the database to resolve connectivity to the database. This is atypical but may be necessary in certain cases.
On target container database, run the following command:
alter system set SEC_CASE_SENSITIVE_LOGON=FALSE scope=both;

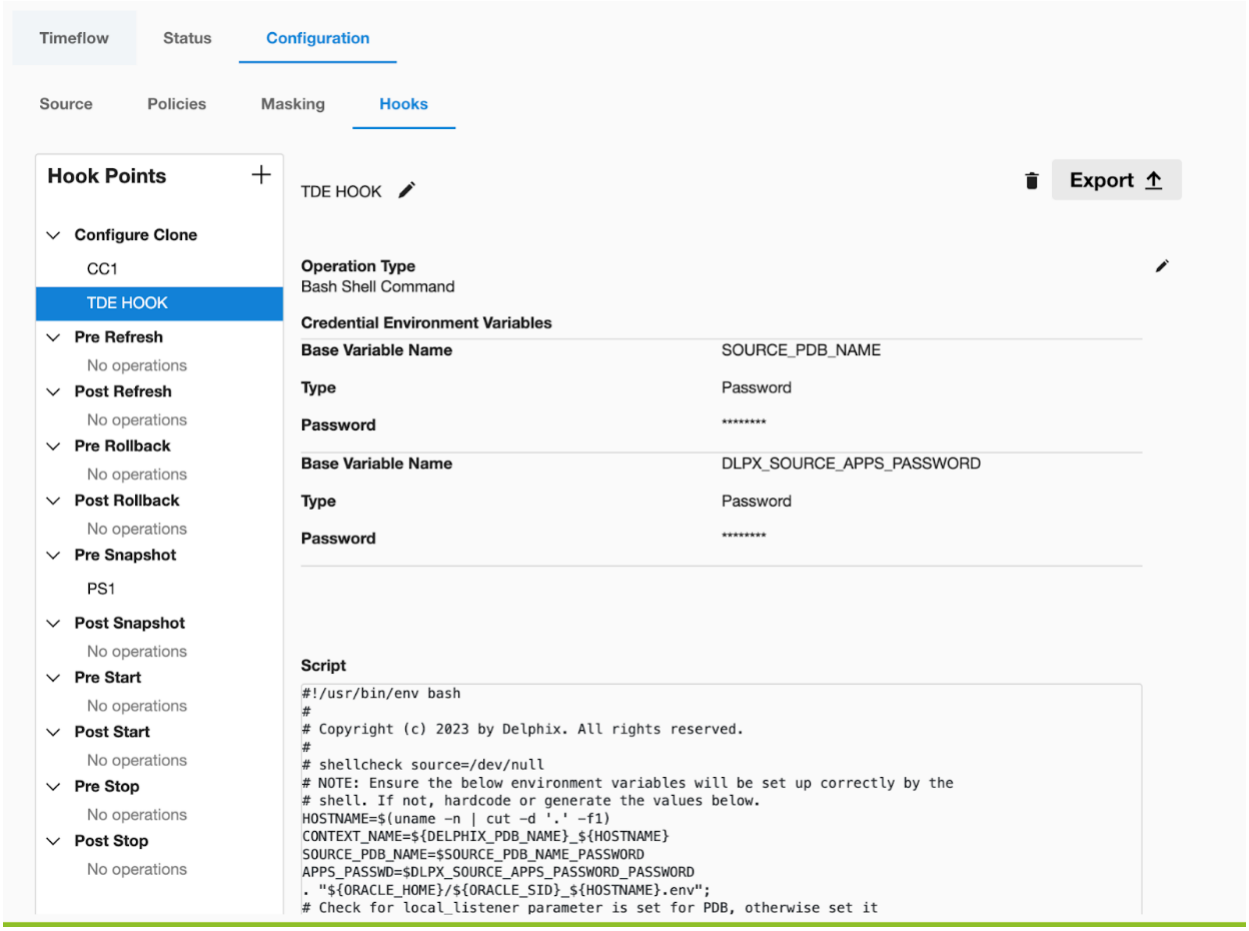
The above ALTER command should run as a first configure clone hook.

```
"${ORACLE_HOME}/${ORACLE_SID}_${HOSTNAME}.env";
sqlplus -s "/ as sysdba" <<EOF
alter system set SEC_CASE_SENSITIVE_LOGON=FALSE scope=both;
EOF
```

Configure clone first hook for a 19c multi-tenant high privileged user:

Configure clone hook will create and start the PDB specific services

> **Note**: It is mandatory to declare `DLPX_SOURCE_APPS_PASSWORD` for providing the source apps schema password and `SOURCE_PDB_NAME` for source PDB name as credential environment variables in hook sections.If you are linking from a RAC dbTier, select the environment for a single running node of the RAC cluster.




```
#!/usr/bin/env bash
#
# Copyright (c) 2023 by Delphix. All rights reserved.
```

```
#
# shellcheck source=/dev/null
# NOTE: Ensure the below environment variables will be set up correctly by the
# shell. If not, hardcode or generate the values below.
HOSTNAME=$(uname -n | cut -d '.' -f1)
CONTEXT_NAME=${DELPHIX_PDB_NAME}_${HOSTNAME}
SOURCE_PDB_NAME=$SOURCE_PDB_NAME_PASSWORD
APPS_PASSWD=$DLPX_SOURCE_APPS_PASSWORD_PASSWORD
. "${ORACLE_HOME}/${ORACLE_SID}_${HOSTNAME}.env";
# Check for local_listener parameter is set for PDB, otherwise set it
# appropriately
check_value=$(sqlplus -s "/ as sysdba" <<EOF
alter session set container="${DELPHIX_PDB_NAME}";
show parameter local_listener;
EOF
)
local_listener=$(echo "$check_value" | awk '{print $11}')
value=("${local_listener//:/ }")
host="${value[0]}"
port="${value[1]}"
curr_port=$(grep PORT < "${ORACLE_HOME}/network/admin/listener.ora" | awk '{print
$9}' | sed 's/)//g')
if [[ $port != "$curr_port" || $host != "${HOSTNAME}" ]]; then
sqlplus -s "/ as sysdba" <<EOF
alter session set container="${DELPHIX_PDB_NAME}";
alter system set local_listener='${HOSTNAME}:${curr_port}';
alter system register;
EOF
fi

#Create, Start EBS PDB services and Delete Source PDB services
. "${ORACLE_HOME}/${ORACLE_SID}_$(hostname -s).env";
sqlplus -s "/ as sysdba" <<EOF
alter session set container="${DELPHIX_PDB_NAME}";
BEGIN DBMS_SERVICE.STOP_SERVICE( service_name => '${SOURCE_PDB_NAME}_ebs_patch');
end ;
/
BEGIN DBMS_SERVICE.STOP_SERVICE( service_name => 'ebs_${SOURCE_PDB_NAME}'); end;
/
BEGIN DBMS_SERVICE.DELETE_SERVICE( service_name => '${SOURCE_PDB_NAME}_ebs_patch');
end ;
/
BEGIN DBMS_SERVICE.DELETE_SERVICE( service_name => 'ebs_${SOURCE_PDB_NAME}'); end;
/
BEGIN DBMS_SERVICE.CREATE_SERVICE( service_name => 'ebs_${DELPHIX_PDB_NAME}',
network_name => 'ebs_${DELPHIX_PDB_NAME}');
end;
/
BEGIN DBMS_SERVICE.CREATE_SERVICE( service_name => '${DELPHIX_PDB_NAME}_ebs_patch',
network_name => '${DELPHIX_PDB_NAME}_ebs_patch');
end;
/
BEGIN DBMS_SERVICE.START_SERVICE( service_name => 'ebs_${DELPHIX_PDB_NAME}');
```

```
end;
/
BEGIN DBMS_SERVICE.START_SERVICE( service_name => '${DELPHIX_PDB_NAME}_ebs_patch');
end;
/
alter system register;
EOF

#For compatibility with version 6.0.16.0 or later
. ${ORACLE_HOME}/${ORACLE_SID}_${HOSTNAME}.env; sqlplus -s "/ as sysdba" <<EOF
shutdown immediate;
startup;
ALTER PLUGGABLE DATABASE ALL OPEN read write services=all;
alter pluggable database ${DELPHIX_PDB_NAME} open read write services=all;
alter pluggable database all save state instances=all;
EOF
sqlplus "/ as sysdba" <<EOF
@${ORACLE_HOME}/appsutil/install/${CONTEXT_NAME}/adupdlib.sql so
EOF
. "${ORACLE_HOME}/${CONTEXT_NAME}.env";
perl "${ORACLE_HOME}/appsutil/clone/bin/adcfgclone.pl" dbconfig "${ORACLE_HOME}/
appsutil/${CONTEXT_NAME}.xml" <<EOF # noqa
${APPS_PASSWD}
EOF
```

Configure clone first hook for a 19c multi-tenant low privileged user:

```
#!/usr/bin/env bash
#
# Copyright (c) 2022 by Delphix. All rights reserved.
#
# shellcheck source=/dev/null
# NOTE: Ensure the below environment variables will be set up correctly by the
# shell. If not, hardcode or generate the values below.
# Input the SOURCE_PDB_NAME enclosed in double quotes.
# Input the DLPX_PRIV_USER according to your environment setup.
SOURCE_PDB_NAME="TDEPDB"
DLPX_DB_EXEC_SCRIPT="<Remote BIN location till dlpx_db_exec script>"
DLPX_PRIV_USER=oravis
SOURCE_PDB_NAME=$SOURCE_PDB_NAME_PASSWORD
APPS_PASSWD=$DLPX_SOURCE_APPS_PASSWORD_PASSWORD
CONTEXT_NAME=${DELPHIX_PDB_NAME}_$(hostname -s)

"${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" ". ${ORACLE_HOME}/${ORACLE_SID}_$
(hostname -s).env;"

# Check for local_listener parameter is set for PDB, otherwise set it appropriately
check_value=$("${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" ". ${ORACLE_HOME}/$
{CONTEXT_NAME}.env; sqlplus -s \"/ as sysdba\" <<-EOF
alter session set container=${DELPHIX_PDB_NAME};
show parameter local_listener;
EOF
```

```
")

local_listener=$(echo "$check_value" | awk '{print $11}')
value=("${local_listener//:/ }")
host="${value[0]}"
port="${value[1]}"

curr_port=$(grep PORT < "${ORACLE_HOME}/network/admin/listener.ora" | awk '{print
$9}' | sed 's/)//g')

if [[ $port != "$curr_port" || $host != "$(hostname -s)" ]]; then
    "${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" ". ${ORACLE_HOME}/${ORACLE_SID}_$
(hostname -s).env; sqlplus -s \"/ as sysdba\" <<EOF
    alter session set container=${DELPHIX_PDB_NAME};
    alter system set local_listener='$(hostname -s):${curr_port}';
    alter system register;
EOF
"
fi

#Create, Start EBS PDB services and Delete Source PDB services
"${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" ". ${ORACLE_HOME}/${ORACLE_SID}_$
(hostname -s).env; sqlplus -s "/ as sysdba" <<EOF
alter session set container="${DELPHIX_PDB_NAME}";
BEGIN DBMS_SERVICE.STOP_SERVICE( service_name => '${SOURCE_PDB_NAME}_ebs_patch');
end ;
/
BEGIN DBMS_SERVICE.STOP_SERVICE( service_name => 'ebs_${SOURCE_PDB_NAME}'); end;
/
BEGIN DBMS_SERVICE.DELETE_SERVICE( service_name => '${SOURCE_PDB_NAME}_ebs_patch');
end ;
/
BEGIN DBMS_SERVICE.DELETE_SERVICE( service_name => 'ebs_${SOURCE_PDB_NAME}'); end;
/
BEGIN DBMS_SERVICE.CREATE_SERVICE( service_name => 'ebs_${DELPHIX_PDB_NAME}',
network_name => 'ebs_${DELPHIX_PDB_NAME}');
end;
/
BEGIN DBMS_SERVICE.CREATE_SERVICE( service_name => '${DELPHIX_PDB_NAME}_ebs_patch',
network_name => '${DELPHIX_PDB_NAME}_ebs_patch');
end;
/
BEGIN DBMS_SERVICE.START_SERVICE( service_name => 'ebs_${DELPHIX_PDB_NAME}');
end;
/
BEGIN DBMS_SERVICE.START_SERVICE( service_name => '${DELPHIX_PDB_NAME}_ebs_patch');
end;
/
alter system register;
EOF
"
```

```
"${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" ". ${ORACLE_HOME}/${ORACLE_SID}_$
(hostname -s).env; sqlplus \"/ as sysdba\" <<-EOF
@${ORACLE_HOME}/appsutil/install/${CONTEXT_NAME}/adupdlib.sql so
EOF
"
"${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" ". ${ORACLE_HOME}/${CONTEXT_NAME}.env;
perl ${ORACLE_HOME}/appsutil/clone/bin/adcfgclone.pl dbconfig ${ORACLE_HOME}/
appsutil/${CONTEXT_NAME}.xml <<-EOF1
${APPS_PASSWD}
EOF1
"
```

The remaining configure clone & Pre-snapshot hooks remain unchanged. It is mandatory to use hooks mentioned in documentation while provisioning the TDE vPDB database.

9.8.6.3.4 Provisioning the Oracle 19C database

9.8.6.3.4.1 Procedure

Perform the following steps to provision Oracle 19c database VDB.

1. Refresh the Target DB environment from **Manage > Environments** so that the Listener brought up during DBTechstack provision gets discovered and the Installation Home to be listed in dropdown during Database provision. If the Target DB environment is not refreshed, you will see `This environment has no compatible Oracle Installation Homes` in the Installation Home dropdown in the **Database provision**.
2. Provision the EBS database to the target dbTier environment by following the steps outlined in Provisioning an Oracle VDB[494]. For Oracle EBS database for 19c MT, provision the database by following the steps outlined in Provisioning an Oracle Virtual Pluggable Database (vPDB)[495]
   Note: Snapshot conflicts: when Snapshot is running against the DBTechstack, database, or AppsTier, the Delphix Continuous Data Engine also executes pre-clone logic to ensure the latest configuration is staged in the captured snapshots. Unfortunately, if multiple Snapshots are running against the same EBS instance concurrently, this pre-clone logic may fail and produce bad snapshots.
   To avoid SnapSync conflicts, spread out your SnapSync policies for an EBS instance by one hour or more.
3. Select the correct **Installation Home**.
   This should be the virtual DBTechstack you just added to the Delphix Continuous Data Engine.
4. Select an Environment User. This user should be the oracle user-outlined in Preparing Target EBS R12.2 Environments for Provisioning.
5. For the **Container Database** option, select the checkbox next to **Create a New Container Database**, and click **Next**.
6. Select a Target Group for the VDB.

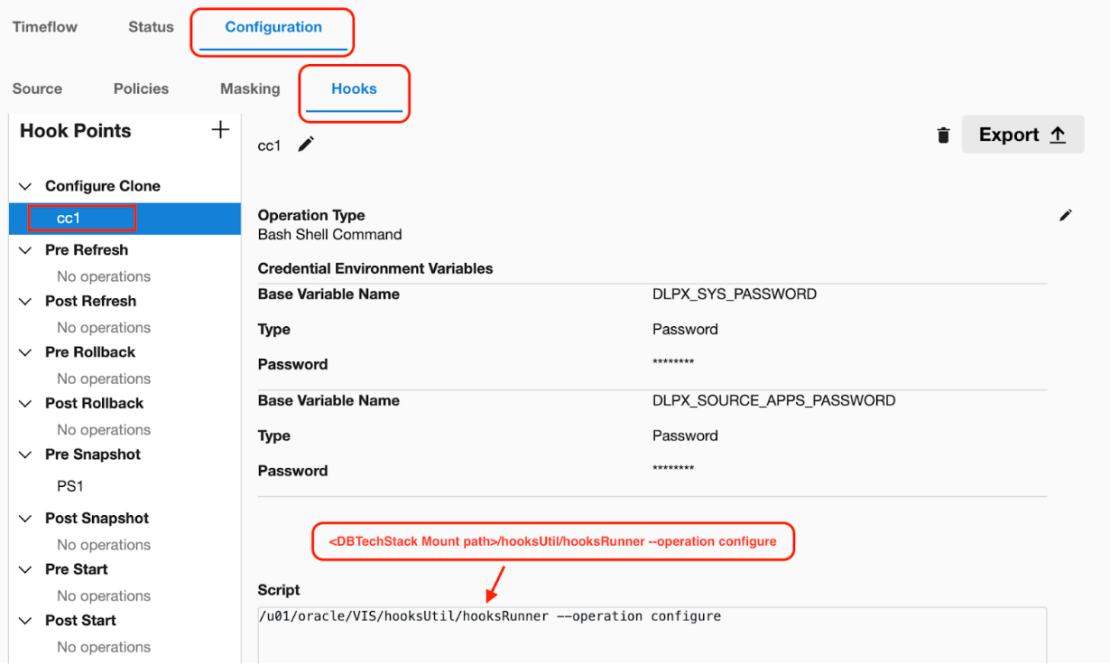
494 https://cd.delphix.com/docs/latest/provisioning-an-oracle-vdb
495 https://cd.delphix.com/docs/latest/provisioning-an-oracle-virtual-pluggable-database-

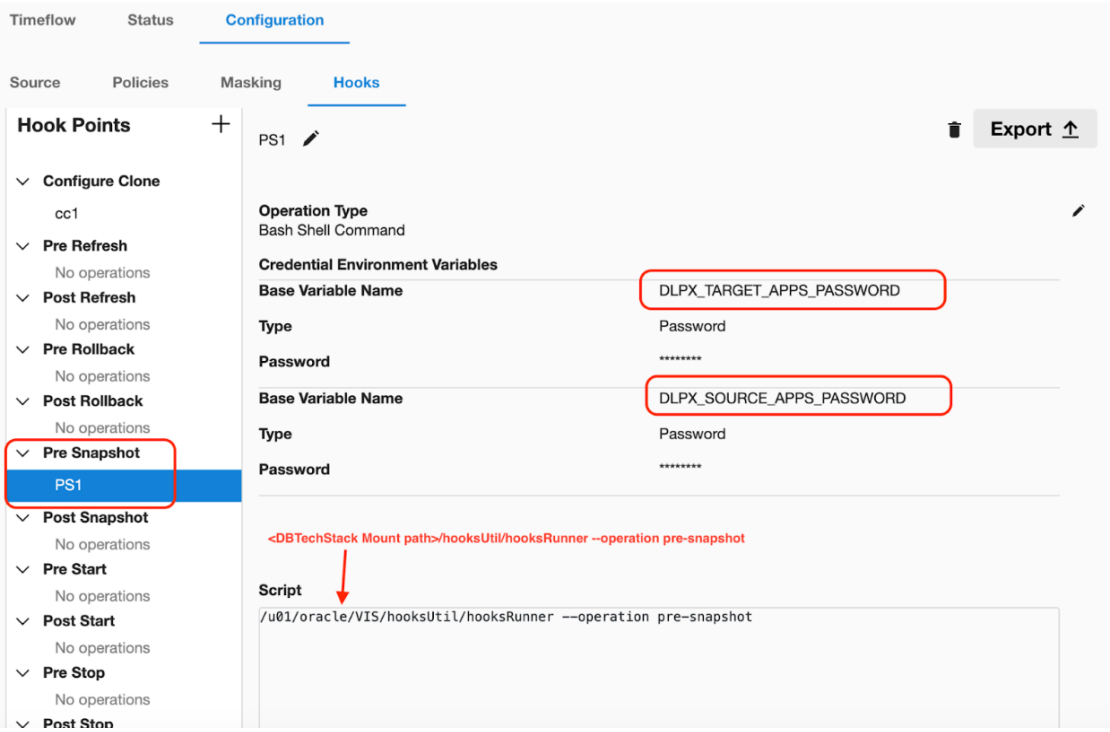
7. In the vPDB **Configuration** section, provide the Target PDB name given during DBTechstack provision in the **Oracle Pluggable Database Name** field and provide any unique user-defined name in **vPDB Name** to be displayed in the Delphix Continuous Data Engine.
8. In vCDB **CONFIGURATION** section, provide the Target DB/CDB SID name given during DBTechstack provision in the **Database Name** field and SID and provide any unique user-defined name in the **vCDB Name** and **Database Unique Name** field, and click **Next**.
9. Click **Advanced**.
10. Select the correct **Oracle Node Listeners** value.
    This should be the listener corresponding to the virtual DBTechstack you just added to the Delphix Continuous Data Engine.
11. (Optional) Add the EBS R12.2 dbTier environment file as a **Custom Environment Variables** entry.
    This file can be specified as an Environment File with Path Parameters of $ORACLE_HOME/<CONTEXT_NAME> .
12. Replace <CONTEXT_NAME> with the virtual EBS instance's context name. The Delphix Continuous Data Engine will expand the $ORACLE_HOME variable at runtime.
    For more information, see Customizing Oracle VDB environment variables[496]

> **Note**: To complete the DB Provisioning operation, you must provide content to configure clones and pre-snapshot hooks. You must either follow step 12 to use the hooks utility feature OR proceed with the conventional way by following steps 13-18.

13. To complete the DB Provisioning operation, you must provide content to configure clones and pre-snapshot hooks. The following are the content for both the hooks:
    a. Configure Clone hook will be `<DBTechstack Mount point>/hooksUtil/hooksRunner --operation configure` . You must set a hooks environment variable `DLPX_SOURCE_APPS_PASSWORD` for providing the source apps schema password and `DLPX_SYS_PASSWORD` for SYSTEM schema password.
    b. Pre-Snapshot hook : `<DBTechstack Mount point>/hooksUtil/hooksRunner --operation pre-snapshot` . You must set a hooks environment variable `DLPX_SOURCE_APPS_PASSWORD` for providing the source apps schema password. In case the apps password change is required in target VDB, you can set hook environment variables `DLPX_TARGET_APPS_PASSWORD` for providing the target apps schema.

496 https://cd.delphix.com/docs/latest/customizing-oracle-vdb-environment-variables





Note:

For versions 19.3 or later, follow the below guidelines if you see any of the following error:

i. ORA-01017: invalid username/password logon denied
ii. ORA-28040 During cloning: No Matching Authentication Protocol

Occasionally, it is necessary to restart the database to resolve connectivity to the database. This is atypical but may be necessary in certain cases.

On target container database, run the following command:
alter system set SEC_CASE_SENSITIVE_LOGON=FALSE scope=both;
The above ALTER command should run as a first configure clone hook.

```
. "${ORACLE_HOME}/${ORACLE_SID}_${HOSTNAME}.env";
sqlplus -s "/ as sysdba" <<EOF
alter system set SEC_CASE_SENSITIVE_LOGON=FALSE scope=both;
EOF
```

Warning: EBS application database user accounts created in the earlier release uses a case-insensitive password version from an earlier release authentication protocol, such as the 10G password version. If your release user account passwords have not been reset and the following conditions persists, then take the action as described below:
On source, the database server has been configured with SEC_CASE_SENSITIVE_LOGON set to FALSE, so that it can only authenticate users who have a 10G case-insensitive password version. Then, on the target container VDB, it becomes necessary to set the SEC_CASE_SENSITIVE_LOGON to FALSE, to make authentication of users successful.

Note: To take advantage of the password protections introduced in Oracle Database 19c, users must change their passwords.

14. Add a **Run Bash Shell Command** operation to the Configure Clone hook to ensure that `adcfgclone` utility is run against the newly provisioned database for a **high privileged user**, as shown in the script below.

Note: You can pass the credentials securely to Hook Operations by setting up the base variables. See Other hook operations[497] for more details.

497 https://cd.delphix.com/docs/latest/other-operations

Note: It is mandatory to set a hooks environment variable `DLPX_SOURCE_APPS_PASSWORD` for providing the source apps schema password. In case the apps password change is required in target VDB, you can set hook environment variables `DLPX_TARGET_APPS_PASSWORD` and `DLPX_SYS_PASSWORD` for providing the target apps schema and system schema passwords in respective hooks. The variables will be declared in the Credential **Environment Variables** hook sectionYou can pass the credentials securely to Hook Operations by setting up the base variables. See Other hook operations[498] for more details.

15. The hooks environment variable will be declared as, `DLPX_SOURCE_APPS_PASSWORD` and then its password value will be the same variable used in hooks, along with the `PASSWORD` keyword appended at the end so that it will be like `DLPX_SOURCE_APPS_PASSWORD_PASSWORD`
    We assign the value of $ `DLPX_SOURCE_APPS_PASSWRD` to `APPS_PASSWD`.
    In hooks, it is written as

```
APPS_PASSWD=$DLPX_SOURCE_APPS_PASSWORD_PASSWORD
```

Same goes for `SYSTEM_PASSWD:`

```
SYSTEM_PASSWD=$DLPX_SYS_PASSWORD_PASSWORD
```

498 https://cd.delphix.com/docs/latest/other-operations

Note: Customers have to identify the DLPX_DB_EXEC_SCRIPT value from their remote environment, this would always be inside the toolkit directory provided while adding the environment on Delphix Continuous Data Engine.You should replace the value for `DLPX_DB_EXEC_SCRIPT` from all the given hooks according to the path that exists in your environment. Please replace the following lines from hook scripts accordingly.You can pass the credentials securely to Hook Operations by setting up the base variables. See Other hook operations[499] for more details.

```
DLPX_DB_EXEC_SCRIPT="<Remote BIN location till dlpx_db_exec script>"
DLPX_PRIV_USER=<replace with your db user> eg:oravis
```

Configure clone first hook for a 19c multi-tenant high privileged user.

```
#!/usr/bin/env bash
#
# Copyright (c) 2022 by Delphix. All rights reserved.
#
# shellcheck source=/dev/null
# NOTE: Ensure the below environment variables will be set up correctly by the
# shell. If not, hardcode or generate the values below.
HOSTNAME=$(uname -n | cut -d '.' -f1)
CONTEXT_NAME=${DELPHIX_PDB_NAME}_${HOSTNAME}
APPS_PASSWD=$DLPX_SOURCE_APPS_PASSWORD_PASSWORD
. "${ORACLE_HOME}/${ORACLE_SID}_${HOSTNAME}.env";
# Check for local_listener parameter is set for PDB, otherwise set it
# appropriately
check_value=$(sqlplus -s "/ as sysdba" <<EOF
alter session set container="${DELPHIX_PDB_NAME}";
show parameter local_listener;
EOF
)
local_listener=$(echo "$check_value" | awk '{print $11}')
value=("${local_listener//:/ }")
host="${value[0]}"
port="${value[1]}"
curr_port=$(grep PORT < "${ORACLE_HOME}/network/admin/listener.ora" | awk '{print
$9}' | sed 's/)//g')
if [[ $port != "$curr_port" || $host != "${HOSTNAME}" ]]; then
sqlplus -s "/ as sysdba" <<EOF
alter session set container="${DELPHIX_PDB_NAME}";
alter system set local_listener='${HOSTNAME}:${curr_port}';
alter system register;
EOF
fi
```

499 https://cd.delphix.com/docs/latest/other-operations

```
#For compatibility with 6.0.16.0 Delphix Continuous Data Engine version
. ${ORACLE_HOME}/${ORACLE_SID}_${HOSTNAME}.env; sqlplus -s "/ as sysdba" <<EOF
shutdown immediate;
startup;
ALTER PLUGGABLE DATABASE ALL OPEN read write services=all;
alter pluggable database ${DELPHIX_PDB_NAME} open read write services=all;
alter pluggable database all save state instances=all;
EOF
sqlplus "/ as sysdba" <<EOF
@${ORACLE_HOME}/appsutil/install/${CONTEXT_NAME}/adupdlib.sql so
EOF
. "${ORACLE_HOME}/${CONTEXT_NAME}.env";
perl "${ORACLE_HOME}/appsutil/clone/bin/adcfgclone.pl" dbconfig "${ORACLE_HOME}/
appsutil/${CONTEXT_NAME}.xml" <<EOF # noqa
${APPS_PASSWD}
EOF
```

16. Configure Clone hook to ensure that `adcfgclone` is run against the newly provisioned database for a **low privileged user**, as shown in the script below. Configure clone first hook for a 19c multi-tenant low privileged user

```
#!/usr/bin/env bash
#
# Copyright (c) 2022 by Delphix. All rights reserved.
#
# shellcheck source=/dev/null
# NOTE: Ensure the below environment variables will be set up correctly by the
# shell. If not, hardcode or generate the values below.
DLPX_DB_EXEC_SCRIPT="<Remote BIN location till dlpx_db_exec script>"
DLPX_PRIV_USER=oravis
HOSTNAME=$(uname -n | cut -d '.' -f1)
APPS_PASSWD=$DLPX_SOURCE_APPS_PASSWORD_PASSWORD
CONTEXT_NAME=${DELPHIX_PDB_NAME}_${HOSTNAME}

"${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" ". ${ORACLE_HOME}/${ORACLE_SID}
_${HOSTNAME}.env;"

# Check for local_listener parameter is set for PDB, otherwise set it
appropriately
check_value=$("${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" ". ${ORACLE_HOME}/
${CONTEXT_NAME}.env; sqlplus -s \"/ as sysdba\" <<-EOF
alter session set container=${DELPHIX_PDB_NAME};
show parameter local_listener;
EOF
")

local_listener=$(echo "$check_value" | awk '{print $11}')
value=("${local_listener//:/ }")
host="${value[0]}"
```

```
port="${value[1]}"

curr_port=$(grep PORT < "${ORACLE_HOME}/network/admin/listener.ora" | awk
'{print $9}' | sed 's/)//g')

if [[ $port != "$curr_port" || $host != "${HOSTNAME}" ]]; then
    "${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" ". ${ORACLE_HOME}/$
{ORACLE_SID}_${HOSTNAME}.env; sqlplus -s \"/ as sysdba\" <<EOF
    alter session set container=${DELPHIX_PDB_NAME};
    alter system set local_listener='${HOSTNAME}:${curr_port}';
    alter system register;
EOF
"
fi

#For compatibility with 6.0.16.0 Delphix Continuous Data Engine version
"${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" ". ${ORACLE_HOME}/${ORACLE_SID}
_${HOSTNAME}.env; sqlplus -s "/ as sysdba" <<EOF
shutdown immediate;
startup;
ALTER PLUGGABLE DATABASE ALL OPEN read write services=all;
alter pluggable database ${DELPHIX_PDB_NAME} open read write services=all;
alter pluggable database all save state instances=all;
EOF
"
"${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" ". ${ORACLE_HOME}/${ORACLE_SID}
_${HOSTNAME}.env; sqlplus \"/ as sysdba\" <<-EOF
@${ORACLE_HOME}/appsutil/install/${CONTEXT_NAME}/adupdlib.sql so
EOF
"
"${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" ". ${ORACLE_HOME}/$
{CONTEXT_NAME}.env; perl ${ORACLE_HOME}/appsutil/clone/bin/adcfgclone.pl
dbconfig ${ORACLE_HOME}/appsutil/${CONTEXT_NAME}.xml <<-EOF1
${APPS_PASSWD}
EOF1
"
```

17. While setting up hooks environment variables, if your environment is on R12.TXK.C.Delta.13 or later, set `DLPX_SYS_PASSWORD` variable's password as the password of the EBS_SYSTEM user, else if your environment is on R12.TXK.C.Delta.12 or earlier, set `DLPX_SYS_PASSWORD` variable's password as the password of the SYSTEM user in the below second hook script.
This requirement is outlined in the FAQ: [Oracle E-Business Suite and System Schema Migration (DocID 2758999.1)](500) [Using UTL_FILE_DIR or Database Directories for PL/SQL File I/O in Oracle E-

500 https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=516370417411931&id=2758999.1&_adf.ctrl-state=qra3xc5e9_53

Business Suite Releases 12.1 and 12.2 (DocID 2525754.1)[501] found at http://docs.oracle.com[502]
Configure clone second hook for a 19c multi-tenant high privileged user:

```
#!/usr/bin/env bash
#
# Copyright (c) 2022 by Delphix. All rights reserved.
#
# shellcheck source=/dev/null
# Set the target UTL_FILE_DIR values in the Oracle 19c database in the below
# variable UTL_FILE_DIR_PATH separated by commas
# Example UTL_FILE_DIR_PATH=/u01/oracle/VIS/temp/VISPDB,\
# /u01/oracle/VIS/19.3.0/appsutil/outbound/UTL,/u01/tmp

# UTL_FILE_DIR_PATH=<provide utl_file_dir locations separated by commas>
# If your environment is on R12.TXK.C.Delta.13 or later, enter the password
# for the EBS_SYSTEM user. If your environment is on R12.TXK.C.Delta.12 or
# earlier, enter the password for the SYSTEM user.
SYSTEM_PASSWD=$DLPX_SYS_PASSWORD_PASSWORD
DLPX_HOSTNAME=$(uname -n | cut -d '.' -f1)
TIMESTAMP=$(date +%d-%m-%Y_%H-%M-%S)
CONTEXT_NAME=${DELPHIX_PDB_NAME}_${DLPX_HOSTNAME}
APPS_PASSWD=$DLPX_SOURCE_APPS_PASSWORD_PASSWORD
DIR_OBJ_PATH="${ORACLE_HOME}/appsutil/outbound/${DELPHIX_PDB_NAME}_$
{DLPX_HOSTNAME}"
checkExists() {
if [[ ! -f $1 ]]; then
    echo "$1 does not exist."
    exit 1
fi
}
checkExists "${ORACLE_HOME}/${CONTEXT_NAME}.env"
checkExists "${ORACLE_HOME}/appsutil/bin/txkCfgUtlfileDir.pl"
. "${ORACLE_HOME}/${CONTEXT_NAME}.env";
echo "${APPS_PASSWD}" | perl "${ORACLE_HOME}/appsutil/bin/txkCfgUtlfileDir.pl"
 -contextfile="${ORACLE_HOME}/appsutil/${DELPHIX_PDB_NAME}_$
{DLPX_HOSTNAME}.xml" -oraclehome="${ORACLE_HOME}" -outdir="${ORACLE_HOME}/
appsutil/log" -mode=getUtlFileDir
checkExists "${ORACLE_HOME}/dbs/${DELPHIX_PDB_NAME}_utlfiledir.txt"
cp "${ORACLE_HOME}/dbs/${DELPHIX_PDB_NAME}_utlfiledir.txt" "${ORACLE_HOME}/dbs/
${DELPHIX_PDB_NAME}_utlfiledir.txt${TIMESTAMP}"
cat /dev/null > "${ORACLE_HOME}/dbs/${DELPHIX_PDB_NAME}_utlfiledir.txt"
if [ -z "$UTL_FILE_DIR_PATH" ]; then
    mkdir -p "${ORACLE_HOME}/appsutil/outbound/${DELPHIX_PDB_NAME}_$
{DLPX_HOSTNAME}"
    echo "${ORACLE_HOME}/appsutil/outbound/${DELPHIX_PDB_NAME}_$
{DLPX_HOSTNAME}" > "${ORACLE_HOME}/dbs/${DELPHIX_PDB_NAME}_utlfiledir.txt"
```

501 https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=516418361250069&id=2525754.1&_adf.ctrl-state=qra3xc5e9_110
502 https://docs.oracle.com/

```
    mkdir -p "${ORACLE_BASE}/temp/${DELPHIX_PDB_NAME}"
    echo "${ORACLE_BASE}/temp/${DELPHIX_PDB_NAME}" >> "${ORACLE_HOME}/dbs/$
{DELPHIX_PDB_NAME}_utlfiledir.txt"
else
    echo "$UTL_FILE_DIR_PATH" | awk -F, '{ for (i = 1; i < NF+1; ++i ) print $i
>> "${ORACLE_HOME}/dbs/${DELPHIX_PDB_NAME}_utlfiledir.txt" }'
fi
{ echo "${APPS_PASSWD}"; echo "${SYSTEM_PASSWD}"; } | perl "${ORACLE_HOME}/
appsutil/bin/txkCfgUtlfileDir.pl" -contextfile="${ORACLE_HOME}/appsutil/$
{DELPHIX_PDB_NAME}_${DLPX_HOSTNAME}.xml" -oraclehome="${ORACLE_HOME}" -outdir="
${ORACLE_HOME}/appsutil/log" -mode=setUtlFileDir
{ echo "${APPS_PASSWD}"; echo "${SYSTEM_PASSWD}"; echo "${DIR_OBJ_PATH}"; } |
perl "${ORACLE_HOME}/appsutil/bin/txkCfgUtlfileDir.pl" -contextfile="$
{ORACLE_HOME}/appsutil/${DELPHIX_PDB_NAME}_${DLPX_HOSTNAME}.xml" -oraclehome="$
{ORACLE_HOME}" -outdir="${ORACLE_HOME}/appsutil/log" -mode=createDirObject
echo "${APPS_PASSWD}" | perl "${ORACLE_HOME}/appsutil/bin/txkCfgUtlfileDir.pl"
 -contextfile="${ORACLE_HOME}/appsutil/${DELPHIX_PDB_NAME}_$
{DLPX_HOSTNAME}.xml" -oraclehome="${ORACLE_HOME}" -outdir="${ORACLE_HOME}/
appsutil/log" -mode=syncUtlFileDir -skipautoconfig=yes
```

18. Configure Clone hook to ensure that sqlnet.ora or sqlnet_ifile.ora specify a value for SQLNET.ALLOWED_LOGON_VERSION_SERVER for a **low privileged user**, as shown in the script below.
    While setting up hooks environment variables, if your environment is on R12.TXK.C.Delta.13 or later, set `DLPX_SYS_PASSWORD` variable's password as the password of the EBS_SYSTEM user, else if your environment is on R12.TXK.C.Delta.12 or earlier, set `DLPX_SYS_PASSWORD` variable's password as the password of the SYSTEM user in the below second hook script.
    This requirement is outlined in the FAQ: Oracle E-Business Suite and System Schema Migration (DocID 2758999.1)[503] Using UTL_FILE_DIR or Database Directories for PL/SQL File I/O in Oracle E-Business Suite Releases 12.1 and 12.2 (DocID 2525754.1)[504] found at http://docs.oracle.com[505].
    Configure clone second hook for a 19c multi-tenant low privileged user

```
#!/usr/bin/env bash
#
# Copyright (c) 2022 by Delphix. All rights reserved.
#
# shellcheck source=/dev/null
# Set the target UTL_FILE_DIR values in the Oracle 19c database
# Example UTL_FILE_DIR_PATH=/u01/oracle/VIS/temp/VISPDB,\
# /u01/oracle/VIS/19.3.0/appsutil/outbound/UTL

# UTL_FILE_DIR_PATH=<provide utl_file_dir locations separated by commas>
```

503 https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=516370417411931&id=2758999.1&_adf.ctrl-state=qra3xc5e9_53

504 https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=516418361250069&id=2525754.1&_adf.ctrl-state=qra3xc5e9_110

505 https://docs.oracle.com/

```
# If your environment is on R12.TXK.C.Delta.13 or later, enter the password
# for the EBS_SYSTEM user. If your environment is on R12.TXK.C.Delta.12 or
# earlier, enter the password for the SYSTEM user.
SYSTEM_PASSWD=$DLPX_SYS_PASSWORD_PASSWORD
DLPX_DB_EXEC_SCRIPT="<Remote BIN location till dlpx_db_exec script>"
DLPX_PRIV_USER=oravis
APPS_PASSWD=$DLPX_SOURCE_APPS_PASSWORD_PASSWORD
HOSTNAME=$(uname -n | cut -d '.' -f1)
CONTEXT_NAME=${DELPHIX_PDB_NAME}_${HOSTNAME}
DIR_OBJ_PATH="${ORACLE_HOME}/appsutil/outbound/${CONTEXT_NAME}"
TIMESTAMP=$(date +%d-%m-%Y_%H-%M-%S)
checkExists() {
if [[ ! -f $1 ]]; then
    echo "$1 does not exist."
    exit 1
fi
}
checkExists "${ORACLE_HOME}/${CONTEXT_NAME}.env"
checkExists "${ORACLE_HOME}/appsutil/bin/txkCfgUtlfileDir.pl"
. "${ORACLE_HOME}/${CONTEXT_NAME}.env";

"${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" ". ${ORACLE_HOME}/$
{CONTEXT_NAME}.env; perl ${ORACLE_HOME}/appsutil/bin/txkCfgUtlfileDir.pl
-contextfile=${ORACLE_HOME}/appsutil/${CONTEXT_NAME}.xml -oraclehome=$
{ORACLE_HOME} -outdir=${ORACLE_HOME}/appsutil/log -mode=getUtlFileDir <<-EOF
${APPS_PASSWD}
EOF
"

checkExists "${ORACLE_HOME}/dbs/${DELPHIX_PDB_NAME}_utlfiledir.txt"
"${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" "cp ${ORACLE_HOME}/dbs/$
{DELPHIX_PDB_NAME}_utlfiledir.txt" "${ORACLE_HOME}/dbs/${DELPHIX_PDB_NAME}
_utlfiledir.txt${TIMESTAMP}"
"${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" "echo ""  > \"${ORACLE_HOME}/
dbs/${DELPHIX_PDB_NAME}_utlfiledir.txt\""

# Initialise for shell check to succeed
i=0
if [ -z "$UTL_FILE_DIR_PATH" ]; then
    "${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" "mkdir -p ${ORACLE_HOME}/
appsutil/outbound/${CONTEXT_NAME}"
    "${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" "echo ${ORACLE_HOME}/
appsutil/outbound/${CONTEXT_NAME} > ${ORACLE_HOME}/dbs/${DELPHIX_PDB_NAME}
_utlfiledir.txt"
    "${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" "mkdir -p ${ORACLE_BASE}/
temp/${DELPHIX_PDB_NAME}"
    "${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" "echo ${ORACLE_BASE}/temp/$
{DELPHIX_PDB_NAME} >> ${ORACLE_HOME}/dbs/${DELPHIX_PDB_NAME}_utlfiledir.txt"
else
    "${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" "echo $UTL_FILE_DIR_PATH |
awk -F, '{ for (i = 1; i < NF+1; ++i ) print $i }' > ${ORACLE_HOME}/dbs/$
{DELPHIX_PDB_NAME}_utlfiledir.txt"
```

```
fi

"${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" ". ${ORACLE_HOME}/$
{CONTEXT_NAME}.env; perl ${ORACLE_HOME}/appsutil/bin/txkCfgUtlfileDir.pl
-contextfile=${ORACLE_HOME}/appsutil/${CONTEXT_NAME}.xml -oraclehome=$
{ORACLE_HOME} -outdir=${ORACLE_HOME}/appsutil/log -mode=setUtlFileDir <<-EOF1
${APPS_PASSWD}
${SYSTEM_PASSWD}
EOF1
"

"${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" ". ${ORACLE_HOME}/$
{CONTEXT_NAME}.env; perl ${ORACLE_HOME}/appsutil/bin/txkCfgUtlfileDir.pl
-contextfile=${ORACLE_HOME}/appsutil/${CONTEXT_NAME}.xml -oraclehome=$
{ORACLE_HOME} -outdir=${ORACLE_HOME}/appsutil/log -mode=createDirObject <<-EOF2
${APPS_PASSWD}
${SYSTEM_PASSWD}
${DIR_OBJ_PATH}
EOF2
"

"${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" ". ${ORACLE_HOME}/$
{CONTEXT_NAME}.env; perl ${ORACLE_HOME}/appsutil/bin/txkCfgUtlfileDir.pl
-contextfile=${ORACLE_HOME}/appsutil/${CONTEXT_NAME}.xml -oraclehome=$
{ORACLE_HOME} -outdir=${ORACLE_HOME}/appsutil/log -mode=syncUtlFileDir
-skipautoconfig=yes <<-EOF3
${APPS_PASSWD}
EOF3
"
```

19. Set up a Pre-Snapshot hook **Run Bash Shell Command** operation to run any pre-clone steps necessary and specific to your EBS database for a **high privileged user**, as shown in the script below. Normally, these steps will include running Oracle's adpreclone tool.
Pre-snapshot hook for a 19c multi-tenant high privileged user.

```
#!/usr/bin/env bash
#
# Copyright (c) 2022 by Delphix. All rights reserved.
#
# shellcheck source=/dev/null
# As grep has to check all the matches in making decision on L58
# shellcheck disable=SC2143
# NOTE: Ensure the below environment variables will be set up correctly by
# the shell. If not, hardcode or generate the values below.
HOSTNAME=$(uname -n | cut -d '.' -f1)
CONTEXT_NAME=${DELPHIX_PDB_NAME}_${HOSTNAME}
SOURCE_APPS_PASSWD=$DLPX_SOURCE_APPS_PASSWORD_PASSWORD
TARGET_APPS_PASSWD=$DLPX_TARGET_APPS_PASSWORD_PASSWORD
timeout=3600
waittime=0
```

```
ORACLE_USER="oravis"

# Query active PDB services
. "${ORACLE_HOME}/${ORACLE_SID}_${HOSTNAME}.env";
check_value=$(sqlplus -s "/ as sysdba" <<EOF
SELECT NAME FROM v\$active_services;
exit;
EOF
)

checkService() {
if ! echo "$1" | grep -q "$2"; then
. "${ORACLE_HOME}/${ORACLE_SID}_${HOSTNAME}.env";
sqlplus -s "/ as sysdba" <<EOF
alter session set container="${DELPHIX_PDB_NAME}";
BEGIN DBMS_SERVICE.START_SERVICE( service_name => '$2');
end;
/
EOF
fi
}

# Check for ebs_PDB service
checkService "$check_value" "ebs_${DELPHIX_PDB_NAME}"
# Check for PDB_ebs_patch service
checkService "$check_value" "${DELPHIX_PDB_NAME}_ebs_patch"

. "${ORACLE_HOME}/${CONTEXT_NAME}.env";

testAppsPassword() {
    local passwordInQuestion=$1
    ERROR=$(sqlplus "apps/${passwordInQuestion}@${DELPHIX_PDB_NAME}" <<<
"exit;")
    grep ORA-01017 <<< "${ERROR}" >/dev/null && return 1
    return 0
}

testAppsPassword "${SOURCE_APPS_PASSWD}"
testResult=$?

if [[ ${testResult} == 0 ]]; then
    APPS_PASSWD="${SOURCE_APPS_PASSWD}"
else
    APPS_PASSWD="${TARGET_APPS_PASSWD}"
fi

while [[ -f "${ORACLE_HOME}/.delphix_adpreclone.lck" || $(pgrep -au "$
{ORACLE_USER}" | grep "^${ORACLE_HOME}" | grep -v grep | grep "adpreclone") ]]
; do
    if [[ $waittime -gt $timeout ]]; then
        echo "Another adpreclone process is still running from last 60
mins.Delphix cannot proceed until it is complete. Exiting Now."
```

```
        exit 1
    fi
    echo " Another adpreclone process is running. waiting for the process to
complete before starting the adpreclone of the database...."
    (( timeout += 10 ))
    sleep 10
done
echo "No other adpreclone process is running on the database, proceeding ...."
"${ORACLE_HOME}"/perl/bin/perl "${ORACLE_HOME}/appsutil/scripts/$
{CONTEXT_NAME}/adpreclone.pl" dbTier <<-EOF
${APPS_PASSWD}
EOF
```

20. Pre-Snapshot hook **Run Bash Shell Command** operation to run any pre-clone steps necessary and specific to your EBS database for a **low privileged user**, as shown in the script below.

    Pre-snapshot hook for a 19c multi-tenant low privileged user

```
#!/usr/bin/env bash
#
# Copyright (c) 2022 by Delphix. All rights reserved.
#
# shellcheck source=/dev/null
# As grep has to check all the matches in making decision on L58
# shellcheck disable=SC2143
# NOTE: Ensure the below environment variables will be set up correctly by the
# shell. If not, hardcode or generate the values below.
DLPX_DB_EXEC_SCRIPT="<Remote BIN location till dlpx_db_exec script>"
DLPX_PRIV_USER=oravis
HOSTNAME=$(uname -n | cut -d '.' -f1)
CONTEXT_NAME=${DELPHIX_PDB_NAME}_${HOSTNAME}
SOURCE_APPS_PASSWD=$DLPX_SOURCE_APPS_PASSWORD_PASSWORD
TARGET_APPS_PASSWD=$DLPX_TARGET_APPS_PASSWORD_PASSWORD
timeout=3600
waittime=0

# Query active PDB services
. "${ORACLE_HOME}/${ORACLE_SID}_${HOSTNAME}.env";
check_value=$(sqlplus -s "/ as sysdba" <<EOF
SELECT NAME FROM v\$active_services;
exit;
EOF
)

checkService() {
if ! echo "$1" | grep -q "$2"; then
. "${ORACLE_HOME}/${ORACLE_SID}_${HOSTNAME}.env";
sqlplus -s "/ as sysdba" <<EOF
alter session set container="${DELPHIX_PDB_NAME}";
```

```
BEGIN DBMS_SERVICE.START_SERVICE( service_name => '$2');
end;
/
EOF
fi
}

# Check for ebs_PDB service
checkService "$check_value" "ebs_${DELPHIX_PDB_NAME}"
# Check for PDB_ebs_patch service
checkService "$check_value" "${DELPHIX_PDB_NAME}_ebs_patch"

. "${ORACLE_HOME}/${CONTEXT_NAME}.env";
testAppsPassword() {
    local passwordInQuestion=$1
    ERROR=$("${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" ". ${ORACLE_HOME}/$
{CONTEXT_NAME}.env; sqlplus apps/\"${passwordInQuestion}\"@${DELPHIX_PDB_NAME}
<<< \"exit;\"")
    grep ORA-01017 <<< "${ERROR}" >/dev/null && return 1
    return 0
}
testAppsPassword "${SOURCE_APPS_PASSWD}"
testResult=$?

if [[ ${testResult} == 0 ]]; then
    APPS_PASSWD=${SOURCE_APPS_PASSWD}
else
    APPS_PASSWD=${TARGET_APPS_PASSWD}
fi
while [[ -f "${ORACLE_HOME}/.delphix_adpreclone.lck" || $(pgrep -au "$
{DLPX_PRIV_USER}" | grep "^${ORACLE_HOME}" | grep -v grep | grep "adpreclone")
]] ; do
    if [[ $waittime -gt $timeout ]]; then
        echo "Another adpreclone process is still running from the last 60
mins.Delphix cannot proceed until it is complete. Exiting Now."
        exit 1
    fi
    echo " Another adpreclone process is running. waiting for the process to
complete before starting the adpreclone of the database...."
    (( timeout += 10 ))
    sleep 10
done
echo "No other adpreclone process is running on the database, proceeding ...."
${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" ". ${ORACLE_HOME}/$
{CONTEXT_NAME}.env; ${ORACLE_HOME}/perl/bin/perl ${ORACLE_HOME}/appsutil/
scripts/${CONTEXT_NAME}/adpreclone.pl dbTier <<-EOF
${APPS_PASSWD}
EOF
"
```

9.8.6.3.5 Provisioning the Oracle 12C database

9.8.6.3.5.1 Procedure

Perform the following steps to provision Oracle 12c database VDB.

1. Refresh the Target DB environment from **Manage > Environments** so that the Listener brought up during DBTechstack provision gets discovered and the Installation Home to be listed in dropdown during Database provision. If the Target DB environment is not refreshed, you will see `This environment has no compatible Oracle Installation Homes` in the Installation Home dropdown in the **Database provision**.
2. Provision the EBS database to the target dbTier environment by following the steps outlined in Provisioning an Oracle VDB[506]. **Note:** When Snapshot is running against the DBTechstack, database, or AppsTier, the Delphix Continuous Data Engine also executes pre-clone logic to ensure the latest configuration is staged in the captured snapshots. Unfortunately, if multiple Snapshots are running against the same EBS instance concurrently, this pre-clone logic may fail and produce bad snapshots.
   To avoid SnapSync conflicts, spread out your SnapSync policies for an EBS instance by one hour or more.
3. Select the correct **Installation Home**.
   This should be the virtual DBTechstack you just added to the Delphix Continuous Data Engine.
4. Select an Environment User.
5. This user should be the oracle user-outlined in Preparing Target EBS R12.2 Environments for Provisioning.
6. Select a Target Group for the VDB.
7. In the VDB **Configuration** section, provide Target DB/CDB SID name provided during DBTechstack provisioning in the **Oracle Database Name** and **Oracle SID** field. Provide any unique user-defined name in **VDB Name** and **Oracle Database Unique Name** and click **Next**.
8. Click **Advanced**.
9. Select the correct **Oracle Node Listeners** value.
   This should be the listener corresponding to the virtual DBTechstack you just added to the Delphix Continuous Data Engine.
10. Add the EBS R12.2 dbTier environment file as a **Custom Environment Variables** entry.
11. This file can be specified as an Environment File with Path Parameters of `$ORACLE_HOME/<CONTEXT_NAME>` .
    Replace < `CONTEXT_NAME` > with the virtual EBS instance's context name. The Delphix Continuous Data Engine will expand the $ `ORACLE_HOME` variable at runtime.

506 https://cd.delphix.com/docs/latest/provisioning-an-oracle-vdb

```
. "${ORACLE_HOME}/${ORACLE_SID}_${HOSTNAME}.env";
sqlplus -s "/ as sysdba" <<EOF
alter system set SEC_CASE_SENSITIVE_LOGON=FALSE scope=both;
EOF
```

For more information, see Customizing Oracle VDB Environment Variables[507].

> **Note:** To complete the DB Provisioning operation, you must provide content to configure clones and pre-snapshot hooks. You must either follow step 12 to use the hooks utility feature OR proceed with the conventional way by following steps 13-20.

12. To complete the DB Provisioning operation, you must provide content to configure clones and pre-snapshot hooks. The following are the content for both the hooks:
    a. Configure Clone hook will be `<DBTechstack Mount point>/hooksUtil/hooksRunner --operation configure` . You can set a hooks environment variable `DLPX_SOURCE_APPS_PASSWORD` for providing the source apps schema password.
    b. Pre-Snapshot hook : `<DBTechstack Mount point>/hooksUtil/hooksRunner --operation pre-snapshot` . You can set a hooks environment variable `DLPX_SOURCE_APPS_PASSWORD` for providing the source apps schema password. In case of apps password change in target VDB, you can set hook environment variables `DLPX_TARGET_APPS_PASSWORD` for providing the target apps schema password.

> **Note**: It is mandatory to set a hooks environment variable `DLPX_SOURCE_APPS_PASSWORD` for providing the source apps schema password. In case the apps password change is required in target VDB, you can set hook environment variables `DLPX_TARGET_APPS_PASSWORD` for providing the target apps schema password in respective hooks. The variables will be declared in the Credential **Environment Variables** hook section.

The hooks environment variable will be declared as, `DLPX_SOURCE_APPS_PASSWORD` and then its password value will be the same variable used in hooks, along with the `PASSWORD` keyword appended at the end so that it will be like `DLPX_SOURCE_APPS_PASSWORD_PASSWORD`
We assign the value of $ `DLPX_SOURCE_APPS_PASSWRD` to `APPS_PASSWD` .
In hooks, it is written as:

507 https://cd.delphix.com/docs/latest/customizing-oracle-vdb-environment-variables

```
APPS_PASSWD=$DLPX_SOURCE_APPS_PASSWORD_PASSWORD
```

13. for `DLPX_DB_EXEC_SCRIPT` from all the given hooks according to the path that exists in your environment. Please replace the following lines from hook scripts accordingly.

```
DLPX_DB_EXEC_SCRIPT="<Remote BIN location till dlpx_db_exec script>"
DLPX_PRIV_USER=<replace with your db user> eg:oravis
```

14. For the EBS database, add a **Run Bash Shell Command** operation to the Configure Clone hook to ensure that sqlnet.ora or sqlnet_ifile.ora specify a value for SQLNET.ALLOWED_LOGON_VERSION_SERVER for a **high privileged user**, as shown in the script below.
This requirement is outlined in **Cloning Oracle E-Business Suite Release 12.2 with Rapid Clone (Doc ID 1383621.1)** found at http://docs.oracle.com[508]
Configure clone second hook for a high privileged user.

```
#!/usr/bin/env bash
#
# Copyright (c) 2022, 2024 by Delphix. All rights reserved.
#
# shellcheck source=/dev/null
# NOTE: Ensure the below environment variables will be set up correctly by
# the shell. If not, hardcode or generate the values below.
# If you are using sqlnet_ifile.ora, change the script below to reflect
# sqlnet_ifile.ora Initialize dummy variable `parameter` to avoid shellcheck
# to fail
CONTEXT_NAME=${ORACLE_SID}_$(uname -n | cut -d '.' -f1)
. "${ORACLE_HOME}/${CONTEXT_NAME}.env";
TNS_ADMIN=${ORACLE_HOME}/network/admin/${CONTEXT_NAME}
FILE="${TNS_ADMIN}/sqlnet.ora"

if [[ ! -f "${FILE}" ]];  then
     {
    echo "#Creating sqlnet.ora file for Delphix configure clone hook"
    echo "##############################################################"
    echo "NAMES.DIRECTORY_PATH=(TNSNAMES, ONAMES, HOSTNAME)"
    echo "SQLNET.EXPIRE_TIME=10"
    echo "SQLNET.INBOUND_CONNECT_TIMEOUT=60"
    echo "SQLNET.ALLOWED_LOGON_VERSION_SERVER=8"
    } >> "${TNS_ADMIN}/sqlnet.ora"
fi
echo "sqlnet.ora is created at location ${TNS_ADMIN}"

check_value=$(sqlplus -s "/ as sysdba" <<EOF
set head off termout off feedback off wrap off
```

508 https://docs.oracle.com/en/

```
show parameter sec_case_sensitive_logon;
EOF
)

TNS_ADMIN=${ORACLE_HOME}/network/admin
check_value=$(echo "${check_value}" | awk '{{print $3}}'| xargs)
echo "sec_case_sensitive_logon initialization parameter value in the database
is : ${check_value}"

if [[ ${check_value} = "FALSE" ]]; then
    sed 's/^SQLNET.ALLOWED_LOGON_VERSION_SERVER=.*/
SQLNET.ALLOWED_LOGON_VERSION_SERVER=8/g' "${TNS_ADMIN}/sqlnet.ora" > /var/tmp/
temp.ora && mv /var/tmp/temp.ora "${TNS_ADMIN}/sqlnet.ora"
elif [[ ${check_value} = "TRUE" ]]; then
    sed 's/^SQLNET.ALLOWED_LOGON_VERSION_SERVER=.*/
SQLNET.ALLOWED_LOGON_VERSION_SERVER=10/g' "${TNS_ADMIN}/sqlnet.ora" > /var/tmp/
temp.ora && mv /var/tmp/temp.ora "${TNS_ADMIN}/sqlnet.ora"
else
    echo "sec_case_sensitive_logon parameter is not set in the database. So the
sqlnet.ora has not been updated."
fi
```

15. Configure Clone hook to ensure that sqlnet.ora or sqlnet_ifile.ora specify a value for SQLNET.ALLOWED_LOGON_VERSION_SERVER for a **low privileged user**, as shown in the script below.
    Configure clone second hook for a low privileged user.

```
#!/usr/bin/env bash
#
# Copyright (c) 2022, 2024 by Delphix. All rights reserved.
#
# shellcheck source=/dev/null
# NOTE: Ensure the below environment variables will be set up correctly by the
# shell. If not, hardcode or generate the values below.
# If you are using sqlnet_ifile.ora, change the script below to reflect
# sqlnet_ifile.ora.
# Initialize dummy variable `parameter` to avoid shellcheck to fail
DLPX_DB_EXEC_SCRIPT="<Remote BIN location for dlpx_db_exec script>"
DLPX_PRIV_USER=oravis
CONTEXT_NAME=${ORACLE_SID}_$(uname -n | cut -d '.' -f1)
TNS_ADMIN=${ORACLE_HOME}/network/admin/${CONTEXT_NAME}
FILE="${TNS_ADMIN}/sqlnet.ora"

if [[ ! -f "${FILE}" ]];  then
     {
    echo "#Creating sqlnet.ora file for Delphix configure clone hook"
    echo "##############################################################"
    echo "NAMES.DIRECTORY_PATH=(TNSNAMES, ONAMES, HOSTNAME)"
    echo "SQLNET.EXPIRE_TIME=10"
    echo "SQLNET.INBOUND_CONNECT_TIMEOUT=60"
```

```
    echo "SQLNET.ALLOWED_LOGON_VERSION_SERVER=8"
    } >> "${TNS_ADMIN}/sqlnet.ora"
fi
echo "sqlnet.ora is created at location ${TNS_ADMIN}"

check_value=$("${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" ". ${ORACLE_HOME}/
${CONTEXT_NAME}.env; sqlplus \"/ as sysdba\" <<-EOF
set head off termout off feedback off wrap off
show parameter sec_case_sensitive_logon;
EOF
")

TNS_ADMIN=${ORACLE_HOME}/network/admin
check_value=$(echo "${check_value}" | awk '{{print $3}}'| xargs)
echo "sec_case_sensitive_logon initialization parameter value in the database
is : ${check_value}"

if [[ ${check_value} = "FALSE" ]]; then
    "${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" "sed 's/
^SQLNET.ALLOWED_LOGON_VERSION_SERVER=.*/SQLNET.ALLOWED_LOGON_VERSION_SERVER=8/
g' \"${TNS_ADMIN}/sqlnet.ora\" > /var/tmp/temp.ora && mv /var/tmp/temp.ora \"$
{TNS_ADMIN}/sqlnet.ora\""
elif [[ ${check_value} = "TRUE" ]]; then
    "${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" "sed 's/
^SQLNET.ALLOWED_LOGON_VERSION_SERVER=.*/SQLNET.ALLOWED_LOGON_VERSION_SERVER=10/
g' \"${TNS_ADMIN}/sqlnet.ora\" > /var/tmp/temp.ora && mv /var/tmp/temp.ora \"$
{TNS_ADMIN}/sqlnet.ora\""
else
    echo "sec_case_sensitive_logon parameter is not set in the database. So the
sqlnet.ora has not been updated."
fi
```

16. Add a **Run Bash Shell Command** operation to the Configure Clone hook to ensure that `adcfgclone.pl script` is run against the newly provisioned database for a **high privileged user**, as shown in the script below.

    Note: You must set a hooks environment variable `DLPX_SOURCE_APPS_PASSWORD` for providing the source apps schema password. In case the apps password change is required in target VDB, you can set hook environment variables `DLPX_TARGET_APPS_PASSWORD` and `DLPX_SYS_PASSWORD` for providing the target apps schema and system schema passwords.

    Configure clone first hook for a high privileged user

```
#!/usr/bin/env bash
#
# Copyright (c) 2022 by Delphix. All rights reserved.
#
# shellcheck source=/dev/null
# NOTE: Ensure the below environment variables will be set up correctly by
# the shell. If not, hardcode or generate the values below.
```

```
HOSTNAME=$(uname -n | cut -d '.' -f1)
CONTEXT_NAME=${ORACLE_SID}_${HOSTNAME}
APPS_PASSWD=$DLPX_SOURCE_APPS_PASSWORD_PASSWORD

. "${ORACLE_HOME}/${CONTEXT_NAME}.env";
# Check for local_listener parameter is set for SID, otherwise set it
# appropriately
check_value=$(sqlplus -s "/ as sysdba" <<EOF
show parameter local_listener;
EOF
)
local_listener=$(echo "$check_value" | awk '{print $11}')
value=("${local_listener//:/ }")
host="${value[0]}"
port="${value[1]}"
curr_port=$(grep PORT < "${TNS_ADMIN}/listener.ora" | awk '{print $9}' | sed
's/)//g')

if [[ $port != "$curr_port" || $host != "${HOSTNAME}" ]]; then
sqlplus -s "/ as sysdba" <<EOF
alter system set local_listener='${HOSTNAME}:${curr_port}';
alter system register;
EOF
fi

sqlplus "/ as sysdba" <<EOF
@${ORACLE_HOME}/appsutil/install/${CONTEXT_NAME}/adupdlib.sql so
EOF

perl "${ORACLE_HOME}/appsutil/clone/bin/adcfgclone.pl" dbconfig "$
{ORACLE_HOME}/appsutil/${CONTEXT_NAME}.xml" <<EOF
${APPS_PASSWD}
EOF
```

17. Configure Clone hook to ensure that adcfgclone is run against the newly provisioned database for a **low privileged user**, as shown in the script below.Customers have to identify the DLPX_DB_EXEC_SCRIPT value from their remote environment, this would always be inside the toolkit directory provided while adding the environment on Delphix Continuous Data Engine.
Configure clone first hook for a low privileged user.

```
#!/usr/bin/env bash
#
# Copyright (c) 2022 by Delphix. All rights reserved.
#
# shellcheck source=/dev/null
# NOTE: Ensure the below environment variables will be set up correctly by the
# shell. If not, hardcode or generate the values below.

DLPX_DB_EXEC_SCRIPT="<Remote BIN location for dlpx_db_exec script>"
DLPX_PRIV_USER=oravis
```

```
APPS_PASSWD=$DLPX_SOURCE_APPS_PASSWORD_PASSWORD
HOSTNAME=$(uname -n | cut -d '.' -f1)
CONTEXT_NAME=${ORACLE_SID}_${HOSTNAME}

. "${ORACLE_HOME}/${CONTEXT_NAME}.env";
# Check for local_listener parameter is set for SID, otherwise set it
# appropriately
check_value=$(sqlplus -s "/ as sysdba" <<EOF
show parameter local_listener;
EOF
)
local_listener=$(echo "$check_value" | awk '{print $11}')
value=("${local_listener//:/ }")
host="${value[0]}"
port="${value[1]}"
curr_port=$(grep PORT < "${TNS_ADMIN}/listener.ora" | awk '{print $9}' | sed
's/)//g')

if [[ $port != "$curr_port" || $host != "${HOSTNAME}" ]]; then
sqlplus -s "/ as sysdba" <<EOF
alter system set local_listener='${HOSTNAME}:${curr_port}';
alter system register;
EOF
fi

"${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" ". ${ORACLE_HOME}/$
{CONTEXT_NAME}.env; sqlplus \"/ as sysdba\" <<-EOF
@${ORACLE_HOME}/appsutil/install/${CONTEXT_NAME}/adupdlib.sql so
EOF
"

"${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" ". ${ORACLE_HOME}/$
{CONTEXT_NAME}.env; perl ${ORACLE_HOME}/appsutil/clone/bin/adcfgclone.pl
dbconfig ${ORACLE_HOME}/appsutil/${CONTEXT_NAME}.xml <<-EOF1
${APPS_PASSWD}
EOF1
"
```

18. Configure Clone hook to ensure that the ebs_<*PDB_SID*>listener services do not disappear during a start/stop VDB operation, as shown in the script in point 19.
19. Set up a Pre-Snapshot hook **Run Bash Shell Command** operation to run any pre-clone steps necessary and specific to your EBS database for a **high privileged user**, as shown in the script below. Normally, these steps will include running Oracle's adpreclone tool.
    Pre-snapshot hook for a high privileged user.

```
#!/usr/bin/env bash
#
# Copyright (c) 2023 by Delphix. All rights reserved.
#
```

```
# shellcheck source=/dev/null
# As grep has to check all the matches in making decision on L58
# shellcheck disable=SC2143
# NOTE: Ensure the below environment variables will be set up correctly by
# the shell. If not, hardcode or generate the values below.
HOSTNAME=$(uname -n | cut -d '.' -f1)
CONTEXT_NAME=${ORACLE_SID}_${HOSTNAME}
SOURCE_APPS_PASSWD=$DLPX_SOURCE_APPS_PASSWORD_PASSWORD
TARGET_APPS_PASSWD=$DLPX_TARGET_APPS_PASSWORD_PASSWORD
ORACLE_USER="oravis"

timeout=3600
waittime=0
. "${ORACLE_HOME}/${CONTEXT_NAME}.env";

testAppsPassword() {
    local passwordInQuestion=$1
    ERROR=$(sqlplus "apps/${passwordInQuestion}" <<< "exit;")

    grep ORA-01017 <<< "${ERROR}" >/dev/null && return 1
    return 0

}

testAppsPassword "${SOURCE_APPS_PASSWD}"
testResult=$?

if [[ ${testResult} == 0 ]]; then
    APPS_PASSWD=${SOURCE_APPS_PASSWD}
else
    APPS_PASSWD=${TARGET_APPS_PASSWD}
fi

# Query active SID services
check_value=$(sqlplus -s "/ as sysdba" <<EOF
SELECT NAME FROM v\$active_services;
exit;
EOF
)
checkService() {
if ! echo "$1" | grep -q "$2"; then
. "${ORACLE_HOME}/${CONTEXT_NAME}.env";
sqlplus -s "/ as sysdba" <<EOF
BEGIN DBMS_SERVICE.START_SERVICE( service_name => '$2');
end;
/
EOF
fi
}
# Check for SID_ebs_patch or ebs_patch service based on the AD-TXK codelevel
# version. In AD-TXK Delta 8 and earlier, the service name for connections
# to the patch edition of the database was ebs_patch. In AD-TXK Delta 9, the
```

```
# service name to connect to the patch edition has been changed to
# <instance_name>_ebs_patch
patch_level=$(sqlplus -s "apps/${APPS_PASSWD}" <<EOF
select PATCH_LEVEL from apps.fnd_product_installations where PATCH_LEVEL like
'R12.AD.%';
EOF
)
patch_service="${ORACLE_SID}_ebs_patch"
if [ "${patch_level: -1}" -le 8 ]; then
    patch_service="ebs_patch"
fi
checkService "$check_value" "${patch_service}"


while [[ -f "${ORACLE_HOME}/.delphix_adpreclone.lck" || $(pgrep -au "$
{ORACLE_USER}" | grep "^${ORACLE_HOME}" | grep -v grep |  grep "adpreclone") ]]
 ; do
    if [[ $waittime -gt $timeout  ]]; then
        echo "Another adpreclone process is still running from the last 60
mins.Delphix cannot proceed until it is complete. Exiting Now."
        exit 1
    fi
    echo  " Another adpreclone process is running. waiting for the process to
complete before starting the adpreclone of the database...."
    (( timeout += 10 ))
    sleep 10
done
echo "No other adpreclone process is running on the database, proceeding ...."
"${ORACLE_HOME}"/perl/bin/perl "${ORACLE_HOME}/appsutil/scripts/$
{CONTEXT_NAME}/adpreclone.pl" dbTier <<-EOF
${APPS_PASSWD}
EOF
```

20. Pre-Snapshot hook **Run Bash Shell Command** operation to run any pre-clone steps necessary and specific to your EBS database for a **low privileged user**, as shown in the script below.
Pre-snapshot hook for a low privileged user.

```
#!/usr/bin/env bash
#
# Copyright (c) 2022 by Delphix. All rights reserved.
#
# shellcheck source=/dev/null
# NOTE: Ensure the below environment variables will be set up correctly by the
# shell. If not, hardcode or generate the values below.
# As grep has to check all the matches in making decision on L58
# shellcheck disable=SC2143
DLPX_DB_EXEC_SCRIPT="<Remote BIN location for dlpx_db_exec script>"
DLPX_PRIV_USER=oravis
HOSTNAME=$(uname -n | cut -d '.' -f1)
CONTEXT_NAME=${ORACLE_SID}_${HOSTNAME}
```

```
SOURCE_APPS_PASSWD=$DLPX_SOURCE_APPS_PASSWORD_PASSWORD
TARGET_APPS_PASSWD=$DLPX_TARGET_APPS_PASSWORD_PASSWORD

timeout=3600
waittime=0

testAppsPassword() {
    local passwordInQuestion=$1
    ERROR=$("${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" ". ${ORACLE_HOME}/$
{CONTEXT_NAME}.env; sqlplus apps/\"${passwordInQuestion}\" <<< \"exit;\"")

    grep ORA-01017 <<< "${ERROR}" >/dev/null && return 1
    return 0
}

testAppsPassword "${SOURCE_APPS_PASSWD}"
testResult=$?

if [[ ${testResult} == 0 ]]; then
    APPS_PASSWD=${SOURCE_APPS_PASSWD}
else
    APPS_PASSWD=${TARGET_APPS_PASSWD}
fi

# Query active SID services
. "${ORACLE_HOME}/${CONTEXT_NAME}.env";
check_value=$(sqlplus -s "/ as sysdba" <<EOF
SELECT NAME FROM v\$active_services;
exit;
EOF
)
checkService() {
if ! echo "$1" | grep -q "$2"; then
. "${ORACLE_HOME}/${CONTEXT_NAME}.env";
sqlplus -s "/ as sysdba" <<EOF
BEGIN DBMS_SERVICE.START_SERVICE( service_name => '$2');
end;
/
EOF
fi
}
# Check for SID_ebs_patch or ebs_patch service based on the AD-TXK codelevel
# version. In AD-TXK Delta 8 and earlier, the service name for connections
# to the patch edition of the database was ebs_patch. In AD-TXK Delta 9, the
# service name to connect to the patch edition has been changed to
# <instance_name>_ebs_patch
patch_level=$(sqlplus -s "apps/${APPS_PASSWD}" <<EOF
select PATCH_LEVEL from apps.fnd_product_installations where PATCH_LEVEL like
'R12.AD.%';
EOF
)
patch_service="${ORACLE_SID}_ebs_patch"
```

```
if [ "${patch_level: -1}" -le 8 ]; then
    patch_service="ebs_patch"
fi
checkService "$check_value" "${patch_service}"

while [[ -f "${ORACLE_HOME}/.delphix_adpreclone.lck" || $(pgrep -au "$
{DLPX_PRIV_USER}" | grep "^${ORACLE_HOME}" | grep -v grep |  grep "adpreclone")
 ]] ; do
    if [[ $waittime -gt $timeout  ]]; then
        echo "Another adpreclone process is still running from the last 60
mins.Delphix cannot proceed until it is complete. Exiting Now."
        exit 1
    fi
    echo  " Another adpreclone process is running. waiting for the process to
complete before starting the adpreclone of the database...."
    (( timeout += 10 ))
    sleep 10
done
echo "No other adpreclone process is running on the database, proceeding ...."
"${DLPX_DB_EXEC_SCRIPT}" -u"${DLPX_PRIV_USER}" ". ${ORACLE_HOME}/$
{CONTEXT_NAME}.env; ${ORACLE_HOME}/perl/bin/perl ${ORACLE_HOME}/appsutil/
scripts/${CONTEXT_NAME}/adpreclone.pl dbTier <<-EOF
${APPS_PASSWD}
EOF
"
```

9.8.6.3.6 Provisioning the EBS AppsTier

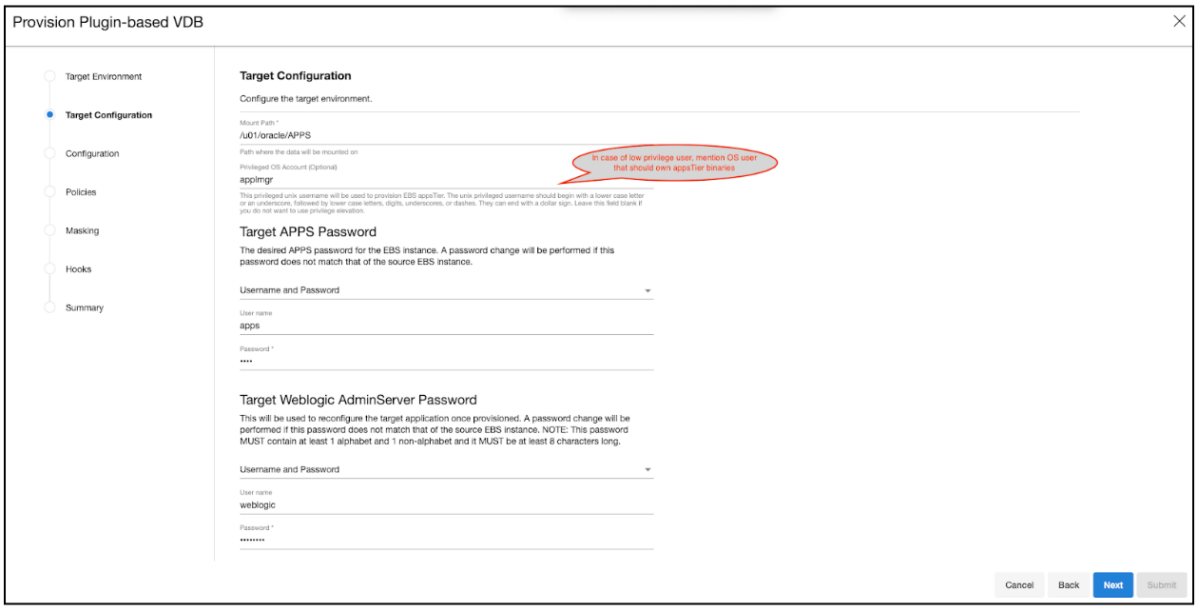
9.8.6.3.6.1 Procedure

Perform the following steps to provision EBS AppsTier VDB.

1. Login to the **Delphix Management** application using **Admin** credentials.
2. Click **Manage**.
3. Select **Datasets**.
4. Select the **AppsTier dSource**.
5. Select a dSource **snapshot**.
   All snapshots will have staged configuration prepared by adpreclone.pl and any hook operations placed on the dSource.
6. Click **Provision**.
   The **Provision VDB** wizard will open.
7. Select an **Environment.**
   This environment will host the virtual AppsTier and be used to execute hook operations specified in a few steps. This environment will also run the WebLogic Admin server (Web Administration service) for the virtual AppsTier.

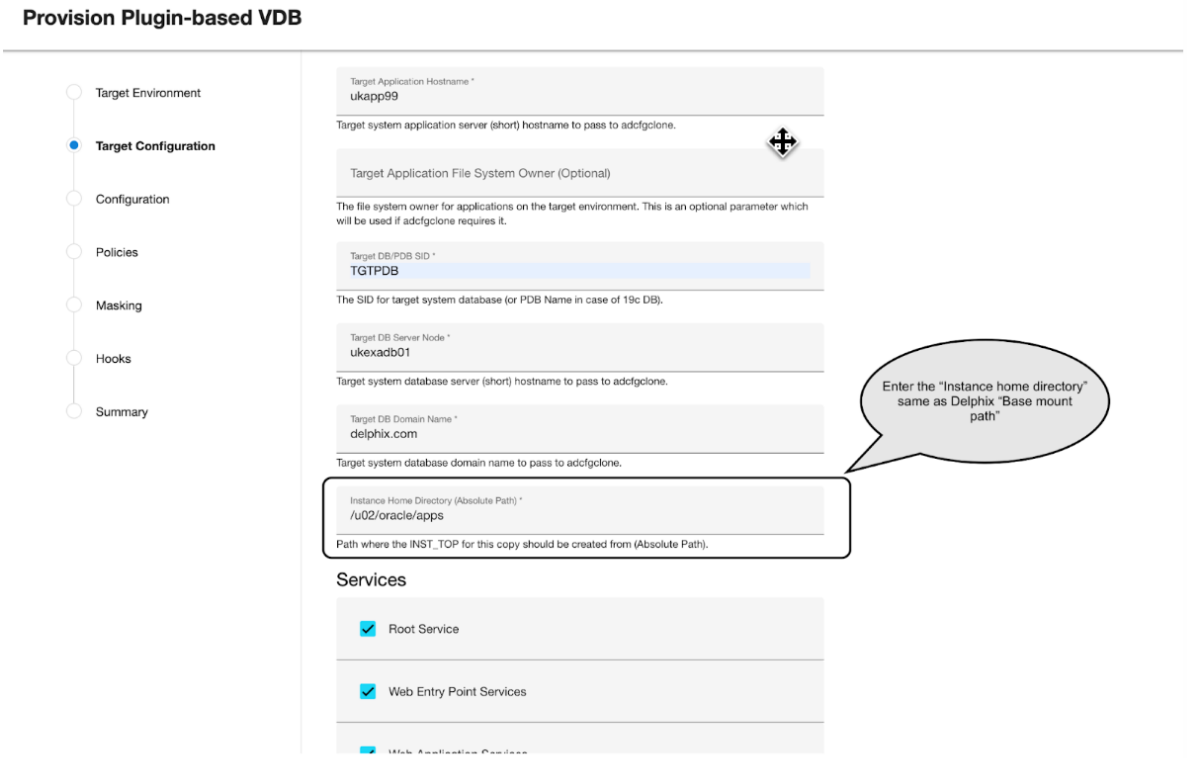
If you are provisioning a multi-node AppsTier, you will be able to specify additional environments to host the virtual AppsTier in a few steps.

8. Select E-Business Suite R12.2 AppsTier from the **Installation** dropdown.
9. Select an Environment User.
   This user should be the applmgr user-outlined in Requirements for Oracle EBS target databases for provisioning (see page 1667)
10. Enter a **Mount Path** for the virtual AppsTier VDB.If you are provisioning a multi-node AppsTier, this mount path will be used across all target environments.
11. Enter the **EBS-specific parameters** for the virtual AppsTier. A subset of these parameters are discussed in more detail below.



12. **Privileged OS Account (Optional)** field should contain a high privileged user when a low privileged user is being used for provisioning.
13. The **Target APPS Password** is the new apps password that is required to configure and manage the virtual AppsTier.
    This password is encrypted when stored within the Delphix Continuous Data Engine and is available as an environment variable to the adcfgclone, adstrtal, and adstpall processes.
14. The **Weblogic AdminServer Password** is the new WebLogic password required to configure the virtual AppsTier. A password change will be performed if this password does not match that of the source EBS instance. This password is encrypted when stored within the Delphix Continuous Data Engine and is available as an environment variable to the adcfgclone, adstrtal, and adstpall processes.
15. The **SYSTEM Password** is required to configure the virtual AppsTier with the new apps password. This password is not required if the Target Apps Password is the same as the source.
16. Provide the Target Application hostname.

17. The **Target DB/PDB SID** is the new database SID (PDB SID in case of 19c) that is required to configure and manage the virtual AppsTier.
18. Provide the Target DB server node hostname.
19. Provide **Target DB Domain Name**, this will be provided to the ***adcfgclone.***
20. Delphix recommends specifying an **Instance Home Directory** under the **Mount Path** so that instance-specific EBS files live on Delphix-provided mount path. Instance Home Directory(Absolute Path) should be the same as Mount Path as mentioned in the first input field.



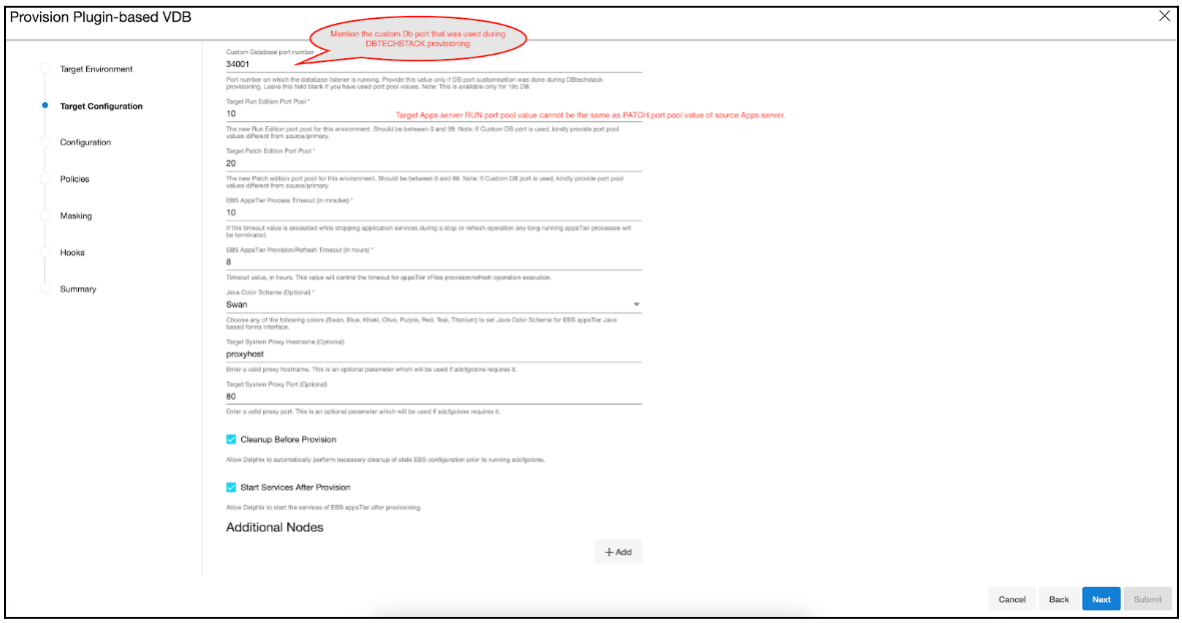
21. For example, if the provided **Mount Path** is /u01/oracle/VIS, then providing an **Instance Home Directory** of /u01/oracle/VIS would allow EBS to generate virtual application INST_TOP in /u01/oracle/VIS/fs1/inst/apps/<CONTEXT_NAME > and /u01/oracle/VIS/fs2/inst/apps/<CONTEXT_NAME >.
    a. If you are provisioning a single-node AppsTier, this recommendation is OPTIONAL; putting instance-specific EBS files on Delphix-provided storage merely eases the administration of the virtual EBS instance.
    b. If you are provisioning a multi-node AppsTier, this recommendation is REQUIRED; the Delphix Continuous Data Engine's automation requires that all nodes in the AppsTier have access to instance-specific files via Delphix-provided storage.
22. Ensure that the **Target Application Hostname** and **Target DB Server Node** values are the short hostnames, not the fully-qualified hostnames.

23. Select **Services**, by default all the ***Root***, ***Web Entry Point***, ***Web Application*** and ***Batch Processing*** would be checked. In case you are using a multi-node environment and want to start a few services on a secondary node then this field is going to be very useful for you.
24. Provide **DISPLAY Variable**, the default value is hostname:0.0
    For example - ukapps99.delphix.com:0.0
25. [Optional]Provide the **Custom Database Port Number** on which the database listener must run.

> **Note:** Enter the numeric port value in this field only if you want to use DB port customisation, and this value should match with the custom database port number given during the DBTechstack provisioning. Leaving this field blank automatically uses the default port based on the port pool. The custom database port number is applicable only for Oracle 19c database.

26. Provide **Target Run Edition Port Pool**, the value of this field should be 0 to 99. This should **match** with the **port pool value** given during the **DBTechstack provisioning** if the Custom **Database Port Number** input field is left blank.
    Provide **Target Patch Edition Port Pool**, the value of this field should be between 0 to 99.
    Target Run Edition Port Pool and **Target Patch Edition Port Pool** ranges shouldn't intersect with each other.
    For example: if **Run Edition Port Pool** is 9 then the **patch FS port pool value** can be anything between 0 to 99 except 9.

> **Note:** In case, the database listener is running on a **Custom Database Port Number** then Target Apps server's **RUN Edition Port Pool** and **PATCH Edition Port Pool** value cannot have the same port pool value of source Application server.



27. The **EBS AppsTier Timeout** is required to terminate all the long-running AppsTier processes which have exceeded the timeout value when stopping the applications as part of a refresh. This timeout will be calculated in minutes. For example, if set to 30, then we run adstpall at the start of the refresh, and if after 30 minutes the application has not stopped, then the processes will be terminated to allow the refresh to continue.
28. The **EBS AppsTier Provision/Refresh Timeout** is required to terminate configure and cloning via adcfgclone process which have exceeded the timeout as part of a provision/refresh. This timeout will be calculated in hours and the default value will be 8 hours. For example, if set to 9, then the connector will detect a timeout after 9 hours instead of getting in a hung state and UI error for the same will be prompted.
29. The **Java Color Scheme** is an **optional** parameter and is used to change the JAVA color for provisioned Oracle EBS JAVA based form interface. Please choose any of the following colors (Swan, Blue, Khaki, Olive, Purple, Red, Teal, Titanium) to set Java Color Scheme for EBS AppsTier java-based forms interface.
30. Provide **Target System Proxy Hostname (Optional)**, This is an optional parameter that will be used if ***adcfgclone*** requires it.
31. Provide **Target System Proxy Port (Optional)**, This is an optional parameter that will be used if ***adcfgclone*** requires it.
32. Enable the **Cleanup Before Provision Option** if you want to permit the Delphix Continuous Data Engine to automatically clean up stale EBS configuration during a refresh. This option is recommended, but only available if your Oracle Home is patched with Oracle Universal Installer (OUI) version 10.2 or above.
    a. With this option enabled, the Delphix Continuous Data Engine will inspect the target environment's oraInventory prior to refreshing this virtual AppsTier. If any Oracle Homes are already registered within the specified **Mount Path**, the Delphix Continuous Data Engine will detach them from the inventory prior to running adcfgclone. These homes must be detached/unregistered prior to running post-clone configuration. If they are not detached/unregistered,

adcfgclone will fail, citing conflicting oraInventory entries as an issue. The Delphix Continuous Data Engine will also remove any conflicting INST_TOP directories left in the environment. Non-conflicting INST_TOP directories will not be modified.

b. Without this option enabled, Oracle Homes or INST_TOP directories that conflict with the specified **Mount Path** or desired INST_TOP location will be reported in errors instead of automatically cleaned up. For refresh to succeed, you must manually detach/register conflicting Oracle Homes and manually remove conflicting INST_TOP directories prior to refresh.

33. Enable the **Start Services After Provision** option if you want to permit the Delphix Continuous Data Engine to automatically start the services for EBS AppsTier after provisioning.
    a. With this option enabled, the Delphix Continuous Data Engine will start all the services of EBS AppsTier after provisioning.
    b. Without this option enabled, the Delphix Continuous Data Engine will **NOT** start the services of EBS AppsTier after provisioning. This will allow customers to prevent EBS services from being started at the conclusion of provisioning or refreshing. That way, customers can perform post-clone processing automation using a configure-clone hook without having to stop services first.
34. If you are provisioning a multi-node AppsTier, enter additional AppsTier nodes as **Additional Nodes**.
    a. Select the Environment for the secondary node.
    b. The **Environment User** for each node should be the applmgr user-outlined in Requirements for Oracle EBS target databased for provisioning (see page 1667)
    c. Ensure that the **Hostname** value for each node is the short hostname, not the fully-qualified hostname.
    d. Provide DISPLAY Variable, the default value is hostname:0.0.
35. The **Mount Path** is not configurable for each node individually. The **Mount Path** provided for the primary environment will be used for each additional node.
36. Click **Next**.
37. Enter a **VDB Name**.
38. Select a **Target Group** for the VDB.
    If necessary, click the **Plus** icon to add a new group.
39. (Optional)When provisioning a new VDB in the VDB Provisioning wizard, check the Auto VDB Restart box. Reference - Automatic VDB restart on target server after reboot[509]
40. Select a **Snapshot Policy** for the VDB.
    If necessary, click the **Plus** icon to create a new policy.

Note: When Snapshot is running against the DBTechstack, database, or AppsTier, the Delphix Continuous Data Engine also executes pre-clone logic to ensure the latest configuration is staged in the captured snapshots. Unfortunately, if multiple Snapshots are running against the

509 https://cd.delphix.com/docs/latest/automatic-vdb-restart-on-target-server-after-reboo

same EBS instance concurrently, this pre-clone logic may fail and produce bad snapshots.
To avoid SnapSync conflicts, spread out your SnapSync policies for an EBS instance by one hour or more.

41. Click **Next**.
42. Enter any **custom hook operations** that are needed to help correctly manage the virtual AppsTier.
    The Configure Clone hook will be run after the adcfgclone.pl tool has both mounted and configured the AppsTier.
    All hook operations run against the environment specified for provision. For a multi-node AppsTier, hook operations never run against additional nodes specified.
43. Click **Next.**
44. Click **Submit**.

Warning: dbTier must be accessible during AppsTier provisioning
Info: Post-clone configuration will fail if the AppsTier cannot connect to the database. Ensure the target dbTier is accessible to the AppsTier during the provisioning process. Ensure both the virtual database and database listener are running.

Note: If your source setup is EBS ISG enabled, then during virtual AppsTier creation on the target server, perform post cloning steps as described in Installing Oracle E-Business Suite Integrated SOA Gateway, Release 12.2[510] (Doc ID 1311068.1).

45. When provisioning starts, you can review the progress of the job in the **Datasets** panel, or in the **Job History** panel of the **Dashboard**. When provisioning is complete, the AppsTier VDB will be included in the group you designated and listed in the **Datasets** panel. If you select the AppsTier VDB in the **Datasets** panel and click the **Configuration** tab, you can view information about the virtual files and its Data Management settings.
46. For tips on monitoring the progress of AppsTier provisioning, see Monitoring EBS R12.2 AppsTier Provisioning Progress[511]
    Once all three EBS virtual datasets have been provisioned successfully, your virtual EBS instance should be running and accessible.

510 https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=225107018848416&parent=DOCUMENT&sourceId=2293004.1&id=1311068.1&_afrWindowMode=0&_adf.ctrl-state=o0lz4uv6d_53#clone
511 https://cd.delphix.com/docs/latest/monitoring-ebs-r12-2-appstier-provisioning-progres

Note: AppsTier topology Is configurable. The AppsTier topology of the virtual EBS instance does NOT need to match the AppsTier topology of the source EBS instance. The Delphix Continuous Data Engine will automate scale up or scale down logic as required during provisioning.

9.8.6.4 Refreshing and rewinding a TDE-enabled vPDB for Oracle EBS

Reference Refreshing and rewinding a TDE-enabled vpdb[512]

9.8.6.4.1 Rewind operation on snapshot

Perform the rewind operation on the snapshot after AppsTier provisioning in the following sequence.

1. Take a snapshot of the Database VDB after the AppsTier provision.
2. Stop the AppsTier, DB, and DBTechstack one by one.
3. Rewind DBTechstack, DB, and AppsTier one by one.

If the DB snapshot was taken before AppsTier provisioning, rewinding the DB will cause a loss of new data updated by AppsTier during provisioning and the next operation of AppsTier rewind will fail. So, to ensure that the Rewind operation of AppsTier succeeds, take a snapshot of DB after AppsTier provision.

9.8.6.5 Starting and stopping a virtual EBS instance

This topic describes the process of starting and stopping a virtual Oracle E-Business Suite (EBS) instance.

You can start and stop virtual EBS instances through the **Delphix Management** application or through the standard Oracle command-line interface (CLI) utilities, `adstrtal` and `adstpall`. The Delphix Engine will show the dbTechStack and appsTier as running as long as there are processes using the dbTechStack and appsTier filesystem mounts on the target environments.

Be careful! Services running on the appsTier depend on the availability of services on the dbTier. The steps below are explicitly ordered with these dependencies in mind. Executing steps out of order may lead to errors in accessing the virtual EBS instance.

9.8.6.5.1 Stopping

1. Login to the **Delphix Management** application.
2. Click **Manage**.

512 https://cd.delphix.com/docs/latest/refreshing-and-rewinding-a-tde-enabled-vpdb

3. Select **Datasets**.
4. Select the **appsTier vFiles** for your EBS instance.
5. Click the **Stop** icon.

   Stopping the appsTier will run Oracle's `adstpall.sh` utility. For multi-node appsTier, secondary nodes will be stopped sequentially, followed by the primary node.
   Note: Stopping the appsTier may take a long time. The Delphix Engine will wait for all Oracle application processes to exit before declaring the appsTier as stopped.
6. Select the **VDB** utilized by your EBS instance.
7. Click the **Stop** icon. This action will shutdown the database instance.
8. Select the **dbTechStack vFiles** hosting your virtual EBS database.
9. Click the **Stop** icon. Stopping the dbTechStack will shutdown the database listener.

9.8.6.5.2 Starting

1. Login to the **Delphix Management** application using **Delphix Admin** credentials.
2. Click **Manage**.
3. Select **Datasets**.
4. Select the **dbTechStack vFiles** hosting your virtual EBS database.
5. Click the **Start** icon.

   Starting the dbTechStack will start the database listener.
6. Select the **VDB** utilized by your EBS instance.
7. Click the **Start** icon. Starting the database will open the database.
8. Select the **appsTier vFiles** for your EBS instance.
9. Click the **Start** icon. Starting the appsTier will run Oracle's `adstrtal.sh` utility. For multi-node appsTier, the primary node will be started first, followed by secondary nodes sequentially.

9.8.6.6 Rewinding a virtual EBS instance

This topic describes the process of rewinding a virtual Oracle E-Business Suite (EBS) instance.

9.8.6.6.1 Rewind Rules

To Rewind an EBS Environment, rewind has to be done in the following order for each dataset -

1. Rewind all tiers:

   a. Disable/Stop AppsTier VDB
   b. Disable/Stop Database tier VDB
   c. Rewind DBTechStack
   d. Rewind DB tier
   e. Rewind Appstier

2. Rewind of Database only:
   a. Disable/Stop AppsTier VDB
   b. Rewind DB tier
   c. Rewind Appstier

3. Rewind of AppsTier needs to be handled with care. Rewinding Appstier should be only done to a snapshot of AppsTier compatible with the database snapshot. It is not recommended to perform only AppsTier rewind.

To find a compatible AppsTier snapshot with Database snapshot - Check the timestamps of AppsTier snapshot & Database snapshot from timeflow UI. AppsTier & database snapshot timing should be close.

> Changes applied to EBS and picked up only in some dSource snapshots may make certain combinations of snapshots across the appsTier and dbTier incompatible. When provisioning, refreshing or rewinding a virtual EBS instance, be sure the points in time you select for each dataset are compatible with each other.

> Be careful! Services running on the appsTier depend on the availability of services on the dbTier. The steps below are explicitly ordered with these dependencies in mind. Executing steps out of order may lead to errors in accessing the virtual EBS instance.

9.8.6.6.2 Prerequisites

The appsTier **Instance Home Directory** of the virtual EBS instance must reside under the specified **Mount Path**.

9.8.6.6.3 Procedure

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Datasets**.
4. Select the **appsTier vFiles** for your EBS instance.
5. Click the **Stop** icon to shutdown the appsTier services.
6. Select the **VDB** utilized by your EBS instance.

7. Click the **Stop** icon to shutdown the database.
8. Select the **dbTechStack vFiles** hosting your virtual EBS database.
9. Click the **Stop** icon to shutdown the database listener.
10. Rewind the dbTechStack vFiles.
    a. Select a **snapshot**.
    b. Click the **Rewind** button.
11. Rewind the EBS VDB.
    a. Select a **snapshot**.
    b. Click the **Rewind** button.
12. Rewind the appsTier vFiles.
    a. Select a **snapshot**.
    b. Click the **Rewind** button.

Once you have rewound all three EBS virtual datasets successfully, your virtual EBS instance should be running and accessible.

9.8.6.7 Refreshing a virtual EBS instance

This topic describes the process of refreshing a virtual Oracle E-Business Suite (EBS) instance.

> Changes applied to EBS and picked up only in some dSource snapshots may make certain combinations of snapshots across the appsTier and dbTier incompatible. When provisioning, refreshing, or rewinding a virtual EBS instance, be sure the points in time you choose for each dataset are compatible with each other.

> Be careful! Services running on the appsTier depend on the availability of services on the dbTier. The steps below are explicitly ordered with these dependencies in mind. Executing steps out of order may lead to errors in accessing the virtual EBS instance.

9.8.6.7.1 Procedure

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Datasets**
4. Select the **appsTier vFiles** for your EBS instance.
5. Click the **Stop** icon to shut down the appsTier services.

6. Select the **VDB** utilized by your EBS instance.
7. Click the **Stop** icon to shut down the database.
8. Select the **dbTechStack vFiles** hosting your virtual EBS database.
9. Click the **Stop** icon to shut down the database listener.

> **Warning:** If you did NOT specify the **Cleanup Before Provision** option for either your virtual dbTechStack or appsTier, you must manually clean up your target environments prior to refreshing. If you have specified this option for both datasets, no manual work is required.

To manually clean up a target environment before the refresh, remove instance-specific directories and or inventory entries that will conflict with files and entries recreated during the refresh. Without this clean-up, post-clone configuration performed during a refresh will fail with an error claiming that a conflicting EBS instance is already installed.

10. Refresh the dbTechStack vFiles.
    a. On the vFiles, click the **Refresh**.
    b. Select a **snapshot** from which to refresh.
11. Refresh the EBS VDB.
    a. On the VDB, click the **Refresh**.
    b. Select a **snapshot** from which to refresh.
12. Refresh the appsTier vFiles.
    a. On the vFiles, click the **Refresh**.
    b. Select a **snapshot** from which to refresh.

Once you have refreshed all three EBS virtual datasets successfully, your virtual EBS instance should be running and accessible.

9.8.6.8 Enabling and disabling a virtual EBS instance

This topic describes the process of enabling and disabling a virtual Oracle E-Business Suite (EBS) instance.

An enabled virtual EBS instance will be running and fully available to end-users. A disabled virtual EBS instance will be neither running nor mounted to the target environments.

> Be careful! Services running on the AppsTier depend on the availability of services on the DBTier. The steps below are explicitly ordered with these dependencies in mind. Executing steps out of order may lead to errors in accessing the virtual EBS instance.

Please refer to the Sequence of provisioning (see page 1652) for details on the order in which provisioning of datasets must be done.

9.8.6.8.1 Disabling

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Datasets**.
4. Select the **AppsTier vFiles** for your EBS instance.
5. From the Actions menu (...) select **Disable**. Disabling the AppsTier vFiles will stop the AppsTier services and unmount the AppsTier files.

> **Note**: Stopping the AppsTier may take a long time. The Delphix Engine will wait for all Oracle application processes to exit before declaring the AppsTier as stopped..

6. Select the **VDB** utilized by your EBS instance.
7. From the Actions menu (...) select **Disable**. Disabling the VDB will stop the database instance and unmount the data files.
8. Select the **DBTechStack vFiles** hosting your virtual EBS database.
9. From the Actions menu (...) select **Disable**. Disabling the DBTechStack vFiles will stop the database listener and unmount the DBTechStack files.

Once you have disabled all three EBS virtual datasets successfully, your virtual EBS instance should be fully removed from the target environment.

9.8.6.8.2 Enabling

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Datasets**.
4. Select the **DBTechStack vFiles** hosting your virtual EBS database.
5. From the Actions menu (...) select **Enable**. Enabling the DBTechStack vFiles will mount the DBTechStack files and start the database listener.
6. Select the **VDB** utilized by your EBS instance.
7. From the Actions menu (...) select **Enable.** Enabling the VDB will mount the data files and start the database instance.
8. Select the **AppsTier vFiles** hosting your virtual EBS database.
9. From the Actions menu (...) select **Enable.** Enabling the DBTechStack vFiles will mount the AppsTier files and start the application services.

Once you have enabled all three EBS virtual datasets successfully, your virtual EBS instance should be running and accessible.

9.8.6.9 Deleting a virtual EBS instance

This topic describes the process of deleting a virtual Oracle E-Business Suite (EBS) instance.

> Be careful! Services running on the AppsTier depend on the availability of services on the DBTier. The steps below are explicitly ordered with these dependencies in mind. Executing steps out of order may lead to errors in accessing the virtual EBS instance.

9.8.6.9.1 Procedure

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Datasets**.
4. Select the **AppsTier vFiles** for your EBS instance.
5. Delete the AppsTier vFiles by clicking the **Trash Can** icon in the lower left-hand corner.
6. Select the **VDB** utilized by your EBS instance.
7. Delete the VDB by clicking the **Trash Can** icon in the lower left-hand corner.
8. Click **Manage**.
9. Select **Environments**.
10. Select the **target DBTier environment**.
11. Click the **Databases** tab.
12. In the list of **Installation Homes** on the environment, click the **Trash Can** icon next to the dbTechStack Oracle Home you want to delete.
13. Click **Manage**.
14. Select **Datasets**.
15. Select the **DBTechStack vFiles** for your EBS instance.
16. Delete the DBTechStack vFiles by clicking the **Trash Can** icon in the lower left-hand corner.
17. Clean up any files that the virtual EBS instance might have created outside of the Delphix mount points on the target environments. These typically include the instance-specific directories, oraInventory files, and oraTab entries.

Once you have deleted all three EBS virtual datasets successfully, your virtual EBS instance should be fully removed from the target environments.

9.8.6.10 Modifying the AppsTier topology

This topic describes the process of modifying the AppsTier topology of a virtual EBS instance.

AppsTier Topology Changes Invalidate Rewind

If you modify the appsTier topology of a virtual EBS instance, all snapshots in existence prior to the modification will no longer be valid for rewind. The AppsTier topology in the snapshots will no longer match the Delphix Engine's configuration and rewind targeting these snapshots will fail if attempted.

9.8.6.10.1 Procedure

1. Disable the virtual EBS instance by following the procedure outlined in Enabling and disabling a virtual EBS instance (see page 1749)
2. Select the **AppsTier vFiles** for your EBS instance.
3. On the back of the card, modify the **EBS-specific parameters** for the virtual AppsTier. This process will normally entail adding or deleting **Additional Nodes** and configuring its corresponding **Services**.
4. Apply the configuration changes by refreshing the entire virtual EBS instance. Follow the procedure outlined in Refreshing a virtual EBS instance (see page 1748)

9.8.6.11 Replicating a virtual EBS instance

This topic describes the process of replicating a virtual Oracle E-Business Suite (EBS) instance.

9.8.6.11.1 Replication

1. Login to the *source* Delphix Engine's **Delphix Management** application.
2. Configure replication between the source Delphix Engine and a target Delphix Engine. For a detailed outline of the replication process, see Configuring replication (see page 2381).[513]
3. Select the **DBTechStack vFiles**, **VDB**, and **AppsTier vFiles** objects to be replicated. These objects have dependencies on all other Delphix Engine objects relevant to the virtual EBS instance: you do not need to specify any additional objects for EBS replication.
4. Schedule or perform the replication.

513 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/configuring-replication

9.8.6.11.2 Failover

1. Log in to the *target* Delphix Engine's **Delphix Management** application**.**
2. Failover the replica **DBTechStack vFiles**, **VDB**, and **AppsTier vFiles**. Failing over these object will sever future replication, but will not enable the datasets. For a detailed outline of the failover process, see Controlled failover (see page 2385)
3. Enable the DBTier and AppsTier environments.
   a. Click **Manage**.
   b. Select **Environments**.
   c. For each environment, from the Actions menu (...) select **Disable**.
4. Enable the virtual EBS instance by following the procedure outlined in Enabling and disabling a virtual EBS instance (see page 1749)

9.8.6.11.3 Provisioning from a replicated EBS instance

You can provision new virtual EBS instances from replicated datasets regardless of whether these datasets have been failed over. The provisioning process for each version of EBS is outlined in the Virtualizing Oracle E-Business suite (see page 1652).[514] For more information about provisioning from replicated datasets, see Replicating a virtual EBS instance (see page 1752)

9.8.6.12 Upgrading a Delphix Engine hosting a virtual EBS instance

This topic describes the process of upgrading a Delphix Engine hosting a virtual Oracle E-Business Suite (EBS) instance.

9.8.6.12.1 Procedure

1. Disable the virtual EBS instance by following the procedure outlined in Enabling and Disabling a Virtual EBS Instance (see page 1749)
2. Once you have safely disabled the virtual EBS instance, upgrade the Delphix Engine by following the procedure outlined in Upgrading the Delphix Engine (see page 925)
3. Enable the virtual EBS instance by following the procedure outlined in Enabling and Disabling a Virtual EBS Instance (see page 1749)

9.8.6.13 Migrating an EBS VDB

This topic describes how to migrate an EBS Virtual Database (VDB) from one target environment to another.

There may be situations in which you want to migrate a virtual database to a new target environment, for example when upgrading the host on which the VDB resides, or as part of a general data center migration.

---

514 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/datasets-virtualizing-oracle-e-business-suite

This is easily accomplished by first disabling the database, then using the Migrate VDB feature to select a new target environment.

There are three components for any EBS environment - DBTechstack, Database, and AppsTier. Whenever we create virtual copies for any EBS environment, we provision these components in the following sequence:

1. DBTechstack
2. Database
3. AppsTier

We can **only** migrate the EBS database (VDB) with the following prerequisite.

9.8.6.13.1 Prerequisite

You should have already set up a new target environment that is compatible with the VDB that you want to migrate.

9.8.6.13.2 Procedure

1. Login to your Delphix Management application.
2. Click **Manage**.
3. Select **Datasets**.
4. Select the **VDB** you want to migrate.
5. From the Actions menu (...) select **Disable.**
6. Click **Disable** to confirm. When the VDB is disabled, its icon will turn gray.
7. From the Actions menu (...) select **Migrate**.
8. Select the new target environment for the VDB, the user for that environment, and the database installation where the VDB will reside.
9. Select **Migrate** to confirm your selections.
10. From the Actions menu (...) select the **Enable**
11. Click **Enable** to confirm.
    Within a few minutes, your VDB will restart in the new environment, and you can continue using it.

- You need to Provisioning the Oracle EBS DBTechStack (see page 1701) on the new target environment with a similar DB name VDB that you want to migrate.
- EBS appsTier vFile cannot be migrated, you need to re-configure the EBS appsTier (see page 1738) by modifying the custom configuration for Target DB server node information to the target environment where VDB is migrated.

9.8.7 Troubleshooting Oracle EBS

This section covers the following topics:

- EBS connector logging (see page 1755)

9.8.7.1 EBS connector logging

Remote Logging for EBS 12.2 connector adds the following options to provide enhanced logging:

- set debug log levels
- set active log file size
- set log retention levels

You can modify the parameters defined in the ebs_plugin.conf file located in remote target host at <Toolkit directory>/<Delphix_COMMON>/plugin/ebs122-vsdk-db_9f7f5a63-1da7-495c-80df-e0fa61941342/ebs_plugin.conf for DBTechstack and <Toolkit directory>/<Delphix_COMMON>/plugin/ebs122-vsdk-app_403e7034-0028-4200-ad0b-15031f0f8871/ebs_plugin.conf for AppsTier to configure the logging mechanism.

9.8.7.1.1 Setting log levels (level)

You can set log levels for connector-generated logs by setting the parameter level in the ebs_plugin.conf file. Higher logging levels will help to expedite debugging issues. There are two levels of logging: info and debug.

Log level description

- level=INFO: This level will print only informational logs. This is the default log level.
- level=DEBUG: This level will print informational logs and debug statements.

9.8.7.1.2 Setting maximum size of active log file (logFileSize)

You can set this parameter inside the ebs_plugin.conf file to set the maximum size of the active log file in MB. Once this limit is reached, the connector rotates the log as per the retention property defined below. The minimum value of this parameter is 1 MB and the maximum value is 10 MB. This parameter only takes in a positive integer value. The default value is 1 MB.

9.8.7.1.3 Setting log retention levels (retention)

You can set a retention level for diag logs (diag.log) using parameter retention in the ebs_plugin.conf file. As per this parameter, the log files are moved (archived), renamed, or deleted once they reach the value set in the logFileSize parameter. New incoming log data is directed into a new fresh file (at the same location).

By default, this value will be set to a minimum value of 2. The user can change this value and set its value within the range 2 and 50. For example; if retention is set to 4, the connector will have the following log files: diag.log, diag.log.1, diag.log.2, diag.log.3, diag.log.4.

- File diag.log is the active log file.
- File diag.log.1 is the most recent archive log file

- File diag.log.4 is the oldest one

ebs_plugin.conf file

```
[plugin_logging_parameters]
#
# This flag will set the debug level of plugin logs on the remote
server. There are two levels:
# Info
# Debug
# The above are the only valid values that can be assigned. If any other value is
assigned, the plugin will set the
# default level as Info. The string is case insensitive so info, Info and INFO are
acceptable.
#
level=INFO
#
# This parameter will set the maximum size of the active log file in MB. Once this
 limit is reached, the plugin will
# rotate the log as per the retention property defined below. The minimum value of
this parameter is 1 MB and maximum
# value is 10 MB. This parameter only takes in a positive integer value. The default
 value is 1 MB
# logFileSize=<positive integer>
#
logFileSize=1
#
# Whenever the size of diag.log file exceeds the value provided by logFileSize
parameter then the plugin will
# rename the active log file to diag.log.<number> and a new log file with name
diag.log will be
# generated. For example, if LogRetention is set to 4, the plugin will have the
following log files: diag.log,
# diag.log.1, diag.log.2, diag.log.3, diag.log.4.
# File diag.log.4 will be the oldest one.
# File diag.log.1 will be the most recent archive log file.
# File diag.log will be the active log file.
# The minimum and default value for the retention flag is 2.
# retention=<positive integer>
retention=2
```

9.8.8 Oracle EBS additional information

This section includes the following operations:

- Oracle central inventory (see page 1757)
- ETCC patch level (see page 1758)
- Clean up before provisioning option (see page 1758)
- Configuring a Delphix self-service data template for EBS (see page 1758)
- Managing the APPS password (see page 1759)

- Refreshing a target DBTier environment (see page 1761)
- Managing the Weblogic password (see page 1761)
- Working with EBS snapshots (see page 1763)
- Monitoring Oracle EBS R12.2 AppsTier provisioning progress (see page 1764)
- Monitoring Oracle EBS R12.2 DBTechstack provisioning progress (see page 1764)
- Unsupported EBS versions and features (see page 1765)

9.8.8.1 Oracle central inventory

The Oracle Central Inventory is a repository for all Oracle products (software) installed on a system. Since the Central Inventory consists of system-specific information, it is required that the Central Inventory be saved in a local non-shared directory on the system. While software can be shared across nodes, the inventory should be local to each system. The global inventory/central inventory(oraInventory) for each node should be on a local file system other than the Delphix mount base path.

For more info, see the following notes

- **Note 360079.1- Global and Local Inventory explained.**[515]
- **Note 564192.1- FAQs on Central Inventory and Oracle Home Inventory (Local Inventory) in Oracle RDBMS.**[516]
- **Note 834894.1- How to Create a Clean oraInventory in Release 12?**[517]
- **Note 742477.1 - How to create, update, or rebuild the Central Inventory for Applications R12?**[518]
- **Note 295185.1- How to Recreate the Global oraInventory?**[519]

9.8.8.1.1 oraInst.loc

An oraInst.loc file must exist on every AppsTier and database node before provisioning. This file will specify where the oraInventory directories live or where they should be created if they do not already exist.

The oraInst.loc file is typically located at `/etc/oraInst.loc` on Linux or `/var/opt/oracle/oraInst.loc` on Solaris. Ensure that the oraInventory to which this file points is writable by the application or database tier OS user.

If you are provisioning a single-node AppsTier,

- [Required] The global inventory/central inventory(oraInventory) for each node should be on a local file system other than the Delphix mount base path.
- If you are provisioning a multi-node AppsTier,
- The following recommendations are mandatory;
    - While configuring SHARED APPL_TOP application, Delphix expects the oraInventory directory path should be beneath the Delphix mount base path.

---

515 https://support.oracle.com/knowledge/Oracle%20E-Business%20Suite/360079_1.html
516 https://support.oracle.com/knowledge/Oracle%20Database%20Products/564192_1.html
517 https://support.oracle.com/knowledge/Oracle%20E-Business%20Suite/834894_1.html
518 https://metalink.oracle.com/metalink/plsql/showdoc?db=NOT&id=742477.1
519 https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=515087639867449&parent=DOCUMENT&sourceId=878717.1&id=295185.1&_afrWindowMode=0&_adf.ctrl-state=qra3xc5e9_4

    - The `inventory_loc` variable defined in `file /etc/oraInst.loc` on all application servers should point to the same shared oraInventory directory location on the shared Delphix mount base path.
    - The Delphix Continuous Data Engine's cloning workflow requires that all nodes in the AppsTier have access to the shared oraInventory directories via Delphix-provided storage.
- [Required] The Delphix Continuous Data Engine's automation requires that all nodes in the AppsTier have access to the oraInventory directories via Delphix-provided storage.

9.8.8.2 ETCC patch level

The E-Business Suite Technology Codelevel Checker (ETCC) utility identifies patches that need to be applied to your Oracle E-Business Suite 12.2 technology stack for the application and database tiers.
Check https://blogs.oracle.com/ebstech/category/ebs-etcc for the latest patches that are required.

9.8.8.3 Clean up before provisioning option

If you plan to utilize the Cleanup Before Provision option available during DBTechstack provisioning, the Delphix Continuous Data Engine requires the Database Oracle Home to be patched with Oracle Universal Installer (OUI) version 10.2 or above. You can read more about the Clean Up Provisioning option in Provisioning a Virtual EBS R12.2 Instance. Note that provisioning is still possible without this option specified, but you will need to manage the target dbTier's Oracle Inventory manually to ensure that conflicting entries do not cause provisions to fail.

9.8.8.4 Configuring a Delphix self-service data template for EBS

This topic describes the process of configuring a Delphix Self-Service (Jet Stream) data template for Oracle E-Business Suite (EBS).

By configuring a data template for EBS, you eliminate the need to order operations applied across the EBS datasets.

The ordering of operations is a result of the dataset dependencies discussed in topics under Managing Data Operations of Virtual EBS Instances, such as Refreshing a Virtual EBS Instance.[520]

9.8.8.4.1 Procedure

1. Create a Delphix Self-Service data template by following the procedure outlined in Understanding Data Templates
2. For EBS, the data template will have three data sources: the DBTechstack, database, and AppsTier.
3. Be sure to set the correct sequence of **order for the data sources** when creating the data template. This ordering will ensure that Delphix Self-Service operations do not violate the EBS dataset dependencies. For order please refer: Sequence of provisioning (see page 1652)

520 http://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/refreshing-a-virtual-ebs-instance

4. Once you have created a Delphix Self-Service data template, you can configure Delphix Self-Service data containers to manage virtual EBS instances. Delphix Self-Service data containers will follow the ordering of data sources configured in the template. All Delphix Self-Service operations should work as expected for virtual EBS instances.

9.8.8.5 Managing the APPS password

This topic outlines how to manage the APPS password on a virtual EBS instance.

The Delphix Engine requires a virtual EBS instance's APPS password in order to manage its DBTechStack and AppsTier. This password is encrypted when stored within the Delphix Engine and is exposed as an environment variable to RapidClone tools.

When a virtual EBS instance is being provisioned, the Delphix Engine's copy of the APPS password should match the source EBS instance's APPS password during its time of the snapshot. After provisioning, you can change the APPS password of the virtual instance in both EBS and the Delphix Engine.

9.8.8.5.1 Changing the APPS password

1. Provision a virtual EBS instance.
2. Change the APPS password in the virtual EBS instance. You can perform the password change using Oracle's `FNDCPASS` utility.
3. Stop the virtual AppsTier manually using `adstpall`. You cannot stop the virtual AppsTier through the Delphix Engine, because the APPS password being stored is now out of sync.
4. Disable the entire virtual EBS instance. For an outline of this process, see Enabling and Disabling a Virtual EBS Instance (see page 1753).[521]
5. Change the APPS password on the DBTechStack.
   a. Select the **DBTechStack vFiles** hosting your virtual EBS database.
   b. On the back of the card, click the **Custom** tab.
   c. Edit the **APPS Password** field.
6. Change the APPS password in any hook operations defined on the EBS virtual database.
   a. Select the **VDB** utilized by your EBS instance.
   b. On the back of the card, click the **Hooks** tab.
   c. Edit the relevant hook operations. Typically, you will need to edit the Pre-Snapshot hook operation running `adpreclone`.
7. Change the APPS password on the appsTier.
   a. Select the **AppsTier vFiles** hosting your virtual EBS database.
   b. On the back of the card, click the **Custom** tab.
   c. Edit the **APPS Password** field.

521 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/enabling-and-disabling-a-virtual-ebs-instance

8. Enable the entire virtual EBS instance. For an outline of this process, see Enabling and Disabling a Virtual EBS Instance (see page 1749).[522]

9.8.8.5.2 Refreshing or rewinding

The APPS password stored across individual snapshots of a virtual EBS instance will not be consistent after a password change. Old snapshots of EBS data will refer to a different APPS password than new snapshots of EBS data. To perform a refresh or rewind, you must explicitly manipulate the Delphix Engine's copy of the APPS password to ensure that the virtual EBS instance is being accessed with the correct APPS password at every step.

> Recipe Not Needed If Password Not Changed
> These steps are only necessary if the virtual EBS instance has a different APPS password than the snapshots being targeted by the refresh or rewind.
> If the APPS password has not been changed, follow the instructions in Refreshing a Virtual EBS Instance (see page 1748) or Rewinding a Virtual EBS Instance (see page 1746).[523]

1. Before refreshing or rewinding the virtual EBS instance, disable the entire virtual EBS instance. For an outline of this process, see Enabling and Disabling a Virtual EBS Instance (see page 1749).[524]
2. Identify the APPS password for the snapshots being targeted by the refresh or rewind. Modify the virtual EBS instance to refer to this password.
   a. Change the APPS password on the dbTechStack.
      i. Select the **DBTechStack vFiles** hosting your virtual EBS database.
      ii. On the back of the card, click the **Custom** tab.
      iii. Edit the **APPS Password** field.
   b. Change the APPS password in any hook operations defined on the EBS virtual database.
      i. Select the **VDB** utilized by your EBS instance.
      ii. Select the Configuration tab, then select the **Hooks** tab.
      iii. Edit the relevant hook operations. Typically, you will need to edit the Pre-Snapshot hook operation running `adpreclone` and the Configure Clone operation running `adcfgclone`.
   c. Change the APPS password on the appsTier.
      i. Select the **AppsTier vFiles** hosting your virtual EBS database.
      ii. Click the **Custom** tab.
      iii. Edit the **APPS Password** field.
3. Perform the refresh or rewind while the EBS instance is disabled. See Refreshing a Virtual EBS Instance (see page 1748) or Rewinding a Virtual EBS Instance (see page 1746) for an outline of these processes.

---

522 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/enabling-and-disabling-a-virtual-ebs-instance
523 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/rewinding-a-virtual-ebs-instance
524 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/enabling-and-disabling-a-virtual-ebs-instance

9.8.8.6 Refreshing a target DBTier environment

This topic describes how to properly refresh a target DBTier environment hosting an EBS VDB.

When an environment is initially registered with the Delphix Engine, a discovery process is responsible for registering resources such as Oracle Homes, listeners, and databases. When an environment is modified – for example, when an Oracle Home is upgraded or removed – you must run the discovery process again in order to register new resources and unregister removed ones.

The virtual DBTechStack plays a unique role in the Delphix Engine because it acts both as a virtual dataset AND a discovered resource on the target DBTier environment. Because of this unique role, you must mount and start this dataset (hosting the EBS database listener) before you refresh the target DBTier environment.

If you unmount or stop the virtual DBTechStack, the discovery process will assume that the EBS Oracle Home and database listener have been removed from the environment. Further interactions with the EBS VDB will result in an error because neither the database listener nor Oracle Home appears available to the database.

9.8.8.6.1 Procedure

1. Prior to refreshing a target DBTier environment, ensure that the virtual DBTechStack is both enabled and started. For more information, see Enabling and Disabling a Virtual EBS Instance (see page 1749) and Starting and Stopping a Virtual EBS Instance (see page 1745).[525]
2. Refresh the target DBTier environment.

9.8.8.7 Managing the Weblogic password

9.8.8.7.1 Changing the Weblogic password

For existing EBS 12.2 customers who intend to use this WebLogic password change feature need to follow the below steps for their EBS application:

1. Disable all the virtual datasets created on Delphix Engine in the following order:
   a. appsTier vFiles
   b. VDB utilized by your EBS instance
   c. dbTechStack vFiles
2. Disable all the EBS dSource created on Delphix Engine once you have disabled all three EBS virtual datasets in the above step.
3. Upgrade Delphix Engine to 5.3.2.0 release and upload EBS 1.4.0 connector release.

   Go to **Manage>Plugins**

   - Upload a plugin file to upgrade the existing connector by clicking plus (+) symbol.
4. Refresh all host environments added on the Delphix Engine.

525 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/starting-and-stopping-a-virtual-ebs-instance

Go to Manage> Environments

- From the symbol (...) located in the upper left-hand corner, select Refresh All.

5. Change the Weblogic admin password on EBS appsTier vFile.
   - Select the appsTier vFiles hosting your virtual EBS database.
   - Go to Configurations >Custom
   - Update the **Target Weblogic AdminServer Password** field with the password you would like to set for EBS appsTier.
6. Enable all the EBS dSource created on Delphix Engine.
7. Enable all the virtual datasets created on Delphix Engine in the following order:
   a. appsTier vFiles
   b. VDB utilized by your EBS instance
   c. dbTechStack vFiles
8. Verify the changed WebLogic password by logging into the WebLogic console.

EBS 12.2 customers configuring appsTier on a multi-node environment will need to manually enable passwordless SSH on all target nodes *only when* they want to change the Weblogic password and then proceed with EBS appsTier provision. The passwordless SSH **should** be enabled between all nodes and for both low privileged and high privileged user.

9.8.8.7.2 Enabling passwordless SSH between target application nodes

To enable passwordless SSH between target application nodes complete the following

The following is an example of a two-node EBS appsTier.

Node 1: test1-ap1

Node 2: test1-ap2

1. Create Authentication SSH-Kegen Keys on Node 1.

```
[applvis@test1-ap1 ~]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/applvis/.ssh/id_rsa):
Created directory '/home/applvis/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/applvis/.ssh/id_rsa.
Your public key has been saved in /home/applvis/.ssh/id_rsa.pub.
The key fingerprint is:
b8:1d:83:ef:40:d3:54:d6:50:2d:74:28:66:8a:4d:15 applvis@test1-ap1.dcenter.delphix.com
The key's randomart image is:
+--[ RSA 2048]----+
|          .E*oo.  |
|         .o+ +..  |
|        +.+ . .   |
|       .=o        |
|       = S        |
|      . = o       |
|       o o        |
|        o         |
|         .        |
+-----------------+
[applvis@test1-ap1 ~]$
```

2. Create .ssh Directory on Node 2.

```
[applvis@test1-ap1 ~]$ ssh applvis@test1-ap2 mkdir -p .ssh
The authenticity of host 'test1-ap2 (10.43.17.170)' can't be established.
RSA key fingerprint is 22:83:2b:d2:53:03:3e:b6:8b:fe:6a:87:21:35:d5:fc.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'test1-ap2,10.43.17.170' (RSA) to the list of known hosts.
applvis@test1-ap2's password:
[applvis@test1-ap1 ~]$
```

3. Upload Generated Public Keys to Node 2.

```
[applvis@test1-ap1 ~]$ cat .ssh/id_rsa.pub | ssh applvis@test1-ap2 'cat >> .ssh/authorized_keys'
applvis@test1-ap2's password:
[applvis@test1-ap1 ~]$
```

4. Set Permissions on Node 2 and Login to verify without password.

```
[applvis@test1-ap1 ~]$ ssh applvis@test1-ap2 "chmod 700 .ssh; chmod 640 .ssh/authorized_keys"
[applvis@test1-ap1 ~]$
[applvis@test1-ap1 ~]$ ssh test1-ap2
Last login: Mon Jan 28 21:46:12 2019 from 10.43.15.129
[applvis@test1-ap2 ~]$
[applvis@test1-ap2 ~]$ hostname -s
test1-ap2
```

9.8.8.8 Working with EBS snapshots

Taking a snapshot creates a new snapshot entry in the dSource's Timeflow. The connector-managed dSource allows you to select the parameters before taking the snapshot. These parameters are provided as an input to pre-snapshot and post snapshot functions. On the other hand, in the default snapshot, the parameters are pre-defined in the dSource environment.

Snapshot (Default)

This section lists the steps to take a snapshot and delete the same.

1. Login to the **Delphix Management** application.
2. Click **Manage** and select **Datasets** from the dropdown list.
3. Select the dSource you want to Snapshot.
4. Click the **Camera** icon. Alternatively, click the arrow next to the Camera icon and select **Snapshot (default)**.
5. From the Snapshot dialog, select **Perform Snapshot**.
6. Click **View:** under **Timeflow** to verify the Snapshot you just created.
7. To delete, select the snapshot you just created, and from the Actions menu (...) select **Delete Snapshot**.
8. In the Delete Snapshot dialog, select **Delete**.
9. Click **View:** under **Timeflow** to verify the Snapshot, you just deleted.

9.8.8.9 Monitoring Oracle EBS R12.2 AppsTier provisioning progress

This topic describes how to monitor the progress of EBS R12.2 AppsTier provisioning.

The Delphix Continuous Data Engine automates the configuration of the AppsTier during provisioning. It spends the majority of provisioning time running RapidClone utilities such as *adcfgclone*. The Delphix Continuous Data Engine does not report the progress of RapidClone utilities through the progress of the **DB_Provision** job. To track provisioning progress more accurately, Delphix recommends monitoring EBS log files generated on the target AppsTier environment.

9.8.8.9.1 Procedure

1. Connect to the target AppsTier environment using SSH or an alternative utility.
2. Change directories to the <INST_TOP>/admin/log/.
   - Replace **<INST_TOP>** with the value of **INST_TOP** on the virtual EBS instance.
3. After *adcfgclone* utility has begun running, a file matching ApplyAppsTier_*.log will exist. Identify this log file and use the tail or an equivalent utility to monitor it.
4. Leave the Mount Point before Refresh

If provisioning fails, you will need to fix the cause of the failure and perform a refresh of the dataset before you can attempt configuration again. Before refreshing or disabling the dataset, be sure to change directories to a location outside of the mount path on the target environment. If you leave a shell session with a current working directory inside the mount path, the dataset will fail to unmount cleanly during a refresh or disable.

9.8.8.10 Monitoring Oracle EBS R12.2 DBTechstack provisioning progress

This topic describes how to monitor the progress of EBS R12.2 DBTechstack provisioning.

The Delphix Continuous Data Engine automates the configuration of the DBTechstack during provisioning. It spends the majority of provisioning time running RapidClone utilities such as adcfgclone. The Delphix Continuous Data Engine does not report the progress of RapidClone utilities through the progress of the

DB_Provision job. To track provisioning progress more accurately, Delphix recommends monitoring EBS log files generated on the target dbTier environment.

9.8.8.10.1 Procedure

1. Connect to the target dbTier environment using SSH or an alternative utility.
2. Change directories to the <ORACLE_HOME>/appsutil/log/<CONTEXT_NAME>/
   - Replace <ORACLE_HOME> with the path to the DBTechstack's Oracle Home. This path will be under the mount path specified during provisioning.
   - Replace <CONTEXT_NAME> with the virtual EBS instance's context name.
3. After *adcfgclone* utility has begun running, a file matching ApplyDBTechstack_*.log will exist. Identify this log file and use tail or an equivalent utility to monitor it.

Leave the mount point before refreshing

If provisioning fails, you will need to fix the cause of the failure and perform a refresh of the dataset before you can attempt configuration again. Before refreshing or disabling the dataset, be sure to change directories to a location outside of the mount path on the target environment. If you leave a shell session with a current working directory inside the mount path, the dataset will fail to unmount cleanly during a refresh or disable.

9.8.8.10.2 Caveats

The Delphix Continuous Data Engine will only register the default WLS-managed server for each server type during multi-node AppsTier provisioning. Additional WLS-managed servers need to be registered manually or as a **Configure clone** hook.

9.8.8.11 Unsupported EBS versions and features

- Support SSL enabled Environment
- V2P

For specific plugin version release notes, see Oracle EBS 12.2 Connector Release Notes (see page 495).

9.8.9 Oracle EBS glossary

9.8.9.1 delphix_os user

This is the low-privilege user for both Appstier & DBTechstack that is referenced in EBS documentation. This user does not have the necessary sudo privileges.

This is the user that is used when the customer doesn’t want to give sudo permissions to a user profile used in Delphix Engine and just wants to give privileges for specific required commands using sudoers entry.

To separate authentication and perform privileged operations with a non-privileged OS account, first, an OS user account (i.e. `delphix_os`) needs to be created on the EBS DB Tier and AppsTier node to be used as a target. This user is easily created by the **createDelphixOSUser.sh** script (located in EBS documentation in the section creating delphix_os (see page 1664)user).

The primary OS group of the Delphix Continuous Data Engine software owner account (i.e. `delphix_os`) should be the same as the EBS Database/AppsTier software owner account (i.e. `oravis` or `applmgr`).

Primary group = Oracle Install Group (typically `oinstall`), secondary group = OSDBA Group (typically `dba`). There are lots of cases where the OS group named dba fills this role, so be sure to check the group membership of the EBS Database and AppsTier software owner account.

Please note, that the non-privileged OS account must have the same group as assigned to EBS Database or AppsTier privileged account (like `oravis` or `applmgr`).

Delphix recommends giving the delphix_os user the same primary OS group as the Oracle home owner. This ensures the Delphix engine can fully and automatically discover Oracle homes, databases, and listeners.

Often, this is an OS group named `oinstall`. However, the `oinstall` group is not always necessary depending on your Oracle configuration. This user requires access to the libobk_proxy.so library in the toolkit for the child processes triggered by RMAN.

9.8.9.2 oravis

This is the primary user for DBTechstack and DBtier which is referenced in EBS documentation. This user has the necessary sudo privileges.

9.8.9.3 applvis

This is the primary user for Appstier which is referenced in EBS documentation. This user has the necessary sudo privileges.

9.8.9.4 WLS (Web Logic Server)

In Oracle E-Business Suite (EBS), WLS stands for WebLogic Server. In the context of Oracle EBS, WebLogic Server is used as the application server tier, responsible for hosting the web-based components of EBS applications.

9.8.10 Oracle EBS FAQs

This section covers the following FAQs:

- What are important notes that are needed for EBS? (see page 1767)
- How to use the Delphix Continuous Data Engine key pair for Oracle EBS? (see page 1767)

9.8.10.1 What are important notes that are needed for EBS?

Oracle E-Business Suite Installation and Upgrade Notes Release 12 (12.2) for Linux x86-64 (Doc ID 1330701.1)

Delphix Continuous Data Engine version 6.0.17.x along with EBS plugin version 4.2.1 is certified with 19c database, EBS apps 12.2.x and Solaris 11 U4.

9.8.10.2 How to use the Delphix Continuous Data Engine key pair for Oracle EBS?

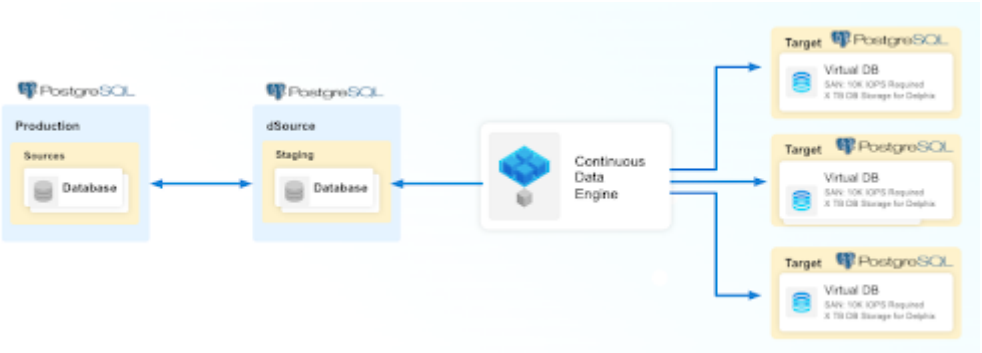
Perform the following steps to use the Delphix Continuous Data Engine key pair:

1. Select **Public Key** for the **Login Type.**
2. Click **View Public Key**.
3. Copy the public key that is displayed, and append it to the end of your ~/.ssh/authorized_keys file. If this file does not exist, you must create it.
   a. Run `chmod 600 ~/.ssh/authorized_key` to allow only the file's owner to read and write to it (make sure the file is owned by you).
   b. Run `chmod 755 ~` to restrict access to your home directory so that no other may write to it.
   c. Run `chmod 700 ~/.ssh` so that others cannot write to it. The ~/.ssh directory is not writable by group or others. Otherwise, authentication will fail.

9.9 PostgreSQL data sources

9.9.1 Overview

The Delphix PostgreSQL virtualization solution consists of the Delphix Continuous Data Engine and the Connector for PostgreSQL which must be uploaded and installed into the Engine. This solution ingests source data using the staging architecture (see page 1014), which copies (logically or physically, depending on the ingestion method chosen) the production database from the source environment into a staging environment. The Delphix Continuous Data Engine captures the persistent storage layer of the staging database, which then allows the Delphix Continuous Data Engine to create virtualized copies in target environments.



The PostgreSQL connector supports both staging pull and staging push for data ingestion.

The following specific ingestion methods are supported:

1. Staging Pull: This ingestion method allows Delphix to execute the necessary operations to prepare the data content in the Staging Environment. Staging pull supports the following ingestion methods:
    - Delphix-initiated single database backup
    - PostgreSQL streaming replication with Delphix-initiated Backup
    - PostgreSQL streaming replication with Externally-initiated backup
    - Log Shipping with Externally-initiated backup
2. Staging Push: This ingestion method allows the user to choose and tailor the specific operations and processes to prepare the data content in the Staging Environment.

9.9.2 Limitations of the PostgreSQL solution

The Delphix Continuous Data Engine and the PostgreSQL connector should not be considered a replacement for the backup and recovery of your source instances and databases. The Delphix Continuous Data Engine solution is intended for different purposes, primarily to create non-production database copies.

PostgreSQL will not allow provisioning of a single PostgreSQL vDB filesystem into a pre-existing PostgreSQL cluster. The Delphix solution must create a new PostgreSQL instance and vDB on the target Environment with its own unique HTTP port.

9.9.3 Support

License for the PostgreSQL connector is included with the standard Delphix License Agreement, and Delphix Technical Support backs the solution. For questions, bugs, or feature requests, contact us via the Delphix Support[526] or the Delphix Community[527] portals.

9.9.4 PostgreSQL virtualization process

The following material introduces you to the overall process of deploying the PostgreSQL virtualization solution. A sequence of high-level steps is presented; each step briefly describes the task and the expected result. Refer to the embedded links for more detailed information.

- Begin by reviewing the general Delphix Continuous Data Engine architecture diagrams (see page 1014). Identify existing or new PostgreSQL source, staging, and target Environments, the underlying Operating Systems, and PostgreSQL database versions. This will give a good understanding of the architectural strategy and confirm support.
- Review the PostgreSQL Support Matrix (see page 1769). Confirm that your combination of PostgreSQL software, host operating system, Delphix Engine version, and PostgreSQL connector version are supported for the various environments which will integrate into your PostgreSQL deployment. Early validation will reduce unexpected issues later.

526 https://support.delphix.com/
527 https://community.delphix.com/contactus

- Deploy and configure Delphix Continuous Data.
- Install the PostgreSQL connector (see page 1792). The PostgreSQL connector extends the Delphix Continuous Data Engine to PostgreSQL datasets. It is recommended to stay current with the latest version of the connector.
- Create staging and target environments (see page 1769). Create the staging and target hosts and ensure that they are running and reachable. Follow the host requirements to configure the respective environments. The staging environment will host a PostgreSQL instance that the Delphix Continuous Data Engine will ingest from. The target environment will host one or more PostgreSQL instance(s) in which the virtual database copies will be created.
    a. Consult the source environment (see page 1776) and staging/target environment (see page 1780) requirements documentation for configuration details.
- Add Staging and Target Environments in the Delphix Continuous Data Engine (see page 1796). Use the "Add Environment" operation in Delphix Continuous Data Engine for each host, which causes Delphix to register the specific host and catalog the PostgreSQL software installation(s) on that host.
- Link the PostgreSQL source database (see page 1802). Configure and create a dSource which allows the Delphix Continuous Data Engine to capture a copy of the source data. There are many ingestion methods that can offer a variety of methods to capture the data to be virtualized by Delphix. Therefore, we encourage you to identify the ingestion method that best suits your organization and infrastructure policies.
- Provision virtual PostgreSQL databases (see page 1828). Administrators and users can provision multiple copies of the source data as virtual databases.

9.9.5 PostgreSQL requirements and prerequisites

This section covers the following topics:

- PostgreSQL Support matrix (see page 1769)
- Requirements for source databases (see page 1776)
- Requirements for staging and target environments (see page 1780)

9.9.5.1 PostgreSQL Support matrix

Refer to the following support matrices to ensure compatibility between the environment, database, connector, and Delphix Continuous Data Engine.

Compatible upgrades have a green **Y**, while incompatible upgrades show a red **N**.

Per Delphix Support Policies, when a minor release is certified, all the patch releases falling under that minor release are also considered certified.

9.9.5.1.1 Environment and database compatibility

9.9.5.1.1.1 Operating system

The PostgreSQL connector is certified for use in source, staging, and target environments with the following operating systems and database versions.

OS version	Database version (open-source & EDB)									
	9.4.x	9.5.x	9.6.x	10.x	11.x	12.x	13.x	14.x	15.x	16.x
RHEL/CentOS 7.3	Y	Y	Y	Y	Y	Y	N	N	N	N
RHEL/CentOS 7.4	Y	Y	Y	Y	Y	Y	N	N	N	N
RHEL/CentOS 7.5	Y	Y	Y	Y	Y	Y	N	N	N	N
RHEL/CentOS 7.6	Y	Y	Y	Y	Y	Y	N	N	N	N
RHEL/CentOS 7.7	Y	Y	Y	Y	Y	Y	N	N	N	N
RHEL/CentOS 7.8	N	N	Y	Y	Y	Y	N	N	N	N
RHEL 7.9	N	N	Y	Y	Y	Y	Y	Y	N	N
Centos 7.9	N	N	N	N	N	Y	Y	Y	N	N
RHEL 8.0	N	N	N	N	N	Y	Y	Y	Y	N
RHEL 8.1	N	N	N	N	N	Y	Y	Y	Y	N
RHEL 8.2	N	N	N	N	N	Y	Y	Y	Y	N
RHEL 8.3	N	N	N	N	N	Y	Y	Y	Y	N
RHEL 8.4	N	N	N	N	N	Y	Y	Y	Y	N
RHEL 8.5	N	N	N	N	N	Y	Y	Y	Y	N
RHEL 8.6	N	N	N	N	N	Y	Y	Y	Y	N
RHEL 8.7	N	N	N	N	N	Y	Y	Y	Y	N
RHEL 8.8	N	N	N	N	N	Y	Y	Y	Y	N

RHEL 8.9	N	N	N	N	N	Y	Y	Y	Y	N
RHEL 9.0	N	N	N	N	N	N	N	N	Y	Y
RHEL 9.1	N	N	N	N	N	N	N	N	N	Y
RHEL 9.2	N	N	N	N	N	N	N	N	N	Y
RHEL 9.3	N	N	N	N	N	N	N	N	N	Y
SUSE 11	Y	N	N	N	N	N	N	N	Y	N
SUSE 12	Y	Y	Y	Y	Y	N	N	N	N	N
SUSE 15	N	N	N	N	N	N	Y	N	N	N
SUSE 15SP3	N	N	N	N	N	N	Y	N	N	N
Ubuntu 16.04	N	N	N	Y	N	N	N	N	N	N
Ubuntu 18.04	N	N	N	Y	Y	Y	Y	Y	Y	N

"x" represents the minor version.

- SUSE 15SP3 is certified on Power 8.
- Additionally, ensure that either libncurses.5 or libncurses.6 is installed on the environment to support both RHEL and SUSE. For actionable steps, refer to KBA 5622[528].

9.9.5.1.2 Supported Cloud Platform-as-a-Service (PaaS) database services

The PostgreSQL connector is certified for use with the following cloud PaaS database services and versions.

528 https://support.delphix.com/Continuous_Data_Engine_(formerly_Virtualization_Engine)/Delphix_Admin/Environment_Operations_Fail_With_%22bash%3A_error_while_loading_shared_libraries%3A_libncurses.so.5%3A_cannot_open_shared_object_file%3A_No_such_file_or_directory%22_(KBA5622)

A Cloud PaaS database service cannot be used as staging or target environments due to strict limitations imposed by the cloud vendors. In these cases, the Cloud PaaS database service can be used only as a source, and self-managed hosts, with a supported operating system and equivalent open-source PostgreSQL database versions, must be used for the staging and target environments. We recommend the Delphix Continuous Data Engine, staging, and target environments be hosted in the same Cloud and region as the PaaS database service.

Cloud database service	**Database version**				
	11	**12**	**13**	**14**	**15**
Amazon Aurora PostgreSQL	Y	Y	Y	Y	Y
Amazon RDS PostgreSQL	Y	Y	Y	Y	Y
Azure Flexible Server	Y	Y	Y	Y	N

> Only the single DB ingestion feature is supported for the cloud vendors using PostgreSQL through the pg_dump/pg_restore method with Staging Pull. With Staging Push, other ingestion methods may be possible.

If you wish to combine multiple individual Cloud PaaS databases into a single PostgreSQL instance, it is possible to do so with Staging Push. In this scenario, the individual databases are combined and restored into a single, common staging instance, resulting in one dSource which contains multiple databases, and is accessed via one HTTP port.

9.9.5.1.3 PostgreSQL connector and Delphix Continuous Data Engine compatibility

The PostgreSQL connector versions are certified to work with the following Delphix Continuous Data Engines.

Engine version	**Connector version**																		
	1.4.1	**1.4.2**	**1.4.3**	**1.4.4**	**1.5.0**	**1.5.1**	**2.0.0**	**2.1.0**	**2.1.1**	**3.0.0**	**3.1.0**	**3.2.0**	**4.0.0**	**4.1.0**	**4.1.1**	**4.2.0**	**4.2.1**	**4.2.2**	**4.2.3**

6.0.0.x	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.1.x	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.2.x	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.3.x	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.4.x	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.5.x	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.6.x	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.7.x	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.8.x	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.9.x	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.10.x	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N

6.0.11.x	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.12.x	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.13.x	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N	N
6.0.14.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N
6.0.15.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N
6.0.16.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
6.0.17.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
7.0.0.x	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
8.0.0.x	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
9.0.0.x	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
10.0.0.x	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

11.0.0.x	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
12.0.0.x	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
13.0.0.x	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
14.0.0.x	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
15.0.0.x	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
16.0.0.x	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
17.0.0.x	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
18.0.0.x	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
19.0.0.x	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
20.0.0.x	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
21.0.0.x	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

22.0.0.x	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

For more information, refer to the Delphix End of Support[529] policy to determine whether a connector version is in primary, extended, or legacy support. Contact Delphix customer support with any questions.

9.9.5.2 Requirements for source databases

This topic outlines the requirements for the PostgreSQL source database. The requirements are divided based on the linking methods. For more information about these parameters, refer to the Important PostgreSQL settings (see page 1842) section.

1. Externally Initiated backup with WAL shipping

Source system requirements	Description
The value of **wal_level** must be set to **replica** (or **archive**).	Add an entry to the **postgresql.conf** file to set the **wal_level** value. This is required to have enough transactional information necessary for the staging PostgreSQL instance to support point-in-time recovery. `wal_level = 'replica'  # Default is minimal`

(Optional) The **archive_mode** is set to **on** and the **archive_command** should be defined.	Add entries to the **postgresql.conf** file to set the **archive_mode** or **archive_command** values.
	Enabling archiving is necessary so that WAL files can be shared with the staging environment. If the **archive_mode** is set to **on**, the completed WAL segments are sent to archive storage by setting **archive_command**.
	`archive_mode = 'on'  # Enable Archiving`
	`# Ex: Copy completed WAL files to /u01/archive directory`
	`archive_command = 'cp %p /u01/archive/%f'`

2. PostgreSQL streaming replication with externally and Delphix initiated backup

529 https://support.delphix.com/Support_Policies_and_Technical_Bulletins/Support_Policies/Software_Product_Line_Policies_(KBA1003)#:~:text=Delphix%20may%20discontinue%20a%20product,18%20months%20of%20Extended%20Support

Source system requirements	Description
The source PostgreSQL cluster must not be in a shutdown state.	Configure the source database to be in a read-write, read-only, or standby mode (streaming site).
The staging environment must have access to a PostgreSQL role on the source database that has replication and login privileges.	Create a new or modify an existing user to stream transaction logs from the primary to the standby instance. Avoid using the superuser. `CREATE USER delphix WITH REPLICATION ENCRYPTED [ PASSWORD 'password'];`

The PostgreSQL client and replication connection to the source PostgreSQL instance from the staging environment must be allowed.	Add an entry to the **pg_hba.conf** file of the source PostgreSQL instance to allow replication connections from the staging environment.
	`host all <role> <ip-address_of_staging_target>/0 <auth-method>`
	`host replication <role> <ip-address_of_staging_target>/0 <auth-method>`
	The default location of the pg_hba.conf file is the data directory. However, you may modify the location of this file in the postgresql.conf file.
	The **<auth-method>** can be set to trust, scram-sha-256, md5, etc. on the source, as allowed by PostgreSQL. For more information on how to configure pg_hba.conf, refer to Client authentication[530].

The value of **wal_level** must be set to **replica** (or **archive**).	Add an entry to the **postgresql.conf** file to set the **wal_level** value. This is required to have enough transactional information necessary for the staging PostgreSQL instance to support point-in-time recovery. `wal_level = 'replica'  # Default is minimal`

530 https://cd.delphix.com/docs/latest/certificate-management-for-throughput-tests#id-(15.0.0.0)Certificatemanagementforthroughputtests-Enablingclientauthentication

Source system requirements	Description
The minimum value of **max_wal_senders** must be set to 2.	Add an entry to the **postgresql.conf** file to set the **max_wal_senders** value. The PostgreSQL connector triggers a pg_basebackup of the source PostgreSQL cluster and also sets up streaming replication, so a minimum value of “2” is required. Increase the value by 2 for each additional standby. `max_wal_senders = 2  # Existing value plus (+) 2 for each standby, default is 0`

The value of the **wal_keep_size** (known as **wal_keep_segments** in PostgreSQL 12 and lower) parameter should be large enough to support the WAL sync process.	Add an entry to the **postgresql.conf** file to set the **wal_keep_size** or **wal_keep_segments** values**.**
	If using streaming replication without file-based continuous archiving, the server might recycle old WAL segments before the standby has received them. If it occurs, the standby will need to be resynchronized with a new base backup.
	To avoid this, you must set the **wal_keep_size** to a value large enough to ensure that WAL segments are not recycled too early, or by configuring a replication slot for the standby.
	Example: PostgreSQL 13 and greater
	`wal_keep_size = 20gb`
	Example: PostgreSQL 12 and lower
	`# Equivalent to 20gb of WAL files wal_keep_segments = 1280 # In mb wal_segment_size = 16 # Default value in mb`
	The value for **wal_keep_segments** can be estimated as the average number of new WAL segments per minute in the pg_wal directory, multiplied by the number of minutes across which protection is required. Therefore, **wal_keep_size = wal_keep_segments * wal_segment_size.**

Source system requirements	Description

(Optional) The **archive_mode** is set to **on** and the **archive_command** should be defined.	Add entries to the **postgresql.conf** file to set the **archive_mode** or **archive_command** values.
	Enabling archiving is necessary so that WAL files can be shared with the staging environment. If the **archive_mode** is set to **on**, the completed WAL segments are sent to archive storage by setting **archive_command**.
	`archive_mode = 'on'  # Enable Archiving`
	`# Ex: Copy completed WAL files to /u01/ archive directory`
	`archive_command = 'cp %p /u01/archive/ %f'`

3. Delphix initiated single database backup

Source system requirements	Description
The source PostgreSQL cluster must not be in a shutdown state.	Configure the source database to be in a read-write, read-only, or standby mode (streaming site).
The staging environment must have access to a PostgreSQL role on the source database that has privileges to run pg_dumpall/pg_dump.	That pg_dumpall/pg_dump does not operate with special permissions. In particular, it must have **read access to all tables** that you want to back up, so in order to back up the **entire database** you almost always have to run it as a database **superuser**. `CREATE ROLE delphix SUPERUSER LOGIN PASSWORD 'password';`

Source system requirements	Description

The PostgreSQL client connection to the source PostgreSQL instance from the staging environment must be allowed.	Add entries to the **pg_hba.conf** file of the source PostgreSQL instance to allow connections from the staging environment.
	1. Connection to the database (single_db) that we are taking the pg_dump off.
	2. Connection to the postgres database, if it doesn't exist then connection to the template1 database for pg_dumpall.
	`host <single_db> <role> <ip-address_of_staging_target>/0 <auth-method>`
	`host <postgres/template1> <role> <ip-address_of_staging_target>/0 <auth-method>`
	The default location of the pg_hba.conf file is the data directory. However, you may modify the location of this file in the postgresql.conf file.
	The **<auth-method>** can be set to trust, scram-sha-256, md5, etc. on the source, as allowed by PostgreSQL. For more information on how to configure pg_hba.conf, refer to Client authentication[531].

9.9.5.3 Requirements for staging and target environments

Environments must follow a set of requirements that have been defined in the Minimum Staging and Target environment requirements (see page 1780). Additionally, for operations (1) and (2), sudo privileges for the environment's operating system (OS) user (see page 1780) requirements are necessary.

Optionally, you can configure Privilege Elevation (see page 1780), which allows a low-privilege user to perform higher-privileged operations; improving the environment's security profile. This feature can be added later at any time.

As background, the Delphix Continuous Data Engine connects via SSH to the staging and target environments to perform the following operations:

1. Mount and unmount directories for linking and provisioning operations. These privileges are essential to Delphix functionality; mount and unmount are typically privileged and likely require additional privileges to be granted during installation and configuration.
2. Create and remove directories for linking and provisioning operations.

531 https://cd.delphix.com/docs/latest/certificate-management-for-throughput-tests#id-(15.0.0.0)Certificatemanagementforthroughputtests-Enablingclientauthentication

3. Run a variety of commands requested by the PostgreSQL connector.

9.9.5.3.1 Minimum staging and target environment configuration requirements

Follow these requirements to ensure that your environment is configured accordingly:

Staging/Target environment requirements	Explanation
Environment compatibility check The operating system and architecture of the target environment must match the source environment.	It is required that the source and the target environments have compatible operating systems and hardware configurations. This is to ensure that the database runs correctly.
PostgreSQL installation and compatibility check There must be an installation of PostgreSQL on the staging/target environment that is compatible with the installation of PostgreSQL on the source environment.	Two installations of PostgreSQL are compatible **only if**: 1. They share the same vendor (for example, PostgreSQL is incompatible with EnterpriseDB Postgres Plus Advanced Server). 2. They share the same major version number (for example, 14.4 is compatible with 14.x; however, it is incompatible with 12.x, 13.x, or 15.x).
"Toolkit Path" There must be a directory on the staging/target environment with the following properties: 1. The directory must be writable by the operating system user mentioned above. 2. The directory must have at least 1.5 GB of available storage.	The **"Toolkit Path"** in the Environment creation UI page is the directory on the staging/target environment where the PostgreSQL Connector support files are installed (for example, /opt/toolkit) and the connector logs are located. The Delphix Toolkit facilitates communication between Delphix CD Engine and this specific Environment. Files will be deposited into this path by the Delphix CD Engine during the Environment Discovery operation.
There must be a directory (e.g. "/mnt/provision/") or ("/mnt/staging/") that will be used as a container for the NFS mount points that are created when provisioning a VDB or linking a staging push dSource.	The delphix_os(or delphix) user and primary OS group (i.e. postgres) must own the directory. The directory must have at least permissions -rwxrwx--- (0770). There must be no symbolic links in the path of this directory, because NFS can mount into a directory with symlinks in its path, but cannot unmount.

Staging/Target environment requirements	Explanation
The delphix_os(delphix) user must have Postgres as a primary or secondary OS group. Also, the Postgres socket directory (eg. /mnt/run/postgresql) should have the 775 permission.	Delphix recommends giving the delphix_os(delphix) user the same OS group as the Postgres user. This ensures the delphix_os(delphix) user can access the Postgres binary directory.
Locale en_US.UTF-8 should be available at OS level on staging/target environment.	The PostgreSQL connector uses en_US.UTF-8 LANG settings to capture snapshot metadata in English.

Required utilities and tools
The **netstat** or **ss** utility must be installed on the staging/target environment.
	The PostgreSQL connector checks the status of the running PostgreSQL instance and validates if ports are available on the staging/target environment using the **netstat** or **ss** utility.
	Checking if **netstat** or **ss** utility is installed with:
	`which netstat`
	`which ss`

Zip and unzip must be installed on the staging/target environment for external backup support.
Note: This is required if providing the source backup to the PostgreSQL Connector as a ZIP file.
	PostgreSQL connector requires zip and unzip utility to unzip archived external backup files.
	Check if zip and unzip utility is installed with:
	`which zip`
	`which unzip`

The DELPHIX_PG_PATH must be set if the PostgreSQL installation you want to discover is in a shutdown state.	The following syntax should be used for the DELPHIX_PG_PATH environment variable: `DELPHIX_PG_PATH="<postgres_binary_path1>:<data_path1>;<postgres_binary_path2>:<data_path2>"` For more information, refer to the Discovering installations (see page 1796) sections.

Staging/Target environment requirements	Explanation

(Optional) Hostname and IP may need to be set in /etc/hosts file to point to the correct IP address.	The output of the “hostname -i” command should produce the correct result as the IP address of the server.
	For example:
	`[postgres@source postgres]$ hostname -i`
	`10.110.207.113`

9.9.5.3.2 Sudo privileges for the environment users

An environment user account on the environment must be selected to communicate with both the staging and target environments. As required by the PostgreSQL connector, you must grant this environment user the following two key privileges:

1. **Disable requiretty:** The Delphix Continuous Data Engine requires the `requiretty` setting to be disabled. This allows the environment user to run sudo commands without a logged-in terminal session (tty).
2. **Add NOPASSWD:** By including the NOPASSWD qualifier, you ensure that the "sudo" command does not prompt for a password when executing certain commands such as mount, umount, mkdir, and rmdir.

In the example configurations, the environment username **postgres** is used. Make sure to modify the username based on the environment user available in the environment for performing the linking and/or provisioning operations.

9.9.5.3.2.1 Procedure

Perform the following steps on both the staging and target environments.

1. The sudo configuration is located in the `/etc/sudoers` file. Run the following commands to open the `sudoers’ file:

   ```
   vi /etc/sudoers
   ```

> Third party tools like `visudo`, can be helpful in minimizing syntax errors.

2. Next, update the file with the following snippet to grant the required privileges to the postgres` user:

```
Defaults:postgres !requiretty
postgres ALL=NOPASSWD:  /bin/mount,/bin/umount,/bin/mkdir,/bin/rmdir
```

9.9.5.3.2.2 Additional examples

The examples below demonstrate how a user's privileges can be further refined. Note that these examples are for illustrative purposes only and are subject to change.

Example one

This example restricts the use of sudo privileges to the `/postgres` directory.

Note that wildcards are allowed for the options with mount and umount commands because those commands expect a fixed number of arguments after the options. The option wildcard on the mount command also enables you to specify the file-system being mounted from the Delphix Continuous Data Engine. However, wildcards are not allowed with mkdir and rmdir because they can have any number of arguments after the options. For those commands, you must specify the exact options (-p, -p -m 755) used by the Delphix Continuous Data Engine.

Example: `/etc/sudoers` *file configuration on the target environment for sudo privileges on the VDB mount directory only (Linux OS)*

```
Defaults:postgres !requiretty
delphix_os ALL=(root) NOPASSWD: \
/bin/mount  *        /postgres/*, \
/bin/umount *        /postgres/*, \
/bin/umount          /postgres/*, \
/bin/mkdir -p        /postgres/*, \
/bin/mkdir -p -m 755 /postgres/*, \
/bin/mkdir           /postgres/*, \
/bin/rmdir           /postgres/*
```

Example two

This example restricts use of sudo privileges to the /postgres directory, restricts the mount commands to a specific Delphix Continuous Data Engine hostname and IP, and does not allow user-specified options for the umount command. This configuration is more secure, but there is a tradeoff with deployment simplicity. This approach would require a different sudo configuration for targets configured for different Delphix Continuous Data Engines.

Example: *Configuring the /etc/sudoers file on the target environment for privileges on the VDB mount directory only (Linux OS)*

```
Defaults:delphix_os !requiretty
delphix_os ALL=(root) NOPASSWD: \
/bin/mount           <delphix-server-name>* /postgres/*, \
/bin/mount *         <delphix-server-name>* /postgres/*, \
/bin/mount           <delphix-server-ip>*   /postgres/*, \
/bin/mount *         <delphix-server-ip>*   /postgres/*, \
/bin/mount "", \
```

```
/bin/umount          /postgres/*, \
/bin/umount *        /postgres/*, \
/bin/mkdir [*]       /postgres/*, \
/bin/mkdir           /postgres/*, \
/bin/mkdir -p        /postgres/*, \
/bin/mkdir -p -m 755 /postgres/*, \
/bin/rmdir           /postgres/*
```

9.9.5.3.3 Privilege elevation requirements

The optional Privilege Elevation feature allows a non-privileged environment user (e.g., delphix_os) to gain the permissions of a privileged environment user (e.g., postgres) for operations on the staging or target environments. This configuration can be valuable to maximize an environment's security posture.

9.9.5.3.3.1 Procedure

First, the privilege elevation script, dlpx_db_exec, must be created and uploaded onto the Delphix Continuous Data Engine. Then, the non-privileged environment user must be granted the appropriate permissions.

1. Upload the privilege elevation script.

Perform one of the following methods to upload the privilege elevation script:

a) CURL file upload method

Before running any of the following commands, update *<CONTINUOUS-DATA-FQDN-or-IP>*, *<USERNAME>*, and *<PASSWORD>* placeholder values.

1. Create a session with the Delphix Continuous Data Engine.
    a. Also, update the version values for `major`, `minor`, and `micro` fields.
        i. The correct values for your Delphix Continuous Data Engine version can be found in the API version information[532].

```
curl -i -c cookies.txt -X POST -H "Content-Type:application/json"
 https://<CONTINUOUS-DATA-FQDN-or-IP>/resources/json/delphix/
session -d '{
    "version":{
        "major":1,
        "minor":11,
        "micro": 5,
        "type":"APIVersion"
    },
    "type":"APISession"
}'
```

2. Login to Delphix Continuous Data Engine as the `*admin*` user.

532 https://cd.delphix.com/docs/latest/api-version-information

```
curl -i -c cookies.txt -b cookies.txt -X POST -H "Content-Type:application/
json" https://<CONTINUOUS-DATA-FQDN-or-IP>/resources/json/delphix/login -d '{
    "username":"<USERNAME>",
    "password":"<PASSWORD>",
    "type":"LoginRequest",
    "target":"DOMAIN"
}'
```

3. Copy *DLPX_DB_EXEC* contents to the Delphix Continuous Data Engine.

```
curl -i -b cookies.txt -X POST -H "Content-Type:application/json" https://
<CONTINUOUS-DATA-FQDN-or-IP>/resources/json/delphix/host/privilegeElevation/
profileScript/HOST_PRIVILEGE_ELEVATION_PROFILE_SCRIPT-7 -d '{
    "type": "HostPrivilegeElevationProfileScript",
    "contents": "#\n# Copyright (c) 2018 by Delphix. All rights reserved.
\n#\n#\n# This script allows customization of command execution with an
alternate user\n# account.\nif [[ $1 != -u* ]]; then\n    echo \"Incorrect
command line parameters, -u<optional user account> is required as the first
parameter\"\n    exit 1\nfi\nuser_id=`echo $1 | sed -e \"s\/^-u\/\/\"`\nshift
1\nif [[ $user_id != \"delphix_os\" ]]; then\n    command=$(printf \"%s\"
\"$@\")\n    cd /tmp\n    sudo -E su $user_id -p -c \"$command\"\nelse\n
$@\nfi\n"
}'
```

If the *dlpx_db_exec* script is updated after the environment(s) is added, refresh the environment to propagate the changes.

b) CLI method

For steps on creating a Privilege Elevation Profile, refer to [CLI Cookbook: How to create or edit privilege elevation profiles and profile scripts](533).

1. Configure sudo privileges to grant Privilege Elevation

With Privilege Elevation, all PostgreSQL connector commands are run via the **dlpx_db_exec** script, and within this script, the commands are run using 'sudo'. In order to execute the connector commands as the high-privileged environment user, sudoers entries are required.

To add additional sudoers entries you need to edit the **/etc/sudoers** file. Run the following commands to open the `sudoers' file:

533 https://cd.delphix.com/docs/latest/cli-cookbook-how-to-create-or-edit-a-privilege-ele

```
vi /etc/sudoers
# or
visudo
```

> Third party tools like `visudo`, can be helpful in minimizing syntax errors.

Templates for different connector versions are given below. These examples assume the following:

- The Operating System is Linux
- The privileged environment user is named “postgres”
- The low privileged environment user is named “delphix_os”

You must update the `privileged environment user` and the `low privileged environment user` as per your environment.

Entries required for both linking and provisioning via low-privileged user (delphix_os):

Connector Version >= 4.1.0 (as a drop-down)

```
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD:SETENV: /bin/mount,\
/bin/umount,\
/bin/mkdir,\
/bin/rmdir,\
/bin/ps,\
/bin/su postgres -p -c '*'/createdb -p * '*',\
/bin/su postgres -p -c '*'/dropdb -p * '*',\
/bin/su postgres -p -c '*'/pg_dump -Fd '*' -p * -j * -h '*' -f '*/*_backup' -U * --
verbose,\
/bin/su postgres -p -c '*'/pg_dumpall --globals-only --no-role-password --clean -p *
-h '*' -f '*/*dumpall_file.sql' -U *,\
/bin/su postgres -p -c '*'/pg_restore -p * -d '*' -j * '*/*_backup' --verbose,\
/bin/su postgres -p -c */pg_basebackup -D */data -F t -v -w * -p * -U * --
checkpoint\=fast,\
/bin/su postgres -p -c */pg_controldata */data,\
/bin/su postgres -p -c */pg_ctl --version,\
/bin/su postgres -p -c */pg_ctl initdb -D */data,\
/bin/su postgres -p -c */pg_ctl initdb -D */data -o '*',\
/bin/su postgres -p -c */pg_ctl initdb -D */postgres_init,\
/bin/su postgres -p -c */pg_ctl initdb -D */postgres_init -o '*',\
/bin/su postgres -p -c */pg_ctl start -D */data -t * &> */.delphix/postgres_db_log,\
/bin/su postgres -p -c */pg_ctl status -D */data,\
/bin/su postgres -p -c */pg_ctl stop -D */data &> */.delphix/postgres_db_log,\
/bin/su postgres -p -c */pg_isready -h * -p *,\
/bin/su postgres -p -c */psql -t -U * -p * -d * -h * -c 'select 1;',\
/bin/su postgres -p -c */psql -p * -d postgres -f '*/dumpall_file.sql',\
/bin/su postgres -p -c */psql -t -U * -p * -d * -h * -c 'select name\, setting from
pg_settings;' -o */.delphix/source_postgresql_config_file.conf,\
```

```
/bin/su postgres -p -c */psql -t -U * -p * -d * -h * -c 'show server_version;',\
/bin/su postgres -p -c base64 -d $DLPX_PG_PLUGIN_ECHO_PATH/'database_oid' >
*/.delphix/database_oid,\
/bin/su postgres -p -c base64 -d $DLPX_PG_PLUGIN_ECHO_PATH/'ingestion_method' >
*/.delphix/ingestion_method,\
/bin/su postgres -p -c base64 -d $DLPX_PG_PLUGIN_ECHO_PATH/'postgresql_tmp.conf' > */
data/postgresql_tmp.conf,\
/bin/su postgres -p -c base64 -d $DLPX_PG_PLUGIN_ECHO_PATH/'recovery.conf' >> */data/
recovery.conf,\
/bin/su postgres -p -c base64 -d $DLPX_PG_PLUGIN_ECHO_PATH/'tablespace_file' >
*/.delphix/tablespace_file,\
/bin/su postgres -p -c base64 -d $DLPX_PG_PLUGIN_ECHO_PATH/'walControl.sh' > */data/
scripts/walControl.sh,\
/bin/su postgres -p -c cat '*/.delphix/postgres_db_log',\
/bin/su postgres -p -c cat '*/data/PG_VERSION',\
/bin/su postgres -p -c cat '*/data/current_logfiles',\
/bin/su postgres -p -c cat '*/data/postgresql.conf',\
/bin/su postgres -p -c cat '*/dumpall_file.sql',\
/bin/su postgres -p -c cat '*/scratch_file.sql',\
/bin/su postgres -p -c cat '*/source_postgresql_config_file.conf',\
/bin/su postgres -p -c chmod 0700 '*/data',\
/bin/su postgres -p -c chmod 0700 '*/data/recovery.conf.delphix',\
/bin/su postgres -p -c chmod 0755 '*/data/scripts/walControl.sh',\
/bin/su postgres -p -c chmod 0666 '*/scratch_file.sql',\
/bin/su postgres -p -c cp '*' '*/data/postgresql.conf',\
/bin/su postgres -p -c cp '*/data/postgresql.conf' '*/data/postgresql.conf_backup',\
/bin/su postgres -p -c cp '*/dumpall_file.sql' '*/scratch_file.sql',\
/bin/su postgres -p -c echo '' > '*/data/postgresql.auto.conf',\
/bin/su postgres -p -c echo '' > '*/data/recovery.conf',\
/bin/su postgres -p -c echo '' > '*/data/standby.signal',\
/bin/su postgres -p -c echo '' > '*/data/recovery.signal',\
/bin/su postgres -p -c echo '' > '*/.delphix/staging_push',\
/bin/su postgres -p -c find * -type f -name PostgreSQL_T\[0-9\]\[0-9\]\[0-9\]\[0-9\]\
[0-9\]\[0-9\]\[0-9\]\[0-9\].zip,\
/bin/su postgres -p -c find * -type d -name PostgreSQL_T\[0-9\]\[0-9\]\[0-9\]\[0-9\]\
[0-9\]\[0-9\]\[0-9\]\[0-9\]\[0-9\]\[0-9\]\[0-9\]\[0-9\]\[0-9\]\[0-9\],\
/bin/su postgres -p -c find * -type f -name base.tar,\
/bin/su postgres -p -c find */data -type f -name \*.tar,\
/bin/su postgres -p -c find * -type f -name \[0-9\]\*tar,\
/bin/su postgres -p -c grep -w '^log_directory' '*/data/postgresql.conf',\
/bin/su postgres -p -c grep -w '^max_connections' '*/data/postgresql.auto.conf',\
/bin/su postgres -p -c grep -w '^max_connections' '*/data/postgresql_source.conf',\
/bin/su postgres -p -c grep -w '^max_wal_senders' '*/data/postgresql.auto.conf',\
/bin/su postgres -p -c grep -w '^max_wal_senders' '*/data/postgresql_source.conf',\
/bin/su postgres -p -c grep -w '^port' '*/data/postgresql.auto.conf',\
/bin/su postgres -p -c grep -w '^port' '*/data/postgresql.conf',\
/bin/su postgres -p -c grep -w '^wal_keep_segments' '*/data/postgresql.auto.conf',\
/bin/su postgres -p -c grep -w '^wal_keep_segments' '*/data/postgresql_source.conf',\
/bin/su postgres -p -c grep -w '^wal_keep_size' '*/data/postgresql.auto.conf',\
/bin/su postgres -p -c grep -w '^wal_keep_size' '*/data/postgresql_source.conf',\
/bin/su postgres -p -c grep -w '^timezone' '*/data/postgresql.conf',\
/bin/su postgres -p -c ln -s '../tablespace/*' '*/data/pg_tblspc',\
/bin/su postgres -p -c ls -lrt '*' | tail -n+2,\
```

```
/bin/su postgres -p -c mkdir -p '*',\
/bin/su postgres -p -c du -sb '*',\
/bin/su postgres -p -c mv '*/data/postgresql.auto.conf' '*/data/
postgresql.auto.conf.delphix',\
/bin/su postgres -p -c mv '*/data/postgresql.auto.conf' '*/data/
postgresql.auto.source.conf',\
/bin/su postgres -p -c mv '*/data/postgresql.conf' '*/data/postgresql_source.conf',\
/bin/su postgres -p -c mv '*/data/postgresql_tmp.conf' '*/data/postgresql.conf',\
/bin/su postgres -p -c mv '*/data/postmaster.opts' '*/data/postmaster.opts.delphix',\
/bin/su postgres -p -c mv '*/data/recovery.conf' '*/data/recovery.conf.delphix',\
/bin/su postgres -p -c mv '*/data/standby.signal' '*/data/standby.signal.delphix',\
/bin/su postgres -p -c mv '*/postgres_init/postgresql.auto.conf' '*/data',\
/bin/su postgres -p -c mv '*/postgres_init/postgresql.conf' '*/data',\
/bin/su postgres -p -c mv '*/postgres_init/pg_hba.conf' '*/data',\
/bin/su postgres -p -c mv '*/postgres_init/pg_ident.conf' '*/data',\
/bin/su postgres -p -c mv '*/scratch_file.sql' '*/dumpall_file.sql',\
/bin/su postgres -p -c printenv param_value >> */data/recovery.conf,\
/bin/su postgres -p -c rm -f '*/data/*.tar',\
/bin/su postgres -p -c rm -f '*/data/postmaster.pid',\
/bin/su postgres -p -c rm -f '*/data/tablespace_map',\
/bin/su postgres -p -c rm -r -f '*',\
/bin/su postgres -p -c tail -n 1 '*/data/scripts/WalBreakChainDetected',\
/bin/su postgres -p -c tail -n 1 '*/data/scripts/InvalidRecordFileFoundData',\
/bin/su postgres -p -c tail -n * '*/data/log/*',\
/bin/su postgres -p -c tail -n * '*/data/pg_log/*',\
/bin/su postgres -p -c tar -xf */base.tar -C */data,\
/bin/su postgres -p -c tar -xf */pg_wal.tar -C */data/pg_wal,\
/bin/su postgres -p -c tar -xf */*.tar -C */data/tablespace*,\
/bin/su postgres -p -c test -d '*',\
/bin/su postgres -p -c test -f '*',\
/bin/su postgres -p -c test -r '*' && test -w '*' && test -x '*',\
/bin/su postgres -p -c test -r '*' && test -x '*',\
/bin/su postgres -p -c test -r '*',\
/bin/su postgres -p -c test -w '*' && test -x '*',\
/bin/su postgres -p -c unzip '*/PostgreSQL_T[0-9][0-9][0-9][0-9][0-9][0-9][0-9]
[0-9].zip' -d '*/data',\
/bin/su postgres -p -c zipinfo* '*/PostgreSQL_T[0-9][0-9][0-9][0-9][0-9][0-9][0-9]
[0-9].zip'
```

3.1.0 <= Connector Version < 4.1.0 (as a drop-down)

```
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD:SETENV: /bin/mount,\
/bin/umount,\
/bin/mkdir,\
/bin/rmdir,\
/bin/ps,\
/bin/su postgres -p -c '*'/createdb -p * '*',\
/bin/su postgres -p -c '*'/dropdb -p * '*',\
/bin/su postgres -p -c '*'/pg_dump -Fd '*' -p * -j * -h '*' -f '*/*_backup' -U * --
verbose,\
```

```
/bin/su postgres -p -c '*'/pg_dumpall --globals-only --no-role-password --clean -p *
-h '*' -f '*/*dumpall_file.sql' -U *,\
/bin/su postgres -p -c '*'/pg_restore -p * -d '*' -j * '*/*_backup' --verbose,\
/bin/su postgres -p -c */pg_basebackup -D */data -F t -v -w * -p * -U * --
checkpoint\=fast,\
/bin/su postgres -p -c */pg_controldata */data,\
/bin/su postgres -p -c */pg_ctl --version,\
/bin/su postgres -p -c */pg_ctl initdb -D */data,\
/bin/su postgres -p -c */pg_ctl initdb -D */postgres_init,\
/bin/su postgres -p -c */pg_ctl start -D */data -t * &> */.delphix/postgres_db_log,\
/bin/su postgres -p -c */pg_ctl status -D */data,\
/bin/su postgres -p -c */pg_ctl stop -D */data &> */.delphix/postgres_db_log,\
/bin/su postgres -p -c */pg_isready -h * -p *,\
/bin/su postgres -p -c */psql -t -U * -p * -d * -h * -c 'select 1;',\
/bin/su postgres -p -c */psql -p * -d postgres -f '*/dumpall_file.sql',\
/bin/su postgres -p -c */psql -t -U * -p * -d * -h * -c 'select name\, setting from
pg_settings;' -o */.delphix/source_postgresql_config_file.conf,\
/bin/su postgres -p -c */psql -t -U * -p * -d * -h * -c 'show server_version;',\
/bin/su postgres -p -c */psql -t -U * -p * -c "select
SUM(pg_database_size(pg_database.datname)) from pg_database;",\
/bin/su postgres -p -c */psql -t -U * -p * -d * -c "select
pg_database_size(pg_database.datname) from pg_database where pg_database.datname\=
'*';",\
/bin/su postgres -p -c base64 -d $DLPX_PG_PLUGIN_ECHO_PATH/'database_oid' >
*/.delphix/database_oid,\
/bin/su postgres -p -c base64 -d $DLPX_PG_PLUGIN_ECHO_PATH/'ingestion_method' >
*/.delphix/ingestion_method,\
/bin/su postgres -p -c base64 -d $DLPX_PG_PLUGIN_ECHO_PATH/'postgresql_tmp.conf' > */
data/postgresql_tmp.conf,\
/bin/su postgres -p -c base64 -d $DLPX_PG_PLUGIN_ECHO_PATH/'recovery.conf' >> */data/
recovery.conf,\
/bin/su postgres -p -c base64 -d $DLPX_PG_PLUGIN_ECHO_PATH/'tablespace_file' >
*/.delphix/tablespace_file,\
/bin/su postgres -p -c base64 -d $DLPX_PG_PLUGIN_ECHO_PATH/'walControl.sh' > */data/
scripts/walControl.sh,\
/bin/su postgres -p -c cat '*/.delphix/postgres_db_log',\
/bin/su postgres -p -c cat '*/data/PG_VERSION',\
/bin/su postgres -p -c cat '*/data/current_logfiles',\
/bin/su postgres -p -c cat '*/data/postgresql.conf',\
/bin/su postgres -p -c cat '*/dumpall_file.sql',\
/bin/su postgres -p -c cat '*/scratch_file.sql',\
/bin/su postgres -p -c chmod 0700 '*/data',\
/bin/su postgres -p -c chmod 0700 '*/data/recovery.conf.delphix',\
/bin/su postgres -p -c chmod 0755 '*/data/scripts/walControl.sh',\
/bin/su postgres -p -c chmod 0666 '*/scratch_file.sql',\
/bin/su postgres -p -c cp '*' '*/data/postgresql.conf',\
/bin/su postgres -p -c cp '*/data/postgresql.conf' '*/data/postgresql.conf_backup',\
/bin/su postgres -p -c cp '*/dumpall_file.sql' '*/scratch_file.sql',\
/bin/su postgres -p -c echo '' > '*/data/postgresql.auto.conf',\
/bin/su postgres -p -c echo '' > '*/data/recovery.conf',\
/bin/su postgres -p -c echo '' > '*/data/standby.signal',\
/bin/su postgres -p -c echo '' > '*/data/recovery.signal',\
/bin/su postgres -p -c echo '' > '*/.delphix/staging_push',\
```

```
/bin/su postgres -p -c find * -type f -name PostgreSQL_T\[0-9\]\[0-9\]\[0-9\]\[0-9\]\
[0-9\]\[0-9\]\[0-9\]\[0-9\].zip,\
/bin/su postgres -p -c find * -type d -name PostgreSQL_T\[0-9\]\[0-9\]\[0-9\]\[0-9\]\
[0-9\]\[0-9\]\[0-9\]\[0-9\]\[0-9\]\[0-9\]\[0-9\]\[0-9\]\[0-9\]\[0-9\],\
/bin/su postgres -p -c find * -type f -name base.tar,\
/bin/su postgres -p -c find */data -type f -name \*.tar,\
/bin/su postgres -p -c find * -type f -name \[0-9\]\*tar,\
/bin/su postgres -p -c grep -w '^log_directory' '*/data/postgresql.conf',\
/bin/su postgres -p -c grep -w '^max_connections' '*/data/postgresql.auto.conf',\
/bin/su postgres -p -c grep -w '^max_connections' '*/data/postgresql_source.conf',\
/bin/su postgres -p -c grep -w '^max_wal_senders' '*/data/postgresql.auto.conf',\
/bin/su postgres -p -c grep -w '^max_wal_senders' '*/data/postgresql_source.conf',\
/bin/su postgres -p -c grep -w '^port' '*/data/postgresql.auto.conf',\
/bin/su postgres -p -c grep -w '^port' '*/data/postgresql.conf',\
/bin/su postgres -p -c grep -w '^wal_keep_segments' '*/data/postgresql.auto.conf',\
/bin/su postgres -p -c grep -w '^wal_keep_segments' '*/data/postgresql_source.conf',\
/bin/su postgres -p -c grep -w '^wal_keep_size' '*/data/postgresql.auto.conf',\
/bin/su postgres -p -c grep -w '^wal_keep_size' '*/data/postgresql_source.conf',\
/bin/su postgres -p -c grep -w '^timezone' '*/data/postgresql.conf',\
/bin/su postgres -p -c ln -s '../tablespace/*' '*/data/pg_tblspc',\
/bin/su postgres -p -c ls -lrt '*' | tail -n+2,\
/bin/su postgres -p -c mkdir -p '*',\
/bin/su postgres -p -c mv '*/data/postgresql.auto.conf' '*/data/
postgresql.auto.conf.delphix',\
/bin/su postgres -p -c mv '*/data/postgresql.auto.conf' '*/data/
postgresql.auto.source.conf',\
/bin/su postgres -p -c mv '*/data/postgresql.conf' '*/data/postgresql_source.conf',\
/bin/su postgres -p -c mv '*/data/postgresql_tmp.conf' '*/data/postgresql.conf',\
/bin/su postgres -p -c mv '*/data/postmaster.opts' '*/data/postmaster.opts.delphix',\
/bin/su postgres -p -c mv '*/data/recovery.conf' '*/data/recovery.conf.delphix',\
/bin/su postgres -p -c mv '*/data/standby.signal' '*/data/standby.signal.delphix',\
/bin/su postgres -p -c mv '*/postgres_init/postgresql.auto.conf' '*/data',\
/bin/su postgres -p -c mv '*/postgres_init/postgresql.conf' '*/data',\
/bin/su postgres -p -c mv '*/postgres_init/pg_hba.conf' '*/data',\
/bin/su postgres -p -c mv '*/postgres_init/pg_ident.conf' '*/data',\
/bin/su postgres -p -c mv '*/scratch_file.sql' '*/dumpall_file.sql',\
/bin/su postgres -p -c printenv param_value >> */data/recovery.conf,\
/bin/su postgres -p -c rm -f '*/data/*.tar',\
/bin/su postgres -p -c rm -f '*/data/postmaster.pid',\
/bin/su postgres -p -c rm -f '*/data/tablespace_map',\
/bin/su postgres -p -c rm -r -f '*',\
/bin/su postgres -p -c tail -n 1 '*/data/scripts/WalBreakChainDetected',\
/bin/su postgres -p -c tail -n 1 '*/data/scripts/InvalidRecordFileFoundData',\
/bin/su postgres -p -c tail -n * '*/data/log/*',\
/bin/su postgres -p -c tail -n * '*/data/pg_log/*',\
/bin/su postgres -p -c tar -xf */base.tar -C */data,\
/bin/su postgres -p -c tar -xf */pg_wal.tar -C */data/pg_wal,\
/bin/su postgres -p -c tar -xf */*.tar -C */data/tablespace*,\
/bin/su postgres -p -c test -d '*',\
/bin/su postgres -p -c test -f '*',\
/bin/su postgres -p -c test -r '*' && test -w '*' && test -x '*',\
/bin/su postgres -p -c test -r '*' && test -x '*',\
/bin/su postgres -p -c test -r '*',\
```

```
/bin/su postgres -p -c test -w '*' && test -x '*',\
/bin/su postgres -p -c unzip '*/PostgreSQL_T[0-9][0-9][0-9][0-9][0-9][0-9][0-9]
[0-9].zip' -d '*/data',\
/bin/su postgres -p -c zipinfo* '*/PostgreSQL_T[0-9][0-9][0-9][0-9][0-9][0-9][0-9]
[0-9].zip'
```

The dlpx_db_exec script

The formatted dlpx_db_exec script is presented here for completeness.

```
#!/bin/sh
#
# This script allows customization of command execution with an              #
alternate user account.
if [[ $1 != -u* ]]; then
    echo "Incorrect command line parameters, -u <optional user account> is required
as the first parameter"
    exit 1
fi
user_id=`echo $1 | sed -e "s/^-u//"`

shift 1
if [[ $user_id != "delphix_os" ]]; then
    command=$(printf "%s" "$@")
    cd /tmp
    sudo -E su $user_id -p -c "$command"
else
    $@
fi
```

9.9.6 PostgreSQL installation and upgrade

The PostgreSQL connector is managed through the Delphix Continuous Data Engine's plugin framework. Plugins are distributed separately from the engine. The plugin must be installed before PostgreSQL Environments can be added, and before attempting any PostgreSQL virtualization operations.

9.9.6.1 Downloading the connector

Perform the following steps to download the PostgreSQL connector to your Desktop system.

1. Log in to the Delphix Download Portal[534] using your credentials from the Delphix Support Portal.
2. Find the PostgreSQL plugin folder by navigating to the "Delphix Product Releases / Continuous Data Connectors / PostgreSQL" path and selecting the desired version's folder.
3. Download the .zip plugin file by clicking "Plugin_postgreSQL_<version>.zip".

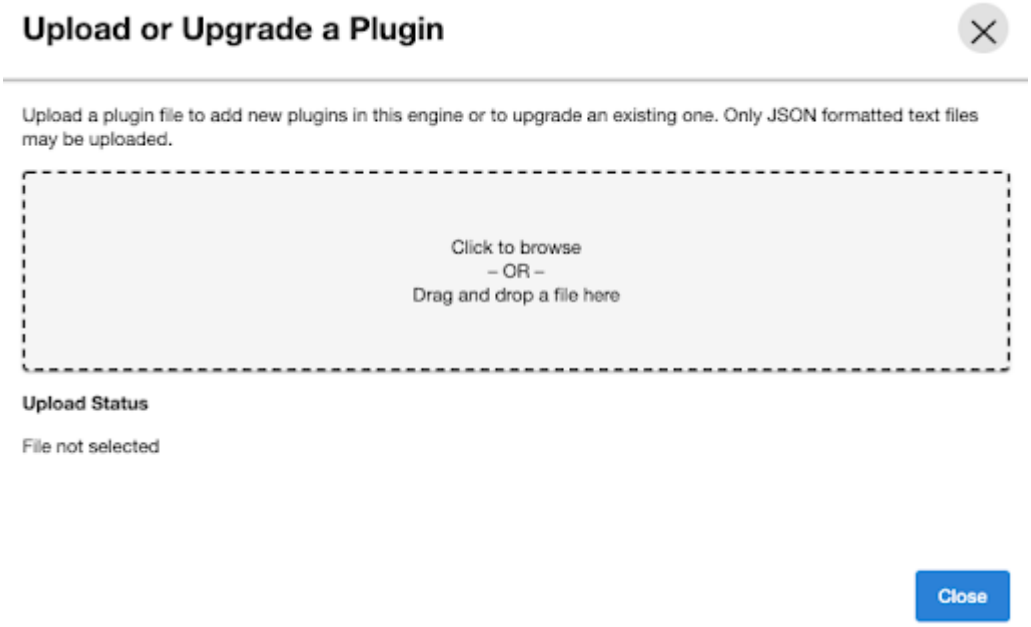
534 https://download.delphix.com/

4. After the download is complete, extract the contents of the downloaded .zip file to obtain a JSON file. For example, postgres.json.

9.9.6.2 Installing the connector

Perform the following steps to upload and install a new connector.

1. Log in to the Delphix Management application.
2. Navigate to the **Manage** > **Plugins** page.
3. To upload a plugin, click the plus icon. This action displays a "**Upload or Upgrade a Plugin**" dialog window.
4. Browse to the JSON file on your desktop which you downloaded and extracted in the previous step
5. Upload the downloaded JSON (for example, postgres.json) file.



For more information, refer to the [Delphix Continuous Data Engine connector management](#)[535] section.

9.9.6.3 **Upgrading the connector**

Users must upgrade their connectors to use new features and apply bug fixes.

Connectors are not upgraded through engine upgrades. However, some connector upgrades are only compatible with a minimum Delphix Continuous Data Engine version. Please ensure your engine is upgraded to a compatible version before upgrading the connector. For full compatibility details, refer to the [PostgreSQL support matrix](#)[536].

535 https://cd.delphix.com/docs/latest/delphix-engine-plugin-management
536 https://cd.delphix.com/docs/latest/postgresql-support-matrix

Perform the following process to upgrade your PostgreSQL connector to a later version.

Warning: Since v2.0.0, connector upgrades have been streamlined. However, skipping a version upgrade step before upgrading to v2.0.0 may lead to configuration issues. If upgrading from a pre-v2.0.0, follow the Upgrade matrix (see page 0)instructions closely.

9.9.6.3.1 Upgrade process

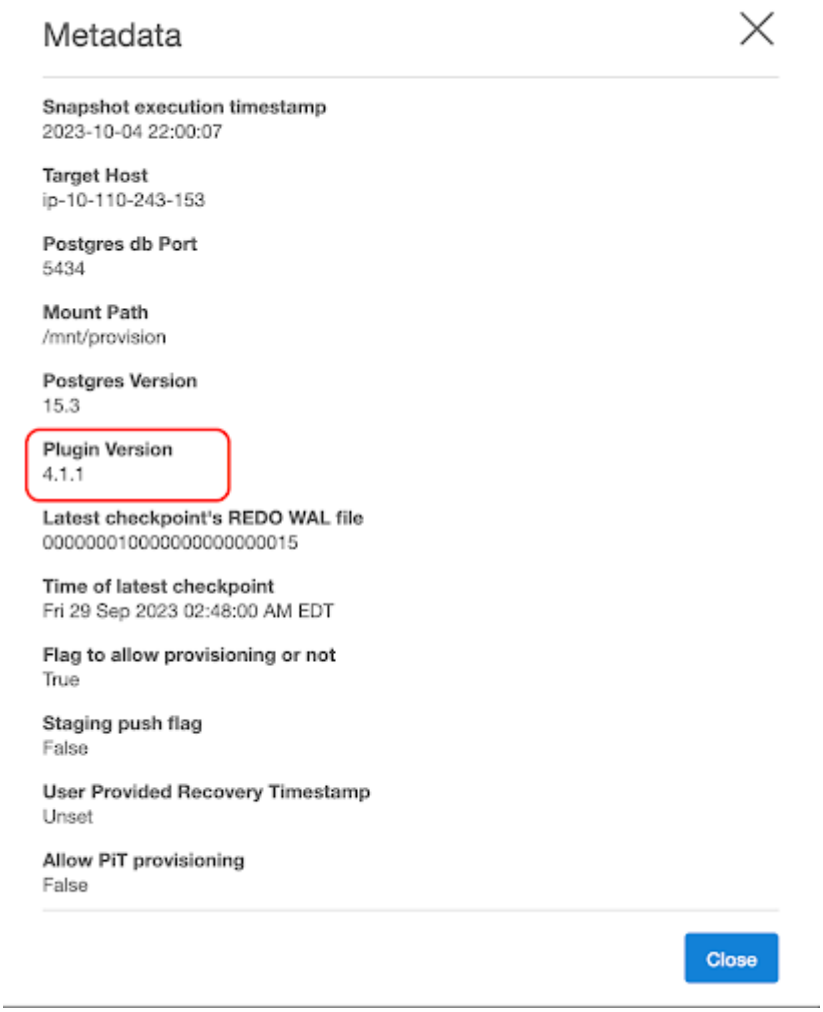
Perform the following steps when upgrading your PostgreSQL connector.

1. Disable all PostgreSQL dSources and VDBs. (Only required for upgrades of Postgres connector version lower than v2.0.0).

Note: Consult the Upgrade matrix (see page 0)instructions for more details.

2. Upload the newer connector version into the Delphix Continuous Data Engine. For more information, refer to the Installing the PostgreSQL connector[537] documentation.
3. Refresh all the associated PostgreSQL environments to trigger the discovery workflow. Verify that the workflow is completed.
4. Enable dSources and VDBs, if disabled, and create new dSource and VDB snapshots.
5. Review the newly created snapshot's metadata for the new plugin version to confirm the upgrade was successful.

537 https://cd.delphix.com/docs/latest/postgresql-installation-and-upgrade



9.9.6.3.2 Upgrade matrix

This section lists PostgreSQL connector versions that a user can upgrade from and to.

> Warning! You must adhere to the following upgrade considerations. Ignoring the below upgrade matrix may result in incompatible PostgreSQL data sources.
> - It is mandatory to upgrade to the v1.5.x of the PostgreSQL connector and then upgrade to only v2.0.0. of the PostgreSQL connector.
> - After upgrading to v2.0.0, you may upgrade directly to any higher version.

Follow the matrix below to avoid connector corruption:

Your installed version	Required upgrade path	dSource and VDB Downtime Required?
v1.0.1, 1.0.2	Upgrade to v1.1.0.	Yes
v1.1.0	Upgrade to v1.2.0.	Yes
v1.2.0	Upgrade to v1.3.0	Yes
v1.3.0	Upgrade to v1.4.4	Yes
v1.4.1, v1.4.2, v1.4.3, and v1.4.4	Upgrade to v1.5.1	Yes
v1.5.0 and v1.5.1	Upgrade to v2.0.0	Yes
v2.0.0+ (vSDK)	Upgrade to any more recently released version.	No

9.9.7 PostgreSQL connector operations

To successfully ingest and virtualize a PostgreSQL instance, you must perform the following connector operations in sequence:

1. Discovering PostgreSQL installations (see page 1796)
2. Linking PostgreSQL data sources (see page 1802)
3. Provisioning PostgreSQL virtual databases (see page 1828)

The respective sections provide detailed information about prerequisites, limitations (if any), and the procedure to be followed for completing each of the operations.

9.9.7.1 Discovering PostgreSQL installations

Discovery is a process that allows the PostgreSQL connector to gather information about the PostgreSQL installation details in an environment. The database's discovery typically occurs as part of the initial environment setup process.

During the discovery process, PostgreSQL connector performs the following steps:

1. Identifies the installations from the running PostgreSQL instances on the host.
2. Identifies the installations provided by the user via the environment variable DELPHIX_PG_PATH. Check the below contents to know more about setting the DELPHIX_PG_PATH variable.

3. If the DELPHIX_PG_PATH is not set, then the connector does a brute-force search for installations in the following directories: /var, /opt, and /usr.

To optimize performance, it is recommended to create the environment variable DELPHIX_PG_PATH. The variable must be available to the environment user in a non-interactive way. You can test this variable for non-interactive logins using:

```
ssh <environment_user>@<host_ip> "env | grep DELPHIX_PG_PATH"
```

The following syntax should be used for the DELPHIX_PG_PATH environment variable:

```
DELPHIX_PG_PATH="<postgres_binary_path1>:<data_path1>;
<postgres_binary_path2>:<data_path2>"
```

For example:

```
DELPHIX_PG_PATH="/u01/pgsql-9.6/bin:/var/lib/pgsql/9.6/data;
/u02/edb/as9.6/bin:/opt/edb/as9.6/data; /u05/pgsql-9.6/bin:/tmp/TESTING/data;"
```

Whenever there is any modification, such as installing a new database home, to an environment, it is necessary to perform an environment refresh operation.

9.9.7.1.1 Prerequisites for PostgreSQL discovery operation

The following requirements must be met before starting the discovery process.

When you add an environment to Delphix Continuous Data Engine, the PostgreSQL connector will automatically perform a discovery of all the PostgreSQL installed binaries on the server.

1. The PostgreSQL connector must be installed.
2. The DELPHIX_PG_PATH must be set if the PostgreSQL installation you want to discover is in a shutdown state.
3. The <postgresql bin> path [each directory in the path] must have read and execute permissions for the environment user.
4. The environment user must be able to execute the <postgresql bin>/postgres --version command.

9.9.7.1.2 Adding an PostgreSQL environment

This topic describes the process of adding a PostgreSQL staging and target environment.

9.9.7.1.2.1 Prerequisites

To set up a PostgreSQL staging environment, ensure all the essential requirements, as described in Requirements and Prerequisites (see page 1769) are met.

9.9.7.1.2.2 Procedure

Perform the following steps to add an environment to the Delphix Continuous Data Engine:

1. Login to the **Delphix Management** application.
2. Navigate to the **Manage > Environments** page.
3. Click the **Actions** ··· menu and select **Add Environment**.
4. From the **Add Environment** dialog window, select **Unix/Linux.**
5. Select **Standalone Host**.
6. Click **Next**.
7. Provide the name of the environment.
8. Provide the **Host IP address** or **host FQDN**.
9. Provide the **SSH port**. The default value is 22.
10. Provide an **Environment Username** for the environment. If you intend to use a low-privileged environment user, ensure that you have fulfilled the Prerequisites for Privilege Elevation using DLPX_DB_EXEC Script (see page 1780).
11. Choose the **Login Type** that fits your setup.
    - Username and Password – provide the environment username and password
    - Username and Public Key – provide the environment username
    - Password Vault – select from an existing Enterprise Password Vault

> **Using Public Key Authentication**
> If you want to use public-key authentication for logging into your Unix-based environment, you have two options:
>
> - Use the Delphix Continuous Data Engine's key pair. For more information on how to use the Delphix Continuous Data Engine's key pair, refer to the How to use the Continuous Data Engine's key-pair (see page 1845) section.
> - Provide a key pair for this environment. As an alternative, a key pair specific for this environment can be provided via the API or CLI. For instructions, refer to CLI Cookbook: Setting up SSH key authentication for UNIX environment users[538].

12. For Password, enter the password associated with the user in step 11.
13. If you want to use Public Key Encryption for logging into your environment:
    a. Select Public Key for the Login Type.
    b. Click View Public Key.

538 https://cd.delphix.com/docs/latest/cli-cookbook-setting-up-ssh-key-authentication-for

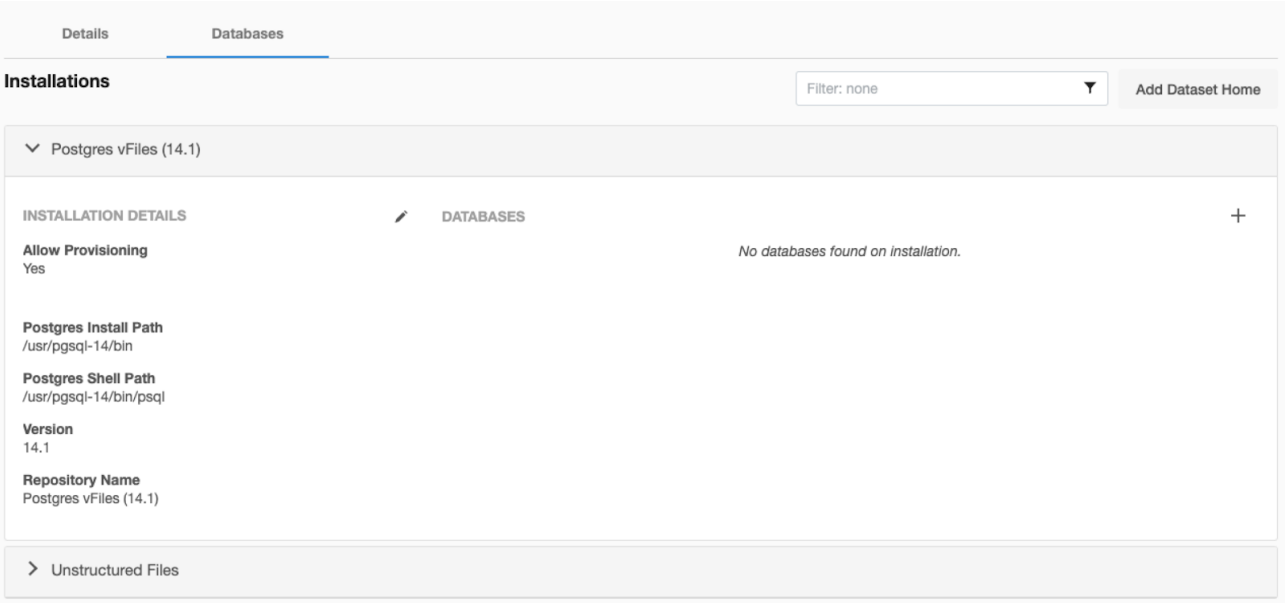
c. Copy the public key that is displayed, and append it to the end of your `~/.ssh/authorized_keys` file. If this file does not exist, it must be created.

> The addition of the public key is a one-time requirement per user and per environment. Additionally, you have the option to add public key authentication to an environment user's profile via CLI (for details, refer to the CLI Cookbook: Setting up SSH key authentication for UNIX environment users[539] topic).

14. For Password Login, click Verify Credentials to test the username and password.
15. Enter **Toolkit Path** (ensure that the toolkit path does not have any spaces).
16. To provide your own Oracle Java Development Kit, select the **Provide my own JDK** checkbox and click **Next**.
    a. On the Java Development Kit page, provide an absolute path to your Oracle JDK and click **Next**.
17. Click **Submit**.

After successfully adding and discovering the environment, you should be able to view the newly added environment listed under the Environments tab. In case it does not appear, click the refresh icon in your browser.

After the environment has been successfully discovered if you wish to establish further links, add a source configuration to the previously discovered installation.



Perform the following steps to access the information about any environment you have created:

1. Navigate to **Manage** > **Environments**.

539 https://cd.delphix.com/docs/latest/cli-cookbook-setting-up-ssh-key-authentication-for

2. Select the environment name.

9.9.7.1.3 Adding a PostgreSQL installation

If you have added a new PostgreSQL Installation to your environment and want it to be reflected in the Delphix Continuous Data Engine UI, you must refresh the environment. For more information, refer to the "Environment Refresh" section on the Managing environments and hosts[540] page.

9.9.7.1.4 PostgreSQL environment-related operations via CLI

There are many operations related to a Delphix Continuous Data Engine environment:

- Create/Discover
- Update
- Refresh
- Disable
- Enable
- Delete

CLI operations for each will appear below:

To create/discover an environment via CLI, login to the Delphix Continuous Data Engine using the admin user account.

```
ssh admin@<CONTINUOUS-DATA-FQDN-or-IP>
```

9.9.7.1.4.1 Creating/ Discovering the environment

Run the following commands to discover the environment.

```
environment
create
set hostEnvironment.name=rh86-pg15-s1
set hostParameters.host.address=rh86-pg15-s1.dlpxdc.co
set hostParameters.host.toolkitPath=/tmp
set primaryUser.name=postgres
set primaryUser.credential.password=postgres
commit
```

9.9.7.1.4.2 Updating the environment

Run the following commands to update the environment.

540 https://cd.delphix.com/docs/latest/managing-environments-and-hosts

The name of the environment can be updated from *rh86-pg15-s* to *rh86-pg15-s1.dlpxdc.co*[541] using the following command.

```
environment
select rh86-pg15-s1
update
set name=rh86-pg15-s1.dlpxdc.co
commit
```

9.9.7.1.4.3 Refreshing the environment

Run the following commands to refresh the environment.

```
environment
select rh86-pg15-s1.dlpxdc.co
refresh
commit
```

9.9.7.1.4.4 Disabling the environment

Run the following commands to disable the environment.

```
environment
select rh86-pg15-s1.dlpxdc.co
disable
commit
```

9.9.7.1.4.5 Enabling the environment

Run the following commands to enable the environment.

```
environment
select rh86-pg15-s1.dlpxdc.co
enable
commit
```

9.9.7.1.4.6 Deleting the environment

Run the following commands to delete the environment.

541 http://rh86-pg15-s1.dlpxdc.co

```
environment
select rh86-pg15-s1.dlpxdc.co
delete
commit
```

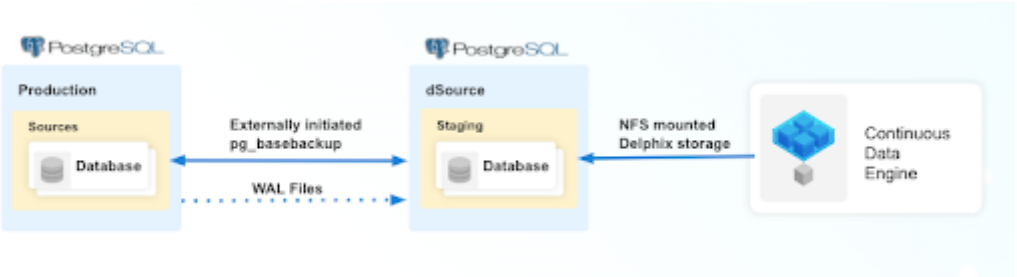
9.9.7.2 Linking PostgreSQL data sources

The Delphix Continuous Data Engine provides various ingestion methods for linking the dSource from a PostgreSQL source database.

1. Log Shipping with Externally-initiated backup (see page 1802)
2. PostgreSQL Streaming Replication with Externally Initiated Backup Ingestion (see page 1807)
3. PostgreSQL Streaming Replication with Delphix Initiated Backup Ingestion (see page 1812)
4. Delphix Initiated Single database Backup (see page 1816)
5. Staging Push (see page 1821)

9.9.7.2.1 Log shipping with externally-initiated backup

The following diagram demonstrates the architecture of the Delphix Continuous Data Engine support for the PostgreSQL database with external backup + WAL application ingestion between the source and staging database.



Perform the following steps to create a dSource with External Backup that uses WAL logs to stay in sync with the source.

9.9.7.2.1.1 Prerequisites

Ensure that you meet the following requirements before you begin creating a dSource.

- The environment requirements are met by the source and staging environments as described in the Requirements for PostgreSQL Staging/Target environment (see page 1780).
- The environment user or the privileged-user must have the following permissions:
  a. access to the PostgreSQL commands
  b. read+write+execute permissions on the `unix_socket_directories` configured in `postgresql.conf` file (The default directories are `/var/run/postgresql` & `/tmp`).
- To use privilege elevation
  a. You must meet the Privilege Elevation Requirements (see page 1780).

b. The high privilege user must have the following permissions:
   i. read+execute permissions on the **PostgreSQL Backup File Path**
   ii. read permission on the backup file(s)
   iii. read+write+execute permissions on the **PostgreSQL Backup WAL Log Files Path**.
   iv. read permission on the WAL logs within **WAL Log Files Path**.

- In connector versions 3.2.0 and greater**,** the *pg_basebackup* database full backup can be provided in three ways, ranked in the order of high-to-low precedence:
  a. The backup files are placed within the organized folder(s) inside the PostgreSQL Backup File Path named PostgreSQL_T<YYYYMMDDhhmmss>. During the dSource creation/resync operation, backups from the folder with the latest timestamp in the name will be selected.
     i. Run the following command to create a folder(s) on the staging server:

```
DLPXPGBACKUPDIR=PostgreSQL_T$(date "+%Y%m%d%H%M%S")
mkdir -p  <backup_path>/$DLPXPGBACKUPDIR
```

     ii. Run the following command to take the source backup on the staging server using pg_basebackup:

```
# For PostgreSQL versions 9.4.x,9.5.x,9.6.x
<path_to_bin>/pg_basebackup -D <backup_path>/$DLPXPGBACKUPDIR -F t
-v -w -P -x -h <source_ip_domain_name> -p <source_port> -U
<source_user>
# For PostgreSQL versions 10.x and above
<path_to_bin>/pg_basebackup -D <backup_path>/$DLPXPGBACKUPDIR -F t
-v -w -P -Xs -h <source_ip_domain_name> -p <source_port> -U
<source_user>
```

  b. The backup files are stored directly within the PostgreSQL Backup File Path.
     i. Make sure to redirect the *pg_basebackup* output to an empty directory and then move it to <backup_path>.

```
# For PostgreSQL versions 9.4.x,9.5.x,9.6.x
<path_to_bin>/pg_basebackup -D <path_to_empty_directory> -F t -v
-w -P -x -h <source_ip_domain_name> -p <source_port> -U
<source_user>
# For PostgreSQL versions 10.x and above
<path_to_bin>/pg_basebackup -D <path_to_empty_directory> -F t -v
-w -P -Xs -h <source_ip_domain_name> -p <source_port> -U
<source_user>
# Move files
mv <path_to_empty_directory>/* <backup_path>
```

  c. The backup files are compressed and stored within a ZIP file named PostgreSQL_T<YYYYMMDD>.zip and placed within PostgreSQL Backup File Path.
     i. Run the following command to take the source backup on the source server using *pg_basebackup*:

```
# For PostgreSQL versions 9.4.x,9.5.x,9.6.x
<path_to_bin>/pg_basebackup -D <destination_path> -F t -v -w -P -x
# For PostgreSQL versions 10.x and above
<path_to_bin>/pg_basebackup -D <destination_path> -F t -v -w -P
-Xs
```

ii. Run the following command to create a ZIP file that can be moved to the staging server (ensure that the backup files are stored at the root of the ZIP file):

d.

```
cd <destination_path>
# The back files should be at the root of the ZIP file
DLPXPGBACKUPZIP=PostgreSQL_T$(date +"%Y%m%d").zip
zip $DLPXPGBACKUPZIP *
```

For more information, refer to Requirements and Prerequisites (see page 1769).

9.9.7.2.1.2 Limitation

For PostgreSQL versions 9.4 and 9.5, the connector creates the staging instance as a warm standby (see page 1841). This instance is not accessible because it is always in recovery mode and does not allow any connection.

> The limitation does not affect the provisioning of a VDB.

9.9.7.2.1.3 Procedure

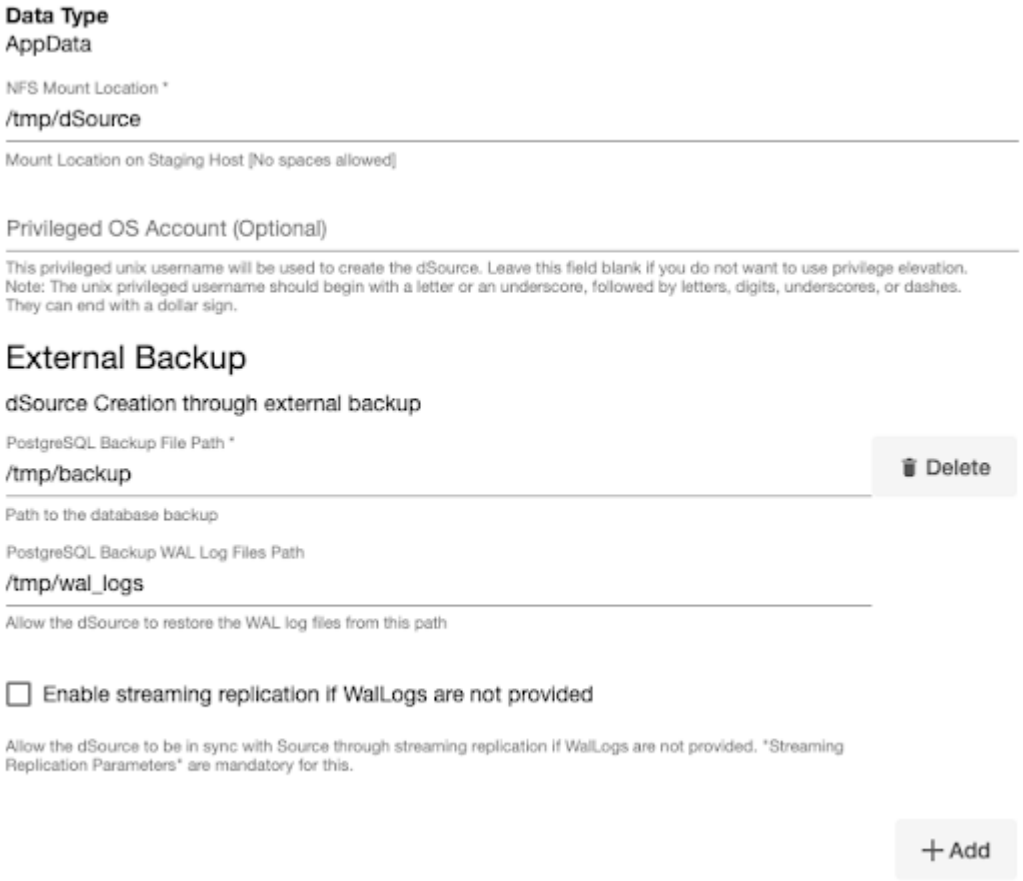
Perform the following steps to create a dSource using the externally initiated backup ingestion with WAL log shipping method.

1. Login to the **Delphix Management** application.
2. Navigate to the **Manage > Environments** page.
3. On the **Databases** tab, choose a repository for your dSource and click the + icon.
4. In the **Add Database** dialog window, provide the **name** of your source configuration and click **Add**.
5. Select your source configuration and click the **Add dSource** link located on the right.
6. On the **Source** tab, do the following:
   a. In the **NFS Mount Location** field, provide a mount location on the staging environment.
   b. If you want to use Privilege Elevation, provide the username of the **Privileged OS Account**. **Note**: Once the dSource is created, you will not be able to update the privileged OS account. Recreating the dSource account is the only way to make changes.

c. Click on the **+Add** icon next to **External Backup** Parameter.
Note: You must click the “Add Button” for External Backup only once as the solution supports a single backup path. Providing multiple paths will result in an error during the linking process.

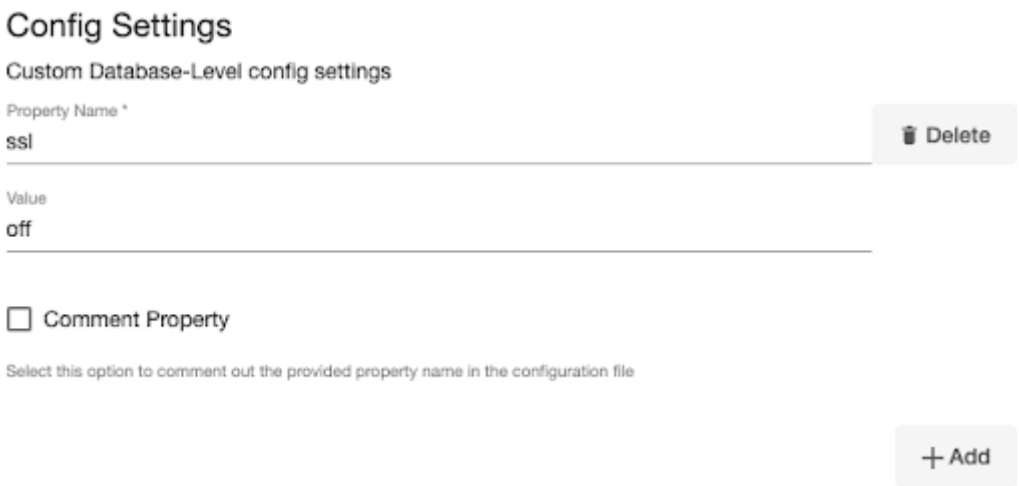
d. Specify the **PostgreSQL Backup File Path**. This is the location of the backup file.

e. Specify the **PostgreSQL Backup WAL Log Files Path**. This is the location of the WAL log files.
Note: Once the dSource has been set up using a full external backup from the staging environment, it operates in recovery mode and patiently awaits the arrival of the next expected WAL (Write-Ahead Logging) file in the specified **PostgreSQL Backup WAL Log Files Path**. When the expected WAL files are placed in this path, the dSource proceeds to apply these WAL files and subsequently waits for the next expected files in sequential order. However, if an invalid or out-of-sequence WAL file is provided, the dSource will throw an error on the UI when the next dSource snapshot is taken. In the event of an error in the WAL chain, you have two options: resynchronize the dSource or provide the missing WAL file. After addressing the issue, it may be necessary to disable and then enable the dSource database.



f. Specify the **Staging Instance Port Number**. This number must be unique for a given instance.

g. Optionally, you can define database configuration parameters during the linking operation in the “Config Settings" section if needed.



Note: During dSource creation, you now have the capability to configure the parameters and their values in the postgresql.conf file using the "Config Settings" section in the UI. For example, if you wish to disable SSL via UI, you can specify the "ssl" parameter and its value through the UI. This action will update the existing parameter value in the postgresql.conf file with the value provided. Similarly, if you want to disable a parameter, specify the parameter name and select the **Comment Property** check box. This action will comment out the parameter in the postgresql.conf configuration file.

h. Click **Next**.

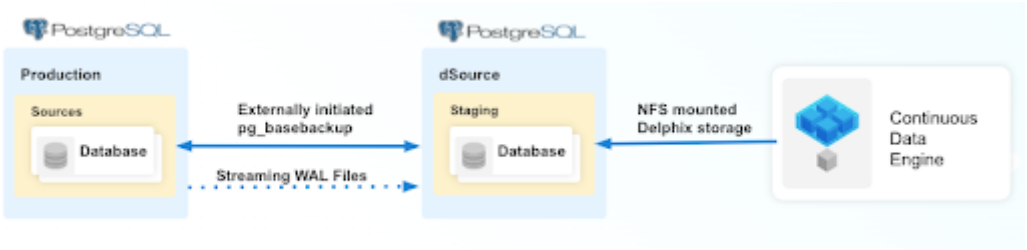
7. In the **dSource Configuration** tab, provide a dSource name and select a group for your dSource. Adding a dSource to a dataset group allows you to manage user permissions for that database and its objects, including snapshots. For more information, refer to the topics under Users and Groups[542].
8. In the **Data Management** tab, specify your staging environment, user, and snapshot parameters. Select the **Resynchronize dSource** checkbox if you want to resynchronize the dSource. This operation will force a non-incremental load of data from the source. This operation is similar to creating a new dSource, but prevents duplication of the storage requirements and maintains timeflow history. Click **Next** to proceed.
9. In the **Policies** tab, apply policy details to the dSource if needed, then click **Next** to continue.
10. In the **Hooks** tab, select a Hook Point and then click **+** to add a script that you want to run at that hook point. You have the flexibility to define scripts to run at multiple hook points in the process.
11. In the **Summary** tab, review the configuration profile for your dSource.
12. Click **Submit**. The Delphix Continuous Data Engine will initiate two jobs, namely DB_Link and DB_Sync to create the dSource. You can monitor these jobs by selecting **System>Jobs** or by selecting **System >Events** in the top menu bar. Upon successful completion of the DB_link job jobs, the database icon will change to a dSource icon on the **Environments > Host > Databases** screen. The dSource will also be listed under Datasets within its assigned group.

542 https://cd.delphix.com/docs/latest/users-and-groups

After you have successfully created a dSource, you can access the dSource Configuration tab to view details about it and make modifications to its policies and permissions. In the Datasets panel, select the specific dSource you want to inspect. Then, navigate to the Configuration tab to view details associated with it such as the source files, data management configuration, and hook operations.

9.9.7.2.2 PostgreSQL streaming replication with externally-initiated backup

The following diagram demonstrates the architecture of the Delphix Continuous Data Engine support for the PostgreSQL database linking with the PostgreSQL streaming replication with externally-initiated backup between the source and staging database.



Perform the following steps to create a dSource with External Backup that uses streaming replication to keep the staging WAL files in sync with the source.

9.9.7.2.2.1 Prerequisites

Ensure that you meet the following requirements before you begin creating a dSource.

- The environment requirements are met by the source and staging environments as described in Requirements for PostgreSQL Staging/Target environments (see page 1780).
- The environment user or the privileged user must have the following permissions:
  a. access to the Postgres commands
  b. read+write+execute permissions on the `unix_socket_directories` configured in `postgresql.conf` file (The default directories are `/var/run/postgresql` & `/tmp`).
- To use privilege elevation
  a. You must meet the Privilege Elevation Requirements (see page 1780).
  b. The high-privilege user must have the following permissions:
     i. read+execute permissions on the **PostgreSQL Backup File Path**.
     ii. read permission on the backup file(s)
- **Starting with connector versions 3.2.0 and greater,** the *pg_basebackup* database full backup can be provided in three ways, ranked in the order of high-to-low precedence:
  a. The backup files are placed within the organized folder(s) inside the PostgreSQL Backup File Path named PostgreSQL_T<YYYYMMDDhhmmss>. During the dSource creation/resync operation, backups from the folder with the latest timestamp in the name will be selected.
     i. Run the following command to create a folder(s) on the staging server:

```
DLPXPGBACKUPDIR=PostgreSQL_T$(date "+%Y%m%d%H%M%S")
mkdir -p <backup_path>/$DLPXPGBACKUPDIR
```

ii. Run the following command to take the source backup on the staging server using pg_basebackup:

```
# For PostgreSQL versions 9.4.x,9.5.x,9.6.x
<path_to_bin>/pg_basebackup -D <backup_path>/$DLPXPGBACKUPDIR -F t
-v -w -P -x -h <source_ip_domain_name> -p <source_port> -U
<source_user>
# For PostgreSQL versions 10.x and above
<path_to_bin>/pg_basebackup -D <backup_path>/$DLPXPGBACKUPDIR -F t
-v -w -P -Xs -h <source_ip_domain_name> -p <source_port> -U
<source_user>
```

b. The backup files are stored directly within the PostgreSQL Backup File Path.

i. Make sure to redirect the *pg_basebackup* output to an empty directory and then move it to <backup_path>.

```
# For PostgreSQL versions 9.4.x,9.5.x,9.6.x
<path_to_bin>/pg_basebackup -D <path_to_empty_directory> -F t -v
-w -P -x -h <source_ip_domain_name> -p <source_port> -U
<source_user>
# For PostgreSQL versions 10.x and above
<path_to_bin>/pg_basebackup -D <path_to_empty_directory> -F t -v
-w -P -Xs -h <source_ip_domain_name> -p <source_port> -U
<source_user>
# Move files
mv <path_to_empty_directory>/* <backup_path>
```

c. The backup files are compressed and stored within a ZIP file named PostgreSQL_T<YYYYMMDD>.zip and placed within PostgreSQL Backup File Path.

i. Run the following command to take the source backup on the source server using *pg_basebackup*:

```
# For PostgreSQL versions 9.4.x,9.5.x,9.6.x
<path_to_bin>/pg_basebackup -D <destination_path>  -F t -v -w -P
-x
# For PostgreSQL versions 10.x and above
<path_to_bin>/pg_basebackup -D <destination_path> -F t -v -w -P
-Xs
```

ii. Run the following command to create a ZIP file that can be moved to the staging server (ensure that the backup files are stored at the root of the ZIP file):

```
cd <destination_path>
```

```
# The back files should be at the root of the ZIP file
DLPXPGBACKUPZIP=PostgreSQL_T$(date +"%Y%m%d").zip
zip $DLPXPGBACKUPZIP *
```

For more information, refer to Requirements and Prerequisites (see page 1769).

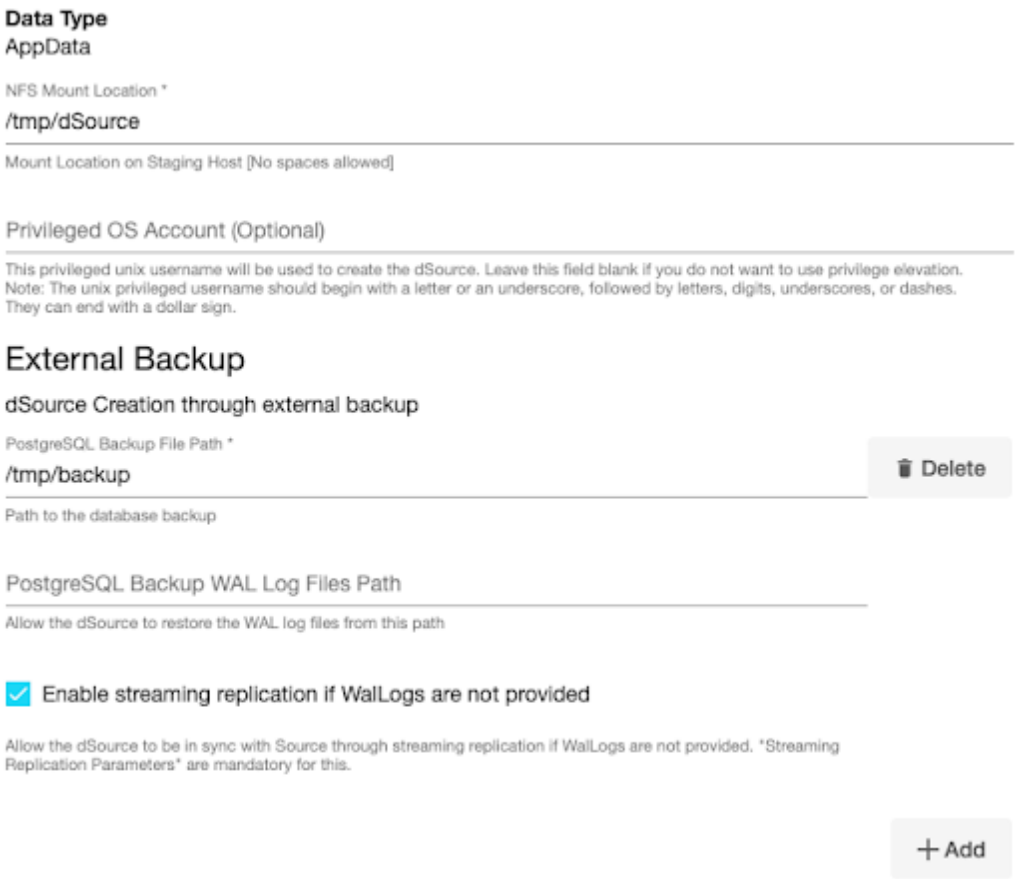
9.9.7.2.2.2 Limitation

With PostgreSQL versions 9.4 and 9.5, the connector creates the staging instance as a warm standby (see page 1841). This instance is not accessible because it is always in recovery mode and does not allow any connection.

9.9.7.2.2.3 Procedure

Perform the following steps to create a dSource using the PostgreSQL streaming replication with externally initiated backup ingestion.

1. Login to the **Delphix Management** application.
2. Navigate to the **Manage > Environments** page.
3. On the **Databases** tab, choose a repository for your dSource and click + the icon.
4. In the **Add Database** dialog window, provide the **name** of your source configuration and click **Add**.
5. Select your source configuration and click the **Add dSource** link located on the right.
6. In the **Source** tab, do the following:
   a. In the **NFS Mount Location** field, provide a mount location on the staging environment.
   b. If using Privilege Elevation, provide the username of the **Privileged OS Account**.
      Note:
      Once the dSource is created, you will not be able to update the privileged OS account. Recreating the dSource account is the only way to make changes.
   c. Click on the **+Add** icon next to **External Backup** Parameter.
   d. Specify the **PostgreSQL Backup File Path**. This is the location of the backup file.
   e. Select the '**Enable streaming replication if WalLogs are not provided**' checkbox.
      - This checkbox will take precedence over the PostgreSQL Backup WAL Log Files Path and ignore any value provided.



f. Click on the **+Add** icon next to the **Delphix Initiated Backup/External Backup - Streaming Replication Parameters** option.
Note: You must click the “Add Button” for Delphix Initiated Backup/External Backup - Streaming Replication only once as the solution supports a single backup path. Providing multiple paths will result in an error during the linking process.

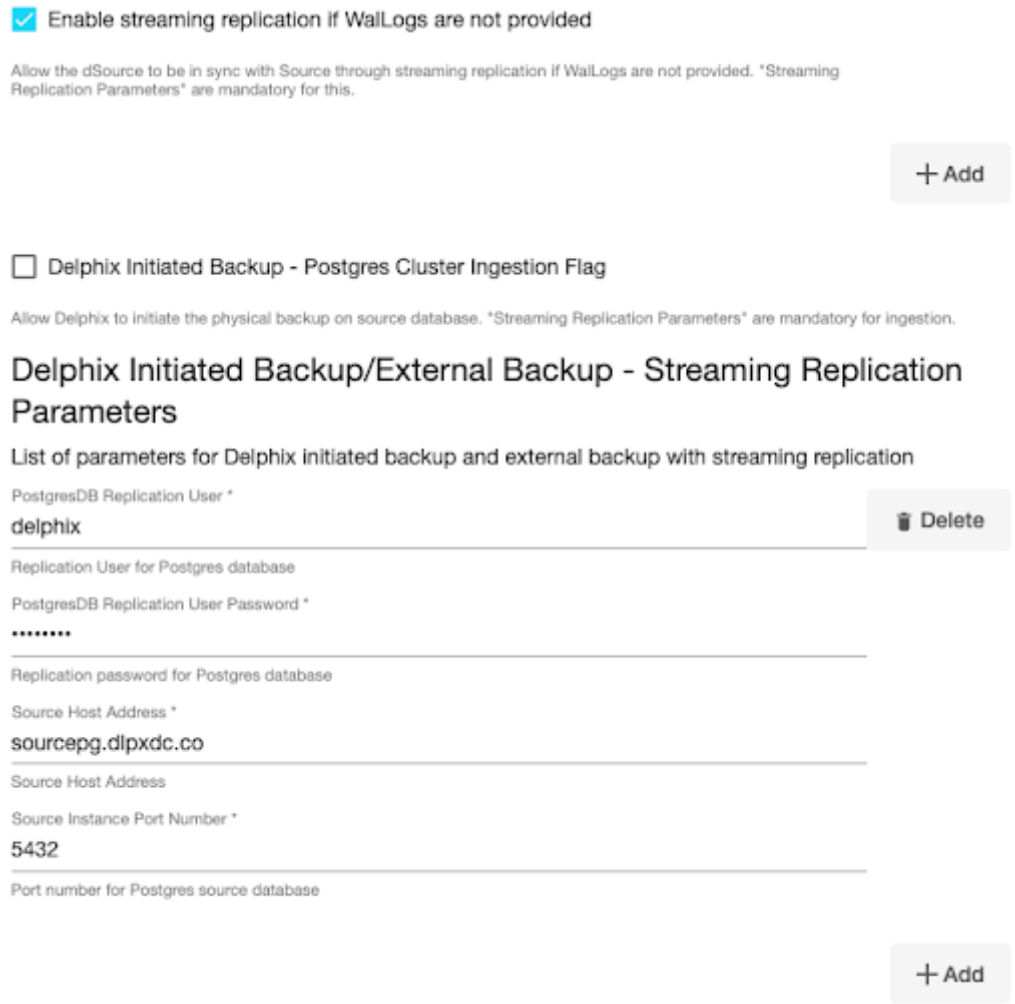
g. Provide the values in the following fields.

i. **PostgresDB Replication User**

ii. **PostgresDB Replication User Password**

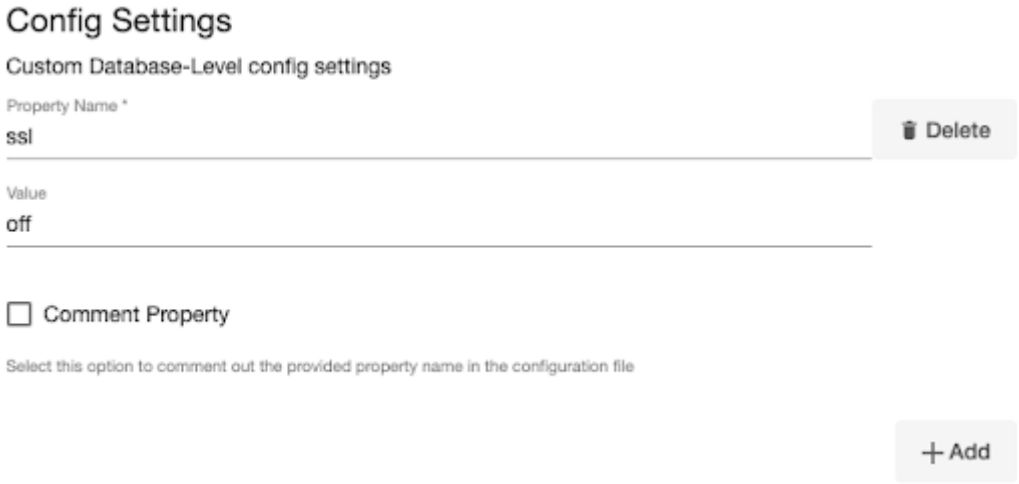
iii. **Source Host Address**

iv. **Source Instance Port Number**



v. Specify the **Staging Instance Port Number**. You must provide a port number that is available for the staging instance.

vi. Optionally, you can define database configuration parameters during the linking operation in the "Config Settings" section if needed.



Note: During dSource creation, you now have the capability to configure the parameters and their values in the postgresql.conf file using the "Config Settings" section in the UI. For example, if you wish to disable SSL via UI, you can specify the "ssl" parameter and its value through the UI. This action will update the existing parameter value in the postgresql.conf file with the value provided.
Similarly, if you want to disable a parameter you can specify the parameter name and then select the **Comment Property** check box. This action will comment out the parameter in the postgresql.conf configuration file.

vii. Click **Next**.

7. In the **dSource Configuration** tab, provide a dSource name and then select a group for your dSource. Adding a dSource to a dataset group allows you to manage user permissions for that database and its objects, including snapshots. For more information, refer to the topics under Users and Groups[543].
8. In the **Data Management** tab, specify your staging environment, user, and snapshot parameters. Select the **Resynchronize dSource** checkbox if you want to resynchronize the dSource. This operation will force a non-incremental load of data from the source. This operation is similar to creating a new dSource, but prevents duplication of the storage requirements and maintains timeflow history. Click **Next** to proceed.
9. In the **Policies** tab, apply policy details to the dSource if needed, and then click **Next** to continue.
10. In the **Hooks** tab, select a Hook Point and then click **+** to add a script that you want to run at that hook point. You have the flexibility to define scripts to run at multiple hook points in the process.
11. In the **Summary** tab, review the configuration profile for your dSource.
12. Click **Submit**. The Delphix Continuous Data Engine will initiate two jobs, namely DB_Link and DB_Sync to create the dSource. You can monitor these jobs by selecting **System>Jobs** or by selecting **System >Events** in the top menu bar. Upon successful completion of the DB_link job jobs, the database icon will change to a dSource icon on the **Environments > Host > Databases** screen. The dSource will also be listed under Datasets within its assigned group.

In the case of PostgreSQL streaming replication with Externally Initiated backup, the replication process is continuously monitored by reading the PostgreSQL database logs every minute. If, for any reason, the streaming replication stops, a fault will be generated, and an error message will be prominently displayed under the "Status" tab. It's important to note that if the streaming replication encounters a fault and stops, the snapshot operation will also fail.

9.9.7.2.3 PostgreSQL streaming replication with Delphix-initiated backup

The following diagram demonstrates the architecture of the Delphix Continuous Data Engine support for the PostgreSQL database with PostgreSQL streaming replication with Delphix-initiated backup between the source and staging database.

Perform the following steps to create a dSource using PostgreSQL streaming replication with Delphix-initiated backup to stay in sync with the source.

543 https://cd.delphix.com/docs/latest/users-and-groups

9.9.7.2.3.1 Prerequisites

Ensure that you meet the following requirements before you begin creating a dSource.

- The environment requirements are met by the source and staging environments as described in Requirements for PostgreSQL Staging/Target environments (see page 1780).
- To use privilege elevation, you must meet the Sudo Privilege Requirements (see page 1780).
- The OS or privileged user must have the following permissions:
  - access to the Postgres commands
  - read+write+execute permissions on the `unix_socket_directories` configured in `postgresql.conf` file (The default directories are `/var/run/postgresql` & `/tmp`).

For more information, refer to Requirements and Prerequisites (see page 1769).

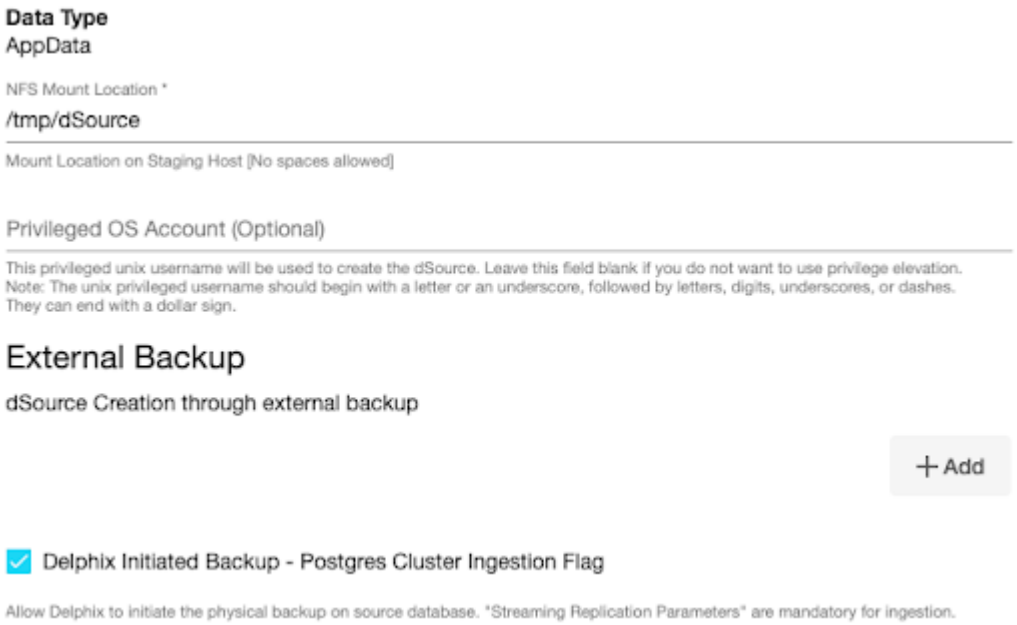
9.9.7.2.3.2 Limitation

With PostgreSQL versions 9.4 and 9.5, the connector creates the staging instance as a warm standby (see page 1841). This instance is not accessible because it is always in recovery mode and does not allow any connection.

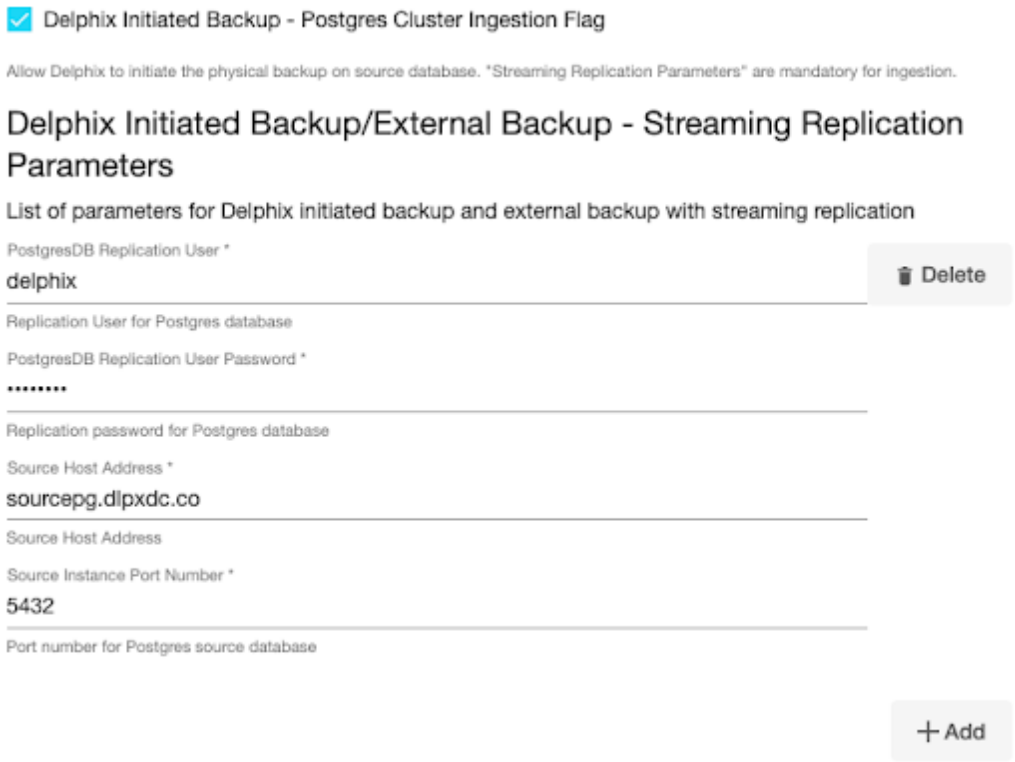
9.9.7.2.3.3 Procedure

Perform the following steps to create a dSource using the externally initiated backup ingestion with the WAL log shipping method.

1. Login to the **Delphix Management** application.
2. Navigate to the **Manage > Environments** page.
3. On the **Databases** tab, choose a repository for your dSource and click the + icon.
4. In the **Add Database** dialog window, provide the **name** of your source configuration and click **Add**.
5. Select your source configuration and click the **Add dSource** link located on the right.
6. In the **Source** tab, do the following:
   a. In the **NFS Mount Location** field, provide a mount location on the staging environment.
   b. If using Privilege Elevation, provide the username of the **Privileged OS Account**.
   Note: Once the dSource is created, you will not be able to update the privileged OS account. Recreating the dSource account is the only way to make changes.
   c. Select the **Delphix Initiated Backup - Postgres Cluster Ingestion Flag** checkbox. Enabling this checkbox allows Delphix to initiate the physical backup on the source database.

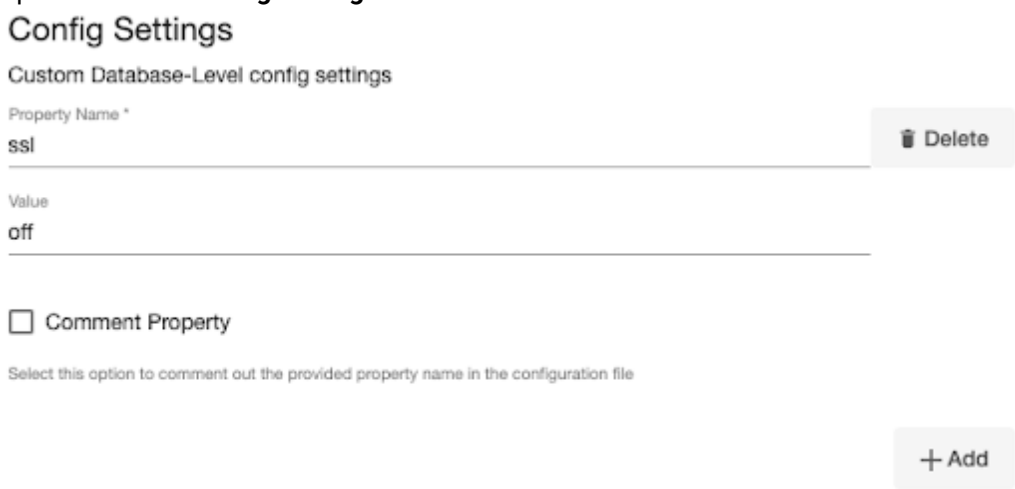


d. Click on the **+Add** icon next to the **Delphix Initiated Backup/External Backup - Streaming Replication Parameters** parameter.
   Note: You must click the “Add Button” for External Backup only once as the solution supports a single backup path. Providing multiple paths will result in an error during the linking process.
e. Provide the values in the following fields.
   i. **Delphix Initiated Backup - Postgres Cluster Ingestion Flag** - You must select this checkbox.
   ii. **PostgresDB Replication User**
   iii. **PostgresDB Replication User Password**
   iv. **Source Host Address**
   v. **Source Instance Port Number**



vi. Specify the **Staging Instance Port Number**. This port number must be unique and the port must be free for a given instance.

vii. Optionally, you can define database configuration parameters during the linking operation in the **Config Settings** section if needed.



Note: During dSource creation, you can configure the parameters and their values in the `postgresql.conf` file using the "Config Settings" section in the UI. For example, if you wish to support replication slots via UI, you can specify the "primary_slot_name" parameter and its value through the UI. This action will update the existing parameter value in the `postgresql.conf` file with the value provided. Similarly, if you want to disable a parameter, specify the parameter name and then

select the **Comment Property** check box. This action will comment out the parameter in the `postgresql.conf` configuration file.

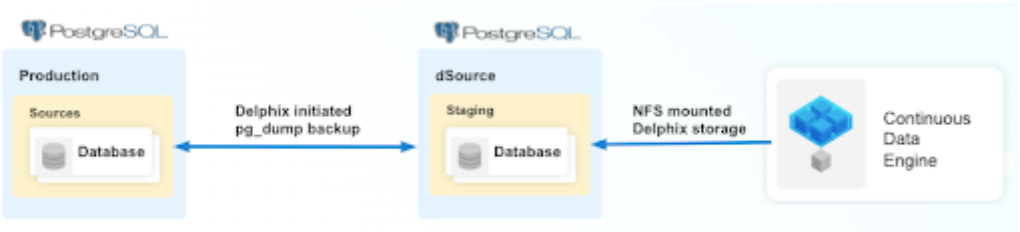
viii. Click **Next.**

7. In the **dSource Configuration** tab, provide a dSource name and then select a group for your dSource. Adding a dSource to a dataset group allows you to manage user permissions for that database and its objects, including snapshots. For more information, refer to the topics under Users and Groups[544].
8. In the **Data Management** tab, specify your staging environment, user, and snapshot parameters. Select the **Resynchronize dSource** checkbox if you want to resynchronize the dSource. This operation will force a non-incremental load of data from the source. This operation is similar to creating a new dSource, but prevents duplication of the storage requirements and maintains timeflow history. Click **Next** to proceed.
9. In the **Policies** tab, apply policy details to the dSource if needed, and then click **Next** to continue.
10. In the **Hooks** tab, select a Hook Point and then click **+** to add a script that you want to run at that hook point. You have the flexibility to define scripts to run at multiple hook points in the process.
11. In the **Summary** tab, review the configuration profile for your dSource.
12. Click **Submit**. The Delphix Continuous Data Engine will initiate two jobs, namely DB_Link and DB_Sync to create the dSource. You can monitor these jobs by selecting **System>Jobs** or by selecting **System >Events** in the top menu bar. Upon successful completion of the DB_link job jobs, the database icon will change to a dSource icon on the **Environments > Host > Databases** screen. The dSource will also be listed under Datasets within its assigned group.

> In the case of a Delphix-initiated backup, the streaming replication process is continuously monitored by reading the PostgreSQL database logs every minute. If, for any reason, the streaming replication stops, a fault will be generated, and an error message will be prominently displayed under the "Status" tab. It's important to note that if the streaming replication encounters a fault and stops, the snapshot operation will also fail.

9.9.7.2.4 Delphix-initiated single database backup

The following diagram demonstrates the architecture of the Delphix Continuous Data Engine support for the PostgreSQL database with pg_dump logical backup between source and staging database.



544 https://cd.delphix.com/docs/latest/users-and-groups

Perform the following steps to create a dSource with a Delphix-initiated backup that uses a single database backup to stay in sync with the source.

9.9.7.2.4.1 Prerequisites

Ensure that you meet the following requirements before you begin creating a dSource.

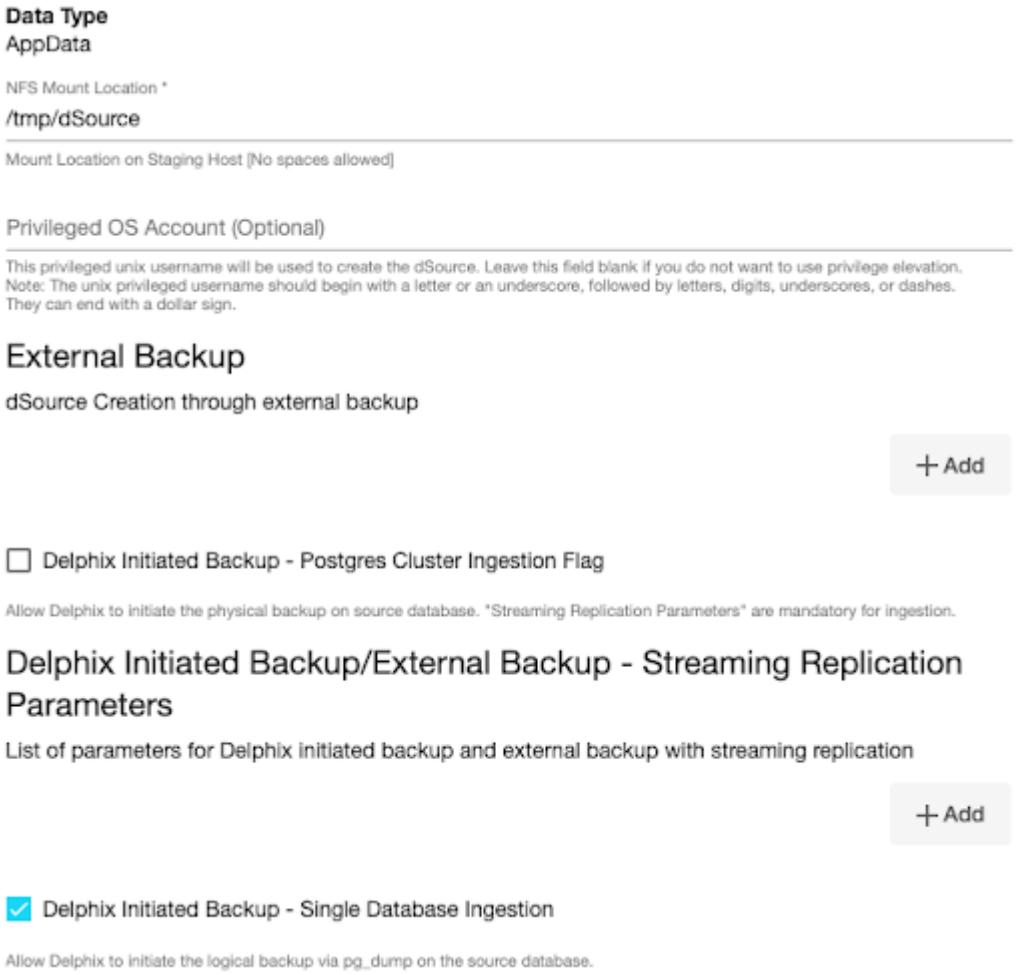
- The environment requirements are met by the source and staging environments as described in Requirements for PostgreSQL Staging and Target environments (see page 1780).
- The environment user or the privileged user must have the following permissions:
  - access to the Postgres commands
  - read+write+execute permissions on the `unix_socket_directories` configured in `postgresql.conf` file (The default directories are `/var/run/postgresql` & `/tmp`).
- To use privilege elevation,
  - You must meet the Privilege Elevation Requirements (see page 1780).
  - The high-privilege user must have the following permissions:
    - Write+execute permissions on the **PostgreSQL Backup Dump Directory**.

For more information, refer to Requirements and Prerequisites (see page 1769).

9.9.7.2.4.2 Procedure

Perform the following steps to create a dSource using the Delphix-initiated single database backup ingestion.

1. Login to the **Delphix Management** application.
2. Navigate to the **Manage > Environments** page.
3. On the **Databases** tab, choose a repository for your dSource and click the + icon.
4. In the **Add Database** dialog window, provide the **name** of your source configuration and click **Add**.
5. Select your source configuration and click the **Add dSource** link located on the right.
6. In the **Source** tab, do the following:
   a. In the **NFS Mount Location** field, provide a mount location on the staging environment.
   b. If using Privilege Elevation, provide the username of the **Privileged OS Account**.
   Note: Once the dSource is created, you will not be able to update the privileged OS account. Recreating the dSource account is the only way to make changes.
   c. Select the **Delphix Initiated Backup - Single Database Ingestion** checkbox. Enabling this checkbox allows Delphix to initiate the logical backup on the source database.

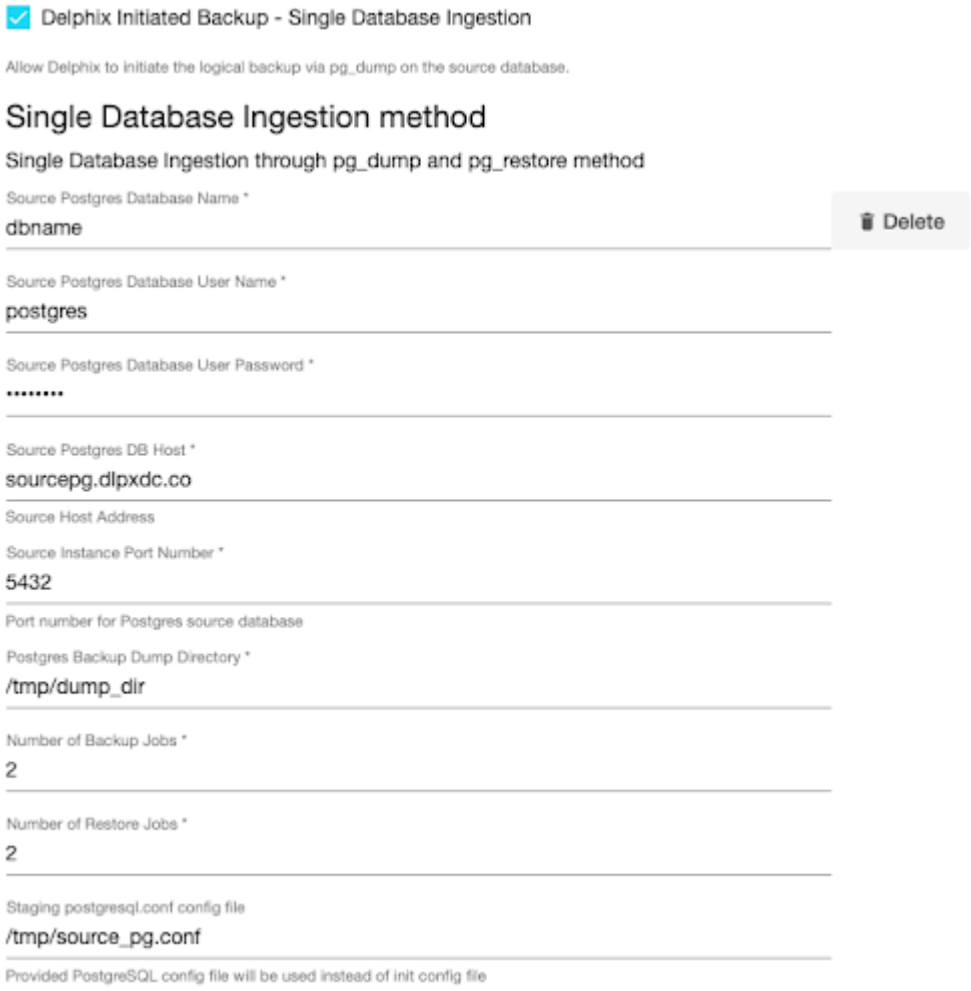


d. Click on the **+Add** icon next to the **Single Database Ingestion method** parameter.
Note: You must click the “Add Button” for External Backup only once as the solution supports a single backup path. Providing multiple paths will result in an error during the linking process.

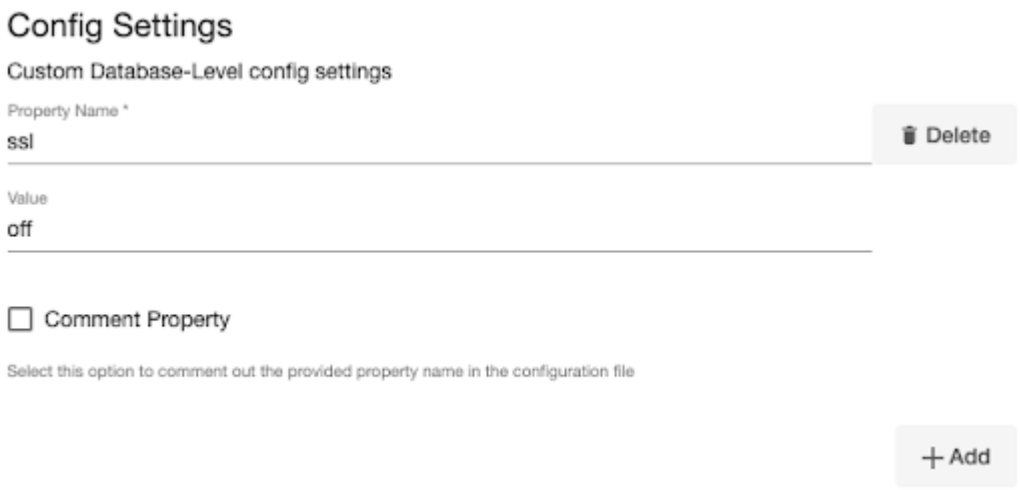
e. Provide the values in the following fields.

- **Source Postgres Database Name**
- **Source Postgres Database User Name.** The default value is postgres.
- **Source Postgres Database User Password**
- **Source Postgres DB Host**
- **Source Instance Port Number.** The default value is 5432.
- **Postgres Backup Dump Directory:** Provide the directory on the staging environment on which the backup will be created. This directory should already exist in the staging environment and must have read and write access for the environment user.
- **Number of Backup Jobs:** The default value is 2.
- **Number of Restore Jobs:** The default value is 2.

- **(Optional) Staging postgresql.conf config file:** Provide the path to a PostgreSQL configuration template file that will be used when starting the dSource.



f. Specify the **Staging Instance Port Number**. This port number must be unique and the port must be free for a given instance.
g. Optionally, you can define database configuration parameters during the linking operation in the "Config Settings" section if needed.



Note: During dSource creation, you now can configure the parameters and their values in the `postgresql.conf` file using the "Config Settings" section in the UI. For example, if you wish to disable SSL via UI, you can specify the "ssl" parameter and its value through the UI. This action will update the existing parameter value in the `postgresql.conf` file with the value provided. Similarly, if you want to disable a parameter then you can specify the parameter name and then select the **Comment Property** check box. This action will comment out the parameter in the `postgresql.conf` configuration file.

h. Click **Next**.

7. In the **dSource Configuration** tab, provide a dSource name and then select a group for your dSource. Adding a dSource to a dataset group allows you to manage user permissions for that database and its objects, including snapshots. For more information, refer to the topics under Users and Groups[545].
8. In the **Data Management** tab, specify your staging environment, user, and snapshot parameters. Select the **Resynchronize dSource** checkbox if you want to resynchronize the dSource. This operation will force a non-incremental load of data from the source. This operation is similar to creating a new dSource, but prevents duplication of the storage requirements and maintains timeflow history. Click **Next** to proceed.
9. In the **Policies** tab, apply policy details to the dSource if needed, and then click **Next** to continue.
10. In the **Hooks** tab, select a Hook Point and then click **+** to add a script you want to run at that hook point. You can define scripts to run at multiple hook points in the process.
11. In the **Summary** tab, review the configuration profile for your dSource.
12. Click **Submit**. The Delphix Continuous Data Engine will initiate two jobs, namely DB_Link and DB_Sync to create the dSource. You can monitor these jobs by selecting **System>Jobs** or by selecting **System >Events** in the top menu bar. Upon successful completion of the DB_link job jobs, the database icon will change to a dSource icon on the **Environments > Host > Databases** screen. The dSource will also be listed under Datasets within its assigned group.

545 https://cd.delphix.com/docs/latest/users-and-groups

> Starting with PostgreSQL connector version 4.0.0 and onwards, support has been added for using pg_dumpall with single database ingestion. This allows you to obtain global objects such as roles and tablespaces during the ingestion process.

9.9.7.2.5 Staging push

The Staging Push transfers the responsibility of preparing dSource to the end user. This ingestion method allows you to manually push the source data into the mount path backed by the Delphix storage. It offers an alternative approach to the Staging Pull model where the Delphix Continuous Data Engine takes care of preparing the dSource.

The Staging Push ingestion method has been verified with the following mechanisms:

- Backups + Logs w/ pgBackRest
- Backups + Logs w/ pg_basebackup
- Logical Replication
- pg_dump and pg_restore
- Streaming Replication

9.9.7.2.5.1 Prerequisites

Ensure that you meet the following requirements before you begin creating a dSource.

- The environment requirements are met by the source and staging environments as described in Requirements for PostgreSQL Staging/Target environments (see page 1780).
- To use privilege elevation, you must meet the Privilege Elevation Requirements (see page 1780).

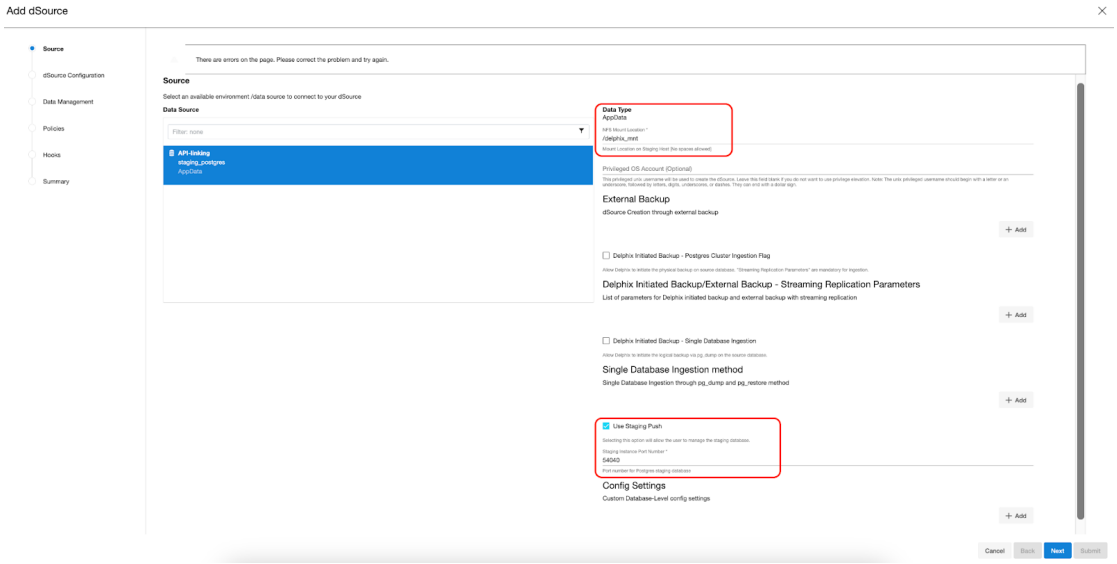
For more information, refer to Requirements and Prerequisites (see page 1769).

9.9.7.2.5.2 Procedure

Perform the following steps to create a dSource using the Staging Push ingestion method.

1. Login to the **Delphix Management** application.
2. Navigate to the **Manage > Environments** page.
3. On the **Databases** tab, choose a repository for your dSource and click the icon.
4. In the **Add Database** dialog window, provide the **name** of your source configuration and click **Add**.
5. Select your source configuration and click the **Add dSource** link located on the right.
6. In the **Source** tab, do the following:
   a. In the **NFS Mount Location** field, provide a mount location on the staging environment.
   b. Select the **Use Staging Push** checkbox.

c. Specify the **Staging Instance Port Number**. You must provide a port number that is available for the staging instance.



d. Click **Next**.

7. In the **dSource Configuration** tab, provide a dSource name and then select a group for your dSource. Adding a dSource to a dataset group allows you to manage user permissions for that database and its objects, including snapshots. For more information, refer to the topics under Users and Groups[546].
8. In the **Data Management** tab, specify your staging environment, user, and snapshot parameters. Select the **Resynchronize dSource** checkbox if you want to resynchronize the dSource. This operation will force a non-incremental load of data from the source. This operation is similar to creating a new dSource, but prevents duplication of the storage requirements and maintains timeflow history. Click **Next** to proceed.
9. In the **Policies** tab, apply policy details to the dSource if needed, and then click **Next** to continue.
10. In the **Hooks** tab, select a Hook Point and then click **+** to add a script you want to run at that hook point. You can define scripts to run at multiple hook points in the process.
11. In the **Summary** tab, review the configuration profile for your dSource.
12. Click **Submit**. The Delphix Continuous Data Engine will initiate two jobs, namely DB_Link and DB_Sync to create the dSource. You can monitor these jobs by selecting **System>Jobs** or by selecting **System >Events** in the top menu bar. Upon successful completion of the DB_link job jobs, the database icon will change to a dSource icon on the **Environments > Host > Databases** screen. The dSource will also be listed under Datasets within its assigned group.

9.9.7.2.6 Taking a snapshot

Perform the following steps to safely take snapshots of the dSource.

546 https://cd.delphix.com/docs/latest/users-and-groups

1. Recover the physical backup (pg_basebackup or pgBackRest) in the data directory. When restoring backups created with pg_dump, start by initializing the PostgreSQL cluster in the target data directory, and then proceed to recover the backup using pg_restore.
2. During the recovery of a physical backup, it's important to ensure that all tablespace locations, if any, are contained within the specified mount point. The following screenshot illustrates the expected file system structure after the recovery process is completed.

```
[postgres@target data]$ tree -L 1
.
├── base
├── current_logfiles
├── global
├── log
├── logfile
├── pg_commit_ts
├── pg_dynshmem
├── pg_hba.conf
├── pg_ident.conf
├── pg_logical
├── pg_multixact
├── pg_notify
├── pg_replslot
├── pg_serial
├── pg_snapshots
├── pg_stat
├── pg_stat_tmp
├── pg_subtrans
├── pg_tblspc
├── pg_twophase
├── PG_VERSION
├── pg_wal
├── pg_xact
├── postgresql.auto.conf
├── postgresql.conf
├── postmaster.opts
└── postmaster.pid

18 directories, 9 files
```

3. Stop the cluster.

> Failure to stop the cluster will result in a snapshot operation failure.

4. Take a snapshot.
5. Start the cluster.
6. Provision, refresh, or rewind VDBs using the new snapshot.

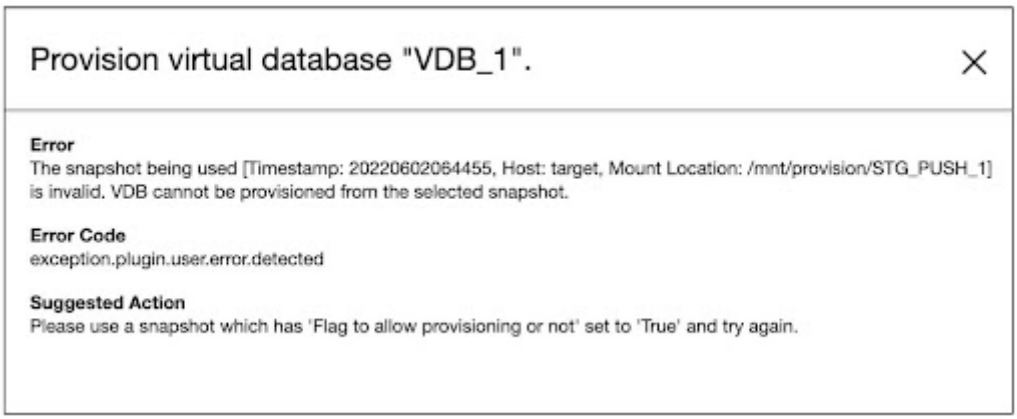
> The initial snapshot that is created during the creation of a dSource includes an empty data directory located within the mount point created by Delphix. Consequently, creating Virtual Databases (VDBs) directly from this snapshot is not possible. Resolve this issue by completing the ingestion and then taking a new snapshot of the dSource and provisioning a VDB from the new one.

The following snippet illustrates this scenario:

```
[postgres@target Staging_Push]$ tree
.
└── data

1 directory, 0 files
[postgres@target Staging_Push]$ pwd
/mnt/provision/Staging_Push
```

Attempting to create a Virtual Database (VDB) from this empty snapshot will result in a failure, and you will receive the following error message.

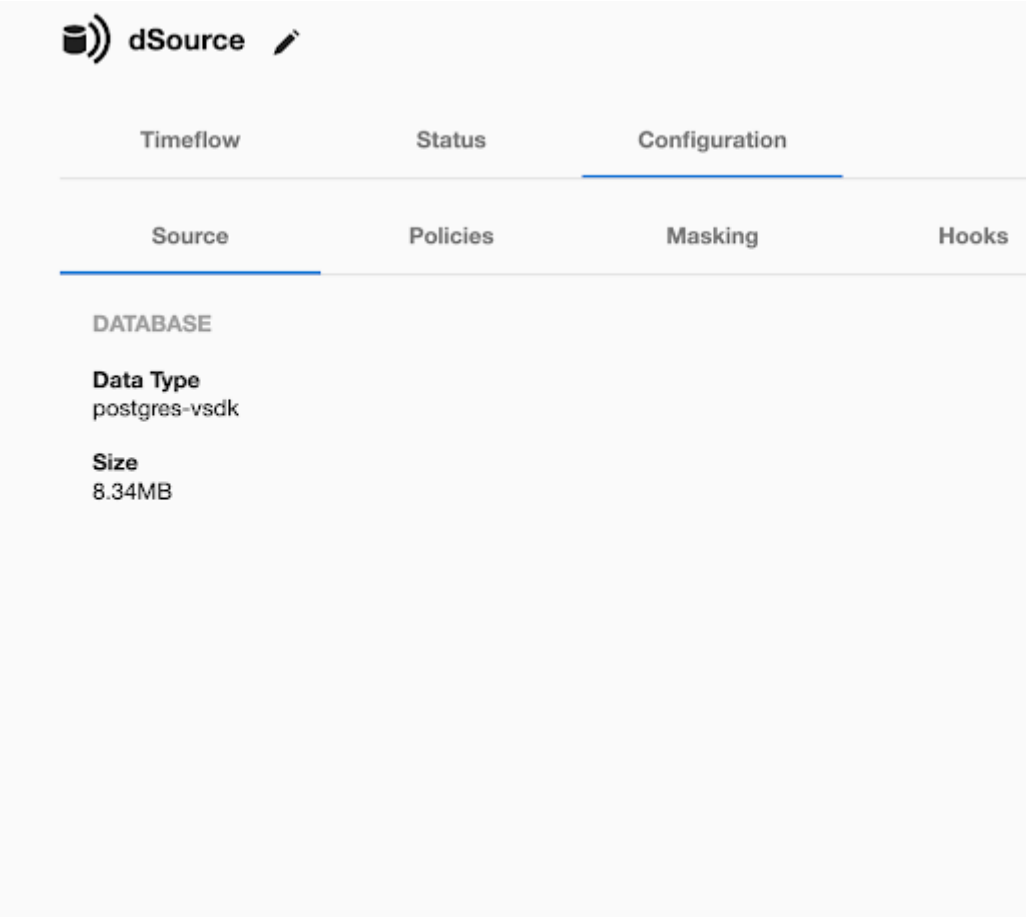


9.9.7.2.7 Source sizing implementation for dSource (PostgreSQL)

The Source Sizing feature aligns the data source experience with Delphix database standards that will improve your ingestion reporting experience on the per TB pricing model. This feature enables you to calculate the database size based on whether they are ingested at the PostgreSQL cluster level or database level.

9.9.7.2.7.1 Procedure

For all dSources, the dataset size can be calculated by executing `du -sb <dataset mount path>` command, which provides the correct volume and apparent size of the datasets.



9.9.7.2.8 PostgreSQL link operations via CLI

9.9.7.2.8.1 dSource creation from CLI

To create a dSource via CLI, you must create a source configuration first. For more information on creating a source configuration, refer to Add Source Config using CLI (see page 1041).

To create a dSource and perform operations via CLI, login to the Delphix Continuous Data Engine using the admin user account.

```
ssh admin@<CONTINUOUS-DATA-FQDN-or-IP>
```

9.9.7.2.8.2 Set default linking parameters

Run the following commands to set the default linking parameters. The following example uses `testscfg` source config.

```
ip-10-110-238-142> database link
ip-10-110-238-142 database link *> set linkData.type=AppDataStagedLinkData
ip-10-110-238-142 database link *> set name=testds
ip-10-110-238-142 database link *> set group=Untitled
ip-10-110-238-142 database link *> set linkData.config=testscfg
ip-10-110-238-142 database link *> set linkData.environmentUser=rh83-pg14-
tec.dlpxdc.co/postgres

ip-10-110-238-142 database link *> set linkData.stagingEnvironment=rh83-pg14-
tec.dlpxdc.co

ip-10-110-238-142 database link *> set linkData.stagingEnvironmentUser=rh83-pg14-
tec.dlpxdc.co/postgres

ip-10-110-238-142 database link *> set
linkData.syncParameters.parameters={\"resync\":true}
```

9.9.7.2.8.3 Link parameters for Delphix initiated backup with memory and DB config

Run the following commands to link parameters for Delphix-initiated backup with memory and DB configuration.

```
set linkData.parameters="{\"mountLocation\":\"/tmp/DIBR\",\"externalBackup\":[],
\"delphixInitiatedBackupFlag\":true, \"privilegedOsUser\":\"\",
\"delphixInitiatedBackup\":[{\"userName\":\"delphix\",\"userPass\":\"password\",
\"sourceHostAddress\":\"c78-pg12-s2.dlpxdc.co\", \"postgresSourcePort\":5432}],
\"singleDatabaseIngestionFlag\":false,\"singleDatabaseIngestion\":[],
\"postgresPort\":5442, \"configSettingsStg\":[{\"propertyName\":
\"wal_keep_segments\", \"value\": \"9\"}]}"
```

9.9.7.2.8.4 Link parameters for log shipping with externally initiated backup

Run the following commands to link parameters for Log Shipping with Externally Initiated Backup.

```
set linkData.parameters="{\"mountLocation\":\"/tmp/DIBR\",\"externalBackup\":[],
\"delphixInitiatedBackupFlag\":true, \"privilegedOsUser\":\"\",
\"delphixInitiatedBackup\":[{\"userName\":\"delphix\",\"userPass\":\"password\",
\"sourceHostAddress\":\"c78-pg12-s2.dlpxdc.co\", \"postgresSourcePort\":5432}],
\"singleDatabaseIngestionFlag\":false,\"singleDatabaseIngestion\":[],
```

```
\"postgresPort\":5442, \"configSettingsStg\":[{\"propertyName\":
\"wal_keep_segments\", \"value\": \"9\"}]}"
```

9.9.7.2.8.5 Link parameters for PostgreSQL streaming replication with externally initiated backup:

Run the following commands to link parameters for PostgreSQL streaming replication with externally initiated backup.

```
set linkData.parameters="{\"mountLocation\":\"/tmp/mntp\",\"externalBackup\":
[{\"backupPath\":\"/tmp/backup\", \"walLogPath\":\"/tmp/backup\",
\"keepStagingInSync\":false}], \"delphixInitiatedBackupFlag\":false,
\"delphixInitiatedBackup\":[], \"singleDatabaseIngestionFlag\":false,
\"singleDatabaseIngestion\":[], \"stagingPushFlag\":true,\"postgresPort\":5436,
\"configSettingsStg\":[]}"
```

9.9.7.2.8.6 Link parameters for Delphix-initiated single database backup ingestion

Run the following commands to link parameters for Delphix-initiated Single Database backup ingestion.

```
set linkData.parameters="{\"mountLocation\":\"/tmp/singledb2\",\"privilegedOsUser\":
\"\",\"externalBackup\":[], \"delphixInitiatedBackupFlag\":false,
\"delphixInitiatedBackup\":[],\"singleDatabaseIngestionFlag\":true,
\"singleDatabaseIngestion\":[{\"databaseName\":\"postgres\", \"databaseUserName\":
\"postgres\", \"databaseUserPassword\":\"postgres\", \"sourceHost\":\"rh83-pg14-
sec.dlpxdc.co\", \"sourcePort\":5432, \"dumpDir\":\"/home/postgres\", \"dumpJobs\":2,
\"restoreJobs\":2, \"postgresqlFile\":\"\"}],\"stagingPushFlag\":false,
\"postgresPort\":5435, \"configSettingsStg\":[]}"
```

9.9.7.2.8.7 Link parameters for staging push

Run the following commands to link parameters for staging push.

```
set linkData.parameters="{\"mountLocation\":\"/tmp/pg_basebackupD\",
\"privilegedOsUser\":\"\",\"externalBackup\":[],
\"delphixInitiatedBackupFlag\":false, \"delphixInitiatedBackup\":[],
\"singleDatabaseIngestionFlag\":false,\"singleDatabaseIngestion\":[],
\"stagingPushFlag\":true,\"postgresPort\":5433, \"configSettingsStg\":[]}"
```

9.9.7.2.8.8 Commit

Run the following command to commit the change.

```
commit;
```

9.9.7.2.8.9 **Disable**

Run the following command to disable the dSource.

```
source
select testdb
disable
commit
```

9.9.7.2.8.10 **Enable**

Run the following command to enable the dSource.

```
source
select testdb
enable
commit
```

9.9.7.3 Provisioning PostgreSQL virtual databases

The virtual database (VDB) is a Delphix virtualized copy of your source database that runs on your Target Environment. A VDB is provisioned from a dSource or VDB snapshot.

9.9.7.3.1 Prerequisites

Meet the following requirements to create a VDB:

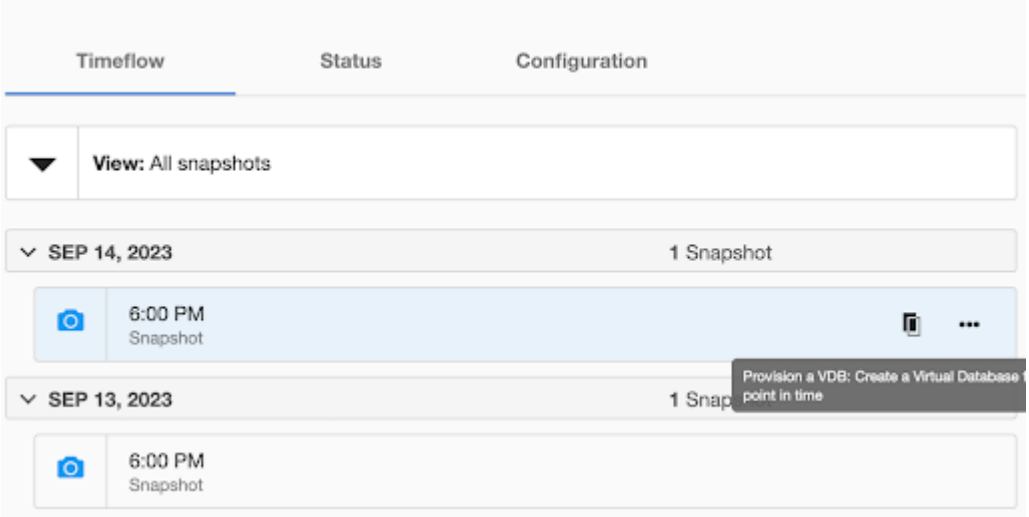
1. Identify an originating dataset and snapshot.
    a. Link a dSource by following the instructions in Linking a PostgreSQL dSource (see page 1802).
    b. Select a VDB that has been previously provisioned.
2. Set up a PostgreSQL target environment as described in PostgreSQL support and requirements (see page 1780).
3. To use privilege elevation, you must meet the Privilege Elevation Requirements (see page 1780).

9.9.7.3.2 Procedure

Perform the following steps to provision a PostgreSQL VDB.

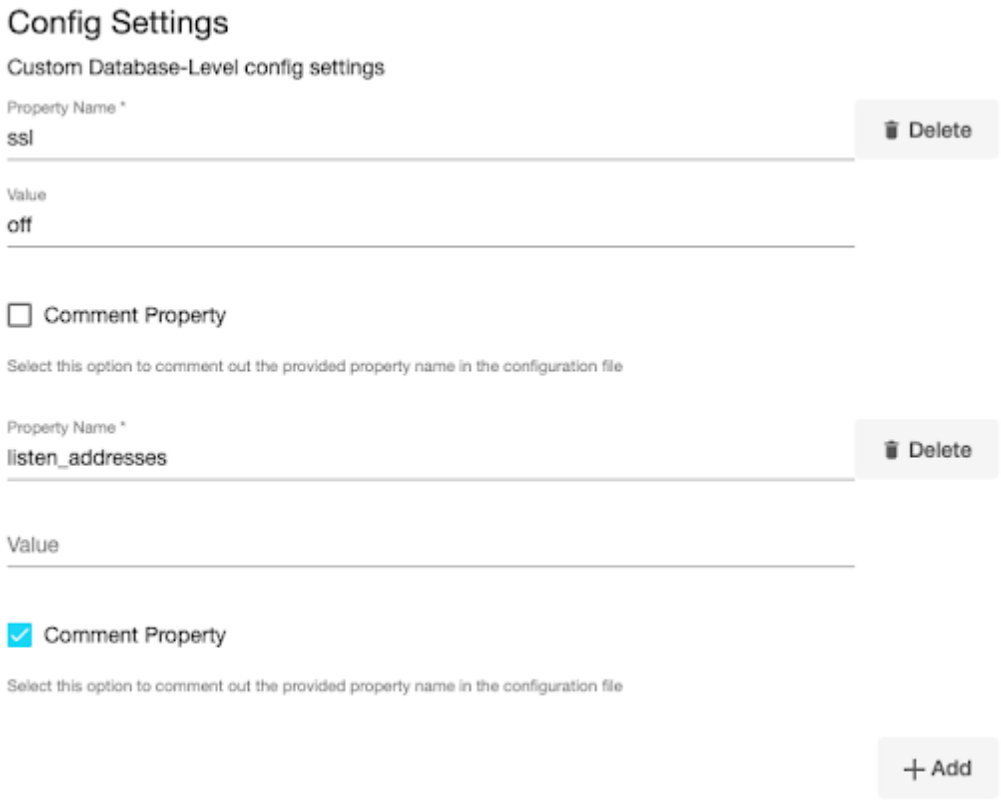
1. Login to the **Delphix Management** application.

2. Navigate to the **Manage > Datasets** page and select a dataset.
3. Under the Timeflow tab, identify a snapshot, hover over it, and select the **Provision a VDB** button**.**



4. On the **Target Environment** page, perform the following actions:
   - From the **Environment** dropdown list, select a target environment. This will dynamically populate the other properties.
   - From the **Installation** dropdown list, select the specific PostgreSQL installation that you want to use.
   - From the **User** dropdown list, select the environment user that matches the PostgreSQL Instance Owner as configured in the prerequisites.
     Note: The ability to select different instance owners is only applicable when you have multiple users configured in that environment.
5. On the **Target Configuration** page, perform the following actions:
   - Specify the target environment's **NFS Mount Location**. The specified path must not exist or be empty.

- (Optional) Enable **Privilege Elevation** by providing the username of the **Privileged OS Account**.
- Specify the **Virtual Postgres Port Number** of the target database. The default value is 5434
- (Optional) Define database configuration parameters during the provisioning operation in the "Config Settings" section.
    - During VDB creation, the **Config Settings** section provides the capability to configure the parameters of the VDB's `postgresql.conf` file. For example, if you wish to disable SSL via UI, you can specify the "ssl" parameter and its value through the UI. This action will replace the existing parameter value in the M file. Similarly, if you want to disable a parameter then specify the parameter name and then select the **Comment Property** check box. This action will comment out the parameter in the `postgresql.conf` file.



6. On the **Configuration** page, perform the following:
   - Specify the **VDB Name** of the PostgreSQL VDB.
   - Select the **Target Group** to organize the VDB under a dataset group
     - Click the “Add Dataset Group” button to create a new group.
7. (Optional) On the **Policies** page, select a previously defined **VDB Snapshot Policy** or the **Use group policy** checkbox.
8. (Optional) On the **Masking** page, enable the **Mask this VDB** checkbox to mask the database following a successful provision.
   - This feature may be disabled or unavailable if the chosen snapshot is already masked or the Delphix Continuous Compliance Engine is disconnected.
9. (Optional) On the **Hooks** page, select a Hook Point and then click the plus button (**+**) to add a script that you want to run. You can define scripts to run at multiple hook points in the process.
10. In the **Summary** tab, review the configuration profile for your VDB.
11. Click **Submit**.

9.9.7.3.3 Provisioning a PostgreSQL virtual database Point-in-Time (PiT) with external logs

Starting with PostgreSQL connector version 4.0.0 and later, you can leverage Point-in-Time recovery to provision a VDB by providing the WAL files on the target environment. PostgreSQL allows you to perform a

Point-in-Time (PiT) recovery by replaying archived WAL files from a base backup. PiT recovery relies on three components.

- **Base Backup:** This represents the file system backup, commonly referred to as a dataset snapshot.
- **Archived WAL Files:** These are the archived Write-Ahead Logging (WAL) files that contain all transaction changes.
- **Recovery Target Timestamp:** The specific timestamp at which you intend to recover your database. This value is greater than the backup's timestamp and has been recorded by the provided WAL files.

For more information about WAL file archiving and Point-in-Time (PiT) Recovery, refer to the PostgreSQL Documentation[547].

9.9.7.3.3.1 Prerequisites

Ensure that you meet the following requirements before you begin creating a dSource.

1. Create a dSource snapshot from one of the following ingestion methods:
   - dSource snapshot created using WAL Shipping with Externally Initiated backup (see page 1831)
   - dSource snapshot created using PostgreSQL streaming replication with Externally Initiated backup (see page 1807)
   - dSource snapshot created using PostgreSQL streaming replication with Delphix Initiated backup (see page 1812)

> ⚠ Other snapshot sources are not supported, such as VDBs, Staging Push ingestion dSources, or Delphix Initiated Single Database backup ingestion dSources.

2. Set up a PostgreSQL target environment that satisfies all the necessary requirements described in PostgreSQL support and requirements (see page 1780).
3. The target environment must have the required WAL Logs (at least from the REDO WAL Log captured in the snapshot metadata) to perform point-in-time recovery. For more information, refer to Copy WAL logs to the target environment (see page 1843).
4. Identify a target timestamp value with the following criteria:
   - Greater than the checkpoint timestamp available in the dSource's snapshot metadata.
   - Recorded by the WAL files.
   - Follows ISO 8601 basic format.

   For more information, refer to Understanding the target timestamp value (see page 1831).
5. To use privilege elevation
   - You must meet the Privilege Elevation Requirements (see page 1780).
   - The high-privilege user must have the following permissions:

547 https://www.postgresql.org/docs/current/continuous-archiving.html

- access to the directory that contains the WAL logs and read+execute permissions on the same.
- read permission on individual WAL Logs.

> If the VDB PiT recovery is configured to use the same Write-Ahead Logging (WAL) path as provided to the dSource created using Log Shipping with Externally initiated Backup (in connector versions below 4.0.0), the provisioning of the VDB PiT may fail. This failure can manifest with the error message "cp: cannot stat," which can be observed in the PostgreSQL logs. The reason for the error is that during the restore process, the dSource renames the WAL Logs with a ".done" suffix (e.g., 000000010000000000000020.done). To prevent this error from occurring, it is advisable to proactively execute the **Resynchronize dSource** operation on the dSource and then provision the VDB PiT.

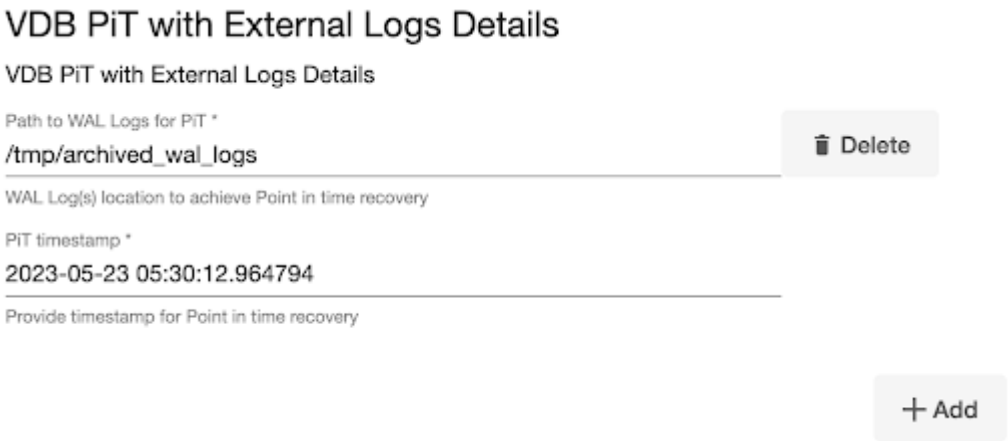
9.9.7.3.3.2 Procedure

Perform the following steps to perform a PiT recovery.

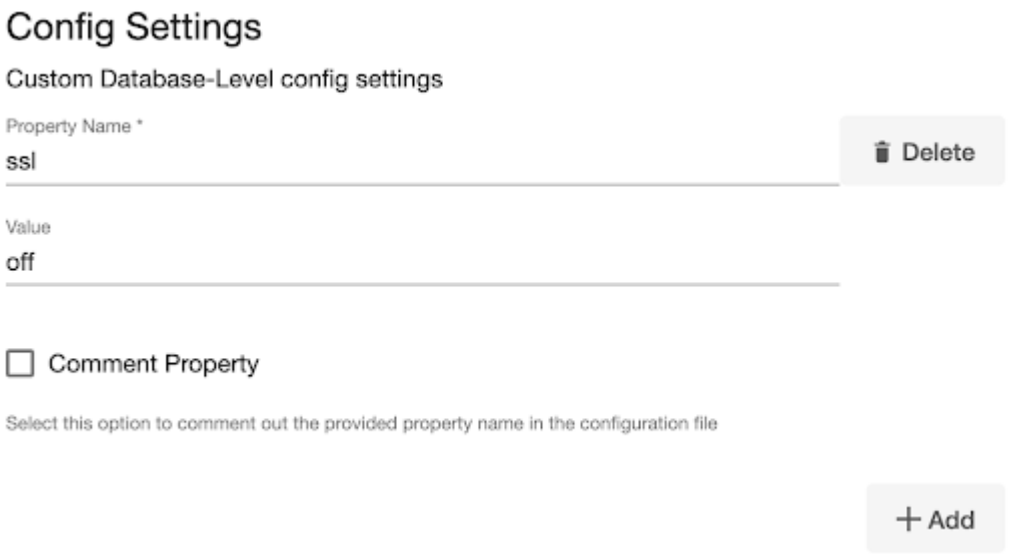
1. Login to the **Delphix Management** application.
2. Navigate to the **Manage > Datasets** page.
3. Select the **VDB PiT supported dSource** and the snapshot that contains the data you want to provision from. Click the provision VDB icon.
4. On the **Target Environment** page, perform the following actions:
   a. From the **Environment** dropdown list, select a target environment. This will dynamically populate the other properties.
   b. From the **Installation** dropdown list, select the specific PostgreSQL installation that you want to use.
   c. From the **User** dropdown list, select the environment user that matches the PostgreSQL Instance Owner as configured in the prerequisites.
   Note: The ability to select different instance owners is only applicable when you have multiple users configured in that environment.
5. On the **Target Configuration** page, perform the following actions:
   a. Specify the target environment's **NFS Mount Location**. The specified path must not exist or be empty.

   b. (Optional) Enable **Privilege Elevation** by providing the username of the **Privileged OS Account**.
   c. Specify the Virtual Postgres Port Number of the target database.

6. Select the **VDB PiT with External Logs Details** section's **+ Add** button and provide the following: **Note**: Users must click the "Add Button" for **VDB PiT with External Logs details** only once as the solution supports a single backup path. Providing multiple paths will result in an error during the linking process.
   a. **Path to WAL Logs for PiT:** Provide an **absolute path** to the archived WAL files available on the target environment.
   b. **PiT timestamp:** Provide the target timestamp.



7. Optionally, you can define database configuration parameters during the linking operation in the "Config Settings" section if needed.



Note: During dSource creation, you have the capability to configure the parameters and their values in the `postgresql.conf` file using the "Config Settings" section in the UI. For example, if you wish to disable SSL via UI, you can specify the "ssl" parameter and its value through the UI. This action will update the existing parameter value in the `postgresql.conf` file with the value provided. Similarly, if you want to disable a parameter then you can specify the parameter name and then select the **Comment Property** check box. This action will comment out the parameter in the `postgresql.conf` configuration file.

8. Click **Next**.
9. On the **Configuration** page, provide the name of the PostgreSQL VDB that will be displayed on the UI.
10. Select a **Target Group** for the VDB to select an existing group or click the green **Plus** icon to add a new group, if necessary.
11. In the **Policies** tab, apply policy details to the dSource if needed, and then click **Next** to continue.
12. In the **Hooks** tab, select a Hook Point and then click **+** to add a script that you want to run at that hook point. You have the flexibility to define scripts to run at multiple hook points in the process.
13. In the **Summary** tab, review the configuration profile for your dSource.
14. Click **Submit**. The VDB provisioning must complete and create a new VDB timeflow.

For more information on VDB Database permissions, refer to Database authentication for PostgreSQL[548]datasets.

After a successful VDB provision, two snapshot metadata values can be used to identify the operation's results:

- **User-Provided Recovery Timestamp**. This timestamp records the timestamp specified by the user.
- **PiT recovery Restore Timestamp**. This timestamp records the timestamp of the last committed transaction that the recovery process has reached up to.

548 https://cd.delphix.com/docs/latest/database-permissions-for-provisioned-postgresql-vd

9.9.7.3.3.3 Understanding the target timestamp value

In PostgreSQL, a recovery point is defined by the transaction commit timestamp. During the recovery process, the database is restored to a state where the commit timestamp is less than or equal to the provided target time. Follow the requirements above in the Prerequisite section and the additional following details to ensure a successful provision.

ISO 8601 string format

The connector accepts timestamps in ISO 8601 basic format. The permitted timestamp formats can be summarized as follows:

- Timezone inherited from the postgresql.conf file:
  - YYYY-mm-dd HH:MM:SS
  - YYYY-mm-dd HH:MM:SS.ssssss
- Time Zone is defined and a UTC offset:
  - YYYY-mm-dd HH:MM:SS(+/-)HHMM
  - YYYY-mm-dd HH:MM:SS.ssssss(+/-)HHMM

Here YYYY represents the year, mm represents the month, dd represents the day, HH represents the hour in 24-hour format, MM represents the minutes, SS represents the seconds and ssssss represents the microseconds. If a timezone is not specified, it will be obtained from the virtual database's `postgresql.conf` file.

Additional details

- Starting with PostgreSQL version 13 and higher, performing a PiT recovery will fail if there have been no transactions or activity occurring after the requested target time.
- You can use pg_waldump (or pg_xlogdump for PostgreSQL 9.x versions) as a tool to analyze WAL files and identify commit timestamps.

9.9.7.3.4 Provision PostgreSQL operations via CLI

To create a VDB via CLI, you must first create a dSource. For more information on creating a dSource, refer to Link operation via CLI (see page 1825) section.

To create a VDB and perform operations via CLI, log in to the Delphix Continuous Data Engine using the admin user account.

```
ssh admin@<CONTINUOUS-DATA-FQDN-or-IP>
```

9.9.7.3.4.1 Provision

Run the following commands to provision a VDB. The following example uses `testdb` dSource.

```
database provision
set type=AppDataProvisionParameters
set container.name=testvdb
set container.group=Untitled
set source.allowAutoVDBRestartOnHostReboot=true
set sourceConfig.repository="rh86-pg15-t1.dlpxdc.co/Postgres vFiles (15.2)"
set timeflowPointParameters.container=testdb
set sourceConfig.name=testscfgv
set source.parameters="{\"mountLocation\":\"/tmp/singldb2v\", \"privilegedOsUser\":
\"\", \"postgresPort\":5443, \"configSettingsStg\":[{\"propertyName\":
\"wal_keep_segments\", \"value\": \"8\"}]}"
commit
```

9.9.7.3.4.2 **Disable**

Run the following commands to disable a dSource. You must select the same sourceConfig name that was given during VDB creation.

```
source
select testscfgv
disable
commit
```

9.9.7.3.4.3 **Enable**

Run the following command to enable a dSource.

```
source
select testscfgv
enable
commit
```

9.9.7.3.4.4 **Stop**

Run the following command to stop a dSource.

```
source
select testscfgv
stop
commit
```

9.9.7.3.4.5 Start

Run the following command to start a dSource.

```
source
select testscfgv
start
commit
```

9.9.8 PostgreSQL troubleshooting

This section covers the following topics:

- Advanced configuration (see page 1838)
- Logging (see page 1839)
- Managing log rotation and log level (see page 1840)

9.9.8.1 Advanced configuration

This topic details customizing PostgreSQL connector logging and controlling supported dataset configurations. You can modify the following parameters at any stage of the dataset lifecycle.

Parameter value	Usage
MAX_FILE_SIZE_KB	This parameter allows you to set the maximum size for the connector log files.
NUM_OF_FILES_TO_KEEP	This parameter allows you to define the maximum number of connector log files to retain.
LOG_LEVEL	This parameter allows you to adjust the connector debug log levels.
PGCTLTIMEOUT	This parameter allows you to set the PG_CTL timeout value for the database. This parameter defines the time (in seconds) to wait for the PostgreSQL start/stop (using pg_ctl) operation to complete. It can be customized to compensate for any network latency between the Delphix Continuous Data Engine and the remote environment. The default value is 600 seconds.

Parameter value	Usage
PIT_VDB_RECOVERY_MAX_WAIT_TIME	This parameter allows you to set the maximum wait time for PiT VDB recovery. Provisioning of VDB PiT with External Logs has a time-bounded wait for the recovery to complete. The PIT_VDB_RECOVERY_MAX_WAIT_TIME defines the maximum period (in seconds) the connector will wait for the recovery operation to successfully complete. The default value is 7200 seconds (i.e. 2 hours).

9.9.8.1.1 Connector config file

After adding the environment to Delphix Continuous Data Engine, you can find the connector configuration file, "dlpx_postgres_config.ini," on the remote environment. It is typically located at the following specified path:

```
<toolkit path>/Delphix_COMMON_<long id>/plugin/postgres-vsdk_12c9a11f-fab6-4792-
a5da-49d726c004df/dlpx_postgres_config.ini
```

This file is divided into 3 sections - `log_config`, `database_config`, and `plugin_config`.

```
# Sample of file dlpx_postgres_config.ini
[log_config]
MAX_FILE_SIZE_KB = 10240
NUM_OF_FILES_TO_KEEP = 50
LOG_LEVEL = INFO # Accepted values: DEBUG, INFO, WARNING and ERROR

[database_config]
PGCTLTIMEOUT = 600


[plugin_config]
PIT_VDB_RECOVERY_MAX_WAIT_TIME = 7200
```

9.9.8.2 Logging

You can find the PostgreSQL connector logs within the scratch path on the remote environment, located at the following path:

```
<toolkit path>/Delphix_COMMON_<long id>/plugin/postgres-vsdk_12c9a11f-fab6-4792-
a5da-49d726c004df/<environment_user>_logs/delphix_postgres_debug.log
```

You can configure the logging mechanism by modifying the following parameters defined in the *dlpx_postgres_config.ini file*[549]. This feature offers options to provide customized connector logging.

1. MAX_FILE_SIZE_KB - sets the maximum connector log file size.
2. NUM_OF_FILES_TO_KEEP - sets the maximum number of connector log files.
3. LOG_LEVEL - sets the connector debug log levels.

9.9.8.3 Managing log rotation and log level

9.9.8.3.1 Setting maximum size

MAX_FILE_SIZE_KB - You can set this parameter to customize the maximum size of the active log file in KB. Once this limit is reached, the connector will automatically initiate the log rotation. The default/maximum value of this parameter is set at 10240 KB/10 MB. This parameter accepts only a positive integer value, such as 10240.

9.9.8.3.2 Setting the number of log files to retention

NUM_OF_FILES_TO_KEEP - You can set this parameter to customize the retention number for the rotated log files. The default value of the parameter is set to 50, which means that the connector will retain a maximum of 50 rotated log files and each file will be of a size as defined in the `MAX_FILE_SIZE_KB` parameter. If the number of files exceeds this value, then the old files will be deleted in a First In First Out (FIFO) order.

9.9.8.3.3 Setting the log level

LOG_LEVEL - You can define the `LOG_LEVEL` parameter to set the amount of logs being generated on the remote environment. Accepted values are **DEBUG, INFO, WARNING,** and **ERROR**, with **INFO** being the default value. The DEBUG setting generates the highest amount of logs and ERROR, the least.

For known issues, refer to the Known issues[550] section from the release notes.

9.9.9 PostgreSQL additional information

This section covers the following topics:

- PostgreSQL glossary (see page 1841)
- Important PostgreSQL connector settings (see page 1842)
- PostgreSQL architecture for streaming replication (see page 1842)

549 https://docs.google.com/document/d/1WUkjGmv194W3WQ_UW5NxHjTFzjUPpF_ZZw0RdgvrN-8/edit#heading=h.2feelut4n0ly
550 https://cd.delphix.com/docs/latest/known-issues-postgresql

9.9.9.1 PostgreSQL glossary

9.9.9.1.1 File-Based Log Shipping

This is a method of syncing parent and child databases by applying WAL log files. Each WAL log file can store up to 16MB of data, and these files are sent only when they reach this size threshold. Note: This approach will cause a replication delay and also increase the chances of losing data if the parent database crashes and logs are not archived.

9.9.9.1.2 Hot standby (aka Streaming Replication)

This mechanism provides asynchronous binary replication to one or more standby databases. Standby databases may also become hot standby databases meaning they can be queried as a read-only database. This is the fastest type of replication because WAL data is sent immediately, without waiting for a complete segment to be produced and shipped. At the hot standby level, the same information is logged as with the archive, in addition to the information needed to reconstruct the status of running transactions from the WAL. To enable read-only queries on a standby server, wal_level must be set to 'hot_standby' or higher on the primary database, and hot_standby must be enabled in the standby database.

9.9.9.1.3 Streaming WAL records

WAL record chunks are streamed by database servers to maintain data synchronization. Standby servers establish connections with the parent server to receive these WAL chunks. WAL records are streamed as soon as they are generated and do not wait for the WAL file to be filled. *This allows a standby server to stay more up-to-date than is possible with file-based log shipping.* By default, streaming replication is asynchronous although it also supports synchronous replication.

9.9.9.1.4 WAL sender

WAL sender is a process in the primary server that is responsible for sending segments to a receiver**.**

9.9.9.1.5 WAL receiver

It is a process, in the standby server, that continuously listens for any incoming WAL segments from the primary server in its receive queue and applies the same on the source database.

9.9.9.1.6 WAL segments

The WAL files are stored as a set of WAL segment files. By default, each segment is 16 MB, and it is internally divided into pages of 8192 bytes (8 KB).

9.9.9.1.7 Warm standby (aka Log Shipping)

This is a high availability (HA) solution that replicates a database cluster to either an archive or a warm standby server (can be brought up quickly, but not available for querying). It offers minimal overhead and is

easy to set up. This solution is well-suited for scenarios where the primary objectives are continuous backup and achieving short failover times.

9.9.9.1.8 Write Ahead Logging (WAL)

These are files that record all modifications made to the database before they are actually applied/written to data files.

9.9.9.2 Important PostgreSQL connector settings

Parameter/ Filename	Usage
pg_hba.conf	The pg_hba.conf file is used to specify the IP addresses from which the server will accept logins, as well as which databases and users are allowed access.
wal_level	This parameter determines the level of information that is written to the WAL files.
max_wal_senders	This parameter defines the maximum allowable number of concurrent connections from standby servers or streaming base backup clients. It should be increased by two (+2) from its desired value.
archive_command	This parameter sets the shell command that will be called to archive a WAL file.

9.9.9.3 PostgreSQL architecture for streaming replication

Transaction logs or Write-Ahead Logging (WAL) is replayed from the source to the dSource to ensure data synchronization between the two.

The architectural setup involves the following steps:

1. Ensure that the end-user application connecting to the PostgreSQL connector source database can perform both the read/write queries on the database.
2. Understand that the database changes are recorded as WAL segments in the PostgreSQL connector database under the directory **pg_xlog** or **pg_wal**.
3. Configure the source PostgreSQL connector server as a Standby Node.
4. Configure replication security by creating a replication user and specifying the authentication protocol.
5. Initiate a base backup on the standby environment.

6. Configure `postgresql.conf` file as per the source environment.
7. Start the standby server.

9.9.10 PostgreSQL FAQs

This section covers the following topics:

- How to copy WAL files to the target environment? (see page 1843)
- Database authentication methods for PostgreSQL connector datasets (see page 1844)
- How to use the Delphix Continuous Data Engine key pair? (see page 1845)

9.9.10.1 How to copy WAL files to the target environment?

Use any one of the following methods to copy the WAL files to the target environment:

1. Archiving the WAL files on the source environment and mounting (NFS) the archive directory on the target environment. Perform the following steps to enable WAL archiving:
   a. Set the wal_level configuration parameter to replica or higher.
   b. Set the archive_mode to on.
   c. Use the archive_command parameter to copy the WAL files to an archive directory, ex: cp %p <ARCHIVE_DIRECTORY>/%f
2. Use the pg_receivewal utility on the target environment to start a background process. This process will collect the WAL files from the source database using streaming replication.

> This process will require increasing the max_wal_senders count in the source configuration (`postgresql.conf`) and collects only the WAL files from the time you start the process.

3. Copying the required WAL files from the archive directory on the source environment to the target environment.
4. Use unstructured files to mount the archive directory on the source environment as a dSource and later provision the VFiles on the target environment (Delphix recommended approach). With this approach, the archived WAL files can be kept in sync using the dSource SnapSync and VFiles refresh operations.

> For more information, refer to the Unstructured files and app data[551] section.

551 https://cd.delphix.com/docs/latest/unstructured-files-and-app-data

9.9.10.2 Database authentication methods for PostgreSQL connector datasets

This topic describes database authentication on provisioned PostgreSQL connector virtual databases.

9.9.10.2.1 PostgreSQL authentication

PostgreSQL connector offers different client authentication methods. The method used to authenticate a particular application/client connection can be selected based on the (client) environment address, database, and user.

PostgreSQL connector database user names are logically separate from user names of the operating system in which the server runs. If all the users of a particular server also have accounts on the server's machine, it makes sense to assign database user names that match their operating system usernames. However, a server that accepts remote connections might have many database users who have no local operating system account, and in such cases, any connection between database user names and environment user names is not needed.

9.9.10.2.2 Delphix Continuous Data Engine PostgreSQL connector authentication

The Delphix Continuous Data Engine PostgreSQL connector requires the staging and target environments to already have the necessary users and authentication systems created/installed on them. Delphix Continuous Data Engine will neither create users nor change database passwords as part of the provisioning process. As a best practice, use a secure PostgreSQL connector authentication mechanism on the source, staging, and target servers.
To connect to the PostgreSQL connector VDB. Add an entry in pg_hba.conf like below.
You can choose the method as per your requirements.

```
# TYPE  DATABASE        USER            ADDRESS                 METHOD
host    all             all             0.0.0.0/0               scram-sha-256
```

The following section describes some important authentication methods used by PostgreSQL connector architecture.

9.9.10.2.2.1 Trust authentication [552]

When trust authentication is specified, the PostgreSQL connector assumes that anyone who can connect to the server is authorized to access the database with whatever database user name they specify (even superuser names). Of course, restrictions made in the database and user columns still apply. This method should only be used when there is adequate operating-system-level protection on connections to the server.

552 https://www.postgresql.org/docs/9.6/auth-methods.html#GSSAPI-AUTH:~:text=20.3.1.-,Trust%20Authentication,-When%20trust%20authentication

9.9.10.2.2.2 Password authentication[553]

The password-based authentication methods are md5 and password. These methods operate similarly except for the way that the password is sent across the connection, namely MD5-hashed and clear-text respectively.

PostgreSQL connector database passwords are separate from operating system user passwords. The password for each database user is stored in the pg_authid system catalog. Passwords can be managed with the SQL commands CREATE USER[554] and ALTER ROLE[555], e.g., CREATE USER 'foo' WITH PASSWORD 'secret'. If no password has been set up, the stored password is null and password authentication will always fail.
Some more authentication mechanisms are less often used. For more information, refer to the following PostgreSQL connector links.

- GSSAPI Authentication[556]
- SSPI Authentication[557]
- Ident Authentication[558]
- Peer Authentication[559]
- LDAP Authentication[560]
- RADIUS Authentication[561]
- Certificate Authentication[562]
- PAM Authentication[563]
- BSD Authentication[564]

9.9.10.3 How to use the Delphix Continuous Data Engine key pair?

Perform the following steps to use the Delphix Continuous Data Engine key pair:

1. Select **Public Key** for the **Login Type.**
2. Click **View Public Key**.
3. Copy the public key that is displayed, and append it to the end of your ~/.ssh/authorized_keys file. If this file does not exist, you must create it.
    a. Run chmod 600 ~/.ssh/authorized_keys to allow only the file's owner to read and write to it (make sure the file is owned by you).

553 https://www.postgresql.org/docs/9.6/auth-methods.html#GSSAPI-AUTH:~:text=20.3.2.-,Password%20Authentication,-The%20password%2Dbased
554 https://www.postgresql.org/docs/9.6/static/sql-createuser.html
555 https://www.postgresql.org/docs/9.6/static/sql-alterrole.html
556 https://www.postgresql.org/docs/9.6/auth-methods.html#GSSAPI-AUTH
557 https://www.postgresql.org/docs/9.6/static/auth-methods.html#SSPI-AUTH
558 https://www.postgresql.org/docs/9.6/static/auth-methods.html#AUTH-IDENT
559 https://www.postgresql.org/docs/9.6/static/auth-methods.html#AUTH-PEER
560 https://www.postgresql.org/docs/9.6/static/auth-methods.html#AUTH-LDAP
561 https://www.postgresql.org/docs/9.6/static/auth-methods.html#AUTH-RADIUS
562 https://www.postgresql.org/docs/9.6/static/auth-methods.html#AUTH-CERT
563 https://www.postgresql.org/docs/9.6/static/auth-methods.html#AUTH-PAM
564 https://www.postgresql.org/docs/9.6/auth-methods.html#AUTH-BSD

b. Run chmod 755 ~ to restrict access to your home directory so that no other may write to it.
c. Run chmod 700 ~/.ssh so that others cannot write to it. The ~/.ssh directory is not writable by group or others. Otherwise, authentication will fail.

9.10 SAP ASE data sources

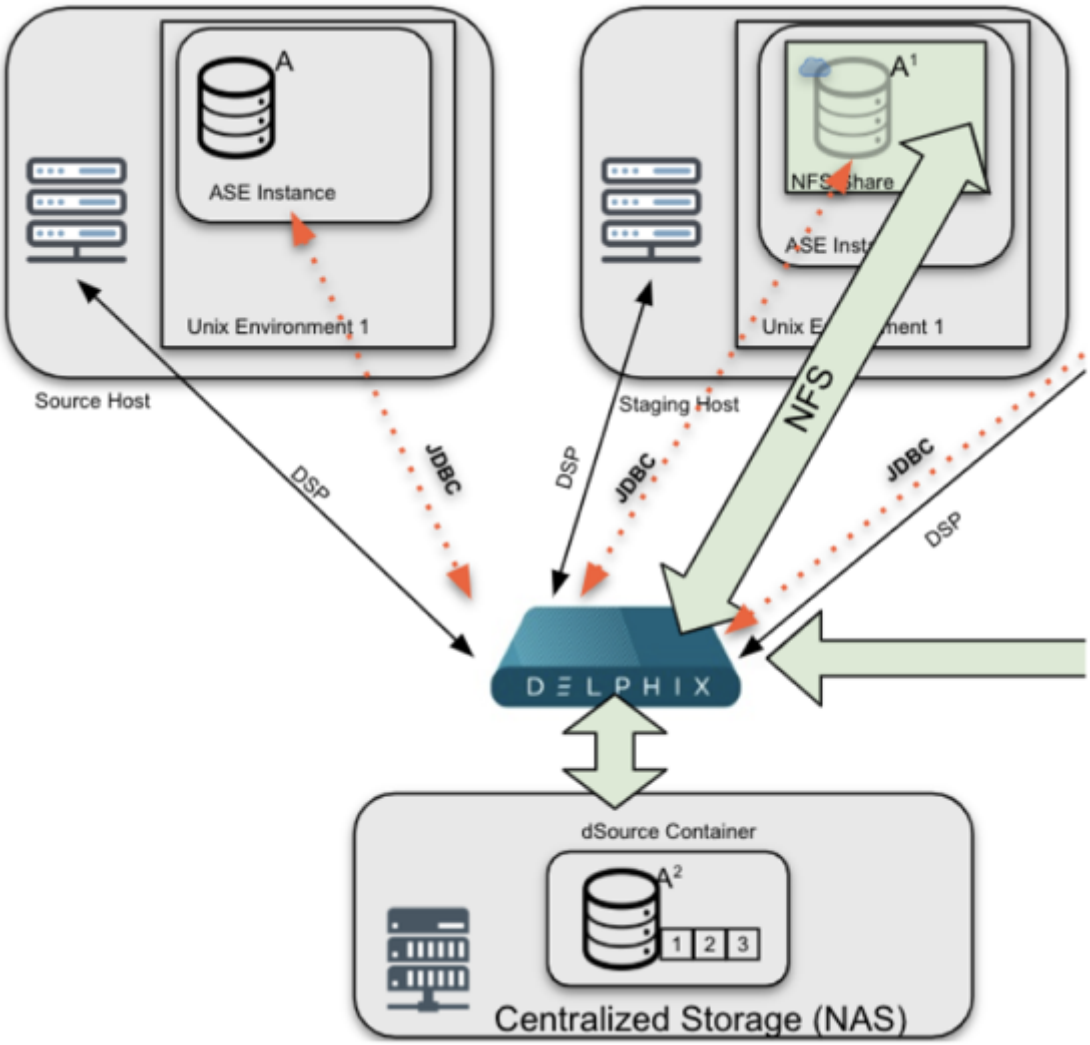
This section covers the following topics:

- Delphix architecture with SAP ASE (see page 1846)
- Overview of ASE database encryption (see page 1850)
- TLS security for Sybase ASE (see page 1853)
- Quick start guide for SAP ASE (see page 1855)
- SAP ASE support and requirements (see page 1877)
- Managing SAP ASE environments and hosts (see page 1903)
- Linking data sources and Syncing Data with SAP ASE (see page 1917)
- Provisioning and managing VDBs from SAP ASE (see page 1929)
- Backup server best practices (see page 1947)
- SAP ASE hook operations (see page 1948)
- Support for dump history file (see page 1952)

9.10.1 Delphix architecture with SAP ASE

This topic describes the high-level process for adding SAP ASE-supported environments, linking SAP ASE databases to the Delphix Engine, and provisioning virtual databases.

Diagram of underlying Linking Architecture to support ingestion workflows between SAP ASE-supported Environments and the Delphix Engine.

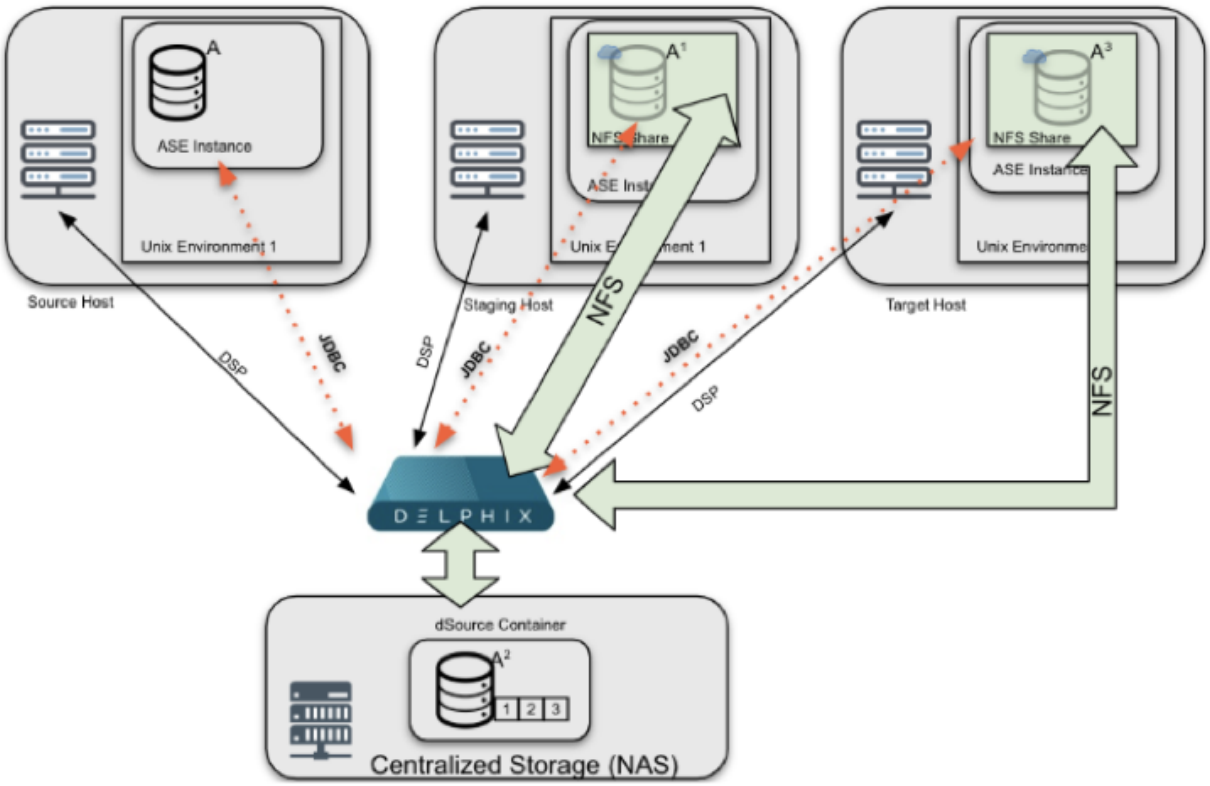


9.10.1.1 Linking architecture between SAP ASE and Delphix engine

The diagram featured above shows a common architecture for SAP ASE-based systems. The diagram shows the infrastructure used to ingest data. (From the above diagram) The workflow starts with a Source Host (Left, which in this example is a Production Database Server), that provides critical data in the form of dump files (used for SnapSync) and transaction logs (used for LogSync).

The Delphix Engine (Bottom), continuously monitors the source database to determine when new dumps are available. When a dump is available, the Delphix Engine will contact the SAP ASE Staging host (Right) via Delphix Session Protocol (DSP). The Staging host will read the dumps from the Source Host and recover them through a Staging Database that is automatically set up on Delphix NFS storage. Once recovery is complete, the backup data is incorporated into the Delphix dSource as a new snapshot card and is available for use to provision a new virtual or physical database.

Diagram of underlying provisioning architecture to support VDB provisioning workflows between the Delphix engine and SAP ASE-supported target environments.



If no remote load location (aka Source Host) is configured, dump files must be made available on the staging host via NFS or some other copy mechanism. If no remote load location is configured, the Delphix Engine searches for the files on the staging server. If they are not found, the Delphix Engine cannot load them

SAP ASE provisioning architecture

This diagram is an extension of the previous diagram with the new content showing how Delphix provisions Virtual Databases (VDBs) to a Target Environment. The Delphix Engine (bottom) creates a set of virtual files from a snapshot that becomes the VDB. These files are presented to the Target host (right) via NFS. Delphix uses DSP to communicate with Target and initiates the creation of a new database. Once complete, the VDB is brought online and made available for use.

9.10.1.2 Environment setup

9.10.1.2.1 Target and staging environments for SAP ASE

For SAP ASE-based Delphix Architectures, the Staging Host plays a critical role:

1. Provide the “staging” point in which the Delphix Engine coordinates data ingestion to a dSource by restoring backups to a staging database and creating a dSource from that ingested data.
2. Tracks changes on disk by running validated sync, a process that identifies when new backups are available and initiates the ingestion process if new backups and/or transaction log backups are found.
3. Minimizes touch to production by providing an intermediate host between source databases and Delphix.

For SAP ASE-based Delphix architectures, the target host has two potential roles:

1. Host a target environment for provisioning virtual databases (VDBs).

> Staging and Target hosts can be run on a target server, although a dedicated staging server is recommended for optimal performance.

SAP ASE dSources are representations of a Source Database replica on the staging database that runs on a target or staging host. There is no requirement for additional local storage with either host option, as the storage is mounted over NFS from the Delphix Engine. For a deeper, technical discussion, please see the Technical Deep Dive section below.

At Delphix, we refer to the creation and maintenance of this staging database on the staging host as "validated sync," because it prepares the dSource data on the Delphix Engine for provisioning VDBs later on. After the Delphix Engine creates the staging database, it continuously monitors the source database for new transaction log/Full backups (if “truncate log on checkpoint” is active, Delphix will need Full Backups for Validated Sync). When it detects a new transaction log backup, it restores that backup to the staging database. The result is a TimeFlow with consistent points from which you can provision a VDB, and a faster provisioning process, because there is no need for any database recovery during provisioning. If Log Sync is enabled, you may provision a VDB to a point in time, in between snapshots.

When you later provision a VDB, you can specify any environment as a target, including the environment that contains the staging database. However, for best performance, we recommend that you choose a different target environment. The only requirement for the target is:

- It must have an operating system that is compatible with the one running on the validated host.

9.10.1.3 Target hosts for ASE

9.10.1.3.1 Container for VDBs

This topic describes the basic concepts involved with provisioning VDBs from SAP ASE dSources or even other SAP ASE VDBs.

A dSource is a virtualized representation of a physical or logical source database. As a virtual representation, it cannot be accessed or manipulated using database tools. Instead, you must create a virtual database (VDB) from a dSource snapshot. A VDB is an independent, writable copy of a dSource snapshot. You can also create VDBs from other VDBs. Once you have provisioned a VDB to a target environment, you can also implement snapshot and retention policies for the VDB, which will determine how frequently Delphix Engine

will take a database snapshot and how long the snapshots will be retained for recovery and provisioning purposes.

When provisioning a VDB, Delphix creates the database with the default SAP ASE database options. If the database options have been altered on the source database and you wish for the VDB to reflect these same options, they would need to be altered via a Post-Script or by Hook Scripts for Automation and Customization (see page 1060)

For an overview of the high-level components involved in provisioning an SAP ASE VDB, see Overview of Provisioning SAP ASE Virtual Databases (see page 1929)

9.10.1.4 Validated sync and logSync

To run validated sync, a staging environment must be specified to host a staging database for the validated sync process. In this process, the Delphix Engine continuously monitors the source database for new full and transaction log backups if the source database is using a simple recovery model, and only transaction log backups if using a full recovery model. When it detects a new backup, it restores that backup to the staging database with the storage residing in Delphix. The result is a Timeflow with consistent points from which you can provision a VDB, also known as snapshots.

> If the “truncate log on checkpoint” is set, Delphix will only apply full backups.

Snapshots accumulate over time. To view a snapshot:

1. From the Datasets panel, click the group containing the dSource.
2. Select dSource.
3. Click the Timeflow tab.

Each snapshot is displayed and includes information about the source database, operating system, and time stamp. You can scroll through these cards to select the one you want, or you can enter a date and time to search for a specific snapshot.

Once you have provisioned a VDB, you can also take snapshots of it. As with the dSource snapshots, you can find these when you select the VDB in the Datasets panel. You can then provision additional VDBs from these VDB snapshots.

9.10.1.4.1 Dependencies

If there are dependencies on the snapshot, you will not be able to delete the snapshot free space; the dependencies rely on the data associated with the snapshot.

9.10.2 Overview of ASE database encryption

Beginning with 16.0, SAP ASE supports the Database Encryption feature. The SAP ASE Database Encryption feature encrypts the data at rest, without changing the applications. This encryption can be done on entire databases or only on columns. This ensures that the authorized users access the data and thus prevents the misuse of the data against theft and security breaches.

Data is encrypted with the help of encryption keys. These encryption keys are stored in the database in an encrypted form. You can encrypt an encryption key using a key encryption key (KEK).

In the SAP ASE Database Encryption, column and database encryption uses a symmetric encryption algorithm, which means that the same key is used for encryption and decryption. SAP ASE tracks the key that encrypts the data.

Starting 6.0.8.0, Delphix Engine will support the SAP ASE encrypted databases.

For more information on SAP ASE Database Encryption, see the SAP ASE Encryption Documentation[565]

9.10.2.1 Delphix implementation of database encryption

This topic describes various configurations to support encrypted databases with Delphix. Follow the mandatory steps below on the ASE instance that hosts the staging databases and virtual databases.

1. If the source database is not encrypted already.
   a. Install the license option ASE_ENCRYPTION.
   b. Create a master key that will serve as the KEK.
      Command

      ```
      > create encryption key master with passwd "sybase"
      ```

   c. If the database is not encrypted already, create the database encryption key and use it.
      Commands

      ```
      > create encryption key <encryption-key-name> for database encryption
      > sp_configure "number of worker processes", 2
      > alter database <database-name> encrypt with <encryption-key-name>
      ```

   d. Export the master key and the encryption key to a location that is shared among source, staging, and target hosts. The command-line version of the ddlgen[566] tool is located at *$SYBASE/$SYBASE_ASE/bin*. You need to find out this location for your instance if it is different.
      i. cd $SYBASE/$SYBASE_ASE/bin
      ii. ddlgen -Usa -Psybase -SASE160_SRC -TEK -N master.dbo.master -XOD -O<shared-path>/master_ddl.sql
      iii. ddlgen -Usa -Psybase -SASE160_SRC -TEK -N master.dbo.<key_name> -XOD -O<shared-path>/<key_name>_ddl.sql
   e. Enable encryption in SAP ASE by executing the below command on the staging/target instance.
      Command

565 https://help.sap.com/doc/a613310dbc2b1014ade0f78b6ecb68ec/16.0.3.3/en-US/SAP_ASE_Database_Encryption_en.pdf
566 https://help.sap.com/viewer/9bedfba95c784f7a99c4fd926063de12/16.0.2.4/en-US/a7ed349ebc2b1014b50fbd20d6921128.html

```
> sp_configure 'enable encrypted columns', 1
```

2. Import the keys on the staging and target instances by running the below commands from the directory where the SQL files are present (the shared location between instances) or mention the entire path of the files to be imported.
   a. isql -Usa -Psybase -SASE160_TGT -w 220 -i master_key.sql
   b. isql -Usa -Psybase -SASE160_TGT -w 220 -i <key_name>.sql
3. Set the encryption password by executing the below command on the staging/target instance.

   Command

```
> set encryption passwd "sybase" for key master
```

4. Setup for automatic master key access. Refer create the master key start-up file[567]. In order to avoid issues on the master key password after the reboot of the ASE instance, a master key startup file needs to be created by running the following steps on the staging and the target instance.
   a. Command

```
> sp_configure 'automatic master key access',1
```

   b. Command

```
> alter encryption key master with passwd 'sybase' add encryption for
 automatic_startup
```

   c. Command

```
> sp_encryption mkey_startup_file,default_location,sync_with_mem
```

   d. Verify if the master key startup file has been successfully created on the instance.
   Command

```
> sp_encryption mkey_startup_file
```

   e. Reboot the ASE instance to get the master key startup file in effect.

567 https://help.sap.com/viewer/6a68f933609a46fabe13f6e4e72ec3ac/16.0.0.0/en-US/a7c39fe1bc2b1014b4eec6e03cf126c5.html

> If you perform a reboot or plan to perform a reboot on the source host, then you would need to repeat step 4 on the source host as well. By doing so, you don't need to set the master key password again after reboot.

9.10.3 TLS security for Sybase ASE

SAP offers a method to configure Sybase ASE instances utilizing TLS, enhancing security for connections to the Sybase ASE database.

Starting with release 23.0.0.0, Delphix Continuous Data Engine has introduced support for Sybase ASE instances configured with TLS. JDBC connections established by Delphix Continuous Data Engine to Sybase ASE instances can now be TLS enabled.

9.10.3.1 Implementing TLS support in the Delphix Continuous Data Engine

This feature enables the TLS support that provides the encrypted JDBC connections between Delphix Continuous Data Engine and Sybase hosts.

Delphix Continuous Data Engine requires the following steps to be taken to support TLS for Sybase ASE instances:

1. Disable all Sybase ASE dSources and VDBs running on the ASE instance to prevent connection failures during TLS setup.
2. Configure the Sybase ASE instance with TLS, ensuring the use of TLSv1.2.
3. If utilizing a self-signed certificate or a private PKI, you must include the signing certificate into the Delphix trust store. This can be accomplished by logging into Delphix Continuous Data Engine using the "sysadmin" user and adding the ASE instance's certificate in the setup application as instructed in the "**Adding a certificate**" section of the TrustStore settings (see page 989) page.
4. Restart the Delphix Continuous Data Engine by selecting the **Restart** button available under the three dots (...) menu at the top of the page where the certificate was added.
5. Access the CLI using the "admin" user and enable TLS support by setting the parameter `enableTls` to `true` for existing environments configured to use TLS following the above steps.

```
ip-10-110-225-100> environment
ip-10-110-225-100 environment> select Src
ip-10-110-225-100 environment 'Src'> update
ip-10-110-225-100 environment 'Src' update *> edit aseHostEnvironmentParameters
ip-10-110-225-100 environment 'Src' update aseHostEnvironmentParameters *> ls
Properties
    type: ASEHostEnvironmentParameters
    credentials:
        type: PasswordCredential
```

```
        password: ********
    dbUser: sa
    enableTls: false
    skipServerCertificateValidation: false
ip-10-110-225-100 environment 'Src' update aseHostEnvironmentParameters *> set
enableTls=true
ip-10-110-225-100 environment 'Src' update aseHostEnvironmentParameters *>
commit
    Dispatched job JOB-39
    ENVIRONMENT_UPDATE job started for "Src".
    ENVIRONMENT_UPDATE job for "Src" completed successfully.
```

6. Next, log in to the Delphix Continuous Data Engine UI using the "admin" user and initiate a refresh of the environment for which TLS support has been enabled. Upon successful completion of the environment refresh, navigate to the database tab on the environment page to locate the TLS port. If Delphix Continuous Data Engine encounters any issues discovering ASE instances (whether auto-discovered or manually added) running on the TLS port during the environment refresh operation, it will display an error or fallback to the TCP port if available, raising a corresponding fault.
7. Re-enable the dSources and VDBs that were disabled in the initial step.
8. If you want to create a new environment with TLS encryption enabled, you can do so by setting the parameter `enableTls` to `true` in `ASEHostEnvironmentParameters`.

```
ip-10-110-225-100 environment create *> ls
Properties
    type: HostEnvironmentCreateParameters
    hostEnvironment:
        type: UnixHostEnvironment
        name: Src (*)
        aseHostEnvironmentParameters:
            type: ASEHostEnvironmentParameters (*)
            credentials:
                type: PasswordCredential (*)
                password: ******** (*)
            dbUser: sa (*)
            enableTls: true (*)
            skipServerCertificateValidation: (unset)
        description: (unset)
        logCollectionEnabled: false
    hostParameters:
        type: UnixHostCreateParameters
        host:
            type: UnixHost
            address: kanojis-centos-79-sybase-160-src.dlpxdc.co (*)
            dspKeystoreAlias: (unset)
            dspKeystorePassword: (unset)
            dspKeystorePath: (unset)
            dspTruststorePassword: (unset)
            dspTruststorePath: (unset)
```

```
            javaHome: (unset)
            nfsAddressList: (unset)
            oracleHostParameters: (unset)
            privilegeElevationProfile: (unset)
            sshPort: 22
            sshVerificationStrategy: (unset)
            toolkitPath: /work (*)
    logCollectionEnabled: false
    primaryUser:
        type: EnvironmentUser
        name: sybase (*)
        credential:
            type: PasswordCredential
            password: ******** (*)
        environment: (unset)
        groupId: (unset)
        userId: (unset)
ip-10-110-225-100 environment create *> commit
    `UNIX_HOST_ENVIRONMENT-15
    Dispatched job JOB-159
    ENVIRONMENT_CREATE_AND_DISCOVER job started for "Src".
    ENVIRONMENT_CREATE_AND_DISCOVER job for "Src" completed successfully.
```

9. Do not set the parameter "**skipServerCertificateValidation**" to "**true**" until or unless you don't want to add the certificate to Delphix Continuous Data Engine. Setting it to "**true**" will skip the Sybase ASE server certificate validation during the TLS handshake. Delphix strongly advises against this setting, as it will compromise the integrity and security of encrypted communication by exposing it to the risk of interception and impersonation. If you still choose to compromise TLS security by not adding the certificate into Delphix Continuous Data Engine and setting the parameter `skipServerCertificateValidation` to `true`, then you can skip adding the certificate & restart the Delphix Continuous Data Engine as outlined above.

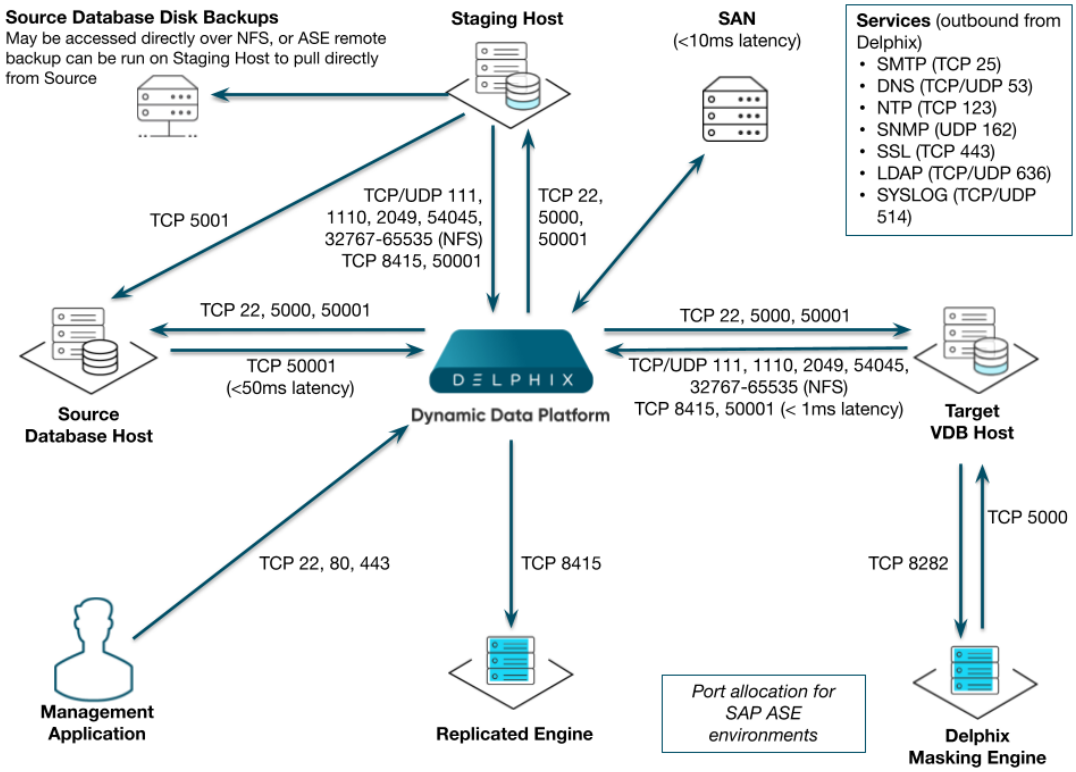
9.10.4 Quick start guide for SAP ASE

This quick start guide, which is excerpted from the larger User Guide, is intended to provide you with a focused overview of working with SAP ASE database objects in the Delphix Dynamic Data Platform. It does not cover advanced configuration options. It does not cover advanced configuration options or best practices for performance. It assumes that you are working in a Lab/Dev setting and attempting to quickly test Delphix Engine functionality. It assumes you will use the VMware Hypervisor.

Overview

In this guide, we will walk through deploying a Delphix Engine, starting with configuring SAP ASE Source and Target environments. We will then create a dSource, and provision a VDB. A detailed list can be referred to in the Network and Connectivity Requirements for SAP ASE Environments (see page 1892) section of the manual.

For purposes of the QuickStart, you can ignore any references to Replication or Masking, such as the engines shown in the diagram below.



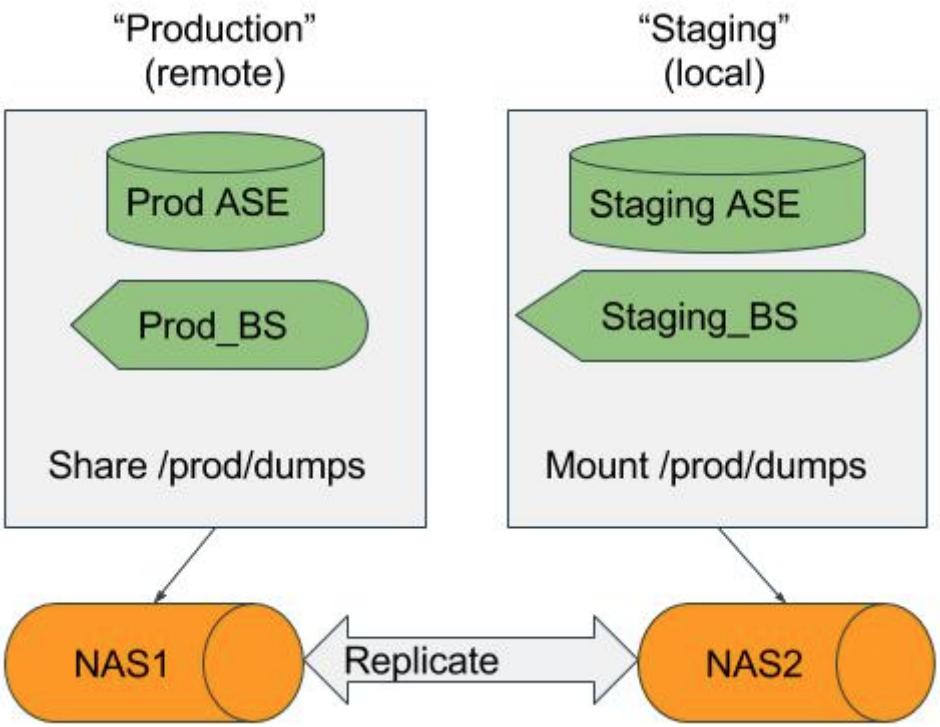
Delphix stays in sync with source databases by monitoring the ASE backup server log. When it sees a new database dump or transaction log has been created, it attempts to load it into the ASE staging instance.

Color	Supported?
Y	Yes
N	No

Validated sync Mode	SnapShot for FULL BACKUP	SnapShot for TLOGS	Point-in-Time Restore
truncate log on chkpt = true	Y	Y	Y
truncate log on chkpt = false + LogSynctruncate log on chkpt = false	N	Y	N
truncate log on chkpt = true + Log Sync	Y	N	N

Validated sync Mode	SnapShot for FULL BACKUP	SnapShot for TLOGS	Point-in-Time Restore
truncate log on chkpt = false + LogSync	N	Y	Y

Before proceeding with setting up environments, decide where the databases will be located relative to the ASE staging instance. In the following diagram, we have two ASE hosts (production and staging).



If the database dumps (or transaction logs) are available locally to the staging Backup Server, Delphix considers them "local". Whereas if they are available to a Backup Server on production, they are "remote". If the files will be local to the staging server, there are many options to get them from the production host to the staging host including but not limited to:

- Sharing the directory on the production host over NFS to the staging host.
- Using Network Attached Storage (NAS) to replicate the files.
- Using scp to copy the files from one host to the other.

If the files are only available on the production host, Delphix will login to the ASE staging instance and issue the "LOAD DATABASE" command using the remote server syntax (for example LOAD DATABASE pubs2 FROM "/dumps/pubs2.full.9_5_16" AT Prod_BS).

9.10.4.1 Deploy OVA on VMware

Use the Delphix-supplied OVA file to install the Delphix Engine. The OVA file is configured with many of the minimum system requirements. The underlying storage for the install is assumed to be redundant SAN storage.

1. Download the OVA file from https://download.delphix.com[568]. You will need a support login from your sales team or a welcome letter.
   a. Navigate to the Delphix Product Releases/<Current Version>/Appliance Images page.
   b.
2. Login using the vSphere client to the vSphere server (or vCenter Server) where you want to install the Delphix Engine.
3. In the vSphere Client, click **File**.
4. Select **Deploy OVA Template**.
5. Browse to the OVA file.
6. Click **Next**.
7. Select a **hostname** for the Delphix Engine. This hostname will also be used in configuring the Delphix Engine network.
8. Select the **data center** where the Delphix Engine will be located.
9. Select the **cluster** and the **ESX host**.
10. Select one (1) **data store** for the **Delphix OS**. This datastore can be **thin-provisioned** and must have enough free space to accommodate the 127GB comprising the Delphix operating system.
11. Select four (4) or more **data stores** for Database Storage for the Delphix Engine. The Delphix Engine will stripe all of the Database Storage across these VMDKs, so for optimal I/O performance, each VMDK must be equal in size and be configured **Thick Provisioned - Eager Zeroed**. Additionally, these VMDKs should be distributed as evenly as possible across all four SCSI I/O controllers.
12. Select the **virtual network** you want to use. If using multiple physical NICs for link aggregation, you must use vSphere NIC teaming. Do not add multiple virtual NICs to the Delphix Engine itself. The Delphix Engine should use a single virtual network. For more information, see Optimal Network Architecture for the Delphix Engine (see page 719)
13. Click **Finish**. The installation will begin and the Delphix Engine will be created in the location you specified.
14. Once the installation has completed, power on the Delphix Engine and proceed with the initial system configuration as described in Setting Up Network Access to the Delphix Engine (see page 569)

568 https://download.delphix.com/

> If your source database is 4 TB, you probably need 4 TB of storage for the Delphix Engine. Add at least 4 data disks of similar size for the Delphix VM. For example: for a source database of 4 TB, create 4 VMDKs of 1 TB each.

> For a full list of requirements and best practice recommendations, see Virtual Machine Requirements for VMware Platform (see page 604)

9.10.4.2 Setup network access to Delphix engine

1. Power on the Delphix Engine and open the Console.
2. Wait for the Delphix Management Service and Delphix Boot Service to come online. This might take up to 10 minutes during the first boot. Wait for the large orange box to turn green.
3. Press any key to access the sysadmin console.
4. Enter `sysadmin@SYSTEM` for the username and `sysadmin` for the password.
5. You will be presented with a description of available network settings and instructions for editing.

```
Delphix Engine Network Setup To access the system setup through the browser,
the system must first be configured for networking in your environment. From
here, you can configure the primary interface, DNS, hostname, and default
 route. When DHCP is configured, all other properties are derived from DHCP
settings. To see the current settings, run "get." To change a property, run
"set =." To commit your changes, run "commit." To exit this setup and return to
the standard CLI, run "discard." defaultRoute    IP address of the gateway for
 the default route -- for example, "1.2.3.4." dhcp            Boolean value
indicating whether DHCP should be used for the primary interface. Setting this
 value to "true" will cause all other properties (address, hostname, and DNS)
to be derived from the DHCP response dnsDomain       DNS Domain -- for example,
"delphix.com" dnsServers      DNS server(s) as a list of IP addresses -- for
 example, "1.2.3.4,5.6.7.8." hostname        Canonical system hostname, used in
alert and other logs -- for example, "myserver" primaryAddress  Static address
for the primary interface in CIDR notation -- for example, "1.2.3.4/22" Current
settings: defaultRoute: 192.168.1.1 dhcp: false dnsDomain: example.com
dnsServers: 192.168.1.1 hostname: Delphix primaryAddress: 192.168.1.100/24
```

6. Configure the `hostname`. If you are using DHCP, you can skip this step.

```
delphix network setup update *> set hostname=<hostname>
```

Note: Use the same `hostname` you entered during the server installation.

7. Configure DNS. If you are using DHCP, you can skip this step.

```
delphix network setup update *> set dnsDomain=<domain> delphix network setup
update *> set dnsServers=<server1-ip>[,<server2-ip>,...]
```

8. Configure either a static or DHCP address.

   DHCP Configuration

```
delphix network setup update *> set dhcp=true
```

   Static Configuration

```
delphix network setup update *> set dhcp=false delphix network setup update *>
set primaryAddress=<address>/<prefix-len>
```

   Note: The static IP address must be specified in CIDR notation (for example, `192.168.1.2/24`)

9. Configure a default gateway. If you are using DHCP, you can skip this step.

```
delphix network setup update *> set defaultRoute=<gateway-ip>
```

10. Commit your changes. Note that you can use the `get` command prior to committing to verify your desired configuration.

```
delphix network setup update *> commit Successfully committed network settings.
Further setup can be done through the browser at: http://<address> Type "exit"
to disconnect, or any other commands to continue using the CLI.
```

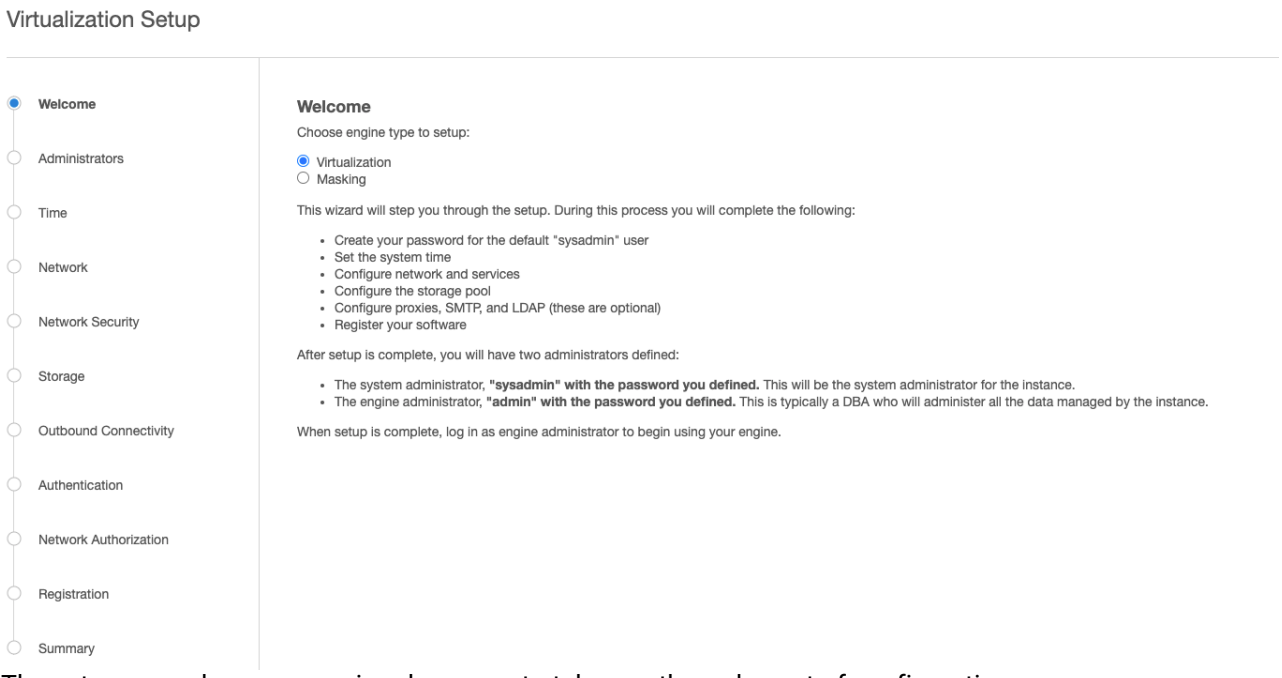
11. Check that you can now access the Delphix Engine through a Web browser by navigating to the displayed IP address, or hostname if using DNS.
12. Exit setup.

```
delphix> exit
```

9.10.4.3 Setting up the Delphix engine

Once you set up the network access for Delphix Engine, navigate to the Delphix Engine URL in your browser for server setup.

The welcome screen below will appear for you to begin your Delphix Engine setup.



The setup procedure uses a wizard process to take you through a set of configuration screens:

- Administrators
- Time
- Network
- Network Security
- Storage
- Outbound Connectivity
- Authentication
- Network Authorization
- Registration
- Summary

1. Connect to the Delphix Engine at `http://<Delphix Engine>/login/index.html#serverSetup` The **Delphix Setup** application will launch when you connect to the server. Enter your **sysadmin** login credentials, which initially defaults to the username **sysadmin**, with the initial default password of **sysadmin**. On first login, you will be prompted to change the initial default password.
2. Click **Next**.

9.10.4.3.1 Administrators

The Delphix Engine supports two types of administrators:

- System Administrator (**sysadmin**) - this is the engine system administrator. The sysadmin password is defined here**.**
- Engine Administrator (**admin)** - this is typically a DBA who will administer all the data managed by the engine.

On the Administrators tab, you set up the sysadmin password by entering an email address and password. The details for the admin are displayed for reference.

> The default domain user created on Delphix Engines from 5.3.1 is known as **admin** instead of delphix_admin. When engines created before 5.3.1 are upgraded to 5.3.1 or later they will retain their old username 'delphix_admin'. To avoid complications Delphix recommends creating users with an admin role and then Disabling delphix_admin.

System Time

The engine time is used as the baseline for setting policies that coordinate between virtual databases and external applications.

Choose your option to set up system time in this section. For a Quick Start, simply set the time and your timezone. You can change this later.

9.10.4.3.2 Network

The initial out-of-the-box network configuration in the OVA file is set to use a single VMXNET3 network adapter.

You have already configured this in the initial configuration. Delphix supports more advanced configurations, but you can enable those later.

9.10.4.3.3 Storage

You should see the data storage VMDKs or RDMs you created during the OVA installation. Click **Next** to configure these for data storage.

9.10.4.3.4 Serviceability

Choose your options to configure serviceability settings.

For a Quick Start, accept the defaults. You can change this later.

9.10.4.3.5 Authentication

Choose your options to configure authentication services.

For a Quick Start, accept the defaults. You can change this later.

9.10.4.3.6 Registration

If the Delphix Engine has access to the external Internet (either directly or through a web proxy), then you can auto-register the Delphix Engine:

1. Enter your **Support Username** and **Support Password**.

2. Click **Register**.

If external connectivity is not immediately available, you must perform manual registration.

1. Copy the **Delphix Engine registration code** in one of two ways:
   a. Manually highlight the registration code and copy it to clipboard. Or,
   b. Click **Copy Registration Code to Clipboard**.
2. Transfer the Delphix Engine's registration code to a workstation with access to the external network Internet. For example, you could e-mail the registration code to an externally accessible e-mail account.
3. On a machine with access to the external Internet, please use your browser to navigate to the Delphix Registration Portal at http://register.delphix.com[569].
4. Login with your Delphix support credentials (username and password).
5. Paste the **Registration Code**.
6. Click **Register**.

> Although your Delphix Engine will work without registration, we strongly recommend that you register each Delphix Engine as part of the setup. Failing to register the Delphix Engine will impact its supportability and security in future versions.

To regenerate the registration code for a Delphix Engine please refer to, Regenerating the Delphix Engine Registration Code (see page 672). Delphix strongly recommends that you regenerate this code and re-register the engine regularly to maximize the Support Security of the Delphix Engine. Delphix recommends doing this every six months.

9.10.4.3.7 Summary

The final summary tab will enable you to review your configurations for System Time, Network, Storage, Serviceability, and Authentication.

1. Click the **Back** button to go back and to change the configuration for any of these server settings.
2. If you are ready to proceed, then click **Submit**.
3. Click **Yes** to confirm that you want to save the configuration.
4. Click **Setup** to acknowledge the successful configuration.
5. There will be a wait of several minutes as the Delphix Engine completes the configuration.

569 http://register.delphix.com/

9.10.4.4 Requirements for SAP ASE hosts and databases

In order to begin using SAP ASE environments with Delphix, you will need to configure the source and target hosts with the requirements described on this page.

9.10.4.5 SAP ASE source host requirements

There must be an operating system user, such as delphix_os, that meets the following requirements:

- The $SYBASE environment variable is defined for non-interactive shells (such as via the .bashrc configuration file).
    - Set the PermitUserEnvironment configuration parameter to "yes" in the sshd_config file
    - Add the variable to the user's .ssh/environment file
    - Restart the SSH daemon
- To test this requirement:

```
ssh delphix_user@ase_hostname env | grep SYBASE
```

- Can login to the source host via SSH (TCP port 22)
- Delphix requires superuser permission to run **pargs** in order to discover Solaris ASE instances. For more information, see Sudo Privilege Requirements for SAP ASE Environments (see page 1897).[570]
- Designating the Delphix operating system user's primary group to be the same as the ASE instance's means the file system permissions can be more restrictive and is a better security practice than granting world read access to the toolkit or the backup files. If the target host is used to host the staging databases, consider the following:
    - If you don't add the Delphix operating system user to the ASE instance owner's group, greater permissions will need to be **granted to the backup files to ensure read access to the dumps and/or transaction logs**. Delphix looks for the backup files on the staging host (unless a "remote" backup server is used in which case, the remote host is used which is often the source environment).
    - If you sync Delphix with a dSource by asking ASE to create a new backup, the ASE instance owner will need the **write permission to the toolkit** (or the mount point if you use the CLI to specify a directory other than the toolkit). Delphix will issue the "DUMP DATABASE" command to write to the staging database's "temp" directory which is mounted on the staging host.
    - Has **write permission for the mount-point directory** (by default the toolkit directory but can be a separate mount point specified in the command line interface).
- There must be a directory on the source host where you can install the Delphix platform toolkit, for example: /var/opt/delphix/Toolkit
    - The delphix_os user must own the directory

570 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/sudo-privilege-requirements-for-sap-ase-environments

- The directory must have permissions 0770, for example, -rwxrwx---. However, you can also use more permissive settings.
- The directory should have 256MB of available storage.

9.10.4.6 Source database requirements

When adding a source ASE environment to Delphix, you may use a single login to discover the ASE instances and link the source databases OR you may use a single login to discover all of the ASE instances and separate logins to link each dSource.

- Delphix uses a single database user for the discovery of all ASE instances and their databases for each environment added to Delphix.
  - The discovery database user (delphix_disc for example) must have SELECT privileges on the following tables for each ASE instance on the source host:
    - **sysdatabases**
    - **sysservers**
    - **syslisteners**
    - **sysconfigures**
    - **syscurconfigs**
- Another user must be specified when linking each dSource (delphix_link for example) that has SELECT privileges on the above tables.
  - If you will select New Full Backup when linking, this user must also have privileges to take a new full database dump of the source database. For more information about linking options, see Linking an SAP ASE Data Source (see page 1917)
  - The link database user can be different for each instance and database on the source host.

> If the source database is resized and trunc log on chkpt is disabled, take a transaction log dump immediately after the resize operation completes. If trunc log on chkpt is enabled, take a full database dump immediately after the resize operation completes. If multiple resizing operations are performed without taking transaction log dumps between each operation it may be necessary to manually sync the dSource with a new full database dump for Delphix to be able to continue ingesting source database dumps.

9.10.4.7 Target host requirements

- The operating system on the target environment must be the same as, or binary compatible with, the operating system on the source environment.
- As the Delphix Engine supports both NFSv3 and NFSv4 for mounting target host filesystems, the prerequisite packages that support NFSv3 or NFSv4 client communication are required for normal

operation, and the required services to support NFS client communications (including file locking) must be running.This includes:

a. portmapper / rpcbind
b. status daemon (rpc.statd)
c. NFS lock manager (rpc.lockd/lockmgr)

- The SAP ASE major version on the target environment must be the same as the version on the source environment. However, EBF/SP version on target environment can be different. If the target is used as a staging server, the ASE major version must be the same. The only caveat is for ASE 15.7 - where the source and staging patch levels must both be either any version below SP64 or both be any version above SP100.
- There must be an operating system user, such as delphix_os, that meets the following requirements:
  a. Set the PermitUserEnvironment configuration parameter to "yes" in the sshd_config file
  b. Add the variable to the user's .ssh/environment file
  c. Restart the SSH daemon
  d. The $SYBASE environment variable is set for non-interactive shells (such as via the .bashrc configuration file). Set the variable as follows:
  e. To test this requirement:

```
ssh delphix_user@ase_hostname env | grep SYBASE
```

- Can login to the target host via Secure Shell (SSH)
- Can login to ASE instances using isql with LANG=C set
- Designating the Delphix operating system user's primary group to be the same as the ASE instance's means the file system permissions can be more restrictive and is a better security practice than granting world read access to the toolkit or the backup files. If the target host is used to host the staging databases, consider the following:
  - If you don't add the Delphix operating system user to the ASE instance owner's group, greater permissions will need to be **granted to the backup files to ensure read access to the dumps and/or transaction logs**. Delphix looks for the backup files on the staging host (unless a "remote" backup server is used in which case, the remote host is used which is often the source environment).
  - If you sync Delphix with a dSource by asking ASE to create a new backup, the ASE instance owner will need **write permission to the toolkit** (or the mount point if you use the CLI to specify a directory other than the toolkit). Delphix will issue the "DUMP DATABASE" command to write to staging database's "temp" directory which is mounted on the staging host.
  - Has **write permission for the mount-point directory** (by default the toolkit directory but can be a separate mount point specified in the command line interface).
- The following permissions are usually granted via sudo authorization of the commands. See Sudo Privilege Requirements for SAP ASE Environments (see page 1897) for further explanation of this requirement, and Sudo File Configuration Examples for SAP ASE Environments (see page 1900) for example file configurations.
  a. Permission to run **mount** and **umount**as super-user.
  b. On Solaris, permission to run **pargs**on Solaris
  c. On AIX, permission to run the **nfso**command as super-user.

d. (Optional) On AIX and Linux, permission to run **ps**as super-user.
e. Disable **tty** for the delphix_os user for mount and umount.

- There must be a directory on the source host where you can install the Delphix platform toolkit, for example: /var/opt/delphix/Toolkit
  - The delphix_os user must own the directory
  - The directory must have permissions 0770, for example, -rwxrwx--. However, you can also use more permissive settings.
  - The directory should have 1GB of available storage
  - Avoid using the home directory of the delphix_os user
  - If you intend to use the LogSync feature, it is recommended to make the toolkit directory as short as possible to keep the full path to the transaction log file names under ASE's 127 character limit. For example, create the toolkit directory as /tk. Alternatively, link the dSource using the command line interface and specify the "mountBase (see page 2616)" parameter to mount the staging database's devices under a directory following your own naming convention.

9.10.4.8 Target database requirements

- There must be a database user, such as delphix_db, with the **sa_role** on each instance on the target environment
- The database user such as delphix_db for any staging instances must also have the **sybase_ts_role**
- If the target host will be used as a staging target environment, at least one of the following two options must be configured:
  - You must use **sp_addserver** to add the staging ASE instance's Backup Server to **sysservers** on the source ASE instance (so that remote database dump/load works)

    OR
  - Full and transaction dump files from the source database must be available locally to the staging database (over NFS, replication, scp, etc.)

9.10.4.8.1 Specific ASE tuning recommendations

- In ASE 15.7 SP60 and higher, there is a configuration parameter named "**enable large pool for load**". ASE automatically tunes the caches for recovery on reboot, but does not do this for load database and load transaction recovery, since it could impact other databases on a production server. Delphix recommends that the staging ASE instance be separate from the ASE production instance and ASE instances hosting VDBs so enabling this parameter should have ***a beneficial impact on performance for the ASE staging instance.***
- Specify the "**relaxed**" strategy for replacing the cache in the default data cache (**sp_cacheconfig 'default data cache', relaxed**) in the ASE staging instance. Since the mount/unmount process invalidates the pages anyway, the page chain is really just unneeded overhead on the staging server.
- Staging and target ASE instances should have disk mirroring disabled.

sp_configure "disable disk mirroring" – run value should be 1, which is the default. If it is 0, change it using

sp_configure "disable disk mirroring", 1 – this parameter is static so the ASE instance must be restarted for this change to take effect.

- Delphix will mirror the number of devices used on the source database for the staging database (dSource) and each VDB created from that source database. The number of devices parameter should be scaled appropriately based on the max number of virtual databases that will be provisioned to the ASE instance. This parameter can be changed using: **sp_configure "number of devices",** .

> ASE 15.7.0 SP100 and later releases support the shrink command. In some cases, Delphix must increase the number of devices used for databases if this command is used. Delphix creates a minimum of the same number of devices as the source database for the staging database (dSource) and each VDB and will add more devices for every 4TB of fragment holes. See SAP ASE issue CR#799273[571] for additional details.

To support multiple VDBs and the staging databases, you may need to increase the parameter number **of alarms**.

> Delphix uses ASE operations that use alarm structures such as **MOUNT** and **UNMOUNT**. The number of alarms limits the number of these operations which can be run concurrently. Various ASE instance failures can occur if the available alarm structures are exhausted. The amount of memory consumed by increasing the number of alarm structures is small. Delphix recommends that the **number of alarms** value is increased to at least 4096.

9.10.4.9 Adding SAP ASE source and target environments

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Environments**.
4. From the **Actions (...)** menu select **Add Environment**.
5. In the **Host and Server** tab window, select **Unix/Linux**.
6. Select **Standalone Host.**
7. Click **Next.**
8. Enter a **Name** for the environment.
9. Enter the **Host IP** address.

571 https://launchpad.support.sap.com/#/notes/2332779

10. For NFS Addresses (Optional): Enter one or more comma-separated **IP Address/Hostname**
    Note: If specified, Delphix Engine only allows NFS requests (mount, etc) originated from IP Addresses specified for the host.
11. Enter the **SSH** port. The default value is **22**.
12. Select a **Login Type**. — Username and Password - enter the OS username and password — Username and Public Key - enter the OS username. — Password Vault - select from an existing Enterprise Password Vault

    Using public key authentication

    If you want to use public-key authentication for logging into your Unix-based environment, there are two options: use the engine's key pair or provide a key pair for this environment.

    To use the engine's key pair:
    a. Select **Public Key** for the **Login Type.**
    b. Click **View Public Key**.
    c. Copy the public key that is displayed, and append it to the end of your `~/.ssh/authorized_keys` file. If this file does not exist, you will need to create it.
        i. Run chmod 600 ~/.ssh/authorized_keys to allow only the file's owner to read and write to it (make sure the file is owned by the user).
        ii. Run chmod 755 ~ to restrict access to the user's home directory so no other user may write to it.
        iii. Run chmod 700 ~/.ssh so that others cannot write to it. The ~/.ssh directory cannot be writable by group or other users. Otherwise, authentication will fail.

    As an alternative, you can provide a key pair specific for this environment via the API or CLI. See Option 2 in this CLI Cookbook article (see page 1855) for instructions
13. For **Password Login**, click **Verify Credentials** to test the username and password.
14. Enter a **Toolkit Path**. The toolkit directory stores scripts used for Delphix Engine operations. It must have a persistent working directory rather than a temporary one. The toolkit directory will have a separate subdirectory for each database instance. The toolkit path must have 0770 permissions.
15. To provide your own Oracle Java select the **Provide my own JDK** checkbox and click **Next**.
16. In the Java Development Kit tab enter the absolute path to your Oracle JDK and click **Next**.
17. Click the **Discover SAP ASE** checkbox.
18. Click **Next**.
19. In the Summary, tab confirm your selections.
20. Click **Submit**.

9.10.4.10 ASE manual discovery

When an environment is added, Delphix discovers your ASE instances. Manual discovery allows users to add instances that were not automatically discovered. This feature is currently only supported via the CLI.

This topic describes how to use CLI commands to manually add ASE repositories to an SAP ASE environment. Discovery is the process by which the Delphix platform identifies data sources and data dependencies on a remote environment. ASE repository discovery is done automatically when an environment is added to the Delphix platform or when an already added environment is refreshed. In some cases, automatic discovery does not discover all of the repositories in an SAP ASE environment. These repositories may be added using manual discovery.

Unlike automatically discovered instances, manually discovered instances are not automatically deleted if the environment is refreshed when the instance isn't running. Manually discovered instances are not updated during an environment refresh either. So for example, if you upgrade ASE to a new version or change the listener port, you must manually update the repository.

To manually discover an ASE repository you will need to:

- Add the environment to Delphix
- Use CLI to manually discover a repository

9.10.4.11 Creating an ASE environment

Please refer to Adding an SAP ASE Environment (see page 1905) for detailed steps.

9.10.4.12 Manually discovering a repository

> In the following example, we are using **sc-dev3.dc2** as our example environment.

1. Log into CLI and cd to repository menu:

```
$ ssh delphix_admin@sc-dev3.dc2
Password:
sc-dev3.dc2> cd repository
sc-dev3.dc2 repository>
```

2. Add (manually discover) an ASE repository instance:

Note: The values used in the following code block are specific to the example instance we are adding.

```
sc-dev3.dc2 repository> create
sc-dev3.dc2 repository create *> ls
Properties
  type: ASEInstance
  credentials: (unset)
  dbUser: (unset)
  environment: (required)
  installationPath: (required)
  instanceName: (required)
```

```
  instanceOwner: (required)
  ports: (required)
  version: (unset)
sc-dev3.dc2 repository create *> set credentials.password=sybase
sc-dev3.dc2 repository create *> set dbUser=sa
sc-dev3.dc2 repository create *> set environment=sc-rhel64-sybase-ase-0
sc-dev3.dc2 repository create *> set installationPath=/opt/sybase/15-7
sc-dev3.dc2 repository create *> set instanceName=ASE1570_S1
sc-dev3.dc2 repository create *> set instanceOwner=sybase
sc-dev3.dc2 repository create *> set ports=5100
sc-dev3.dc2 repository create *> ls
Properties
  type: ASEInstance
  credentials:
      type: PasswordCredential (*)
      password: ******** (*)
  dbUser: sa (*)
  environment: sc-rhel64-sybase-ase-0 (*)
  installationPath: /opt/sybase/15-7 (*)
  instanceName: ASE1570_S1 (*)
  instanceOwner: sybase (*)
  ports: 5100 (*)
  version: (unset)
sc-dev3.dc2 repository create *> commit
  `ASE_INSTANCE-22
sc-dev3.dc2 repository>
```

9.10.4.13 Updating a repository

Adding onto the above, the following example illustrates updating an ASE instance's version after upgrading ASE:

```
sc-dev3.dc2> repository
sc-dev3.dc2 repository> select ASE1570_S1
sc-dev3.dc2 repository 'ASE1570_S1'> update
sc-dev3.dc2 repository 'ASE1570_S1' update *> set version="15.7 SP138"
sc-dev3.dc2 repository 'ASE1570_S1' update *> ls
Properties
   type: ASEInstance
   credentials:
       type: PasswordCredential
       password: ********
   dbUser: sa
   installationPath: /opt/sybase/15-7
   instanceOwner: sybase
   linkingEnabled: true
   ports: 5100
   provisioningEnabled: true
   servicePrincipalName: (unset)
   staging: false
```

```
   version: 15.7 SP138 (*)
sc-dev3.dc2 repository 'ASE1570_S1' update *> commit
```

Be careful when setting the version string. Make sure that it matches output as displayed by the "select @@version" query all the way out to the patch level (PL). For example "15.7 SP138" or "16.0 SP02 PL01".

9.10.4.14 Enable linking and provisioning for SAP ASE environments

This topic describes how to enable and disable provisioning and linking for SAP ASE databases.

Before a database can be used as a dSource, you must first make sure that you have enabled linking to its SAP ASE instance. Similarly, before you can provision a VDB to a target database, you must make sure that you have enabled provisioning to its SAP ASE instance.

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Environments**.
4. Select the **Databases** tab.
5. Click the **pencil** icon located next to the database **Installation Details**.
6. Select the **Allow Provisioning** checkbox to enable provisioning, deselect the checkbox to disable provisioning.
7. Click **show details** for the database.
8. Slide the button next to **Allow Linking** to **On** or **Off** to enable or disable linking.

9.10.4.15 Linking an SAP ASE data source

The dSource is an object that the Delphix Virtualization Engine uses to create and update virtual copies of your database. As a virtualized representation of your source data, it cannot be managed, manipulated, or examined by database tools. For an overview of all dSource related actions, please visit Managing Data Sources.[572] Delphix Virtualization for SAP ASE databases leverages backup-based ingestion, which means that Delphix will look for, or sometimes initiate the creation of, a backup through your SAP ASE backup server. From there, the backup is restored on a staging server and the staging copy is then ingested into Delphix. See Delphix Architecture with SAP ASE (see page 1855) for more information.

When linking a dSource from an SAP ASE source database, Delphix offers several different methods of capturing backup information:

- ASE Managed Backups, where the SAP ASE source database schedules and initiates backups. This method supports various backup types which include:

572 https://documentation.delphix.com/continuous-data-11-0-0-0/docs/getting-started-managing-data-sources-and-syncing-data

    - Full backups
    - Transaction log backups (with LogSync disabled)
    - Transaction log backups (with LogSync enabled)
- Delphix Managed Backups, where the Delphix Engine schedules and initiates the backups from the source database, and captures them.

9.10.4.16 ASE managed backups

Further contextual information on the various backup types (listed above):

- Full Backups - A snapshot will be created on the Delphix Timeflow for each Full backup.
- Transaction log backups (with LogSync disabled) - A snapshot will be created on the Delphix Timeflow for each transaction log backup.

> Transaction Logs
> Transaction logs are not collected if:
> - a) there are gaps in the sequence of log backups (a break in the “log chain”).
> - b) the available log backups do not include any changes since the last successful Delphix snapshot.

Transaction log backups (with LogSync enabled) - A snapshot will be created on the Delphix Timeflow for each transaction log backup. In addition, point-in-time provisioning will be an available option if you would like to provision from any point in between snapshots.

> Log Files
> Log files consume additional space on the Delphix Engine and are managed according to the defined retention policy for logs.

9.10.4.17 Delphix managed backups

When the checkbox for Delphix Managed Backups is selected, the Delphix Engine will initiate a full backup of the source database for the initial load of the dSource. Thereafter, the Delphix Engine will initiate full backups of the source database using the schedule specified by the selected SnapSync Policy. If you select the None policy, the Delphix Engine will not automatically initiate a full backup, but you can initiate them manually using the snapshot (camera) icon.

Delphix looks for the backup files on the staging host (unless a "remote" backup server is used in which case, the remote host is used which is often the source environment)

9.10.4.18 Procedure

1. Login to the **Delphix Management** application.
2. Navigate to **Manage > Datasets.**
3. Click the plus icon and select **Add dSource**.
   Note: Delphix looks for the backup files on the staging host (unless a "remote" backup server is used in which case, the remote host is used which is often the source environment.
4. In the **Add dSource** wizard, select the source environment with the correct environment-based user.
5. Enter your login credentials for the source database and click **Next**.
6. Enter a name and select a group for your dSource. Adding a dSource to a dataset group lets you set Delphix Domain user permissions for that database and its objects, such as snapshots. See the topics under Users and Groups (see page 674) for more information.
7. Select the **Data Management** settings needed, as described in Data Management Settings for ASE Data Sources (see page 1921). [573]
8. From the **Data Management** tab under the **Initial Load** option, select and enter any **additional settings** needed. There are three options for the initial load of the dSource:
   a. If the source ASE instance resides on the same server as the staging ASE instance, the staging database's NFS mounted "temp" directory will be present for the source database to write to in response to the "DUMP DATABASE" command.
   b. If the source and staging ASE instances are configured to allow remote access to the backup servers and the dSource is linked using the Remote Server option as described below. Delphix will then issue the "DUMP DATABASE" command and append the "AT <staging_backup_server_name>" clause so that the dump is written to the staging backup server.
   c. (Recommended) **New Full Backup** - Lets Delphix create a new full backup file and load it. Note - that when Delphix creates the backup, it is moved to Delphix's NFS-mounted storage located on the stage host rather than. The backup will be located in the "temp" directory and will be deleted once the Delphix Engine has restored the backup and created a dSource from the restored staging database. This means that this option will work under two scenarios:
   d. Most Recent **Existing Full Backu**p – Find the most recent existing full backup file in the Backup Location and load it.
      Note: If Dump History is not active on the Source Database: Choosing this option can delay completion of the dSource link as Delphix attempts to find and catalog every single backup listed in the source database's backup server log file.
   e. Specific **Existing Full Backu**p – Specify which backup files in the Backup Location you want to load. Choosing this option is much faster because Delphix will skip directly to loading the desired backup and only start to search for and catalog backups in the background after the linking of the source database has completed.
      Note: When using a dump taken with the deprecated compression syntax, select the Specific

573 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/data-management-settings-for-sap-ase-dsources

Existing Full Backup option for Initial Load and, for each stripe, type compress::<file name> into the text box.

9. Select the Staging environment and ASE instance that will be used to manage the staging database used for validated sync of the dSource.
10. Select any policies for the new dSource.
11. Click **Next**, then specify any pre-hook and post-hook scripts.
12. Review the dSource Configuration and Data Management information, and click **Submit** to begin provisioning the VDB.

9.10.4.19 Provisioning an SAP ASE VDB

9.10.4.20 Procedure

1. In the Datasets panel on the left-hand side, click the group containing the dSource or VDB from which you want to provision.
2. Click the **Timeflow** tab.
3. Select a snapshot or open LogSync timeline to provision by a specific log or point in time.
    - Find more detail about initial provisioning options in the section 'Provisioning by Snapshot or LogSync' below.
4. Click to open the Provision VDB wizard, and select a compatible Target Environment for the new ASE VDB
5. Review the information presented for Target Configuration and edit as necessary.
6. Select a Snapshot Policy for the VDB.
7. (Optional) - Selective Data Distribution - After policies, there is a masking option.
8. Enter any operations that should be run in the Hooks page. These scripts can be managed after provision in the VDB's configuration page.

When provisioning starts, you can review the progress of the job by selecting the VDB and clicking on the Status tab, or by selecting Manage/Dashboards and viewing the Job History panel. Alternatively, you could see this in the Actions Sidebar. When provisioning is complete, the VDB will be included in the group you designated and listed in the Datasets panel. If you select the VDB in the Datasets panel and click the Configuration tab, you can view information about the database and its Data Management settings.

9.10.4.20.1 Provisioning by snapshot or LogSync

When provisioning by Snapshot, you can provision to the start of any particular snapshot by time.

Provisioning by snapshot/time	description
Provisioning By Snapshot	You can provision by using a Snapshot. In that case, a new VDB will be provisioned to the database state as of the Snapshot.
Provision by Time	If you have enabled Log Sync, you can provision a new database to a point in time. You can select a snapshot and then using time entry fields, specify a Point in Time. Delphix will use the selected snapshot to restore the VDB and use the log files to roll forward the VDB to the selected time.

9.10.4.21 Configuration settings for ASE virtual databases

Each VDB has its own data management settings, found during the provisioning workflow as well as in the configuration page for that VDB. When you create a SAP ASE VDB, Delphix copies most configuration settings from the dSource and uses them to create the VDB. However, you can customize these with the following settings:

Setting	Explanation
Recovery Model	The current recovery model of the source database. This field will auto-populate with information from the dSource.
Auto VDB Restart	Enabling this option will automatically restart this VDB whenever its target host is rebooted.

9.10.4.22 Automatic VDB restart on target server after reboot

The Delphix platform now automatically detects whether a target server has been rebooted, and proactively restarts any VBD on that server that was previously up and running. This is independent of the data platform. It is done as if you realized a target server was restarted and issued a start command from the Delphix platform. This feature is compatible with Self-Service ordering dependencies and is limited to non-clustered VDBs.

To enable automatic restart, complete the following steps:

- When provisioning a new VDB in the VDB Provisioning wizard, check the **Auto VDB Restart** box.

Once the VDB has been provisioned, you will be able to turn **Automatic VDB Restart** on.

1. In the **Datasets** panel, select the VDB.
2. Select the **Configuration** tab.
3. Select **Source** sub-tab.

4. Select **Database edit**.

9.10.4.23 Next steps

Congratulations! You have provisioned your first virtual database!

Now, perform some simple functional tests with your application. You can connect your app to the VDB using standard TNS/JDBC techniques. Delphix has already registered the VDB for you on the target listener.

We suggest the following next steps:

1. Drop a table and use the VDB Rewind feature to test the recovery of your VDB.
2. Take a snapshot of your dSource and refresh your VDB to quickly get fresh production data.
3. Provision a new VDB from your VDB to test sharing data quickly with other teams.
4. Mask your new VDB to protect sensitive data. Provision new VDBs from that masked VDB to quickly provide safe data to development and QA teams.

9.10.5 SAP ASE support and requirements

In order to begin using SAP ASE environments with Delphix, you will need to configure the source and target hosts with the requirements described in this section.

This section covers the following topics:

- SAP ASE matrix (see page 1877)
- Requirements for SAP ASE environments and databases (see page 1884)
- Network and connectivity requirements for SAP ASE environments (see page 1892)
- Sudo privilege requirements for SAP ASE environments (see page 1897)
- Sudo file configuration examples for SAP ASE environments (see page 1900)

9.10.5.1 SAP ASE matrix

This topic describes supported operating systems and database versions for SAP ASE.

9.10.5.1.1 Supported DBMS versions

- ASE 15.5
- ASE 15.7
- ASE 16.0

9.10.5.1.2 Supported operating systems and database versions for SAP ASE

Source and Target OS and DBMS compatibility

The source and target must be running the same DBMS/Operating System combination, (*although users can run different patch/sp levels*) in order to successfully provision a VDB to the target. For example, if the source is running SAP ASE 16, the target can be running ASE 16SP1. However, for ASE 15.7 the versions of the source and target should both be in the 15.7 SP100 to 15.7 SP140 or 15.7 ESD#1 to 15.7 SP64 ranges. There can be no mixing of versions between these two ranges. If the target is used as a staging server, the same rule applies (prior to Delphix 5.2.5.0, the staging server had to match the source server's version down to the SP level). The Operating System platform must be the same between the source and target, even when the operating system version may differ. For example, if the source is running Red Hat Enterprise Linux 6.2 x86_64 then the target could be running Red Hat Enterprise Linux 6.4 x86_64, but not Solaris 10 SPARC.
ASE itself may or may not allow the **MOUNT** command to work between versions. Be sure to check that the **UNMOUNT** and **MOUNT** commands work between your desired ASE versions.
So for example, if we upgraded the ASE staging instance to ASE 15.7 SP140 but we keep the ASE instance hosting the VDBs at ASE 15.7 SP135, we would want to make sure that we can **UNMOUNT** a database from ASE 15.7 SP140 and then **MOUNT** it on ASE 15.7 SP135:
isql -Usa -Psybase -SASE157SP140 1> UNMOUNT DATABASE testdb to "/tmp/manifest" 2> GO isql -Usa -Psybase -S ASE157SP135 1> MOUNT DATABASE testdb from "/tmp/manifest" 2> go Msg 14580, Level 16, State 1: Server 'ASE157SP135', Line 1: You cannot mount the database(s) because at least one database contains functionality that is available only on the server on which it originated.

In the above example, ASE would not allow the database "testdb" to be mounted on the older release of ASE.

Required HP-UX patch for target servers
PHNE_37851 - resolves a known bug in HP-UX NFS client prior to HP-UX 11.31.

- Delphix Support Policies specifically list Major and Minor release coverage. If a minor release is listed as covered, then all patch releases under that minor release are certified.
- Delphix only supports 64-bit operating systems.

Key:

Color	Supported?
Y	Yes
N	No

Color	Supported?
NA	Not Applicable

9.10.5.1.2.1 Red Hat Enterprise Linux (RHEL)

> Support for RHEL 8.x requires installing the `libncurses.5` library onto the host. Please reference KBA 5622[574] for actionable steps.

Supported OS version	Supported DBMS version		
	15.5	15.7	16
RHEL 5.5	Y	Y	**NA**
RHEL 5.6	Y	Y	NA
RHEL 5.7	Y	Y	NA
RHEL 5.8	Y	Y	NA
RHEL 5.9	Y	Y	NA
RHEL 5.10	Y	Y	NA
RHEL 5.11	Y	Y	NA
RHEL 6.0	N	N	NA
RHEL 6.1	N	N	N
RHEL 6.2	Y	Y	N

574 https://support.delphix.com/Delphix_Virtualization_Engine/Delphix_Admin/Environment_Operations_Fail_With_%22bash%3A_error_while_loading_shared_libraries%3A_libncurses.so.5%3A_cannot_open_shared_object_file%3A_No_such_file_or_directory%22_(KBA5622)

RHEL 6.3	Y	Y	N
RHEL 6.4	Y	Y	N
RHEL 6.5	Y	Y	Y
RHEL 6.6	Y	Y	Y
RHEL 6.7	Y	Y	Y
RHEL 6.8	Y	Y	Y
RHEL 6.9	Y	Y	Y
RHEL 6.10	Y	Y	Y
RHEL 7.0	NA	Y	Y
RHEL 7.1	NA	Y	Y
RHEL 7.2	NA	Y	Y
RHEL 7.3	NA	Y	Y
RHEL 7.4	NA	Y	Y
RHEL 7.5	NA	Y	Y
RHEL 7.6	NA	Y	Y
RHEL 7.7	NA	Y	Y
RHEL 7.8	NA	Y 6.0.3+	Y 6.0.3+
RHEL 7.9	NA	Y 6.0.4+	Y 6.0.4+
RHEL 8.0	NA	NA	Y

RHEL 8.1	NA	NA	Y 6.0.4+
RHEL 8.2	NA	NA	Y 6.0.4+
RHEL 8.3	NA	NA	Y 6.0.7+

9.10.5.1.2.2 SUSE Linux Enterprise Server (SLES)

> Support for SLES 15 requires installing the `libncurses.5` library onto the host. Please reference KBA 5622[575] for actionable steps.

Supported OS version	**Supported DBMS version**		
	15.5	**15.7**	**16**
SLES 11	N	N	N
SLES 11 SP1	N	N	N
SLES 11 SP 2	N	N	N
SLES 11 SP 3	N	N	N
SLES 11 SP 4	N	N	N
SLES 12	N	N	N
SLES 12 SP 1	N	N	N
SLES 12 SP 2	N	N	N
SLES 12 SP 3	N	N	N

575 https://support.delphix.com/Delphix_Virtualization_Engine/Delphix_Admin/Environment_Operations_Fail_With_%22bash%3A_error_while_loading_shared_libraries%3A_libncurses.so.5%3A_cannot_open_shared_object_file%3A_No_such_file_or_directory%22_(KBA5622)

Supported OS version	Supported DBMS version		
	15.5	15.7	16
SLES 12 SP 4	N	N	Y 6.0.4+

9.10.5.1.2.3 Solaris SPARC

Supported OS Version	Supported DBMS version		
	15.5	15.7	16
Solaris 10 U9	Y	Y	N
Solaris 10 U10	Y	Y	N
Solaris 10 U11	Y	Y	N
Solaris 11	N	N	N
Solaris 11 U1	N	N	N
Solaris 11 U2	N	N	N
Solaris 11 U3	N	Y	N
Solaris 11 U4	N	Y 6.0.1.0+	Y 6.0.1.0+

9.10.5.1.2.4 Solaris x86

Supported OS version	Supported DBMS version		
	15.5	15.7	16
Solaris 10 U9[1]	Y	Y	N

Solaris 10 U10[1]	Y	Y	N
Solaris 10 U11	NA	Y	N
Solaris 11	NA	Y	Y
Solaris 11 U1	NA	Y	Y
Solaris 11 U2	NA	Y	Y
Solaris 11 U3	NA	Y	Y
Solaris 11 U4	NA	N	N

9.10.5.1.2.5 Hewlett Packard Unix (HP-UX)

	Supported DBMS version		
Supported OS version	**15.5**	**15.7**	**16**
HP-UX 11.31	N	N	N

9.10.5.1.2.6 Advanced Interactive executive (AIX)

	Supported DBMS version		
Supported OS version	**15.5**	**15.7**	**16**
AIX 6.1	N	Y	N
AIX 7.1	N	Y	Y 6.0.1.0+
AIX 7.2	N	Y	Y

9.10.5.2 Requirements for SAP ASE environments and databases

In order to begin using SAP ASE environments with Delphix, you will need to configure the source and target hosts with the requirements described on this page.

9.10.5.2.1 SAP ASE host requirements

> When working with ASE dSources, it is necessary that the Staging and Target ASE instances have an identical **Page Size** configuration to the dSource (e.g. 2KB / 4KB / 8KB). This is set during ASE instance installation.
> The Delphix Engine will prevent the use of incompatible instances.

These requirements are applicable to source, staging, and target environments. Target environments have additional requirements, which are detailed in the **Target Host Requirements** section below.

On each host with SAP ASE, there must be an operating system user (e.g. delphix_os) configured to the required specifications for Delphix.

Host requirement	Explanation
Profile and privileges should be the same as the ASE user (e.g. Sybase) on the host.	For example, delphix_os should have the same environment variable (e.g. $PATH), umask, and ulimit settings as the user sybase.
Delphix operating system user's primary group should be the same as SAP ASE instance's group.	To perform Sync/Validated Sync operations on the source database, Delphix reads a number of files like SAP ASE Backup server log file, ASE Dump History file, and backup files. If Delphix OS user does not have the same group as of the SAP ASE instance, these files will need "everyone" read permissions, which is less restrictive.
Delphix Engine Toolkit Directory on Source Host (e.g. /var/opt/delphix/toolkit) • The delphix_os user must own the directory • The directory must have permissions 0770, for example, -rwxrwx---. However, you can also use more permissive settings. • The directory should have 256MB of available storage	• If you sync Delphix with a dSource by asking SAP ASE to create a new backup, the SAP ASE instance owner will need write permissions to the toolkit (or the mount point if you use the CLI to specify a directory other than the toolkit). Delphix will issue the "DUMP DATABASE" command to write to the staging database's "temp" directory which is mounted on the staging host. • Has write permission for the mount-point directory (by default the toolkit directory but can be a separate mount point specified in the command line interface)

Host requirement	Explanation
Backup Server Log File Permissions	• In the case that the Delphix OS user is not in the same group as the SAP ASE instance user, you need to change permissions to SAP ASE Backup server log file from “rw-r---” to “rw-r-r--”. • The “rw-r---” to “rw-r-r--” permissions change is also required if Dump History files are in use.
The Delphix Engine must be able to make an SSH connection to the source host (typically port 22).	

9.10.5.2.2 SAP ASE database requirements

Delphix for SAP ASE requires a specific login to run discovery and linking activities for instances and databases. Delphix requires specific user permissions for database discovery and linking (detailed below).

> ⚠ Discovery and Linking can be separated into separate users if desired.

> ⚠ If you are using an ASE backup server instance shared with multiple ASE data server instances, the use of the dump history file feature is mandatory. This should be enabled on all instances and the dump history file in a location accessible by the Delphix OS user account from the Staging host.

9.10.5.2.3 SAP ASE source database requirements

The discovery database user (delphix_disc for example) must have SELECT privileges on the following tables for each SAP ASE instance on the source host:

- Sysdatabases
- Sysservers
- Syslisteners
- Sysconfigures
- Syscurconfigs

9.10.5.2.4 Database linking

Permissions for linking environments can be included under the “discovery” user or can be a separate, dedicated role - e.g. delphix_link. If separate, the “linking” user must be specified when linking each dSource (delphix_link for example) that has SELECT privileges on the above tables.

If you will select **New Full Backup** when linking, this user must also have privileges to take a new full database dump of the source database. For more information about linking options, see Linking an SAP ASE Data Source (see page 1917)

> The link database user can be different for each instance and database on the source host.

Sample Script to create delphix_link on Linux

```
note: run as sa

sp_addlogin delphix_link, "StrongPassword"

go

sp_adduser delphix_link

go

grant select on sysdatabases to delphix_link

go

grant select on sysservers to delphix_link

go

grant select on syslisteners to delphix_link

go
```

9.10.5.2.5 Requirements when resizing source databases

For SAP ASE on Delphix, transaction logs are helpful for maintaining a Timeflow via logsync. However, not all customers will want to maintain these logs for bandwidth purposes. The following section will help guide users on requirements to maintain timeflows when it comes to resizing databases with and without (`trunc log on chkpt`).

Active SAP ASE Feature on Source Host	Requirement
(`trunc log on chkpt`) is disabled	take a transaction log dump immediately after the resize operation completes.
(`trunc log on chkpt`) is enabled	take a full database dump immediately after the resize operation completes.

If multiple resizing operations are performed without taking transaction log dumps between each operation it may be necessary to manually sync the source database dumps.

9.10.5.2.6 Additional target host requirements

This section describes the user privileges, and environment discovery requirements, that are required for SAP ASE target hosts and databases, collectively referred to as target environments.

Target host requirement	Explanation
The operating system on the target environment must be the same as, or binary compatible with, the operating system on the source environment	
As the Delphix Engine uses NFSv3 for mounting target host filesystems, the prerequisite packages to support NFSv3 client communication are required for normal operation, and the required services to support NFS client communications (including file locking) must be running This includes: • portmapper / rpcbind • status daemon (rpc.statd) • NFS lock manager (rpc.lockd/lockmgr)	The Delphix Engine uses NFSv3 as a defacto mounting standard for target host file systems.

Target host requirement	Explanation

SAP ASE Source/Staging/Target Environment Version Compatibility	• If the Staging environment is different than the Target environment then:
	  • The major version of the Staging ASE must be the same as the version of the source environment. However, EBF/SP version on the Staging environment can be different.
	  • The major/minor/macro version of the Target ASE instance should be the same or higher than the version of the Source ASE instance. Delphix supports the ASE databases to be restored from lower to higher version or within different patch/sp levels as long as SAP ASE supports it.
	• If the Target environment is used as staging then the SAP ASE version on the Target environment must be the same as the version on the Source environment. However, EBF/SP version on the Target environment can be different.

Create OS User (e.g. delphix_os)	There must be an operating system user, such as delphix_os, that meets the following requirements:
OS User must have - The $SYBASE environment variable set for non-interactive shells (such as via the .bashrc configuration file)	Additional Settings for the $SYBASE environment variable: • Set the PermitUserEnvironment configuration parameter to "yes" in the sshd_config file • Add the variable to the user's .ssh/environment file • Restart the SSH daemon
OS User must have appropriate access to Target Hosts/Instances	• Can login to the target host via Secure Shell (SSH) • Can login to ASE instances using isql with LANG=C set
OS User’s Primary Group must be the same as ASE Instance’s Primary Group	• If not, the file system permissions can be more restrictive. • This also is a better security practice than granting world read access to the toolkit or the backup files.

Target host requirement	Explanation
Permissions granted via sudo authorization of the commands: • Disable tty for the delphix_os user for mount and umount • Permission to run mount and umount as super-user. • On Solaris, permission to run pargs on Solaris • On AIX, permission to run the nfso command as super-user. • (Optional) On AIX and Linux, permission to run ps as super-user.	See Sudo Privilege Requirements for SAP ASE Environments (see page 1897) for further explanation of this requirement, and Sudo File Configuration Examples for SAP ASE Environments (see page 1900) for example file configurations.
There must be a database user, such as delphix_db, with the sa_role on each instance on the target environment	
The database user such as delphix_db for any staging instances must also have the sybase_ts_role	
There must be a directory on the target environment where you can install the Delphix Engine toolkit, for example,/var/opt/delphix/Toolkit.	• The delphix_os user must own the directory • The directory must have permissions 0770, for example, -rwxrwx--. However, you can also use more permissive settings. • The directory should have 1GB of available storage • Avoid using the home directory of the delphix_os user • If you intend to use the LogSync feature, it is recommended to make the toolkit directory as short as possible to keep the full path to the transaction log file names under ASE's 127 character limit. For example, create the toolkit directory as /tk. Alternatively, link the dSource using the command line interface and specify the "mountBase (see page 2616)" parameter to mount the staging database's devices under a directory following your own naming convention.

Target host requirement	Explanation
(Optional - If the target host will be used as a staging target environment) Must have at least one of the following two options configured: • use sp_addserver to add the staging ASE instance's Backup Server to sysservers on the source ASE instance (so that remote database dump/load works) OR • Full and transaction dump files from the source database must be available locally to the staging database (over NFS, replication, scp, etc.)	(see Managing SAP ASE Environments (see page 1903))

9.10.5.2.7 Special considerations - OS user settings (If target host is hosting staging DBs)

Target host requirement	Explanation
If you don't add the Delphix operating system user to the SAP ASE instance owner's group, greater permissions will need to be granted to the backup files to ensure read access to the dumps and/or transaction logs.	Delphix looks for the backup files on the staging host (unless a "remote" backup server is used in which case, the remote host is used which is often the source environment).
ASE Instance Owner Write Permissions to the toolkit (or mount point if you use the CLI to specify a directory other than the toolkit)	If you sync Delphix with a dSource by asking ASE to create a new backup, the SAP ASE instance owner will need write permission to the toolkit (or the mount point if you use the CLI to specify a directory other than the toolkit). Delphix will issue the "DUMP DATABASE" command to write to the staging database's "temp" directory which is mounted on the staging host.
Has write permission for the mount-point directory	The mount-point directory defaults to the toolkit directory, which has write permissions. However, the mount-point directory can be customized, it only needs to have write permissions.

9.10.5.2.8 Additional ASE tuning recommendations

- In SAP ASE 15.7 SP60 and higher, there is a configuration parameter named "enable large pool for load". SAP ASE automatically tunes the caches for recovery on reboot but does not do this for load database and load transaction recovery, since it could impact other databases on a production server. Delphix recommends that the staging SAP ASE instance be separate from the SAP ASE production instance and SAP ASE instances hosting VDBs so enabling this parameter should have *a beneficial impact on performance for the SAP ASE staging instance.*
- Specify the "relaxed" strategy for replacing the cache in the default data cache (sp_cacheconfig 'default data cache', relaxed) in the SAP ASE staging instance. Since the mount/unmount process invalidates the pages anyway, the page chain is really just unneeded overhead on the staging server.
- Staging and target SAP ASE instances should have disk mirroring disabled.
  sp_configure "disable disk mirroring" – run value should be 1, which is the default. If it is 0, change it using
  sp_configure "disable disk mirroring", 1 – this parameter is static so the ASE instance must be restarted for this change to take effect.
- Delphix will mirror the number of devices used on the source database for the staging database (dSource) and each VDB created from that source database. The number of devices parameter should be scaled appropriately based on the max number of virtual databases that will be provisioned to the SAP ASE instance. This parameter can be changed using: sp_configure "number of devices", . SAP ASE 15.7.0 SP100 and later releases support the shrink command. In some cases, Delphix must increase the number of devices used for databases if this command is used. Delphix creates a minimum of the same number of devices as the source database for the staging database (dSource) and each VDB and will add more devices for every 4TB of fragment holes. See SAP ASE issue CR#799273[576] for additional details.

To support multiple VDBs and the staging databases, you may need to increase the parameter number of alarms.

> Delphix uses SAP ASE operations that use alarm structures such as MOUNT and UNMOUNT. The number of alarms limits the number of these operations which can be run concurrently. Various SAP ASE instance failures can occur if the available alarm structures are exhausted. The amount of memory consumed by increasing the number of alarm structures is small. Delphix recommends that the number of alarms value is increased to at least 4096.

9.10.5.2.9 ASE manual discovery

When an SAP ASE environment is added Delphix automatically discovers your SAP ASE instances. Manual discovery allows users to add instances that were not automatically discovered. This feature is currently only supported via the CLI. For more information please refer to Configuring ASE Manual Discovery (see page 2539)

576 https://launchpad.support.sap.com/#/notes/2332779

9.10.5.3 Network and connectivity requirements for SAP ASE environments

9.10.5.3.1 General outbound from the Delphix engine port allocation

Protocol	Port numbers	Use
TCP	25	Connection to a local SMTP server for sending email
TCP/UDP	53	Connections to local DNS servers
UDP	123	Connection to an NTP server
UDP	162	Sending SNMP TRAP messages to an SNMP Manager
TCP	443	HTTPS connections from the Delphix Engine to the Delphix Support upload server
TCP/UDP	636	Secure connections to an LDAP server
TCP	8415	Connections to a Delphix replication target. See Configuring Replication (see page 2381)
TCP	50001	Connections to source and target environments for network performance tests.

9.10.5.3.2 General inbound to the Delphix engine port allocation

Protocol	Port number	Use
TCP	22	SSH connections to the Delphix Engine
TCP	80	HTTP connections to the Delphix GUI
UDP	161	Messages from an SNMP Manager to the Delphix Engine
TCP	443	HTTPS connections to the Delphix Management Application

Protocol	Port number	Use
TCP	8415	Delphix Session Protocol connections from all DSP-based network services including Replication, SnapSync for Oracle, V2P, and the Delphix Connector.
TCP	50001	Connections from source and target environments for network performance tests via the Delphix CLI.

9.10.5.3.3 Firewalls and intrusion detection systems (IDS)

Production databases on source environments (for dSources) are often separated from the non-production environment by firewalls. Firewalls can add milliseconds to the latency between servers. Accordingly, for best performance, there should be no firewalls between the Delphix Engine and the virtual database (VDB) target environments. If the Delphix Engine is separated from a source environment by a firewall, the firewall must be configured to permit network connections between the Delphix Engine and the source environments for the application protocols (ports) listed above.

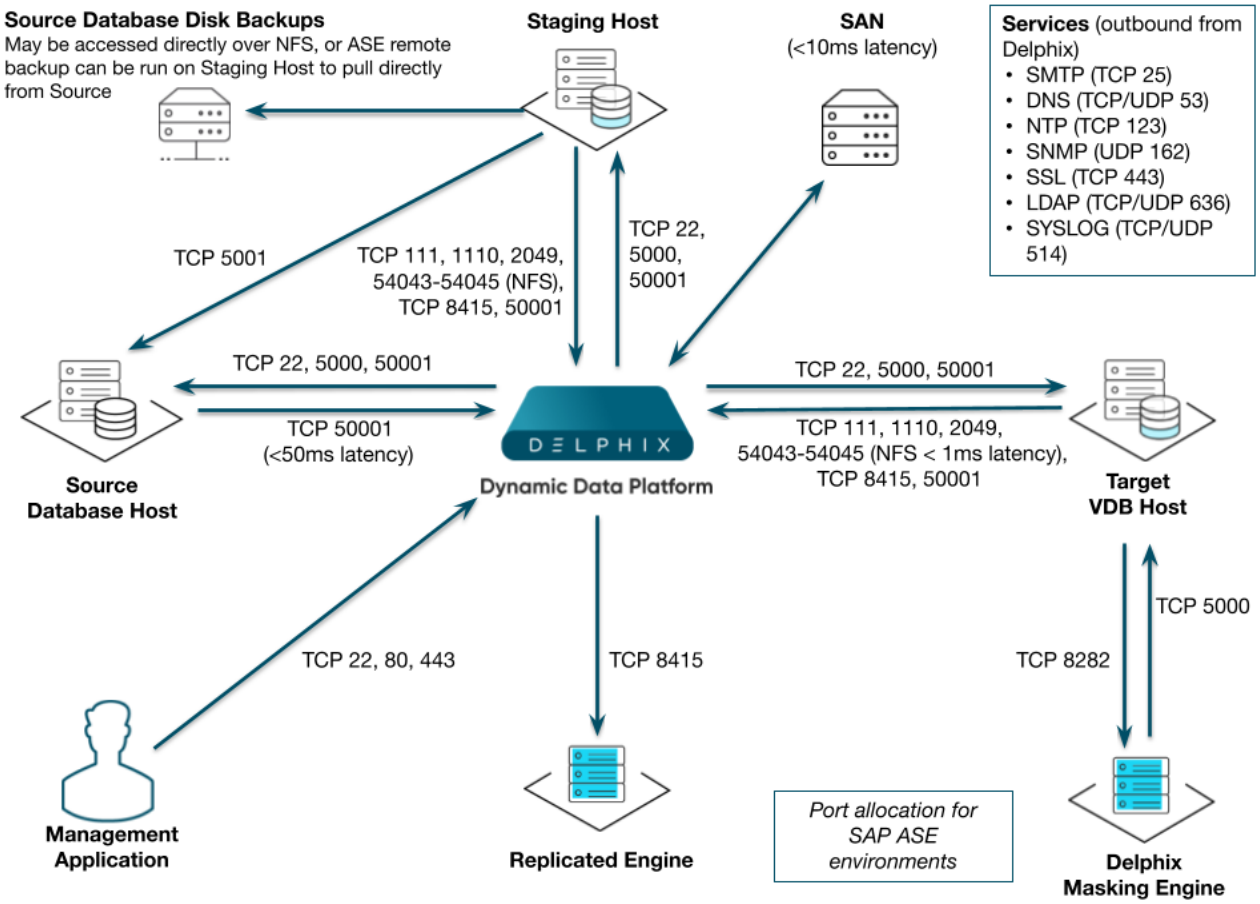
Intrusion detection systems (IDSs) should also be made permissive to the Delphix Engine deployment. IDSs should be made aware of the anticipated high volumes of data transfer between dSources and the Delphix Engine.

9.10.5.3.4 Connection requirements for SAP ASE environments

- The Delphix Engine uses an **SSH** connection to each source environment and **SAP ASE client** connections to the SAP ASE instances on the source environment.
- The Delphix Engine uses an **SSH** connection to each target environment, **NFS** connections from each target environment to the Delphix Engine, and **SAP ASE JDBC** connections to the virtual databases on the target environment.

9.10.5.3.5 Port allocation for SAP ASE environments

The following diagram describes the port allocations for SAP ASE environments. It illustrates the ports that we recommend to be open from Delphix to remote services, to the Delphix Engine, and to the Target Environments.



Refer to Managing SAP ASE Environments (see page 1904) for information on SAP ASE environments. The Delphix Engine makes use of the following network ports for SAP ASE dSources and VDBs:

9.10.5.3.5.1 Outbound from the Delphix engine port allocation

Protocol	Port numbers	Use
TCP	Configuration dependent	JDBC Connections to the SAP ASE instances on the source environments

9.10.5.3.5.2 Inbound to the Delphix engine port allocation

Protocol	Port number	Use
UDP	33434-33464	Traceroute from source and target database servers to the Delphix Engine (optional)

Protocol	Port number	Use
TCP/UDP	111	Remote Procedure Call (RPC) port mapper used for NFSv3 mounts
TCP	2049	NFS client from target hosts to the Delphix Engine (NFSv3 and NFSv4)
TCP	1110	Network Status Monitor (NSM) client from target hosts to Delphix Engine
TCP	54043	Client mount daemon (NFSv3 only)
TCP	54044	Lock state notification service (NFSv3 only)
TCP	54045	Network Lock Manager (NLM) client from target hosts to Delphix Engine (NFSv3 only)
TCP	54046	Connections from Source and Target Environments to the Engine When NFS Encryption is enabled

9.10.5.3.5.3 Port allocation between source and staging target environments

Protocol	Port numbers	Use
TCP	Configuration dependent	SAP ASE Remote Backup Server protocol. Applies if linking using the **New Full Backup** option, or if linking with the **Remote Backup Server** option.

9.10.5.3.6 Port allocation between staging target environments and shared backup fileserver

Protocol	Port numbers	Use
TCP/UDP	NFS and related port numbers: • Portmap (111) • NFS (2049) • Network Lock Manager (NLM) • Network Status Monitor (NSM)	NFS mount point exported by an NFS shared backup fileserver. Applies if linking using the **Local Backup Server** option.

9.10.5.3.7 AppData port requirements

The use of AppData requires the following ports/protocols. Two important notes about these specifications:

1. The next release of the Delphix Engine will significantly augment the port/protocol utilization of AppData. The upcoming-only requirements have been marked with a *.
2. AppData V2P uses RSYNC to export to the target. RSYNC between the target and Delphix Engine is not required for general virtualization usage. The V2P-only requirements have been marked with a ^.

From Source to Delphix Engine	From Delphix Engine to Source	From Target to Delphix Engine	From Delphix Engine to Target
RSYNC to rsyncd[577] (TCP Port 873)	RSYNC to rsyncd[578] (TCP Port 873)	DSP (Default TCP Port 8415)	DSP (Default TCP Port 8415)
DSP (Default TCP Port 8415)	SSH (TCP Port 22)	NFS	SSH (TCP Port 22)

577 https://manpages.ubuntu.com/manpages/focal/man5/rsyncd.conf.5.html
578 https://manpages.ubuntu.com/manpages/focal/man5/rsyncd.conf.5.html

From Source to Delphix Engine	From Delphix Engine to Source	From Target to Delphix Engine	From Delphix Engine to Target
*NFS	DSP (Default TCP Port 8415)	^RSYNC to rsyncd[579] (TCP Port 873)	^RSYNC to rsyncd[580] (TCP Port 873)

9.10.5.4 Sudo privilege requirements for SAP ASE environments

This topic describes the rationale behind specific `sudo` privilege requirements for virtualizing SAP ASE Databases.

Privilege	Sources	Targets	Rationale
`pargs`	**Required on Solaris**	**Required on Solaris**	Delphix attempts to call `pargs` to discover the arguments of the ASE processes. It needs the name of each running dataserver or backupserver process so that it can try to connect to the instances to gather further information during the discovery process.
`ps`	Optional on Linux, AIX	Optional on Linux, AIX	Delphix attempts to call `ps` to discover the arguments of the ASE processes. It needs the name of each running dataserver or backupserver process so that it can try to connect to the instances to gather further information during the discovery process. Unlike Solaris, Delphix can usually determine the arguments without sudo privileges on Linux/AIX. But Delphix will attempt " `sudo ps` " before attempting a regular `ps` command, and this could cause locking of the **delphix_os** account. To avoid locking issues, you can grant `sudo ps` to **delphix_os** .

579 https://manpages.ubuntu.com/manpages/focal/man5/rsyncd.conf.5.html
580 https://manpages.ubuntu.com/manpages/focal/man5/rsyncd.conf.5.html

Privilege	Sources	Targets	Rationale
`mount/umount`	Not Required	**Required**	Delphix dynamically mounts and unmounts directories under the provisioning directory during VDB operations. This privilege is required because `mount` and `umount` are typically reserved for superuser.
`nfso`	Not Required	**Required on AIX**	Delphix monitors NFS read and write sizes on an AIX target host. It uses the `nfso` comman d to query the sizes in order to optimize NFS performance for VDBs running on the target host. Only a superuser can issue the `nfso` co mmand.

Default Mount Directory

By default, Delphix mounts the NFS directories for VDBs and staging databases under the toolkit directory. Sudo permissions should be granted to allow the mount/umount commands to execute under these directories unless the dSource is linked using the command-line interface (CLI) and a different NFS mount base is specified. Please refer to the Reference manual for more information on linking the dSource using the CLI and specifying the "mountBase (see page 2616)" parameter.

Specify the NOPASSWD qualifier

It is required to specify the NOPASSWD qualifier within the "sudo" configuration file, as shown here: Sudo File Configuration Examples for SAP ASE Environments (see page 1900).[581] This ensures that the "sudo" command does not demand the entry of a password, even for the "display permissions" (i.e. "sudo -l") command.

Delphix issues "sudo -l" in some scripts to detect if the operating system user has the correct sudo privileges. If it is unable to execute this command, some actions may fail and Delphix will raise an alert suggesting it does not have the correct sudo permissions. Restricting the execution of "sudo -l" by setting “listpw=always” in the “/etc/sudoers” file when the Delphix operating system user is configured to use public key authentication will cause the Delphix

581 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/sudo-file-configuration-examples-for-sap-ase-environments

operating system user to be prompted for a password which will fail certain Delphix actions. Use a less restrictive setting for listpw than "always" when the Delphix operating system user is using public-key authentication.

SAP ASE and appData mount options	
AIX	`-o cio,rw,fg,hard,rsize=$nfs_rsize,wsize=$nfs_wsize,nointr,timeo=600,proto=tcp,noacl`
HPUX	`-o rw,hard,rsize=1048576,wsize=1048576,nointr,timeo=600,proto=tcp,suid`
Solaris	`-F nfs -o rw,fg,hard,rsize=1048576,wsize=1048576,nointr,timeo=600,proto=tcp,suid,sec=sys`
For these platforms, depending on the NFS version used, additional options vers=3 or vers=4.x is added (x varies depending on what that platform supports. e.g. vers=4 or vers=4.1)	
Linux (NFSv3)	-t nfs -o rw,fg,hard,rsize=1048576,wsize=1048576,nointr,timeo=600,tcp,noacl,vers=3
Linux (NFSv4)	-t nfs4 -o rw,fg,hard,rsize=1048576,wsize=1048576,nointr,timeo=600,sec=sys,tcp,noacl
(For some flavors of Linux and NFSv4.1, additional optional 'v4.1' is added)	

1. AppData plugins and toolkits have some additional mount options depending on the type of toolkit/plugin.
2. "port=2049" is added for all the platforms.

SAP ASE and zppData unmount options	"-f" in most cases. Certain cases, SAP ASE uses "-lf". (Lazy unmount option)

Mount and unmount options subject to change

Please note that the mount and unmount options listed above are subject to change. For example, if Delphix finds that a certain option improves performance, Delphix may add, remove or change options at anytime. Therefore, it is highly recommended to create the sudo profiles using wildcards that allow any number of options.

9.10.5.5 Sudo file configuration examples for SAP ASE environments

This topic provides sample `sudo` file privilege configurations for using the Delphix Engine with various operating systems and SAP ASE.

9.10.5.5.1 Configuring `sudo` access on Solaris for SAP ASE source and target environments

Sudo access to `pargs` on the Solaris operating system is required to discover the arguments of the ASE processes both source and target environments.

Example: Solaris /etc/sudoers entries for a Delphix Source for SAP ASE

```
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD:/usr/bin/pargs
```

On a Solaris target, `sudo` access to `mount` and `umount` is also required.

Example: Solaris /etc/sudoers entries for a Delphix Target for SAP ASE

```
# Delphix issues sudo -l so we need to allow it via listpw. Never set it to always
when using public key authentication
Defaults      listpw=all
User_Alias DELPHIX_USER=delphix_os
Cmnd_Alias DELPHIX_CMDS= \
/usr/sbin/mount, \
/usr/sbin/umount, \
/usr/bin/pargs
DELPHIX_USER ALL=(ALL) NOPASSWD: DELPHIX_CMDS
```

9.10.5.5.2 Configuring `sudo` access on Linux for SAP ASE source and target environments

On a Linux target, sudo access to `mount` and `umount` is required.

Example: Linux /etc/sudoers file for a Delphix Target for SAP ASE

```
# Delphix issues sudo -l so we need to allow it via listpw. Never set it to always
when using public key authentication
Defaults      listpw=all

Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD: \
/bin/mount, /bin/umount
```

9.10.5.5.3 Configuring `sudo` access on AIX for SAP ASE source and target environments

In addition to sudo access to the `mount` and umount commands on AIX target hosts, Delphix also requires `sudo` access to `nfso`. This is required on target hosts for the Delphix Engine to monitor the NFS read write sizes configured on the AIX system. Super-user access level is needed to run the `nfso` comman d.

Example: AIX /etc/sudoers File for a Delphix Target

```
# Delphix issues sudo -l so we need to allow it via listpw. Never set it to always
when using public key authentication
Defaults      listpw=all
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD: \
/bin/mount, \
/bin/umount, \
/usr/sbin/nfso
```

9.10.5.5.4 Examples of limiting `sudo` access for the Delphix OS user

In situations where security requirements prohibit giving the Delphix user root privileges to mount, unmount, make a directory, and remove directory on the global level, it is possible to configure the `sudoers` file to provide these privileges only on specific mount points or from specific Delphix Engines, as shown in these two examples.

> The Delphix Engine tests its ability to run the `mount` command using `sudo` on the target environment by issuing the `sudo mount` command with no arguments. Many of the examples shown in this topic do not allow that. This causes a warning during environment discovery and monitoring but otherwise does not cause a problem. If your VDB operations succeed, it is safe to ignore this warning.
> Similarly, the `ps` or `pargs` the command is used for target environment operations such as initial discovery and refresh.

Some organizations configure the security on the target environments to monitor `sudo` failures and lockout the offending account after some threshold. In those situations, the failure of the sudo commands might cause the **delphix_os** account to become locked. One workaround for this situation is to increase the threshold for locking out the user account. Another option is to modify `/etc/sudoers` to permit the **delphix_os** user to run `ps (pargs), mkdir, rmdir,` and `mount` command without parameters.

Note that the following examples are for illustrative purposes and the sudo file configuration options are subject to change.

9.10.5.5.4.1 Example 1

This example restricts the **delphix_os** user's use of `sudo` privileges to the directory `/sybase`.

Note that wildcards are allowed for the options on `mount` and `umount` because those commands expect a fixed number of arguments after the options. The option wildcard on the `mount` command also makes it possible to specify the file-system being mounted from the Delphix Engine.

However, wildcards are not acceptable on `mkdir` and `rmdir` because they can have any number of arguments after the options. For those commands, you must specify the exact options (`-p`, `-p -m 755`) used by the Delphix Engine.

Example /etc/sudoers File Configuration on the Target Environment for sudo Privileges on the VDB Mount Directory Only (Linux OS)

```
# Delphix issues sudo -l so we need to allow it via listpw. Never set it to always
when using public key authentication
Defaults      listpw=all

Defaults:delphix_os !requiretty
delphix_os ALL=(root) NOPASSWD: \
/bin/mount  *        /sybase/*, \
/bin/mount "", \
/bin/umount *        /sybase/*, \
/bin/umount          /sybase/*, \
/bin/mkdir -p        /sybase/*, \
/bin/mkdir -p -m 755 /sybase/*, \
/bin/mkdir           /sybase/*, \
/bin/rmdir           /sybase/*, \
/bin/ps
```

Example /etc/sudoers File Configuration on the Source Environment to grant Super-User privileges when running PS

```
# Delphix issues sudo -l so we need to allow it via listpw. Never set it to always
when using public key authentication
Defaults      listpw=all

Defaults:delphix_os !requiretty
delphix_os ALL=(root) NOPASSWD: /bin/ps
```

9.10.5.5.4.2 Example 2

This example restricts the **delphix_os** user's use of `sudo` privileges to the directory /sybase, restricts the mount commands to a specific Delphix Engine hostname and IP, and restricts user-specified options for the `umount` command.

This configuration is more secure, but there is a tradeoff with deployment simplicity. This approach would require a different sudo configuration for targets configured for different Delphix Engine.

Configuring the /etc/sudoers File on the Target Environment for Privileges on the VDB Mount Directory Only, and Allows Mounting only from a Single Server (Linux OS)

```
# Delphix issues sudo -l so we need to allow it via listpw. Never set it to always
when using public key authentication
Defaults      listpw=all

Defaults:delphix_os !requiretty
delphix_os ALL=(root) NOPASSWD: \
/bin/mount           <delphix-server-name>* /sybase/*, \
/bin/mount *         <delphix-server-name>* /sybase/*, \
/bin/mount           <delphix-server-ip>*   /sybase/*, \
/bin/mount *         <delphix-server-ip>*   /sybase/*, \
/bin/mount "", \
/bin/umount          /sybase/*, \
/bin/umount *        /sybase/*, \
/bin/mkdir [*]       /sybase/*, \
/bin/mkdir           /sybase/*, \
/bin/mkdir -p        /sybase/*, \
/bin/mkdir -p -m 755 /sybase/*, \
/bin/rmdir           /sybase/*, \
/bin/ps
```

9.10.6 Managing SAP ASE environments and hosts

This section describes the attributes of SAP ASE environments such as information for the Delphix Connector and covers the following topics:

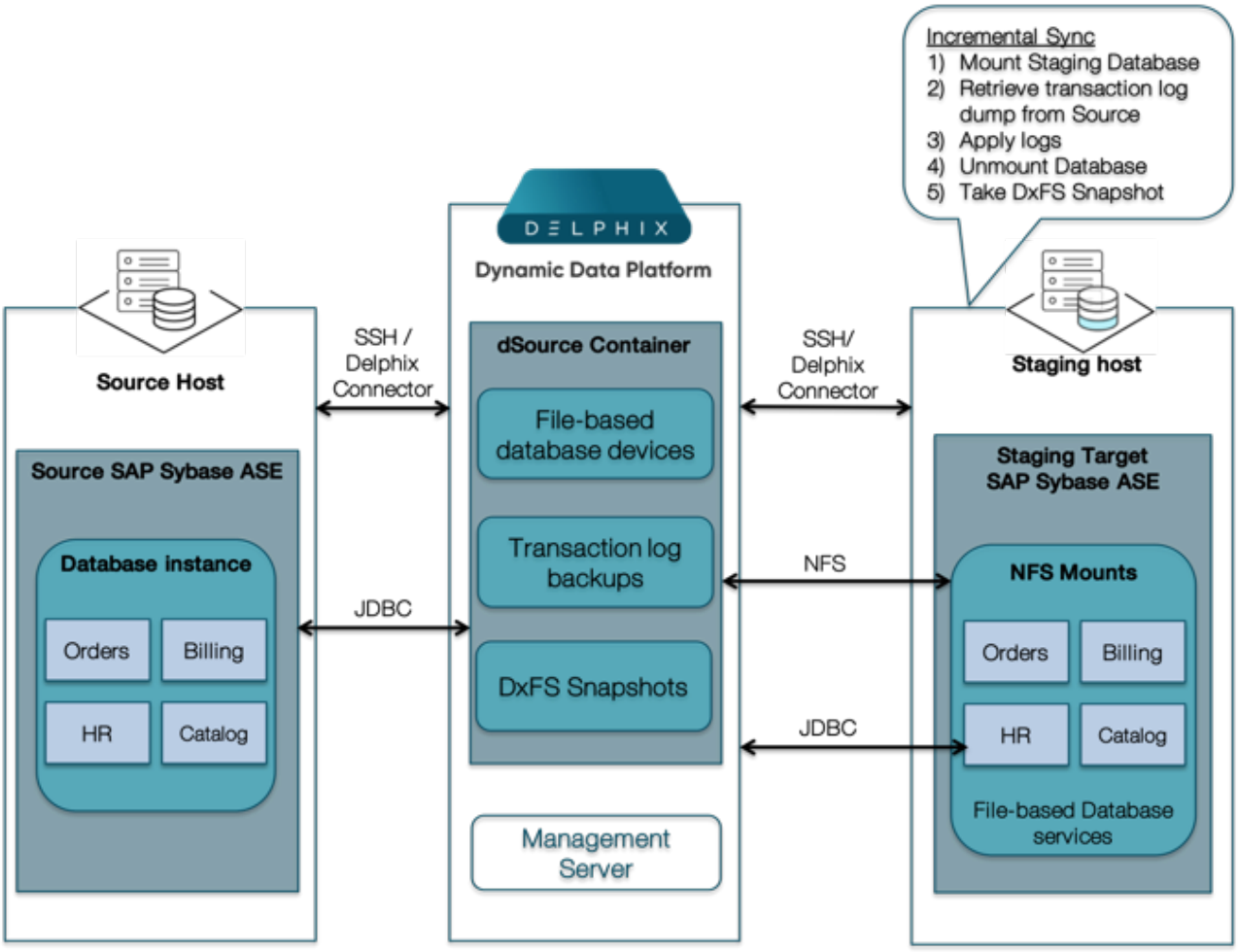
- Managing SAP ASE environments overview (see page 1904)
- Adding an SAP ASE environment (see page 1905)
- Environment attributes for hosts with SAP ASE (see page 1907)
- How to discover SAP ASE instances which use multiple network handlers (see page 1909)

- Changing the host name or IP address of an SAP ASE environment (see page 1911)
- Using HostChecker to validate SAP ASE source and target environments (see page 1912)
- Enabling linking and provisioning for SAP ASE environments (see page 1916)

9.10.6.1 Managing SAP ASE environments overview

This topic describes the high-level process for adding SAP ASE environments, linking SAP ASE databases to the Delphix Engine, and provisioning virtual databases.

9.10.6.1.1 Block diagram of linking architecture between SAP ASE environments and the Delphix engine



9.10.6.1.2 Environment setup

SAP ASE dSources are backed by a staging database that runs on a target host, as shown in the diagram. There is no requirement for additional local storage on this host, as the storage is mounted over NFS from the Delphix Engine. At Delphix, we refer to the creation and maintenance of this staging database on the staging host as "validated sync," because it prepares the dSource data on the Delphix Engine for provisioning VDBs later on. After the Delphix Engine creates the staging database, it continuously monitors the source

database for new transaction log dumps. When it detects a new transaction log dump, it loads that dump to the staging database. The result is a TimeFlow with consistent points from which you can provision a virtual database (VDB), and a faster provisioning process because there is no need for any database recovery during provisioning.

When you later provision a VDB, you can specify any environment as a target, including the environment that contains the staging database. However, for best performance, Delphix recommends that you choose a different target environment. The target must have an operating system that is compatible with the one running on the validated host, as described in Requirements for SAP ASE Environments and Databases (see page 1884)

9.10.6.1.3 Workflow and tasks for SAP ASE environments

1. Add the desired source environments as described in Managing SAP ASE Environments (see page 1903)
2. Add the desired target environments as described in Managing SAP ASE Environments (see page 1903)
3. Link the source database as described in Linking Data Sources with SAP ASE (see page 1918)
4. Provision VDBs as described in Provisioning an SAP ASE VDB (see page 1931)

> Best Practice
> If possible, it is recommended to use separate SAP ASE instances for staging databases (dSources) and VDBs on target environments. This allows better control on target environments based on the resources required for SAP ASE full/transaction log dump ingestion vs VDB performance.

9.10.6.2 Adding an SAP ASE environment

9.10.6.2.1 Prerequisites

See Requirements for SAP ASE Environments and Databases (see page 1884)

9.10.6.2.2 Procedure

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Environments**.
4. From the **Actions (...)** menu select **Add Environment**.
5. In the **Host and Server** tab window, select **Unix/Linux**.
6. Select **Standalone Host.**

7. Click **Next.**
8. Enter a **Name** for the environment.
9. Enter the **Host IP** address.
10. For NFS Addresses (Optional): Enter one or more comma-separated **IP Address/Hostname**

    Note: If specified, Delphix Engine only allows NFS requests (mount, etc) originated from IP Addresses specified for the host.
11. Enter the **SSH** port. The default value is **22**.
12. Select a **Login Type**. — Username and Password - enter the OS username and password — Username and Public Key - enter the OS username. — Password Vault - select from an existing Enterprise Password Vault

    Note:
    Using Public Key Authentication

    If you want to use public-key authentication for logging into your Unix-based environment, there are two options: use the engine's key pair or provide a key pair for this environment.

    To use the engine's key pair:
    a. Select **Public Key** for the **Login Type.**
    b. Click **View Public Key**.
    c. Copy the public key that is displayed, and append it to the end of your `~/.ssh/authorized_keys` file. If this file does not exist, you will need to create it.
        i. Run chmod 600 ~/.ssh/authorized_keys to allow only the file's owner to read and write to it (make sure the file is owned by the user).
        ii. Run chmod 755 ~ to restrict access to the user's home directory so no other user may write to it.
        iii. Run chmod 700 ~/.ssh so that others cannot write to it. The ~/.ssh directory cannot be writable by group or other users. Otherwise, authentication will fail.

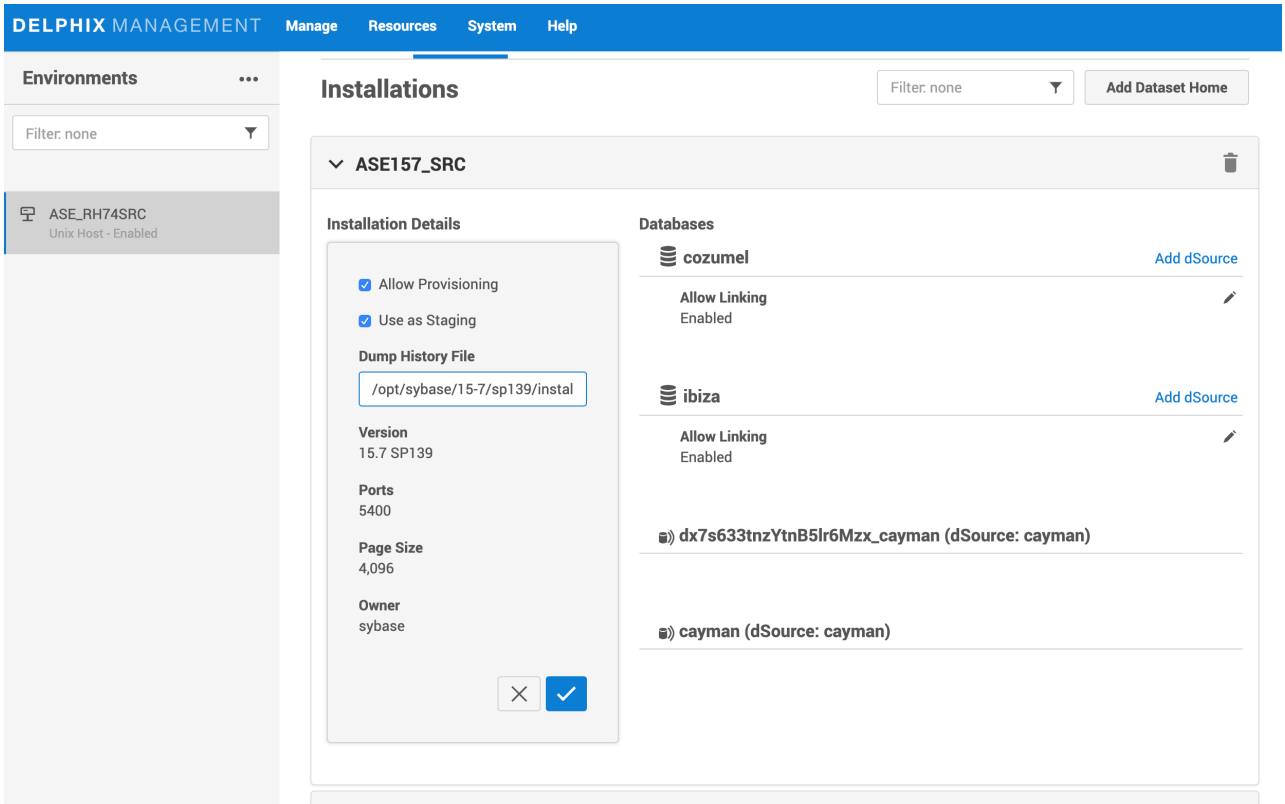
    As an alternative, you can provide a key pair specific for this environment via the API or CLI. See Option 2 in this CLI Cookbook article (see page 1905) for instructions.
13. For **Password Login**, click **Verify Credentials** to test the username and password.
14. Enter a **Toolkit Path**. The toolkit directory stores scripts used for Delphix Engine operations. It must have a persistent working directory rather than a temporary one. The toolkit directory will have a separate subdirectory for each database instance. The toolkit path must have 0770 permissions.
15. To provide your own Oracle Java select the **Provide my own JDK** checkbox and click **Next**.
16. In the Java Development Kit tab enter the absolute path to your Oracle JDK and click **Next**.
17. Click the **Discover SAP ASE** checkbox.
18. Click **Next**.
19. In the Summary, tab confirm your selections.
20. Click **Submit**.

9.10.6.2.3 Post-requisites

After you create the environment, you can view information about it by selecting **Manage > Environments** and then selecting the **environment name**.

9.10.6.2.4 Configuring a non-default dump history file

Delphix, by default, uses the Dump History file that is configured for the source ASE instance. Delphix will query the source ASE instance to find this file name and use this file to find dump history. However, users can specify any other file on the source host to be used for querying the backup history. In that case, Delphix will not use the file configured for the source ASE instance but will use the file specified to get backup information. To specify a custom Dump History file, go to the **source environment > Databases**. Click the pencil button in front of **Installation Details** for the desired instance. Enter the fully qualified name of the Dump History file to be used in the text box next to **Dump History File**.



Delphix does not support rolled over Dump History Files.

9.10.6.3 Environment attributes for hosts with SAP ASE

This topic describes the attributes of SAP ASE environments. Below you will see a section for common environment attributes shared by all types of environments as well as SAP ASE specific ones.

9.10.6.3.1 Procedure

1. From your Delphix Engine, click **Manage**.
2. Select **Environments**.
3. In the **Environments** panel, click the name of an environment to view its attributes.
4. Next to **Attributes**, click the Pencil icon to edit an attribute.

9.10.6.3.2 Common environment attributes

Attribute	Description
Environment Users	The users for that environment. These are the users who have permission to ssh into an environment or access the environment through the Delphix Connector. For more information on the environment user requirements, see the Requirements topics for specific data platforms.
Host Address	The IP address of the environment host.
DSP KeyStore Path	The path to the user-managed DSP Keystore.
DSP KeyStore Alias	The lowercase alias to use inside the user-managed DSP Keystore.
DSP KeyStore Password	The password for the user-managed DSP Keystore.
DSP TrustStore Path	The path to the user managed DSP truststore.
DSP TrustStore Password	The password for the user managed DSP truststore.
OS	The name of the host operating system.
Version	The version of the host operating system.
Release	The release of the host operating system.
Time Zone	The timezone of the host operating system.

Attribute	Description
Total RAM	The amount of RAM on the host machine.
Processor Type	The processor type of the host machine.
Traceroute	Traceroute info from target host to Delphix Engine.
Notes	Any other information you want to add about the environment.
Java Development Kit	The currently selected JDK kit will be shown.
Java Development Kit (JDK) Path	Location of the Java Development Kit (JDK) used for the host. Only specified if the feature to provide your own JDK is enabled, otherwise, the defaults are used per our Java Support Matrix (see page 1025).

9.10.6.3.3 SAP ASE environment attributes

Attribute	Description
Database User	User for Delphix to use for SAP ASE database operations.
DB Password	Credentials to use for the database user in this environment.

9.10.6.4 How to discover SAP ASE instances which use multiple network handlers

When adding a SAP ASE or Sybase ASE environment, the Delphix Engine may not be able to discover ASE instances that are not listening on the same network address that the engine used for discovery. In these cases, you may receive the following warning:

```
WARNING: Error during discovery for instance "INSTANCENAME" : "Failed to connect to instance "INSTANCENAME over JDBC.". Skipping discovery for instance "INSTANCENAME".
```

9.10.6.4.1 Troubleshooting

An ASE instance may have been configured to listen on other network interfaces for one of the following reasons:

- To support the use of clustering/failover technologies such as Veritas Cluster Server
- To support the concept of "virtual hostnames" or "virtual IPs" (VIPs) to provide abstraction for client applications
- To facilitate a previous database migration to new infrastructure
- Database administrators configured it manually for other reasons

For guidance and best practices for configuring the Delphix Engine to work with clustering or failover technologies, contact your Professional Services or Customer Success representative.

To verify whether an ASE instance is using multiple network interfaces, use the `sp_listener` stored procedure when connected to the instance using `isql`:

```
1> sp_listener status 2> go  proto host port status
------------------------------------------------------------------  tcp
secondary-hostname-or-ip-address 4559 active
```

As described in the ASE document Configuring the Server for Multiple Network Handlers[582], this configuration is read from the `$SYBASE/interfaces` file on your ASE server during instance startup.

9.10.6.4.2 Resolution

To allow discovery of instances using multiple network interfaces, use the **Add Environment** dialog to add the environment ONCE per network interface.

Each environment added in this way should have both of the following:

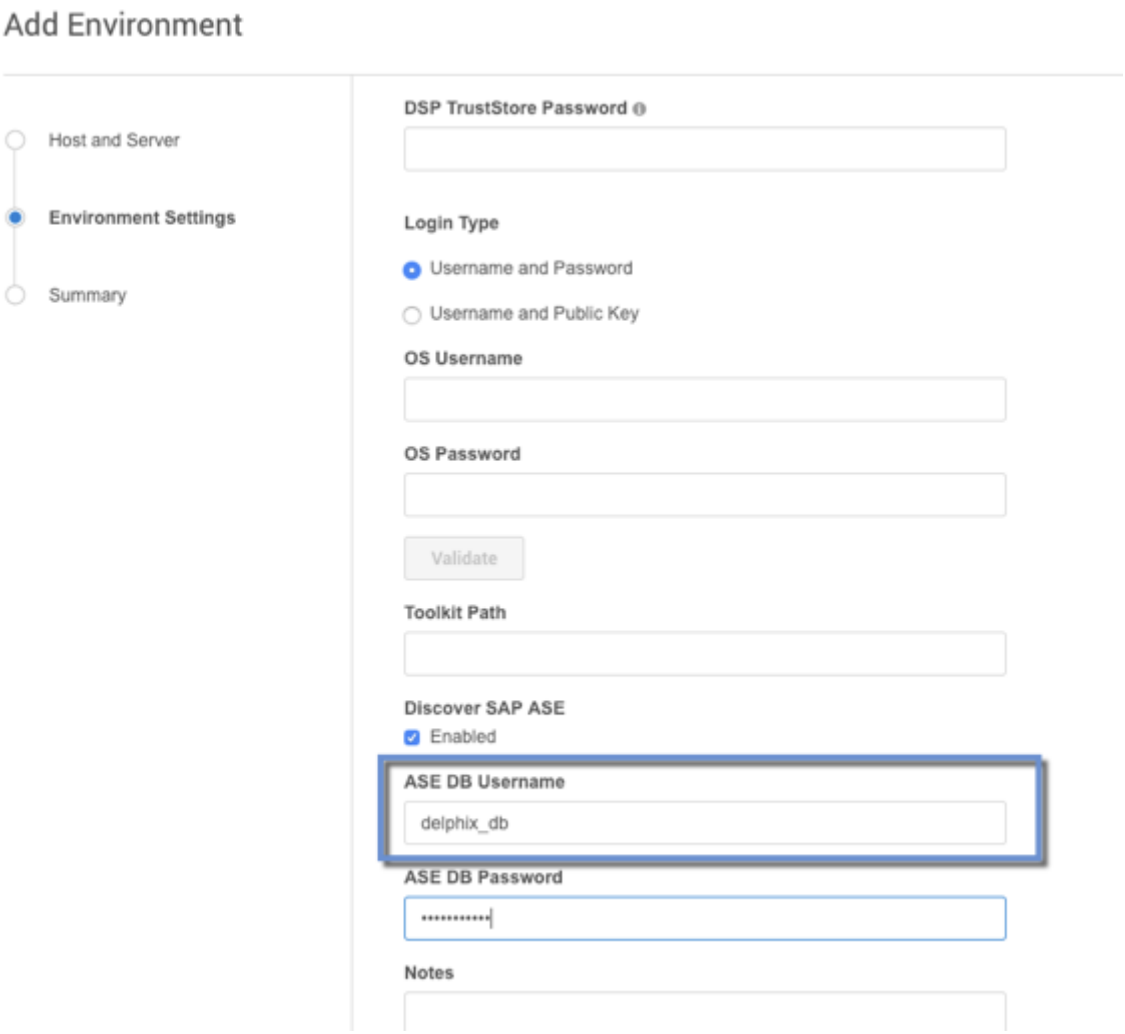
- A unique Host Address; and
- A unique Toolkit Path

The Delphix Engine expects that the files deployed to the Delphix Toolkit directory are persistent. Any failover of the ASE service to a different host must ensure that the Delphix Toolkit directory remains consistent between both hosts, with all files and permissions intact.

Deploying the Delphix Toolkit multiple times will result in increased disk space consumption. It may circumvent concurrency limitations that the Delphix Engine introduces to minimize its impact on your source environments. If your environment requires you to add the same host several times using this process, contact your Delphix Professional Services representative or Customer Success Manager to discuss best practices or alternative solutions.

582 http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.infocenter.dc35823.1600/doc/html/san1334282784204.html

Using the example interfaces file provided above, you could use the following inputs to add the `INSTANCENAME` instance:



9.10.6.5 Changing the host name or IP address of an SAP ASE environment

This topic describes how to change the hostname or IP address for source and target environments, and for the Delphix Engine.

9.10.6.5.1 Procedure

9.10.6.5.1.1 For source environments

1. Disable the dSource as described in Enabling and Disabling SAP ASE dSources (see page 1927)
2. If the **Host Address** field contains an IP address, edit the IP address.

3. If the **Host Address** field contains a hostname, update your Domain Name Server to associate the new IP address to the hostname. The Delphix Engine will automatically detect the change within a few minutes.
4. In the **Environments** screen of the Delphix Engine, refresh the host.
5. Enable the dSource.

9.10.6.5.1.2 For VDB target environments

1. Disable the VDB as described in Enabling and Disabling SAP ASE dSources (see page 1927)
2. If the **Host Address** field contains an IP address, edit the IP address.
3. If the **Host Address** field contains a hostname, update your Domain Name Server to associate the new IP address to the hostname. The Delphix Engine will automatically detect the change within a few minutes.
4. In the **Environments** screen of the Delphix Engine, refresh the host.
5. Enable the VDB.

9.10.6.5.1.3 For the Delphix engine

1. To stop running your VDB select the red **Stop** button located on the VDB **Configuration** tab.
2. Disable all dSources as described in Enabling and Disabling SAP ASE dSources (see page 1927)
3. You can use either the command-line interface or the Delphix Setup application to change the IP address of the Delphix Engine.
   a. To use the command-line interface, press **F2** and follow the instructions described in Setting Up Network Access to the Delphix Engine (see page 569)
   b. To use the Delphix Setup application, go to **Delphix Management > Engine Setup** in the Delphix Management application, or click **Server Setup**in the Delphix Engine login screen.
      i. In the **Network** panel, click **Modify**.
      ii. Under **DNS Services**, enter the new IP address.
      iii. Click **OK**.
4. Refresh all Environments by clicking the **Refresh** symbol on the **Environments** screen.
5. Enable all dSources as described in Enabling and Disabling SAP ASE dSources (see page 1927)
6. To start all VDBs, click the **Start** button located on the VDB **Configuration** tab.

9.10.6.6 Using HostChecker to validate SAP ASE source and target environments

9.10.6.6.1 What is HostChecker?

The HostChecker is a standalone program which validates that host machines are configured correctly before the Delphix Engine uses them as data sources and provision targets.

Please note that HostChecker does not communicate changes made to hosts back to the Delphix Engine. If you reconfigure a host, you must refresh the host in the Delphix Engine in order for it to detect your changes.

You can run the tests contained in the HostChecker individually, or all at once. You must run these tests on both the source and target hosts to verify their configurations. As the tests run, you will either see validation messages that the test has completed successfully, or error messages directing you to make changes to the host configuration.

9.10.6.6.2 Prerequisites

- Make sure that your source and target environments meet the requirements specified in SAP ASE Support and Requirements (see page 1877)

9.10.6.6.3 Procedure

1. Download the HostChecker tarball matching the host's operating system from https://download.delphix.com/(for example: hostchecker_linux_x86.tar).
2. Create a working directory and extract the HostChecker files from the HostChecker tarball.

```
mkdir dlpx-host-checker
cd dlpx-host-checker/
tar -xf hostchecker_linux_x86.tar
```

3. Change to the working directory and enter this command. Note that for the target environments, you would change `source` to `target`.

```
$ cd hostchecker
$ ./hostchecker.sh
Extracting the JDK from the tarball jdk.tar.gz.
Please enter whether this machine is a source or a target:target
1: Check ASE environment
2: Check all ASE instances
3: Check all the Oracle installations
4: Check homedir permissions
5: Check Linux Performance Settings
6: Check mkdir and rmdir
7: Check the MySQL installation
8: Check network port access
9: Check the Oracle CRS home
10: Check for ssh connectivity
11: Check sshd_config for timeout configuration
12: Check user sudo privileges
13: Check toolkit path
all: Execute all checks
quit: Exits
Please select an option:
```

Note:
Don't Run as Root

Don't run the HostChecker as root; this will cause misleading or incorrect results from many of the checks.

4. Select which checks you want to run. We recommend you run all checks (excluding Oracle and MySQL) if you are running Hostchecker for the first time.
5. Pass in the arguments the checks ask for.
6. Read the output of the check.
7. The error or warning messages will explain any possible problems and how to address them. Resolve the issues that the HostChecker describes. Don't be surprised or undo your work if more errors appear the next time you run HostChecker, because the error you just fixed may have been masking other problems.
8. Repeat steps 3–7 until all the checks return no errors or warnings.

9.10.6.6.4 Tests run

Test	SAP ASE Source	SAP ASE Target	Description
Check SAP ASE environment	X	X	• Checks that the **$SYBASE** environment variable exists for interactive logs. **NOTE**: It is still necessary to check that it is defined for non-interactive logins (ssh user@host env \| grep SYBASE). • Checks that the "**isql_r64**" binary can be found underneath the **$SYBASE** environment variable.

Test	SAP ASE Source	SAP ASE Target	Description
Check All SAP ASE instances		X	Attempts to connect to each running SAP ASE instance via "**isql_r64**" (utilizing the $SYBASE/interfaces file) and execute the following queries: select @@servername select count(*) from master..syslisteners select count(*) from master..sysservers select count(*) from master..sysdatabases select count(*) from master..sysusages select @@version select @@maxpagesize SELECT srvnetname FROM master..sysservers WHERE srvname='SYB_BACKUP'
Check all the Oracle installations	N/A	N/A	
Check homedir permissions	X	X	Check that the home directory, the ~/.ssh directory, and the ~/.ssh/authorized_keys file exist, that they are owned by the user invoking this check, and that they are not 'group' or 'other' writable.
Check performance="">>		X	Check target's kernel settings necessary to optimize performance.
Check mkdir and rmdir		X	Tests that the user can mkdir and rmdir under both the toolkit directory and the specified mount path.
Check the MySQL installation	N/A	N/A	

Test	SAP ASE Source	SAP ASE Target	Description
Check network port access	X	X	Can be used to test access to specified ports on the Delphix Engine. See Network and Connectivity Requirements for SAP ASE Environments (see page 1892) for a list of ports.
Check the Oracle CRS home	N/A	N/A	
Check Oracle DB Instance	N/A	N/A	
Check for ssh connectivity	X	X	Verifies that the environment is accessible via SSH
Check sshd_config for timeout configuration		X	Check that sshd_config does not contain ClientAliveInterval or ClientAliveCountMax entries.
Check toolkit Path	X	X	Verifies that the toolkit installation location has the proper ownership, proper permissions, and enough free space.
Check user sudo privileges	X	X	Verifies that the operating system user can execute certain commands with necessary privileges via `sudo`. This only needs to be run on target environments. See the topic Requirements for SAP ASE Environments and Databases (see page 1884) for more information.

9.10.6.7 Enabling linking and provisioning for SAP ASE environments

This topic describes how to enable and disable provisioning and linking for SAP ASE databases.

Before a database can be used as a dSource, you must first make sure that you have enabled linking to it. Similarly, before you can provision a VDB to a target database, you must make sure that you have enabled provisioning to it.

9.10.6.7.1 Procedure

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Environments**.
4. Select the **Databases** tab.
5. Click the **pencil** icon located next to the database **Installation Details**.
6. Select the **Allow Provisioning** checkbox to enable provisioning, deselect the checkbox to disable provisioning.
7. Click **show details** for the database.
8. Slide the button next to **Allow Linking** to **On** or **Off** to enable or disable linking.

9.10.7 Linking data sources and Syncing Data with SAP ASE

Creating a dSource will ingest data from the source and create a dSource on the engine. The dSource is an object that the Continuous Data Engine uses to create and update virtual copies of your database. As a virtualized representation of your source data, it cannot be managed, manipulated, or examined by database tools.

This section covers the following topics:

- Linking SAP ASE data sources: an overview (see page 1917)
- Linking data sources with SAP ASE (see page 1918)
- Linking data sources with encrypted SAP ASE database (see page 1921)
- Data management settings for SAP ASE dSources (see page 1921)
- Upgrading a SAP ASE dSource (see page 1924)
- Detaching and re-attaching SAP ASE dSources (see page 1924)
- Deleting an SAP ASE dSource (see page 1926)
- Enabling and disabling SAP ASE dSources (see page 1927)
- Working with SAP ASE snapshots (see page 1928)

9.10.7.1 Linking SAP ASE data sources: an overview

This topic describes basic concepts behind the creation of dSources from SAP ASE databases.

9.10.7.1.1 Initial linking and staging databases

A dSource is a copy of a physical database that is created when the Delphix Engine links to and loads the database from a backup. The database backup can be a new full database backup that the Delphix Engine initiates, the most recent existing database backup, or an existing database backup specified by the user. When loading from an existing backup, the backup should be in a location that the source environment user can access.

After obtaining the initial snapshot and linking the dSource, the Delphix Engine keeps the dSource and the source database in sync by monitoring the source database for new transaction log dumps and then applying those backups on a standby database. This database is called the "staging database." A target environment that hosts one or more staging databases is referred to as a "staging target."

After you have linked a database into the Delphix Engine, you can re-initialize it by performing a sync on the dSource.

> There is an SAP ASE feature that affects the size of database dumps which correspondingly affect the amount of space consumed on the storage attached to the Delphix Engine. If the SAP ASE "sp_dumpoptimize" feature is set to maximum as follows, more disk space will be consumed on Delphix storage because SAP ASE writes both allocated and unallocated pages into the dump files:

```
sp_dumpoptimize 'archivespace = maximum'
```

9.10.7.1.2 SAP ASE source and staging database compatibility

You may use different SAP ASE patch level versions between source and staging hosts. The major version of SAP ASE must still be the same. The only caveat is for SAP ASE 15.7 - where the source and staging patch levels must both be either any version below SP64 or both be any version above SP100.

9.10.7.2 Linking data sources with SAP ASE

The dSource is an object that the Continuous Data Engine uses to create and update virtual copies of your database. As a virtualized representation of your source data, it cannot be managed, manipulated, or examined by database tools. For an overview of all dSource related actions, please visit Managing Data Sources (see page 1040).[583]

Continuous Data for SAP ASE databases leverages backup-based ingestion, which means that Delphix will look for, or sometimes initiate the creation of, a backup through your SAP ASE backup server. From there, the backup is restored on a staging server and the staging copy is then ingested into Delphix. See Delphix Architecture with SAP ASE (see page 1846) for more information.

When linking a dSource from an SAP ASE source database, Delphix offers several different methods of capturing backup information:

- ASE Managed Backups, where the SAP ASE source database schedules and initiates backups. This method supports various backup types which include:
  - Full backups
  - Transaction log backups (with LogSync disabled)
  - Transaction log backups (with LogSync enabled)

583 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/getting-started-managing-data-sources-and-syncing-data

- Delphix Managed Backups, where the Delphix Engine schedules and initiates the backups from the source database, and captures them.

9.10.7.2.1 ASE managed backups

Further contextual information on the various backup types (listed above):

- Full Backups - A snapshot will be created on the Delphix Timeflow for each Full backup.
- Transaction log backups (with LogSync disabled) - A snapshot will be created on the Delphix Timeflow for each transaction log backup.

> **Transaction logs**
> Transaction logs are not collected if:
> - a) there are gaps in the sequence of log backups (a break in the “log chain”).
> - b) the available log backups do not include any changes since the last successful Delphix snapshot.

- Transaction log backups (with LogSync enabled) - A snapshot will be created on the Delphix Timeflow for each transaction log backup. In addition, point-in-time provisioning will be an available option if you would like to provision from any point in between snapshots.

> **Log files**
> Log files consume additional space on the Delphix Engine and are managed according to the defined retention policy for logs.

9.10.7.2.2 Delphix managed backups

When the checkbox for Delphix Managed Backups is selected, the Delphix Engine will initiate a full backup of the source database for the initial load of the dSource. Thereafter, the Delphix Engine will initiate full backups of the source database using the schedule specified by the selected SnapSync Policy. If you select the None policy, the Delphix Engine will not automatically initiate a full backup, but you can initiate them manually using the snapshot (camera) icon.

Delphix looks for the backup files on the staging host (unless a "remote" backup server is used in which case, the remote host is used which is often the source environment)

9.10.7.2.3 Procedure

1. Login to the **Delphix Management** application.
2. Navigate to **Manage > Datasets.**

3. Click the plus icon and select **Add dSource**.
   Note:
   Delphix looks for the backup files on the staging host (unless a "remote" backup server is used in which case, the remote host is used which is often the source environment.
4. In the **Add dSource** wizard, select the source environment with the correct environment-based user.
5. Enter your login credentials for the source database and click **Next**.
6. Enter a name and select a group for your dSource. Adding a dSource to a dataset group lets you set Delphix Domain user permissions for that database and its objects, such as snapshots. See the topics under Users and Groups (see page 674) for more information.
7. Select the **Data Management** settings needed, as described in Data Management Settings for ASE Data Sources (see page 1921). [584]
8. From the **Data Management** tab under the **Initial Load** option, select and enter any **additional settings** needed. There are three options for the initial load of the dSource:
   a. If the source ASE instance resides on the same server as the staging ASE instance, the staging database's NFS mounted "temp" directory will be present for the source database to write to in response to the "DUMP DATABASE" command.
   b. If the source and staging ASE instances are configured to allow remote access to the backup servers and the dSource is linked using the Remote Server option as described below. Delphix will then issue the "DUMP DATABASE" command and append the "AT <staging_backup_server_name>" clause so that the dump is written to the staging backup server.
   c. (Recommended) **New Full Backup** - Lets Delphix create a new full backup file and load it. Note - that when Delphix creates the backup, it is moved to Delphix's NFS-mounted storage located on the stage host rather than. The backup will be located in the "temp" directory and will be deleted once the Delphix Engine has restored the backup and created a dSource from the restored staging database. This means that this option will work under two scenarios:
   d. Most Recent **Existing Full Backu**p – Find the most recent existing full backup file in the Backup Location and load it.
      Note:
      If Dump History is not active on the Source Database: Choosing this option can delay completion of the dSource link as Delphix attempts to find and catalog every single backup listed in the source database's backup server log file.
   e. Specific **Existing Full Backu**p – Specify which backup files in the Backup Location you want to load. Choosing this option is much faster because Delphix will skip directly to loading the desired backup and only start to search for and catalog backups in the background after the linking of the source database has completed.
      Note:
      When using a dump taken with the deprecated compression syntax, select the Specific Existing Full Backup option for Initial Load and, for each stripe, type compress::<file name> into the text box.

584 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/data-management-settings-for-sap-ase-dsources

9. Select the Staging environment and ASE instance that will be used to manage the staging database used for validated sync of the dSource.
10. Select any policies for the new dSource.
11. Click **Next**, then specify any pre-hook and post-hook scripts.
12. Review the dSource Configuration and Data Management information, and click **Submit** to begin provisioning the VDB.

9.10.7.3 Linking data sources with encrypted SAP ASE database

This topic describes the behavior of the Delphix Engine when linking to a dSource based on an encrypted SAP ASE database.

Beginning with version 16.0, SAP ASE supports the Database Encryption feature. You can link the encrypted source databases to a dSource by following the basic procedure described in Linking Data Sources with SAP ASE (see page 1918) but the user needs to ensure that the same key with the same name is present on the staging instance as it was on the source instance. The keys should be imported from the source instance. Refer to the configuration steps 4 and 5 of Delphix Implementation of Database Encryption (see page 1850) to export and import the master and encryption key.

- If a user links a database when it is unencrypted and then encrypts it, the subsequent syncs will fail.
- If a user links an encrypted database and then decrypts the source database, the subsequent syncs will fail.
- If a user changes the encryption key, the subsequent syncs will fail.

In the case of the above failures, the user will be prompted to apply a full backup with the “Drop and Recreate Devices” option enabled (for once), and subsequent snap and validate sync will run without any failures. See Working with Snapshots (see page 1928)

Exceptions with suggested actions are raised in the following scenarios:

- If there has been a change in the encryption status of a source database since it was last synced or the key has changed.
- If the required encryption key is not present on the staging instance during the linking.

9.10.7.4 Data management settings for SAP ASE dSources

Each dSource has its own data management settings, which can be configured during the linking workflow as well as on the configuration page for that dSource. You can configure data management settings to improve overall performance and meet your requirements.

The following settings are available for SAP ASE data sources:

Setting	Explanation
Backup Path	The directory where Delphix will search for backups for data ingestion. Delphix recursively searches this location, so the database backups or transaction logs can reside in any subdirectories below the path entered.
Validated Sync Mode	ASE Validated Sync is either enabled or disabled. Delphix stays in sync with source databases by monitoring the ASE backup server log or dump history file (depending on the dSource configuration). When it sees a new database dump or transaction log has been created, it attempts to load it into the ASE staging instance. • Enabled - Delphix automatically ingests either transaction logs or full database dumps depending on the source database's configuration. • Disabled - Delphix does not actively monitor the source database's backup server log or dump history file and does not automatically ingest backups.
Dump History	Delphix Validated Sync will leverage Dump History files to locate and ingest backups. This setting is recommended to improve the data ingestion and syncing processes.
LogSync	When enabled, Delphix copies the transaction logs to Delphix storage to allow precise point-in-time provisioning. **Note**: If Dump History is not enabled, due to SAP ASE CR 800569, Delphix can only support transaction logs generated in intervals greater than one minute apart for SAP ASE versions 16.0 SP02 through SAP ASE 16.0 SP02 PL04. This ASE bug inadvertently removed the second and millisecond precision from the dump header sequence dates preventing Delphix from knowing what order to apply the transaction logs when there are multiple transaction logs dumped within the same minute.
Source of Production Backup	You may select a staging or remote server as the location for the Delphix Engine to ingest backups. **Note** - Selecting a staging server is the most common application for ingesting source-based backups. One major reason for this is the ability to use Validated Sync to coordinate backup ingestion.
Dump Password	SAP ASE has an optional field for database dumps (backups) to prevent unauthorized loads. If enabled, Delphix will need to store the dump password in order to restore the SAP ASE backup for ingestion. Delphix also provides the option to propagate password protection to established dSources in the case of an active password on the originating database dump or transaction log dump files. Setting this option causes Delphix to add the "WITH passwd=" clause to the "LOAD" commands.

9.10.7.4.1 Procedure

1. Enter the **Backup location**. This is the directory where the database backups are stored. Delphix recursively searches this location, so the database backups or transaction logs can reside in any subdirectories below the path entered.
2. Select the **Staging environment** and ASE instance name.
3. Enable or disable Validated Sync Mode. Validated Sync Mode (also known as ValidatedSync) is a background process that looks for new full or transaction log backups either by monitoring ASE Backup Server's log file or Dump History file for a new database or transaction log dumps. When Delphix detects a new dump is available it attempts to load it into the staging ASE database.
4. Enable or disable **Use dump history**. If Dump History is enabled, Delphix uses the ASE Dump History file to find information about backups being performed on the source database and to find the next backup to be used for Validated Sync.
5. Enable or disable **LogSync**. LogSync copies the transaction logs to Delphix storage which enables provisioning VDBs from a specific point-in-time in rather than just a particular backup

   Note:
   LogSync support limitations

   [If Dump History is not active on the Source Database] Due to ASE CR 800569, Delphix can only support transaction logs generated in intervals greater than one minute apart in ASE versions 16.0 SP02 through ASE 16.0 SP02 PL04. This ASE bug inadvertently removed the second and millisecond precision from the dump header sequence dates preventing Delphix from knowing what order to apply the transaction logs in when there are multiple transaction logs dumped within the same minute.
6. Select **Backup location type**.
7. Click **Advanced**to edit Source of Production Dump, External Data Directory, Retention policies, or Dump Password.
   Note:
   External Data Directory

   The External Data Directory feature is not currently used with ASE dSources and is targeted for removal in a future release of Delphix.

   Remote Server should be selected when database dumps cannot be found on the Staging Environment. This option can be used with any of the initial load selections (New Full Backup, Most Recent Existing Full Backup or Specific Existing Full Backup). If selected, fill out additional settings as needed:
   a. Enter the Remote Server Name. This is the name of the backup server used when the dump was created.
   b. Select the Remote Host and Remote User that the backup server is located on.
   c. As noted, the interfaces file on both the staging and remote environments must be modified to point at each other's backup servers.

d. The Create Dump Password sets a dump password for the dSource. Select this only if the dump password option was used to create a password on the database dump or transaction log dump files. Setting this option causes Delphix to add the "WITH passwd=" clause to the "LOAD" commands.

9.10.7.5 Upgrading a SAP ASE dSource

This topic describes how to upgrade SAP ASE dSources.

9.10.7.5.1 Prerequisites

- The staging SAP ASE instance must be the same version as the SAP ASE instance hosting the upgraded source database.

9.10.7.5.2 Procedure

1. Login to the **Delphix Management** application.
2. In the top menu bar, click **Manage** and select **Datasets**.
3. Select the **dSource** to be upgraded.
4. Disable the dSource.
   a. From the **Actions (...) menu** select **Disable**.
   b. In the Disable dialog select **Disable**.
5. Upgrade the source and staging ASE instances (Delphix requires the source and staging ASE instances to be at the same major release level).
6. Click the **Refresh** icon (under the **Manage > Environments** menu) to refresh the Delphix environment hosting the source and staging ASE instances so that Delphix registers the new version of ASE.
7. Enable the dSource.
   a. From the **Actions (...) menu** select **Enable**.
   b. In the Enable dialog select **Enable**.

9.10.7.5.3 Additional steps

If Delphix was unable to discover the SAP ASE instances automatically using Manual Discovery, you will need to manually update the version by completing the following procedure, CLI Cookbook: Configuring SAP ASE Manual Discovery (see page 2539)

9.10.7.6 Detaching and Re-attaching SAP ASE dSources

This topic describes how to detach dSources and re-attach them to a different source database.

Each dSource contains metadata that associates it with the source database, as well as the data it has ingested from the source database in the form of snapshots up to that point. It is possible to detach, or unlink, a dSource from its source database. This breaks the association with the source database without

affecting the data within the Delphix Engine. Detached dSources and their source databases have these properties:

- A detached dSources can still be used to provision a virtual database (VDB).
- You can re-link the source database as a different dSource.
- Any child VDBs that were provisioned from this dSource will only be able to be refreshed from the most recent snapshot available on the dSource.
- If you need a VDB from a newer snapshot, you would need to provision a new VDB. Once you have provisioned the new VDB you can delete the old VDBs provisioned from this dSource. You can delete the old dSource when it is no longer needed.

9.10.7.6.1 Detaching a dSource

1. Login to the **Delphix Management** application as a user with **OWNER** privileges on the dSource, group, or domain.
2. Click **Manage**.
3. Select **Datasets**.
4. Select the **Dataset** you want to unlink.
5. From the **Actions** menu (...) select **Unlink dSource.** A warning message will appear.
6. Click **Unlink** to confirm. The status of the dSource will show as Detached.

9.10.7.6.2 Rebuilding source databases and using VDBs

In situations where you want to rebuild a source database but retain the existing dSource, you will need to detach the original dSource and create a new one from the rebuilt data source.

1. Detach the dSource as described in the procedure on this page.
2. You cannot attach a dSource with the same name as a dSource that is already attached. If you intend to give the new dSource the same name as the original one, rename the detached dSource.
    a. At the top of the **Configuration** tab, next to the dSource's name, click the **Edit** (pencil) icon.
    b. After renaming the dSource, click the **checkmark**.
3. Create the new dSource from the rebuilt database.

You will now be able to provision VDBs from both the detached dSource and the newly created one, but the detached dSource will only represent the state of the source database prior to being detached.

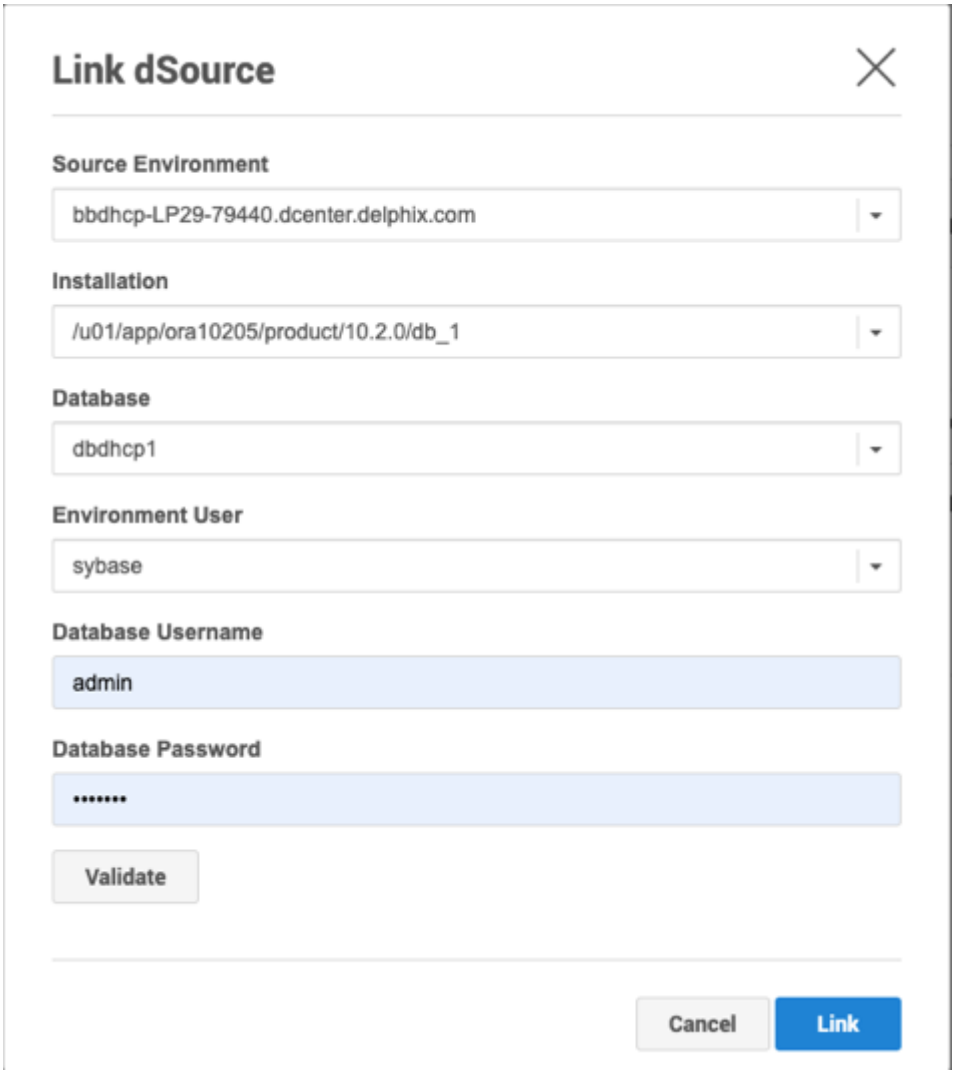
9.10.7.6.3 Attaching a previously detached dSource

The attach operation is supported via the UI. You can only re-attach databases that represent the same physical database.

Attaching a dsource via the UI:

1. Login to the **Delphix Management** application.

2. Click **Manage**.
3. Select **Datasets**.
4. Select the **Dataset** you want to link.
5. From the **Actions** menu (...) select **Link dSource.**
6. In the Link dSource window click **Link** to confirm.



9.10.7.7 Deleting an SAP ASE dSource

This topic describes how to delete an SAP ASE dSource.

Deleting a dSource will delete the dSource metadata for a particular source database, along with all snapshots, logs, and policies stored in Delphix. This is a permanent operation, and re-attaching the dSource will require a new linking operation (see page 1918).[585] If a dSource is deleted, it does not affect your source database.

585 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/linking-data-sources-with-sap-ase

If you wish to temporarily disable your dSource without deleting it, you can follow the steps to Enabling and Disabling SAP ASE dSources (see page 1927) instead.

9.10.7.7.1 Prerequisites

You cannot delete a dSource that has dependent virtual databases (VDBs). Before deleting a dSource, make sure that you have deleted all dependent VDBs as described in Deleting a VDB.

9.10.7.7.2 Procedure

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Click **Datasets**.
4. In the **Datasets** list, select the **dSource** you want to delete.
5. From the **Actions** menu (...) select **Delete.**
6. Click **Delete** to confirm.

Deleting a dSource will also delete all snapshots, logs, and descendant VDB refresh policies for that dSource. You cannot undo the deletion.

9.10.7.8 Enabling and disabling SAP ASE dSources

This topic describes how to enable and disable dSources when certain operations against the source database must occur outside of Delphix.

Some operations, such as restoring the source database from a backup, will require that the dSource be temporarily disabled. Disabling a dSource turns off communication between it and the source database, but it does not tear down the configuration that enables communication and data updating to take place. When a disabled dSource is later enabled, it will resume communication and incremental data updates from the source database according to the original policies and data management configurations that you set.

Disabling a dSource is also a prerequisite for several other operations, such as database migration and upgrading the dSource metadata after upgrade of the associated data source.

9.10.7.8.1 Procedure

Disabling a dSource will stop further operations on the Delphix Engine related to the dSource.

1. Login to the Delphix Management application.
2. Click **Manage**.
3. Select **Datasets**.
4. Select the **dSource** you want to disable.
5. In the upper right-hand corner, from the **Actions** menu (...) select **Disable.**
6. In the Disable dialog select **Disable**.

When you are ready to enable the dSource again, from the Actions menu (...) select **Enable**, and the dSource will continue to function as it did previously.

9.10.7.9 Working with SAP ASE snapshots

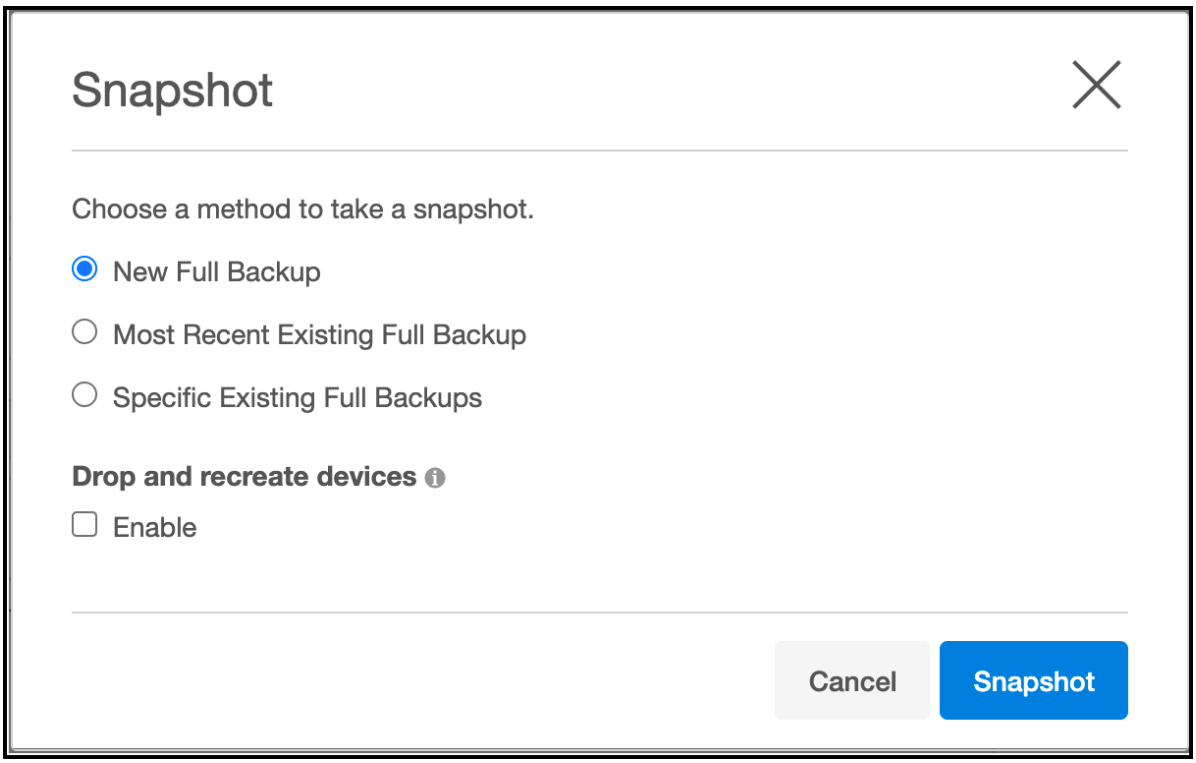
Taking a snapshot creates a new snapshot entry in the dSource's Timeflow.

Snapshot

This section lists the steps to take a snapshot and delete the same.

Login to the **Delphix Management** application.

1. Click **Manage** and select **Datasets** from the dropdown list.
2. Select the dSource you want to Snapshot.
3. Click the **Camera** icon.
4. Select the method to take the snapshot. See Quick Start Guide for SAP ASE (see page 1855) for information on Delphix Managed Backups.
   You can optionally select the **Enable** checkbox under **Drop and recreate devices**. This option can be used if the previous sync failed because of device mapping issues. When the sync operation is performed with this option set to true, Delphix Engine will not try to re-map the existing devices; instead, it will delete them and create new devices on the staging database. This option should only be used if the previous sync had failed due to device remapping, as this will increase disk usage.



5. From the Snapshot dialog, click **Snapshot**.

6. Under the **Timeflow** tab, choose an option to view the snapshot and verify the snapshot you just created.
7. To delete, select a snapshot you want to delete, and from the Actions menu (...) select **Delete Snapshot**.
8. Under the **Timeflow** tab, choose an option to view the snapshot and verify the snapshot you just deleted.

9.10.8 Provisioning and managing VDBs from SAP ASE

Virtual databases are a key data management concept for Delphix. In order to create or provision a virtual database, you will need a linked dSource from a source host and a compatible target environment.

From a dSource, you can select a snapshot or point in time to create a VDB. SQL Server VDBs each have their own configuration settings as described in Configuration Settings for ASE Virtual Databases below.

This section covers the following topics:

- Overview of provisioning SAP ASE virtual databases (see page 1929)
- Provisioning an SAP ASE VDB (see page 1931)
- Provisioning a VDB from an encrypted SAP ASE database (see page 1932)
- Provisioning an SAP ASE VDB from a replicated VDB or dSource (see page 1933)
- Resizing an SAP ASE VDB (see page 1934)
- V2P with an SAP ASE dSource or VDB (see page 1939)
- Additional SAP ASE VDB topics (see page 1946)

9.10.8.1 Overview of provisioning SAP ASE virtual databases

Virtual databases are a key data management concept for Delphix. In order to create or provision a virtual database, you will need a linked dSource from a source host and a compatible target environment, as described in the overview for Managing Environments and Hosts[586] and Requirements for SAP ASE Environments and Databases (see page 1884).[587]

From a dSource, you can select a snapshot or point in time to create a VDB. SQL Server VDBs each have their own configuration settings as described in Configuration Settings for ASE Virtual Databases below.

9.10.8.1.1 Procedure

1. In the Datasets panel on the left-hand side, click the group containing the dSource or VDB from which you want to provision.
2. Click the **Timeflow** tab.
3. Select a snapshot or open LogSync timeline to provision by a specific log or point in time.

586 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/6295693/Managing+environments+and+hosts
587 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/requirements-for-sap-ase-environments-and-databases

- Find more detail about initial provisioning options in the section 'Provisioning by Snapshot or LogSync' below.

4. Click to open the Provision VDB wizard, and select a compatible Target Environment for the new ASE VDB
5. Review the information presented for Target Configuration and edit as necessary.
6. Select a Snapshot Policy for the VDB.
7. (Optional) - Selective Data Distribution - After policies, there is a masking option.
8. Enter any operations that should be run in the Hooks page. These scripts can be managed after provision in the VDB's configuration page.

When provisioning starts, you can review the progress of the job by selecting the VDB and clicking on the Status tab, or by selecting Manage/Dashboards and viewing the Job History panel. Alternatively, you could see this in the Actions Sidebar. When provisioning is complete, the VDB will be included in the group you designated and listed in the Datasets panel. If you select the VDB in the Datasets panel and click the Configuration tab, you can view information about the database and its Data Management settings.

9.10.8.1.1.1 Provisioning by snapshot or LogSync

When provisioning by Snapshot, you can provision to the start of any particular snapshot by time.

Provisioning by snapshot/time	Description
Provisioning By Snapshot	You can provision by using a Snapshot. In that case, a new VDB will be provisioned to the database state as of the Snapshot.
Provision by Time	If you have enabled Log Sync, you can provision a new database to a point in time. You can select a snapshot and then using time entry fields, specify a Point in Time. Delphix will use the selected snapshot to restore the VDB and use the log files to roll forward the VDB to the selected time.

9.10.8.1.2 Configuration settings for ASE virtual databases

Each VDB has its own data management settings, found during the provisioning workflow as well as in the configuration page for that VDB. When you create a SAP ASE VDB, Delphix copies most configuration settings from the dSource and uses them to create the VDB. However, you can customize these with the following settings:

Setting	Explanation
Recovery model	The current recovery model of the source database. This field will auto-populate with information from the dSource.

Setting	Explanation
Auto VDB restart	Enabling this option will automatically restart this VDB whenever its target host is rebooted.

9.10.8.1.3 Automatic VDB restart on target server sfter reboot

The Delphix platform now automatically detects whether a target server has been rebooted, and proactively restarts any VBD on that server that was previously up and running. This is independent of the data platform. It is done as if you realized a target server was restarted and issued a start command from the Delphix platform. This feature is compatible with Self-Service ordering dependencies and is limited to non-clustered VDBs.

To enable automatic restart, complete the following steps:

- When provisioning a new VDB in the VDB Provisioning wizard, check the **Auto VDB restart** box.

Once the VDB has been provisioned, you will be able to turn **Automatic VDB restart** on.

1. In the **Datasets** panel, select the VDB.
2. Select the **Configuration** tab.
3. Select **Source** sub-tab.
4. Select **Database edit**.

9.10.8.2 Provisioning an SAP ASE VDB

This topic describes how to provision a virtual database (VDB) from an SAP ASE dSource.

9.10.8.2.1 Prerequisites

- You must have already linked a dSource from a source database, as described in Linking Data Sources with SAP ASE (see page 1918), or have already created a VDB from which you want to provision another VDB.
- You must have already set up target environments as described in Adding an SAP ASE Environment (see page 1905).[588]
- Ensure that you have the required privileges on the target environment, as described in Requirements for ASE Environments and Databases (see page 1884).[589]

588 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/adding-an-sap-ase-environment
589 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/requirements-for-sap-ase-environments-and-databases

- If you are provisioning to a target environment that is different from the one in which you set up the staging database, you must make sure that the two environments have compatible operating systems, as described in Requirements for SAP ASE Target Hosts and Databases (see page 1884).[590] For more information on the staging database and the validated sync process, see Managing SAP ASE Environments Overview (see page 1904).[591]

9.10.8.2.2 Procedure

1. Login to the **Delphix Management** application.
2. Click **Manage**
3. Select **Datasets**.
4. Select a **dSource**.
5. Select a **means of provisioning**.
6. Click **Provision**. The **Provision VDB** panel will open, and the **Instance** and **Database Name** fields will auto-populate with information from the dSource.
7. Select whether to enable **Truncate Log on Checkpoint** database option for the VDB.
8. Click **Next**.
9. Select a **Target Group** for the VDB. Click the green **Plus** icon to add a new group, if necessary.
10. Select a **Snapshot Policy** for the VDB. Click the green **Plus** icon to create a new policy, if necessary.
11. Click Auto VDB Restart to enable VDBs to be automatically restarted when staging/target host gets rebooted, if necessary.
12. Specify any **Hooks** to be used during the provisioning process. For more information, see SAP ASE Hook Operations (see page 1948).[592]
13. If your Delphix Engine system administrator has configured the Delphix Engine to communicate with an SMTP server, you will be able to specify one or more people to notify when the provisioning is done. You can choose other Delphix Engine users or enter email addresses.
14. Click **Submit**. When provisioning starts, the VDB will appear in the **Datasets** panel. Select the VDB and navigate to the **Status** tab to see the progress of the job. When provisioning is complete, more information can be seen on the **Configuration** tab.

9.10.8.3 Provisioning a VDB from an encrypted SAP ASE database

This topic describes how to provision a VDB from an encrypted database.

590 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/requirements-for-sap-ase-environments-and-databases
591 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/managing-sap-ase-environments-overview
592 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/sap-ase-hook-operations

The Delphix Engine supports provisioning from a dSource linked to a physical database that has been encrypted with SAP ASE Database Encryption, which can be used to encrypt columns or complete databases.

You can provision a VDB from a dSource linked to an encrypted database using the normal process to Provision an SAP ASE VDB (see page 1931) but the user needs to ensure that the same key with the same name is present on the target instance as it was on the staging instance because we mount the database onto the target that was created on the staging instance. The keys should be imported from the source instance. Refer to configuration steps 4 and 5 of Delphix Implementation of Database Encryption (see page 1850) to export and import the master and encryption key.

Exception with suggested actions is raised in the following scenario:

- If the required encryption key is not present on the target instance during the provisioning.

9.10.8.4 Provisioning an SAP ASE VDB from a replicated VDB or dSource

This topic describes how to provision from a replicated dSource or virtual database (VDB). The process for provisioning from replicated objects is the same as the typical VDB provisioning process, except that first you need to select the **replica** containing the replicated object.

9.10.8.4.1 Prerequisites

- You must have replicated a dSource or a VDB to the target host, as described in Replication Overview (see page 2368)
- You must have added a compatible target environment on the target host

9.10.8.4.2 Procedure

1. Login to the **Delphix Management** application for the target host.
2. Click **Manage**.
3. Select **Datasets**.
4. In the list of replicas, select the **replica** that contains the dSource or VDB you want to provision.
5. The provisioning process is now identical to the process for provisioning standard objects.

9.10.8.4.3 Post-requisites

Once the provisioning job has started, the user interface will automatically display the new VDB in the live system.

9.10.8.5 Resizing an SAP ASE VDB

When resizing ASE VDBs, we recommend:

- Ensure that the VDB's database devices reside on the Delphix storage, in the same directory as the database's other devices. Use the ASE "sp_helpdb" and "sp_helpdevice" commands to get details on the VDB's devices.
- If you need to add a new device, use the "skip_alloc=true" clause because the storage on Delphix is already zeroed (initialized). This should allow the "DISK INIT" command to complete almost instantaneously.
- If you add a new device to the database by issuing the ASE "DISK INIT" followed by the "ALTER DATABASE" commands, click the camera icon to take a new snapshot of the VDB.
  Note: It is important to take a snapshot after altering the VDB's device layout so that Delphix rewrites the database's "manifest" file to include the new database device.
- Delphix 6.x uses the new "ALLOW_DBID_MISMATCH,FIXDBID" clause to fix in dbid mismatches on versions of ASE that support these clauses. If you are on an older version of Delphix or ASE that does not support this clause, you may be required to run the DBCC CHECKALLOC() command on the VDB.

If you use the ASE **ALTER DATABASE** command to increase the amount of space allocated to a VDB, you may need to run the **DBCC CHECKALLOC()** command on the VDB. If the VDB is unmounted and remounted without this command being run on it, you may run into the following ASE CR which leaves the VDB in an unusable state:

SAP ASE change request number	Description
798271	The errors 14545, 14547 and 14519 will be raised by MOUNT DATABASE if the unmounted database had previously been mounted on a server where the database ID was used and a new database ID was assigned, and this database was extended without previously running DBCC CHECKALLOC(dbname, fix). Additionally, two new options have been added to the MOUNT command: WITH FIXDBID, to instruct the MOUNT command to fix any possible database ID mismatch, and WITH ALLOW_DBID_MISMATCH, to prevent that MOUNT DATABASE fails if the database has different database ID values in the allocation pages. The fix for this issue has been ported to **15.7 SP138** and **16.0 SP02 PL05**.

9.10.8.5.1 Determining the need to run DBCC CHECKALLOC

DBID mismatches are most likely to occur under the following circumstances:

- When provisioning VDBs to the same ASE instance hosting the staging database of the dSource the VDB was provisioned from.
- When provisioning multiple copies of the same VDB to the same ASE instance.

Delphix has found the following technique as a reliable way to determine whether or not you need to run DBCC CHECKALLOC() after increasing the amount of space allocated to the VDB.

In the following example, a VDB has been provisioned back to the same ASE instance where the staging database was being hosted. The staging database had a **dbid=11865**, so ASE automatically assigns the VDB the next lowest available dbid (**dbid=22** in this case).

1. Resize the database:

```
1> SELECT name, dbid FROM sysdatabases WHERE name = 'Vdb5'
2> go
 name                           dbid
 ------------------------------ ------
 Vdb5                                22

1> DISK RESIZE name="AAA4$ff367xUvd67P7II_db1_dev26", size="4M"
2> go

1> ALTER DATABASE Vdb5 on AAA4$ff367xUvd67P7II_db1_dev26="4M"
2> go
Extending database by 1024 pages (4.0 megabytes) on disk
AAA4$ff367xUvd67P7II_db1_dev26
Database Vdb5 which is currently offline has been altered from size 2816
 logical pages (2816 physical pages) to 3840 logical pages
```

2. Run the following queries to determine of there is a dbid mismatch on the pages of the VDB. **DBCC PAGE**() requires the **sybase_ts_role** role.

```
1> -- Run a query to generate DBCC commands:
1> SELECT 'dbcc page (Vdb5,' + convert(varchar(5), lstart) + ', 0, 0)' FROM
master..sysusages WHERE dbid=db_id('Vdb5') AND segmap != 0
2> go

 ----------------------------
 dbcc page (Vdb5,0, 0, 0)
 dbcc page (Vdb5,1536, 0, 0)
 dbcc page (Vdb5,1792, 0, 0)
 dbcc page (Vdb5,2816, 0, 0)

1> -- Turn on DBCC traceflag 3604 to direct output to stdout
2> dbcc traceon(3604)
3> GO

1> dbcc page (Vdb5,0, 0, 0)
2> dbcc page (Vdb5,1536, 0, 0)
3> dbcc page (Vdb5,1792, 0, 0)
4> dbcc page (Vdb5,2816, 0, 0)
5> go | grep dbid
    bmass_next=0x0 bmass_prev=0x0 bdbid=22
```

```
pageno=0 dealloc_count=102 allocnode=0 ptnid=99 allocation_page dbid=11865
 timestamp=0000 00002730, segmap=0x00000003 (0x00000002
    bmass_next=0x0 bmass_prev=0x0 bdbid=22
pageno=1536 dealloc_count=0 allocnode=0 ptnid=99 allocation_page dbid=11865
 timestamp=0000 00000001, segmap=0x00000004 (0x00000004
    bmass_next=0x0 bmass_prev=0x0 bdbid=22
pageno=1792 dealloc_count=0 allocnode=0 ptnid=99 allocation_page dbid=11865
 timestamp=0000 00000001, segmap=0x00000003 (0x00000002
    bmass_next=0x0 bmass_prev=0x0 bdbid=22
pageno=2816 dealloc_count=0 allocnode=0 ptnid=99 allocation_page dbid=22
 timestamp=0000 00000001, segmap=0x00000003 (0x00000002
```

Notice that three pages (0, 1536 and 1792) have the staging database's dbid (11865) and one page (2816) has the correct dbid (22) that matches the dbid of 22 stored in sysdatabases.

If all the dbid's match the dbid in sysdatabases, there is no need to run DBCC CHECKALLOC().

This means that DBCC CHECKALLOC() must be run on the VDB to correct the dbid mismatch:

```
1> dbcc checkalloc(Vdb5)
2> go
...etc...
11 allocation pages have been corrected to match database ID 22.
```

Note: In the code block above 11 pages were fixed.

Now if you run the **DBCC PAGE()** query again you can see that all of the dbid's match:

```
1> dbcc page (Vdb5,0, 0, 0)
2> dbcc page (Vdb5,1536, 0, 0)
3> dbcc page (Vdb5,1792, 0, 0)
4> dbcc page (Vdb5,2816, 0, 0)
5> go | grep dbid
    bmass_next=0x0 bmass_prev=0x0 bdbid=22
pageno=0 dealloc_count=102 allocnode=0 ptnid=99 allocation_page dbid=22 timestamp=000
0 00002730, segmap=0x00000003 (0x00000002
    bmass_next=0x0 bmass_prev=0x0 bdbid=22
pageno=1536 dealloc_count=0 allocnode=0 ptnid=99 allocation_page dbid=22 timestamp=00
00 00000001, segmap=0x00000004 (0x00000004
    bmass_next=0x0 bmass_prev=0x0 bdbid=22
pageno=1792 dealloc_count=0 allocnode=0 ptnid=99 allocation_page dbid=22 timestamp=00
00 00000001, segmap=0x00000003 (0x00000002
    bmass_next=0x0 bmass_prev=0x0 bdbid=22
pageno=2816 dealloc_count=0 allocnode=0 ptnid=99 allocation_page dbid=22 timestamp=00
00 00000001, segmap=0x00000003 (0x00000002
```

9.10.8.5.2 Recovering from DBID mismatch errors

If a VDB containing mismatched dbids gets unmounted, it will likely get ASE errors 14545, 14547 and 14519 during the next attempt to **MOUNT** it and it will now be in an unusable state. Options for recovery include:

- For ASE versions that do not have the fix for ASE CR 798271, you may rewind the VDB to a snapshot taken prior to resizing the VDB. Any data changes made since this point in time will be lost.

On newer versions of ASE where ASE CR 798271 has been fixed, you can manually mount the VDB using the new **MOUNT** command options:

```
1> -- Get the device listing for the VDB from the manifest
2>  MOUNT DATABASE Vdb5 FROM '/export/home/sybase/toolkit/564dfe60-40b6-
aec9-26fb-3baeb4e7b23b-vdb-19/datafile/manifest' WITH LISTONLY
3> go
[database]
Vdb5
[device]
'/export/home/sybase/toolkit/564dfe60-40b6-aec9-26fb-3baeb4e7b23b-vdb-19/datafile/
dxnff367xUvd67P7II_db1_dev26' =
'AAA4$ff367xUvd67P7II_db1_dev26'
'/export/home/sybase/toolkit/564dfe60-40b6-aec9-26fb-3baeb4e7b23b-vdb-19/datafile/
dx9279VtfPAvfOm5xY_db1_dev27' =
'AAA5$279VtfPAvfOm5xY_db1_dev27'

1> -- Mount the VDB using the WITH ALLOW_DBID_MISMATCH clause
2> MOUNT DATABASE [Vdb5] AS [Vdb5] FROM '/export/home/sybase/toolkit/564dfe60-40b6-
aec9-26fb-3baeb4e7b23b-vdb-19/datafile/manifest'
3> WITH ALLOW_DBID_MISMATCH USING
4> '/export/home/sybase/toolkit/564dfe60-40b6-aec9-26fb-3baeb4e7b23b-vdb-19/datafile/
dxnff367xUvd67P7II_db1_dev26' = 'AAA4$ff367xUvd67P7II_db1_dev26',
5> '/export/home/sybase/toolkit/564dfe60-40b6-aec9-26fb-3baeb4e7b23b-vdb-19/datafile/
dx9279VtfPAvfOm5xY_db1_dev27' = 'AAA5$279VtfPAvfOm5xY_db1_dev27'
6> go
The physical device '/export/home/sybase/toolkit/564dfe60-40b6-
aec9-26fb-3baeb4e7b23b-vdb-19/datafile/dxnff367xUvd67P7II_db1_dev26'
has been automatically assigned the logical device name
'AAA6$ff367xUvd67P7II_db1_dev26'.
The physical device '/export/home/sybase/toolkit/564dfe60-40b6-
aec9-26fb-3baeb4e7b23b-vdb-19/datafile/dx9279VtfPAvfOm5xY_db1_dev27'
has been automatically assigned the logical device name
'AAA7$279VtfPAvfOm5xY_db1_dev27'.
Started estimating recovery log boundaries for database 'Vdb5'.
Database 'Vdb5', checkpoint=(1543, 13), first=(1543, 13), last=(1543, 13).
Completed estimating recovery log boundaries for database 'Vdb5'.
Started ANALYSIS pass for database 'Vdb5'.
Completed ANALYSIS pass for database 'Vdb5'.
Started REDO pass for database 'Vdb5'. The total number of log records to process is
1.
Completed REDO pass for database 'Vdb5'.
MOUNT DATABASE: Completed recovery of mounted database 'Vdb5'.
MOUNT DATABASE: A new database id was required for database 'Vdb5' in order to mount
it. Execute DBCC CHECKALLOC(Vdb5, fixdbid) to
correct it.
```

```
1> DBCC CHECKALLOC(Vdb5, fixdbid)
2> go
Checking Vdb5: Logical pagesize is 4096 bytes
Total (# alloc pages = 15, # of alloc pages modified = 15).
DBCC execution completed. If DBCC printed error messages, contact a user with System
Administrator (SA) role.

1> -- Confirm pages all have the correct dbid
2> SELECT 'dbcc page (Vdb5,' + convert(varchar(5), lstart) + ', 0, 0)' FROM
master..sysusages WHERE dbid=db_id('Vdb5') AND segmap != 0
3> go

 ----------------------------
 dbcc page (Vdb5,0, 0, 0)
 dbcc page (Vdb5,1536, 0, 0)
 dbcc page (Vdb5,1792, 0, 0)
 dbcc page (Vdb5,2816, 0, 0)

1> dbcc traceon(3604)
2> dbcc page (Vdb5,0, 0, 0)
3> dbcc page (Vdb5,1536, 0, 0)
4> dbcc page (Vdb5,1792, 0, 0)
5> dbcc page (Vdb5,2816, 0, 0)
6> go | grep dbid
    bmass_next=0x0 bmass_prev=0x0 bdbid=23
pageno=0 dealloc_count=102 allocnode=0 ptnid=99 allocation_page dbid=23 timestamp=000
0 00002730, segmap=0x00000003 (0x00000002
    bmass_next=0x0 bmass_prev=0x0 bdbid=23
pageno=1536 dealloc_count=0 allocnode=0 ptnid=99 allocation_page dbid=23 timestamp=00
00 00000001, segmap=0x00000004 (0x00000004
    bmass_next=0x0 bmass_prev=0x0 bdbid=23
pageno=1792 dealloc_count=0 allocnode=0 ptnid=99 allocation_page dbid=23 timestamp=00
00 00000001, segmap=0x00000003 (0x00000002
    bmass_next=0x0 bmass_prev=0x0 bdbid=23
pageno=2816 dealloc_count=0 allocnode=0 ptnid=99 allocation_page dbid=23 timestamp=00
00 00000001, segmap=0x00000003 (0x00000002
```

9.10.8.5.3 More information

VDBs are unmounted and remounted during operations such as:

- Disable/Enable of the VDB.
- Stop/Start of the VDB.
- Rebooting of the server hosting the VDB's ASE instance.

DBAs are unlikely to have found the need to run **DBCC CHECKALLOC()** on physical databases in the normal workflow of resizing a database unless the ASE **MOUNT/UNMOUNT** command is used. When you **MOUNT** a database on a server that already has the mounted database's dbid in use, ASE assigns the mounted database a new dbid and this gets reflected in the **sysdatabases** and **sysusages** tables in the master database and in the **dbtable** memory structure for the mounted database. However, the dbid in the dbinfo structure and allocation pages in the mounted database are not updated by mount. Doing so can take a long time on a big database. **DBCC CHECKALLOC()** will correct the values in all the allocation pages.

In Delphix version 5.1.3.0 and higher, Delphix will create databases using a high `dbid` (using the `"CREATE DATABASE with dbid="` clause). By using a high `dbid` (ASE supports a maximum of 32,767 dbids), it is unlikely that the `dbid` will be reused and thus Delphix will be able to avoid the errors (14545, 14547 and 14519). It attempts to assign unique **dbid** values for staging databases in the range 10000-30000.

The second part of this fix involves keeping the staging database mounted so that the **dbid** is pinned and no other database gets created and uses the staging database's **dbid**. The staging database will remain mounted except for a brief window where we unmount it to create a manifest file.

The final part of this fix utilizes the ASE **DBCC PAGE()** command to confirm the **dbid** on select pages from the sysusages table all match the database's actual **dbid**. If there is a mismatch found, Delphix will call **DBCC CHECKALLOC()** to fix the database. Delphix calls **DBCC PAGE()** in order to avoid calling **DBCC CHECKALLOC()** as the **CHECKALLOC()** routine can add significant time to database recovery (performance considerations for CHECKALLOC[593]).

9.10.8.6 V2P with an SAP ASE dSource or VDB

9.10.8.6.1 Requirements

This topic describes the procedure for exporting a virtual database (VDB) to a physical one, also known as V2P. Before performing the V2P operation, you must have created a database on the target instance into which you will load the exported data. It must be sufficiently large, and it must have been created with the **for load** SAP ASE option.

> The Delphix Engine will initiate a load command using the database specified. The V2P operation will overwrite any existing data in this database. After a V2P operation takes place, the target physical database is no longer in FOR LOAD state.

9.10.8.6.2 Procedure

1. Login to the **Delphix Management** application.
2. Select the **dSource** or **VDB** you want to export.
3. Select the **snapshot** of the dSource or VDB state you want to export.
4. Click **V2P**.
5. Select the **target environment**. The target environment should have been added to Delphix previously and should meet all target host requirements, see Requirements for SAP ASE Environments and Databases (see page 1884).[594]
6. Under **Installation**, select which **instance** that you want to export to.

593 http://infocenter.sybase.com/help/topic/com.sybase.infocenter.dc31644.1600/doc/html/san1371158591037.html
594 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/requirements-for-sap-ase-environments-and-databases

7. Enter the **Name** of the database on the target instance into which you want to load the exported data.
8. Select whether or not to **Run Recovery After V2P**. When this option is set, the Delphix Engine will online the database when the export is done.
9. Click **Next**.
10. Select whether you want to have an email sent to you when the export process completes.
11. Click **Submit**.

9.10.8.7 Additional SAP ASE VDB topics

Deleting an SAP ASE VDB (see page 1941)

Procedure Deleting a VDB is an unrecoverable operation. Proceed only if you want to permanently destroy the unique data that was created in the VDB. Login to the Delphix Management application. Click Manage . Select Datasets . Click th... (see page 1941)

Updated on : 25 May 2023

Enabling and disabling SAP ASE VDBs (see page 1941)

This topic describes how to enable and disable a virtual database (VDB). Disabling a VDB is a pre-requisite for procedures such as VDB migration or upgrade. Disabling a VDB removes all traces of it, including any configuration files, from the targe... (see page 1941)

Updated on : 25 May 2023

Migrating an SAP ASE VDB (see page 1942)

This topic describes how to migrate a virtual database (VDB) from one target environment to another. In certain situations, you may want to migrate a VDB to a new target environment. For example: When upgrading the host on which the VDB resides ... (see page 1942)

Updated on : 25 May 2023

Refreshing an SAP ASE VDB (see page 1943)

This topic describes how to manually refresh a virtual database (VDB). Refreshing a VDB will re-provision it from the dSource. As with the normal provisioning process, you can choose to refresh the VDB from a snapshot or a specific point in time. H... (see page 1943)

Updated on : 25 May 2023

Rewinding an SAP ASE VDB (see page 1945)

This topic describes the procedure for rewinding an SAP ASE virtual database (VDB). Rewinding a VDB rolls it back to a previous point in its TimeFlow and re-provisions the VDB. The VDB will no longer contain changes that were made after the rewind ... (see page 1945)

Updated on : 25 May 2023

Upgrading SAP ASE VDBs (see page 1946)

This topic describes how to upgrade an SAP ASE VDB to a higher version of SAP ASE instance. Procedure Login to the Delphix Management application. Click Manage . Select Datasets . Select the VDB to be upgraded. From the ... (see page 1946)

Updated on : 25 May 2023

9.10.8.7.1 Deleting an SAP ASE VDB

9.10.8.7.1.1 Procedure

Deleting a VDB is an unrecoverable operation. Proceed only if you want to permanently destroy the unique data that was created in the VDB.

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Datasets**.
4. Click the **VDB** that you want to delete.
5. From the Actions menu (...) select **Delete.**
6. If stopping or starting the VDB requires particular credentials for the target environment other than those of the default environment user:
   a. Check **Provide Privileged Credentials**.
   b. Enter the **username** and **password**.
   c. Click **Validate Credentials**.
7. Click **Delete** to confirm that you want to delete the VDB.

If the VDB was currently active, the Delphix Engine will shut it down, unmount all filesystems from the target environment, and finally delete the VDB itself.

9.10.8.7.2 Enabling and disabling SAP ASE VDBs

This topic describes how to enable and disable a virtual database (VDB).

Disabling a VDB is a pre-requisite for procedures such as VDB migration or upgrade. Disabling a VDB removes all traces of it, including any configuration files, from the target environment to which it was provisioned. When you later enable the VDB again, these configuration files are restored on the target environment.

9.10.8.7.2.1 Procedure

1. Click **Manage**.
2. Select **Datasets**.
3. Click the **VDB** you want to disable.
4. From the Actions menu (...) select **Disable**.
5. Click **Disable** to acknowledge the warning.

When you are ready to enable the VDB again, from the Actions menu (...) select **Enable**, and the VDB will continue to function as it did previously.

9.10.8.7.3 Migrating an SAP ASE VDB

This topic describes how to migrate a virtual database (VDB) from one target environment to another.

In certain situations, you may want to migrate a VDB to a new target environment. For example:

- When upgrading the host on which the VDB resides
- During a general data center migration
- To distribute the VDB load across target environments

This is easily accomplished by first disabling the database, then using the Migrate VDB feature to select a new target environment.

9.10.8.7.3.1 Prerequisites

- You must first disable the VDB before migrating it. Follow the steps outlined in Enabling and Disabling SAP ASE VDBs (see page 1941)
- You must have already set up a new target environment that is compatible with the VDB that you want to migrate.

9.10.8.7.3.2 Procedure

1. Login to your Delphix Management application.
2. Click **Manage**.
3. Select **Datasets**.
4. Click the **VDB** you want to migrate.
5. From the **Actions** menu (...) select **Disable**.
6. Click **Disable** to confirm.
7. From the **Actions** menu (...) select **Migrate.**
8. In the Migrate window select:
    a. Environment: the new **target environment** for the VDB

   b. User: select a user
   c. Installation: the **installation** where you want to migrate

9. Click Migrate to confirm your selections.
10. From the **Actions** menu (...) select **Enable**.
11. Click **Enable** to confirm.

Within a few minutes, your VDB will re-start in the new environment, and you can continue to work with it as you would with any other VDB.

9.10.8.7.4 Refreshing an SAP ASE VDB

This topic describes how to manually refresh a virtual database (VDB).

Refreshing a VDB will re-provision it from the dSource. As with the normal provisioning process, you can choose to refresh the VDB from a snapshot or a specific point in time. However, you should be aware that refreshing a VDB would delete any changes that have been made to it over time. When you refresh a VDB, you are essentially re-setting it to the state you select during the refresh process. You can refresh a VDB manually, as described in this topic, or you can set a VDB refresh policy, as described in Managing Policies[595]

> Although the VDB no longer contains the previous contents, the previous Snapshots and Timeflow still remain in Delphix and are accessible through the Command Line Interface (CLI).

9.10.8.7.4.1 Prerequisites

To refresh a VDB, you must have the following permissions:

- **PROVISIONER** permissions on the dSource associated with the VDB
- **PROVISIONER** permissions on the group that contains the VDB
- **Owner** permissions on the VDB itself
- **Data** is a role that allows DB_ROLLBACK, DB_REFRESH, READ_ACTION, DB_SYNC, JOB_CANCEL**.**
- **Read** is a role that allows the user to inspect objects via the READ_ACTION permission.

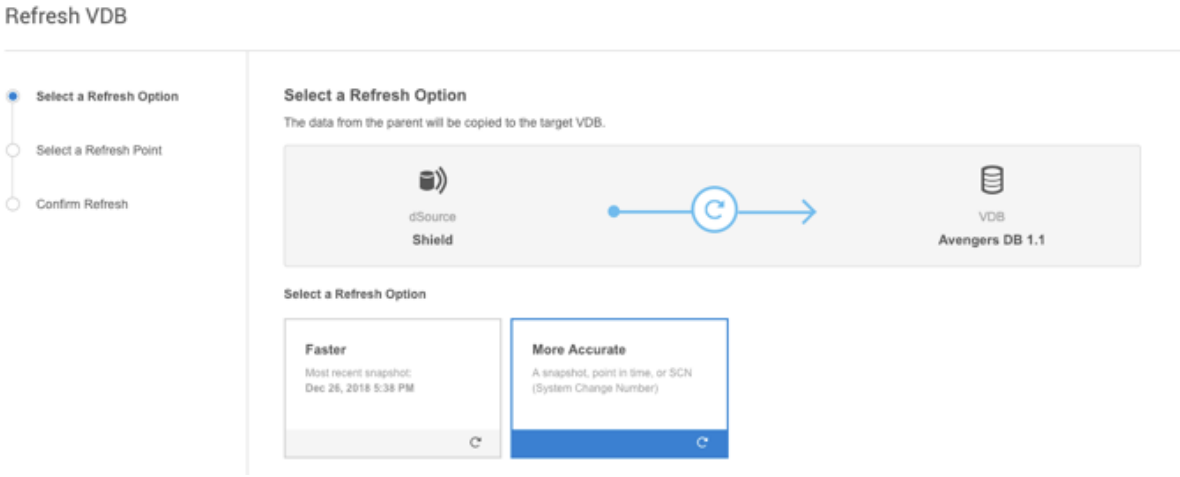
A user with admin credentials can perform a VDB Refresh on any VDB in the system.

9.10.8.7.4.2 Procedure

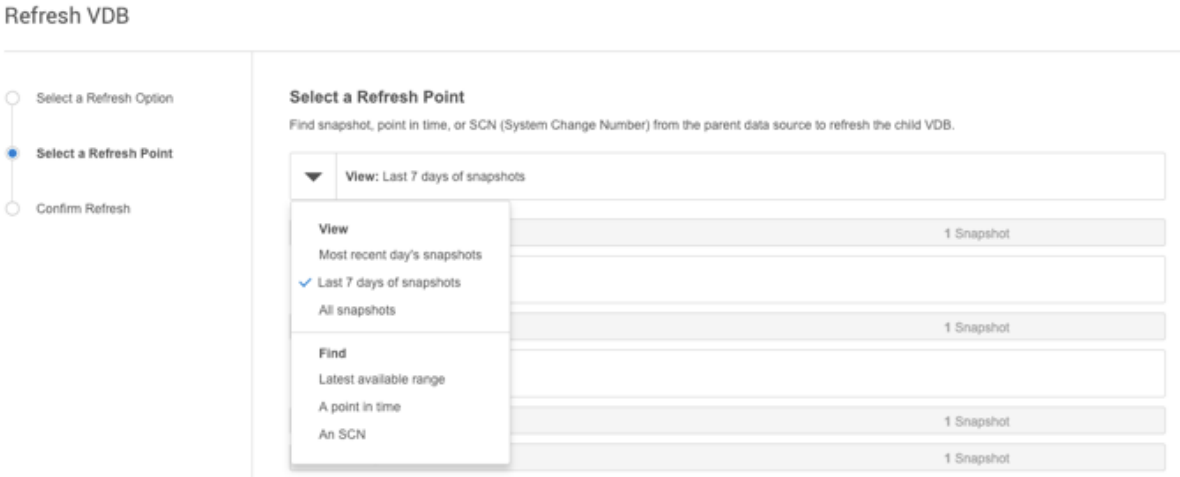
1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Datasets**.
4. Select the **VDB** you want to refresh.
5. Click the **Refresh VDB** button.

595 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/6783897/Managing+policies

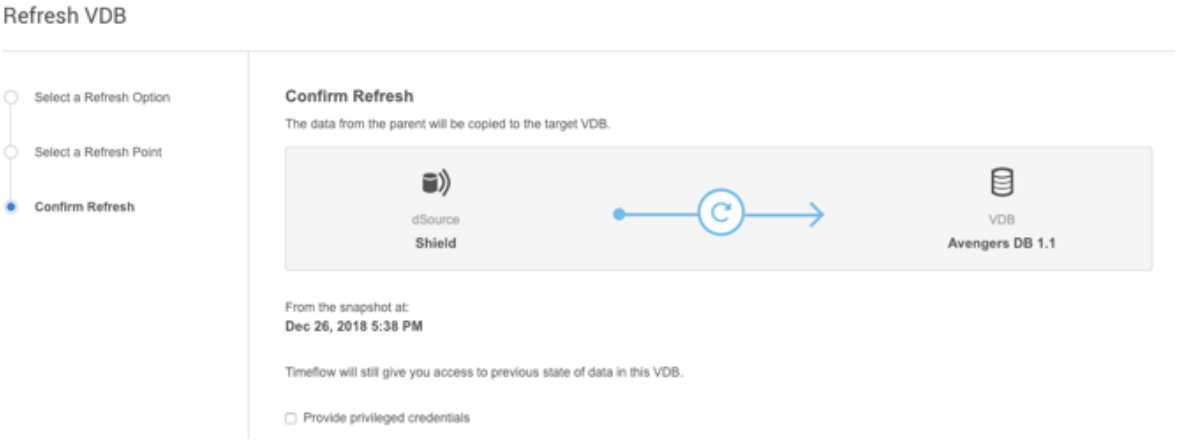
6. Select **More Accurate** and **Next**.



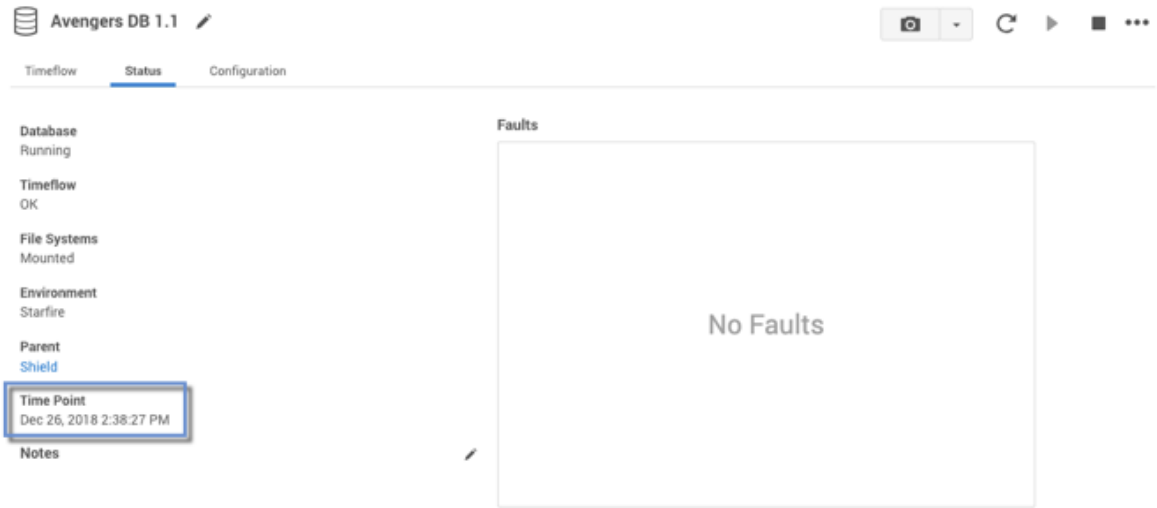
7. Select the desired **refresh point** snapshot or click the **down arrow** icon to choose the Latest available range, A point in time, or An SCN to refresh from.



8. Click **Next**.
9. Click **Submit** to confirm.



10. Click the **Actions**link to watch the progress of the refresh job.
11. To see when the VDB was last refreshed/provisioned, check the **Time Point** on the Status page



9.10.8.7.5 Rewinding an SAP ASE VDB

This topic describes the procedure for rewinding an SAP ASE virtual database (VDB).

Rewinding a VDB rolls it back to a previous point in its TimeFlow and re-provisions the VDB. The VDB will no longer contain changes that were made after the rewind point.

> Although the VDB no longer contains changes made after the rewind point, the rolled-over snapshots and TimeFlow still remains in Delphix and are accessible through the Command Line Interface (CLI). See the topic CLI Cookbook: Rolling Forward a VDB (see page 2660) for instructions on how to use these snapshots to refresh a VDB to one of its later states after it has been rewound.

9.10.8.7.5.1 Prerequisites

To rewind a VDB, you must have the following permission:

- **Owner** permissions on the VDB itself

You do NOT need owner permissions for the group that contains the VDB. A user with Delphix Admin credentials can perform a VDB rewind on any VDB in the system.

9.10.8.7.5.2 Procedure

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Datasets**.
4. Select the **VDB** you want to rewind.
5. Click the **TimeFlow** tab.
6. Select the rewind point as a snapshot or a point in time.
7. Click **Rewind**.
8. If you want to use login credentials on the target environment other than those associated with the environment user, click **Provide Privileged Credentials**.
9. Click **Yes** to confirm.

> You can use TimeFlow bookmarks as the rewind point when using the CLI. Bookmarks can be useful to:
> - Mark where to rewind to – before starting a batch job on a VDB, for example.
> - Provide a semantic point to revert back to, in case the chosen rewind point turns out to be incorrect.
>
> For a CLI example using a TimeFlow bookmark, see CLI Cookbook: Provisioning a VDB from a Timeflow Bookmark (see page 2639).[596]

9.10.8.7.6 Upgrading SAP ASE VDBs

This topic describes how to upgrade an SAP ASE VDB to a higher version of SAP ASE instance.

Procedure

1. Login to the **Delphix Management** application.

596 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/cli-cookbook-provisioning-a-vdb-from-a-timeflow-bookmark

2. Click **Manage**.
3. Select **Datasets**.
4. Select the **VDB** to be upgraded.
5. From the **Actions** menu (...) select **Disable**.
6. Click **Disable** to confirm.
7. Upgrade the ASE instance(s) hosting the VDBs.
8. Click the **Refresh** icon (under the **Manage > Environments** menu) to refresh the Delphix environment hosting the VDBs so that Delphix registers the new version of ASE.
9. Enable the VDB. The VDBs will be upgraded the first time they're enabled.
10. Repeat steps 4 to 9 for each VDB you want to upgrade.

9.10.9 Backup server best practices

- The backupserver log should have read permissions for the Delphix environment user
- Dump file names should be unique per source backup server.
- Dumps should be done to a local backupsever. Dumps done using the "at <backupserver>" syntax are not discovered and ingested/applied.
- Dumps of production databases that use the same backup server and same load backup path should be available on the same staging host that was pointed at during the lin
- If using a remote backup server as load location, make sure "interfaces" file changes are done to ensure staging host backupserver can communicate with the remote backup server.
- A user running backupserver or process owner of the backupserver on the staging host should have read access for all the dump files that belong to dSources that use this backupserver.

9.10.9.1 Recommendations/best practices

1. Name the dump files as unique as possible.
2. Delphix recursively searches for backup files in a given load backup path. So assign or choose the load backup path for a given dSource to be as specific and unique as possible. For e.g, if the load backup path is /dumps for dSource db1, but the database dumps are copied to /dumps/db1, choose /dumps/db1 instead to avoid duplicate files that can happen with other dSources or databases.
3. Avoid linking sources where it's backup location is a child directory of another dSource's backup location. Ideally, each dSource using the same staging host should have its backup path set to sibling directories for ease of search.
4. Avoid copying files (of any type) with the same name as backup files that are being ingested.
5. Avoid copying other dSource or database backups to the same load backup path of the dSource of interest. You can copy, as long as dump file names are unique.

6. Only use a remote backup server as the load location if dumps cannot be available on the staging host.
7. Avoid, if possible, using production backup server as load location and the backup files are on production host when configuring a dSource with remote backupserver.
8. DE by default assumes backup files are available on the staging host when searching for them. If the backup files are available on the staging host, avoid using the remote backupserver option on the link/data management wizard.
9. If validated sync reports any fault, look at other faults that were raised, resolve all of them as well along with the current fault.
10. When copying striped backups to staging, keep all stripes in the same location.
11. If possible, create separate database dumps and/or transaction logs by more than 1 minute apart so that the dump header name is unique.
12. The Delphix Engine, by default, looks for backups that happened within the last six months from the current date/time when a source database is linked for the first time
    a. If there is ever a need to ingest an older backup, use “Specific Backup” sync parameters and specify the location of the backup files to ingest.
    b. Please reach out to Delphix Support for help if there is a reason for concern regarding the 6 months window.
13. The Delphix Engine periodically purges discovered backup files from its internal catalog that are older than 6 months (default). If there is a need to ingest an old backup, use the “Specific Backup” option as there is no mechanism to rediscover old backups.

9.10.10 SAP ASE hook operations

9.10.10.1 Shell operations

9.10.10.1.1 RunCommand operation

The RunCommand operation runs a shell command on a Unix environment using whatever binary is available at `/bin/sh`. The environment user runs this shell command from their home directory. The Delphix Engine captures and logs all output from this command. If the script fails, the output is displayed in the Delphix Management application and command-line interface (CLI) to aid in debugging.

If successful, the shell command must exit with an exit code of `0`. All other exit codes will be treated as an operation failure.

9.10.10.1.1.1 Examples of RunCommand operations

You can input the full command contents into the RunCommand operation.

```
remove_dir="$DIRECTORY_TO_REMOVE_ENVIRONMENT_VARIABLE"

if test -d "$remove_dir"; then
    rm -rf "$remove_dir" || exit 1
fi

exit 0
```

If a script already exists on the remote environment and is executable by the environment user, the RunCommand operation can execute this script directly.

```
/opt/app/oracle/product/10.2.0.5/db_1/dbs/myscript.sh "$ARG_ENVIRONMENT_VARIABLE"
"second argument in double quotes" 'third argument in single quotes'
```

9.10.10.1.2 RunBash operation

The RunBash operation runs a Bash command on a Unix environment using a `bash` binary provided by the Delphix Engine, unless it's a Linux environment, in which case it uses the system's native bash binary. The environment user runs this Bash command from their home directory. The Delphix Engine captures and logs all output from this command. If the script fails, the output is displayed in the Delphix Management application and command-line interface (CLI) to aid in debugging.

If successful, the Bash command must exit with an exit code of `0`. All other exit codes will be treated as an operation failure.

9.10.10.1.2.1 Example of RunBash Operations

You can input the full command contents into the RunBash operation.

```
remove_dir="$DIRECTORY_TO_REMOVE_ENVIRONMENT_VARIABLE"

# Bashisms are safe here!
if [[ -d "$remove_dir" ]]; then
    rm -rf "$remove_dir" || exit 1
fi

exit 0
```

9.10.10.1.3 Shell operation tips

9.10.10.1.3.1 Using `nohup`

You can use the `nohup` command and process backgrounding from resource in order to "detach" a process from the Delphix Engine. However, if you use `nohup` and process backgrounding, you MUST redirect `stdout` and `stderr.`

Unless you explicitly tell the shell to redirect `stdout` and `stderr` in your command or script, the Delphix Engine will keep its connection to the remote environment open while the process is writing to either `stdout` or `stderr` . Redirection ensures that the Delphix Engine will see no more output and thus not block waiting for the process to finish.

For example, imagine having your `RunCommand` operation background a long-running Python process. Below are the bad and good ways to do this.

Bad Examples
• `nohup python file.py & # no redirection` • `nohup python file.py 2>&1 & # stdout is not redirected` • `nohup python file.py 1>/dev/null & # stderr is not redirected` • `nohup python file.py 2>/dev/null & # stdout is not redirected`

Good Examples
• `nohup python file.py 1>/dev/null 2>&1 & # both stdout and stderr redirected, Delphix Engine will not block`

9.10.10.2 Other operations

9.10.10.2.1 RunExpect operation

The RunExpect operation executes an Expect script on a Unix environment. The Expect utility provides a scripting language that makes it easy to automate interactions with programs which normally can only be used interactively, such as `ssh` . The Delphix Engine includes a platform-independent implementation of a subset of the full Expect functionality.

The script is run on the remote environment as the environment user from their home directory. The Delphix Engine captures and logs all output of the script. If the operation fails, the output is displayed in the Delphix Management application and CLI to aid in debugging.

If successful, the script must exit with an exit code of `0`. All other exit codes will be treated as an operation failure.

9.10.10.2.1.1 Example of a RunExpect operation

Start an `ssh` session while interactively providing the user's password.

```
spawn ssh user@delphix.com
expect {
    -re {Password: } {
        send "${env(PASSWORD_ENVIRONMENT_VARIABLE)}\n"
    }
    timeout {
        puts "Timed out waiting for password prompt."
        exit 1
    }
}
exit 0
```

9.10.10.3 SAP ASE environment variables

Operations that run user-provided scripts have access to environment variables. For operations associated with specific dSources or virtual databases (VDBs), the Delphix Engine will always set environment variables so that the user-provided operations can use them to access the dSource or VDB.

9.10.10.3.1 dSource environment variables

Environment variables	Description
`ASE_ENVUSER`	Environment username for the dSource
`ASE_DBUSER`	Database username for the dSource
`ASE_DATABASE`	Database name for the dSource
`ASE_INSTANCE`	SAP ASE Instance name for the dSource

Environment variables	Description
`ASE_PORT`	SAP ASE Instance port for the dSource

9.10.10.3.2 VDB environment variables

Environment variables	Description
`ASE_ENVUSER`	Environment username for the VDB
`ASE_DBUSER`	Database username for the VDB
`ASE_DATABASE`	Database name for the VDB
`ASE_INSTANCE`	SAP ASE Instance name for the VDB
`ASE_PORT`	SAP ASE Instance port for the VDB

9.10.10.3.3 Staging server environment variables

Environment variables	Description
`ASE_ENVUSER`	

9.10.11 Support for dump history file

9.10.11.1 Overview

Prior to ASE version 15.7 ESD#2, ASE did not provide any structured information about backup/restore operations performed on databases. Delphix relied on the backup server log file to find information about any backups performed using a particular backup server. The Delphix Engine parsed this log file, read each backup file to find which database the backup belonged to and restored the ones required for linked databases. As backup information is not very structured in this log file, getting backup history from this file resulted in the collection and parsing of unnecessary data. Starting with ASE 15.7 ESD#2, users can enable

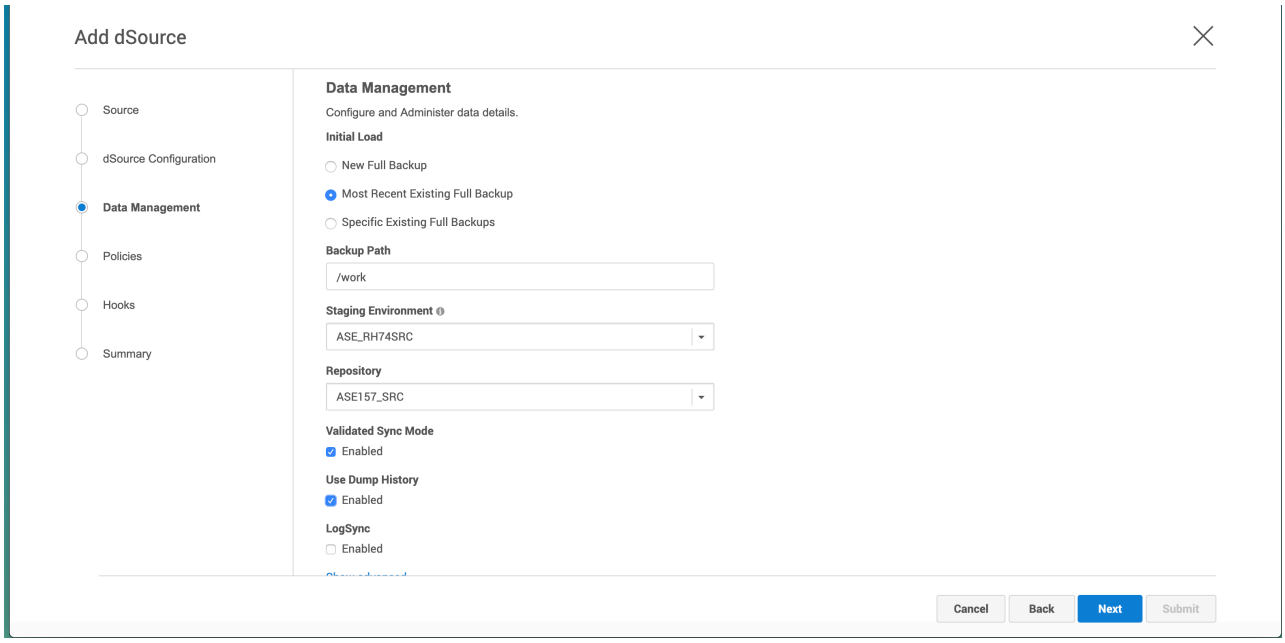
an option at the ASE instance level to record all backup and restore operations in a particular file, known as the Dump History file. Backup and restore operation information is stored in this file in a structured way and can be reliably used to get information about backups performed on a given database. When Dump History is enabled for a linked source, Delphix will use this feature to get information about backups for all sync operations including validated sync.

9.10.11.2 Prerequisites

To use this feature, Dump history needs to be enabled at the ASE Source instance level.

9.10.11.2.1 Linking a database with dump history enabled

By default, Delphix uses the backup server log file to discover backup information. Users can switch to Dump History by explicitly enabling it for the dSource. Before linking a database with Dump History enabled, make sure that Dump History is enabled on the source ASE instance. To enable Dump History for the database being linked, go through the linking wizard and on the **Data Management** page, check the box marked "Use Dump History".



9.11 SAP HANA data sources

> Starting 6.0.8.0 and HANA plugin version 6.0.0, Delphix provides you with a vSDK-based solution. **Delphix highly recommends their users to use this HANA Staged (vSDK-based) plugin over the existing HANA Direct (Lua-based) plugin**.

> Currently, the Lua-based HANA plugin 4.3.1 and earlier versions are also supported. Please refer to SAP HANA Release Notes (see page 521) for supported versions. For Lua-based HANA plugin documentation, refer to Delphix Engine 6.0.7.0 Documentation[597] and the earlier versions.

This section covers the following topics:

- SAP HANA overview (see page 1954)
- Delphix architecture (see page 1955)
- SAP HANA 2.0 plugin installation (see page 1959)
- SAP HANA 2.0 plugin tools (see page 1959)
- Quick start guide for SAP HANA plugin (see page 1963)
- SAP HANA support and requirements (see page 1982)
- Managing SAP HANA environments and hosts (see page 1992)
- Linking data sources and syncing data with SAP HANA (see page 1998)
- Provisioning and managing VDBs from SAP HANA dSources (see page 2008)

9.11.1 SAP HANA overview

SAP HANA is an in-memory database, columnar-oriented database management system developed by SAP. The key differentiators of HANA are its tight integration into the SAP application stack coupled with its columnar store, the latter of which is intended to enable customers to fold both OLTP and OLAP operations into a single database entity. In practical terms, this means an SAP HANA database stores both analytical and transactional data. Analysis of the data combination is possible on the fly, helping businesses make real-time analytical decisions.

SAP HANA as a database product continues to evolve dramatically. Key among recent changes has been the movement from a Single Database Container (SDC) to Multi-Tenant Database Containers (MDC), an architectural change that has become the exclusive architecture of HANA version 2.0.

9.11.1.1 Implications of the HANA column store

In a row-based relational database, data is stored by row, usually in large files. Data is then accessed from the database by reading and returning full rows of data.

Row

```
Country Product Sales
1   US Alpha   3000
2   US Beta    1250 <-----storing this line across------>
3   JP Alpha   400
4   UK Alpha   700
```

Column

597 https://docs.delphix.com/docs607

```
Country Product Sales    ^
US      Alpha 3000       |
US      Beta 1250    storing each column up / down
JP      Alpha 400        |
UK      Alpha 700        v
```

9.11.1.2 Select

In a columnar database, the entire row does not need to be read, only the columns associated with the query. This columnar structure improves (read) efficiency and, therefore, query performance, particularly for extensive data sets.

Columnar data storage allows highly efficient compression because most of the columns contain only a few distinct values (compared to the number of rows). The data, therefore, has a smaller memory footprint and takes fewer processing cycles to interrogate.

Better parallel Processing: in a column store, data is already vertically partitioned. This means that operations on different columns can easily be processed in parallel. If multiple columns need to be searched or aggregated, each of these operations can be assigned to a different processor core.

9.11.1.3 Insert / Update

Storing data in columns instead of rows is challenging for workloads with many data modifying operations. Therefore, the concept of a differential buffer was introduced, where new entries are written to a differential buffer first. In SAP HANA, this is referred to as the delta store. In contrast to the main store, the delta store is optimized for inserts. At a later point in time and depending on thresholds, e.g. the frequency of changes and new entries, the data in the delta store is merged into the main store. This process is referred to as delta merge.

The delta merge process has important implications for filesystem-based table modifications. Delta merge on any table will reorganize all data in that table and as a corollary will re-write the entire table to different filesystem blocks. *The space savings Delphix can offer by tracking block changes between snapshots becomes far less effective or predictable when entire tables are reorganized as a function of delta merge.*

9.11.2 Delphix architecture

The Delphix HANA plugin is based on a staging architecture, i.e. the linking process and SnapSync requires a staging environment. The staging environment is a HANA DB server host that leverages storage allocated directly on the Delphix Engine using NFS, and it must contain an instance of SAP HANA that is higher or matches the same version of the Source. To create a dSource on a staging server, a full backup would be required. The Recover command deposits its backup data onto the mounted storage area. The Delphix Engine snapshots the Staging server and the snapshots can be provisioned out to one or more target servers.

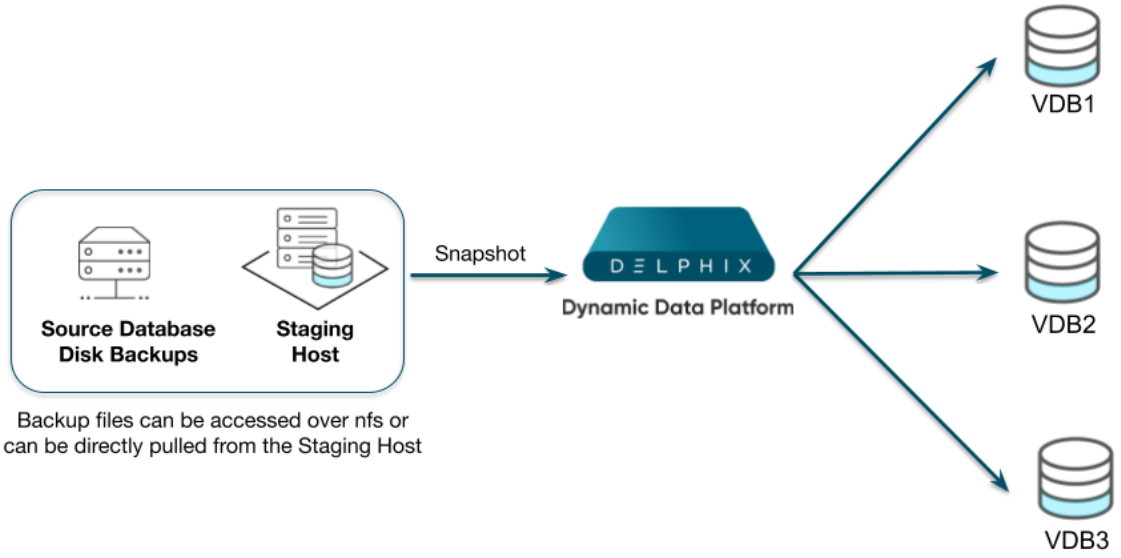
To create a dSource on a staging server, a full backup of one particular tenant database is required. With this solution, virtualization will be done at the tenant level only. This topic describes the high-level architecture of the Delphix HANA plugin solution.

This solution invokes HANA recovery of database backup files when a user selects to create a dSource on the staging server. The necessary backups/logs are created outside of Delphix and must exist before dSource creation.

Virtual databases can only be provisioned from a dSource (inheriting the state of the dSource).

The following is the high-level Delphix architecture with SAP HANA.

Delphix for HANA Architecture



There are two mechanisms to create dSource on the staging server.

- Pull architecture, where the HANA plugin pulls the data into the Delphix provided mount location.
- Staging Push architecture, where the user pushes the data into the Delphix engine.

9.11.2.1 HANA plugin with pull architecture

In Delphix HANA with Pull architecture, dSource is created using External backups which must be a HANA Native backup (ie: no support for third-party Backint tools). With External backups, the first dSource snapshot must include a Full backup, but can optionally include one or more incremental or differential backups. Subsequent dSource snapshots can be based upon additional incremental or differential backups (This will require a full backup along with incremental/differential backups, as recovery in Hana always requires a full backup).
HANA plugin supports an External backup, also known as Native Backups. Native backup implies that the backups are written in native storage format to local storage without a third-party backup tool.

The dSource creation operation invokes the HANA database recovery of the relevant HANA backup files. HANA database recovery can be a lengthy process. The duration of recovery time is subject to the size of the

database being recovered. After completion of HANA recovery, the HANA instance will contain the newly recovered dSource tenant.

VDB provision operations from a dSource snapshot do not require HANA database recovery and can occur very quickly.

Failure to follow this recommendation diminishes the key value proposition for HANA, which is agility. Below are the recommended tiers:

- **dSource:** A virtual HANA database constructed by recovery.
- **Virtual Database (VDB):** A copy of a specific dSource snapshot, cloned using Delphix technology. These copies are fast to create with a small initial storage footprint. VDBs are the recommended workbench for developers. Multiple VDBs can exist from a single dSource. Each VDB corresponds to a dSource snapshot.

9.11.2.2 SAP HANA plugin with staging push architecture

Starting Delphix Engine version 6.0.12.0 and HANA plugin version 6.2.0, a new data ingestion mechanism has been introduced that will help users to push data into the Delphix provided mount point on their own.

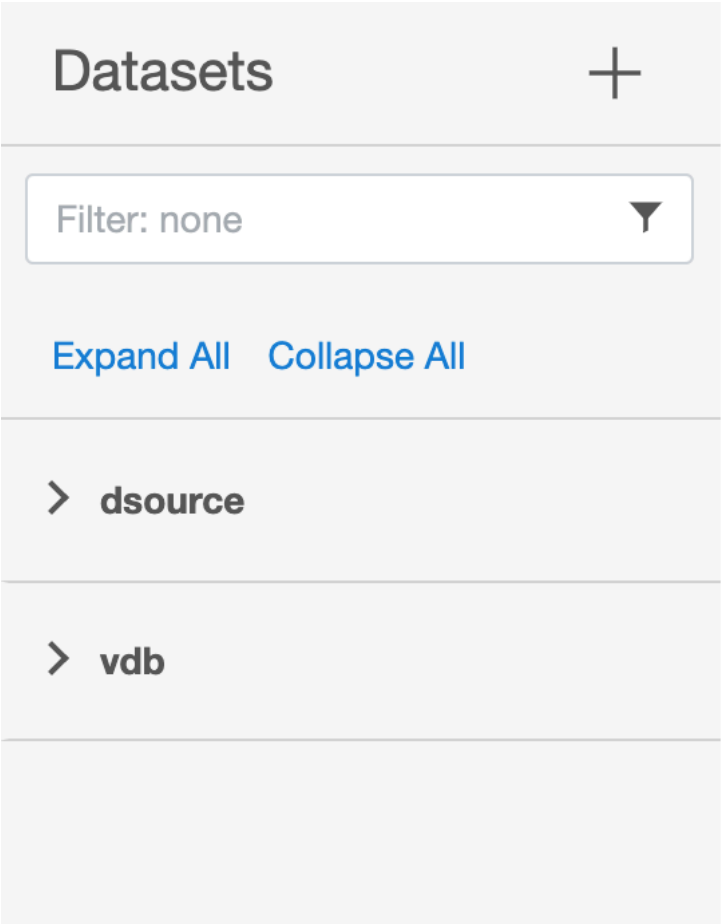
The previous versions of the HANA plugin were based only on the Pull approach and it was difficult to support the different ingestion mechanisms as it required a lot of effort to analyze, code, and develop the solution. With the Staging Push mechanism, data ingestion can be achieved with minimum effort.

Few points to be noted about the Staging Push mechanism.

- dSource creation creates an empty mount point on the staging host
- User managed backup and restore

9.11.2.3 Operational best practice

- Construct a dSource using the backup files. Construct multiple VDBs descended from the dSource.
- Create a meaningful object name.
- Create Dataset Groups that help delineate to which tier an object belongs.



- Segregate user targets (where VDBs reside) from the staging where HANA recovery must take place to create the dSource.
- Provision end-user VDBs, so that they can be created, refreshed, and rewound quickly.
- Layer VDBs to perform any common post-recovery operations to further streamline provisioning steps.
- Closely monitor and manage the quantity and lifespan of dSource and VDB snapshots, since each can consume additional Delphix storage.

9.11.2.4 Benefits of HANA plugin

- Virtualization of individual tenant databases instead of the entire system.
- Ability to mix and match virtualized tenants across any HANA system.
- Reduced Storage footprint. VDBs can use significantly less storage when initially provisioned from dSource.
- Relaxes strict requirement of exact HANA version match between source and target, leveraging HANAs ability to recover an earlier version against a more recent version.
- Source backup files can be ingested from a surrogate host without impact to the true production source host.

9.11.2.5 Limitations of HANA plugin

- The tenant's name should contain at least one alphanumeric character or one underscore. Also, it should not start with a digit.
- A single quote, double quote, and space characters are not allowed in the password.
- Datasets created using the Lua version cannot be upgraded to the vSDK version.
- Manual cleanup/unmounting would be required after **force delete** operation on any dataset.

9.11.3 SAP HANA 2.0 plugin installation

The HANA 2.0 Plugin is provided as a zip file on the Delphix download site. Follow the steps below to download the plugin:

1. In the web browser, go to the Delphix download[598] site.
2. Login to the download site using email and password credentials.
3. Navigate to the Connectors > Continuous Data > SAP HANA folder.
4. Go to the Plugins > HANA > HANA 2.0 and download the Plugin_<version_number>.zip file.

For initial and upgrade installations refer to the section Delphix Engine Plugin Management (see page 2257) for further details.

9.11.4 SAP HANA 2.0 plugin tools

Delphix provides the **create_backup_metadata.sh** script, which is bundled with the plugin. This script is required to be executed by customer for customer-created HANA backups, which are to be used to create or refresh a dSource. This script can be found at:

```
<path>/scripts/create_backup_metadata.sh
```

This script must be run prior to creating or refreshing a dSource from a customer-created backup. The script is not required for Delphix-initiated backups. The script must be run on the source instance where the source tenant database exists. The script retrieves and persists important metadata about the specific backup set. This information will be used by the plugin during the dSource creation and will then be stored in the metadata section of the dSource.

9.11.4.1 Prerequisites

- The script assumes the user has the credentials of the tenant on the source HANA instance.
- The script must be copied to the source host as the operating system user should have access to hdbsql.
- The script must be executable.

598 https://download.delphix.com/

- Correct backup files of the source must exist on the source host or the backup files should be accessible from the source host.
- The script should have the write permissions in the directory where the backups have been kept.

You can run the `create_backup_metadata.sh` script in the following two ways.

1. Interactive mode
2. Providing the complete list of parameters (Non-Interactive mode)

9.11.4.2 Interactive mode

The mode allows you to input the values to the required/optional parameters one by one on the CLI. See the sample script run example below the Parameters table.

9.11.4.2.1 Parameters

The following table lists the optional and required parameters used by the script.

Parameter	Optional/Required	Description
Please enter the location of the Backup files:	Required	Location of the Backup files For example: */usr/sap/HDB/HDB00/source_backup*
If the above provided location of the Backup files is different than the original location where backups were taken then please enter original location, otherwise press enter(OPTIONAL):	Optional	This is the original location where the backup files were created during the dSource creation which can be used later for recovery purposes Provide this parameter when the location provided at the "*Please enter the location of the Backup files:*" parameter is different from the location where the backup files are kept now For example: */usr/sap/HDB/HDB00/backup/data/DB_TEST1*
Please enter the location of the Log files(OPTIONAL):	Optional	Location of the Log files You may use logs for the recovery For example: */usr/sap/HDB/HDB00/source_log*

Parameter	Optional/Required	Description
If the above provided location of the log files is different than the original location where log files were created then please enter original location, otherwise press enter(OPTIONAL):	Optional	This is the original location where the log files were created during the dSource creation which can be used later for recovery purposes Provide this parameter when the location provided at the "*Please enter the location of the Backup files:*" parameter is different from the location where the log files are kept now For example: */usr/sap/HDB/HDB00/backup/log/DB_TEST1* (Note that this is not a valid directory)
Please enter the Tenant Database Name:	Required	Name of the tenant database for which backups/logs were created For example: TEST1
Please enter the SID of the HANA database:	Required	SID of the tenant database for which backups/logs are present For example: HDB
Enter the Tenant User Name:	Required	Username to connect to the tenant database For example: SYSTEM
Tenant User Password:	Required	Password to connect to the tenant database
Please enter the Instance Number:	Required	Instance number of the tenant database For example: 00

9.11.4.2.2 Example

The following example shows an execution of the script with input parameters:

```
hdbadm@ip-10-110-244-207.delphix:/usr/sap/HDB/HDB00> ./create_backup_metadata.sh
Starting create_backup_metadata.sh
Please enter the location of the Backup files:/usr/sap/HDB/HDB00/source_backup

If the above provided location of the Backup files is different than the original location where backups were taken then please enter original locatio
n, otherwise press enter(OPTIONAL):/usr/sap/HDB/HDB00/backup/data/DB_TEST1

Please enter the location of the Log files(OPTIONAL):/usr/sap/HDB/HDB00/backup/log/DB_TEST1

If the above provided location of the log files is different than the original location where log files were created then please enter original locati
on, otherwise press enter(OPTIONAL):

Please enter the Tenant Database Name:TEST1

Please enter the SID of the HANA database:HDB

Please enter the Tenant User Name:SYSTEM

Tenant User Password:

Please enter the Instance Number:00
```

9.11.4.2.3 Usage

```
./create_backup_metadata.sh
```

9.11.4.3 Providing the complete list of parameters

This mode allows you to input the parameter values in a specified order at the run time. See the sample script run example below the Parameters table.

9.11.4.3.1 Parameters

The following table lists the optional and required parameters in the order that in which you need to pass the parameter values along with the script.

Parameter	Optional/Required	Description
TENANT_DATABASE_NAME	Required	Tenant database name for which the script needs to be run
SID	Required	SID of the HANA database
TENANT_USER_NAME	Required	Username to connect to the tenant database
TENANT_USER_PASSWORD	Required	Password to connect to the tenant database
INSTANCE_NUMBER	Required	Instance number of the HANA database
DATA_BACKUP_PATH	Required	The location where the backups are present
LOG_BACKUP_PATH	Optional	The location where the logs are present
ORIG_DATA_BACKUP	Optional	Original location where these backups were created. Do not provide any value if backups are present at the same location where they were created

Parameter	Optional/Required	Description
ORIG_LOG_BACKUP_PATH	Optional	Original location where logs were created. Do not provide any value if the logs are present at the same location where they were created

9.11.4.3.2 Example 1

The following example shows an execution of the script with input parameter values in the above-specified order.

```
hdbadm@target:/usr/sap/HDB/HDB00> ./create_backup_metadata.sh TEST HDB SYSTEM Delphix_123 00 /usr/sap/HDB/HDB00/source_backup
Starting create_backup_metadata.sh
Checking : /usr/sap/HDB/HDB00/source_backup



Running the script for TEST tenant
BACKUP ID of latest complete backup: 1633404668463
Metadata information of backup is stored in file: /usr/sap/HDB/HDB00/source_backup/backupTimeInfo
```

If all the mandatory parameters are not provided or parameters are not provided in the specified order, then the plugin will throw an error.

9.11.4.3.3 Example 2

The following example shows an error when all the mandatory parameter values are not provided.

```
hdbadm@target:/usr/sap/HDB/HDB00> ./create_backup_metadata.sh TEST HDB SYSTEM Delphix_123
Starting create_backup_metadata.sh


Incorrect number of parameters provided to the script. Please provide the below parameters to the script or do not pass any parameters which will invo
ke the script in interactive mode.

TENANT_DATABASE_NAME - Tenant Database name for which script needs to be run.

SID - SID of the HANA Database

TENANT_USER_NAME - User name to connect to the Tenant Database

TENANT_USER_PASSWORD - Password to connect to the Tenant Database

INSTANCE_NUMBER - Instance number of the HANA Database

DATA_BACKUP_PATH - Location where backups are present

LOG_BACKUP_PATH - Location where logs are present. This is an optional field.

ORIG_DATA_BACKUP - Original location where these backups were created. This is an optional field. Do not provide any value if Backups are present at t
he same location where they were created.

ORIG_LOG_BACKUP_PATH - Original location where logs were created. This is an optional field. Do not provide any value if Logs are present at same local
```

9.11.5 Quick start guide for SAP HANA plugin

This quick start guide, which is excerpted from the larger User Guide, is intended to provide you with a quick overview of working with HANA database objects in the Delphix Engine. It does not cover any advanced configuration options or best practices, which can have a significant impact on performance. It assumes that you are working in a Lab/Dev setting and attempting to quickly test Delphix functionality. It assumes you will use the VMware Hypervisor.

9.11.5.1 Overview

In this guide, we will walk through deploying a Delphix Engine, starting with configuring Source and Target database environments on Windows servers. We will then create a dSource, and provision a VDB.

For purposes of the QuickStart, you can ignore any references to Replication or Masking.

9.11.5.2 Deploy OVA on VMware

Use the Delphix-supplied OVA file to install the Delphix Engine. The OVA file is configured with many of the minimum system requirements. The underlying storage for the install is assumed to be redundant SAN storage.

1. Download the OVA file from https://download.delphix.com[599]. You will need a support login from your sales team or a welcome letter.
    a. Navigate to the Delphix Product Releases/<Current Version>/Appliance Images page.
2. Login using the vSphere client to the vSphere server (or vCenter Server) where you want to install the Delphix Engine.
3. In the vSphere Client, click **File**.
4. Select **Deploy OVA Template**.
5. Browse to the OVA file.
6. Click **Next**.
7. Select a **hostname** for the Delphix Engine. This hostname will also be used in configuring the Delphix Engine network.
8. Select the **data center** where the Delphix Engine will be located.
9. Select the **cluster** and the **ESX host**.
10. Select one (1) **data store** for the **Delphix OS**. This datastore can be **thin-provisioned** and must have enough free space to accommodate the 127GB comprising the Delphix operating system.
11. Select four (4) or more **data stores** for Database Storage for the Delphix Engine. The Delphix Engine will stripe all of the Database Storage across these VMDKs, so for optimal I/O performance, each VMDK must be equal in size and be configured **Thick Provisioned - Eager Zeroed**. Additionally, these VMDKs should be distributed as evenly as possible across all four SCSI I/O controllers.
12. Select the **virtual network** you want to use. If using multiple physical NICs for link aggregation, you must use vSphere NIC teaming. Do not add multiple virtual NICs to the Delphix Engine itself. The Delphix Engine should use a single virtual network. For more information, see Optimal Network Architecture for the Delphix Engine (see page 719)
13. Click **Finish**. The installation will begin and the Delphix Engine will be created in the location you specified.

599 https://download.delphix.com/

14. Once the installation has completed, power on the Delphix Engine and proceed with the initial system configuration as described in Setting Up Network Access to the Delphix Engine (see page 569)

> If your source database is 4 TB, you probably need 4 TB of storage for the Delphix Engine. Add at least 4 data disks of similar size for the Delphix VM. For example: for a source database of 4 TB, create 4 VMDKs of 1 TB each.

> For a full list of requirements and best practice recommendations, see Virtual Machine Requirements for VMware Platform (see page 604)

9.11.5.3 Setup network access to the Delphix engine

1. Power on the Delphix Engine and open the Console.
2. Wait for the Delphix Management Service and Delphix Boot Service to come online. This might take up to 10 minutes during the first boot. Wait for the large orange box to turn green.
3. Press any key to access the sysadmin console.
4. Enter `sysadmin@SYSTEM` for the username and `sysadmin` for the password.
5. You will be presented with a description of available network settings and instructions for editing.

```
Delphix Engine Network Setup

To access the system setup through the browser, the system must first be
configured for networking in your
environment. From here, you can configure the primary interface, DNS, hostname,
and default route. When DHCP is
configured, all other properties are derived from DHCP settings.

To see the current settings, run "get." To change a property, run "set =." To
commit your changes,
run "commit." To exit this setup and return to the standard CLI, run "discard."

    defaultRoute    IP address of the gateway for the default route -- for
 example, "1.2.3.4."

    dhcp            Boolean value indicating whether DHCP should be used for
 the primary interface. Setting this value
                    to "true" will cause all other properties (address,
hostname, and DNS) to be derived from the DHCP
                    response
```

```
    dnsDomain       DNS Domain -- for example, "delphix.com"

    dnsServers      DNS server(s) as a list of IP addresses -- for example,
"1.2.3.4,5.6.7.8."

    hostname        Canonical system hostname, used in alert and other logs --
for example, "myserver"

    primaryAddress  Static address for the primary interface in CIDR notation
-- for example, "1.2.3.4/22"

Current settings:

    defaultRoute: 192.168.1.1
    dhcp: false
    dnsDomain: example.com
    dnsServers: 192.168.1.1
    hostname: Delphix
    primaryAddress: 192.168.1.100/24
```

6. Configure the `hostname`. If you are using DHCP, you can skip this step.

```
delphix network setup update *> set hostname=<hostname>
```

Note:
Use the same `hostname` you entered during the server installation.

7. Configure DNS. If you are using DHCP, you can skip this step.

```
delphix network setup update *> set dnsDomain=<domain>
delphix network setup update *> set dnsServers=<server1-ip>[,<server2-ip>,...]
```

8. Configure either a static or DHCP address.

DHCP Configuration

```
delphix network setup update *> set dhcp=true
```

Static Configuration

```
delphix network setup update *> set dhcp=false
delphix network setup update *> set primaryAddress=<address>/<prefix-len>
```

Note:
The static IP address must be specified in CIDR notation (for example, `192.168.1.2/24`)

9. Configure a default gateway. If you are using DHCP, you can skip this step.

```
delphix network setup update *> set defaultRoute=<gateway-ip>
```

10. Commit your changes. Note that you can use the `get` command prior to committing to verify your desired configuration.

```
delphix network setup update *> commit
Successfully committed network settings. Further setup can be done through the
browser at:

        http://<address>

Type "exit" to disconnect, or any other commands to continue using the CLI.
```

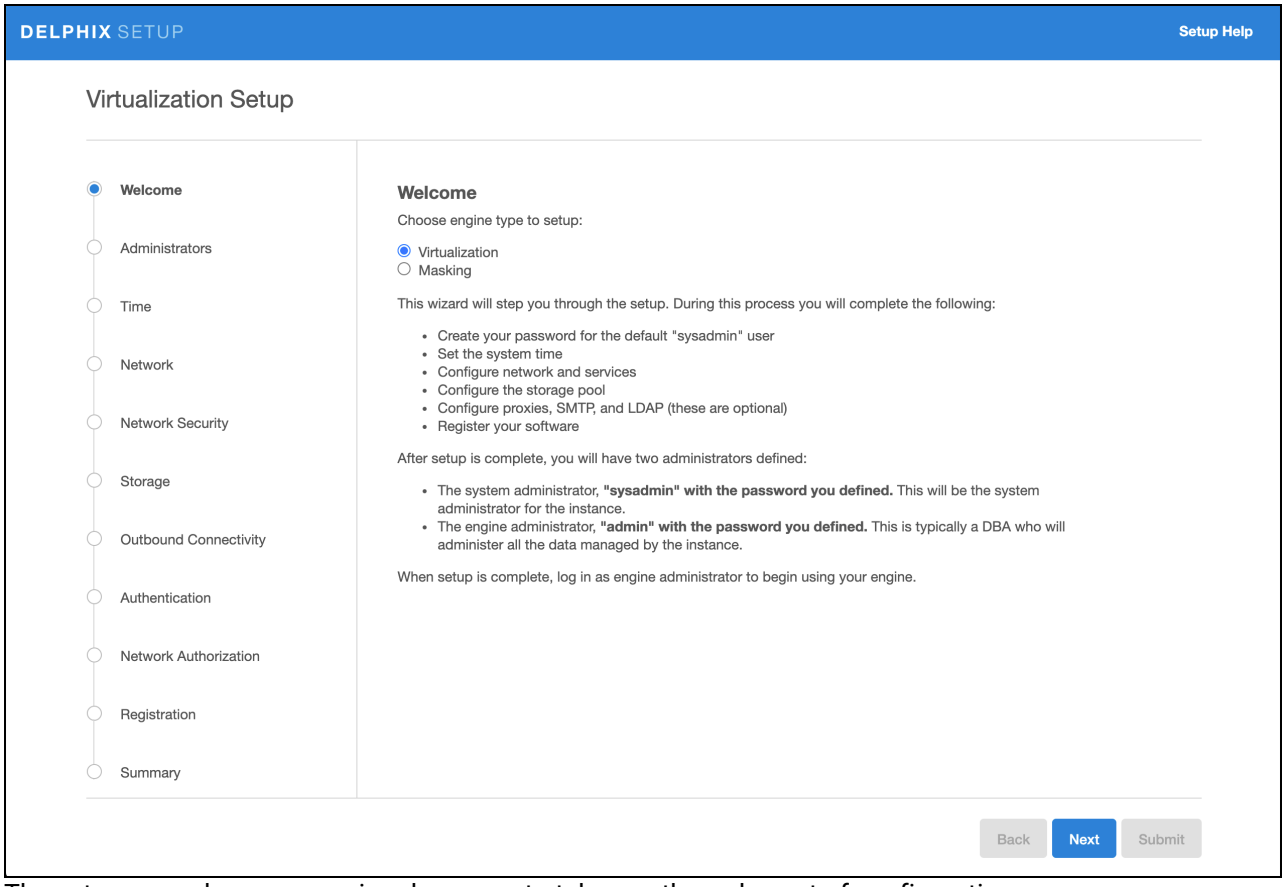
11. Check that you can now access the Delphix Engine through a Web browser by navigating to the displayed IP address, or hostname if using DNS.
12. Exit setup.

```
delphix> exit
```

9.11.5.4 Setting up the Delphix engine

Once you set up the network access for Delphix Engine, navigate to the Delphix Engine URL in your browser for server setup.

The welcome screen below will appear for you to begin your Delphix Engine setup.



The setup procedure uses a wizard process to take you through a set of configuration screens:

- Administrators
- Time
- Network
- Network Security
- Storage
- Outbound Connectivity
- Authentication
- Network Authorization
- Registration
- Summary

1. Connect to the Delphix Engine at `http:///<Delphix Engine>login/index.html#serverSetup`. The **Delphix Setup** application will launch when you connect to the server. Enter your **sysadmin** login credentials, which initially defaults to the username **sysadmin**, with the initial default password of **sysadmin**. On first login, you will be prompted to change the initial default password.
2. Click **Next**.

9.11.5.4.1 Administrators

The Delphix Engine supports two types of administrators:

- System Administrator (**sysadmin**) - this is the engine system administrator. The sysadmin password is defined here**.**
- Engine Administrator (**admin)** - this is typically a DBA who will administer all the data managed by the engine.

On the Administrators tab, you set up the sysadmin password by entering an email address and password. The details for the admin are displayed for reference.

> The default domain user created on Delphix Engines from 5.3.1 is known as **admin** instead of delphix_admin. When engines created before 5.3.1 are upgraded to 5.3.1 or later they will retain their old username 'delphix_admin'. To avoid complications Delphix recommends creating users with an admin role and then disabling the delphix_admin.

9.11.5.4.2 Time

The engine time is used as the baseline for setting policies that coordinate between virtual databases and external applications

Choose your option to setup system time in this section. For a Quick Start, simply set the time and your timezone. You can change this later.

9.11.5.4.3 Network

The initial out-of-the-box network configuration in the OVA file is set to use a single VMXNET3 network adapter.

You have already configured this in the initial configuration. Delphix supports more advanced configurations, but you can enable those later.

9.11.5.4.4 Storage

You should see the data storage VMDKs or RDMs you created during the OVA installation. Click **Next** to configure these for data storage.

9.11.5.4.5 Serviceability

Choose your options to configure serviceability settings.

For a Quick Start, accept the defaults. You can change this later.

9.11.5.4.6 Authentication service

Choose your options to configure authentication services.

For a Quick Start, accept the defaults. You can change this later.

9.11.5.4.7 Registration

If the Delphix Engine has access to the external Internet (either directly or through a web proxy), then you can auto-register the Delphix Engine:

1. Enter your **Support Username** and **Support Password**.
2. Click **Register**.

If external connectivity is not immediately available, you must perform manual registration.

1. Copy the **Delphix Engine registration code** in one of two ways:
   a. Manually highlight the registration code and copy it to clipboard. Or,
   b. Click **Copy Registration Code to Clipboard**.
2. Transfer the Delphix Engine's registration code to a workstation with access to the external network Internet. For example, you could e-mail the registration code to an externally accessible e-mail account.
3. On a machine with access to the external Internet, please use your browser to navigate to the Delphix Registration Portal at http://register.delphix.com[600].
4. Login with your Delphix support credentials (username and password).
5. Paste the **Registration Code**.
6. Click **Register**.

> Although your Delphix Engine will work without registration, we strongly recommend that you register each engine as part of the setup. Failing to register the Delphix Engine will impact its supportability and security in future versions.

To regenerate the registration code for a Delphix Engine please refer to, Regenerating the Delphix Engine Registration Code (see page 672). Delphix strongly recommends that you regenerate this code and re-register the engine regularly to maximize the Support Security of the Delphix Engine. Delphix recommends doing this every six months.

600 http://register.delphix.com/

9.11.5.4.8 Summary

The final summary tab will enable you to review your configurations for System Time, Network, Storage, Serviceability, and Authentication.

1. Click the **Back** button to go back and to change the configuration for any of these server settings.
2. If you are ready to proceed, then click **Submit**.
3. Click **Yes** to confirm that you want to save the configuration.
4. Click **Setup** to acknowledge the successful configuration.
5. There will be a wait of several minutes as the Delphix Engine completes the configuration.

9.11.5.5 Requirements for HANA staging/target host and databases

Backup of the tenant must be present which needs to be recovered. Along with backup, metadata information generated by script create_backup_metadata.sh is also expected in the backup location. As we will be recovering a tenant through backup files the important thing while creating the dSource which should be taken care of is the user should have access to the backup directory where the backup file resides.

General staging Host Requirements

- HANA database must be installed and running.
- Read-write permission on the backup directory.
- There must be an operating system user which has HANA administrative user privileges <sid>adm:
    - The Delphix Engine must be able to make an SSH connection to the staging environment using the operating system user.
    - The operating system user must have read and execute privileges on the HANA binaries installed on the staging environment.
    - The operating system user must have read, write, and execute access to the HANA data directories on the staging environment.
- A plugin directory must exist on the staging host. The directory must have the following properties:
    - Should be writable by the operating system user described above.
    - Should have at least 256 MB of available storage.
- TCP/IP connectivity to and from the staging environment must be configured as described in General Network and Connectivity Requirements (see page 717)
- Hostname and IP address must be correctly configured.

9.11.5.6 Requirements for HANA target host and databases

A **target environment** is where virtualized databases run. A target environment must have a HANA instance installed and configured. The VDB will become a tenant on the target database instance.

9.11.5.7 General target environment requirements

1. The operating system of the target environment must match the staging environment.
2. The SAP HANA installation on the target environment must have the same or higher version than the staging environment.
3. There must be an operating system user with the following privileges:
   a. The Delphix Engine must be able to make an SSH connection to the target environment using the operating system user.
   b. The operating system user must have read and execute privileges on the HANA binaries installed on the target environment.
   c. The operating system user must have permission to run mount and unmount as the superuser via sudo with neither a password nor a TTY. See Sudo Requirements for the HANA 2.0 Plugin (see page 1990)
4. Prior to the discovery, a plugin directory on the target environment must exist with the following properties:
   a. The plugin directory must be writable by the operating system user mentioned above.
   b. The plugin directory must have at least 256 MB of available storage.
5. There must be a mount point directory (for example, /mnt) that will be used as the base for mount points that are created when provisioning a VDB with the following properties:
   a. The mount point directory must be writable by the operating system user mentioned above.
   b. The mount point directory should be empty.
6. TCP/IP connectivity to and from the source environment must be configured as described in General Network and Connectivity (see page 717) Requirements.
7. The hostname and IP address must be correctly configured.
8. HANA instance on which the VDB creation is being done should have the empty ports to host the services for the tenant database. The default port number range for tenant databases is 3<instance>40—3<instance>99. This means that the maximum number of tenant databases that can be created per instance is 20. However, you can increase this by reserving the port numbers of further instances. You do this by configuring the property [multidb] reserved_instance_numbers in the global.ini file. The default value of this property is 0. If you change the value to 1, the port numbers of one further instance are available (for example, 30040—30199 if the first instance is 00). If you change it to 2, the port numbers of two further instances are available (for example, 30040—30299 if the first instance is 00). And so on.

9.11.5.8 Adding a HANA environment

9.11.5.9 Prerequisites

- Make sure that the HANA environment in question meets the requirements described in Requirements for HANA Staging Hosts and Databases (see page 1988) and Requirements for HANA Target Hosts and Databases (see page 1989)

9.11.5.10 Procedure

- Login to the **Delphix Management** application.
- Click **Manage**.
- Select **Environments**.
- Next to **Environments**, click the **Actions (...) menu and** select **Add Environment**.
- In the Add Environment dialog, In the **Host OS** select **Unix/Linux** on the menu.
- In **Server Type**, select **Standalone Host**.
- Click **Next**.
- Enter **Name** for the environment.
- Enter the **Host IP address** or **fully qualified hostname**.
- Enter the **SSH port**. The default value is 22.
- Enter an **OS Username** for the environment which should be ".>'.>
- Select **Login Type**.
  Note:
  Currently, the Enterprise Password Vault option is not supported.
- Username and Password - enter the OS username and password
- Username and Public Key - enter the OS username.
- Password Vault - select from an existing Enterprise Password Vault

Using public key authentication
If you want to use public-key authentication for logging into your Unix-based environment, there are two options: use the engine's key pair or provide a key pair for this environment.
To use the engine's key pair:

1. Select **Public Key** for the **Login Type.**
2. Click **View Public Key**.
3. Copy the public key that is displayed, and append it to the end of your `~/.ssh/authorized_keys` file. If this file does not exist, you will need to create it.
   a. Run chmod 600 ~/.ssh/authorized_keys to allow only the file's owner to read and write to it (make sure the file is owned by the user).

b. Run chmod 755 ~ to restrict access to the user's home directory so no other user may write to it.
c. Run chmod 700 ~/.ssh so that others cannot write to it. The ~/.ssh directory cannot be writable by group or other users. Otherwise, authentication will fail.

As an alternative, you can provide a key pair specific for this environment via the API or CLI. See Option 2 in this CLI Cookbook article (see page 1963) for instructions.

1. For Password, enter the password associated with the user in step 11.
2. Select **Public Key** for the **Login Type.**
3. Click **View Public Key**.
4. Copy the public key that is displayed, and append it to the end of your `~/.ssh/authorized_keys` file. If this file does not exist, you will need to create it.
   a. Run chmod 600 ~/.ssh/authorized_keys to allow only the file's owner to read and write to it (make sure the file is owned by the user).
   b. Run chmod 755 ~ to restrict access to the user's home directory so no other user may write to it.
   c. Run chmod 700 ~/.ssh so that others cannot write to it. The ~/.ssh directory cannot be writable by group or other users. Otherwise, authentication will fail.
5. For Password Login, click 'Validate' to test the username and password.
6. Enter a **Plugin Path.** The location specified must be a string without spaces. Ensure this directory path exists on the host in question. Environment discovery will fail if the directory does not exist on the host.
7. To provide your own Oracle Java select the **Provide my own JDK** checkbox and click **Next**.
8. In the Java Development Kit tab enter the absolute path to your Oracle JDK and click **Next**. For more information, see Providing Your Own Oracle Java (see page 1025)
9. Click **Submit**. On success, the new environment will be added to the list in the Environments tab.
10. Select the "Databases" tab to view discovered HANA databases:

Details | Databases

Installations

Filter: none

Add Dataset Home

HANA 2.00.052.00.1599235305 /usr/sap/HDB/HDB00

INSTALLATION DETAILS

Allow Provisioning
Yes

System Administrator User
hdbadm

SAP Mount Path
/hana/shared

HANA Version
2.00.052.00.1599235305

Hostname
source

HANA Instance Number
00

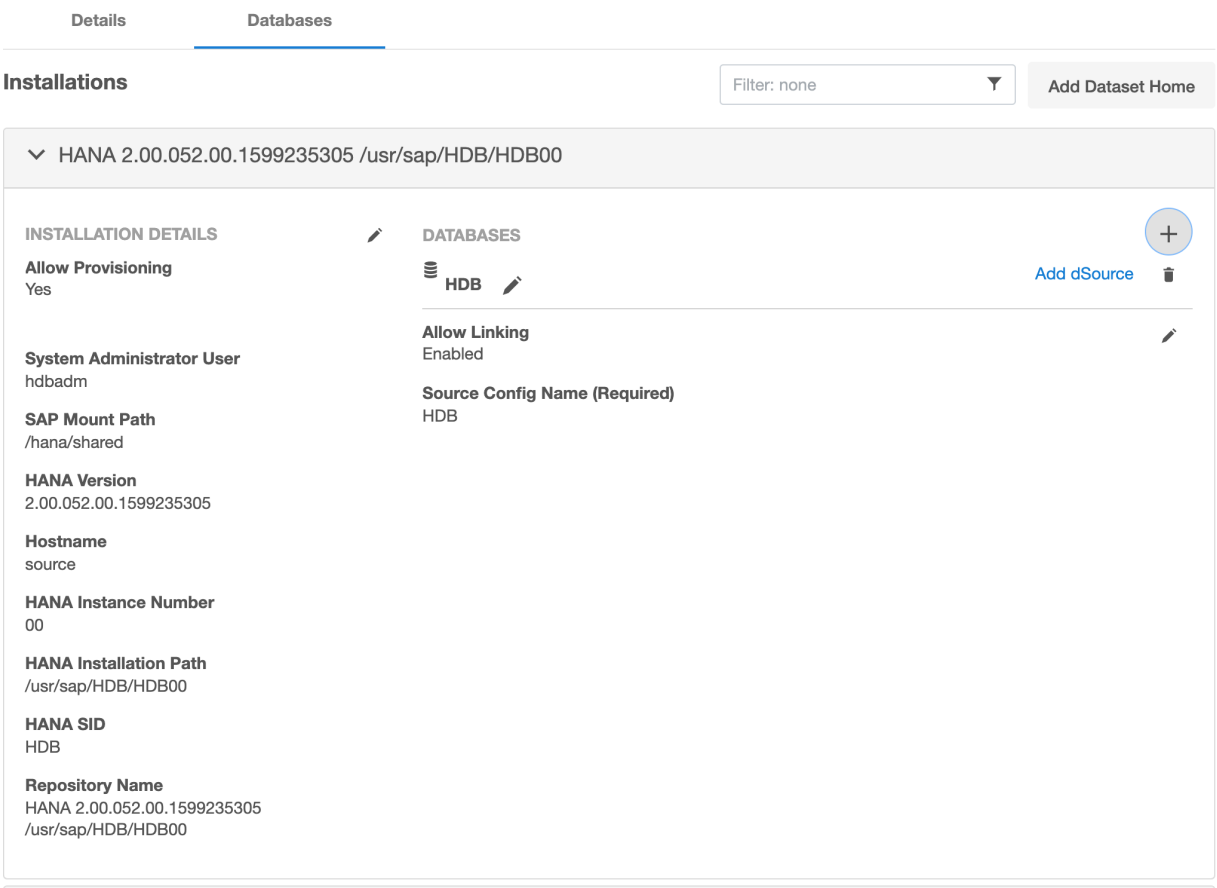
HANA Installation Path
/usr/sap/HDB/HDB00

HANA SID
HDB

Repository Name
HANA 2.00.052.00.1599235305
/usr/sap/HDB/HDB00

DATABASES

No databases found on installation.



9.11.5.11 Linking a HANA data source

This topic describes basic concepts behind the creation of a dSource for a HANA database using conventional pull and Staging push mechanisms.

> For Staging Push, only native/third-party backups are used to populate data in the staging database.

For Third-party Backups, users need to provide a “Backup Path” containing FULL and Incremental backups. These files will be copied to the Delphix Engine during the dSource linking process. In this scenario, the Delphix Engine will not require any credentials of the source database. Although credentials for the tenant database (for which backup has been placed) and systemdb of staging instances would be required.

9.11.5.12 Prerequisites for customer provided backups ingestion mechanism

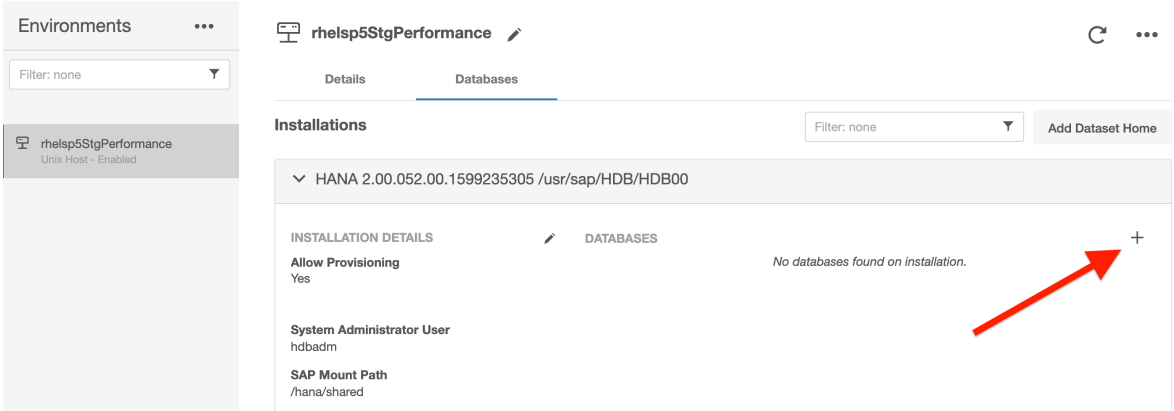
- Users need to run the create_backup_metadata.sh. This script will create the backupTimeInfo file at the same location where backups are kept. This script will gather all the required metadata

information and will store that information in the backupTimeInfo file. This information will then get used by the plugin at the time of creating dSource.

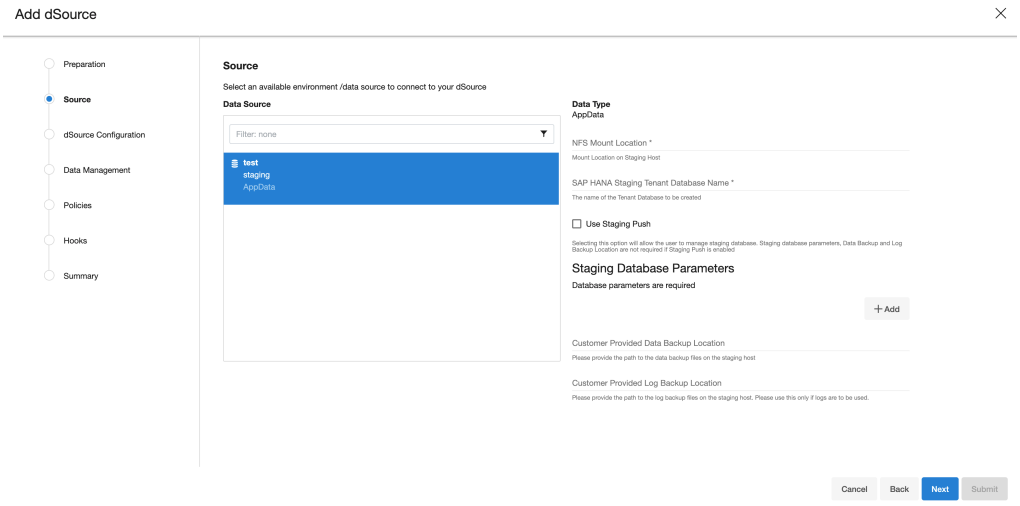
- More details regarding the script have been included in the HANA 2.0 Plugin Tools (see page 1959)

9.11.5.13 Procedure

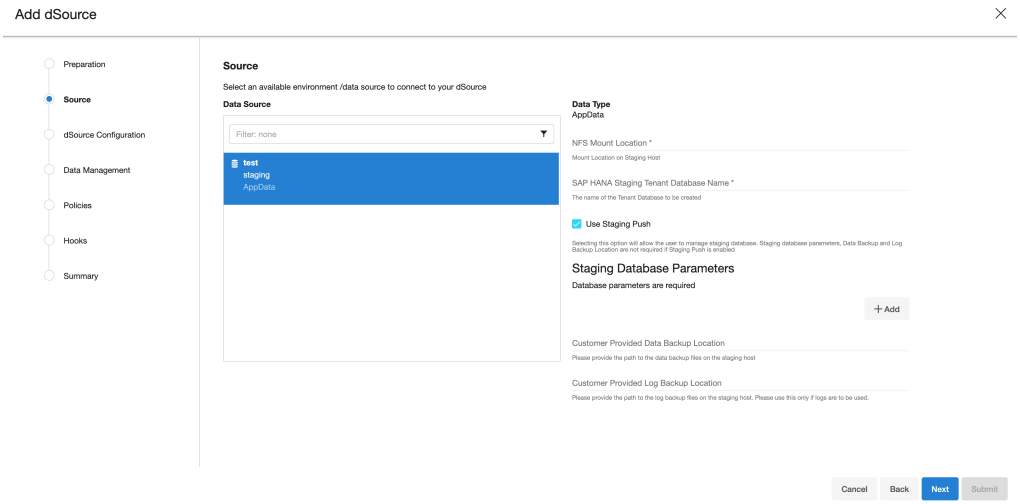
1. Login to the Delphix Management application.
2. Select **Manage > Environments**. Then select the Datasets tab.



3. Create a source configuration on which to build the dSource. You can do this by clicking on the **+** symbol. The Source configuration can have any name.
4. In the **Add dSource** wizard, select the required source configuration.
5. After selecting the source, below are the parameters required to create the dSource.
   a. Parameters required for the conventional **pull** mechanism:
      i. Mount Location on Staging Host
      ii. SAP HANA Source Tenant Database Name
      iii. Source SystemDB User Name
      iv. Source SystemDB Password
      v. Source Tenant Database User Name
      vi. Source Tenant Database Password
      vii. Customer Provided Data Backup Location: This represents the path to the location of the customer-provided backup files.
      viii. Customer Provided Log Backup Location: This represents the path to log the location of the customer-provided log backup files. This parameter is optional.



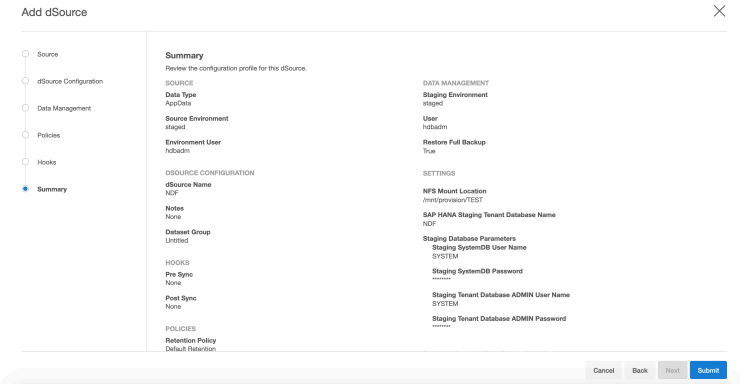
b. Parameters required for the **Staging push** mechanism
   i. Mount Location on Staging Host
   ii. SAP HANA Source Tenant Database Name
   iii. Enable the **Staging Push** feature by selecting the checkbox "**Use Staging Push**". With this feature, the user handles the Source Tenant database and its state on their own and you don't need to specify any other staging database parameters, data backup, and log backup locations. If this check box is selected, the HANA plugin will not invoke the tenant creation and recovery operation and if this checkbox is not selected then the default pull approach will be used.



c. Click **Next.**

6. Enter a name and select a group for your dSource. Adding a dSource to a dataset group lets you set Delphix Domain user permissions for that database and its objects, such as snapshots. See the topics under Users and Groups (see page 674) for more information.

7. Configure the SnapSync and Retention Policy. The SnapSync policy can be set to 'None' for customers who wish not to run a scheduled SnapSync job.
8. Optionally, users can provide the bash script as a plugin-defined hook. Both Pre-Snapshot and Post-SnapShot hooks can be used.
9. Select **Next.**
10. At the end of the workflow, review the dSource Configuration in the Summary pane as shown below:



Once the dSource linking job is submitted it is possible to monitor progress by selecting **Active Jobs** in the top menu bar. On successful completion, the new dSource will be visible in the list of Datasets under its assigned group.

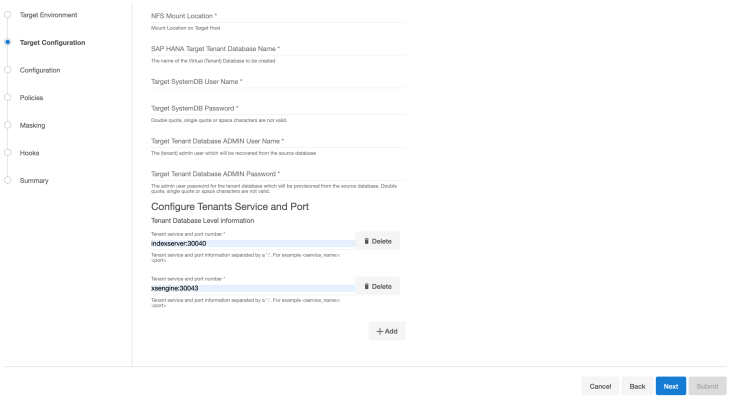
After dSource creation using the Staging Push mechanism, an empty snapshot will be created but users can not create VDB's from the same. Also, before creating the subsequent snapshots, users will have to create the database and point the database to use the Delphix provided mount point.

9.11.5.14 Provisioning a HANA VDB

9.11.5.15 Prerequisites

- The provision requires a linked dSource built off a physical source tenant, as described in Linking a HANA dSource (see page 1999), or a VDB.
- The HANA target environment should adhere to the Requirements for HANA Hosts and Databases (see page 1989)
- Operating system users and database users should be configured.

9.11.5.16 VDB creation UI



9.11.5.17 Procedure

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Datasets**.
4. Select a **dSource**.
5. Select a **snapshot** from which you want to provision.

   Note:
   If the first or empty dSource Snapshot that was created using the Staging Push mechanism is detected by the plugin and an error message is displayed. This check is done for every VDB creation operation.
6. Click the **Provision VDB** icon to open the Provision VDB wizard.
7. Select a target environment from the left pane.
8. The HANA 2.0 Plugin supports one installation on the same host, hence the value in the **Installation** dropdown will be already populated. Multiple drop-down options should not be available.
9. Set the **Environment User** to be the Instance Owner. Note: The picking of instance owner is only possible if you have multiple environment users set on that host.
10. Click **Next**.
11. Target Configuration pane you will need to specify the following:

    Note:

    '*****' indicates the mandatory fields.
    a. NFS Mount Location*: Delphix recommends setting the trailing directory to match the VDB name to delineate mount points and support administration and troubleshooting. In the

example shown, the **NFS Mount Location** is /mnt/provision/dev where dev matches the **SAP HANA Target Tenant Database Name**.

b. SAP HANA Target Tenant Database Name*: This refers to the target tenant database name which will be created upon VDB creation.
c. Target SystemDB User Name*: This refers to the SystemDB username in the target database.
d. Target SystemDB Password*: This refers to the SystemDB password in the target database.
e. Target Tenant Database ADMIN User Name*: This refers to the username of the tenant system user.
f. Target Tenant Database ADMIN Password*: This refers to the password of the tenant system user.
Note:
The SYSTEM username and password derived from the physical source database should be entered during the provision of a VDB because those credentials will be recovered during provision.
g. Configure Tenants Service and Port: This refers to the list of tenant services and port numbers in the following format.

```
<service>:<port>
For example, indexserver:30043
```

You can click "**+Add**" to add more services.
Note:

- This is not a mandatory field. If a user doesn't provide a value to this, then the plugin will proceed with the above fields and it will allocate the service and port on its own.
- If an invalid service is provided, the plugin will proceed with input, but will eventually fail while creating the service and will display an error on the Delphix Engine UI.

12. In the configuration section
    a. Set the vFile Name or accept the default, which is a random string. Delphix strongly recommends applying a consistent naming convention. We suggest this name should match the Target Tenant Name from the previous pane. Maintaining a consistent naming convention will support administration and troubleshooting activities. In the example shown, the vFile Name “dev” matches the actual tenant name and mount point from the previous pane.
    b. Select a **Target Group** for the VDB.Click the **Add Dataset Group** icon to add a new group if required.
    c. Set **Auto VDB Restart** option to automatically restart the VDB, where the target host is rebooted. Delphix recommends enabling this setting.
13. Click **Next**.
14. VDB **Snapshot Policy**. Select the Snapshot policy if an automated VDB snapshot schedule is required. Optionally, the Snapshot policy may be disabled by selecting “None”.
15. Click **Next**.
16. **Masking**. This page offers the opportunity to link the provision job to a masking job.

17. **Hooks**. Run a customized shell script via a hook operation at a particular moment in the provision operation.
18. Click **Next**.
19. **Summary.** Presents an opportunity to review all the information as a part of the initial provision workflow.
20. Click **Submit**.Once a provision job is submitted it is possible to monitor its progress by selecting "Active Jobs" in the top menu bar. On successful completion, VDB will be visible in the list of Datasets under its assigned group.

9.11.5.18 Next steps

Congratulations! You have provisioned your first virtual database!

Now, perform some simple functional tests with your application. You can connect your app to the VDB using standard TNS/JDBC techniques. Delphix has already registered the VDB for you on the target listener.

We suggest the following next steps:

1. Drop a table and use the VDB Rewind feature to test the recovery of your VDB.
2. Take a snapshot of your dSource and refresh your VDB to quickly get fresh production data.
3. Provision a new VDB from your VDB to test the sharing data quickly with other teams.
4. Mask your new VDB to protect sensitive data. Provision of new VDBs from that masked VDB to quickly provide safe data to development and QA teams.

9.11.6 SAP HANA support and requirements

This section covers the following topics:

- SAP HANA matrix (see page 1982)
- Requirements for SAP HANA staging hosts and databases (see page 1988)
- Requirements for SAP HANA target hosts and databases (see page 1989)
- Network and connectivity requirements for SAP HANA environments (see page 1990)
- Sudo requirements for the SAP HANA 2.0 plugin (see page 1990)

9.11.6.1 SAP HANA matrix

9.11.6.1.1 Plugin installation

Plugins to support SAP HANA virtualization are not automatically included in the Virtualization Engine deployment and are installed by Delphix Professional Services or Technical Support. A support case can be opened requesting installation by following this link[601].

601 https://support-tickets.delphix.com/

9.11.6.1.2 Migration and compatibility

This section describes the SAP HANA (DBMS) versions that are supported by the SAP HANA 2.0 Plugin, as well as the compatible operating systems (OS), for use on the source and target environments.

> Delphix Support Policies specifically list Major and Minor release coverage. If a minor release is listed as covered, then all patch releases under that minor release are certified.

9.11.6.1.3 Supported DBMS versions

SAP HANA version (Standard and enterprise editions)	Support package stack (SPS)
2.0	06
2.0	07

> A target may have a higher SPS release than a source; the reverse is not supported.

Key:

Color	Supported?
Y	Yes
N	No

9.11.6.1.4 Supported operating system

Supported OS version	Supported DBMS version		

	SAP HANA 2.0 SPS 04 (Standard and enterprise editions)	SAP HANA 2.0 SPS 05 (Standard and enterprise editions)	SAP HANA 2.0 SPS 06 (Standard and enterprise editions)	SAP HANA 2.0 SPS 07 (Standard and enterprise editions)
RHEL 7.3/7.4/7.5	Y	N	N	N
RHEL 7.6	Y	Y	Y	Y
RHEL 7.9	Y	Y	Y	Y
RHEL 8.1	Y	Y	Y	Y
RHEL 8.6	N	Y	Y	Y
SUSE 12 SP 03	Y	N	N	N
SUSE 15 SP 02	N	Y	Y	Y

9.11.6.1.5 Plugin/Delphix Engine compatibility

The plugin should be installed on compatible Delphix Engines per the table below:

Delphix Engine	Lua-based plugin						vSDK-based Plugin									
	4.1.1	4.1.2	4.2.0	4.3.0	4.3.1	4.3.2	6.0.0	6.0.1	6.1.0	6.1.1	6.2.0	6.2.1	6.3.0	6.3.1	6.3.2	6.3.3
6.0.0.x	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N
6.0.1.x	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N

6.0.1.x	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N
6.0.3.x	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N
6.0.4.x	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N
6.0.5.x	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N
6.0.6.x	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N
6.0.7.x	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N
6.0.8.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N
6.0.9.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N
6.0.10.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N
6.0.11.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N
6.0.12.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N
6.0.13.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N
6.0.14.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N
6.0.15.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N

6.0.16.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N
6.0.17.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N
7.0.0.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N
8.0.0.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N
9.0.0.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N
10.0.0.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
11.0.0.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
12.0.0.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
13.0.0.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
14.0.0.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
15.0.0.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
16.0.0.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
17.0.0.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
18.0.0.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

9.11.6.1.6 Plugin upgrade path

Upgrade from SAP HANA Direct (Lua-based) plugin to SAP HANA Staged (VSDK-based) plugin is not supported.

9.11.6.1.6.1 SAP HANA (Lua-based) plugin upgrade path

Your version	Recommended upgrade path to 4.3.2
3.4.5	Upgrade to version 3.5.1 and then upgrade to 4.0.0 and then to 4.1.1 and then to 4.1.2 and then to 4.2.0
3.5.0	Upgrade to version 3.5.1 and then upgrade to 4.0.0 and then to 4.1.1 and then to 4.1.2 and then to 4.2.0
3.5.1	Upgrade to version 4.0.0 and then to 4.1.1 and then to 4.1.2 and then to 4.2.0
4.0.0	Upgrade to version 4.1.1 and then to 4.1.2 and then to 4.2.0
4.1.1	Upgrade directly to version 4.1.2 and then to 4.2.0
4.1.2	Upgrade directly to version 4.2.0
4.2.0	Upgrade directly to versions 4.3.1 and 4.3.2

9.11.6.1.6.2 SAP HANA (VSDK-based) plugin upgrade path

To upgrade from the existing SAP HANA Staged (VSDK-based) plugin follow the recommended path below.

Your version	Recommended upgrade path to 6.3.2
6.0.0 6.0.1	Upgrade to 6.1.0 and follow the path below

Your version	Recommended upgrade path to 6.3.2
6.1.0 6.1.1	Upgrade to 6.2.0 and follow the path below
6.2.0 6.2.1	Upgrade directly to 6.3.3
6.3.0 6.3.1 6.3.2	Upgrade directly to 6.3.3

9.11.6.1.7 Compatibility Limitations

The following functionalities, users are **not** supported:

- Multiple duplicate services.
- Point-in-Time (PiT) recovery.
- Scale-out source and target environments.
- High isolation level.
- Partitioning.
- Virtual to Physical (V2P).
- We have not validated SAP HANA Stream Replication (HSR), but it should work for the Staging Push ingestion mechanism.
- Pull-based ingestion mechanism using third-party backups tools.
- More than one instance of the same SAP HANA service.
- Switching between the Staging Push to Pull model.
- Non-database users/Low Privileged users/Users for which privileges need to be elevated.

9.11.6.2 Requirements for SAP HANA staging hosts and databases

Backup of the tenant must be present which needs to be recovered. Along with backup, metadata information generated by script create_backup_metadata.sh is also expected in the backup location. As we will be recovering a tenant through backup files the important thing while creating the dSource which should be taken care of is the user should have access to the backup directory where the backup file resides.

General staging Host Requirements

- HANA database must be installed and running.
- Read-write permission on the backup directory.
- There must be an operating system user which has HANA administrative user privileges :
    - The Delphix Engine must be able to make an SSH connection to the staging environment using the operating system user.

    - The operating system user must have read and execute privileges on the HANA binaries installed on the staging environment.
    - The operating system user must have read, write, and execute access to the HANA data directories on the staging environment.
- A plugin directory must exist on the staging host. The directory must have the following properties:
    - Should be writable by the operating system user described above.
    - Should have at least 256 MB of available storage.
- TCP/IP connectivity to and from the staging environment must be configured as described in General Network and Connectivity Requirements (see page 717)
- Hostname and IP address must be correctly configured.

> ⚠ Non-database users/Low Privileged users/Users for which privileges need to be elevated, are not supported for performing Delphix operations.

9.11.6.3 Requirements for SAP HANA target hosts and databases

This section describes user privileges and other requirements for HANA target hosts and databases collectively referred to as the target environment.

A **target environment** is where virtualized databases run. A target environment must have a HANA instance installed and configured. The VDB will become a tenant on the target database instance.

9.11.6.3.1 General target environment requirements

1. The operating system of the target environment must match the staging environment.
2. The SAP HANA installation on the target environment must have the same or higher version than the staging environment.
3. There must be an operating system user with the following privileges:
    a. The Delphix Engine must be able to make an SSH connection to the target environment using the operating system user.
    b. The operating system user must have read and execute privileges on the HANA binaries installed on the target environment.
    c. The operating system user must have permission to run mount and unmount as the superuser via sudo with neither a password nor a TTY. See Sudo Requirements for the HANA 2.0 Plugin (see page 1990)
4. Prior to the discovery, a plugin directory on the target environment must exist with the following properties:
    a. The plugin directory must be writable by the operating system user mentioned above.
    b. The plugin directory must have at least 256 MB of available storage.

5. There must be a mount point directory (for example, /mnt) that will be used as the base for mount points that are created when provisioning a VDB with the following properties:
    a. The mount point directory must be writable by the operating system user mentioned above.
    b. The mount point directory should be empty.
6. TCP/IP connectivity to and from the source environment must be configured as described in General Network and Connectivity (see page 717) Requirements.
7. The hostname and IP address must be correctly configured.
8. HANA instance on which the VDB creation is being done should have the empty ports to host the services for the tenant database. The default port number range for tenant databases is 3<instance>40—3<instance>99. This means that the maximum number of tenant databases that can be created per instance is 20. However, you can increase this by reserving the port numbers of further instances. You do this by configuring the property [multidb] reserved_instance_numbers in the global.ini file. The default value of this property is 0. If you change the value to 1, the port numbers of one further instance are available (for example, 30040—30199 if the first instance is 00). If you change it to 2, the port numbers of two further instances are available (for example, 30040—30299 if the first instance is 00). And so on.

> ⚠ Non-database users/Low Privileged users/Users for which privileges need to be elevated, are not supported for performing Delphix operations.

9.11.6.4 Network and connectivity requirements for SAP HANA environments

Please see General Network and Connectivity Requirements (see page 717) and Architecture Best Practices for Network Configuration (see page 2331) for detailed network and connectivity requirements for the Delphix Engine, including connection requirements, port allocation, and firewall and Intrusion Detection System (IDS) considerations.

9.11.6.5 Sudo requirements for the SAP HANA 2.0 plugin

The following sudo privileges are requirements for the HANA 2.0 Plugin.

Privilege	Source	Target	Rationale
mkdir/rmdir	Not Required	Required	Delphix dynamically makes and removes directories under the provisioning directory during VDB operations. This privilege is optional, provided the provisioning directory permissions allow the delphix os user to make and remove directories.

Privilege	Source	Target	Rationale
mount/umount	Not Required	Required	Delphix dynamically mounts and unmounts directories under the provisioning directory during VDB operations. This privilege is required because `mount` and `umount` are typically reserved for superuser.

It is required to specify the NOPASSWD qualifier within the "sudo" configuration file. This ensures that the "sudo" command does not demand the entry of a password.

Example with user hdbadm

```
Defaults:hdbadm !requiretty
hdbadm ALL=NOPASSWD:/sbin/mount, /sbin/umount, /bin/mkdir, /bin/rmdir
```

9.11.6.5.1 Example of limiting `sudo` access for the Delphix OS user

In situations where security requirements prohibit giving the Delphix user root privileges to mount, unmount, make directory, and remove directory on the global level, it is possible to configure the `sudoers` file to provide these privileges only on specific mount points or from specific Delphix Engines, as shown in the below example.

```
Defaults:hdbadm !requiretty
hdbadm ALL=(root) NOPASSWD: \
/bin/mount    *<delphix-server-name>*   /mnt/provision/*, \
/bin/umount    /mnt/provision/*, \
/bin/mkdir * /mnt/*
```

that the following example is for illustrative purposes and the sudo file configuration options are subject to change.

9.11.6.5.2 Example 1

This example restricts the **delphix_os** user's use of `sudo` privileges to the directory `/hana`.

Note that wildcards are allowed for the options on `mount` and `umount` because those commands expect a fixed number of arguments after the options. The option wildcard on the `mount` command also makes it possible to specify the file-system being mounted from the Delphix Engine.

However, wildcards are not acceptable on `mkdir` and `rmdir` because they can have any number of arguments after the options. For those commands, you must specify the exact options (`-p` , `-p -m 755`) used by the Delphix Engine.

Example /etc/sudoers File Configuration on the Target Environment for sudo Privileges on the VDB Mount Directory Only (Linux OS)

```
Defaults:delphix_os !requiretty
delphix_os ALL=(root) NOPASSWD: \
/bin/mount  *        /hana/*, \
/bin/umount *        /hana/*, \
/bin/umount          /hana/*, \
/bin/mkdir -p        /hana/*, \
/bin/mkdir -p -m 755 /hana/*, \
/bin/mkdir           /hana/*, \
/bin/rmdir           /hana/*
```

9.11.6.5.3 Example 2

This example restricts the **delphix_os** user's use of `sudo` privileges to the directory / `hana` , restricts the mount commands to a specific Delphix Engine hostname and IP, and does not allow user-specified options for the `umount` command.

This configuration is more secure, but there is a tradeoff with deployment simplicity. This approach would require a different sudo configuration for targets configured for different Delphix Engines.

A Second Example of Configuring the /etc/sudoers File on the Target Environment for Privileges on the VDB Mount Directory Only, and Allows Mounting Only from a Single Server (Linux OS)

```
Defaults:delphix_os !requiretty
delphix_os ALL=(root) NOPASSWD: \
/bin/mount           <delphix-server-name>* /hana/*, \
/bin/mount *         <delphix-server-name>* /hana/*, \
/bin/mount           <delphix-server-ip>*   /hana/*, \
/bin/mount *         <delphix-server-ip>*   /hana/*, \
/bin/mount "", \
/bin/umount          /hana/*, \
/bin/umount *        /hana/*, \
/bin/mkdir [*]       /hana/*, \
/bin/mkdir           /hana/*, \
/bin/mkdir -p        /hana/*, \
/bin/mkdir -p -m 755 /hana/*, \
/bin/rmdir           /hana/*
```

9.11.7 Managing SAP HANA environments and hosts

This section covers the following topics:

- Workflow for SAP HANA environments (see page 1993)
- Adding a SAP HANA environment (see page 1993)
- Editing SAP HANA environment details (see page 1996)
- Changing the hostname or IP address for SAP HANA staging and target environments (see page 1997)
- Steps to modify SAP HANA dSource properties (see page 1997)

9.11.7.1 Workflow for SAP HANA environments

A staging environment must be added to the Delphix Engine before linking. Similarly, a compatible target environment must be added to the Delphix Engine in order to provide a location to provision to.

Installation of the HANA Plugin is required before the discovery of a HANA environment can occur.

Once an environment is added to the Delphix Engine, environment discovery takes place. Environment discovery is a process that enables the HANA Plugin to determine HANA installation details on a host. The same Environment Discovery process can be repeated during an Environment Refresh in order to detect new HANA installations.

9.11.7.2 Adding a SAP HANA environment

This topic describes how to add a HANA environment.

9.11.7.2.1 Prerequisites

- Make sure that the HANA environment in question meets the requirements described in Requirements for HANA Staging Hosts and Databases (see page 1988) and Requirements for HANA Target Hosts and Databases (see page 1989)

9.11.7.2.2 Procedure

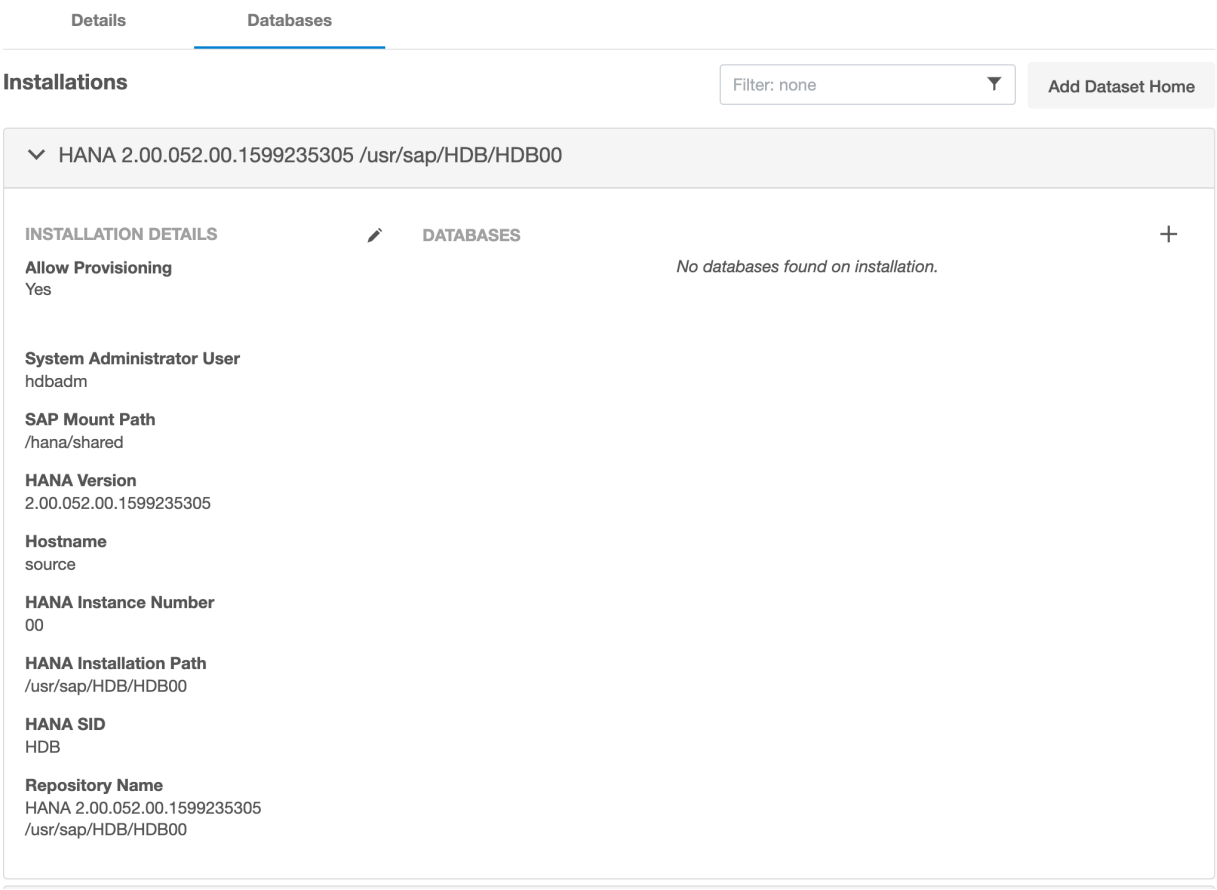
- Login to the **Delphix Management** application.
- Click **Manage**.
- Select **Environments**.
- Next to **Environments**, click the **Actions (...) menu and** select **Add Environment**.
- In the Add Environment dialog, In the **Host OS** select **Unix/Linux** on the menu.
- In **Server Type**, select **Standalone Host**.
- Click **Next**.
- Enter **Name** for the environment.
- Enter the **Host IP address** or **fully qualified hostname**.
- Enter the **SSH port**. The default value is 22.
- Enter an **OS Username** for the environment which should be ''.>'.>
- Select **Login Type**.
  Note:
  Currently, the Enterprise Password Vault option is not supported.
- Username and Password - enter the OS username and password
- Username and Public Key - enter the OS username.

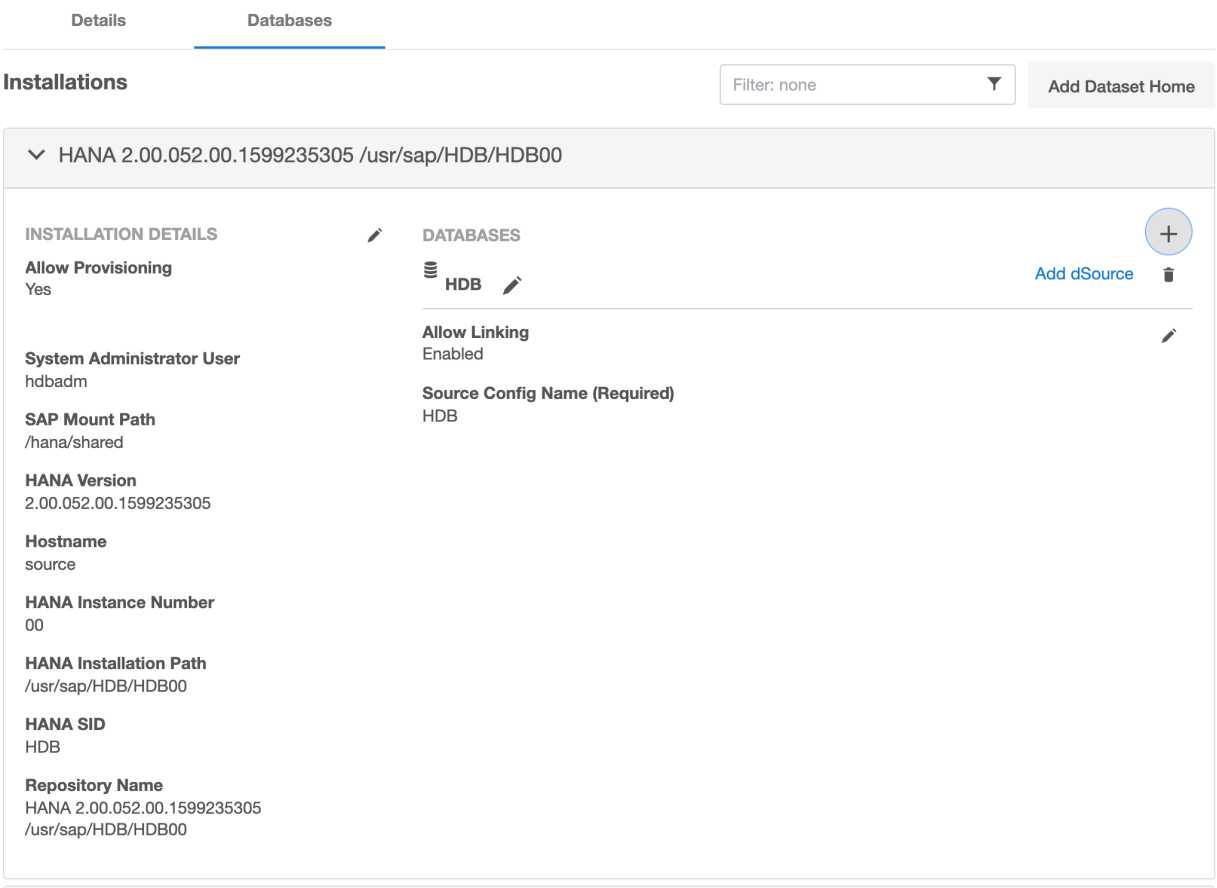
- Password Vault - select from an existing Enterprise Password Vault

> **Using public key authentication**
> If you want to use public-key authentication for logging into your Unix-based environment, there are two options: use the engine's key pair or provide a key pair for this environment.
> To use the engine's key pair:
>
> 1. Select **Public Key** for the **Login Type.**
> 2. Click **View Public Key**.
> 3. Copy the public key that is displayed, and append it to the end of your `~/.ssh/authorized_keys` file. If this file does not exist, you will need to create it.
> a. Run chmod 600 ~/.ssh/authorized_keys to allow only the file's owner to read and write to it (make sure the file is owned by the user).
> b. Run chmod 755 ~ to restrict access to the user's home directory so no other user may write to it.
> c. Run chmod 700 ~/.ssh so that others cannot write to it. The ~/.ssh directory cannot be writable by group or other users. Otherwise, authentication will fail.
>
> As an alternative, you can provide a key pair specific for this environment via the API or CLI. See CLI Cookbook: VDBs (see page 2618) for instructions.

1. For Password, enter the password associated with the user in step 11.
2. Select **Public Key** for the **Login Type.**
3. Click **View Public Key**.
4. Copy the public key that is displayed, and append it to the end of your `~/.ssh/authorized_keys` file. If this file does not exist, you will need to create it.
   a. Run chmod 600 ~/.ssh/authorized_keys to allow only the file's owner to read and write to it (make sure the file is owned by the user).
   b. Run chmod 755 ~ to restrict access to the user's home directory so no other user may write to it.
   c. Run chmod 700 ~/.ssh so that others cannot write to it. The ~/.ssh directory cannot be writable by group or other users. Otherwise, authentication will fail.
5. For Password Login, click 'Validate' to test the username and password.
6. Enter a **Plugin Path.** The location specified must be a string without spaces. Ensure this directory path exists on the host in question. Environment discovery will fail if the directory does not exist on the host.
7. To provide your own Oracle Java select the **Provide my own JDK** checkbox and click **Next**.
8. In the Java Development Kit tab enter the absolute path to your Oracle JDK and click **Next**. For more information, see Providing Your Own Oracle Java (see page 1025)
9. Click **Submit**. On success, the new environment will be added to the list in the Environments tab.

10. Select the “Databases” tab to view discovered HANA databases:





9.11.7.2.3 Post-requisites

To view information about an environment after you have created it:

1. Click **Manage**.
2. Select **Environments**.
3. Select the **environment name**.

9.11.7.3 Editing SAP HANA environment details

This topic describes how to edit the details of an environment. Once an environment has been added it is possible to change any of the following values:

- Environment OS user
- Host IP Address/Name
- SSH port
- Path to plugin location

9.11.7.3.1 Procedure

1. Login to the **Delphix Management** application with admin credentials or as the owner of an environment.
2. Click **Manage**.
3. Select **Environments**.
4. In the **Environments** panel, click the name of an environment to view its details.
5. Next to **Environment Detail**, click the **Pencil** icon to edit.
6. Click the **Check** icon to save your edits.

9.11.7.4 Changing the hostname or IP address for SAP HANA staging and target environments

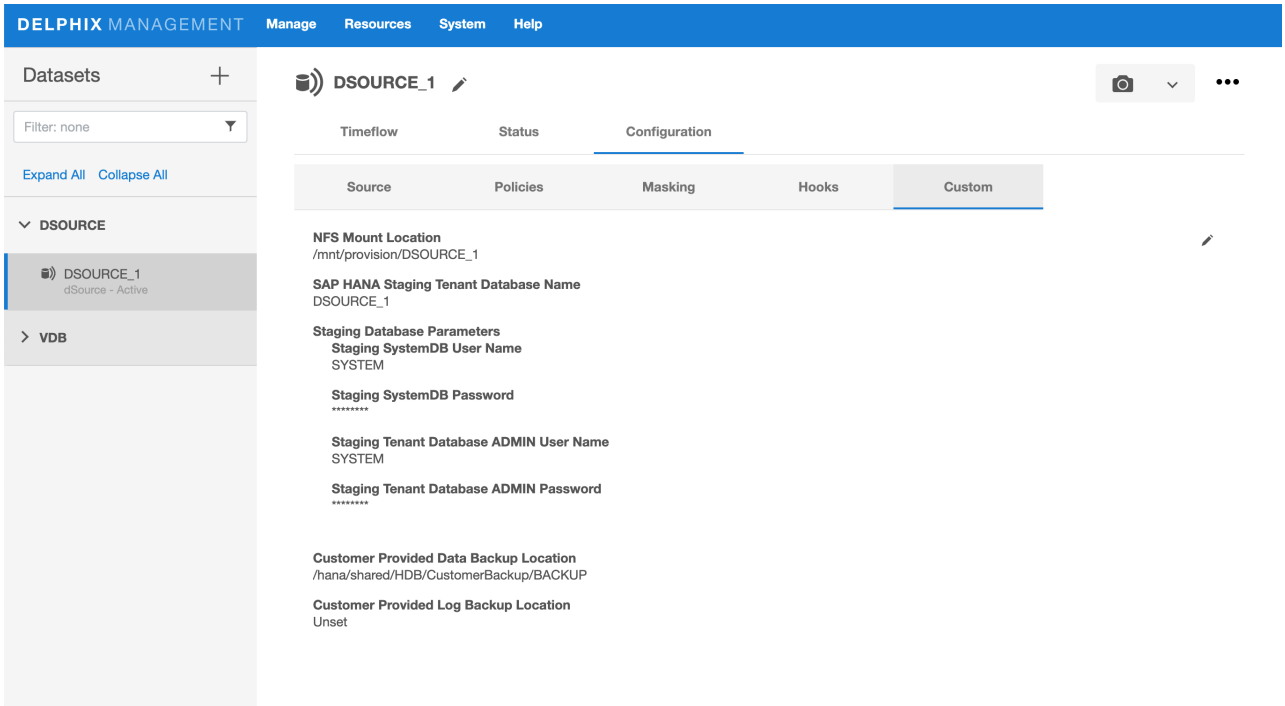
Below is the procedure to change the Host Address of the HANA Staging and Target Environments (If the Host address of the underlying staging and target server has also changed):

9.11.7.4.1 Procedure

1. Disable the dSources.
2. Disable the VDBs.
3. Login to the Delphix Management application.
4. Click **Manage**.
5. Select **Environments**.
6. Click on the existing environment name you want to modify and open the environment information screen.
7. In the Attributes section, click the **Pencil** icon located Next to it.
8. Update the Host address to the new one and save the results by clicking on
9. The New Host address will now be updated.
10. Re-enable the dSources.
11. Re-enable the VDBs.

9.11.7.5 Steps to modify SAP HANA dSource properties

Select the dSource and then the dSource **Configuration** tab to modify dSource properties. For example, to change the **Source SystemDB Username** or **Password**, select the **Dataset**, **Configuration,** and then the **Custom** tab as shown below:



9.11.8 Linking data sources and syncing data with SAP HANA

This section covers the following topics:

- SAP HANA dSource ingestion (see page 1998)
- Linking SAP HANA data sources (see page 1999)
- Staging push implementation details (see page 2002)
- Advanced data management settings for SAP HANA dSources (see page 2005)
- Working with SAP HANA dSource snapshots (see page 2006)
- Source sizing implementation for SAP HANA data sources (see page 2008)

9.11.8.1 SAP HANA dSource ingestion

This release of the HANA 2.0 plugin uses Native Backups/Third-party backups to stage a dSource.

An External Backup is a form of HANA Native Backup that was created by users outside of Delphix. The necessary backup files must exist prior to the dSource operation and must be prepared by the user prior to ingestion by Delphix. In particular, a Delphix-supplied script must be run to persist vital backup metadata to Delphix during the dSource creation process. The backup files must be created on the source HANA instance must be made visible to the Staging host where the dSource is being created (ex: via shared storage, scp, etc).

A new data ingestion mechanism, Staging Push has been introduced that will help users to push data into the Delphix provided mount point on their own.

There are two mechanisms to create dSource on the staging server.

- Pull architecture, where the HANA plugin pulls the data into the Delphix provided mount location.

- Staging Push architecture, where the user pushes the data into the Delphix provided mount point on their own.

9.11.8.2 Linking SAP HANA data sources

This topic describes basic concepts behind the creation of a dSource for a HANA database using conventional pull and Staging push mechanisms.

> For Staging Push, only native/third-party backups are used to populate data in the staging database.

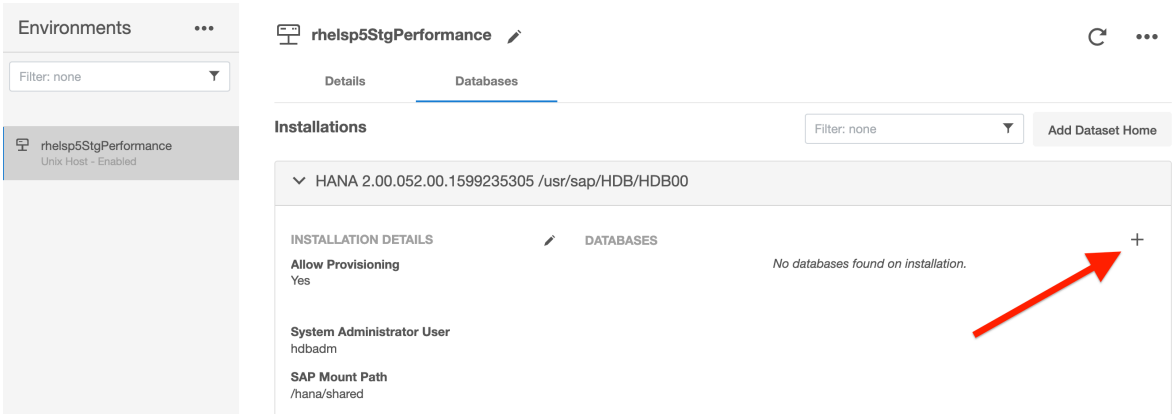
For Third-party Backups, users need to provide a “Backup Path” containing FULL and Incremental backups. These files will be copied to the Delphix Engine during the dSource linking process. In this scenario, the Delphix Engine will not require any credentials of the source database. Although credentials for the tenant database (for which backup has been placed) and system database of staging instances would be required.

9.11.8.2.1 Prerequisites for customer provided backups ingestion mechanism

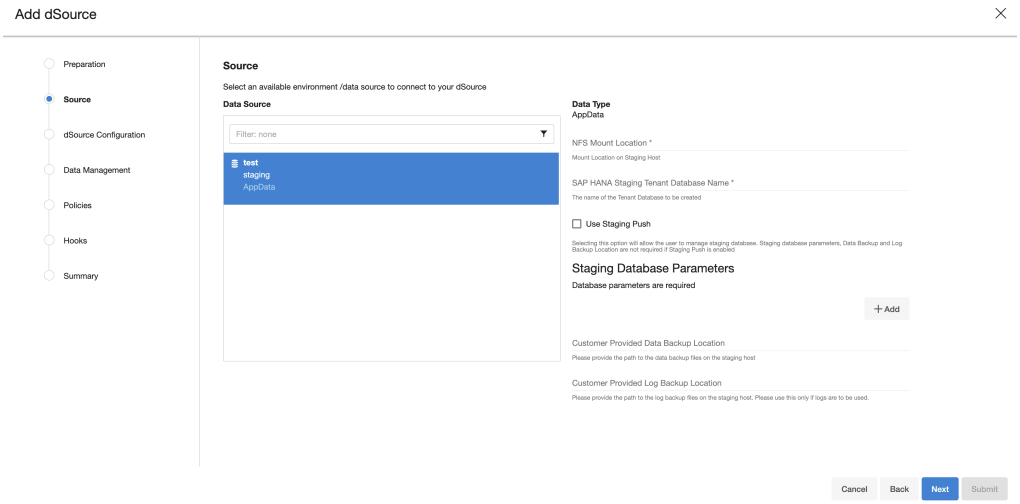
- Users need to run the create_backup_metadata.sh. This script will create the backupTimeInfo file at the same location where backups are kept. This script will gather all the required metadata information and will store that information in the backupTimeInfo file. This information will then get used by the plugin at the time of creating dSource.
- More details regarding the script have been included in the HANA 2.0 Plugin Tools (see page 1959)

9.11.8.2.2 Procedure

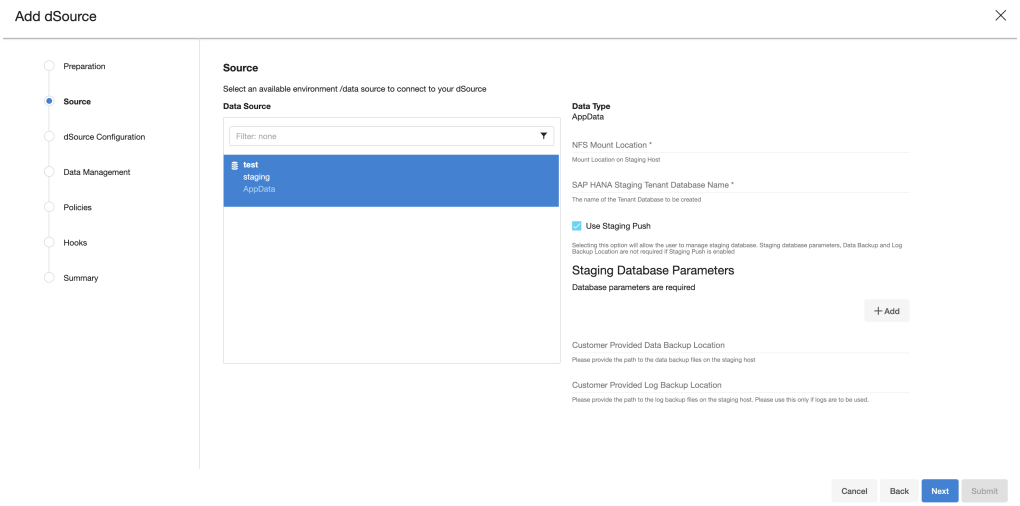
1. Login to the Delphix Management application.
2. Select **Manage > Environments**. Then select the Datasets tab.



3. Create a source configuration on which to build the dSource. You can do this by clicking on the **+** symbol. The Source configuration can have any name.
4. In the **Add dSource** wizard, select the required source configuration.
5. After selecting the source, below are the parameters required to create the dSource.
    a. Parameters required for the conventional **pull** mechanism:
        i. Mount Location on Staging Host
        ii. SAP HANA Staging Tenant Database Name
        iii. Source SystemDB User Name
        iv. Source SystemDB Password
        v. Customer Provided Data Backup Location: This represents the path to the location of the customer-provided backup files.
        vi. Customer Provided Log Backup Location: This represents the path to log the location of the customer-provided log backup files. This parameter is optional.

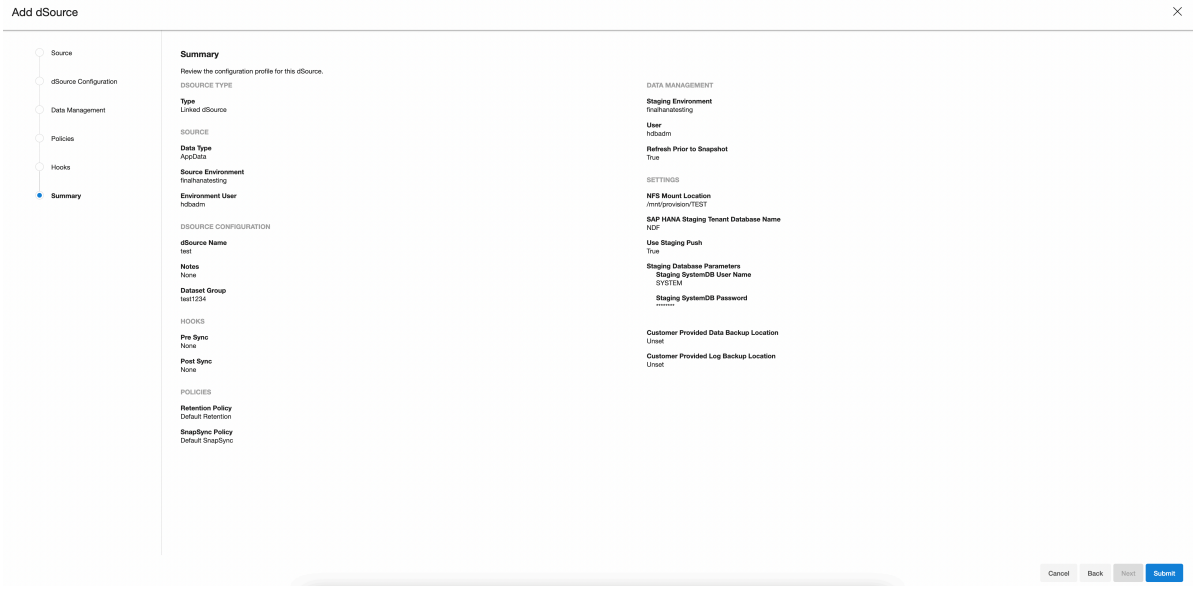


    b. Parameters required for the **Staging push** mechanism
        i. Mount Location on Staging Host
        ii. SAP HANA Source Tenant Database Name
        iii. Enable the **Staging Push** feature by selecting the checkbox "**Use Staging Push**". With this feature, the user handles the Source Tenant database and its state on their own and you don't need to specify any other staging database parameters, data backup, and log backup locations. If this check box is selected, the HANA plugin will not invoke the tenant creation and recovery operation and if this checkbox is not selected then the default pull approach will be used.



c. Click **Next.**

6. Enter a name and select a group for your dSource. Adding a dSource to a dataset group lets you set Delphix Domain user permissions for that database and its objects, such as snapshots. See the topics under Users and Groups (see page 674) for more information.
7. Configure the **SnapSync** and **Retention Policy**. The SnapSync policy can be set to 'None' for customers who wish not to run a scheduled SnapSync job.
8. Optionally, users can provide the bash script as a plugin-defined hook. Both Pre-Snapshot and Post-SnapShot hooks can be used.
9. Select **Next.**
10. At the end of the workflow, review the **dSource Configuration** in the **Summary** pane as shown below:



Once the dSource linking job is submitted it is possible to monitor progress by selecting **Active Jobs** in the top menu bar. On successful completion, the new dSource will be visible in the list of Datasets under its assigned group.

After dSource creation using the Staging Push mechanism, an empty snapshot will be created but users can not create VDB's from the same. Also, before creating the subsequent snapshots, users will have to create the database and point the database to use the Delphix provided mount point.

9.11.8.3 Staging push implementation details

This topic provides implementation details of the Staging Push for the HANA plugin. Staging Push will help users to push data into the Delphix provided mount point on their own.

Staging Push gives users control over some Staging DB processes. It will give control of the staging database to the user to pull from any backup provider. Staging database files will be stored on Delphix Storage. Delphix will still be the one snapshotting the underlying data files, and gathering the metadata required to provision from the snapshot.

The below steps show how to create a dSource using the Staging Push mechanism.

9.11.8.3.1 dSource creation

The first snapshot created as a part of dSource creation contains data and log directories within the Delphix created mount point. These directories are empty for the first snapshot.
The following snippet shows how the file system appears inside the Delphix created mount point.

```
hdbadm@target:/delphix/zfs> tree
.
├── data
└── log

2 directories, 0 files
hdbadm@target:/delphix/zfs> pwd
/delphix/zfs
```

> If a system user attempts to create a VDB from this snapshot, it will fail with the below error message.

Provision virtual database "N5RQVO".

Error
Data volumes are not found under the data directory. Please check that the dSource snapshot contains the valid data and log volumes required for VDB creation.

Error Code
exception.plugin.user.error.detected

Suggested Action
Please take a new snapshot of the dSource with valid data and log volumes and try again.

9.11.8.3.2 Subsequent Snapshots

The system user needs to follow the steps and commands below before taking any subsequent snapshots.

Note: The system user needs to provide the values for all the parameters provided within { }.

1. Create a database in "no start" mode. Here, "no start" implies that the database is not started.

   create database {DATABASE_NAME} SYSTEM USER {DATABASE_PASSWORD} no start.

   The parameters used in the below example are:
   a. DATABASE_NAME - Refers to the database name to be used for staging Push. This is the same name that was provided by the system user on the UI while creating the dSource.
   b. DATABASE_PASSWORD - Refers to the password to be used for the staged database. Example:

   ```
   hdbsql SYSTEMDB=> create database TEST SYSTEM USER PASSWORD Delphix_123 no start;
   0 rows affected (overall time 20.534486 sec; server time 20.531628 sec)
   ```

2. Set the HANA database to use the Delphix created mount point to store the data and log volume files. This is done by setting the `basepath_datavolumes` and `basepath_logvolumes` to *data* and *log*directory under the mount point, respectively. All the tenant data volumes and log volumes reside in the single mount point provided by Delphix.

   ```
   //This query sets the value of basepath_datavolumes to within the delphix
   storage.
   alter system alter configuration('global.ini', 'DATABASE','{DATABASE_NAME}')
   SET ('persistence','basepath_datavolumes') = '{DELPHIX_MOUNT_LOCATION}/data'
   ```

   ```
   //This query sets the value of basepath_logvolumes to within the delphix
   storage.
   alter system alter configuration('global.ini', 'DATABASE','{DATABASE_NAME}')
   SET (''persistence','basepath_logvolumes') = '{DELPHIX_MOUNT_LOCATION}/log'
   ```

Example:
Once the `basepath_datavolumes` and `basepath_logvolumes` parameters are set for the tenant, then their values are reflected in the `global.ini` file in the tenant database. The location of the global.ini file is `<sapmnt>/<SID>/SYS/global/hdb/custom/config/DB_<tenantdbname>.` Example:

```
# global.ini last modified 2021-12-01 12:25:11.693519 by hdbnameserver
 [persistence]
basepath_datavolumes = /delphix/zfs/data
basepath_logvolumes = /delphix/zfs/log
```

3. Create a backup catalog. This is used by HANA to determine whether recovery is possible and which backups are to be used to recover the database.

   Note: Creating a backup catalog is not applicable for third-party backups. For example, Commvault-based backups.

   The parameters used in the below example are:
   a. logDirs - Refers to the location where logs are present for the database. If the system user does not intend to use logs then an empty directory can be created and the location of the empty directory can be provided here.
   b. dataDir - Refers to the location where backups are present.
   c. -d - Refers to the location where the catalog backup will be created.
      Example:

4. Perform recovery on the tenant database. This can be done in the following two ways:
   a. Recovery using data backups only The following parameters for the recovery command are listed below:
      i. Recover database for {DATABASE_NAME} - Represents the name of the database for which recovery is to be performed.
      ii. Until timestamp '{TIMESTAMP}' - Represents the timestamp value till which recovery is to be performed.
      iii. Clear log using catalog path ('{CATALOG_PATH}') - Refers to the path where the catalog backups are kept.
      iv. Using data path ('{DATA PATH}') - Refers to the path where the data backups are kept.
      Example:

      ```
      hdbsql SYSTEMDB=> recover database for TEST until timestamp '2021-09-16 08:41:50.3
      11000000' clear log using catalog path ('/usr/sap/HDB/HDB00/source_backup') using
      data path ('/usr/sap/HDB/HDB00/source_backup') ignore workergroups;
      0 rows affected (overall time 63.235920 sec; server time 63.232622 sec)
      ```

      Recovery using both data and log backup Along with the above parameters, you need to specify the below parameter:
      v. Using log path ('{LOCATION_OF_THE_LOG PATH}') - Refers to the location of log backups.
   b. Recovery using third-party data backups. For example, Commvault-based backups. The following parameters for the recovery command are listed below:

i. STAGING_DATABASE_NAME - Represents the name of the tenant database to be created on the staging area.
ii. RECOVERY_TIMESTAMP - Represents the time limit until which recovery has to be performed. The time has to be in the YYYY-MM-DD HH:MM:SS format.
iii. SOURCE_TENANT_DATABASE - Represents the name of the source tenant database.
iv. SOURCE_SID - Represents the SID of the source HANA installation. Example:

```
recover database for ${STAGING_DATABASE_NAME} until timestamp
'${RECOVERY_TIMESTAMP}' clear log using source
'${SOURCE_TENANT_DATABASE}@${SOURCE_SID}' check access using backint
ignore workergroups;
```

Once the recovery is done, data will be available inside the Delphix mount point and the database will be in an active state. The below snippet shows an example file system after the recovery.

```
hdbadm@target:/delphix/zfs> tree
.
├── data
│   └── mnt00001
│       ├── hdb00002.00004
│       │   ├── datavolume_0000.dat
│       │   └── __DO_NOT_TOUCH_FILES_IN_THIS_DIRECTORY__
│       └── hdb00003.00004
│           ├── datavolume_0000.dat
│           └── __DO_NOT_TOUCH_FILES_IN_THIS_DIRECTORY__
└── log
    └── mnt00001
        ├── hdb00002.00004
        │   ├── __DO_NOT_TOUCH_FILES_IN_THIS_DIRECTORY__
        │   ├── logsegment_000_00000000.dat
        │   ├── logsegment_000_00000001.dat
        │   └── logsegment_000_directory.dat
        └── hdb00003.00004
            ├── __DO_NOT_TOUCH_FILES_IN_THIS_DIRECTORY__
            ├── logsegment_000_00000000.dat
            ├── logsegment_000_00000001.dat
            └── logsegment_000_directory.dat

8 directories, 12 files
```

You can now proceed to take a snapshot and the VDB creation using the new snapshot.

9.11.8.4 Advanced data management settings for SAP HANA dSources

This topic describes advanced data management settings for dSources.

When linking a dSource, you can use custom data management settings to improve overall performance and match the needs of your specific server and data environment. If no specific settings are required, leverage default data management settings.

9.11.8.4.1 Accessing data management settings

There are three ways to set or modify data management settings for dSources:

1. During the dSource linking process, in the Policies tab of the Add dSource wizard select the **Retention Policy**.
2. Or select **Manage**, then select **Policies**.
3. In the Policies screen select the **Retention** tab. To manage retention on target after the dSource snapshots have been deleted from the replication source, select the **Replica Retention** tab.
4. To add a new retention policy click the **+Retention** button, and to add a new replica retention policy on the replication target, click the **+Replica Retention** button.
5. This will open the Retention or the Replica Retention Policy wizard, depending upon the above selection. Enter a policy name along with required retention details and click **Submit**. For more information, see Policies for Scheduled Jobs (see page 1068).

9.11.8.4.2 Retention policies - retention/replica retention

Retention policies define the length of time Delphix Engine retains snapshots within its storage. To support longer retention times, you may need to allocate more storage to the Delphix Engine. The retention policy – in combination with the SnapSync policy – can have a significant impact on the performance and storage consumption of the Delphix Engine.

Replica Retention policies define how long the snapshots are retained on replicated namespaces for dSources and VDBs after they have been deleted on the replication source. Normally, the snapshots that have been deleted on the replication source engine are also deleted on the replication target engine. A new retention policy is introduced to provide an extended lifetime of such snapshots on the replication target.

9.11.8.4.3 Benefits of longer retention

Increased retention times for snapshots extends the quantity of historical data the Delphix Engine will retain.

Common use cases for longer retention include:

- SOX compliance
- Frequent application changes and development
- Caution and controlled progression of data
- Reduction of project risk
- Restore to older snapshots

9.11.8.5 Working with SAP HANA dSource snapshots

Taking a snapshot creates a new snapshot entry in the dSource's Timeflow.

Snapshot (default)

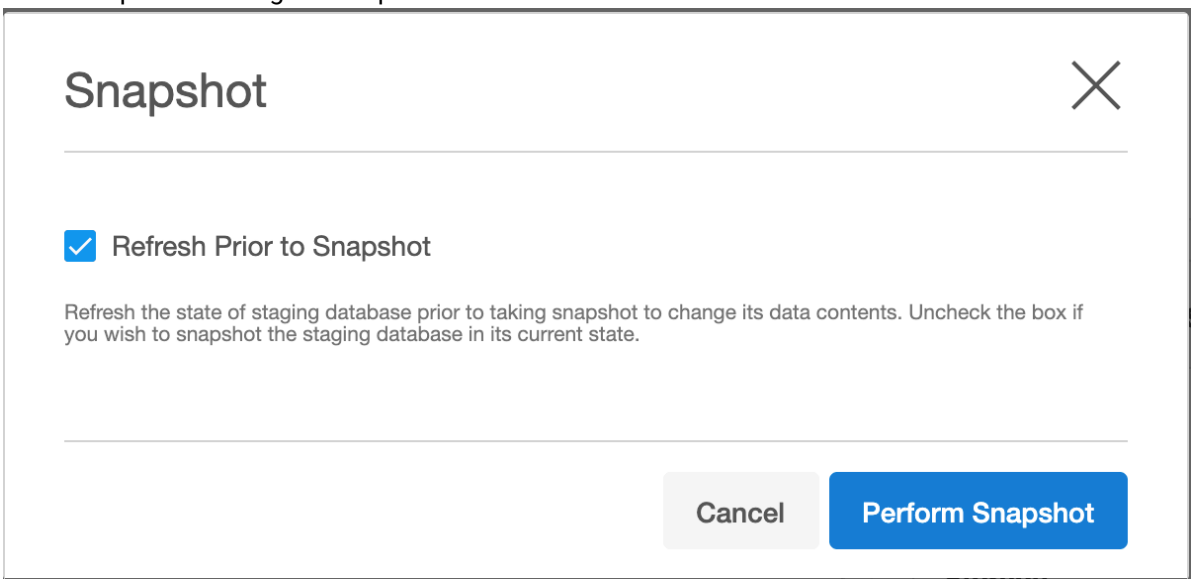
This section lists the steps to take a snapshot and delete the same.

1. Login to the **Delphix Management** application.
2. Click **Manage** and select **Datasets** from the dropdown list.
3. Select the dSource you want to Snapshot.
4. Click the **Camera** icon. Alternatively, click the arrow next to the Camera icon and select **Snapshot (default)**.
5. From the Snapshot dialog, select **Perform Snapshot**.
6. Click **View:** under **Timeflow** to verify the Snapshot you just created.
7. To delete, select the snapshot you just created, and from the Actions menu (...) select **Delete Snapshot**.
8. In the Delete Snapshot dialog, select **Delete**.
9. Click **View:** under **Timeflow** to verify the Snapshot, you just deleted.

Snapshot with parameters

This section lists the steps to take a snapshot with parameters and delete the same.

1. Login to the **Delphix Management** application.
2. Click **Manage** and select **Datasets** from the dropdown list.
3. Select the dSource you want to Snapshot.
4. Click the arrow next to the Camera icon and select **Snapshot with Params...**.
5. Select **Refresh Prior to Snapshot** to refresh the staging database. This will refresh the staging database prior to taking the snapshot.



6. In the Snapshot dialog, select **Perform Snapshot**.
7. Click **View:** under **Timeflow** to verify the Snapshot you just created.
8. To delete, select the snapshot you just created, and from the Actions menu (...) select **Delete Snapshot**.

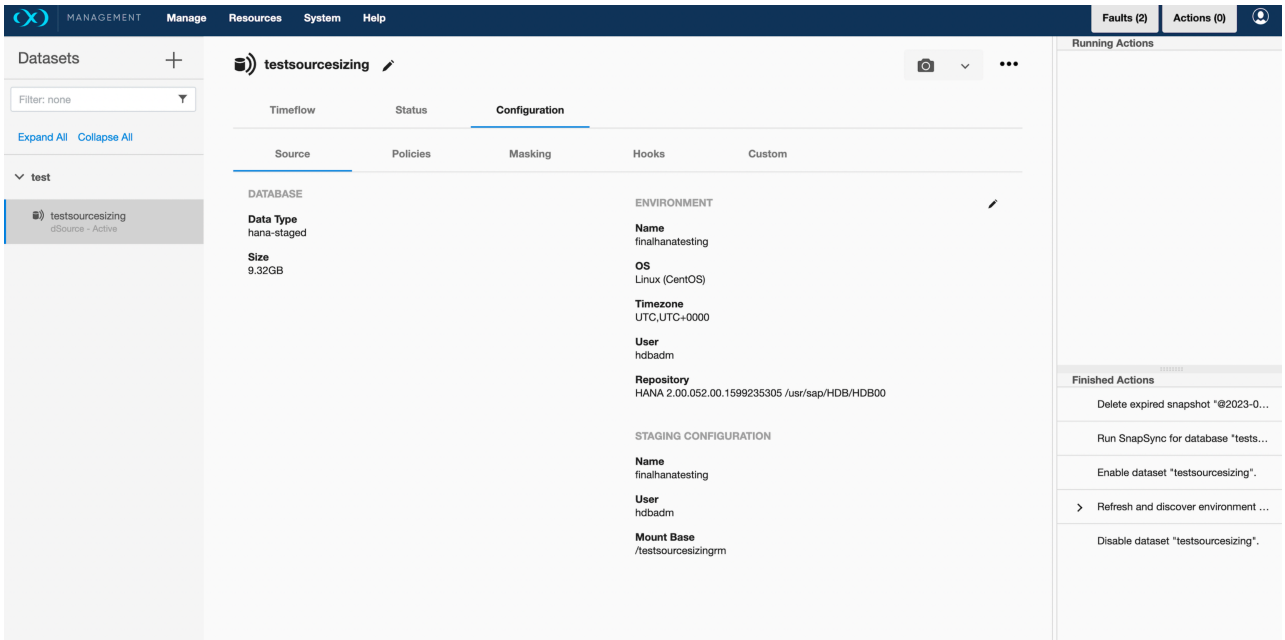
9. Click **View:** under **Timeflow** to verify the Snapshot, you just deleted.

9.11.8.6 Source sizing implementation for SAP HANA data sources

The source sizing feature aligns the data source experience with Delphix database standards that will improve your ingestion reporting experience on the per TB pricing model. This feature enables you to calculate the dataset size based on the total allocated space on the mount point provided by the Delphix Engine for the dataset.

9.11.8.6.1 Procedure

- For all datasets, the size of the dataset can be calculated using `du -sb <dataset>` command that provides the correct volume and apparent size of the datasets.
- Source sizing is implemented for below methods:
  - For the dSource created using external backup ingestion method
  - For the dSource created using staging push method



9.11.9 Provisioning and managing VDBs from SAP HANA dSources

This section covers the following topics:

- Provisioning SAP HANA VDBs: an overview (see page 2009)
- Provisioning SAP HANA VDBs (see page 2010)
- Working with SAP HANA VDB snapshots (see page 2013)
- Failure during provision (see page 2013)
- SAP HANA hook operations (see page 2014)

9.11.9.1 Provisioning SAP HANA VDBs: an overview

This topic describes the basic concepts involved with provisioning of VDBs in HANA.

Provisioning of a VDB results in creation of a new tenant database in the specified HANA target environment, with storage for the new tenant backed by Delphix storage. The data inside the tenant reflects the same data in that exists in the dSource snapshot from which the VDB is being provisioned. During the provision process, Delphix will create a filesystem in Delphix specifically for the new VDB, populate the filesystem with the appropriate data files and datafile contents, then share and mount that filesystem onto the specified HANA target host. The new HANA tenant will have its storage paths re-configured to use a single subdirectory specific to the tenant, then traverse an NFS mount which points to the filesystem on Delphix for that tenant. The VDB provision operation is very fast because Delphix does not move nor copy data to create and populate the VDB filesystem.

During the VDB creation, the location of the basepath_datavolumes and basepath_logvolumes will be changed to use the storage inside the Delphix Engine. This will mean that any new services (even if a user creates a service post VDB creation by the plugin) which will now be created for the virtualized tenant will also be created on the tenant-specific filesystem on the Delphix Engine.

9.11.9.1.1 Setting log mode overwrite

As a part of the VDB creation process, if the correct tenant username and password are provided, then the log mode for the virtualized tenant will be set to 'overwrite'. It means that "No log backups" are created. When savepoints are written, log segments are immediately freed to be overwritten by new log entries.

9.11.9.1.2 HANA port control

Starting Delphix Engine 6.0.10.0 and HANA vSDK plugin, Delphix introduces a solution that keeps the port numbers consistent throughout the VDB life cycle so that the connections made to the VDBs are not disrupted during their life cycle.

9.11.9.1.2.1 Need for user specified ports?

Until now, Delphix HANA was dependent on the HANA-provided ports for any VDB. Every VDB has its own ports and connections for internal and external communication. When operations such as rewind, refresh, or enable are performed on VDBs, this can cause port instability or inconsistency problems because the port numbers could change even when the VDB name remains the same. The below points further elaborates on this limitation.

- The plugin uses HANA-provided port numbers which are selected from the available pool of port numbers in the bottom-up approach that means an idle port number at the bottom will be picked first for use.
- The plugin does not have any mechanism to remember these port numbers and use them from the pool at the time of creating any tenant database again at the time of enabling, refresh, or rewind.
  So if we had a tenant TEST having a service indexserver at 30043 port number, then after the refresh, rewind or enable, there will be a chance that
  a. The same port number(30043) is not available
  b. Any port number lesser than the one being used earlier(30043) is now available.

So, there are chances that the tenant TEST is provided with another port number for the same service by HANA and if there was any connection made to the tenant TEST at 30043 will now be disrupted. Hence there is a need for a solution that allows the database to keep the port number consistent throughout the VDB life cycle.

9.11.9.1.2.2 How does the HANA port allocation works?

The port control feature now provides an option to specify port numbers at the time of VDB creation. This can be done in the following two ways.

User-specified HANA ports

See the procedure for Provisioning HANA VDBs (see page 2009) where the user can specify the HANA service and port number at the time of VDB creation.

By Plugin itself

If the service and port numbers are not provided by the user on the Plugin user interface, then the plugin itself will decide the port number and will keep the ports consistent throughout the life of a VDB.

This is how the plugin chooses the ports:

- It is based on the value of the `reserved_instance_number` parameter in the `global.ini` file. This file is located at `/hana/shared/{DATABASE_SID}/global/hdb/custom/config.` If this parameter is not defined in global.ini, then the value of the instance number will be used.
- Ports will be decided by the HANA plugin in a top-down manner which means that the topmost port which is free will be used first and so on.

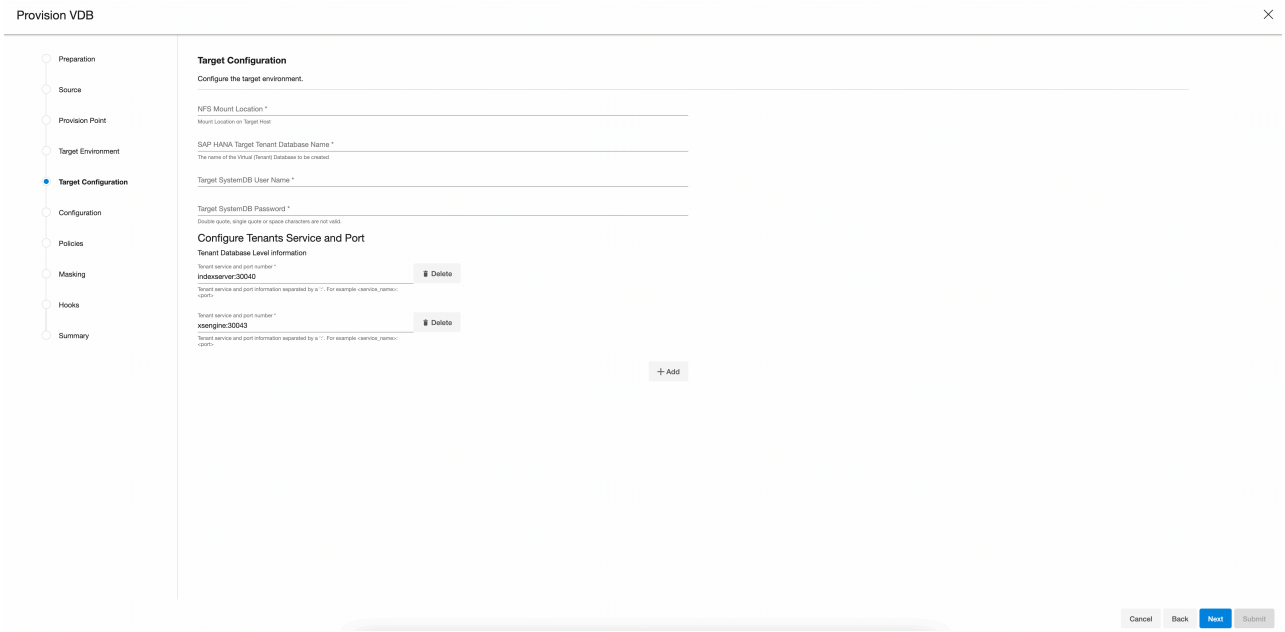
9.11.9.2 Provisioning SAP HANA VDBs

This topic describes how to provision a virtual database (VDB) from a HANA dSource.

9.11.9.2.1 Prerequisites

- The provision requires a linked dSource built off a physical source tenant, as described in Linking a HANA dSource (see page 2008), or a VDB.
- The HANA target environment should adhere to the Requirements for HANA Hosts and Databases (see page 1989)
- Operating system users and database users should be configured.

9.11.9.2.2 VDB creation UI



9.11.9.2.3 Procedure

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Datasets**.
4. Select a **dSource**.
5. Select a **snapshot** from which you want to provision.
   Info:

   If the first or empty dSource Snapshot that was created using the Staging Push mechanism is detected by the plugin and an error message is displayed. This check is done for every VDB creation operation.
6. Click the **Provision VDB** icon to open the Provision VDB wizard.
7. Select a target environment from the left pane.
8. The HANA 2.0 Plugin supports one installation on the same host, hence the value in the **Installation** dropdown will be already populated. Multiple drop-down options should not be available.
9. Set the **Environment User** to be the Instance Owner. Note: The picking of instance owner is only possible if you have multiple environment users set on that host.
10. Click **Next**.
11. In the **Target Configuration**pane you will need to specify the following:

Info: '*****' indicates the mandatory fields.

a. NFS Mount Location *: Delphix recommends setting the trailing directory to match the VDB name to delineate mount points and support administration and troubleshooting. In the example shown, the **NFS Mount Location** is /mnt/provision/dev where dev matches the **SAP HANA Target Tenant Database Name**.
b. SAP HANA Target Tenant Database Name *: This refers to the target tenant database name which will be created upon VDB creation.
c. **Target SystemDB User Name** *: This refers to the SystemDB username in the target database.
d. **Target SystemDB Password** *: This refers to the SystemDB password in the target database.
Note:
The SYSTEM username and password derived from the physical source database should be entered during the provision of a VDB because those credentials will be recovered during provisioning.
e. Configure Tenants Service and Port: This refers to the list of tenant services and port numbers in the following format.

```
<service>:<port>For example, indexserver:30043
```

You can click "**+Add**" to add more services.
Info :

- This is not a mandatory field. If a user doesn't provide a value to this, then the plugin will proceed with the above fields and it will allocate the service and port on its own.
- If an invalid service is provided, the plugin will proceed with input, but will eventually fail while creating the service and will display an error on the Delphix Engine UI.

12. In the configuration section
    a. Set the vFile Name or accept the default, which is a random string. Delphix strongly recommends applying a consistent naming convention. We suggest this name should match the Target Tenant Name from the previous pane. Maintaining a consistent naming convention will support administration and troubleshooting activities. In the example shown, the vFile Name “dev” matches the actual tenant name and mount point from the previous pane.
    b. Select a **Target Group** for the VDB.Click the **Add Dataset Group** icon to add a new group if required.
    c. Set **Auto VDB Restart** option to automatically restart the VDB, where the target host is rebooted. Delphix recommends enabling this setting.
13. Click **Next**.
14. VDB **Snapshot Policy**. Select the Snapshot policy if an automated VDB snapshot schedule is required. Optionally, the Snapshot policy may be disabled by selecting “None”.
15. Click **Next**.
16. **Masking**. This page offers the opportunity to link the provision job to a masking job. At this time, Delphix Masking does not support HANA. Please do not use this option. “Mask this VDB” should remain deselected.

17. **Hooks**. Run a customized shell script via a hook operation at a particular moment in the provision operation.
18. Click **Next**.
19. **Summary.** Presents an opportunity to review all the information as a part of the initial provision workflow.
20. Click **Submit**.Once a provision job is submitted it is possible to monitor its progress by selecting “Active Jobs” in the top menu bar. On successful completion, VDB will be visible in the list of Datasets under its assigned group.

9.11.9.3 Working with SAP HANA VDB snapshots

Taking a snapshot creates a new snapshot entry in the VDB's Timeflow.

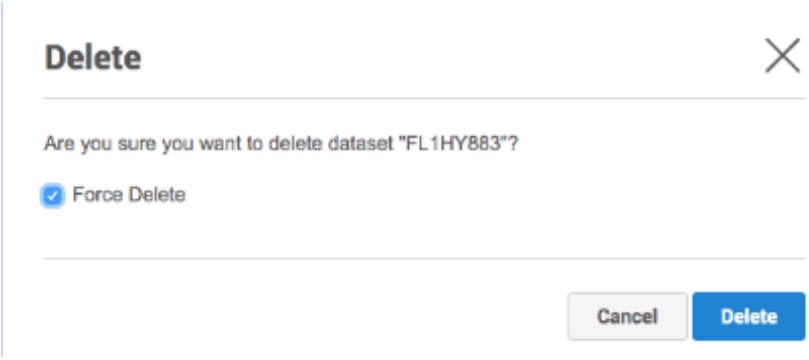
This section lists the steps to take a snapshot and delete the same.

1. Login to the **Delphix Management** application.
2. Click **Manage** and select **Datasets** from the dropdown list.
3. Select the VDB you want to perform Snapshot.
4. Click the **Camera** icon.
5. From the Snapshot dialog, select **Yes**.
6. Click **View:** under **Timeflow** to verify the Snapshot you just created.
7. To delete, select the snapshot you just created, and from the Actions menu (...) select **Delete Snapshot**.
8. In the Delete Snapshot dialog, select **Delete**.
9. Click **View:** under **Timeflow** to verify the Snapshot, you just deleted.

9.11.9.4 Failure during provision

Depending on where in the provision process failure occurs, and the type of failure, extra steps may be required:

- **Force Delete** of the provision object using the User Interface - The **Force Delete** option will remove underlying filesystems shared from the Delphix Engine, irrespective of any perceived dependencies on these filesystems. If provisioning fails very early in the process, it is common for the **Force Delete** option to be the only way to remove an object. This may happen when for example, the wrong password has been entered for the target-side system database user:



- In rare cases, removal of a failed virtual tenant may be required. For example a tenant with the name “TST”.

```
SELECT * FROM M_DATABASES WHERE DATABASE_NAME='TST';
ALTER SYSTEM STOP DATABASE TST IMMEDIATE;
DROP DATABASE TST DROP BACKUPS;
```

- If the **Force Delete** option could not unmount the virtual tenant mount points, then the unmount command should be run to remove the stale mount points.

```
umount -lf /mnt/provision/example
```

9.11.9.5 SAP HANA hook operations

9.11.9.5.1 Shell operations

9.11.9.5.1.1 RunCommand operation

The RunCommand operation runs a shell command on a Unix environment using whatever binary is available at /bin/sh. The environment user runs this shell command from their home directory. The Delphix Engine captures and logs all output from this command. If the script fails, the output is displayed in the Delphix Admin application and command-line interface (CLI) to aid in debugging.

If successful, the shell command must exit with an exit code of 0. All other exit codes will be treated as an operation failure.

Examples of RunCommand operations

You can input the full command contents into the RunCommand operation.

```
remove_dir="$DIRECTORY_TO_REMOVE_ENVIRONMENT_VARIABLE"
if test -d "$remove_dir"; then
    rm -rf "$remove_dir" || exit 1
```

```
fi
exit 0
```

If a script already exists on the remote environment and is executable by the environment user, the RunCommand operation can execute this script directly.

```
/opt/app/oracle/product/10.2.0.5/db_1/dbs/myscript.sh "$ARG_ENVIRONMENT_VARIABLE"
"second argument in double quotes" 'third argument in single quotes'
```

9.11.9.5.1.2 RunBash operation

The RunBash operation runs a Bash command on a Unix environment using a bash binary provided by the Delphix Engine, unless it's a Linux environment, in which case it uses the system's native bash binary. The environment user runs this Bash command from their home directory. The Delphix Engine captures and logs all output from this command. If the script fails, the output is displayed in the Delphix Admin application and command-line interface (CLI) to aid in debugging.

If successful, the Bash command must exit with an exit code of 0. All other exit codes will be treated as an operation failure.

Example of RunBash operations

You can input the full command contents into the RunBash operation.

```
remove_dir="$DIRECTORY_TO_REMOVE_ENVIRONMENT_VARIABLE"
# Bashisms are safe here!
if [[ -d "$remove_dir" ]]; then
    rm -rf "$remove_dir" || exit 1
fi
exit 0
```

9.11.9.5.2 Shell operation tips

9.11.9.5.2.1 Using nohup

You can use the nohup command and process backgrounding from the resource in order to "detach" a process from the Delphix Engine. However, if you use nohup and process backgrounding, you MUST redirect stdout and stderr.

Unless you explicitly tell the shell to redirect stdout and stderr in your command or script, the Delphix Engine will keep its connection to the remote environment open while the process is writing to either stdout or stderr. Redirection ensures that the Delphix Engine will see no more output and thus not block waiting for the process to finish.

For example, imagine having your RunCommand operation background a long-running Python process. Below are the bad and good ways to do this.

Bad examples
• nohup python file.py & # no redirection • nohup python file.py 2>&1 & # stdout is not redirected • nohup python file.py 1>/dev/null & # stderr is not redirected • nohup python file.py 2>/dev/null & # stdout is not redirected

Good examples
• nohup python file.py 1>/dev/null 2>&1 & # both stdout and stderr redirected, Delphix Engine will not block

9.11.9.5.3 Other operations

9.11.9.5.3.1 RunExpect operation

The RunExpect operation executes an Expect script on a Unix environment. The Expect utility provides a scripting language that makes it easy to automate interactions with programs that normally can only be used interactively, such as ssh. The Delphix Engine includes a platform-independent implementation of a subset of the full Expect functionality.

The script is run on the remote environment as the environment user from their home directory. The Delphix Engine captures and logs all output of the script. If the operation fails, the output is displayed in the Delphix Admin application and CLI to aid in debugging.

If successful, the script must exit with an exit code of 0. All other exit codes will be treated as an operation failure.

Example of a RunExpect operation

Start an ssh session while interactively providing the user's password.

```
spawn ssh user@delphix.com
expect {
    -re {Password: } {

        send "${env(PASSWORD_ENVIRONMENT_VARIABLE)}\n"
    }
    timeout {

        puts "Timed out waiting for password prompt."

        exit 1
```

```
    }
}
exit 0
```

9.11.9.5.4 HANA environment variables

Operations that run user-provided scripts have access to environment variables. For operations associated with specific dSources or virtual databases (VDBs), the Delphix Engine will always set certain environment variables so that the user-provided script can use them to access the dSource or VDB.

9.11.9.5.4.1 dSource environment variables

Environment Variable	Description
`DLPX_DATA_DIRECTORY`	The primary mount path provided by the user on the UI. The first one, in case multiple mount specifications are provided.
`USER`	The OS user used to link the dSource

9.11.9.5.4.2 VDB environment variables

Environment Variable	Description
`DLPX_DATA_DIRECTORY`	The primary mount path provided by the user on the UI. The first one, in case multiple mount specifications are provided.
`USER`	The OS user used to provision the VDB

9.11.9.5.5 Shell operations

9.11.9.5.5.1 RunCommand operation

The RunCommand operation runs a shell command on a Unix environment using whatever binary is available at /bin/sh. The environment user runs this shell command from their home directory. The Delphix Engine captures and logs all output from this command. If the script fails, the output is displayed in the Delphix Admin application and command-line interface (CLI) to aid in debugging.

If successful, the shell command must exit with an exit code of 0. All other exit codes will be treated as an operation failure.

Examples of RunCommand operations

You can input the full command contents into the RunCommand operation.

```
remove_dir="$DIRECTORY_TO_REMOVE_ENVIRONMENT_VARIABLE"
if test -d "$remove_dir"; then
    rm -rf "$remove_dir" || exit 1
fi
exit 0
```

If a script already exists on the remote environment and is executable by the environment user, the RunCommand operation can execute this script directly.

/opt/app/oracle/product/10.2.0.5/db_1/dbs/myscript.sh "$ARG_ENVIRONMENT_VARIABLE" "second argument in double quotes" 'third argument in single quotes'

9.11.9.5.5.2 RunBash operation

The RunBash operation runs a Bash command on a Unix environment using a bash binary provided by the Delphix Engine. The environment user runs this Bash command from their home directory. The Delphix Engine captures and logs all output from this command. If the script fails, the output is displayed in the Delphix Admin application and command-line interface (CLI) to aid in debugging.

If successful, the Bash command must exit with an exit code of 0. All other exit codes will be treated as an operation failure.

Example of RunBash operations

You can input the full command contents into the RunBash operation.

```
remove_dir="$DIRECTORY_TO_REMOVE_ENVIRONMENT_VARIABLE"
# Bashisms are safe here!
if [[ -d "$remove_dir" ]]; then
    rm -rf "$remove_dir" || exit 1
fi
exit 0
```

9.11.9.5.6 Shell operation tips

9.11.9.5.6.1 Using nohup

You can use the nohup command and process backgrounding from the resource in order to "detach" a process from the Delphix Engine. However, if you use nohup and process backgrounding, you MUST redirect stdout and stderr.

Unless you explicitly tell the shell to redirect stdout and stderr in your command or script, the Delphix Engine will keep its connection to the remote environment open while the process is writing to either stdout or stderr. Redirection ensures that the Delphix Engine will see no more output and thus not block waiting for the process to finish.

For example, imagine having your RunCommand operation background a long-running Python process. Below are the bad and good ways to do this.

Bad examples
• nohup python file.py & # no redirection • nohup python file.py 2>&1 & # stdout is not redirected • nohup python file.py 1>/dev/null & # stderr is not redirected • nohup python file.py 2>/dev/null & # stdout is not redirected

Good examples
• nohup python file.py 1>/dev/null 2>&1 & # both stdout and stderr redirected, Delphix Engine will not block

9.11.9.5.7 Other operations

9.11.9.5.7.1 RunExpect operation

The RunExpect operation executes an Expect script on a Unix environment. The Expect utility provides a scripting language that makes it easy to automate interactions with programs that normally can only be used interactively, such as ssh. The Delphix Engine includes a platform-independent implementation of a subset of the full Expect functionality.

The script is run on the remote environment as the environment user from their home directory. The Delphix Engine captures and logs all output of the script. If the operation fails, the output is displayed in the Delphix Admin application and CLI to aid in debugging.

If successful, the script must exit with an exit code of 0. All other exit codes will be treated as an operation failure.

Example of a RunExpect operation

Start an ssh session while interactively providing the user's password.

```
spawn ssh user@delphix.com
expect {
    -re {Password: } {
```

```
        send "${env(PASSWORD_ENVIRONMENT_VARIABLE)}\n"
    }
    timeout {

        puts "Timed out waiting for password prompt."

        exit 1
    }
}
exit 0
```

9.11.9.5.8 HANA environment variables

Operations that run user-provided scripts have access to environment variables. For operations associated with specific dSources or virtual databases (VDBs), the Delphix Engine will always set certain environment variables so that the user-provided script can use them to access the dSource or VDB.

9.11.9.5.8.1 dSource environment variables

Environment Variable	Description
`DLPX_DATA_DIRECTORY`	The primary mount path provided by the user on the UI. The first one, in case multiple mount specifications are provided.
`USER`	The OS user used to link the dSource

9.11.9.5.8.2 VDB environment variables

Environment Variable	Description
`DLPX_DATA_DIRECTORY`	The primary mount path provided by the user on the UI. The first one, in case multiple mount specifications are provided.
`USER`	The OS user used to provision the VDB

9.12 SQL Server data sources

This section covers the following topics:

- Delphix architecture with SQL Server (see page 2021)
- Quick start guide for SQL Server (Microsoft SQL Server on Windows) (see page 2028)
- SQL Server support and requirements (see page 2052)
- Managing SQL Server environments and hosts (see page 2092)
- Linking data sources and syncing data with SQL Server (see page 2139)
- Provisioning and managing virtual databases with SQL Server (see page 2180)
- Hooks for SQL Server (see page 2202)
- Using hostChecker to validate target database servers (see page 2217)

9.12.1 Delphix architecture with SQL Server

SQL Server is Microsoft's relational database which is typically run on Windows Server hosts. With Delphix, you can use various SQL Server configurations, ranging from Failover Clusters to Availability Groups. In this section, you'll find an overview of how Delphix works with SQL Server.

There are three key concepts when using Delphix with any data platform:

1. Environments: The server and software required to run a data set.
    a. Source Environment: Source data to be ingested into Delphix. These will be used to create dSources.
    b. Target Environment: Target hosts to provision VDBs.
2. dSources: A database that the Delphix Virtualization Engine uses to create and update or maintain virtual copies of your database
3. VDBs: A database provisioned from either a dSource or another VDB which is a copy of the source data. A VDB is created and managed by the Delphix Virtualization Engine.

With these concepts in mind, let's explore how Delphix connects to SQL Server environments and creates SQL Server dSources and VDBs.

9.12.1.1 Staging push mechanism with SQL server

Starting Delphix Engine version 6.0.13.0 a new data ingestion mechanism has been introduced that will help users to push data into the staging database on their own.

The previous data ingestion mechanism of the SQL Server has a few limitations like dependency on the source access and limited backup vendor's support. Currently, Delphix supports Native, Commvault, NetBackup, Lightspeed, and Redgate backup vendors.

With Staging Push implementation, we have removed the dependency of accessing the customer's production database and also enabled the customers who are using the backup vendors that Delphix does not support.

To summarize the Staging Push mechanism.

- Customers now have ownership of the staging databases.
- Customers are now responsible to keep the staging database in sync with the source database. Note that the database files of the staging database are stored on Delphix Storage.
- Delphix is still responsible to snapshot the underlying data files and gathering any metadata required for provisioning from the snapshots.

9.12.1.2 Delphix in multi-domain windows environments

9.12.1.2.1 General Overview

When considering the Delphix logical architecture, there are four primary components:

1. Source host(s)
2. Continuous Data Engine
3. VDB Target host(s)
4. Staging Target Host(s)

In SQL Server environments, the staging target host is used for staging data from the source database on the source host into Delphix. Although you can use any VDB target host on which the Delphix Connector service has been installed for this purpose, Delphix recommends a dedicated Staging Target Host for load isolation and separation of roles.

This page focuses on the process of getting source SQL Server data into the storage of the Continuous Data Engine (DVE) via the Staging Target Host.

When considering SQL Server deployments in different enterprise environments, we often see cases where the production, development, test, or reporting environments exist in different Windows domains which may or may not have trust relationships. Such varying domain approaches can come into play due to security, organizational, geographical, or other technical reasons, and can make communication between Windows hosts more complicated to manage. Delphix is flexible enough to work in many configurations, but we want to help you choose the solution that best suits your unique environment.

We listed the four primary components of the Delphix logical architecture for SQL Server above. In addition, a fifth component in the Delphix logical architecture might be considered for use-cases #3, #4, and #5 in the Technical Overview below: a Connector host. The function of the Delphix Connector on that host is the discovery of the source environment via remote registry and ODBC calls. There are no Delphix software installation requirements for Windows source hosts, but it might be helpful to note this role can co-reside directly on the Windows source host for consolidation purposes if desired.

9.12.1.2.2 Technical overview

Keep in mind that the Delphix Engine is always syncing with backups of the source database. It is never the live data that is ingested; it is always backups of different flavors.

If SQL Server simple recovery mode is used, these can be full or differential backups initiated by the source database. If full recovery mode is enabled, the Delphix Engine will typically leverage only transaction log backups after the initial data load. Again, the source database would initiate backups, and the Delphix Engine would collect the backup files that have been created by SQL Server. This approach of using transaction logs minimizes spikes in system load by ingesting smaller backups more often. Another option is copy-only backups, which the Delphix Engine initiates in a configuration known as Delphix Managed Backups. For more information refer to Delphix as a Backup Solution to SQL Server (see page 2118).[602]

Delphix can ingest database and log data from native backups, as well as a number of third-party backup products. SQL Server restores the backups onto the shared Delphix storage on the staging target host

602 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/delphix-as-a-backup-solution-to-sql-server

running the databases in recovery mode. We call this process “validated sync,” which is why you may hear the staging target also referred to as a validated sync server.

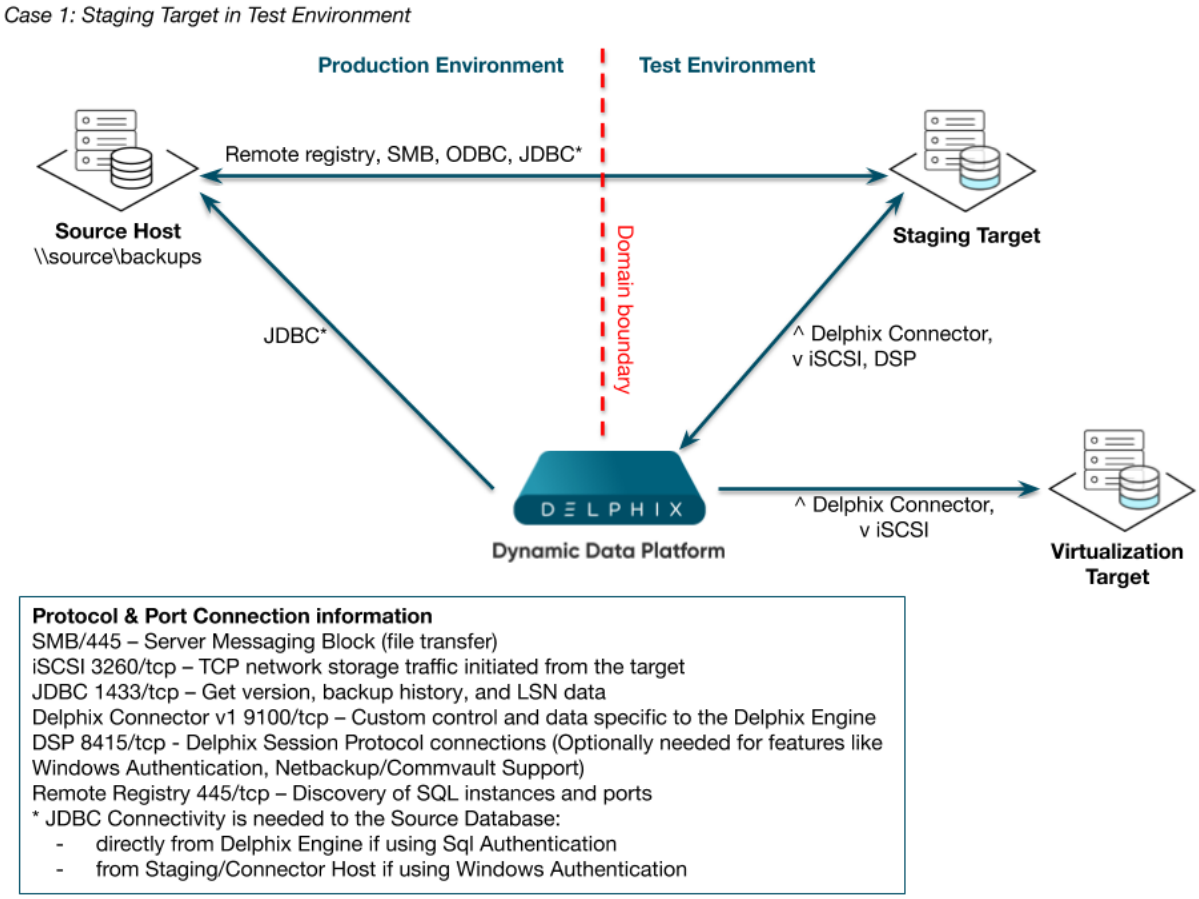
It is important to note that the Continuous Data Engine (based on DxOS, itself derived from a UNIX-based OS) is not a domain member itself. The credentials we discuss in this document are between Windows servers, and the key domain-specific authentication is between the staging host and the UNC path to the SMB share where the backup data is stored.

In the rest of this section, we will describe multiple scenarios. Review them to determine which will fit best in your environment.

9.12.1.2.3 Case 1: Staging target in test environment

In this case, we will review an environment with two domains: PRODUCTION and TEST, which have a domain trust relationship. This is one of the simplest and most straightforward approaches, as illustrated in the *Staging Target in Test Environment diagram below*.

In this example, the staging target host exists in the non-production TEST domain, but because of domain trust, accounts located in that domain can access resources in the PRODUCTION domain. This would allow the staging target host to connect to the PRODUCTION source host both for environment discovery and to the shared backup location “\source\backups” over Server Messaging Block (SMB) to access database and transaction log backups.

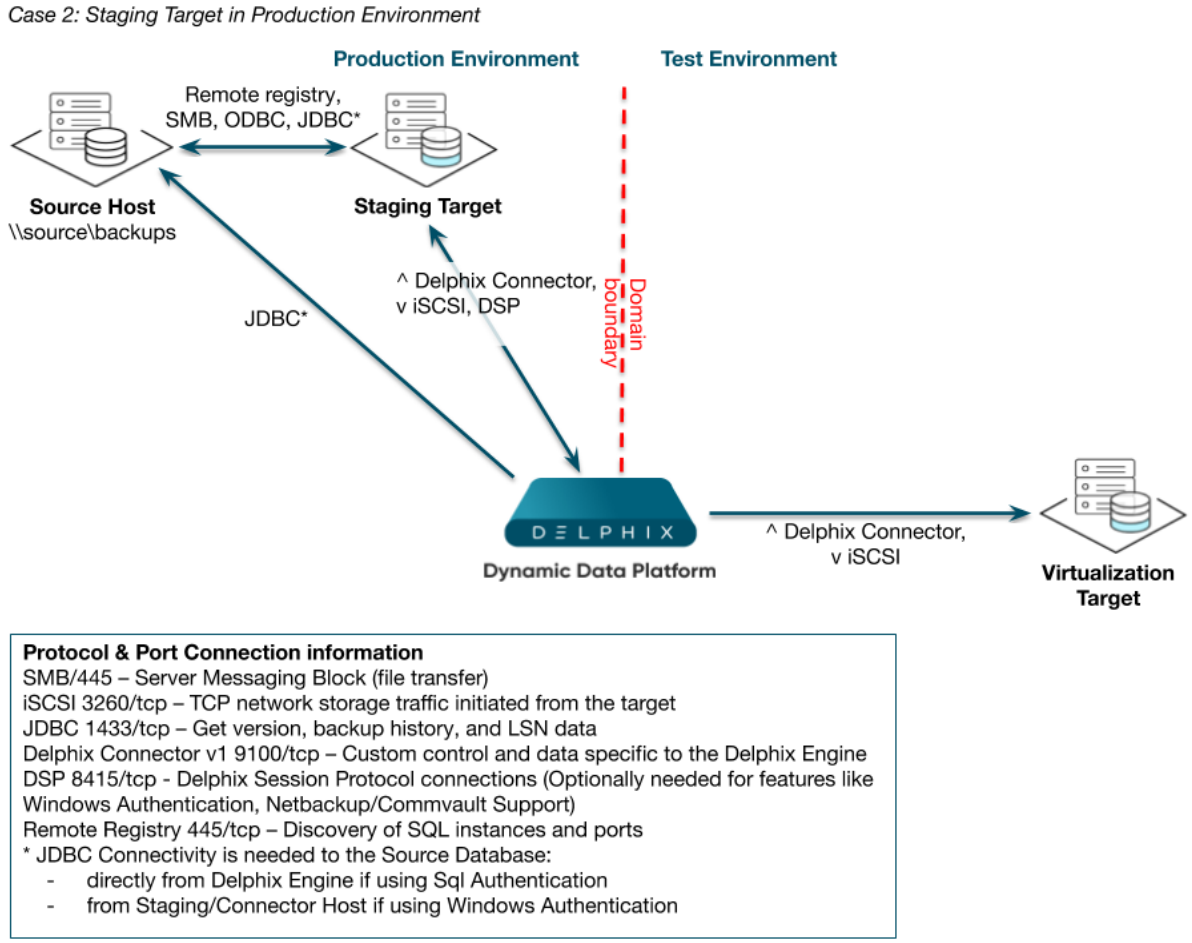


Staging Target in Test Environment

9.12.1.2.4 Case 2: Staging target in production environment

The scenario illustrated in the *Staging Target in Production Environment* diagram below shows a TEST domain that does not have access to resources in the PRODUCTION domain. However, the customer has determined that Delphix VDBs must be provisioned to the TEST domain. In this scenario, you can use the staging target host in the PRODUCTION domain to link to the PRODUCTION database and perform the normal restore of the DB and/or log files to the Delphix storage. You can then provision VDBs in the TEST domain.

In this case, VDBs can be completely isolated from the PRODUCTION domain, and there is no requirement for hosts in the TEST domain to have any direct access to resources in the PRODUCTION domain.



Staging Target in Production Environment

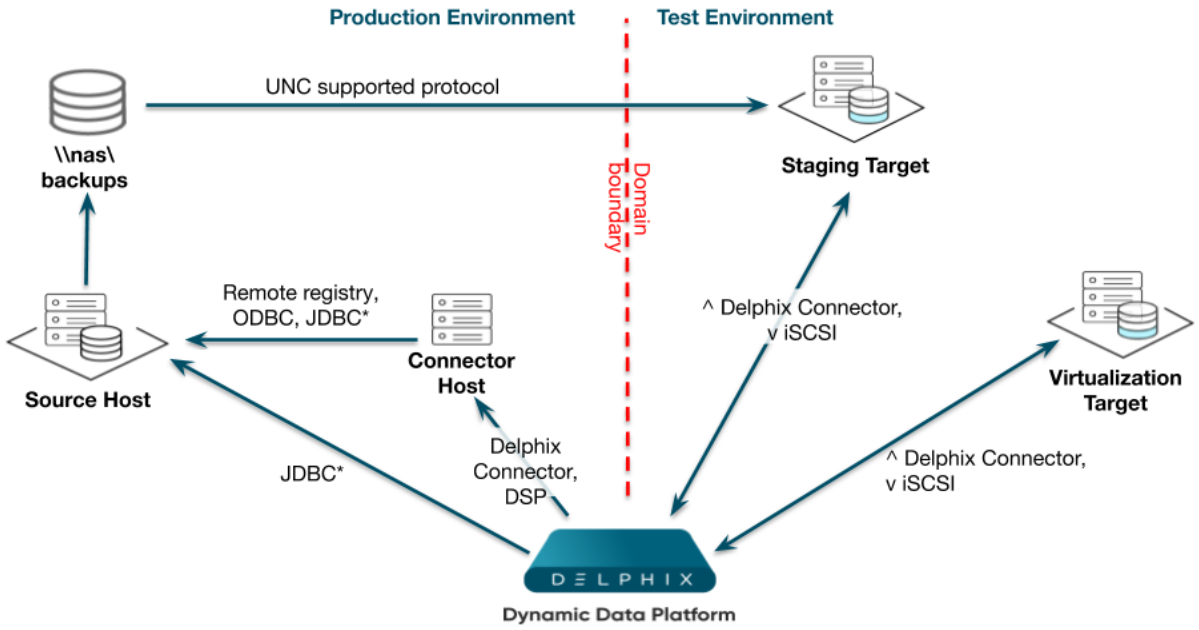
9.12.1.2.5 Case 3: Domain-agnostic storage

This example shows a shared backup location that is not dependent on trust relationships between the PRODUCTION and TEST domains. Because Delphix uses UNC paths, it can support any protocol which provides UNC access for that backup data access – for example, SMB or iSCSI.

This is shown by the *Domain-agnostic Storage* diagram below by the arrow – stretching from bottom-left toward the upper-right and crossing the domain boundary – representing any UNC-compatible protocol

connecting the staging target host to the data on the NAS host. Provided that the Delphix environment users on both the source host and staging target host have read/write access to the shared backup location on network-attached storage (NAS), the SQL Server instance running on the staging target host will be able to access the backup files needed.

Although this option is not specific to this case, you may notice we separated a connector role to its own connector host. As you can infer from the diagram, the Delphix Connector's primary function on that host is the discovery of the source environment via remote registry and ODBC calls. Despite the fact that there are no software installation requirements for the source hosts in PRODUCTION, it may be helpful to note that you can even install this role directly on the source server for consolidation if you want to.



Protocol & Port Connection information
SMB/445 – Server Messaging Block (file transfer)
iSCSI 3260/tcp – TCP network storage traffic initiated from the target
JDBC 1433/tcp – Get version, backup history, and LSN data
Delphix Connector v1 9100/tcp – Custom control and data specific to the Delphix Engine
DSP 8415/tcp - Delphix Session Protocol connections (Optionally needed for features like Windows Authentication, Netbackup/Commvault Support)
Remote Registry 445/tcp – Discovery of SQL instances and ports
* JDBC Connectivity is needed to the Source Database:
- directly from Delphix Engine if using Sql Authentication
- from Staging/Connector Host if using Windows Authentication

Case 3: Domain-agnostic Storage

Domain-agnostic Storage

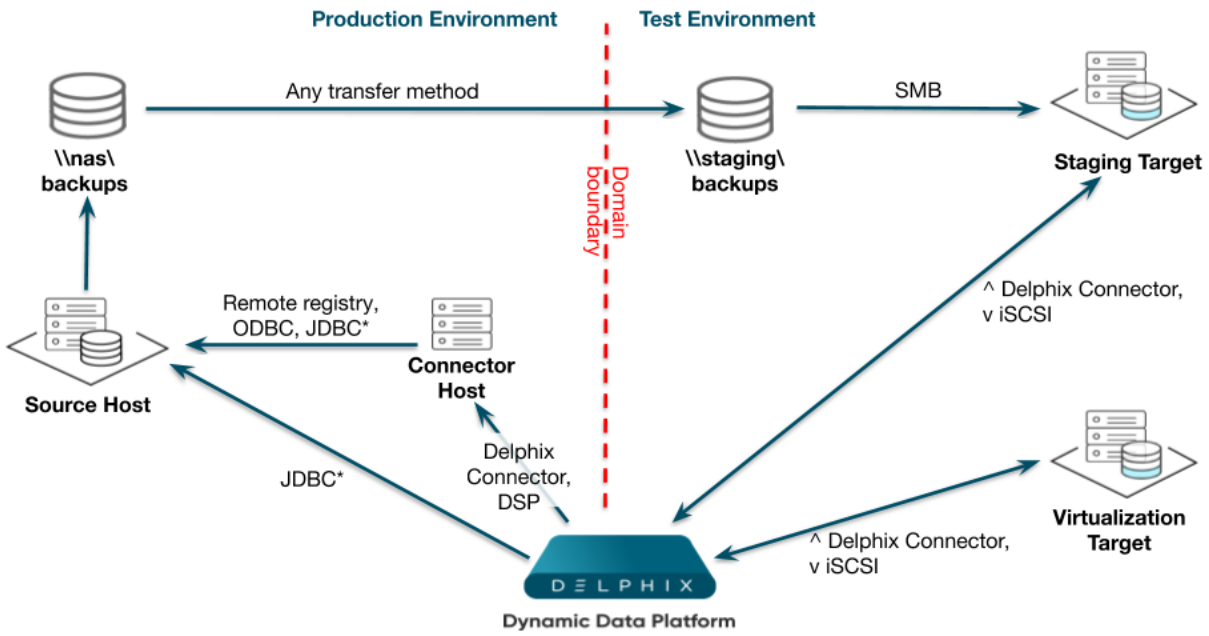
9.12.1.2.6 Case 4: Migrating backup files

In this somewhat more complex configuration, backup files are sent to storage in the PRODUCTION domain, while the host used to link to the source and perform the validated sync is in an isolated TEST domain. We have used a separate connector host in the PRODUCTION domain again, to perform the environment discovery of the source host there. Backup files for SOURCE are being stored on NAS.

We will link using the Staging Target Host and create VDBs in the TEST domain. When the Delphix Engine discovers that a new backup of PRODUCTION has been taken, it will attempt to find the relevant files in the

shared backup location provided during linking. It does this by periodically performing a recursive search for the file names on the shared backup location. If it does not find the specific files, it will try again later. Knowing this, we can specify a shared backup location in the TEST domain and set up an automated process to copy the backup files from \nas\backups in the PRODUCTION domain to \staging\backups in the TEST domain. We can use any copy mechanism to transfer the files, such as FTP or ROBOCOPY. The files must be available long enough for the Delphix Engine to detect and apply them to the recovery database on the Staging Target Host before removal.

We have customers who also use this model in cases with multiple data centers (on-premise deployments) or virtual private clouds (cloud deployments) rather than multiple domains. These customers want database and transaction log backups to be available in secondary data centers or private clouds, but they want to make sure that the data is only copied over the WAN once.



Protocol & Port Connection information
SMB/445 – Server Messaging Block (file transfer)
iSCSI 3260/tcp – TCP network storage traffic initiated from the target
JDBC 1433/tcp – Get version, backup history, and LSN data
Delphix Connector v1 9100/tcp – Custom control and data specific to the Delphix Engine
DSP 8415/tcp - Delphix Session Protocol connections (Optionally needed for features like Windows Authentication, Netbackup/Commvault Support)
Remote Registry 445/tcp – Discovery of SQL instances and ports
* JDBC Connectivity is needed to the Source Database:
- directly from Delphix Engine if using Sql Authentication
- from Staging/Connector Host if using Windows Authentication

Case 4: Migrating Backup Files

Migrating Backup Files

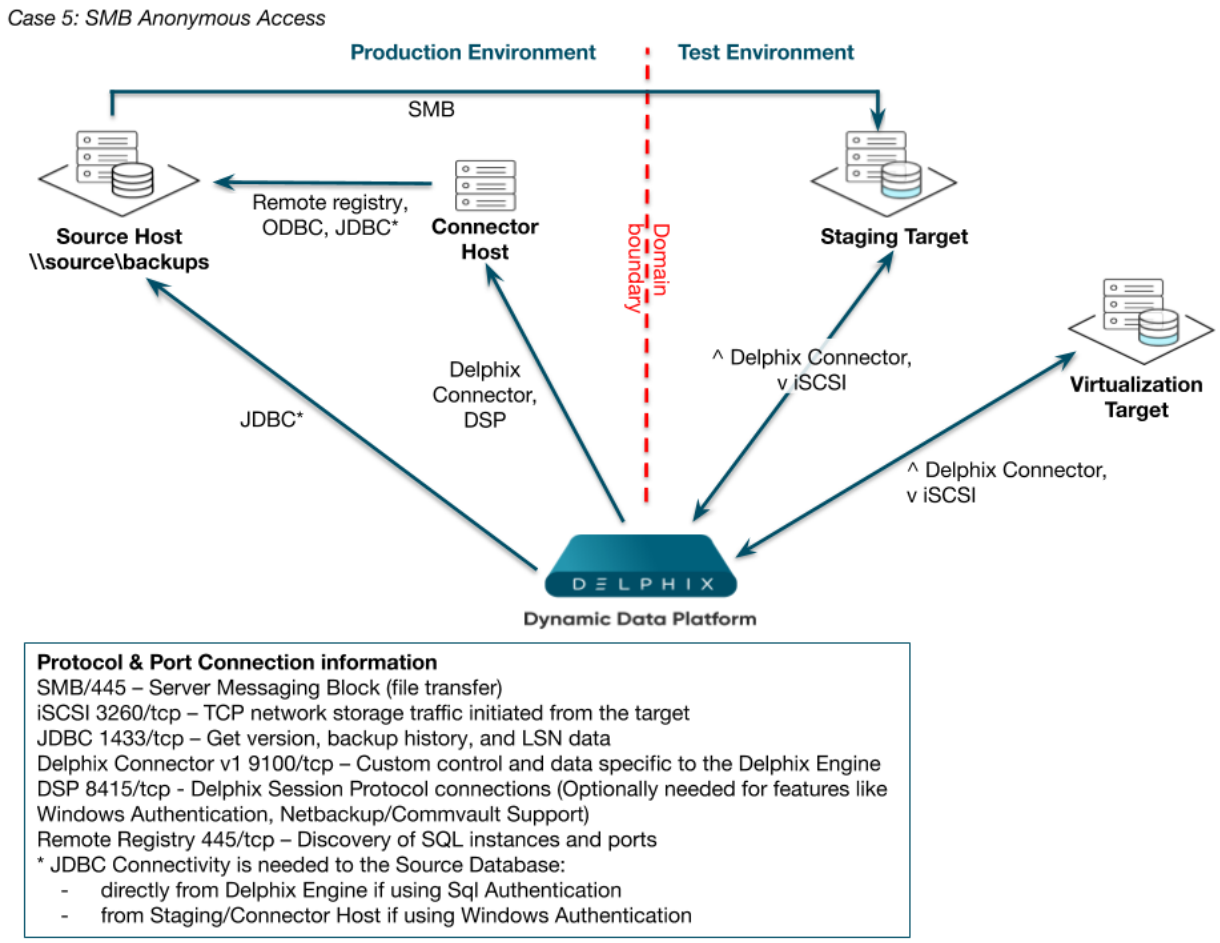
9.12.1.2.7 Case 5: SMB anonymous access

In this example (shown by the *SMB Anonymous Access* diagram below), a Windows SMB connection is traversing domains that do not have a trust relationship. This approach is problematic because there is no simple configuration for SMB file sharing that does not rely on domain trusts. As a result, there is no way to specifically grant accounts in the TEST domain access to SMB shares in the PRODUCTION domain.

Because such users cannot be authenticated, they are treated as “anonymous” users and do not have permission to any resources by default.

Windows provides an “Everyone” group. However, this group still only applies to accounts that can be authenticated in the domain, so you cannot use that group in this case. There is still a way to configure access to the shared backup location on \source\backups by accounts in other domains, such as TEST. However, because it relies on anonymous access, you will need to consider the security implications of enabling this method, as well as measures that could mitigate any additional risk in your environment – for example, a private VLAN or IPSEC between hosts.

1. Enable the “Guest” account on the server source Server – for example, **\SQLPROD**.
2. Create a share where full and transaction log backups will be stored – for example, **\SQLPROD\backups**.
3. Configure read-only security access for both the folder security permissions on the shared directory and the share permissions for the “guest” account.



SMB Anonymous Access

Here are some additional links from Microsoft that relate to anonymous sharing:

- Network access: Let Everyone permissions apply to anonymous users[603]
- Network access: Shares that can be accessed[604]

9.12.2 Quick start guide for SQL Server (Microsoft SQL Server on Windows)

This quick start guide, which is excerpted from the larger User Guide, is intended to provide you with a quick overview of working with SQL Server database objects in the Delphix Engine. It does not cover any advanced configuration options or best practices, which can have a significant impact on performance. It assumes that you are working in a Lab/Dev setting and attempting to quickly test Delphix functionality. It assumes you will use the VMware Hypervisor. It assumes you are running supported combinations of software as explained here: Supported OS, SQL Server, and Backup Software Versions for SQL Server (see page 2052).

9.12.2.1 Overview

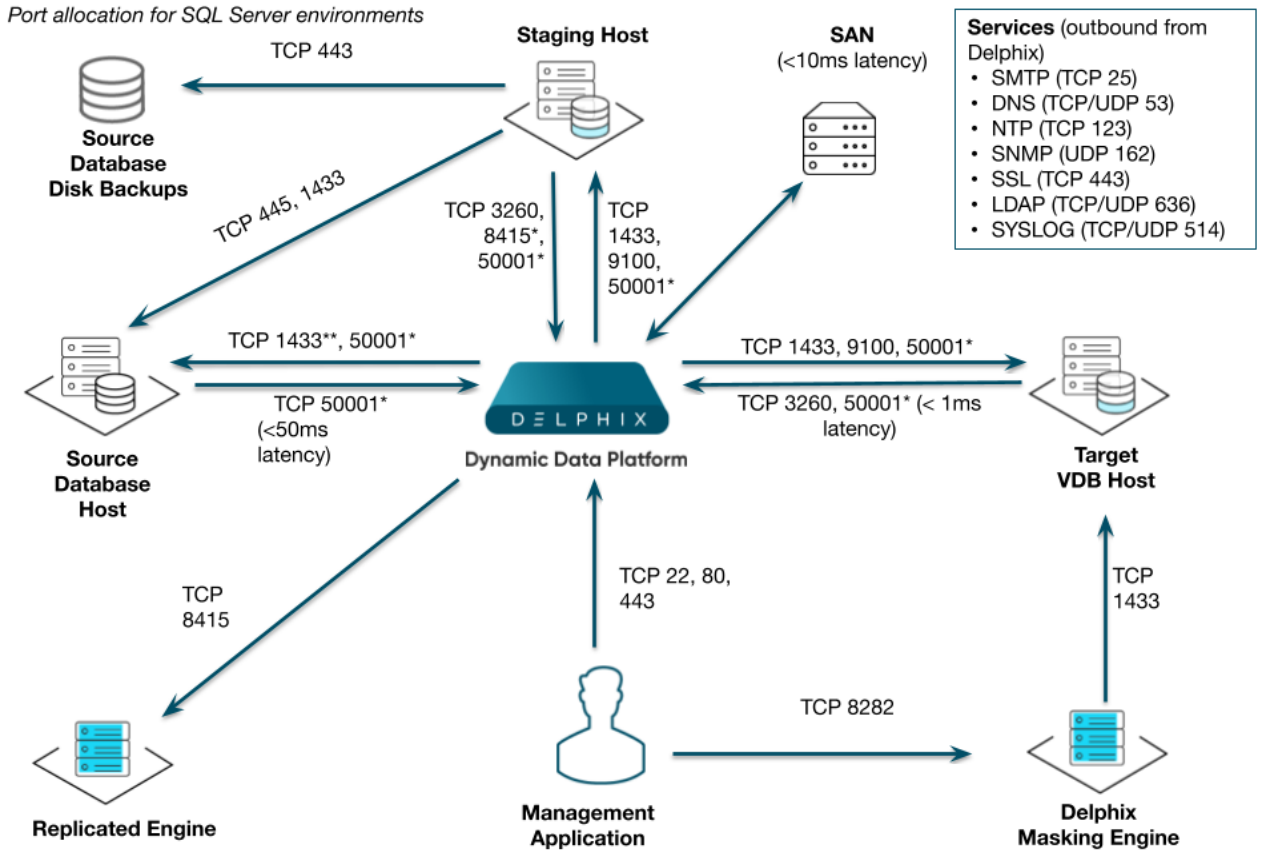
In this guide, we will walk through deploying a Delphix Engine, starting with configuring Source, Staging (aka Validated Sync), and Target database environments on Windows servers. We will then create a dSource, and provision a VDB.

The following diagram describes the engine topology for SQL Server environments. It illustrates the recommended ports to be open from the engine to remote services, to the Delphix Engine, and to the Source, Target and Validated Sync Environments.

> Port 1433 is required between the Delphix Engine and AG (Availability Group) cluster sources.

For purposes of the QuickStart, you can ignore any references to Replication or Masking, such as the engines shown in the diagram below.

603 https://technet.microsoft.com/en-us/library/jj852264(v=ws.11).aspx
604 https://technet.microsoft.com/en-us/library/jj852200(v=ws.10).aspx



9.12.2.2 Deploy OVA on VMware

Use the Delphix-supplied OVA file to install the Delphix Engine. The OVA file is configured with many of the minimum system requirements. The underlying storage for the install is assumed to be redundant SAN storage.

1. Download the OVA file from https://download.delphix.com[605]. You will need a support login from your sales team or a welcome letter.
    a. Navigate to the Delphix Product Releases
2. Login using the vSphere client to the vSphere server (or vCenter Server) where you want to install the Delphix Engine.
3. In the vSphere Client, click **File**.
4. Select **Deploy OVA Template**.
5. Browse to the OVA file.
6. Click **Next**.

605 https://download.delphix.com/

7. Select a **hostname** for the Delphix Engine. This hostname will also be used in configuring the Delphix Engine network.
8. Select the **data center** where the Delphix Engine will be located.
9. Select the **cluster** and the **ESX host**.
10. Select one (1) **data store** for the **Delphix OS**. This datastore can be **thin-provisioned** and must have enough free space to accommodate the 127GB comprising the Delphix operating system.
11. Select four (4) or more **data stores** for Database Storage for the Delphix Engine. The Delphix Engine will stripe all of the Database Storage across these VMDKs, so for optimal I/O performance, each VMDK must be equal in size and be configured **Thick Provisioned - Eager Zeroed**. Additionally, these VMDKs should be distributed as evenly as possible across all four SCSI I/O controllers.
12. Select the **virtual network** you want to use.
    If using multiple physical NICs for link aggregation, you must use vSphere NIC teaming. Do not add multiple virtual NICs to the Delphix Engine itself. The Delphix Engine should use a single virtual network. For more information, see Optimal Network Architecture for the Delphix Engine (see page 719).
13. Click **Finish**. The installation will begin and the Delphix Engine will be created in the location you specified.
14. Once the installation has completed, power on the Delphix Engine and proceed with the initial system configuration as described in Setting Up Network Access to the Delphix Engine (see page 569).

> If your source database is 4 TB, you probably need 4 TB of storage for the Delphix Engine. Add at least 4 data disks of similar size for the Delphix VM. For example: for a source database of 4 TB, create 4 VMDKs of 1 TB each.

> For a full list of requirements and best practice recommendations, see Virtual Machine Requirements for VMware Platform (see page 604).

9.12.2.2.1 Setup network access to Delphix engine

1. Power on the Delphix Engine and open the Console.
2. Wait for the Delphix Management Service and Delphix Boot Service to come online. This might take up to 10 minutes during the first boot. Wait for the large orange box to turn green.
3. Press any key to access the sysadmin console.
4. Enter `sysadmin@SYSTEM` for the username and `sysadmin` for the password.
5. You will be presented with a description of available network settings and instructions for editing.

```
Delphix Engine Network Setup

To access the system setup through the browser, the system must first be
configured for networking in your
environment. From here, you can configure the primary interface, DNS, hostname,
and default route. When DHCP is
configured, all other properties are derived from DHCP settings.

To see the current settings, run "get." To change a property, run "set
<property>=<value>." To commit your changes,
run "commit." To exit this setup and return to the standard CLI, run "discard."

    defaultRoute    IP address of the gateway for the default route -- for
 example, "1.2.3.4."

    dhcp            Boolean value indicating whether DHCP should be used for
 the primary interface. Setting this value
                    to "true" will cause all other properties (address,
hostname, and DNS) to be derived from the DHCP
                    response

    dnsDomain       DNS Domain -- for example, "delphix.com"

    dnsServers      DNS server(s) as a list of IP addresses -- for example,
"1.2.3.4,5.6.7.8."

    hostname        Canonical system hostname, used in alert and other logs --
for example, "myserver"

    primaryAddress  Static address for the primary interface in CIDR notation
-- for example, "1.2.3.4/22"

Current settings:

    defaultRoute: 192.168.1.1
    dhcp: false
    dnsDomain: example.com
    dnsServers: 192.168.1.1
    hostname: Delphix
    primaryAddress: 192.168.1.100/24
```

6. Configure the `hostname`. If you are using DHCP, you can skip this step. **Note** : Use the same `hostname` you entered during the server installation.

```
delphix network setup update *> set hostname=<hostname>
```

7. Configure DNS. If you are using DHCP, you can skip this step.

```
delphix network setup update *> set dnsDomain=<domain>
delphix network setup update *> set dnsServers=<server1-ip>[,<server2-ip>,...]
```

8. Configure either a static or DHCP address. **Note** The static IP address must be specified in CIDR notation (for example, `192.168.1.2/24`)
   - DHCP Configuration

     ```
     delphix network setup update *> set dhcp=true
     ```

   - Static Configuration

     ```
     delphix network setup update *> set dhcp=false
     delphix network setup update *> set primaryAddress=<address>/<prefix-
     len>
     ```

9. Configure a default gateway. If you are using DHCP, you can skip this step.

   ```
   delphix network setup update *> set defaultRoute=<gateway-ip>
   ```

10. Commit your changes. Note that you can use the `get` command prior to committing to verify your desired configuration.

    ```
    delphix network setup update *> commit
    Successfully committed network settings. Further setup can be done through the
    browser at:
            http://<address>
    Type "exit" to disconnect, or any other commands to continue using the CLI.
    ```

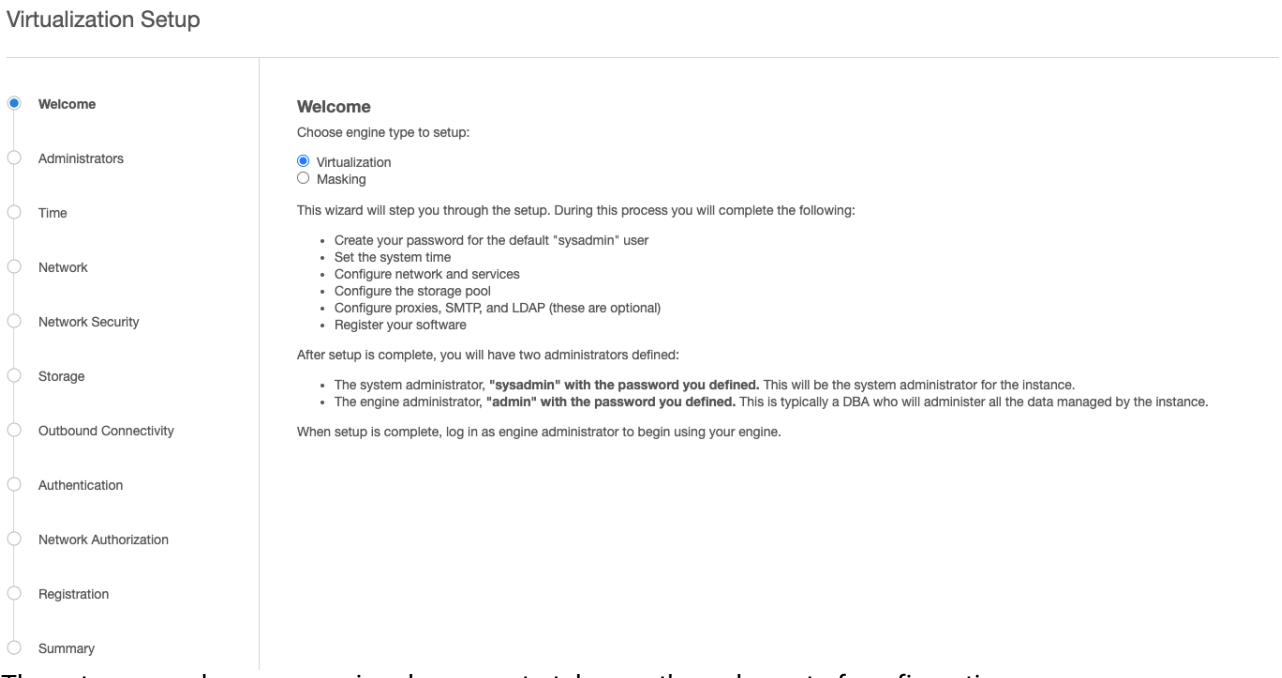
11. Check that you can now access the Delphix Engine through a Web browser by navigating to the displayed IP address, or hostname if using DNS.
12. Exit setup.

    ```
    delphix> exit
    ```

9.12.2.3 Setting up the Delphix engine

Once you set up the network access for Delphix Engine, navigate to the Delphix Engine URL in your browser for server setup.

The welcome screen below will appear for you to begin your Delphix Engine setup.



The setup procedure uses a wizard process to take you through a set of configuration screens:

- Administrators
- Time
- Network
- Network Security
- Storage
- Outbound Connectivity
- Authentication
- Network Authorization
- Registration
- Summary

1. Connect to the Delphix Engine at `http:///login/index.html#serverSetup`.
   The **Delphix Setup** application will launch when you connect to the server.
   Enter your **sysadmin** login credentials, which initially defaults to the username **sysadmin**, with the initial default password of **sysadmin**. On first login, you will be prompted to change the initial default password.
2. Click **Next**.

9.12.2.3.1 Administrators

The Delphix Engine supports two types of administrators:

- System Administrator (**sysadmin**) - this is the engine system administrator. The sysadmin password is defined here**.**
- Engine Administrator (**admin)** - this is typically a DBA who will administer all the data managed by the engine.

On the Administrators tab, you set up the sysadmin password by entering an email address and password. The details for the admin are displayed for reference.

> The default domain user created on Delphix Engines from 5.3.1 is known as **admin** instead of delphix_admin. When engines created before 5.3.1 are upgraded to 5.3.1 or later they will retain their old username 'delphix_admin'. To avoid complications Delphix recommends creating users with an admin role and then Disabling delphix_admin.

9.12.2.4 System time

The engine time is used as the baseline for setting policies that coordinate between virtual databases and external applications.

Choose your option to set up system time in this section. For a Quick Start, simply set the time and your timezone. You can change this later.

9.12.2.4.1 Network

The initial out-of-the-box network configuration in the OVA file is set to use a single VMXNET3 network adapter.

You have already configured this in the initial configuration. Delphix supports more advanced configurations, but you can enable those later.

9.12.2.4.2 Storage

You should see the data storage VMDKs or RDMs you created during the OVA installation. Click **Next** to configure these for data storage.

9.12.2.4.3 Serviceability

Choose your options to configure serviceability settings.

For a Quick Start, accept the defaults. You can change this later.

9.12.2.4.4 Authentication

Choose your options to configure authentication services.

For a Quick Start, accept the defaults. You can change this later.

9.12.2.4.5 Registration

If the Delphix Engine has access to the external Internet (either directly or through a web proxy), then you can auto-register the Delphix Engine:

1. Enter your **Support Username** and **Support Password**.
2. Click **Register**.

If external connectivity is not immediately available, you must perform manual registration.

1. Copy the **Delphix Engine registration code** in one of two ways:
   a. Manually highlight the registration code and copy it to clipboard. Or,
   b. Click **Copy Registration Code to Clipboard**.
2. Transfer the Delphix Engine's registration code to a workstation with access to the external network Internet. For example, you could e-mail the registration code to an externally accessible e-mail account.
3. On a machine with access to the external Internet, please use your browser to navigate to the Delphix Registration Portal at http://register.delphix.com[606].
4. Login with your Delphix support credentials (username and password).
5. Paste the **Registration Code**.
6. Click **Register**.

> Although your Delphix Engine will work without registration, we strongly recommend that you register each Delphix Engine as part of the setup. Failing to register the Delphix Engine will impact its supportability and security in future versions

To regenerate the registration code for a Delphix Engine please refer to, Regenerating the Delphix Engine Registration Code (see page 672). Delphix strongly recommends that you regenerate this code and re-register the engine regularly to maximize the Support Security of the Delphix Engine. Delphix recommends doing this every six months.

9.12.2.4.6 Summary

The final summary tab will enable you to review your configurations for System Time, Network, Storage, Serviceability, and Authentication.

1. Click the **Back** button to go back and to change the configuration for any of these server settings.
2. If you are ready to proceed, then click **Submit**.
3. Click **Yes** to confirm that you want to save the configuration.
4. Click **Setup** to acknowledge the successful configuration.
5. There will be a wait of several minutes as the Delphix Engine completes the configuration.

606 http://register.delphix.com/

9.12.2.5 SQL server source hosts and databases

9.12.2.5.1 Source host requirements

Windows servers that will be added as Source Environments must meet the following requirements:

Source Host Requirement	Explanation
To allow Delphix-initiated backups, the service account running each SQL Server instance (the Instance Owner) should be one of: • A domain user (e.g. `MYDOMAIN\accountname`) (RECOMMENDED) • A Managed Service Account or Group Managed Service Account (`MYDOMAIN\accountname$`) (requires Windows 2012 and later, and SQL Server 2008R2 or later) • The LOCAL SYSTEM account (`NT AUTHORITY\SYSTEM`) • The NETWORK SERVICE account (`NT AUTHORITY\NETWORK SERVICE`)	Backups initiated by the Delphix Engine (Delphix Managed Backups or manual snapshots which request a backup) will fail if the Delphix Engine cannot map the Instance Owner to an Active Directory user or computer object.
The source host, proxy, and staging environments must have appropriate cross-domain trust relationships	For more information on these requirements, see the document Delphix in Multi-domain Windows Environments (see page 2022).

9.12.2.5.1.1 Recommended source Windows user requirement

Aligning to our zero-trust approach, “Delphix OS” user permissions on the Source can now be configured with the least privilege necessary from previous super-user “Backup Operator” requirements.

Recommended source Windows user requirement	Explanation
Have the "Log on as batch" privilege on the source host	This permission is required for remote PowerShell execution. This privilege can be assigned through the Local Security Policy (Local Policies → User Rights Assignment → Log on as batch job).
Have read permission for "`Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg`" on the source host	This permission is required to have access to the remote registry. Delphix uses this privilege to discover SQL Server instances and gather system details, using Windows remote registry access.
Be a member of the Users group on the Staging Host	In order to discover and query SQL Server instances as the Source Windows User, scripts are run on that user at the Staging Host.
Have the "Log on as batch" privilege on the Staging host	This permission is required for remote PowerShell execution. This privilege can be assigned through the Local Security Policy (Local Policies → User Rights Assignment → Log on as batch job).
Be able to login to each SQL Server instance that the Delphix Engine will communicate with	These requirements are described in the *Source Database Login Requirements* section below.

This recommended permission requires editing the registry on each Source host(s) so that membership of the Backup Operators group is not required. The Backup Operators group confers additional permissions that are not used by Delphix, such as being able to shut down the host.

9.12.2.5.2 Deprecated source Windows user requirements

Source Windows User Requirement	Explanation
Be a member of the Backup Operators group on the Source Host	Delphix uses this privilege to discover SQL Server instances and gather system details, using Windows remote registry access.

9.12.2.5.3 Source database login requirements

The Delphix Engine requires SQL Server logins to be created on each SQL Server instance that the Delphix Engine will communicate with:

- A database login for Environment discovery and monitoring, specified when Adding an Environment. This must be a Windows Authentication login for the Source Windows User configured in the previous section.
- A database login for dSource (Source Database) monitoring and interaction, specified when Linking a dSource. This user can be:
- The same as the Source Windows User;
- A different Windows Authentication login (this user must also have Log on as a Batch Job privileges on the Staging host); or
- A SQL Authentication login

These users must have the following permissions on each instance:

Object	Privileges Required	dSource User	Environment User	Purpose
Server	`CONNECT SQL`	X	X	Access to the SQL Server instance
Database: `master`	`db_datareader`	X	X	Access to information about attached databases
Database: `msdb`	`db_datareader`	X		Access to the backup history
Each user database to be linked	`PUBLIC`	X		Delphix will periodically run queries to check the current size of the database
Each user database to be linked	`db_backupoperator`	X		**Optional**: Required for backups to be initiated by Delphix (using Delphix Managed Backups, or when Delphix initiates a backup during a manual Snapshot)
Server	`VIEW ANY DEFINITION`	X	X	**Optional:** Required for the discovery of databases in Availability Groups

Object	Privileges Required	dSource User	Environment User	Purpose
Server	`VIEW SERVER STATE`	X	X	**Optional:** Required for the SnapSync and discovery of Availability Groups
Object: `master.dbo.sqbutility`	`EXECUTE`	X		**Optional:** Required when using backups created by Red Gate SQL Backup
Object: `master.dbo.xp_sqllightspeed_version`	`EXECUTE`	X		**Optional:** Required when using backups created by Quest LiteSpeed for SQL Server

9.12.2.5.4 List of source tables accessed by the Delphix engine

Using the db_datareader permission, the Delphix Engine accesses the following system tables in the master and msdb databases on the source host:

System table	Justification
master.sys.databases	Used to determine the name and recovery model of databases within discover SQL Server instances
master.sys.availability_groups	Used for discovering all the availability groups within an Availability Group source environment.
master.sys.availability_group_listeners	Used for discovering all the availability group listeners within an Availability Group source environment. • A requirement for dSource linking of SQL Server clustered databases (replicas) is to provide an AG (Availability Group) listener for AG cluster source discovery. This is implemented on AG cluster source discovery as a failsafe if AG cluster source database authentication configuration change down the line, ensuring the Delphix engine has a way to reach the cluster and continue certain operations.
master.sys.availability_databases_cluster	Used for discovering all the availability group clusters within an Availability Group source environment.

System table	Justification
master.sys.availability_replicas	Used for discovering all the availability group replicas within an Availability Group source environment.
master.sys.database_files	Used to determine the size of databases and whether filestream is enabled for a database
master.sys.dm_exec_requests	Used to enable Delphix to report backup operation progress
master.sys.master_files	Used to determine the primary file of a database
master.sys.filegroups	Used to determine the filegroups of a database so that Delphix can configure VDBs with the same filegroups
msdb.dbo.backupset	Used to determine new backups that have been taken. This table is regularly queried to find out if a new backup image has been taken and needs to be synchronized with Delphix.
msdb.dbo.backupmediafamily	Used to determine the physical device names of the backup files comprising a backup.

9.12.2.6 SQL server staging hosts and databases

9.12.2.6.1 Staging host requirements

Windows servers which will be added as Staging Environments must meet the following requirements:

Staging Host Requirement	Explanation
The service account running each SQL Server instance (the Instance Owner) must be one of: • A domain user (e.g. `MYDOMAIN\accountname`) (RECOMMENDED) • A Managed Service Account or Group Managed Service Account (`MYDOMAIN\accountname$`)(requires Windows 2012 and later, and SQL Server 2008R2 or later) • The LOCAL SYSTEM account (`NT AUTHORITY\SYSTEM`) • The NETWORK SERVICE account (`NT AUTHORITY\NETWORK SERVICE`)	Access to existing database backups for Snapshots and Validated Sync operations will fail if the service account cannot access backups created by the Source database instance.
The source host, proxy and staging environments must have appropriate cross-domain trust relationships.	For more information on these requirements, see the document Delphix in Multi-domain Windows Environments[607].
The edition of the installed SQL Server instance(s) must support all database features used by linked Source databases.	SQL Server may raise an error during some dSource operations if the Staging SQL Server Instance does not support features used by the Source Database. This is most easily addressed by using the same edition of SQL Server as the Source database. Features in use on the Source database can be checked using the `sys.dm_db_persisted_sku_features` dynamic view.

607 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/delphix-in-multi-domain-windows-environments

Staging Host Requirement	Explanation
Must have both the Source and Staging Windows Users configured as local Windows users.	See the sections *Source Windows User Requirements* and *Staging Windows User Requirements* for more detail.
Delphix Connector software is installed and running.	See Installing the Delphix Connector Service on the Target Database Servers[608] for instructions on installing the Delphix Connector.
Recommended iSCSI Registry settings must be in place.	See Requirements for Windows iSCSI Configuration[609] and Knowledge Base Article KBA1251[610] for instructions on applying these settings*. *Please note that in 6.0.x, these performance settings are subject to change based on further tuning efforts by Delphix Engineering.
sqlcmd command line utility must be installed on the servers hosting VDBs and staging databases. The utility should be in the Delphix operating system user's PATH environment variable.	Delphix does not run sqlcmd (or any other process) directly on the source SQL Server instance. It runs sqlcmd on the staging/connector host.

9.12.2.6.2 Staging Windows user requirements

The Delphix Engine needs a Windows domain user — for example, `MYDOMAIN\delphix_os` — to be specified when adding Staging environments to the Delphix Engine. This user must have the following permissions:

Staging User Requirement	Explanation
Be a member of the Windows "Local Administrators" group on the Staging Host	We require this permission for mounting iSCSI LUNs presented by the Delphix Engine to the staging and target hosts. Microsoft utilities used by the Delphix Engine, such as diskpart, require membership of this group[611].

608 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/replacing-self-signed-certificates-on-the-delphix-connector
609 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/requirements-for-windows-iscsi-configuration
610 https://support.delphix.com/Delphix_Virtualization_Engine/MSSQL_Server/Registry_Settings_for_Optimal_Database_Performance_and_Stability_%28KBA1251%29
611 https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/diskpart

Staging User Requirement	Explanation
Have the "Log on as batch" privilege on the Staging host	We require this permission for remote PowerShell execution. This privilege can be assigned through the Local Security Policy (Local Policies → User Rights Assignment → Log on as batch job).
Be able to access existing backups from the Source Database	If accessing existing native, Red Gate, or LiteSpeed backups, the following permissions are required: • The Staging Windows User and the service account running SQL Server must both have permission to access the database backups via SMB (Windows file sharing). • The Staging Windows User and the service account running SQL Server must both have NTFS Permissions to access the database backups. Separate documents describe requirements if Linking a dSource from a NetBackup SQL Server Backup (see page 2167) or Linking a dSource from a Commvault SQL Server Backup[612].
Be able to login to each SQL Server instance that the Delphix Engine will communicate with	These requirements are described in the *Staging Database Login Requirements* section below.

9.12.2.6.3 Staging database login requirements

The Delphix Engine requires a Windows Authentication login to be created for the Staging Windows User on each SQL Server instance that the Delphix Engine will communicate with, with the following permissions:

Object	Privileges Required	Delphix OS User (Windows Login)	Purpose
Server	`CONNECT SQL`	Y	Access to the SQL Server instance.

612 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/linking-a-dsource-from-a-commvault-sql-server-backup

Object	Privileges Required	Delphix OS User (Windows Login)	Purpose
Server	`sysadmin`	Y	The staging and target databases are managed and administered completely by Delphix. Our functionality requires many administrative operations on those databases and requires full access to them. Since database ownership can be changed by customers as part of configuring virtual databases, we require the sysadmin role to continue to administer the databases.

9.12.2.7 SQL server target hosts and databases

9.12.2.7.1 Target host requirements

Windows servers which will be added as Target Environments must meet the following requirements:

Target Host Requirement	Explanation
Delphix Connector software is installed and running	See Installing the Delphix Connector Service on the Target Database Servers (see page 2098) for instructions on installing the Delphix Connector.
Recommended iSCSI Registry settings must be in place	See Requirements for Windows iSCSI Configuration (see page 2083) and Knowledge Base Article KBA1251[613] for instructions on applying these settings.
The edition of the installed SQL Server instance(s) must support all database features used by linked Source databases	SQL Server may raise an error during some dSource operations if the Staging SQL Server Instance does not support features used by the Source Database. This is most easily addressed by using the same edition of SQL Server as the Source database. Features in use on the Source database can be checked using the `sys.dm_db_persisted_sku_features` dynamic view.

613 https://support.delphix.com/Delphix_Virtualization_Engine/MSSQL_Server/Registry_Settings_for_Optimal_Database_Performance_and_Stability_%28KBA1251%29

Target Host Requirement	Explanation
sqlcmd command line utility must be installed on the servers hosting VDBs and staging databases. The utility should be in the Delphix operating system user's PATH environment variable.	Delphix does not run sqlcmd (or any other process) directly on the source SQL Server instance. It runs sqlcmd on the staging/connector host.

9.12.2.7.2 Target Windows user requirements

The Delphix Engine needs a Windows domain user — for example, `MYDOMAIN\delphix_os` — to be specified when adding Target environments to the Delphix Engine. This user must have the following permissions:

Target User Requirement	Explanation
Be a member of the Windows "Local Administrators" group on the Target Host	We require this permission for mounting iSCSI LUNs presented by the Delphix Engine to the staging and target hosts. Microsoft utilities used by the Delphix Engine, such as diskpart, require membership of this group[614].
Have the "Log on as batch" privilege on the Target Host	We require this permission for remote PowerShell execution. This privilege can be assigned through the Local Security Policy (Local Policies → User Rights Assignment → Log on as batch job).
Be able to login to each SQL Server instance that the Delphix Engine will communicate with	These requirements are described in the *Target Database Login Requirements* section below.
Delphix Connector software is installed and running	See Installing the Delphix Connector Service on the Target Database Servers[615] for instructions on installing the Delphix Connector.

9.12.2.7.3 Target database login requirements

The Delphix Engine requires a Windows Authentication login to be created for the Target Windows User on each SQL Server instance that the Delphix Engine will communicate with, with the following permissions:

614 https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/diskpart
615 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/replacing-self-signed-certificates-on-the-delphix-connector

Object	Privileges Required	Delphix OS User (Windows Login)	Purpose
Server	`CONNECT SQL`	Y	Access to the SQL Server instance
Server	`sysadmin`	Y	The staging and target databases are managed and administered completely by Delphix. Our functionality requires many administrative operations on those databases and requires full access to them. Since database ownership can be changed by customers as part of configuring virtual databases, we require the sysadmin role to continue to administer the databases.

9.12.2.8 Supported roles for failover cluster instances and always on availability groups

Failover Cluster Instances and Always On Availability groups cannot be used as Staging Environments, and VDBs cannot be provisioned into Availability Groups.

> When adding a Failover Cluster Instance or Always On Availability Group, all nodes of the cluster must meet the requirements described in this document.

The following table shows how different SQL Server instance types should be added from the Add Environment screen:

		Environment Role		
Instance Type	**Added As**	**Source Environment**	**Staging Environment**	**Target Environment**
Standalone Instance	Standalone	Y*	Y	Y
Failover Cluster Instance (FCI)	Standalone	Y*	N	N

		Environment Role		
Failover Cluster Instance (FCI)	Cluster	N	N	Y
Always On Availability Group (AG)	Cluster	Y	N	N**

* Databases that are participating in Availability Groups will not be discovered during the discovery of a Standalone environment.

** VDBs cannot be provisioned into availability groups. However, SQL Server instances that participate in an Availability Group can also be added as Standalone Instances.

Using a Failover Cluster Instance as both Source and Target

A Failover Cluster Instance added as an environment once (as either a Source or Target environment) cannot be used as both a Source and Target.

If this is required, the environment can be added twice:

- Once as a Standalone Source environment
- Once as a Cluster Target environment

The Standalone Source environment can be used for linking dSources, and the Cluster Target environment is used for provisioning VDBs.

As suggested by the Best Practice note earlier in this article, this is not a recommended configuration. Where possible, SQL Server failover cluster instances that the Delphix Engine will use as a target should not be used to host databases other than Delphix VDBs.

9.12.2.8.1 Requirements for failover cluster target environments

The following additional requirements exist for Windows Failover Cluster Instances added as Target Environments, to be used for VDB Provisioning.

- You must first add each node in the Window Failover Cluster individually as a standalone target environment, using a non-clustered address. See Adding a SQL Server Standalone Target Environment (see page 2111).
    - The Delphix Engine may show a warning that it cannot discover the Failover Cluster Instances on each standalone host. This is expected.
- Each clustered SQL Server instance must have at least one clustered disk added to the clustered instance resource group, which can be used for creating mount points to Delphix storage.
    - The clustered drive must have a drive letter assigned to it.

    - The clustered drive must be formatted using the "GUID Partition Table (GPT)" partition style, in order for the Delphix Engine to automatically discover the drive letter as a valid option for the cluster instance. An MBR-formatted disk requires manual verification outside of Delphix that the disk has been correctly added to the MSSQL clustered resource group prior to creating the VDB. When provisioning the VDB, you must manually specify the desired MBR disk, because it will not appear in the Delphix GUI.
    - The clustered drive must be added to the clustered instance resource group as a dependency in the Failover Cluster Manager.
- Each node in the cluster must have the Failover Cluster Module for Windows PowerShell feature installed.
    - While running Windows PowerShell as an administrator, enter this command to test that the module is available: `Import-Module FailoverClusters`
- An additional target environment that can be used as a Connector Environment must exist.
    - This environment must **not** be a node in the cluster.
    - This environment should be part of the same Active Directory domain as the cluster.

9.12.2.9 Add the SQL server source environment

Delphix does not require running the Connector on your source. Instead, you'll use the Validated Sync environment as a connector environment to discover the source by proxy.

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Environments**.
4. Next to **Environments**, click the **Actions** menu, and select **Add Environment**.
5. In the **Add Environment**wizard, Host and Server tab select:
    a. Host OS: **Windows**
    b. Host Type: **Source**
    c. Server Type:
    d. If you are adding a Windows Server Failover Cluster (WSFC), add the environment based on which WSFC feature the source databases use:
        - Failover Cluster Instances Add the environment as a **standalone** source using the **cluster name** or **address**.
        - AlwaysOn Availability Groups Add the environment as a **cluster** source using the **cluster name** or **address**.
    e. Otherwise, add the environment as a **standalone** source.
6. Click **Next**.
7. In the Environment Settings tab select a **Connector Environment**. Connector environments are used as a proxy for running discovery on the source. If no connector environments are available for selection, you will need to set them up as described in Adding a SQL Server Standalone Target Environment (see page 2111). Connector environments must:

    - have the Delphix Connector installed
    - be registered with the Delphix Engine from the host machine where they are located.
8. Enter the **Environment Name**, **Node Address**, **OSUsername**, and **OSPassword** for the source environment.
9. Click **Submit.**

As the new environment is added, you will see multiple jobs running in the Delphix Admin Job History to Create and Discover an environment. In addition, if you are adding a cluster environment, you will see jobs to Create and Discover each node in the cluster and their corresponding hosts. When the jobs are complete, you will see the new environment added to the list in the **Environments** panel. If you don't see it, click the **Actions** menu and select **Refresh All**.

9.12.2.10 Linking a SQL server data source (dSource)

Linking a dSource will ingest data from the source and create a dSource object on the engine. The dSource is an object that the Delphix Virtualization Engine uses to create and update virtual copies of your database. As a virtualized representation of your source data, it cannot be managed, manipulated, or examined by database tools.

For an overview of all dSource related actions, please Managing Data Sources and Syncing Data[616].

When linking a dSource from a SQL Server source database, Delphix offers several different methods of capturing backup information:

- SQL Server Managed Backups, where the SQL Server source database schedules and initiates backups and the Delphix Engine captures them
    - Full backups
    - Full or differential backups
    - Transaction log backups (with LogSync disabled)
    - Transaction log backups (with LogSync enabled)
- Delphix Managed Backups, where the Delphix Engine schedules and initiates the backups from the source database, and captures them

9.12.2.11 Procedure

1. Login to the **Delphix Management** application.
2. Navigate to **Manage > Datasets.**
3. Click the plus icon and select **Add dSource**.
4. In the **Add dSource** wizard, select the source database with the correct environment user-specified.
5. Select user type for source database authentication and enter the login credentials. Enter username and password for Database user or Domain (Windows) user. For Environment User, select a source environment user from the dropdown list and click **Next**.

616 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/9962401/Managing+data+sources+and+syncing+data

6. Enter a name and select a group for your dSource. Adding a dSource to a dataset group lets you set Delphix Domain user permissions for that database and its objects, such as snapshots. See the topics under Users and Groups (see page 674) for more information.
7. Select the **Data Management** settings needed. For more information, Data Management Settings for SQL Server Data Sources (see page 2158).
8. Select the Staging environment and SQL Instance that will be used to manage the staging database used for validated sync of the dSource.
9. Select any policies for the new dSource.
10. Enter any scripts that should be run on the **Hooks** page.
11. Review the dSource Configuration and Data Management information, and then click **Submit**.

9.12.2.12 Provisioning a SQL server virtual database (VDB)

1. Login to the **Delphix Management** application.
2. Select **Manage** > **Datasets**.
3. Select a **dSource**.
4. Click **Timeflow** tab.
5. Next to a snapshot select the

   Provision VDB icon. The **Provision VDB** panel will open, and the **Database Name** and **Recovery Model** will auto-populate with information from the dSource.
6. Select a **target environment**.
7. Select an **Instance** to use.
8. If the selected target environment is a Windows Failover Cluster environment, select a drive letter from **Available Drives**. This drive will contain volume mount points to Delphix storage.

> Windows Cluster Volume Management Software Requirements
>
> Only cluster volumes managed by the native Windows Volume Manager are supported. For example, cluster volumes managed by Veritas VxVM are not supported.
>
> If you use third-party volume management software, create a new LU (recommended to be 10GB in size) and add this LU as a clustered resource to the SQL Server instance using native Windows volume management tools. Assign a drive letter for this LU. You can then use this LU as the volume mount point location for Delphix VDB provisioning

9. Enter a **VDB Name** and select a **Target Group** for the VDB.

10. Enable **Auto VDB Restart** to allow the Delphix Engine to automatically restart the VDB when it detects target host reboot.
11. Click **Next**.
12. Select a **Snapshot Policy** for the VDB. Click **Next**.
13. Specify any **Pre-** or **Post-Scripts** that should be used during the provisioning process.
14. Click **Next**.
15. The final summary tab will enable you to review your configurations.
16. Click **Submit**.

When provisioning starts, the VDB will appear in the **Datasets** panel. Select the VDB and navigate to the **Status** tab to see the progress of the job. When provisioning is complete, you can see more information on the **Configuration** tab.

> You can select a SQL Server instance that has a higher version than the source database and the VDB will be automatically upgraded. For more information about compatibility between different versions of SQL Server, see SQL Server Support Matrix.

9.12.2.12.1 Provisioning by snapshot or logSync

When provisioning by snapshot, you can provision to the start of any particular snapshot, either by time or LSN.

> You can take a new snapshot of the dSource and provision from it by clicking the **Camera** icon.

Provisioning By Snapshot	Description
Provision by Time	You can provision to the start of any snapshot by selecting that snapshot card from the **TimeFlow** tab, or by selecting the time icon and entering a value in the time entry fields. The values you enter will snap to the beginning of the nearest snapshot.
Provision by LSN	You can use **Provision by LSN** control to open the LSN entry field. Here, you can type or paste in the LSN to which you want to provision. After entering a value, it will "snap" to the start of the closest appropriate snapshot. Provisioning a SQL Server VDB Procedure

9.12.2.13 Next steps

Congratulations! You have provisioned your first virtual database!

Now, perform some simple functional tests with your application. You can connect your app to the VDB using standard TNS/JDBC techniques. Delphix has already registered the VDB for you on the target listener.

We suggest the following next steps:

1. Drop a table and use the VDB Rewind feature to test the recovery of your VDB.
2. Take a snapshot of your dSource and refresh your VDB to quickly get fresh production data.
3. Provision a new VDB from your VDB to test sharing data quickly with other teams.
4. Mask your new VDB to protect sensitive data. Provision new VDBs from that masked VDB to quickly provide safe data to the development and QA teams.

9.12.3 SQL Server support and requirements

This section covers the following topics:

- SQL Server matrix (see page 2052)
- Overview of requirements for SQL Server environments (see page 2067)
- Requirements for Windows iSCSI configuration (see page 2083)
- Network access requirements for SQL Server (see page 2089)

9.12.3.1 SQL Server matrix

- Delphix Support Policies specifically list Major and Minor release coverage. If a minor release is listed as covered, then all patch releases under that minor release are certified.
- Delphix only supports 64-bit operating systems.

Key:

Color	Supported?
Y	Yes
NA	Not Applicable

9.12.3.1.1 Windows server

Supported OS version	Supported DBMS version					
	SQL Server 2012	SQL Server 2014	SQL Server 2016	SQL Server 2017	SQL Server 2019	SQL Server 2022
Windows Server 2012	Y	Y	Y	Y 5.2.3+	NA	NA
Windows Server 2012 R2	Y	Y	Y	Y 5.2.3+	NA	NA
Windows Server 2016	Y 5.1.8+	Y 5.1.8+	Y 5.1.8+	Y	Y 5.3.7+	Y 8.0.0+
Windows Server 2019	NA	NA	Y 5.3.3+	Y 5.3.3+	Y 5.3.7+	Y 8.0.0+
Windows Server 2022	NA	NA	NA	Y 6.0.13+	Y 6.0.13+	Y 8.0.0+

Windows support

Delphix supports both "Windows with Desktop Experience" and Windows Core (Windows without Desktop Experience) server installations.

Windows support

After applying Windows updates or .NET updates for Windows Server 2016 or 2019, servers used to host Delphix VDBs may experience the following symptoms:

- CPU usage on the Windows target host may increase significantly, possibly reaching 100%.
- PowerShell may take up to 10 seconds to launch from the command line.

For more information on how to resolve this issue, see KBA6024[617]

64-bit Windows only

Delphix supports only 64-bit versions of Windows on VDB Target Hosts and Staging Target Hosts. This restriction does not apply to Source Hosts.

Check OS compatibility

The Windows Server OS versions on the Staging Target Hosts and VDB Target Hosts must be compatible. Please refer to the compatibility matrices below.

SQL server failover clusters

There are additional restrictions on supported Windows and SQL Server versions for SQL Server Failover Cluster target environments.

For details, see Adding a SQL Server Failover Cluster Target Environment (see page 2113)

9.12.3.1.2 Supported SQL versions

SQL server version	Delphix version
SQL Server 2012 (11.0)	Delphix 3.1.2 and beyond
SQL Server 2014 (12.0)	Delphix 4.1.3 and beyond
SQL Server 2016 (13.0)	Delphix 5.1.4 and beyond
SQL Server 2017 (14.0)	Delphix 5.2.3 and beyond
SQL Server 2019 (15.0)	Delphix 5.3.7 and beyond

617 https://support.delphix.com/Delphix_Virtualization_Engine/MSSQL_Server/Resolving_high_CPU_on_Windows_VDB_hosts_after_applying_Windows_Updates_(KBA6024)

SQL server version	Delphix version
SQL Server 2022 (16.0)	Delphix 8.0.0 and beyond

9.12.3.1.3 Supported SQL server editions

- Standard
- Enterprise
- Developer

SQL server editions

For Staging Target Hosts, you can use SQL Server Standard Edition only if the source databases do not use SQL Server Enterprise Edition specific features, such as partitioned tables.

However, it is important for VDB target hosts to use the same edition of SQL Server software as the source database so that all features available in the source are also available in the VDB.

9.12.3.1.4 Supported Windows connector versions

You can always download the current Windows connector version from the Delphix Engine from the following locations:

Delphix Dynamic Data Platform versions before 6.0.5.0:

```
http://<name of your Delphix Engine>/connector/DelphixConnectorInstaller.msi
```

Delphix Dynamic Data Platform versions 6.0.5.0 onwards:

```
http://<name of your Delphix Engine>/connector/DelphixConnectorInstaller.exe
```

The Delphix Connector and Engine are generally backward-compatible.

Delphix connector version 1.18.0 and beyond is compatible with Delphix Engine 6.0.5.0 and later releases.

Delphix connector version 1.17.0 or earlier is compatible with Delphix Engine 6.0.4.0 and earlier releases

9.12.3.1.5 Checking the windows connector version

After installing the connector, you can perform the following steps to determine the Windows connector version.

1. Navigate to the Windows Environment.
2. In the search box on the taskbar, type Add or Remove Programs or launch **Start** > **ControlPanel** > **Programs** > **Programs and Features**.
3. On the Add or Remove Programs screen, locate Delphix Connector and then click on the software to check the version installed.

9.12.3.1.6 Windows connector matrix

The following table details the supported Windows Connector and JRE versions for the corresponding Delphix Dynamic Data Platform versions.

> To know more about the changelogs for the respective Windows connector version, refer to the Windows connector changelog (see page 2065).

Delphix dynamic data platform version	Windows connector version	JRE version
4.2	1.2.0	1.7.0_71 (4.2.5.1)
4.3	1.3.0	1.8.0_40 (4.3.5.1)
5.1.0.0-5.1.4.0	1.4.0	1.8.0_40
5.1.5.0 - 5.1.6.0	1.5.0	1.8.0_112
5.1.7.0	1.6.0	1.8.0_112
5.1.8.0 - 5.1.10.0	1.7.0	1.8.0_112
5.2.0.0-5.2.2.0	1.9.0	1.8.0_131 (5.2.2.1)
5.2.3.0-5.2.4.0	1.10.0	1.8.0_162

Delphix dynamic data platform version	Windows connector version	JRE version
5.2.5.0-5.3.3.0	1.12.0	1.8.0_202
5.3.4.0	1.13.0	1.8.0_202
5.3.5.0	1.14.0	1.8.0_202
5.3.6.0-5.3.7.0	1.15.0	1.8.0_202
5.3.8.0-6.0.0.0	1.16.0	1.8.0_202
6.0.1.0-6.0.4.0	1.17.0	OpenJDK 8u242
6.0.5.0-6.0.8.0	1.18.0	OpenJDK 8u262-b10
6.0.9.0-6.0.10.0	1.20.0	OpenJDK 8u282-b08
6.0.11.0	1.21.0.0	OpenJDK 8u302-b08
6.0.12.0	1.22.0.0	OpenJDK 8u302-b08
6.0.12.1	1.22.0.0	OpenJDK 8u302-b08
6.0.13.0	1.23.0.0	OpenJDK 8u302-b08
6.0.13.1	1.23.0.0	OpenJDK 8u302-b08
6.0.14.0	1.23.0.0	OpenJDK 8u302-b08
6.0.15.0	1.24.0.0	OpenJDK8u332-b09
6.0.16.0	1.24.0.0	OpenJDK8u332-b09
6.0.17.0	1.27.0.0	OpenJDK 8u345-b01
7.0.0.0	1.27.0.0	OpenJDK 8u345-b01

Delphix dynamic data platform version	Windows connector version	JRE version
8.0.0.0	1.27.0.0	OpenJDK 8u345-b01
9.0.0.0	1.28.0.0	OpenJDK 8u362-b09
10.0.0.0	1.29.0.0	OpenJDK 8u362-b09
11.0.0.0	1.30.0.0	OpenJDK 8u362b09
12.0.0.0	1.30.0.0	OpenJDK 8u362b09
13.0.0.0	1.30.0.0	OpenJDK 8u362b09
14.0.0.0	1.31.0.0	OpenJDK 8u362b09
15.0.0.0	1.31.0.0	OpenJDK 8u362b09
16.0.0.0	1.31.0.0	OpenJDK 8u362b09
17.0.0.0	1.32.0.0	OpenJDK 8u382b05
18.0.0.0	1.33.0.0	OpenJDK 8u382b05
19.0.0.0	1.34.0.0	OpenJDK 8u382b05
20.0.0.0	1.34.0.0	OpenJDK 8u382b05
21.0.0.0	1.34.0.0	OpenJDK 8u382b05
22.0.0.0	1.35.0.0	OpenJDK 8u382b05

The following table details the .Net connector version compatibility for the corresponding Windows Connector versions.

.Net connector version	Windows connector version
3.5	1.17.0 and earlier
3.5 and 4.0+	1.18.0 and higher

9.12.3.1.7 File checksum / hashes

The following table provides the MD5 and SHA256 file hash for recent Delphix versions, in the instance that a security policy requires an exception for software installation.

Delphix dynamic data platform version	Connector version	MD5 Hash	SHA256 Hash
6.0.0.0	1.16.0	84616c4f86c0d871127feb26b9b8bd52	bc2e5e4b26c33d739f7be12f53637979481e1371325e0b5c7645462f6cf5f4f3
6.0.1.0	1.17.0	bb6544dcff3f22d6bf9e20429b1cdb0c	8d914b77d2c5aa0695c3fdbe7d7394e8ae051c091f6f5ee59185cb1d9d157542
6.0.1.1	1.17.0	dd3ea3bedf92a1f8d7e89228315ee315	75ed8aa2cf2689916e0b92de6c2aa2fa7660003d62ff07e99978f95dcc9f6abc
6.0.2.0	1.17.0	4770f8c1ad46430a1e4b8764ad687ff5	a98f1c1f90d5053f62d74c1a74bc27b3b48f68a3e067dce04ce9ab2ccf484241
6.0.2.1	1.17.0	97ec3d13db50ad843625b7757c1a2360	7c2ab497ae4ac27b907611e976d7068ed8eaf54e52b836591c082bbe01079512
6.0.3.0	1.17.0	3154c7a5fd7bcb4f0e2c66d328290e6a	efb69b2df884705e488d3405a3958c3bb5457142cbfc71990559bb549c2e8c37
6.0.3.1	1.17.0	2f73be2cf8ceef97cf519c716237462d	7fa7bc269bbc944568231f93ac17ba919b494aa43009967245b75f4abe576a73
6.0.4.0	1.17.0	6a4b7f0a5d180438ff199b2b464506cd	971b88456574e8e11906b5e6850ea46f012490dcb789a5af73f37d71158ffebf

Delphix dynamic data platform version	Connector version	MD5 Hash	SHA256 Hash
6.0.4.1	1.17.0	ea9a564d0cae1066cd9607824292fbd7	674f9896360a351b314d6aa6e624f9270ef9e3e8d113baddf028581f1ddddec9
6.0.4.2	1.17.0	c5f855c1fa26f5cd3d84d70a94fc0480	07d71acafd559e15782e7145225734adf5e50356d2f8abe5a52bbd90f3a44cbe
6.0.5.0	1.18.0	a3ac768989a6c1dae41adf9452548236	71525c31edcf321d2c0220f4edf8932e249e2cd14b261054af1859574dc9e5eb
6.0.6.0	1.18.0	acb09cbd754e9c25d1afa17b35321cd4	3a51bd3b05148dc3563c5b36d83d44ac725295c23c4b7e9c28f54263058fe2b4
6.0.6.1	1.18.0	32202760adb743a62851f8b698a31bf3	ac16e8a5c37807c2fa775ecc18f9e0387f5bd9d235c9e8854275c99d5bf865a1
6.0.7.0	1.18.0	2fed8da4b986b95a7efb9a9a148239ba	2c2a83bad568617bf5dd36b84ad0de22b811f113acac45124ed0effbeb6aef17
6.0.8.0	1.18.0	5bf0b950cdcad8c39db60e1a7406d985	eae5958878a22b4f4f0456210c031ef9c11d064a05ca6cb85535f0e7dec9c0a1
6.0.8.1	1.18.0	a3ac768989a6c1dae41adf9452548236	71525c31edcf321d2c0220f4edf8932e249e2cd14b261054af1859574dc9e5eb
6.0.9.0	1.20.0	127af1d95372f84feb373ab4f7b280b2	b7c7aaceb2b04aebfbd0ba2521dd15520159d54877b0543567b892ee2584c7ce
6.0.10.0	1.20.0	e8a39a57e5a77d106d8040a4537bd8f0	7d4501bf5f1cf22a7039e8f7429d4b42c0d11b1360434380b174830f1155646e
6.0.10.1	1.20.0	baf1223b73e626798dd404bd4d18a3c9	b827d9213e933fce8389322fe25c8d2e1db57ab66709d236b9a8cd5e254bb89d
6.0.11.0	1.21.0.0	412c4c32c7aefb973c21b573dca66d2a	9b689a21249456ddf623042cd2595820531cfbfca0703e713d9795036eef49b7

Delphix dynamic data platform version	Connector version	MD5 Hash	SHA256 Hash
6.0.12.0	1.22.0.0	dc7a97c46ef9e2d91285dcb0712eaf61	db544774ac3eb3235bf90deef3e93962f3955e86bd9f14309ecb0fba9d41dd7f
6.0.12.1	1.22.0.0	8cf9e955438c6393031c5596ca0420cd	130249658f68eab2a2263deceed6a9de6dd54eead66da68886272f3f40b2f9af
6.0.13.0	1.23.0.0	31cc5d3869e4c7f93d38014e5bd40b27	de1c2fd4c79dcb7391a51774fba1821456ecc031283db5b497145cce383005f2
6.0.13.1	1.23.0.0	3236df25c2e0ab6588bcb54974228b52	c18adb5e0cce7e40e9fa9552b022973ae468c746aca311df2e58c46ef09749ac
6.0.14.0	1.23.0.0	5913ddb3336d571e3ecec1327dcc441c	11fc899f72239061ff5eaf8d65dcd6de467ba1adcaa7f70c1143f04f22cf14d64
6.0.15.0	1.24.0.0	df30e6dc33ef48c839fafdc10726c3a6	e9f0a139d759b366165fc8ac10361a1a9a0518692f8d4a08cae64c889fd9e753
6.0.16.0	1.24.0.0	bd3a63d94b7cf6ec2e3c4b7b0d72415f	81a91fb9383d125ca123ca959ea69fe980d01dcfe35474774618b34d2f2a8a84
6.0.17.0	1.27.0.0	ae36f9bdaef53fc967f6504f679ba6c7	f036069d6ec056a0a5ab7e38237d05713f207e84c27dcb65c049bafcc9670214
7.0.0.0	1.27.0.0	3257ed8180aaa53b486afe5b78cf00c5	742cc206acc59942d4773943a6ecaa978be58c95aa3de65b3621b9fa9b791817
8.0.0.0	1.27.0.0	911b83c5ad4fbdb907c11ab3d0ddac18	ea0168077bc104ab59b42b6d76f86000f1f0f3b9ee185be1df1a0f6a5cb1c91f
9.0.0.0	1.28.0.0	64a245d517f05e72ca80453a6d0bf43c	e8f038626c3e1fd9e5bd1c3a2b30df6e0b5e192713005a03ca3a3d0ab329badf
10.0.0.0	1.29.0.0	aa88f148d7fbf2988522841aac780020	b6b5d54be508d9efc2362898a8a8622462fdc42ac15fc3e63d4783edde4c61ee

Delphix dynamic data platform version	Connector version	MD5 Hash	SHA256 Hash
11.0.0.0	1.30.0.0	47a1c9c5817e64f3f68b17e766cac0a	6266b1f4084c2ea48d922ab5760adad40a730745cd27bfc4bcdcadae7db44bd0
12.0.0.0	1.30.0.0	a5ba7f208b474736ee4e0595155d018e	1622ff824abd56ee8921a75bde4e7012075c31b96aae7fb88ee5cb3d3a3af271
13.0.0.0	1.30.0.0	4e2494ca87ff40f4a17461c3c336dd9d	1206eb0797cdd5356be39d9822fe4944ada5426734b4f826e5bb3b47c79deb4e
14.0.0.0	1.31.0.0	bf6a02d54829d974addf7e54744e4ee7	f185bdca7bd771dda49005d21d03f9cf57dac0f7a909eaab38fab25cbec59dc0
15.0.0.0	1.31.0.0	32936765dd8ce3ad622431aaff7736bc	45cb3697eb5e82420774ea79c4645a77ce2502fe4751e281ccba259416889cdd
16.0.0.0	1.31.0.0	dc1dad5da82952461d978c6a4c88d78f	0da5ae67ee82d2145a6234be27ffe562d289d5cb73ecd32ef30717d35e7c3e03
17.0.0.0	1.32.0.0	34cac1765a2e16da32c05db02714ccb6	8e9eef8feb025b55e3a9f01a0e5bd4293e72d12564a7cec47cae9c311ac6266a
18.0.0.0	1.33.0.0	d1465438ca841c49a1263cc525dfb8c0	6087d2aeea4bbb6abe178b7ac83c382f7b561c43c21086e35f04f8bcc822e681
19.0.0.0	1.34.0.0	011a7f9c6393f44b50e23ff3ccd7c58d	80d50f7892f65b963c6261f81851973d77c9589d70e92a7bfad9acb91806ee8c
20.0.0.0	1.34.0.0	857da97353bc54a6e4b27d5cfb8c56cf	dd7b1bc5c1a4aa8ddcae19d216cca95c9c4c2d7ad478e38574c9315daddd8e69
21.0.0.0	1.34.0.0	82a808fd78031c850fc2bc56526ac147	cce3c0058e0f2b1a6a1e024519d60d9bedb5f5f83009452e31d9a71ca5f5ab99
22.0.0.0	1.35.0.0	075b6146101bc332d0d4ac817b411666	e425a84f7f0abbccc9c05a0c215c96a24462a3bbbcf95183ff372a96d38db20d

9.12.3.1.8 Source environment operating system compatibility

Source environments can be running any supported Windows operating system version. There are no compatibility requirements between the source environment's operating system and that of the target environments. For more information, see Overview of Setting Up SQL Server Environments (see page 2092).

9.12.3.1.9 Staging and target environment operating system version compatibility

The operating system version on the target environment that will contain the virtual databases must be equal to or higher than that on the staging target.

9.12.3.1.10 Source and staging environment SQL server version compatibility

The SQL Server version on the staging environment must be equal to that on the source environment.

Provisioning to higher SQL versions when the source is SQL server 2005

For SQL Server 2005, direct provisioning to higher SQL Server versions is only supported for provisioning to SQL Server 2012 or higher. You can first provision a VDB to SQL Server 2005 and then upgrade it to a higher version by following the steps outlined in Upgrading SQL Server VDBs (see page 2190)

9.12.3.1.11 Supported SQL server backup software

Delphix currently supports the following software for dSource backups:

- SQL Server native backups - all formats, including but not limited to
  - Striped
  - Compressed
  - Append
  - Overwrite
- Veritas NetBackup
  - Delphix currently supports NetBackup v7.7.3, v8.0, and v8.1.
  - The version of the NetBackup client on the staging environment must be the same as that on the source.
  - If the dSource is backed up with NetBackup, the source and staging environments must have a NetBackup client installed.
    VDB target environments do not need to have NetBackup installed.
  - Logsync (point-in-time provision) is currently not supported.

    - See Linking a dSource from a NetBackup SQL Server Backup (see page 2167) for more NetBackup requirements and additional information about NetBackup.
- Commvault
    - Delphix currently supports Commvault v11.
    - The version of Commvault SQL Agent on the staging environment must be the same as that on the source.
    - If the dSource is backed up with Commvault, the source and staging environment must have Commvault SQL Agent installed.
      VDB target environments do not need to have Commvault SQL Agent installed.
    - Commvault SQL Agent on staging environment must be registered to the CommServe server, which is managing source database backups.
    - Logsync (point-in-time provision) is currently not supported.
    - See Linking a dSource from a Commvault SQL Server Backup (see page 2162) for more Commvault requirements and additional information about Commvault.
- Quest/NetVault LiteSpeed
    - Delphix currently supports LiteSpeed v5.0.0.0 to v8.9.
    - The LiteSpeed version on the staging environment must be the same or higher than that on the source.
    - If the dSource is backed up with LiteSpeed, the source and the staging environments must also have LiteSpeed installed.
      VDB target environments do not need to have LiteSpeed installed.
    - On source SQL Server Instance, the dSource user must be granted execute privilege on extended stored procedure master.dbo.xp_sqllightspeed_version.
- Red Gate SQL Backup Pro
    - Delphix currently supports SQL Backup Pro v7.3 to v7.7, v8.x, v9.x, and v10.0.
    - The version of SQL Backup Pro on the staging environment must be the same as that on the source.
    - On source SQL Server Instance, the dSource database user must be granted execute privilege on extended stored procedure master.dbo.sqbutility.

If the dSource is backed up with SQL Backup Pro, the source and the staging environments must have SQL Backup Pro installed.
VDB target environments do not need to have SQL Backup Pro.

9.12.3.1.12 Source and provisioning environment SQL server compatibility matrix

When provisioning a VDB, the SQL Server version on the target should be equal to or higher than that on the source.

	Provisioning target environment					
	SQL Server 2012	**SQL Server 2014**	**SQL Server 2016**	**SQL Server 2017**	**SQL Server 2019**	**SQL Server 2022**
Source Environment						
SQL Server 2012	**X**	**X**	**X**	**X**	**X**	**X**
SQL Server 2014		**X**	**X**	**X**	**X**	**X**
SQL Server 2016			**X**	**X**	**X**	**X**
SQL Server 2017				**X**	**X**	**X**
SQL Server 2019					**X**	**X**
SQL Server 2022						**X**

9.12.3.1.13 Windows connector changelog

9.12.3.1.13.1 What's new

Version	New feature	Continuous Data Engine version
1.35.0.0	The Delphix SQL Server user base is rapidly expanding, and as a result, our largest customers have encountered an increase in login entries to their Domain Controllers from Windows target hosts, as well as from environment monitoring to targets and sources. This release implements enhancements in the Delphix code to optimize Active Directory logon requests to the Domain Controller.	22.0.0.0 and later

9.12.3.1.13.2 Fixed issues

Bug number	Description	Fixed in version	Continuous Data Engine version
DLPX-89300	Fixed an issue related to MSSQL full backup hangs by logging the `robocopy` command output to the log file, at the Delphix connector logs folder, during sync operation.	1.34.0.0	19.0.0.0 - 21.0.0.0
DLPX-79498	Fixed an error message when the Delphix Connector service is already installed with the previous version.	1.33.0.0	18.0.0.0
DLPX-88488	Fixed an issue where not enough samples are collected for performance analysis.	1.32.0.0	17.0.0.0
DLPX-82882	Replacing Self-signed Certificates, on the Delphix Connector.	1.28.0.0	9.0.0.0
DLPX-83043	Weak DH 1024-bit exchange key detected by security scanner for the Delphix connector.	1.27.0.0	6.0.17.0 - 8.0.0.0
DLPX-83149	From the Windows connector side, while handshaking, stop using the cipher suites that use Diffie-Hellman key exchange with keys less than 2,048 bits in size.	1.27.0.0	6.0.17.0 - 8.0.0.0
DLPX-73533	Added a fix to prevent multiple connector operations testing for the presence of the same SCRIPT directory. Hence, improved performance by reducing the number of remote PowerShell calls performed during connector operations.	1.27.0.0	6.0.17.0 - 8.0.0.0
DLPX-78791	This release upgrades log4j from 1.2.17 to the latest 2.x in Windows connector.	1.22.0.0	6.0.12.0
DLPX-75335	Added a product name and product version for the Delphix Connector executable so this information can be available before installation.	1.21.0.0	6.0.11.0

Bug number	Description	Fixed in version	Continuous Data Engine version
DLPX-28435	MS SQL instances PatchLevel will be displayed in preference to the version on the UI.	1.20.0	6.0.9.0 - 6.0.10.0

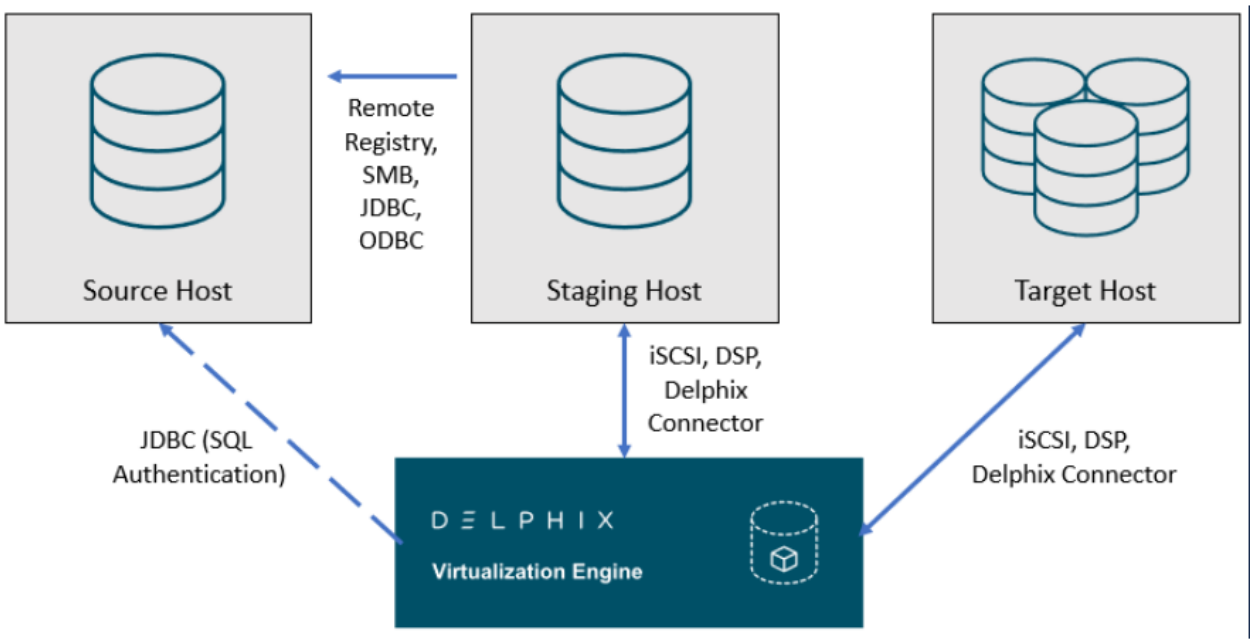
9.12.3.2 Overview of requirements for SQL Server environments

9.12.3.2.1 SQL server staging hosts and databases

This document identifies the requirements for interactions between the Delphix Engine and SQL Server environments and outlines the set of system tables to which we currently require access.

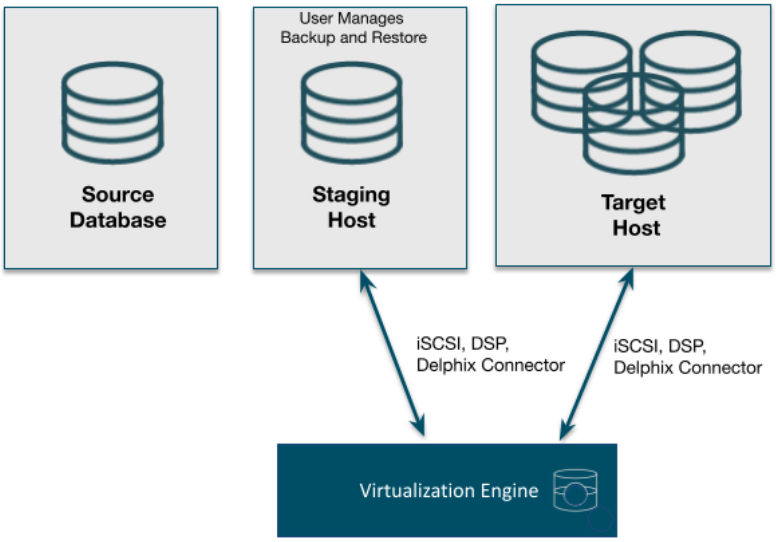
9.12.3.2.2 Delphix SQL server architectural diagram - Traditional pull architecture

This diagram depicts the environments and hosts with which the Delphix Engine interacts. Each type of environment has different requirements, which are described below.



9.12.3.2.3 Delphix SQL server architectural diagram - staging push architecture

This diagram depicts the Staging and target hosts with which the Delphix Engine interacts. Note that the Delphix Engine does not interact with the Source Database. For more details on the Staging Push mechanism, see Staging Push Mechanism with SQL Server (see page 2021) and Staging Push Implementation for SQL Server. (see page 2145)



9.12.3.2.4 SQL server source hosts and databases

9.12.3.2.5 Source host requirements

Windows servers that will be added as Source Environments must meet the following requirements:

> Source databases cannot be in Read-Only mode at the time that a backup is taken. While the snapshot will succeed, attempts to provision VDBs from those snapshots may fail with an error during the provision process.
>
> Temporarily setting the database to read-write mode, and taking a new backup and snapshot while the database is in in this state, will allow VDBs to be provisioned successfully.

Source Host Requirement	Explanation
To allow Delphix-initiated backups, the service account running each SQL Server instance (the Instance Owner) should be one of: • A domain user (e.g. `MYDOMAIN\accountname`) (RECOMMENDED) • A Managed Service Account or Group Managed Service Account (`MYDOMAIN\accountnameundefined`) (requires Windows 2012 and later, and SQL Server 2008R2 or later) • The LOCAL SYSTEM account (`NT AUTHORITY\SYSTEM`) • The NETWORK SERVICE account (`NT AUTHORITY\NETWORK SERVICE`)	Backups initiated by the Delphix Engine (Delphix Managed Backups or manual snapshots which request a backup) will fail if the service account cannot access backups created by the Source database instance
The source host, proxy and staging environments must have appropriate cross-domain trust relationships	For more information on these requirements, see the document Delphix in Multi-domain Windows Environments. (see page 2022)

9.12.3.2.5.1 Recommended source Windows user requirements

Aligning to our zero-trust approach, “Delphix OS” user permissions on the Source can now be configured with the least privilege necessary from previous super-user “Backup Operator” requirements.

Recommended Source Windows User Requirements	Explanation
Have the "Log on as batch" privilege on the source host	This permission is required for remote PowerShell execution. This privilege can be assigned through the Local Security Policy (Local Policies → User Rights Assignment → Log on

Recommended Source Windows User Requirements	Explanation
Have read permission for `"Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg"` on the source host	This permission is required to have access to the remote registry. Delphix uses this privilege to discover SQL Server instances and gather system details, using Windows remote registry access.
Be a member of the Users group on the Staging Host	In order to discover and query SQL Server instances as the Source Windows User, scripts are run on that user at the Staging Host.
Have the "Log on as batch" privilege on the Staging host	This permission is required for remote PowerShell execution. This privilege can be assigned through the Local Security Policy (Local Policies → User Rights Assignment → Log on as batch job).
Be able to login to each SQL Server instance that the Delphix Engine will communicate with	These requirements are described in the Source Database Login Requirements (see page 0) section below.

This recommended permission requires editing the registry on each Source host(s) so that membership of the Backup Operators group is not required. The Backup Operators group confers additional permissions that are not used by Delphix, such as being able to shut down the host.

9.12.3.2.5.2 Deprecated source Windows user requirements

Source Windows User Requirement	Explanation
Be a member of the Backup Operators group on the Source Host	Delphix uses this privilege to discover SQL Server instances and gather system details, using Windows remote registry access.
Be a member of the Users group on the Staging Host	In order to discover and query SQL Server instances as the Source Windows User, scripts are run on that user at the Staging Host.
Have the "Log on as batch" privilege on the Staging host	We require this permission for remote PowerShell execution. This privilege can be assigned through the Local Security Policy (Local Policies → User Rights Assignment → Log on as batch job).

Source Windows User Requirement	Explanation
Be able to login to each SQL Server instance that the Delphix Engine will communicate with	These requirements are described in the Source Database Login Requirements (see page 0) section below.

9.12.3.2.5.3 Source database login requirements

The Delphix Engine requires SQL Server logins to be created on each SQL Server instance that the Delphix Engine will communicate with:

- A database login for Environment discovery and monitoring, specified when Adding an Environment. This must be a Windows Authentication login for the Source Windows User configured in the previous section.
- A database login for dSource (Source Database) monitoring and interaction, specified when Linking a dSource. This user can be:
  - The same as the Source Windows User;
  - A different Windows Authentication login (this user must also have Log on as a Batch Job privileges on the Staging host); or
  - A SQL Authentication login

These users must have the following permissions on each instance:

Object	Privileges Required	dSource User	Environment User	Purpose
Server	`CONNECT SQL`	Y	Y	Access to the SQL Server instance
Database: `master`	`db_datareader`	Y	Y	Access to information about attached databases

Object	Privileges Required	dSource User	Environment User	Purpose
Database: `msdb`	`db_datareader`	Y		Access to the backup history
Each user database to be linked	`PUBLIC`	Y		Delphix will periodically run queries to check the current size of the database
Each user database to be linked	`db_backupoperator`	Y		Optional: Required for backups to be initiated by Delphix (using Delphix Managed Backups, or when Delphix initiates a backup during a manual Snapshot)
Server	`VIEW ANY DEFINITION`	Y	Y	Optional: Required for the discovery of databases in Availability Groups
Server	`VIEW SERVER STATE`	Y	Y	Optional: Required for the SnapSync and discovery of Availability Groups
Object: `master.dbo.sqbutility`	`EXECUTE`	Y		Optional: Required when using backups created by Red Gate SQL Backup

Object	Privileges Required	dSource User	Environment User	Purpose
Object: `master.dbo.xp_sqllightspeed_version`	`EXECUTE`	Y		Optional: Required when using backups created by Quest LiteSpeed for SQL Server

9.12.3.2.5.4 List of source tables accessed by the Delphix engine

Using the db_datareader permission, the Delphix Engine accesses the following system tables in the master and msdb databases on the source host:

System table	Justification
master.sys.databases	Used to determine the name and recovery model of databases within discover SQL Server instances
master.sys.availability_groups	Used for discovering all the availability groups within an Availability Group source environment.
master.sys.availability_group_listeners	Used for discovering all the availability group listeners within an Availability Group source environment. • A requirement for dSource linking of SQL Server clustered databases (replicas) is to provide an AG (Availability Group) listener for AG cluster source discovery. This is implemented on AG cluster source discovery as a failsafe if AG cluster source database authentication configuration change down the line, ensuring the Delphix engine has a way to reach the cluster and continue certain operations.
master.sys.availability_databases_cluster	Used for discovering all the availability group clusters within an Availability Group source environment.

System table	Justification
master.sys.availability_replicas	Used for discovering all the availability group replicas within an Availability Group source environment.
master.sys.database_files	Used to determine the size of databases and whether filestream is enabled for a database
master.sys.dm_exec_requests	Used to enable Delphix to report backup operation progress
master.sys.master_files	Used to determine the primary file of a database
master.sys.filegroups	Used to determine the filegroups of a database so that Delphix can configure VDBs with the same filegroups
msdb.dbo.backupset	Used to determine new backups that have been taken. This table is regularly queried to find out if a new backup image has been taken and needs to be synchronized with Delphix.
msdb.dbo.backupmediafamily	Used to determine the physical device names of the backup files comprising a backup.

9.12.3.2.6 SQL server staging hosts and databases

9.12.3.2.6.1 Staging host requirements

Windows servers which will be added as Staging Environments must meet the following requirements:

Staging Host Requirement	Explanation
The service account running each SQL Server instance (the Instance Owner) must be one of: • A domain user (e.g. `MYDOMAIN\accountname`) (RECOMMENDED) • A Managed Service Account or Group Managed Service Account (`MYDOMAIN\accountname$`)(requires Windows 2012 and later, and SQL Server 2008R2 or later) • The LOCAL SYSTEM account (`NT AUTHORITY\SYSTEM`) • The NETWORK SERVICE account (`NT AUTHORITY\NETWORK SERVICE`)	Access to existing database backups for Snapshots and Validated Sync operations will fail if the service account cannot access backups created by the Source database instance.
The source host, proxy and staging environments must have appropriate cross-domain trust relationships	For more information on these requirements, see the document Delphix in Multi-domain Windows Environments. (see page 2022)
The edition of the installed SQL Server instance(s) must support all database features used by linked Source databases	SQL Server may raise an error during some dSource operations if the Staging SQL Server Instance does not support features used by the Source Database. This is most easily addressed by using the same edition of SQL Server as the Source database. Features in use on the Source database can be checked using the `sys.dm_db_persisted_sku_features` dy namic view.
Must have both the Source and Staging Windows Users configured as local Windows users	See the sections *Source Windows User Requirements* and *Staging Windows User Requirements* for more detail.

Staging Host Requirement	Explanation
Delphix Connector software is installed and running	See Installing the Delphix Connector Service on the Target Database Servers (see page 2098)for instructions on installing the Delphix Connector.
Recommended iSCSI Registry settings must be in place	See Requirements for Windows iSCSI Configuration (see page 2083) and Knowledge Base Article KBA1251[618] for instructions on applying these settings*. *Please note that in 6.0.x, these performance settings are subject to change based on further tuning efforts by Delphix Engineering.
sqlcmd command line utility must be installed on the servers hosting VDBs and staging databases. The utility should be in the Delphix operating system user's PATH environment variable.	Delphix does not run sqlcmd (or any other process) directly on the source SQL Server instance. It runs sqlcmd on the staging/connector host.

9.12.3.2.6.2 Staging Windows user requirements

The Delphix Engine needs a Windows domain user — for example, `MYDOMAIN\delphix_os` — to be specified when adding Staging environments to the Delphix Engine. This user must have the following permissions:

Staging User Requirement	Explanation
Be a member of the Windows "Local Administrators" group on the Staging Host	We require this permission for mounting iSCSI LUNs presented by the Delphix Engine to the staging and target hosts. Microsoft utilities used by the Delphix Engine, such as diskpart, require membership of this group[619].
Have the "Log on as batch" privilege on the Staging host	We require this permission for remote PowerShell execution. This privilege can be assigned through the Local Security Policy (Local Policies → User Rights Assignment → Log on as batch job).

618 https://support.delphix.com/Delphix_Virtualization_Engine/MSSQL_Server/Registry_Settings_for_Optimal_Database_Performance_and_Stability_%28KBA1251%29
619 https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/diskpart

Staging User Requirement	Explanation
Be able to access existing backups from the Source Database	If accessing existing native, Red Gate, or LiteSpeed backups, the following permissions are required: • The Staging Windows User and the service account running SQL Server must both have permission to access the database backups via SMB (Windows file sharing) • The Staging Windows User and the service account running SQL Server must both have NTFS Permissions to access the database backups Separate documents describe requirements if Linking a dSource from a NetBackup SQL Server Backup (see page 2167) or Linking a dSource from a Commvault SQL Server Backup. (see page 2162)
Be able to login to each SQL Server instance that the Delphix Engine will communicate with	These requirements are described in the Staging Database Login Requirements (see page 2067)section below.

9.12.3.2.6.3 Staging database login requirements

The Delphix Engine requires a Windows Authentication login to be created for the Staging Windows User on each SQL Server instance that the Delphix Engine will communicate with, with the following permissions:

Object	Privileges Required	Delphix OS User (Windows Login)	Purpose
Server	`CONNECT SQL`	Y	Access to the SQL Server instance
Server	`sysadmin`	Y	The staging and target databases are managed and administered completely by Delphix. Our functionality requires many administrative operations on those databases and requires full access to them. Since database ownership can be changed by customers as part of configuring virtual databases, we require the sysadmin role to continue to administer the databases.

9.12.3.2.7 SQL server target hosts and databases

9.12.3.2.7.1 Target host requirements

Windows servers which will be added as Target Environments must meet the following requirements:

Target Host Requirement	Explanation
Delphix Connector software is installed and running	See Installing the Delphix Connector Service on the Target Database Servers (see page 2098) for instructions on installing the Delphix Connector.
Recommended iSCSI Registry settings must be in place	See Requirements for Windows iSCSI Configuration (see page 2083) and Knowledge Base Article KBA1251[620] for instructions on applying these settings.
The edition of the installed SQL Server instance(s) must support all database features used by linked Source databases	SQL Server may raise an error during some dSource operations if the Staging SQL Server Instance does not support features used by the Source Database. This is most easily addressed by using the same edition of SQL Server as the Source database. Features in use on the Source database can be checked using the `sys.dm_db_persisted_sku_features` dynamic view.
sqlcmd command line utility must be installed on the servers hosting VDBs and staging databases. The utility should be in the Delphix operating system user's PATH environment variable.	Delphix does not run sqlcmd (or any other process) directly on the source SQL Server instance. It runs sqlcmd on the staging/connector host.

9.12.3.2.7.2 Target Windows user requirements

The Delphix Engine needs a Windows domain user — for example, `MYDOMAIN\delphix_os` — to be specified when adding Target environments to the Delphix Engine. This user must have the following permissions:

620 https://support.delphix.com/Delphix_Virtualization_Engine/MSSQL_Server/Registry_Settings_for_Optimal_Database_Performance_and_Stability_%28KBA1251%29

Target User Requirement	Explanation
Be a member of the Windows "Local Administrators" group on the Target Host	We require this permission for mounting iSCSI LUNs presented by the Delphix Engine to the staging and target hosts. Microsoft utilities used by the Delphix Engine, such as diskpart, require membership of this group[621].
Have the "Log on as batch" privilege on the Target Host	We require this permission for remote PowerShell execution. This privilege can be assigned through the Local Security Policy (Local Policies → User Rights Assignment → Log on as batch job).
Be able to login to each SQL Server instance that the Delphix Engine will communicate with	These requirements are described in the *Target Database Login Requirements* section below.
Delphix Connector software is installed and running	See Installing the Delphix Connector Service on the Target Database Servers (see page 2098) for instructions on installing the Delphix Connector.

9.12.3.2.7.3 Target database login requirements

The Delphix Engine requires a Windows Authentication login to be created for the Target Windows User on each SQL Server instance that the Delphix Engine will communicate with, with the following permissions:

Object	Privileges Required	Delphix OS User (Windows Login)	Purpose
Server	`CONNECT SQL`	Y	Access to the SQL Server instance

621 https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/diskpart

Object	Privileges Required	Delphix OS User (Windows Login)	Purpose
Server	`sysadmin`	Y	The staging and target databases are managed and administered completely by Delphix. Our functionality requires many administrative operations on those databases and requires full access to them. Since database ownership can be changed by customers as part of configuring virtual databases, we require the sysadmin role to continue to administer the databases.

9.12.3.2.8 Supported roles for Failover Cluster instances and Always On Availability Groups

> When considering the deployment of nodes within an AG cluster, please note that the Continuous Data Engine does not support the use of nodes as standalone targets — especially for VDB deployment purposes. This is particularly significant due to potential complications with integrity of engine metadata.

Failover Cluster Instances and Always On Availability groups cannot be used as Staging Environments, and VDBs cannot be provisioned into Availability Groups.

> When adding a Failover Cluster Instance or Always On Availability Group, all nodes of the cluster must meet the requirements described in this document.

The following table shows how different SQL Server instance types should be added from the Add Environment screen:

Color	Supported?
Y	Yes
N	No

		Environment Role		
Instance Type	**Added As**	**Source Environment**	**Staging Environment**	**Target Environment**
Standalone Instance	Standalone	Y *	Y	Y
Failover Cluster Instance (FCI)	Standalone	Y *	N	N
Failover Cluster Instance (FCI)	Cluster	N	N	Y
Always On Availability Group (AG)	Cluster	Supported	N	N **

* Databases that are participating in Availability Groups will not be discovered during the discovery of a Standalone environment.

** VDBs cannot be provisioned into availability groups. However, SQL Server instances that participate in an Availability Group can also be added as Standalone Instances.

A Failover Cluster Instance added as an environment once (as either a Source or Target environment) cannot be used as both a Source and Target.

If this is required, the environment can be added twice:

- Once as a Standalone Source environment
- Once as a Cluster Target environment

The Standalone Source environment can be used for linking dSources, and the Cluster Target environment is used for provisioning VDBs.

As suggested by the Best Practice note earlier in this article, this is not a recommended configuration. Where possible, SQL Server failover cluster instances that the Delphix Engine will use as a target should not be used to host databases other than Delphix VDBs.

9.12.3.2.9 Requirements for failover cluster target environments

The following additional requirements exist for Windows Failover Cluster Instances added as Target Environments, to be used for VDB Provisioning.

- You must first add each node in the Window Failover Cluster individually as a standalone target environment, using a non-clustered address. See Adding a SQL Server Standalone Target Environment. (see page 2111)
    - The Delphix Engine may show a warning that it cannot discover the Failover Cluster Instances on each standalone host. This is expected.
- Each clustered SQL Server instance must have at least one clustered disk added to the clustered instance resource group, which can be used for creating mount points to Delphix storage.
    - The clustered drive must have a drive letter assigned to it.
    - The clustered drive must be formatted using the "GUID Partition Table (GPT)" partition style, in order for the Delphix Engine to automatically discover the drive letter as a valid option for the cluster instance. An MBR-formatted disk requires manual verification outside of Delphix that the disk has been correctly added to the MSSQL clustered resource group prior to creating the VDB. When provisioning the VDB, you must manually specify the desired MBR disk, because it will not appear in the Delphix GUI.
    - The clustered drive must be added to the clustered instance resource group as a dependency in the Failover Cluster Manager.
- Each node in the cluster must have the Failover Cluster Module for Windows PowerShell feature installed.
    - While running Windows PowerShell as an administrator, enter this command to test that the module is available: `Import-Module FailoverClusters`
- An additional target environment that can be used as a Connector Environment must exist.
    - This environment must **not** be a node in the cluster.
    - This environment should be part of the same Active Directory domain as the cluster.

9.12.3.2.10 Additional requirements for azure SQL server availability groups

Microsoft's tutorial[622] and deployment template for building a *SQL Server AlwaysOn Cluster in Azure* does not include the full range of network connectivity that is available in on-premise deployments.

In addition to the connectivity provided by this template:

- The Staging Server must be able to connect to the **Cluster IP Address** using the RPC / Remote Registry port **TCP 445**; and
- The Availability Group Listener must be configured with a TCP Port, such as the SQL Server default 1433

Connectivity to the Cluster IP Address can be tested using the following PowerShell command. This command should be run from the Staging Server, and specify the Cluster Hostname:

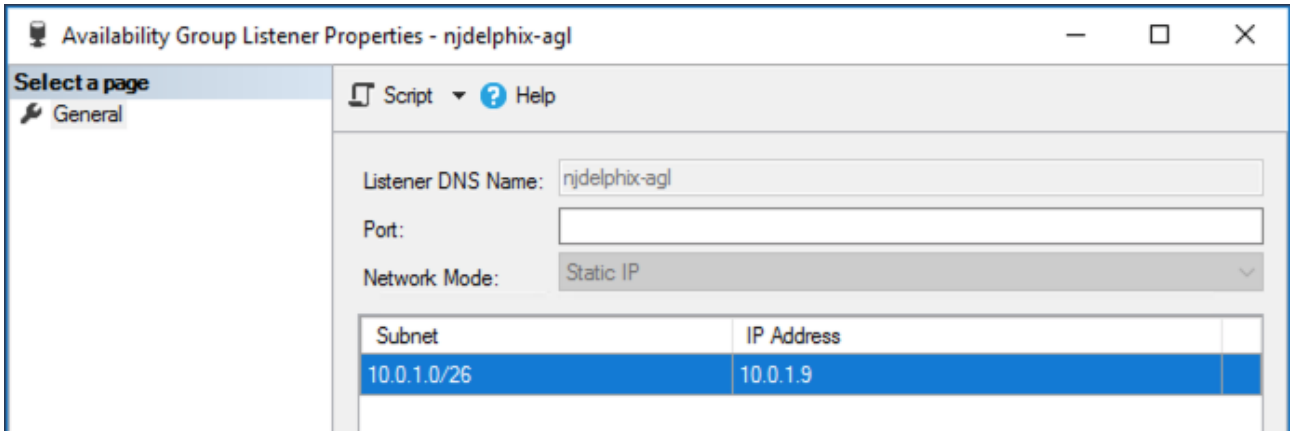
> *New-Object System.Net.Sockets.TcpClient("aodns-fc.mydomain.local", "445")*

622 https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sql/virtual-machines-windows-portal-sql-availability-group-tutorial

> For Azure clusters, Delphix does not support DNN (Distributed Network Name) at the WSFC level. However, DNN (Distributed Network Name) is supported for creating a SQL AG listener in SQL clustering.

To allow this connectivity on the marketplace template, the following additional steps may be required:

1. On a node of the SQL Server cluster, use PowerShell to define a "probe port". The active cluster node will open this port so that the Azure Load Balancer can detect it.
   Get-ClusterResource "Cluster IP Address" | Set-ClusterParameter -Name "ProbePort" -Value 59998
2. From the Failover Cluster Manager, ensure that the Core Cluster Resource has a valid IP address for its subnet (this may be configured with an invalid IP address such as 169.254.1.1).
3. On all nodes of the SQL Server cluster, ensure that the Windows Firewall allows inbound TCP traffic on TCP port 59998.
4. Extend the Azure Load Balancer (sqlLoadBalancer) configuration to route connections to the Cluster IP Address.
   a. Add a Frontend IP Address for the Cluster IP Address, with an IP address matching that configured for the cluster.
   b. Add a Health Probe for TCP port 59998.
   c. Add a Load Balancing Rule for Port 445, using the newly created Frontend IP Address and Health Probe.More complicated Azure AlwaysOn deployments, such as those including multiple networks, may require additional steps to allow this connectivity.
5. The default Availability Group Listener is not configured with a TCP Port. Current versions of Delphix require this to be configured with a valid port number, such as the default port 1433. This can be changed via the AlwaysOn object tree in SQL Server Management Studio:



9.12.3.3 Requirements for Windows iSCSI configuration

Windows iSCSI configuration requirements are split into two types. These requirements are needed on both staging and target servers.

1. iSCSI configuration required for operational stability.
2. Optional iSCSI parameters for performance improvement[623].

> When target environments are discovered, Delphix will configure the Microsoft iSCSI Initiator Service for Automatic startup.

9.12.3.3.1 iSCSI configuration required for operational stability

The following Microsoft iSCSI Initiator configuration parameters are required for the target and staging Hosts. For details about configuring registry settings, see How to modify the Windows registry[624]

> You must reboot the Windows server reboot after changing the iSCSI configuration parameters.

Registry key	Registry value	Type	Data
`HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\iSCSI\Discovery`	MaxRequestHoldTime	REG_DWORD	0x384 (900)
`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Disk`	TimeOutValue	REG_DWORD	0x384 (900)
`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4D36E97B-E325-11CE-BFC1-08002BE10318}\<Instance Number>\Parameters`	MaxRequestHoldTime	REG_DWORD	0x12C (300)

These settings will improve operational stability for VDBs and staging databases. If these settings are not adjusted, SQL Server may raise errors if VDBs are accessed during a temporary infrastructure outage. Affected VDBs may need to be manually restarted using the Continuous Data Engine.

623 https://cd.delphix.com/docs/19.0.0.0/target-host-os-and-database-configuration-options#id-(16.0.0.0)TargethostOSanddatabaseconfigurationoptions-OS-specifictuningrecommendationsforwindows
624 https://support.microsoft.com/kb/136393

Delphix Knowledge Base article KB1251[625] includes scripts to validate or set registry parameters so that they meet current Delphix recommendations.

9.12.3.3.2 Optional iSCSI parameters for performance improvement

The following iSCSI Registry setting may improve the performance of SQL Server dSource and VDB on the staging and target hosts.

Registry Key	Registry Value	Type	Data
`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\<Interface GUID>`	TcpAckFrequency	REG_DWORD	0x1 (1)

This setting is recommended for storage networks in Microsoft's TechNet article iSCSI and the Nagle algorithm[626], described in Microsoft's document TcpAckFrequency to control the TCP ACK behaviour[627]

In some environments, adjusting this setting may not improve performance compared to Windows defaults. Modifications to this registry parameter should be tested in each environment, to confirm that this provides a performance improvement.

9.12.3.3.3 Delphix engine validation for Windows iSCSI configuration

Delphix Engine validates the Windows iSCSI Configurations that are set on any supported windows staging and target host with the Delphix recommended configurations while performing the following operations:

1. Add environment operation
2. Refresh environment operation
3. Enable environment operation

Prerequisites

1. Supported if you are using Powershell 3.0 or above - If you are on Powershell version below 3.0, then the job will be updated with a warning that the Powershell version on your host is not supported for validating iSCSI parameters.
2. The below alerts are applicable only for staging or target Windows hosts.

625 https://support.delphix.com/Continuous_Data_Engine_(formerly_Virtualization_Engine)/MSSQL_Server/Registry_Settings_for_Optimal_Database_Performance_and_Stability_(KBA1251)
626 https://social.technet.microsoft.com/wiki/contents/articles/7636.iscsi-and-the-nagle-algorithm.aspx
627 https://docs.microsoft.com/en-us/troubleshoot/windows-server/networking/registry-entry-control-tcp-acknowledgment-behavior

9.12.3.3.3.1 Additional information

1. Delphix Engine will only validate and will not alter any configuration in the user environment.
2. On update of registry values on the target host to match Delphix recommendations, the faults from the Delphix engine will only be resolved if any of the operations (environment add, refresh or enable) is performed. Delphix engine will not monitor the state of the target host in the background and hence any change will not be picked up unless an operation is triggered. So, the user needs to take action for the change to reflect in faults.
3. On a successful Delphix Engine upgrade, the latest default iSCSI recommendations will be used for validations.

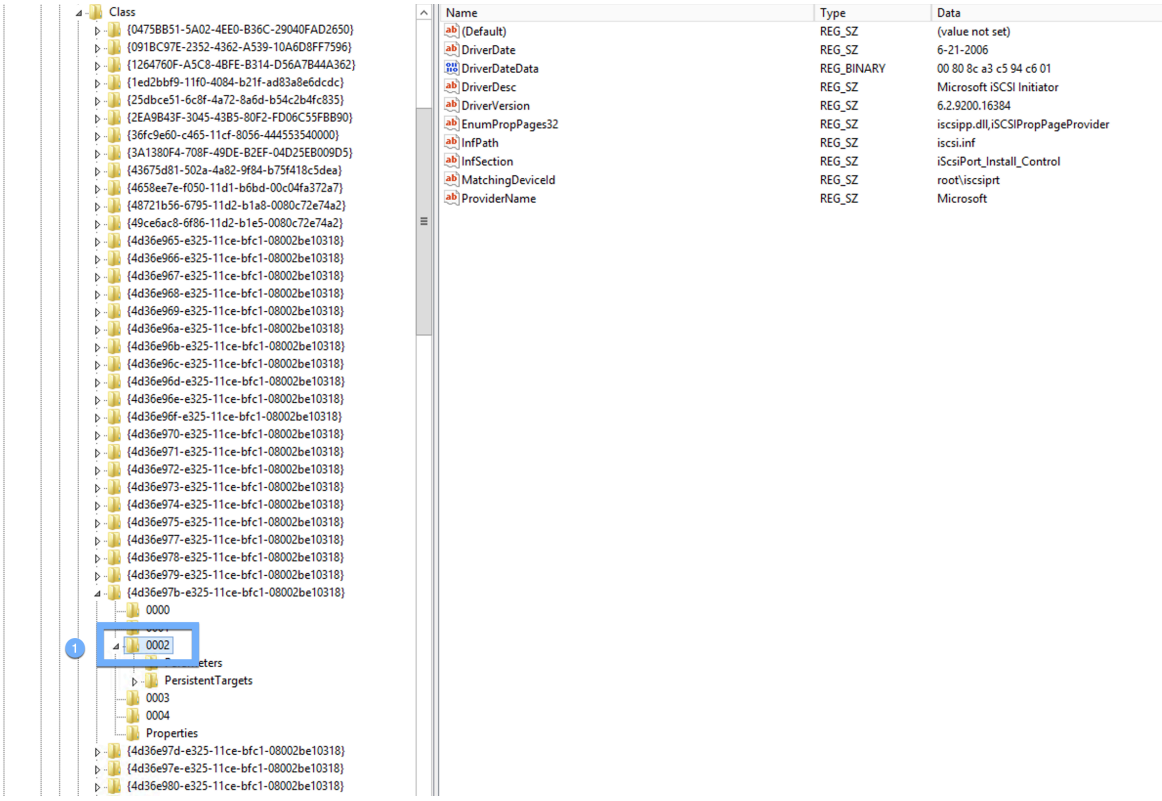
9.12.3.3.3.2 Troubleshooting

Severity	Type	Description
Warning	iSCSI configuration parameter values on environment <env-name> do not match Delphix Engine required values.	This single fault and/or job warning is thrown for all mismatched parameters.
Warning	Failed to fetch some iSCSI configuration parameters for the environment <env-name>.	This single fault and/or job warning is thrown for all parameters where Delphix Engine fails to fetch the value at the target host.
Warning	Failed to fetch some iSCSI configuration parameters for host <env-name> due to time out.	This job warning is raised, if Delphix Engine is unable to get the iSCSI parameters on the host within 5 minutes. No fault thrown at this point.
Warning	PowerShell version 3 (or above) is required for fetching iSCSI configuration parameters.	This job warning is raised if the PowerShell version is below 3 on the host during the validation of the iSCSI parameters. No fault thrown at this point. The validation is skipped.

9.12.3.3.4 Identifying the instance number for iSCSI control class initiator drivers

1. From the Windows toolbar, click **Start** and select **Run** from the menu.
2. Type `regedit` in the **Open** field and click **OK.**

3. Go to the following registry key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4D36E97B-E325-11CE-BFC1-08002BE10318}\<Instance Number> where the value of <Instance Number> is the one that shows a **DriverDesc** value of **Microsoft iSCSI Initiator**. Under the registry key, locate and expand the plus (+) sign next to the instance number. In the example below, the value of the instance number is 0002.



1 Value of Instance Number

9.12.3.3.5 Windows iSCSI configuration and limits for target and staging hosts

> **Windows limitation**
>
> Windows supports up to 255 iSCSI LUNs maximum. This creates a hard limit on the number of VDBs that can be created because each VDB requires one or more iSCSI connections.

9.12.3.3.5.1 iSCSI connections: staging

1. dSource linked with Logsync disabled = 1 LUN (DATA).

2. dSource linked with Logsync enabled = 2 LUNs (DATA and ARCHIVE).
3. dSource linked with Logsync disabled and SnapShot started (new COPY ONLY FULL BACKUP) = 2 LUNs (DATA and TEMP).
    a. Once the SnapShot is completed the TEMP LUN will be destroyed and 1 LUN used.
4. dSource linked with Logsync enabled and SnapShot started (new COPY ONLY FULL BACKUP) = 3 LUNs (DATA, ARCHIVE and TEMP).
    a. Once the SnapShot is completed the TEMP LUN will be destroyed and 2 LUNs used.
5. Microsoft's physical limitation of 255 iSCSI LUNs could be breached by deploying more than 120 dSources to a single staging target (leaving room for other non-Delphix LUNs). Despite physically supporting this many iSCSI LUNs, Delphix does not recommend deploying so many staging databases to a single target, due to other performance considerations.
    a. The proposed scenario would leave 13 additional iSCSI connections available for COPY ONLY FULL BACKUPS.
    b. For dedicated staging hosts, a PowerShell process for monitoring is not used.

9.12.3.3.5.2 iSCSI connections: targets

1. VDB normal operations = 1 LUN (DATA).
    a. No extra mounts required for SnapShot restore or refresh from source Snapsync.
2. VDB point-in-time log actions (such as restore, refresh or provision from logs) = 2 LUNS (DATA and SOURCE_ARCHIVE).
    a. An extra LUN is not required for Snapsync operations, only Logsync.
    b. Most users do not require enablement of the Logsync feature for MSSQL Sources, because sources in FULL RECOVERY mode create a Snapsync for each log file, providing a significant number of restore points even without retaining the logs.
    c. Once recovery is completed the SOURCE_ARCHIVE LUN will be destroyed and 1 LUN used.
3. A maximum of ~120 VDB's per Target is recommended.
    a. In 4.x, target host iSCSI connections are less likely to be a limitation while processing charges for PowerShell threads may become prohibitive, because each target VDB requires a PowerShell process for monitoring.
    b. In 5.x, this has been alleviated with a hard limit on PowerShell processes.

9.12.3.3.5.3 iSCSI connections: V2P

1. V2P normal operation = 1 LUN (DATA).
    a. Once the V2P operation is completed, the DATA LUN will be destroyed leaving no LUNs used.
2. V2P point in time log actions (provision from logs) = 2 LUNS (DATA and SOURCE_ARCHIVE).
    a. Once the V2P operation is completed, both the DATA and SOURCE_ARCHIVE LUNs will be destroyed leaving no LUNs used.

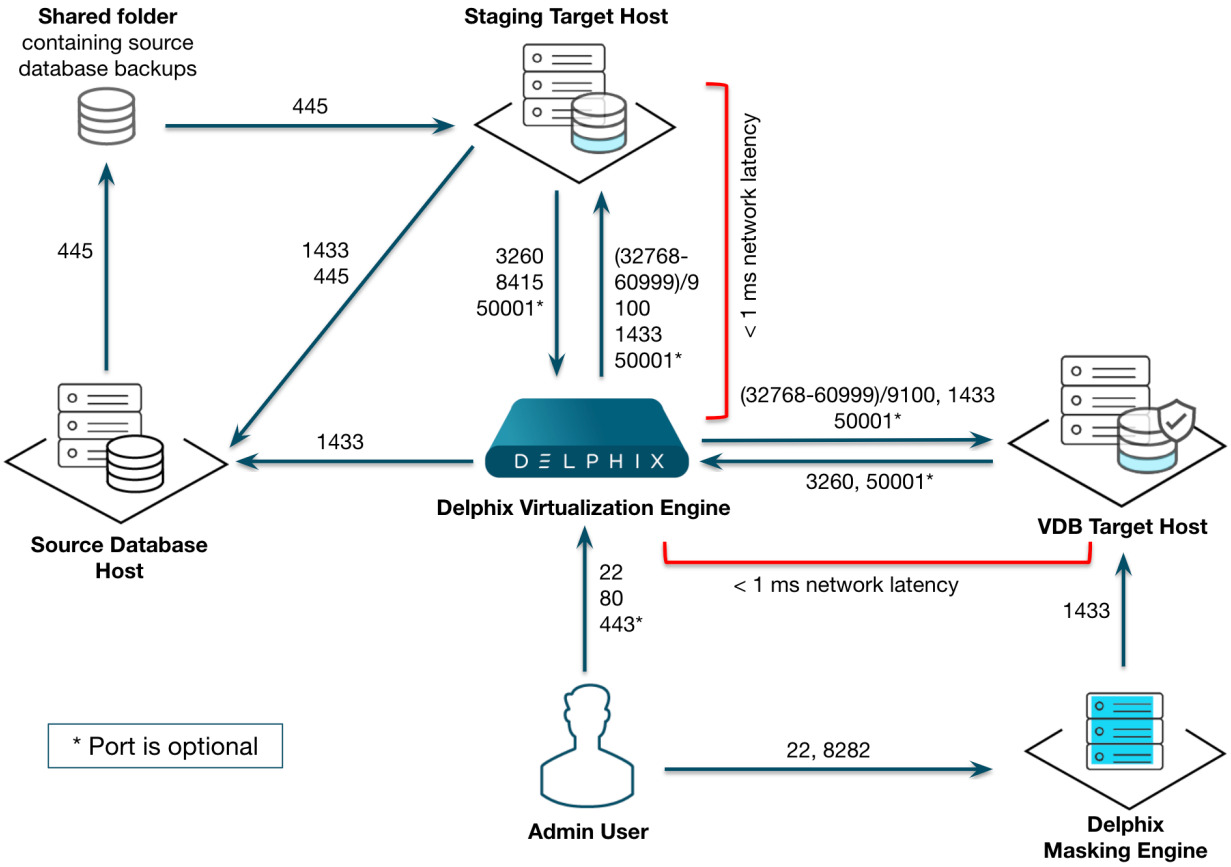
9.12.3.4 Network access requirements for SQL Server

9.12.3.4.1 Network architecture

The diagram *Delphix Virtualizing and Masking SQL Server Network Architecture* below depicts the overall network architecture for Delphix virtualizing and masking SQL Server. In the diagram, each of the arrows represents the direction of a network connection between two nodes. Next to each arrow is a label indicating the network protocol (TCP) and the port number indicating the network service. Also, indicated in red are the recommended network latencies between the major components of the architecture.

> You can optionally configure a separate Connector Environment, specifically used to discover databases on the source during Environment creation. You can also use your Staging Target Host to be used as the Connector Environment, as seen in the image below.

Delphix virtualizing and masking SQL Server network architecture



Delphix Virtualizing and Masking SQL Server Network Architecture

9.12.3.4.2 Ports

Based on the table below, the Windows Network Administrator needs to complete a series of tasks. For each port listed, determine whether it must be opened in your firewall between your Delphix Engine and source or target systems. Work with your Delphix Administrator to understand what requirements are there, and ensure that they have been met before proceeding.

Port	Network Service	Required for virtualization?	Required for masking?	Description and usage
22	SSH	Yes	Yes	Used for accessing command-line interface (CLI) and internal Delphix OS accounts
80	HTTP	Yes	No	Used for GUI console access on Delphix Engine by default, disabled when HTTPS in use
443	HTTPS	Yes	No	Used for GUI console access on Delphix Engine, disabled when HTTP in use
445	SMB	Yes	No	Used for attaching shared folders on Windows. To take a copy-only backup or use Delphix Managed backups, this port is required to allow the source environment access to the staging environment.
1433	JDBC	Yes	Yes	Used for accessing SQL Server databases for queries on data-dictionary. This port is the default, but you can use other ports instead. **Note:** Port 1433 is required between the Delphix Engine and AG (Availability Group) cluster sources.
3260	iSCSI	Yes	No	Used for network-attached storage (NAS) on Windows database servers

Port	Network Service	Required for virtualization?	Required for masking?	Description and usage
8415	DSP	Yes	No	Used for SQL Server hooks, when enabled. This port needs to be open between the Delphix Engine and the VDB target.
(32768-60999)/9100	Delphix Windows Connector	Yes	No	Used for connecting to the Delphix Connector service installed on Windows target database servers.
50001	iPERF	No	No	Used for network throughput testing with the open-source iPerf package through the Delphix CLI, this is purely optional (but useful) functionality

9.12.3.4.3 AppData port requirements

For unstructured files and AppData port requirements, please visit the Network and Connectivity Requirements for Windows Environment (see page 2244) article.

9.12.3.4.4 Applying network access requirements to Windows cluster configurations

Follow the below points to apply Network Access Requirements to Windows Cluster configurations

- For Target Windows Cluster Environments (running Target Failover Cluster Instances), you need to access TCP 445 of the Windows Cluster Virtual IP from the Staging Server.
- For Source Windows Cluster Environments (running Source Always-On Availability Group databases), you need to access the following:
    - From the Staging Server:
        - TCP 445 of the Windows Cluster Virtual IP
        - The SQL Server port (default TCP 1433) of each SQL Server instance running on the cluster
        - The SQL Server port (default TCP 1433) of all Availability Group Listener Virtual IP addresses that contain Source Databases
    - From the Delphix Engine:
        - The SQL Server port (default TCP 1433) of each SQL Server instance running on the cluster

    - The SQL Server port (default TCP 1433) of all Availability Group Listener Virtual IP addresses that contain Source Databases
- For Source Windows Cluster Environments on Azure, some Azure configuration changes may be required, see Additional requirements for Azure SQL Server Availability Groups (see page 0) for more details.
- For both Source and Target Windows Cluster Environments, the required connectivity for a standalone host should be configured for every node of the cluster. For example, Delphix Engine should be able to connect to the Delphix Connector service installed on each target windows cluster node listening on the Delphix Connector port configured at installation.

9.12.4 Managing SQL Server environments and hosts

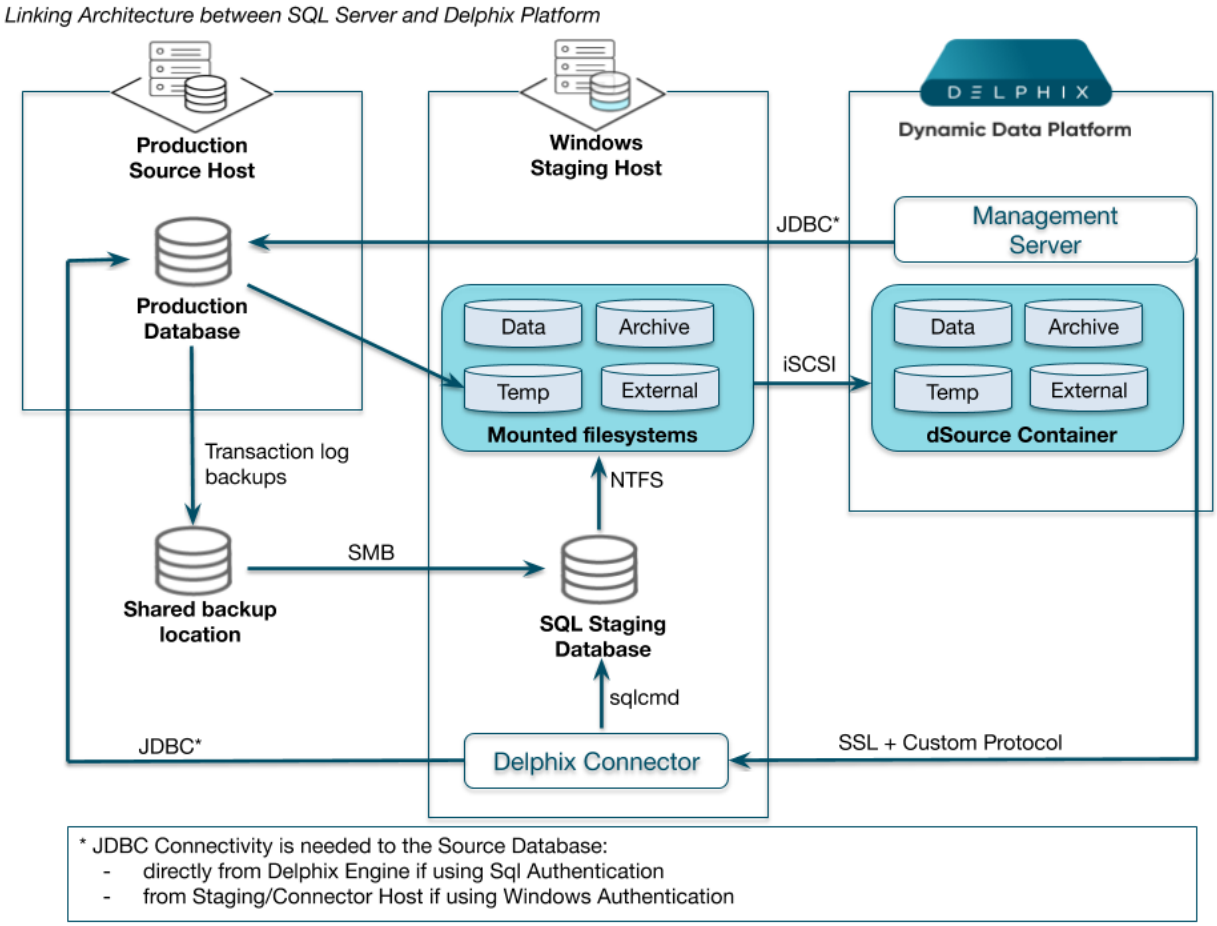
This section describes the attributes of SQL Server environments such as information for the Delphix Connector and covers the following topics:

- Overview of setting Up SQL server environments (see page 2092)
- Environment attributes for hosts with SQL server (see page 2096)
- Installing the Delphix connector service on the target database servers (see page 2102)
- Upgrading the Delphix connector (see page 2105)
- Manual discovery for SQL server instances (see page 2108)
- Adding a SQL server source environment (see page 2110)
- Adding a SQL server standalone target environment (see page 2111)
- Adding a SQL server failover cluster target environment (see page 2113)
- Additional SQL server environment topics (see page 2117)

9.12.4.1 Overview of setting up SQL Server environments

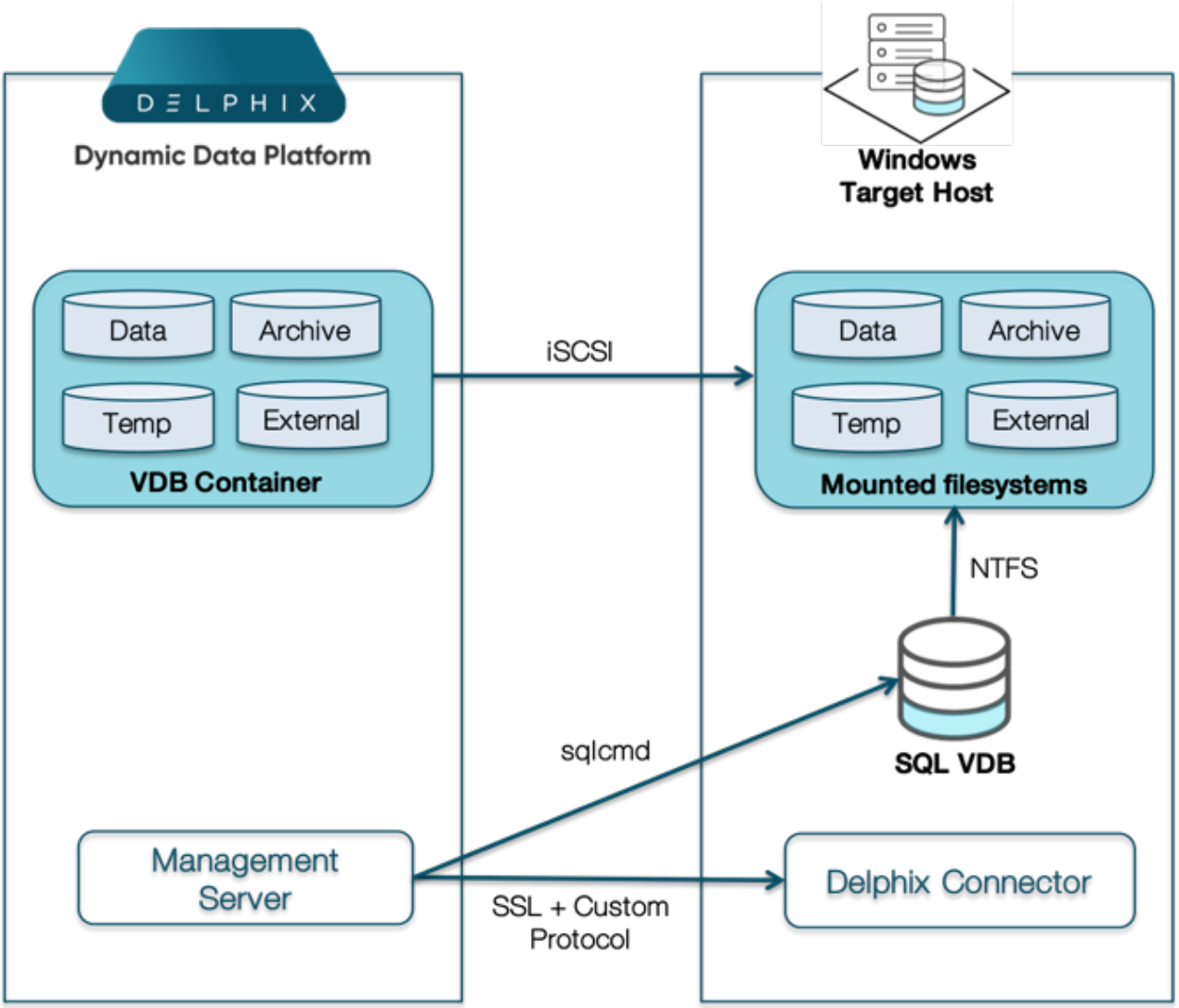
This topic describes the high-level process for adding SQL Server environments, linking SQL Server databases to the Delphix Engine, and provisioning virtual databases.

9.12.4.1.1 Block diagram of linking architecture between SQL server environments and the Delphix engine



This block diagram shows the host architecture when creating SQL Server Environments in the Delphix Engine. On the left, the Source Host (in this example a Production Database Server) creates backups to a shared backup location. The Delphix Engine (far right), continuously monitors the source database to determine when new backups are available. When a backup is available, the Delphix Engine will contact the Windows Staging host (center) via the Delphix Connector service. The Staging host will read the backups from the shared backup location and recover them through a Staging Database that is automatically set up on Delphix iSCSI storage. Once recovery is complete, the backup data is incorporated into the Delphix dSource as a new snapshot card, and is available for use.

9.12.4.1.2 Block diagram of SQL server provisioning architecture



This block diagram shows how Delphix provisions VDBs to a Target Environment. The Delphix Engine (left) creates a set of virtual files from a snapshot that becomes the VDB container. These files are presented to the Target host (right) via iSCSI. Delphix calls the Delphix Connector on the Target and initiates the creation of a new database using the Virtual files. Once complete, the Virtual Database is brought online and made available for use.

9.12.4.1.3 The Delphix connector and environment setup

The Delphix Connector is a Windows service that enables the communication between the Delphix Engine and the Windows target environment where it is installed.

- The directory on which you install the Delphix Connector should have at least 1GB of available space.

This target machine can serve three purposes in a Delphix Engine deployment. It can:

1. Serve as a proxy for database discovery on source hosts.
2. Host a staging database for a linked dSource and run the validated sync process.
3. Host a target environment for provisioning virtual databases (VDBs).

Database discovery is initiated during the environment set up process. When you specify a production source environment that contains the databases you want to manage with the Delphix Engine, you must also specify a target environment where you have installed the Delphix Connector to act as a proxy for communication with the source environment. This is necessary because Delphix does not require that you install the Delphix Connector software on the production source environment. When you register the source environment with the Delphix Engine, the Delphix Engine uses the Delphix Connector on the proxy environment to discover SQL Server instances and databases on the source. You can then create dSources from the discovered databases. If you later refresh the source environment, the Delphix Engine will execute instance and database re-discovery through the proxy host.

SQL Server dSources are backed by a staging database that runs on a target host. There is no requirement for additional local storage on this target host, as the storage is mounted over iSCSI from the Delphix Engine. At Delphix, we refer to the creation and maintenance of this staging database on the staging host as "validated sync," because it prepares the dSource data on the Delphix Engine for provisioning VDBs later on. After the Delphix Engine creates the staging database, it continuously monitors the source database for new transaction log backups. When it detects a new transaction log backup, it restores that backup to the staging database. The result is a TimeFlow with consistent points from which you can provision a VDB, and a faster provisioning process, because there is no need for any database recovery during provisioning.

When you later provision a VDB, you can specify any environment as a target, including the environment that contains the staging database. However, for best performance, we recommend that you choose a different target environment. The only requirements for the target are:

- It must have the Delphix Connector installed.
- It must have an operating system that is compatible with the one running on the validated host, as described in SQL Server Support and Requirements. (see page 2052)

9.12.4.1.4 Toolkit size and predicted growth

Each of the clients that run from the client-side toolkit generates its own logs. Each client generates 4 different log files, one for each level of logging: info, trace, debug, and error. Each level of logging is restricted to a maximum of 10 logfiles, and these logfiles are capped at 10MB each. Therefore, the Delphix Engine will consume a maximum of 400MB per client-side application. On the source server, there are currently two commonly run client-side applications, SnapSyncClient and the Delphix Connector.

> V2P also generates its own logs, so if you intend to V2P to the source, you should account for an additional 400MB in your upper bound.

Thus, the max amount of growth for the toolkit from logging is 800MB without V2P (or 1.2GB with V2P).

Linking additional dSources does not impact the size of the toolkit on production, aside from the log messages generated during linking, which is included in the calculation above.

On the target server, unlike the source server, there would be only one client – the Delphix Connector, which would occupy around 400 MB maximum storage space. In addition, the Delphix Engine pushes new scripts each time you provision a VDB. This requires < 1MB space.

Therefore, the maximum space occupied by the toolkit directory on the source server is its initial size (~ 400MB) + 800MB = 1.2 GB. While on the target server, the maximum toolkit size is initial size (~ 400MB) + 400 MB + Number of VDBs * 1MB.

9.12.4.2 Environment attributes for hosts with SQL Server

This topic describes the attributes of SQL Server environments such as information for the Delphix Connector. Below you will see a section for common environment attributes shared by all types of environments as well as SQL Server-specific ones.

9.12.4.2.1 Procedure

1. Login to the Delphix Management application.
2. Click Manage.
3. Select Environments.
4. In the Environments panel, click the name of an environment to view its attributes.
5. Next to Attributes, click the Pencil icon to edit an attribute.

9.12.4.2.2 Common environment attributes

Attribute	Description
Environment Users	The users for that environment. These are the users who have permission to ssh into an environment or access the environment through the Delphix Connector. For more information on the environment user requirements, see the Requirements topics for specific data platforms.
Host Address	The IP address of the environment host.
DSP KeyStore Path	The path to the user-managed DSP Keystore.
DSP KeyStore Alias	The lowercase alias to use inside the user-managed DSP Keystore.
DSP KeyStore Password	The password for the user-managed DSP Keystore.
DSP TrustStore Path	The path to the user managed DSP truststore.
DSP TrustStore Password	The password for the user managed DSP truststore.

Attribute	Description
OS	The name of the host operating system.
Version	The version of the host operating system.
Release	The release of the host operating system.
Time Zone	The timezone of the host operating system.
Total RAM	The amount of RAM on the host machine.
Processor Type	The processor type of the host machine.
Traceroute	Traceroute info from target host to Delphix Engine.
Notes	Any other information you want to add about the environment.
Java Development Kit	The currently selected JDK kit will be shown.
Java Development Kit (JDK) Path	Location of the Java Development Kit (JDK) used for the host. Only specified if the feature to provide your own JDK is enabled, otherwise, the defaults are used per our Java Support Matrix. (see page 1025)

9.12.4.2.3 SQL server environment attributes

Attribute	Description
Installed Powershell Version	The PowerShell version installed on the windows target host. This is not applicable for Windows source environments.
Delphix Connector Path	The path for the toolkit that resides on the host.
Delphix Connector Port	The port that the connector connects on.
Delphix Connector Version	The Windows Connector version that is installed on the provided host.

Attribute	Description
.NET Framework Version	The .NET Framework version used for Windows Connector Service.
Delphix Connector Host	The connector host for a windows environment. This is not applicable for Windows target environments.

9.12.4.3 Installing the Delphix connector service on the target database servers

This section lists the steps involved in installing a Delphix Connector on your target database server. Installing the Delphix Connector is vital for communication between the Delphix Engine and the targets. A minimum available space of 1GB is a prerequisite to installing the Delphix Connector.

1. From the machine that you want to use as a target, start a browser session and connect to the **Delphix Management** application**.**
2. Click **Manage**.
3. Select **Environments**.
4. Next to **Environments**, click the **Plus** icon.
5. In the **Add Environment** dialog, select **Windows** from the **operating system** menu.
6. Select **Target**.
7. Select **Standalone**.
8. Click **Next**.
9. Click the download link for the **Delphix Connector Installer**.
   The Delphix Connector will download to your local machine.

   Note: You can also download the Delphix Connector directly by navigating to this URL: `http://<name of your Delphix Engine>/connector/DelphixConnectorInstaller.exe`
10. On the Windows machine that you want to want to use as a target, run the Delphix Connector installer. Click **Next** to advance through each of the installation wizard screens.
    Note:

    The installer will only run on 64-bit Windows systems. 32-bit systems are not supported.
    a. For **Connector Configuration**, make sure there is no firewall in your environment blocking traffic to the port on the target environment that the Delphix Connector service will listen to.
    b. For **Java Configuration**, to provide your own Oracle Java enter the absolute path to your Oracle JDK and click **Next**. Otherwise, leave the field blank.
       Note:
       The NETWORK SERVICE user requires read and execute permissions on the Oracle JDK, its subfolders, and files.

c. For **Select Installation Folder**, either accept the default folder or click **Browse** to select another.
d. For **Choose .NET Framework (Delphix Engine 6.0.5.0 onwards),** select which version of .NET should be used by the Delphix Connector Windows service. The installer will default to the latest version detected on the system. This selection can be changed afterwards but make sure that chosen .NET framework version should exist in the system.
e. Click **Next** on the installer final **Confirm Installation** dialog to complete the installation process.
f. For successful installation a popup will come up stating "**DelphixConnector Installed Successfully**".
g. Click **Close** to exit the Delphix Connector Install Program.

9.12.4.3.1 Relocating the Delphix connector

There are times when the Delphix Connector installation requires a move to a different directory or drive. It's not a trivial relocation. This requires disabling dSources and/or VDBs, uninstalling the current install and reinstalling to the new location. In addition to this, an upgrade to the Delphix Connector can also be achieved via the uninstall/reinstall methodology, including a change in location. These instructions cover 4.0 through 5.1. The steps are the same up to 5.1.2.0. At that version and forward it is no longer required to use the CLI to change the Delphix Connector location. All one needs to do is refresh the Delphix Connector environment and the new directory location is discovered and updated on the Delphix Engine. In the steps listed below, the example is moving the connector from "C:\Program Files" to "C:\", so the full connector path is "C:\Delphix\DelphixConnector".

1. **Disable** the **dSources** and/or **VDBs** associated to the **DelphixConnector** host. This will unmount the storage from the Windows host, removing the directories, the **dSources** represented by the staging databases, and the **VDBs**.
   a. **dSource** staging directories are in the form of "guid-staging-xx", where xx is the staging database number
   b. **VDB** directories are appended with "guid-vdb-xx".
2. When all **VDBs** and **dSources** are disabled and you are ready to move the **DelphixConnector** location, stop the **DelphixConnector service**.
3. Backup the remaining directories as a precaution, in particular, the **logs** directory.
4. Uninstall the **DelphixConnector**, using the instructions from Uninstall the Delphix Connector (see page 2105).
5. Reinstall the **DelphixConnector** to the new location, such as "C:\Delphix\DelphixConnector". Check that the **DelphixConnector** service has started.
6. Modify the new **DelphixConnector**location.
   a. On Delphix engine versions **prior to 5.1.2.0**, use the CLI to modify the directory

      i.
      ```
      de4350.dcenter host> select winhost.delphix.com
      de4350.dcenter host 'winhost.delphix.com'> update
      de4350.dcenter host 'winhost.delphix.com' update *> set
      toolkitPath="C:\Delphix\DelphixConnector"
      ```

```
de4350.dcenter host 'winhost.delphix.com' update *> commit
    Dispatched job JOB-3203
    HOST_UPDATE job started for "winhost.delphix.com".
    HOST_UPDATE job for "winhost.delphix.com" completed
successfully.
```

ii. Refresh the windows environment for the change to take effect.

b. On engine versions **5.1.2.0 and higher**, you only need to refresh the windows environment.

7. Enable the **dSources** and/or **VDBs**.

9.12.4.3.2 Replacing self-signed certificates on the Delphix connector

The Delphix Connector relies on a Java Keystore with a self-signed X.509 certificate in order to instantiate SSL. If this certificate does not conform to the customer's business standards, it is possible to run a PowerShell script (ReplaceConnectorKeystore.ps1) to replace the self-signed certificate with a certificate that is signed by a Certificate Authority of their choice (i.e. Verisign).

This script should only be used to replace the self-signed certificate in the Delphix Connector's Java Keystore with a signed certificate. Upon execution, the script will do the following:

1. Validate that a PrivateKeyEntry exists within the input keystore
2. Stop the DelphixConnector service
3. Rename the existing DelphixConnector keystore
4. Import the new keystore
5. Start the DelphixConnector service

9.12.4.3.2.1 Prerequisites:

1. The Delphix Connector is installed
2. The DelphixConnector.jks file exists at <Drive>:\<path to DelphixConnector>\connector\DelphixConnector.jks
3. The DelphixConnector.properties file exists at <Drive>:\<path to DelphixConnector>\connector\DelphixConnector.properties and has not been tampered with (STOREPASS, KEYPASS, UUID are present)
4. The Java Keytool utility exists at <Drive>:\ <path to DelphixConnector>\jre\bin\keytool.exe
5. The script, ReplaceConnectorKeystore.ps1 exists at <Drive>:\<path to DelphixConnector>\connector\ReplaceConnectorKeystore.ps1

9.12.4.3.2.2 User inputs:

1. A JKS/PKCS#12 formatted keystore containing a PrivateKeyEntry with a signed certificate
2. The alias of the PrivateKeyEntry in the new keystore

3. The password for the new JKS/PKCS#12 keystore
4. The password for the private key in the new JKS/PKCS#12 keystore

9.12.4.3.2.3 Running the script:

Open up a PowerShell console, and do the following:

1. Navigate to where ReplaceConnectorKeystore.ps1 lives
2. Run .\ReplaceConnectorKeystore.ps1
3. Enter the full path to the new JKS/PKCS#12 keystore
4. Enter the alias of the PrivateKeyEntry in the input keystore
5. Enter the password for the input keystore
6. Enter the password for the private key in the input keystore

How to check if your Java keystore contains a privateKeyEntry:

```
PS C:\Program Files\Delphix\DelphixConnector\jre\bin> .\keytool.exe -list
-keystore ..\..\connector\DelphixConnector.jks
-storepass <STOREPASS from DelphixConnector.properties file>
Keystore type: JKS
Keystore provider: SUN
Your keystore contains 1 entry
delphixconnector-4ef488a8-85df-4418-b56d-1e61b25c0aa2, Jul 28, 2017, PrivateKeyEntry,
Certificate fingerprint (SHA1): 67:79:DA:E2:64:7A:74:42:62:CA:13:66:29:16:81:0A:B9:7E
:4A:60
```

Example of a successful keystore replacement:

```
PS C:\Users\dtully\Documents> .\ReplaceConnectorKeyStore.ps1
Enter the full path to a JKS/PKCS#12 keystore: C:\Program
Files\Delphix\DelphixConnector\jre\bin\test.jks
Enter alias: leaf
Enter keystore password: ********
Enter private key password: ********
Verifying that a PrivateKeyEntry exists in C:\Program
Files\Delphix\DelphixConnector\jre\bin\test.jks
Stopping the Delphix Connector service
Renaming C:\Program Files\Delphix\DelphixConnector\connector\DelphixConnector.jks to
C:\Program Files\Delphix\DelphixConnector\connector\DelphixConnector.jks.old
Importing the keystore into DelphixConnector.jks
[Storing C:\Program Files\Delphix\DelphixConnector\connector\DelphixConnector.jks]
Starting the Delphix Connector service
```

9.12.4.3.3 Uninstalling the Delphix connector service from the target database servers

On occasion, the installed Delphix Connector may become out of date as newer versions of Delphix are released to improve stability or performance.

The following steps can be used to uninstall and reinstall the Delphix Connector as required:

- Uninstall the Delphix Connector
- Download the latest installer from your Delphix Engine
- Reinstall the Delphix Connector

While the Delphix Connector is uninstalled on target/staging environments, dSource and virtual database (VDB) operations (such as validated sync, manual snapshots, and provisioning operations) initiated by the Delphix Engine against those environments will fail.

When working with target/staging environments, or Delphix Engines with a large number of users, to reduce the risk of confusion or failure for other users consider disabling each dSource in the environment. In the case of VDBs, there is no need to disable or shutdown. If SnapSync, refresh or other VDB operations occur during the uninstall/install they can be restarted once the procedure is complete. This prevents the need to incur more downtime.

9.12.4.3.3.1 Uninstalling the Delphix connector

Before uninstalling the Delphix Connector, the following steps should be performed using the Delphix Management application:

1. Login to the Delphix Management application as an admin.

> As a precaution, you can backup the binary directories, such as bin, connector, service, scripts, etc, to a temporary location.

2. Copy the logs directory, since uninstall will remove this directory. VDB or Staging folders can not be moved since the data is an iSCSI data mount. These will not be removed or manipulated during uninstall/reinstall.
3. Verify that no VDB provisioning or refresh operations are currently being performed on the target/staging environments.
4. If the environment being upgraded is a staging environment for any dSources, disable all dSources using this environment.
5. Just prior to disabling dSources, use the following query on each SQL Server instance to verify that no operations are currently running.

Show Restore Operations

```
SELECT r.session_id, r.command, CONVERT(NUMERIC(6,2), r.percent_complete) AS
[PercentComplete], CONVERT(VARCHAR(20), DATEADD(ms,r.estimated_completion_time,
GetDate()),20) AS [ETACompletionTime], CONVERT(NUMERIC(10,2), r.total_elapsed_time/
1000.0/60.0) AS [ElapsedMin], CONVERT(NUMERIC(10,2), r.estimated_completion_time/
1000.0/60.0) AS [ETAMin], CONVERT(NUMERIC(10,2), r.estimated_completion_time/1000.0/
60.0/60.0) AS [ETAHours], CONVERT(VARCHAR(1000), (SELECT SUBSTRING(text,
r.statement_start_offset/2, CASE WHEN r.statement_end_offset = -1 THEN 1000 ELSE
(r.statement_end_offset-r.statement_start_offset)/2 END) FROM
sys.dm_exec_sql_text(sql_handle)))
FROM sys.dm_exec_requests r
WHERE command IN ('RESTOREDATABASE','BACKUPDATABASE','RESTORELOG','BACKUPLOG')
```

If your Windows Server is used as a Staging / Validated Sync Target for one or more linked dSources, uninstalling the connector while a Validated Sync operation (i.e. database restore) is in progress may leave the database in an inconsistent state. Validated Sync operations are not visible in the Delphix GUI, so it is necessary to query the SQL Server instance(s) on this server directly to check whether a restore is in progress.

6. If any operations are in progress, wait for them to complete before disabling dSources. Once complete, disable dSources and proceed with the uninstall.

On the Target/Staging Environment (where the Delphix Connector is being upgraded):

1. From the Windows Services manager, Stop the **Delphix Connector Service**.
2. Via the **Task Manager** verify that the java process associated with the **Delphix Connector** has terminated.
3. From **Control Panel** select **Programs and Features**, then navigate to **Delphix Connector**, right-click it and select **uninstall**.
4. Install the new connector from the downloaded **DelphixConnectorInstaller.msi** or **DelphixConnectorInstaller.exe** file. For more information refer to Installing the Delphix Connector Service on the Target Database Servers. (see page 2098)
5. Check the **Delphix Connector** version from the **Programs and Features** list. For more information on Connector versions please refer to How To Determine Connector Version[628].
6. Check the **Delphix Connector** service is running and set to automatic.
7. Verify your system is now working properly. You may need to manually start any disabled dSources or VDBs from the Delphix UI. For more information refer to Managing Data Sources and Syncing Data.[629]

628 https://support.delphix.com/Delphix_Virtualization_Engine/MSSQL_Server/Delphix_Windows_Connector_Version_History%2C_and_How_To_Determine_Connector_Version_(KBA4376)
629 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/9962401/Managing+data+sources+and+syncing+data

9.12.4.3.4 Troubleshooting Delphix connector

9.12.4.3.4.1 Troubleshooting removing the Delphix connector

If there are issues removing the original Delphix Connector please refer to these steps:

- You can remove the DelphixConnector from Control Panel > Program and Features.

9.12.4.3.4.2 Troubleshooting Delphix connector install

If you experience the following issues when re-installing the Delphix Connector refer to, Reinstall/Upgrade the Delphix Connector[630].

- Delphix Connector service might disappear from the service console.
- Installation failing with unidentified service error.
- The Delphix connector upgrade failed.
- Manually deleted the Delphix Connector folder to try reinstalling.

9.12.4.3.4.3 Troubleshooting if the Delphix connector service fails to start

- Check if the Delphix Connector service is available and running on the Staging Host. If it is not running, then start the service.
- If the Delphix Connector service automatically shuts down or has restarting problems, kill the running java process started by Delphix connector and then retry starting the Delphix Connector Service.
- Open a command prompt (cmd) as administrator.
- From the prompt run: netstat -noba | findstr "9100 java". If the connector is still running, the output should be similar to this:

  ```
  C:\Users\username>netstat -noba | findstr "9100 java"
   [java.exe]
    TCP    0.0.0.0:9100           0.0.0.0:0              LISTENING       1928
   [java.exe]
    TCP    [::]:9100              [::]:0                 LISTENING       1928
   [java.exe]
  ```

  The final number on each line is the Process ID (PID) of the Delphix Connector process.
- If there is a java process bound to port 9100 as shown above, but the Delphix Connector still fails to start, you can kill the Java process using the taskkill command:

  ```
  C:\Users\username>taskkill /PID 1928 /F
  SUCCESS: The process with PID 1928 has been terminated.
  ```

630 https://support.delphix.com/Delphix_Virtualization_Engine/MSSQL_Server/Reinstall%2F%2FUpgrade_the_Delphix_Connector

Alternatively, you can use Task Manager or Process Explorer to identify and end the correct process/PID and end it from there.
- Start the Delphix Connector service.

If the service still fails to start after performing the above steps, please attempt to reinstall the Delphix Connector. If issues persist, contact Delphix Support for assistance.

9.12.4.3.4.4 Troubleshooting if installation of third party software results in Delphix connect service to fail to start

- If third party software is installed on the Windows host running Delphix Connector and upon the reboot, Delphix Connector doesn't startup, please uninstall the Software and revert the host back to the state it was in before the software deployment and then check if the Delphix Connector service starts
- If it doesn't start check the steps in the section "The Delphix Connector service fails to start".
- If the service still doesn't start please open a Support case with Delphix for further investigation.

9.12.4.4 Upgrading the Delphix connector

The installed Delphix Connector may become outdated as newer versions of Delphix are released to improve stability or performance.

Upgrading the Delphix connector is a two-step process, which includes

- Uninstalling the Delphix connector (see page 2102)
- Installing the Delphix connector (see page 2098)

While the Delphix Connector is uninstalled on target/staging environments, dSource and virtual database (VDB) operations (such as validated sync, manual snapshots, and provisioning operations) initiated by the Delphix Engine against those environments will fail.

When working with target or staging environments or Delphix Engine with a large number of users to reduce the risk of confusion or failure for other users consider disabling each dSource in the environment. In the case of VDBs, there is no need to disable or shut down. If SnapSync, refresh or other VDB operations occur during the uninstall/install they can be restarted once the procedure is complete. This prevents the need to incur more downtime.

If you are using the certificate that is signed by a certificate authority on the Delphix connector, you can use either of the following ways:

1. After upgrading the Delphix connector, re-execute the steps of Replacing Self-signed Certificates on the Delphix Connector. (see page 973)
2. If you want to use the existing certificates, make sure to follow the below steps:

- Before uninstalling, you must back up `DelphixConnector.jks` file and properties such as `Keystore, Keypass, Storepass` and `UUID` from `DelphixConnector.properties` file to a temporary location.
- Uninstall the Delphix connector.
- Install the Delphix connector.
- Stop the Delphix connector service.

- You can now copy the backed-up .jks file and properties.
- Paste the backed-up .jks file inside the installation directory and the properties to the `DelphixConnector.properties` file.
- Ensure that the `Keystore, Keypass, Storepass and UUID` properties are kept unchanged, and `Keystore` property must point to the correct installation directory.

  Note : For example, if the Keystore is `C\:\\Program Files\\Delphix\\Delphix Connector\\connector\\DelphixConnector.jks`, make sure to place this .jks file is in the same path as mentioned in Keystore.

If you are using windows connector host authentication, you can use either of the following ways:

1. After upgrading the Delphix connector, re-execute the steps for windows connector host authentication (see page 969).
2. If you want to use the existing configuration of windows connector host authentication, make sure to follow the below steps:

- Before uninstalling, you must back up `DelphixConnector.jks` file and properties such as `Keystore, Keypass, Storepass` and `UUID` from `DelphixConnector.properties` file to a temporary location.
- Uninstall the Delphix connector (see page 2102) and then
- Install the Delphix connector. (see page 2098)
- Stop the Delphix connector service.
- You can now copy the backed-up .jks file and properties.
- Paste the backed-up .jks file inside the installation directory and the properties to the `DelphixConnector.properties` file.
- Ensure that the `Keystore, Keypass, Storepass and UUID` properties are kept unchanged, and `Keystore` property must point to the correct installation directory.

> For example, if the Keystore is `C\:\\Program Files\\Delphix\\Delphix Connector\\connector\\DelphixConnector.jks`, make sure to place this .jks file is in the same path as mentioned in Keystore.

9.12.4.5 Reducing the number of AD logins to the domain controller

Upon completion of the upgrades for both the Delphix Continuous Data Engine and the Delphix connector, the Delphix connector will optimize its functionality to minimize the frequency of logins to the Domain controller. This optimization is achieved by implementing an efficient caching mechanism to store the necessary login tokens per environment user.

To achieve this, the Delphix connector will acquire the needed login token per environment user and store it in a cache for 15 minutes. It will use the same cached login token across multiple requests requiring the same environment user privileges for 15 minutes before acquiring a new login token.

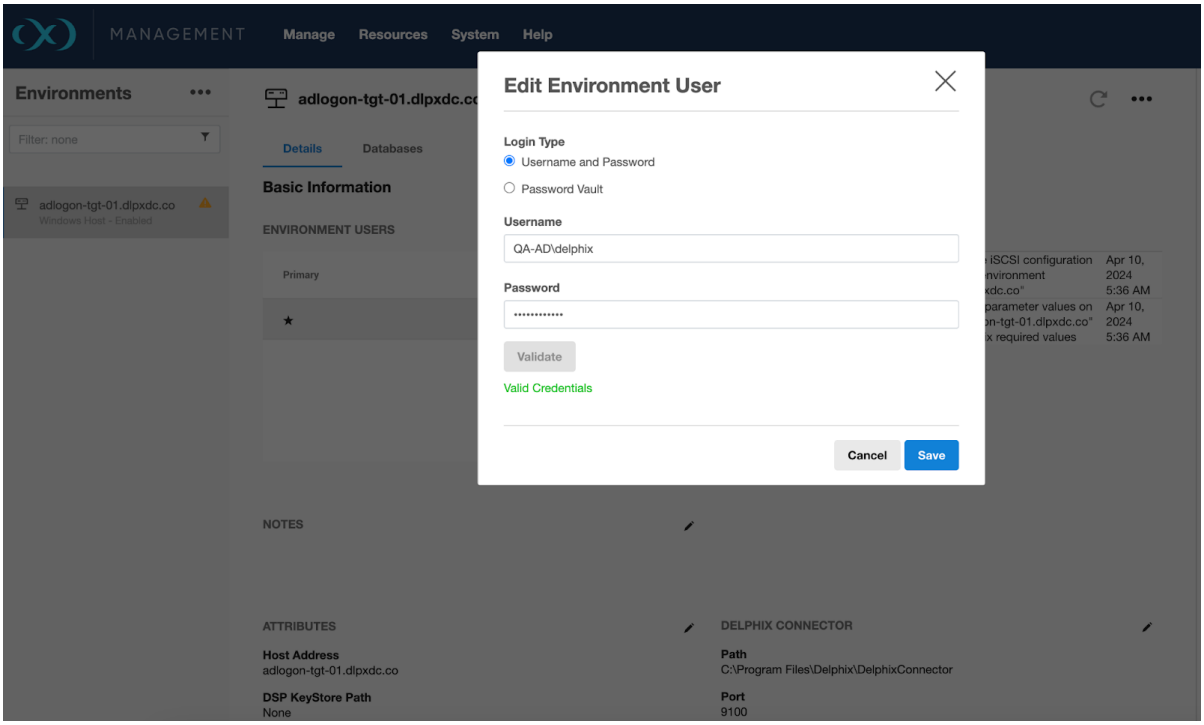
9.12.4.5.1 Manually reloading an existing login token for an environment user

This section describes how to reload an existing login token for an environment user manually.

9.12.4.5.1.1 Procedure

Perform the following steps for manually reloading an existing login token.

1. Login to the Delphix Management application.
2. Click **Manage**.
3. Select **Environments**.
4. In the Environments panel, click the name of an environment to view its attributes.
5. In the Details tab under Environment Users, select the user for which the token needs to be reloaded.
6. Click the edit user icon next to the plus icon to edit the user.
7. Enter the password if the login type is 'User and Password'.
8. Click on **Validate**.



9.12.4.6 Manual discovery for SQL Server instances

Occasionally, SQL Server instances cannot be automatically discovered due to a problem in the Windows registry or if the instance is not running at that time. These SQL Server instances can be discovered using manual discovery.

9.12.4.6.1 Manually adding a SQL server instance

To manually add a SQL Server instance

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Environments** and select the **Databases** tab.
4. Click the **Add Instance** button and enter the following details:
    a. To allow linking and/or provisioning, select **Enabled**.
    b. If this environment will be used as a staging environment, select **Enabled**.
    c. Selecting **Enabled** for Fulltext Installed allows users to do full-text searches.
    d. Add as a **Failover Cluster Source**: When this is Enabled users are asked to enter the Server name. **Note:** This option is only applicable when adding a SQL Server failover cluster instance as a source, it is not available for clusters as a target.
    e. **Name**: Not currently used.
    f. **Port**: Used for IPC connection to the instance.
    g. **Installation path and instance owner**: Installation path of SQL Server. Not currently used.
    h. **Internal version**: The internal version of the SQL Server. This is important during linking and provisioning as the compatible instance is identified based on the internal version.
    i. **Version**: the SQL Server Version required for adding a source instance. **Note:** If the wrong version is entered, Delphix may fail to discover the databases and hence manual discovery may fail.
5. Click **Add**. At this time, if all values are correct, an instance will be created and all its databases will be auto-discovered internally.

> If Delphix Engine is not able to establish a connection using given details, appropriate errors will be displayed and the user will be asked to make the required changes.

9.12.4.6.2 Editing a SQL server instance

1. Login to the **Delphix Management** application.
2. Click **Manage**.

3. Select **Environments**.
4. In the **Environments** panel, select the **Databases** tab.
5. Click the **Pencil** icon to edit an instance.
   If you are editing an instance which you added, you will be able to edit: - You will be able to edit all fields except the name.
6. Click **Save**.

9.12.4.6.3 Refresh a manually added SQL server instance

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Environments**.
4. In the **Environments** panel, click the **refresh** icon on top.
5. Along with auto-discovered instances, it will now also refresh manually added instances and add/remove its databases, based on the current state of the instance.

> If Delphix Engine is not able to establish a connection to the instance, then the appropriate warning will be displayed to the user during database discovery.

9.12.4.6.4 Deleting a manually added SQL server instance

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Environments**.
4. In the **Environments** panel, select the **Databases** tab.
5. You will see a delete icon at the far right of all manually added instances.
6. Click **Delete**. This will delete the instance and all its databases.

> This operation will fail if the chosen instance has databases that have dependencies such as database being used to provision VDBs etc.

9.12.4.7 Adding a SQL Server source environment

This topic describes how to add a SQL Server source environment.

9.12.4.7.1 Prerequisites

- You must have already set up SQL Server target environments, as described in Adding a SQL Server Standalone Target Environment (see page 2111)
    - You will need to specify a target environment that will act as a proxy for running SQL Server instance and database discovery on the source, as explained in Overview of Setting Up SQL Server Environments (see page 2092)
- Make sure your source environment meets the requirements

9.12.4.7.2 Procedure

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Environments**.
4. Next to **Environments**, click the **Actions** menu and select **Add Environment**.
5. In the **Add Environment**wizard, Host and Server tab select:
    a. Host OS: **Windows**
    b. Host Type: **Source**
    c. Server Type:
        - If you are adding a Windows Server Failover Cluster (WSFC), add the environment based on which WSFC feature the source databases use:
            - Failover Cluster Instances Add the environment as a **standalone** source using the **cluster name** or **address**.
            - AlwaysOn Availability Groups Add the environment as a **cluster** source using the **cluster name** or **address**.
        - Otherwise, add the environment as a **standalone** source.
6. Click **Next**.
7. In the Environment Settings tab select a **Connector Environment**.
   Connector environments are used as proxy for running discovery on the source. If no connector environments are available for selection, you will need to set them up as described in Adding a SQL Server Standalone Target Environment. (see page 2111) Connector environments must:
    - have the Delphix Connector installed
    - be registered with the Delphix Engine from the host machine where they are located.

8. Enter the **Environment Name**, **Node Address**, **OS Username**, and **OS Password** for the source environment.
9. Click **Submit.**

As the new environment is added, you will see multiple jobs running in the Delphix Admin Job History to Create and Discover an environment. In addition, if you are adding a cluster environment, you will see jobs to Create and Discover each node in the cluster and their corresponding hosts. When the jobs are complete, you will see the new environment added to the list in the **Environments** panel. If you don't see it, click the **Actions** menu and select **Refresh All**.

9.12.4.8 Adding a SQL Server standalone target environment

This topic describes how to add a SQL Server standalone target environment to the Delphix Engine.

You can use SQL Server targets for three purposes in a Delphix Engine deployment. They can:

- Host a target environment for the provisioning of Virtual Databases (VDBs).
- Host a staging database for a linked dSource and run the validated sync process.
- Serve as a proxy host for database discovery on source hosts.

Regardless of the specific purpose, all Windows targets must have the Delphix Connector installed to enable communication between the host and the Delphix Engine. The instructions in this topic cover initiating the Add Target process in the Delphix Engine interface, running the Delphix Connector installer on the target machine, and then verifying that the target has been added in the Delphix Engine interface.

When target environments are discovered, Delphix will configure the Microsoft iSCSI Initiator Service for Automatic startup.

9.12.4.8.1 Prerequisites

- Make sure that your target environment meets the requirements described in the following sections:
  - SQL Server Support and Requirements (see page 2052)
  - Requirements for Windows iSCSI Configuration (see page 2083)
  - Receive Side Scaling (RSS) for Windows Staging Target and Targets (see page 2117)
- The Directory on which you install the Delphix Connector should have at least 1GB of available space.

If you have taken a snapshot on Windows 2012 or an earlier version, the provisioning, linking, or exporting to Windows 2022 will result in disk errors. Windows event logs list these errors. You can ignore these errors or run the `CHKDSK \F` command.

9.12.4.8.2 Procedure

1. From the machine that you want to use as a target, start a browser session and connect to the **Delphix Management** application**.**
2. Click **Manage**.
3. Select **Environments**.
4. Next to **Environments**, click the **Actions** menu and select **Add Environment**.
5. In the **Add Environment**wizard Host and Server tab, select:
   a. Host OS: **Windows**
   b. Host Type: **Target**.
   c. Server Type: **Standalone**.
6. Click **Next**.
7. In the Environment Settings tab click the download link for the **Delphix Connector Installer**. The Delphix Connector will download it to your local machine.
8. On the Windows machine that you want to want to use as a target, run the Delphix Connector installer. Click **Next**to advance through each of the installation wizard screens.

   The installer will only run on 64-bit Windows systems. 32-bit systems are not supported.
   a. For **Connector Configuration**, make sure there is no firewall in your environment blocking traffic to the port on the target environment that the Delphix Connector service will listen to.
   b. For **Select Installation Folder**, either accept the default folder or click **Browse** to select another.
   c. Click **Next** on the installer final 'Confirm Installation' dialog to complete the installation process and then **Close** to exit the Delphix Connector Install Program.
9. Return to the Delphix Management application.
10. Enter the **Environment Name**, **Host Address**, **Delphix Connector Port**, **OS Username**, and **OS Password** for the target environment.
11. To provide your own Oracle Java select the **Provide my own JDK** checkbox and click **Next**.
12. In the Java Development Kit tab enter the absolute path to your Oracle JDK and click **Next**.
13. Click **Submit.**
    As the new environment is added, you will see two jobs running in the **Delphix Admin Job History**, one to **Create and Discover** an environment, and another to **Create** an environment. When the jobs are complete, you will see the new environment added to the list in the **Environments** panel.

9.12.4.8.3 Post-requisites

1. On the target machine, in the **Windows Start Menu**, click **Services**.
2. Select **Extended Services**.
3. Ensure that the **Delphix Connector** service has a **Status** of **Started**.

4. Ensure that the **Startup Type** is **Automatic**.

9.12.4.9 Adding a SQL Server failover cluster target environment

This topic describes how to add a SQL Server failover cluster target environment to the Delphix Engine.

Adding a failover cluster target environment will discover SQL Server failover cluster instances that are running. You can then provision virtual databases (VDBs) to these failover cluster instances.

9.12.4.9.1 Prerequisites

- You must add each node in the Window Failover Cluster individually as a standalone target environment using a non-cluster address. See Adding a SQL Server Standalone Target Environment. (see page 2111)
    - For a cluster node added as a standalone environment, the Delphix Engine will only discover **non-clustered** SQL Server instances.
    - For a cluster target environment, the Delphix Engine will only discover SQL Server **failover cluster** instances.
- **Each clustered SQL Server instance must have at least one clustered disk added to the clustered instance resource group, which can be used for creating mount points to Delphix storag**e.
    - The clustered drive must have a drive letter assigned to it.
    - The clustered drive must be formatted using the "GUID Partition Table (GPT)" partition style in order for the Delphix Engine to automatically discover the drive letter as a valid option for the cluster instance. An MBR-formatted disk requires manual verification outside of Delphix that the disk has been correctly added to the MSSQL clustered resource group prior to creating the VDB. When provisioning the VDB, you must manually specify the desired MBR disk, because it will not appear in the Delphix GUI.
    - The clustered drive must be added to the clustered instance resource group as a dependency in the Failover Cluster Manager.
- Each node in the cluster must have the **Failover Cluster Module for Windows Powershell**feature installed.
    - While running Windows PowerShell as an administrator, enter this command to load the module: `import-module failoverclusters`
- An additional target environment that can be used as a **Connector Environment** must exist. This environment must NOT be a node in the cluster. See Adding a SQL Server Standalone Target Environment. (see page 2111)

Hotfix required for Windows 2008 R2 hosts

The following hotfix is required for Windows 2008 R2 Cluster nodes:

"0x80070490 Element Not found" error when you enumerate a cluster disk resource by using the WMI MSCluster_Disk class query in a Windows Server 2008 R2-based failover cluster

http://support.microsoft.com/kb/2720218[631]

Windows Cluster Volume Management Software Requirements Only cluster volumes managed by the native Windows Volume Manager are supported. For example, cluster volumes managed by Veritas VxVM are not supported.

If you use third-party volume management software, create a new LU (recommended to be 10GB in size) and add this LU as a clustered resource to the SQL Server instance using native Windows volume management tools. Assign a drive letter for this LU. You can then use this LU as the volume mount point location for Delphix VDB provisioning.

Best Practices

SQL Server failover cluster instances that the Delphix Engine will use should not be used to host databases other than Delphix VDBs.

Cluster environment restrictions

You cannot use failover cluster target environments as staging environments.

However, for VDB **target hosts**, it is important to use the **same edition** of SQL Server software as the **source database**, so that all features available in the source are also available in the VDB.

9.12.4.9.2 Supported roles for each instance type

Failover Cluster Instances, and instances supporting Always On Availability Groups, support a subset of the operations available to Standalone SQL Server instances.

Color	Supported?
Y	Yes
N	No

631 https://support.microsoft.com/kb/2720218

		Environment Role		
Instance Type	**Added As**	**Source Environment**	**Staging Environment**	**Target Environment**
Standalone Instance	Standalone	Y	Y	Y
Failover Cluster Instance (FCI)	Standalone	Y	N	N
Failover Cluster Instance (FCI)	Cluster	N	N	Y
Always On Availability Group (AG)	Cluster	Y	N	N

* Databases that are participating in Availability Groups will not be discovered during the discovery of a Standalone environment.

VDBs cannot be provisioned into availability groups. However, SQL Server instances that participate in an Availability Group can also be added as Standalone Instances.

Using a Failover Cluster Instance as both Source and Target
A Failover Cluster Instance added as an environment once (as either a Source or Target environment) cannot be used as both a Source and Target.
If this is required, the environment can be added twice:

- Once as a Standalone Source environment
- Once as a Cluster Target environment

The Standalone Source environment can be used for linking dSources, and the Cluster Target environment is used for provisioning VDBs.
As suggested by the Best Practice note earlier in this article, this is not a recommended configuration. Where possible, SQL Server failover cluster instances that the Delphix Engine will use as a target should not be used to host databases other than Delphix VDBs.

If you have taken a snapshot on Windows 2012 or an earlier version, the provisioning, linking, or exporting to Windows 2022 Failover Cluster Target might result in disk errors. In such a case, contact Delphix Customer Support

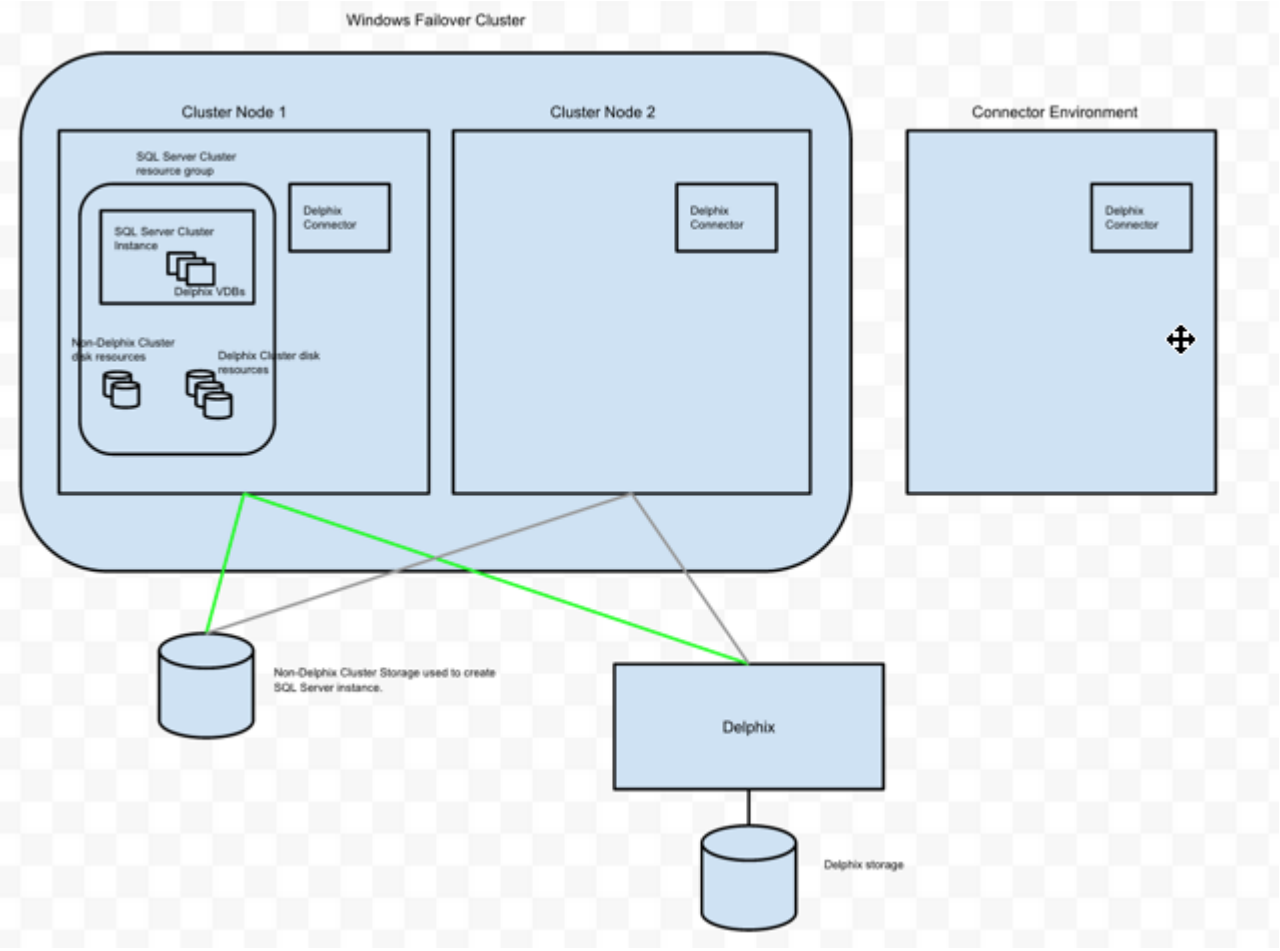
> Error
> The following error may be encountered during the Windows 2022 AG and FCI clusters creation process on VMware ESXi.
> `An error occurred while executing the test.Unable to connect to <hostname.domainname.com> via WMI. This may be due to networking issues or firewall configuration href="http://hostname.domainname.com">hostname.domainname.com>.Invalid namespace`
> This failure is due to `Validate switch enabled Teaming configuration` validation. The workaround is to skip this validation.

9.12.4.9.3 Procedure

1. Click **Manage**.
2. Select **Environments**.
3. Next to **Environments**, click the **Plus** icon and select **Add Environment**.
4. In the **Add Environment**wizard, Host and Server tab select:
   a. Host OS: **Windows**
   b. Host Type: **Target**.
   c. Server Type: **Cluster**.
5. Click **Next**.
6. In the Environment Settings tab specify the environment Name and cluster address of the Windows Failover Cluster.
7. Select a host that is NOT a node in the cluster as the **Connector Environment**.
8. Enter the **OS Username** and **OS Password** for the target environment.
9. Click **Validate Credentials**.
10. Click **Submit** to confirm the target environment addition request.

In the Delphix Engine interface, you will see a new icon for the target environment, and two jobs running in the **Delphix Admin Job History**: one to **Create and Discover** an environment, and another to **Create** an environment. When the jobs are complete, click the icon for the new environment, and you will see the details for the environment.

9.12.4.9.4 Example environment



In this example environment, the Delphix Connector was installed on **Connector Environment**, **Cluster Node 1**, and **Cluster Node 2**. Each host was added to Delphix as standalone target environments.

Next, the **Windows Failover Cluster** was added as a Windows Target Cluster environment using the cluster address. **Cluster Node 1** is currently the active node for the SQL Server Failover Cluster resource group. Delphix has exported iSCSI LUs and has created the corresponding Cluster Disk resources for each VDB.

9.12.4.10 Additional SQL Server environment topics

This section covers the following topics:

- Microsoft SQL Server bypass powerShell execution policy (see page 2118)
- Delphix as a backup solution to SQL Server (see page 2118)
- Delphix managed backups on secondary nodes of SQL Server alwaysOn availability group cluster (see page 2127)
- Receive side scaling for windows staging target and targets (see page 2130)
- Validated sync environment (see page 2133)
- Changing the hostname (IP address) or IQN of a SQL server target or staging host (see page 2135)

9.12.4.10.1 Microsoft SQL Server bypass powerShell execution policy

Hooks are executed with “PowerShell -ExecutionPolicy RemoteSigned” whereas user-configured pre/post sync and pre/post provision PowerShell scripts are executed with “PowerShell -ExecutionPolicy Bypass”. If this PowerShell configuration exceeds the security privileges of business standards, a Group Policy Object can be configured in Windows to manage PowerShell execution policy. If configured, we require that the script execution policy is set to “RemoteSigned”. The primary benefit is that the Group Policy configuration takes precedence and overrides the execution policy set in PowerShell, even if you run PowerShell as an Administrator. Alternatively, using MSSQL hooks instead of pre/post scripts will naturally execute with a “RemoteSigned” execution policy.

Refer to the Microsoft Documentation on Execution Policies for more information: https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_execution_policies?view=powershell-6

> The Group Policy is per machine and will apply to PowerShell 2.0 and later.

9.12.4.10.2 Delphix as a backup solution to SQL Server

9.12.4.10.2.1 Using Delphix as a backup solution to SQL Server

Delphix provides you the option to automatically manage backups from SQL Server source databases into the Delphix Engine. Prior to Delphix 4.2, users could not link source databases that were backed up by unsupported backup software. In Delphix 4.2, a feature known as Delphix Managed Backups was introduced that allows you to have Delphix take and manage backups from your source database directly into Delphix storage. This is the first step in Delphix becoming a full-featured backup solution for SQL Server databases. When the Delphix Engine manages the backups for a dSource, it takes regular, copy-only full backups of the source database, so activating the feature will not interfere with existing backup management solutions. You can configure the schedule of when the Delphix Engine takes these copy-only full backups by specifying a SnapSync policy for the dSource. You can change the SnapSync policy for a dSource at any time by visiting the policy screen; there, you can either select a new SnapSync policy or modify the current one.

If you use a backup solution that is not supported by Delphix, you cannot use your existing backups to keep your dSources in sync. However, enabling Delphix Managed Backups will overcome this issue by using automatic copy-only full backups to keep dSources in sync. Currently, dSources linked when this feature is enabled will not support LogSync functionality, which means that you can only provision VDBs from snapshots and not from any time between snapshots. Additionally, in the current release, the Delphix Engine cannot take differential or transaction log backups of the source database.

9.12.4.10.2.2 Linking SQL Server dSources with Delphix managed backups

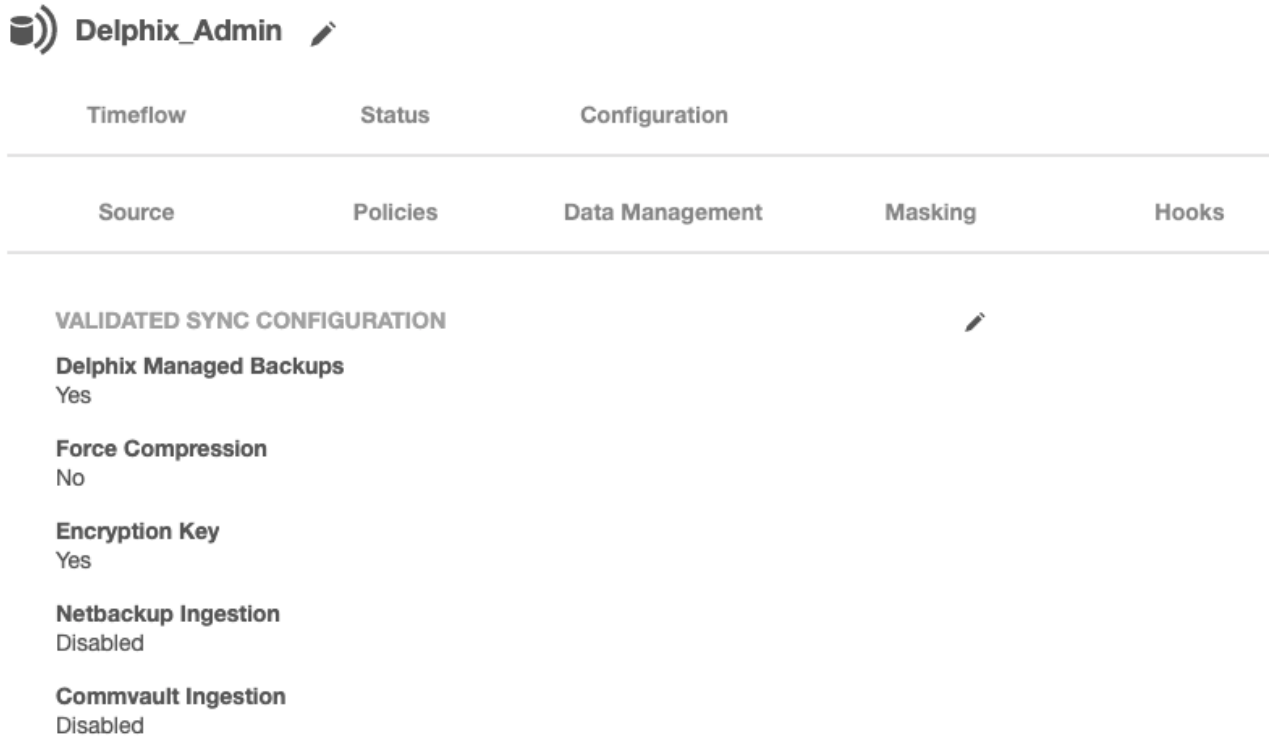
The **Data Management** page of the link wizard for SQL Server dSources provides the option to enable Delphix Managed Backups.

It possible to enable this feature here at link time or toggle it on after the link for future syncs. If you enable this feature, the dSource can only use copy-only full backups taken by the Delphix Engine to stay in sync with

its source; the Delphix Engine will prohibit syncing using existing backups. Checking the **Enabled** box results in the following changes to the **Data Management** page:

- The initial load option is set to a copy-only full backup taken by the Delphix Engine
- The ability to provide a backup path disappears
- A SnapSync policy selection screen appears in the **Policies** page

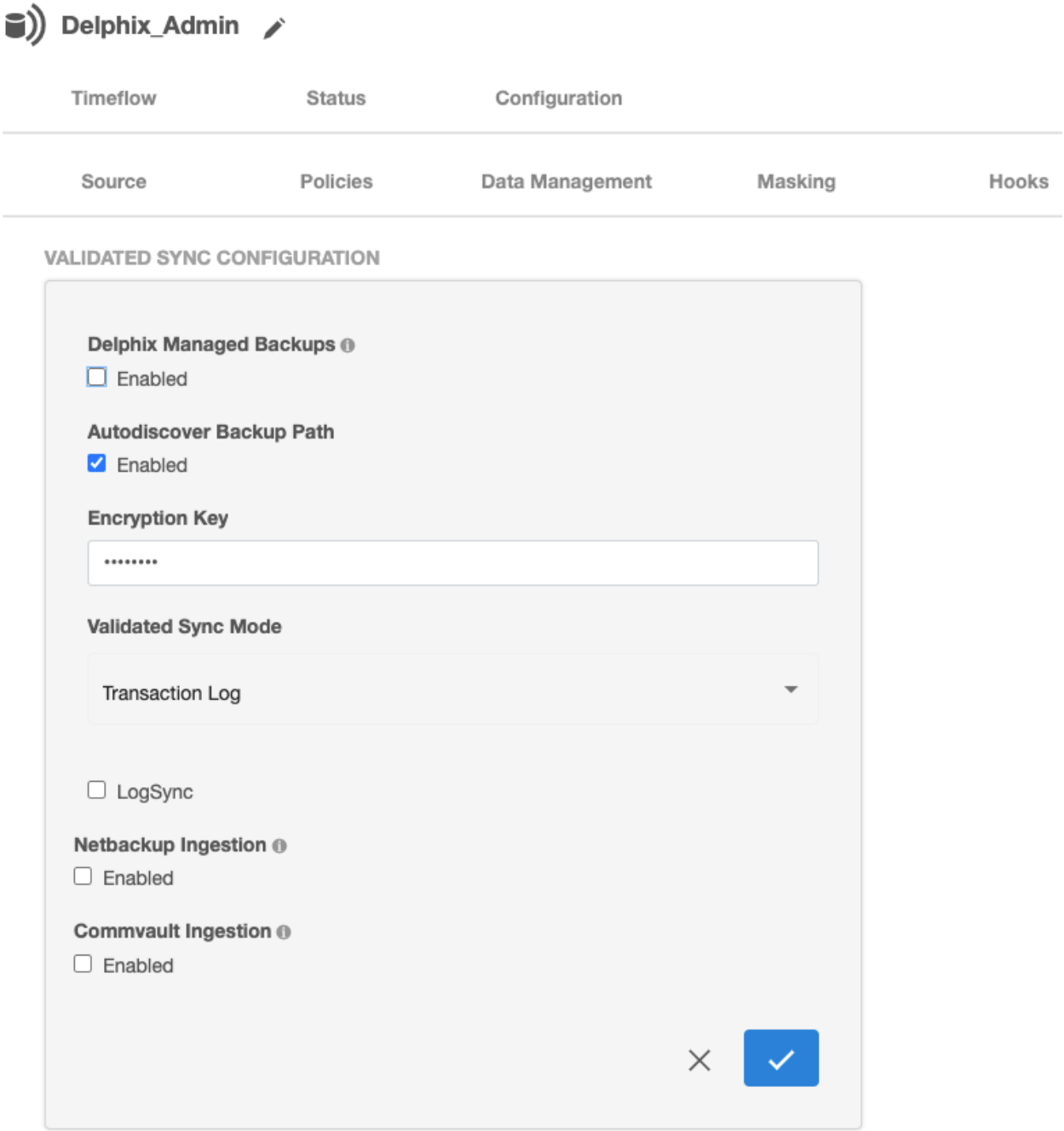
You can select from the list of existing SnapSync policies if the one you want doesn't exist you will need to create a new one on the **Policies** page under the **Manage** dropdown. Proceeding through the remainder of the link wizard will create a dSource with Delphix-managed backups enabled. You can confirm that a dSource has the feature by selecting the dSource and going to the **Configuration** > **Data Management** tab after creation and checking the **Delphix Managed Backups** section, as displayed below:



To disable/enable this feature after linking:

Toggling Delphix Managed backups is supported for SQL Server dSources. This means a dSource that was previously created with Delphix Managed Enabled can have regular validated sync and similarly a dSource that had external backups can use Delphix Managed Backups.

To disable Delphix Managed Backups for a specific dSource, select that specific dataset and go to **Data Management** tab under the **Configuration** tab.



Click the edit button and uncheck this feature. Then update all new inputs.

To enable this feature just go to the same edit menu.

9.12.4.10.2.3 Backup compression feature overview

Since Delphix Engine 5.2, the Delphix Engine has allowed compression to be enabled ("forced") for SQL Server backups which use Delphix Managed Backups. Delphix Managed Backups are used to synchronize SQL Server dSources when existing SQL Server backup files cannot be made available to the Staging server, or if a third-party backup vendor is used that is not yet supported by Delphix. For more information on this

functionality see Delphix as a Backup Solution to SQL Server (see page 2118) and Linking a dSource with SQL Server. (see page 2139)

Backup compression is preferable in the following situations when:

- The default backup compression setting for the Source SQL Server instance is Disabled (0)
- The Source database does not use Transparent Data Encryption (TDE)
- The Source server has available CPU resources to perform compression

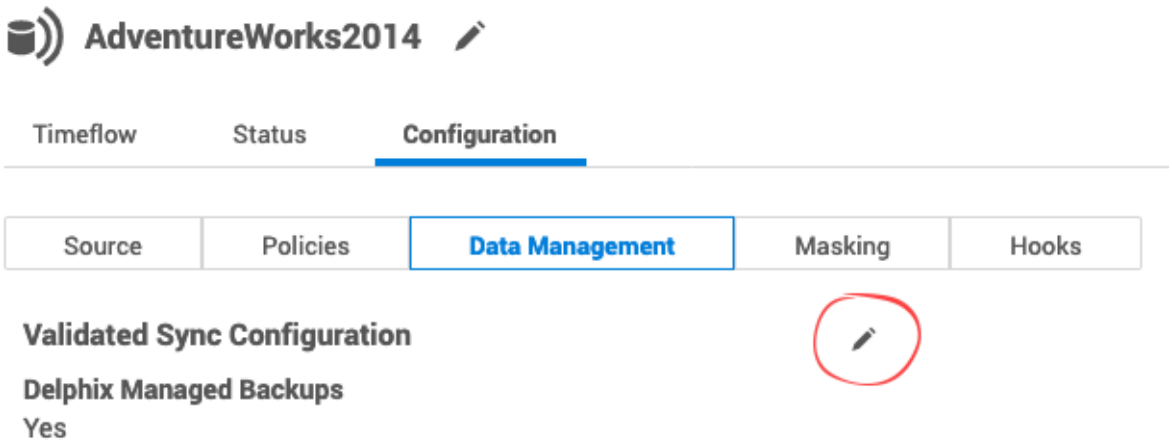
Where the compression ratio of a compressed backup exceeds 3:1, this will reduce the amount of data (by half) that must be transferred over the network to perform a backup, resulting in much faster SnapSync operations.

How to enable backup compression

If the SQL Server instance's default backup compression setting is enabled (see Microsoft's document View or Configure the backup compression default Server Configuration Option[632]), no specific action is required. Backups will automatically be compressed in accordance with this setting.

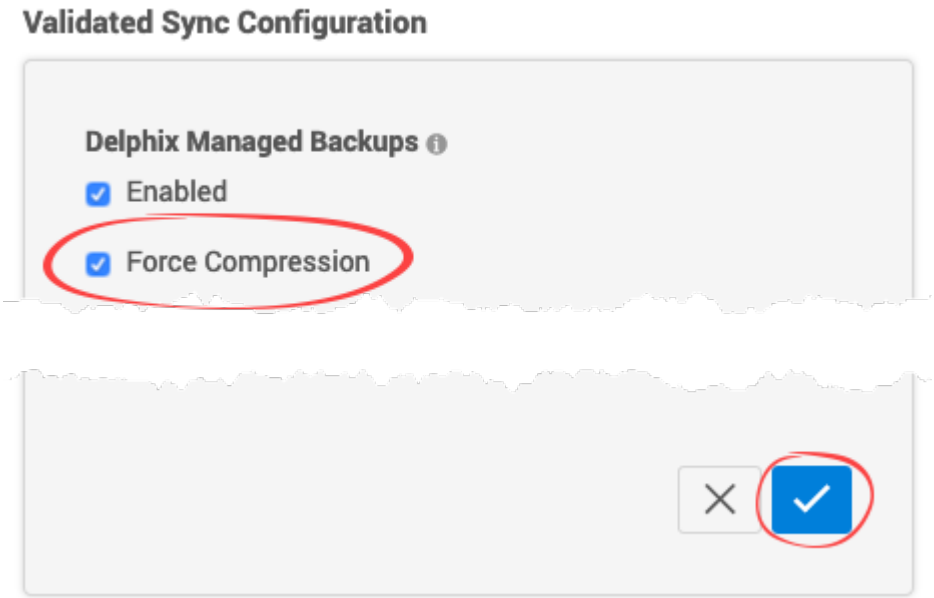
If backups are not automatically compressed, the Delphix Engine (5.2 and later) can be configured to force compressed backups using the following steps:

- Login to the Delphix Engine's Management interface.
- Open the **Manage > Datasets** screen.
- Locate and select the dSource (using the tree on the left).
- Navigate to the **Configuration > Data Management** pane.
- Click the **Edit** (pencil) icon to modify the Validated Sync settings.



- Select the **Force Compression** checkbox, then the select the Tick icon to save the changes

632 https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/view-or-configure-the-backup-compression-default-server-configuration-option?view=sql-server-2017



The change will automatically take effect during the next scheduled Snapshot/SnapSync operation.

9.12.4.10.2.4 Delphix managed backups on secondary nodes of alwaysOn availability group cluster

The Always On availability groups feature is a high-availability and disaster-recovery solution that provides an enterprise-level alternative to database mirroring. Introduced in SQL Server 2012 (11.x), Always On availability groups maximizes the availability of a set of user databases for an enterprise. An *availability group* supports a failover environment for a discrete set of user databases, known as *availability databases*, that failover together. An availability group supports a set of read-write primary databases and one to eight sets of corresponding secondary databases. Optionally, secondary databases can be made available for read-only access and/or some backup operations. Some customers prefer using secondary nodes for backup to reduce the load on the primary node while some customers are OK with using the primary node. To provide more flexibility for customers using Always On availability groups with Delphix Managed backups, delphix provides an option to choose backup policy during linking of a dSource.

They can pick one of the following options that describe their backup policy as shown below.

- Primary (**default option**): Backups are taken only on the primary node.
- Secondary Only: Customers who never want to use primarily for their backups must choose this option as it ensures backups never go to the primary node. If the secondary nodes are down, the backups fail but do not use primary at all.
- Secondary preferred: As the name suggests, one of the secondary nodes is used for backup. The backup will be taken on primary if none of the secondary nodes can be used for backup. So, customers who want reliable backups even if they are taken on the primary node if required should use this option.

Data Management

Configure and Administer data details.

Managed Backups

☑ Enabled

Initial Load

Delphix will take a copy-only full backup of your source database

☐ Force Compression

Backup Policy

◉ Primary only

Backups only go to the primary node.

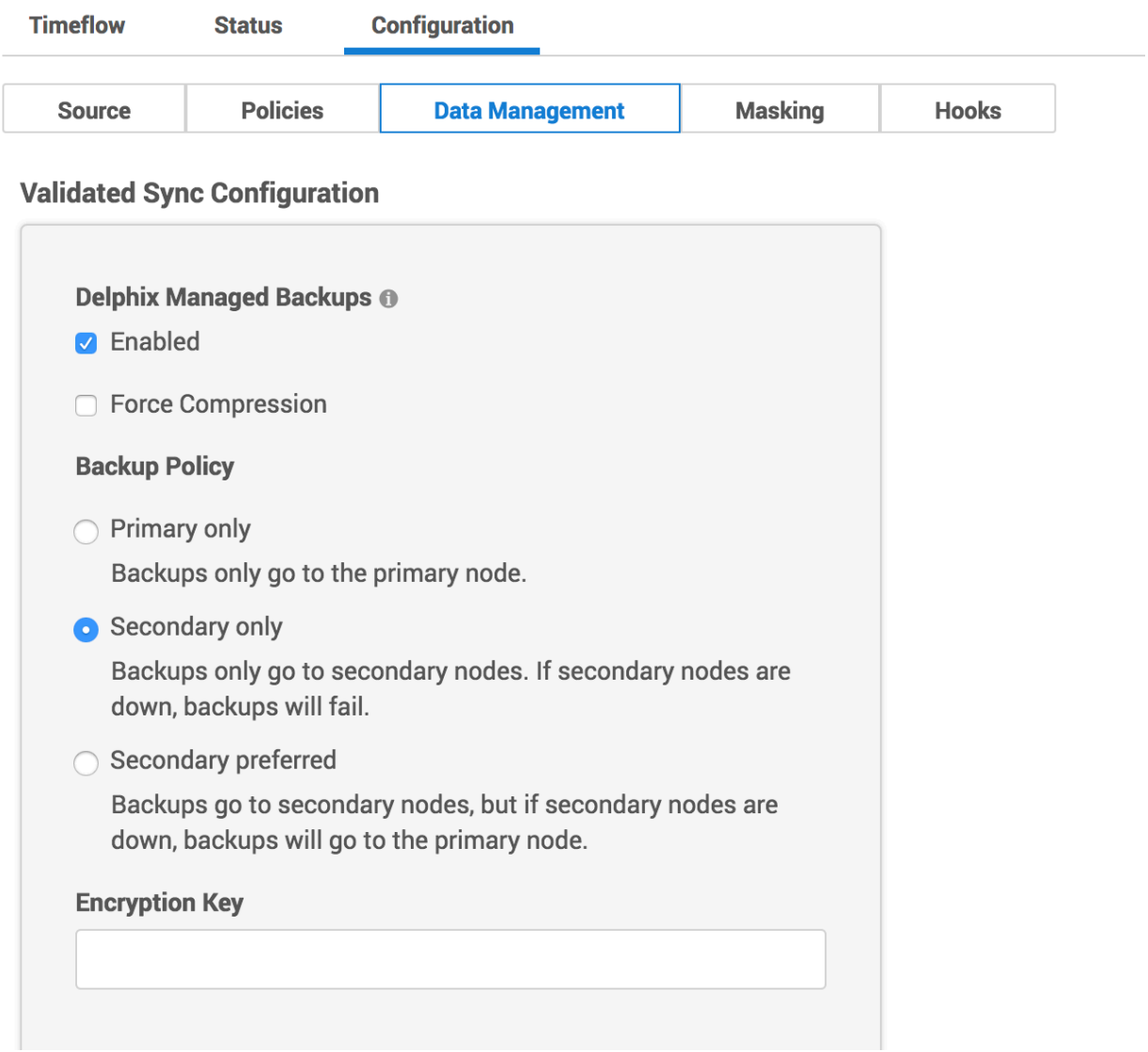
○ Secondary only

Backups only go to secondary nodes. If secondary nodes are down, backups will fail.

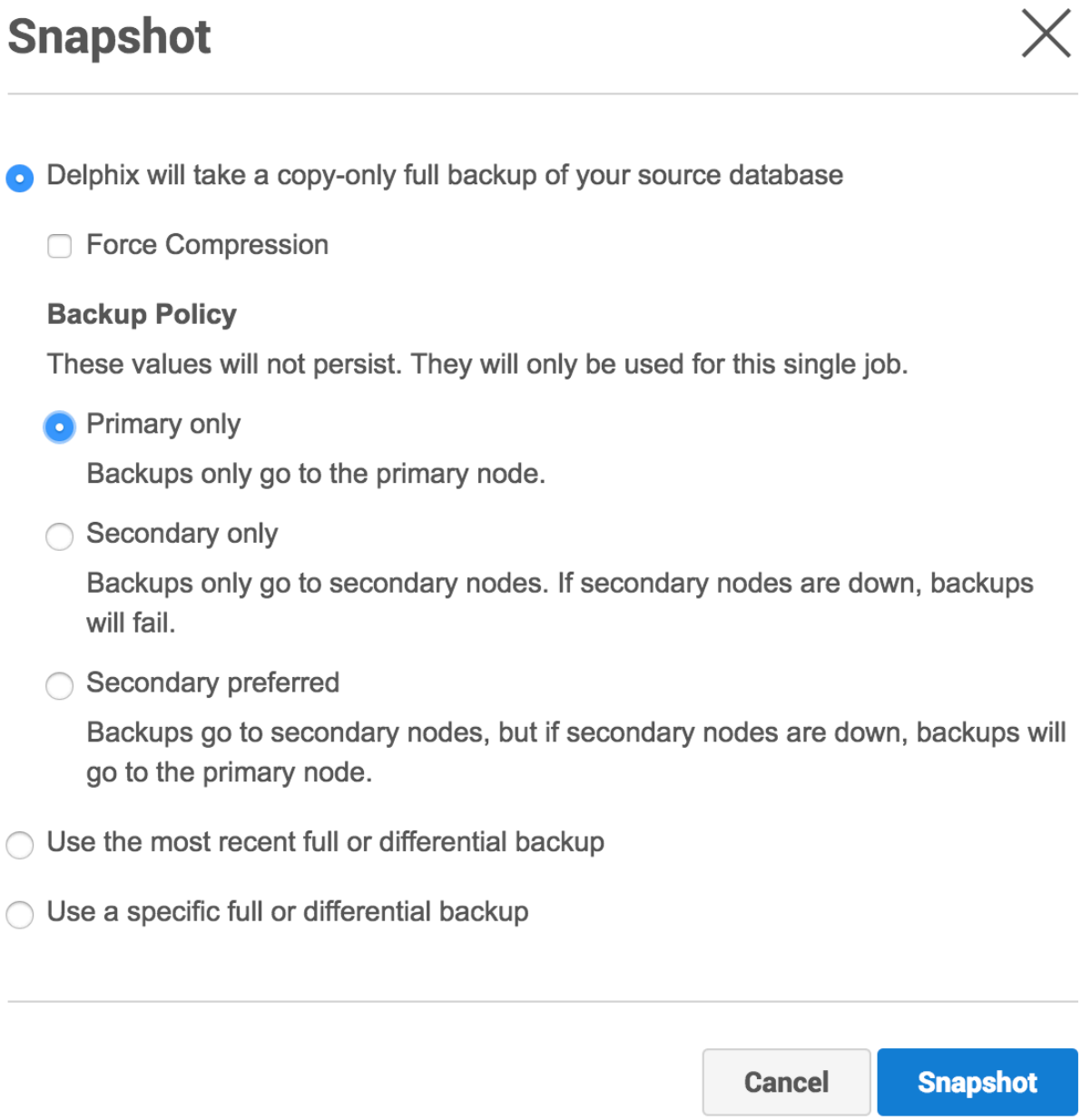
○ Secondary preferred

Backups go to secondary nodes, but if secondary nodes are down, backups will go to the primary node.

The backup policy can be changed after dSource creation too as shown below.



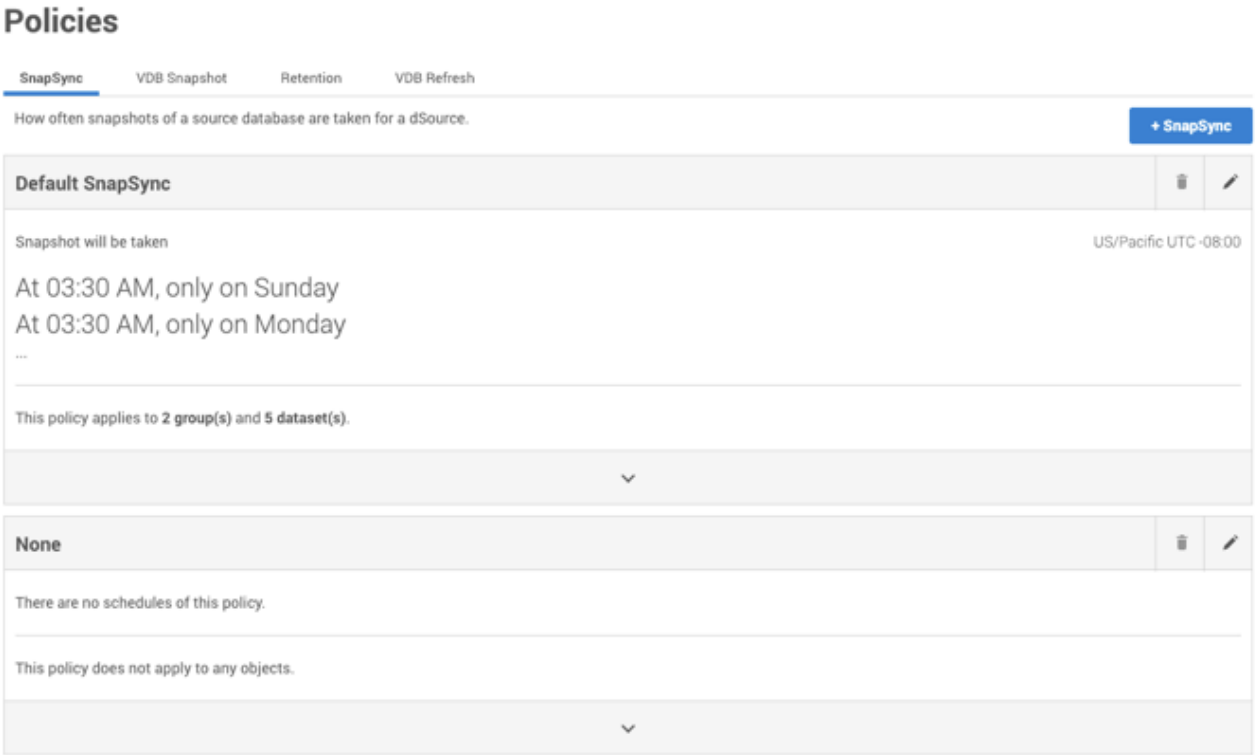
The backup policy is not only provided during the linking phase of a dsource but also during the manual snapsync phase too to provide more flexibility to customers as shown below. Default option is the Backup Policy selected during linking phase.



9.12.4.10.2.5 Understanding snapSync policies

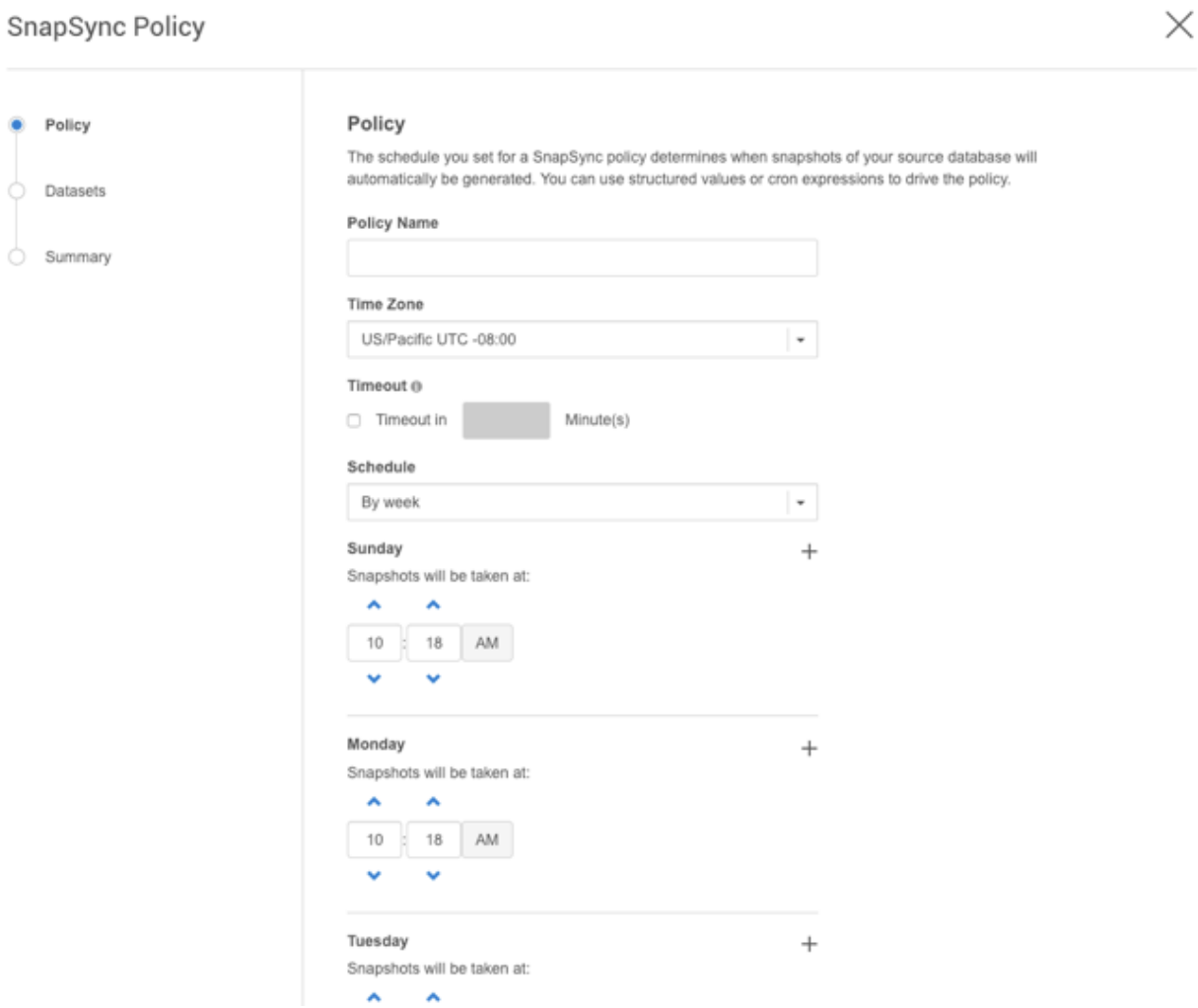
SnapSync policies provide you the ability to specify how frequently the Delphix Engine takes a copy-only full backup of a source database when Delphix Managed Backups are enabled. Selecting an initial SnapSync policy is mandatory at dSource link time. However, you can change the SnapSync policy that the Delphix Engine applies to a dSource at any time by visiting the policy management screen:

1. Click **Manage**.
2. Click **Policies**.



Check SnapSync Policy

For dSources that have Delphix-managed backups enabled, the current SnapSync policy will be displayed under the **SnapSync** column. The rows corresponding to dSources that do not use Delphix Managed Backups will be grayed out. Clicking the **current SnapSync policy** for a dSource will display a drop-down menu of existing SnapSync policies along with the option to create a new SnapSync policy. Selecting a SnapSync policy from this list will change the current SnapSync policy for the dSource. When creating a new policy, you will see the following screen:



Create New SnapSync Policy

Here, you can configure the frequency with which the Delphix Engine takes backups of your source database. You can modify these schedules at any time by clicking the **Modify Policy Templates** button in the upper right-hand corner of the policy management screen.

The **Timeout** field above specifies how long a SnapSync job is allowed to run before it is terminated. If a SnapSync job exceeds its timeout window, the Delphix Engine discards the new backup and rolls back the dSource to the most recent snapshot.

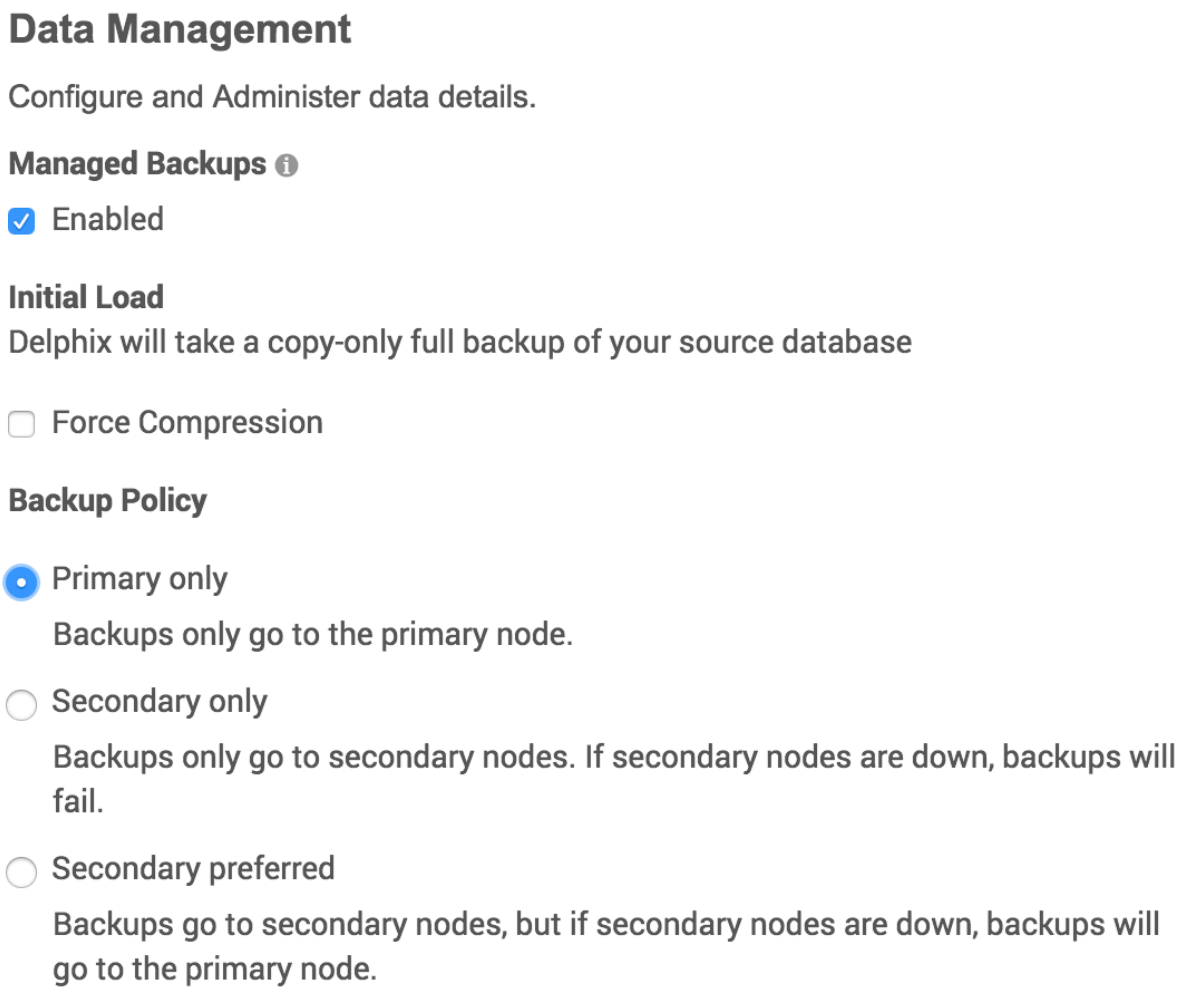
9.12.4.10.3 Delphix managed backups on secondary nodes of SQL Server alwaysOn availability group cluster

The Always On availability groups feature is a high-availability and disaster-recovery solution that provides an enterprise-level alternative to database mirroring. Introduced in SQL Server 2012 (11.x), Always On availability groups maximizes the availability of a set of user databases for an enterprise. An *availability group* supports a failover environment for a discrete set of user databases, known as *availability databases*, that fail over together. An availability group supports a set of read-write primary databases and one to eight sets of corresponding secondary databases. Optionally, secondary databases can be made available for read-only access and/or some backup operations. Some customers prefer using secondary nodes for backup to reduce load on the primary node while some customers are OK with using primary node. To

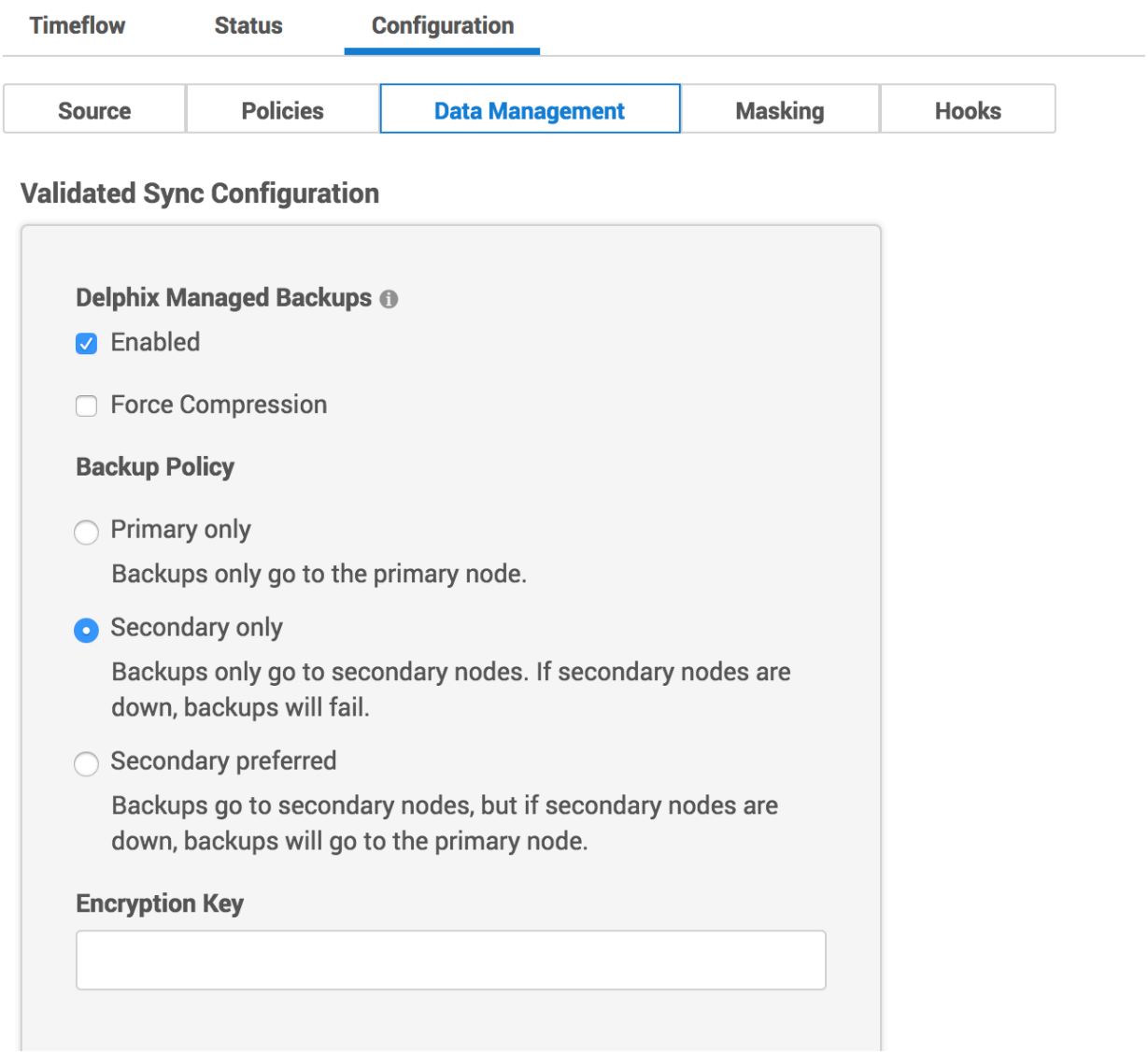
provide more flexibility for customers using Always On availability groups with Delphix Managed backups, delphix provides an option to choose backup policy during linking of dSource.

They can pick one of the following options that describe their backup policy as shown below.

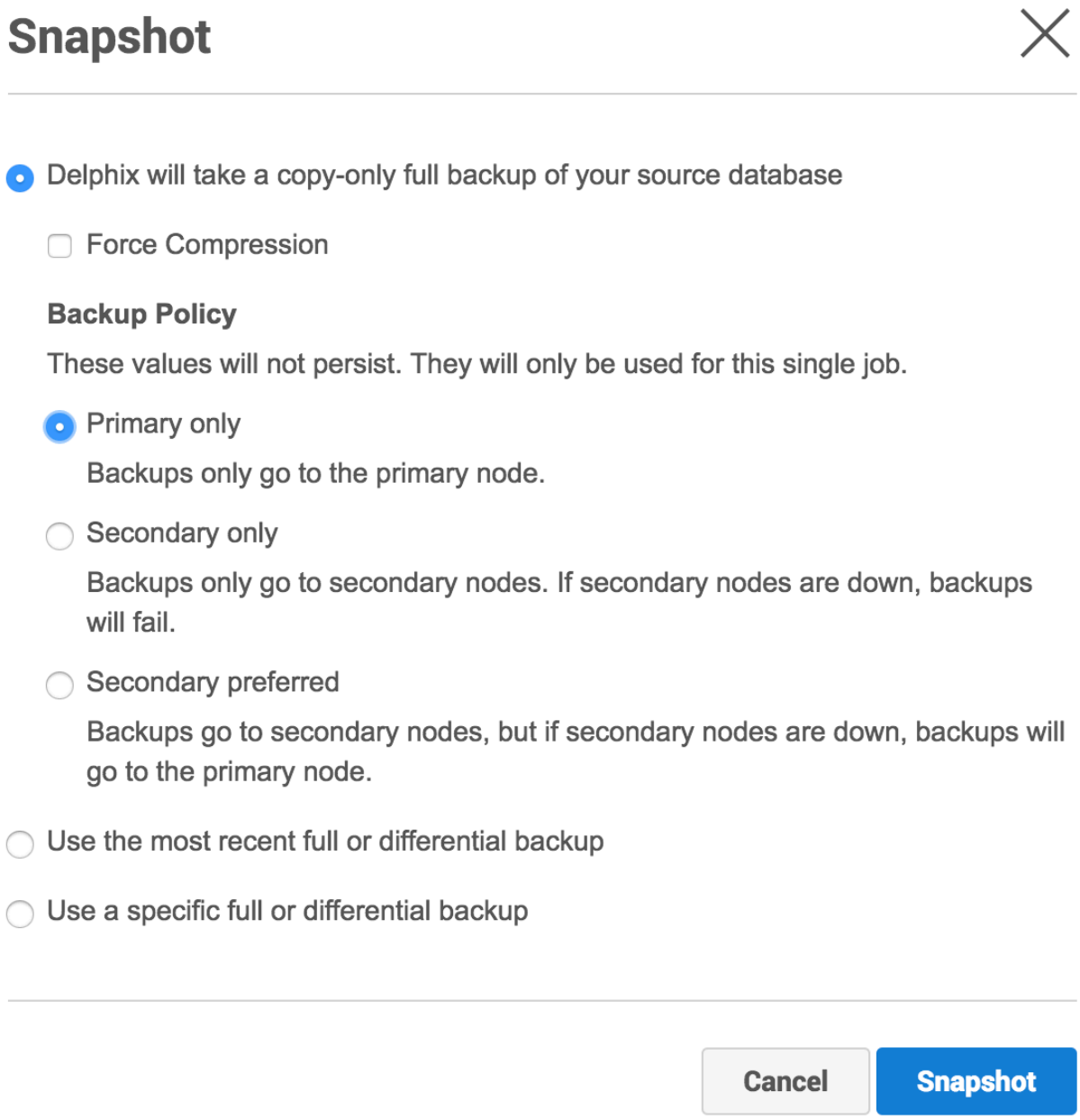
- Primary (**default option**): Backups are taken only on the primary node.
- Secondary Only: Customers who never want to use primarily for their backups must choose this option as it ensures backups never go to the primary node. If the secondary nodes are down, the backups fail but do not use primary at all.
- Secondary preferred: As the name suggests, one of the secondary nodes is used for backup. The backup will be taken on primary if none of the secondary nodes can be used for backup. So, customers who want reliable backups even if they are taken on the primary node if required should use this option.



The backup policy can be changed after dSource creation too as shown below.



The backup policy is not only provided during the linking phase of a dsource but also during the manual snapsync phase too to provide more flexibility to customers as shown below. The default option is the Backup Policy selected during the linking phase.



9.12.4.10.4 Receive side scaling for windows staging target and targets

Enabling Receive Side Scaling (RSS) on a Windows Target and Staging Target can have a significant improvement in the overall IO throughput to the Delphix Engine and is a best practice. RSS enables network adapters to distribute the kernel-mode network processing load across multiple processor cores in multi-core computers. The distribution of this processing makes it possible to support higher network traffic loads than would be possible if only a single core were to be used.

More information on RSS can be found here[633].

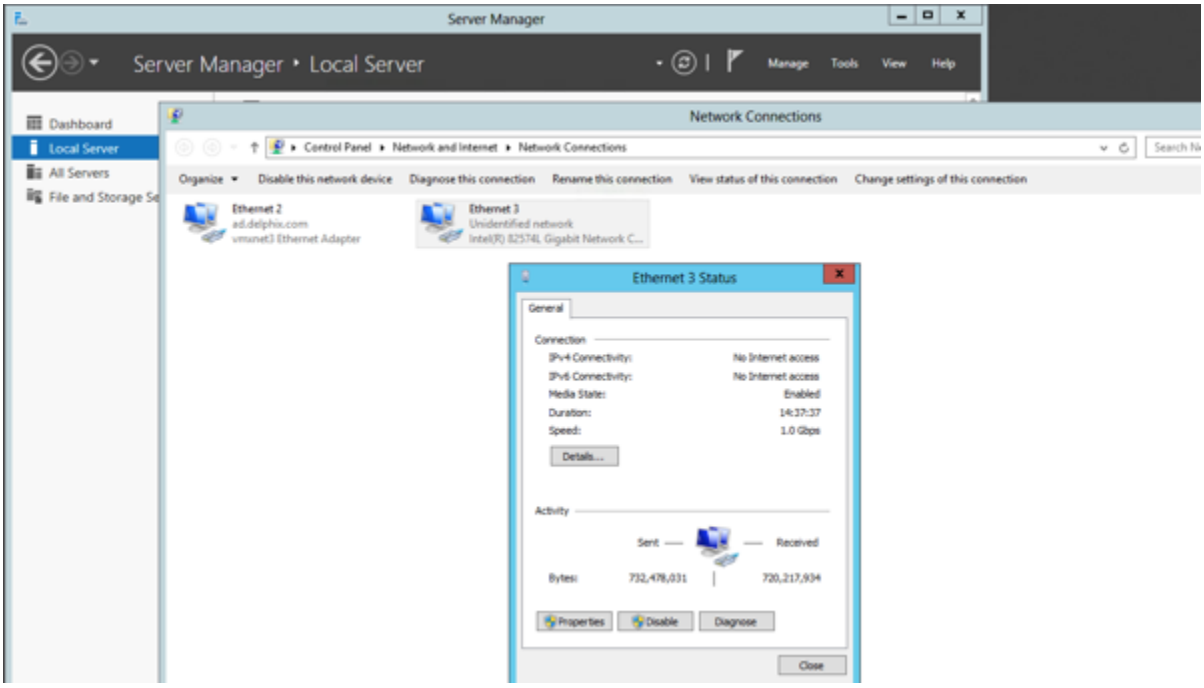
633 https://technet.microsoft.com/library/hh997036.aspx

Enabling RSS on the network interface will force the network service to restart and will cause a momentary loss of connectivity on that network interface

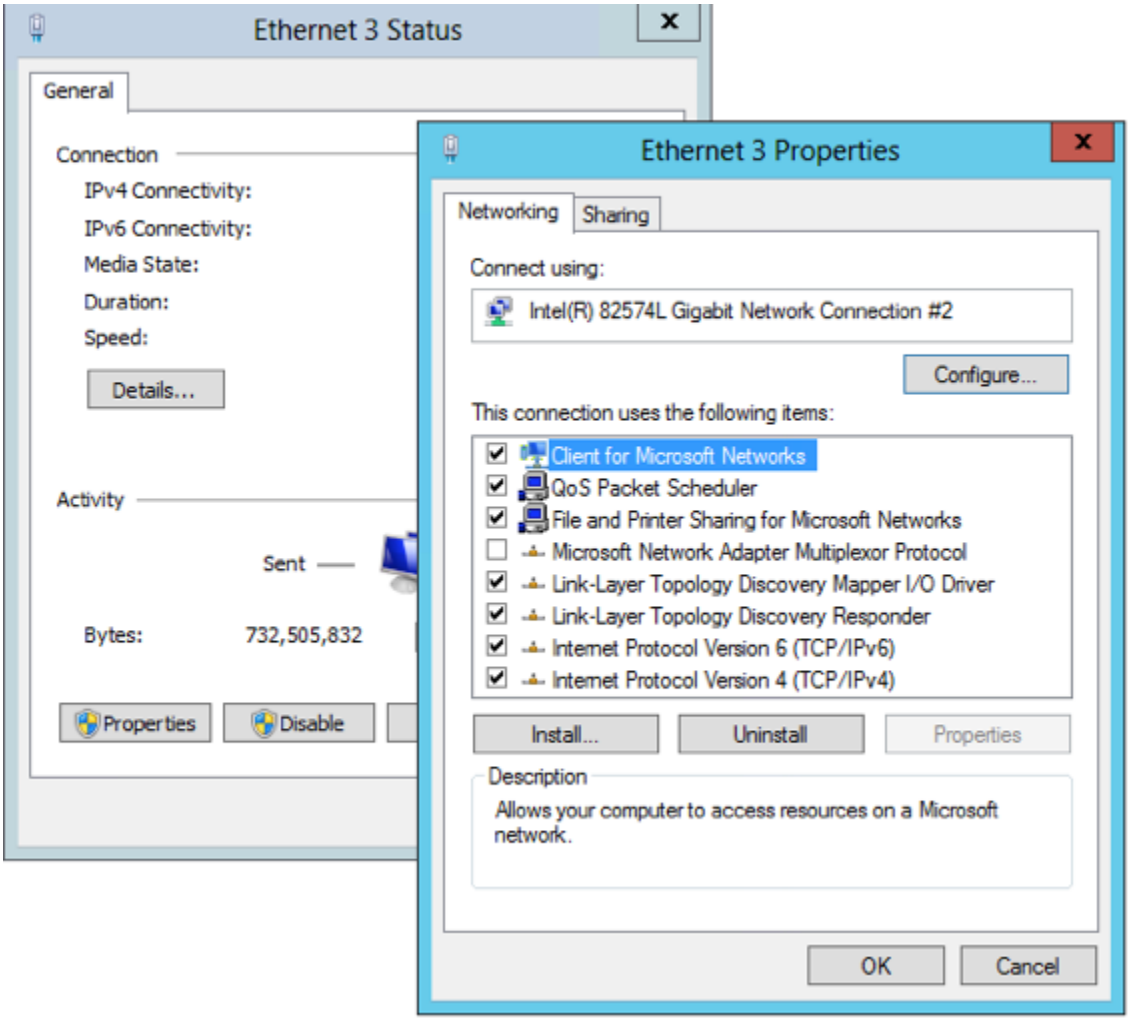
Because hyper-threaded CPUs on the same core processor share the same execution engine, the effect is not the same as having multiple core processors. For this reason, RSS does not use hyper-threaded processors.

9.12.4.10.4.1 Steps to implement RSS on Windows

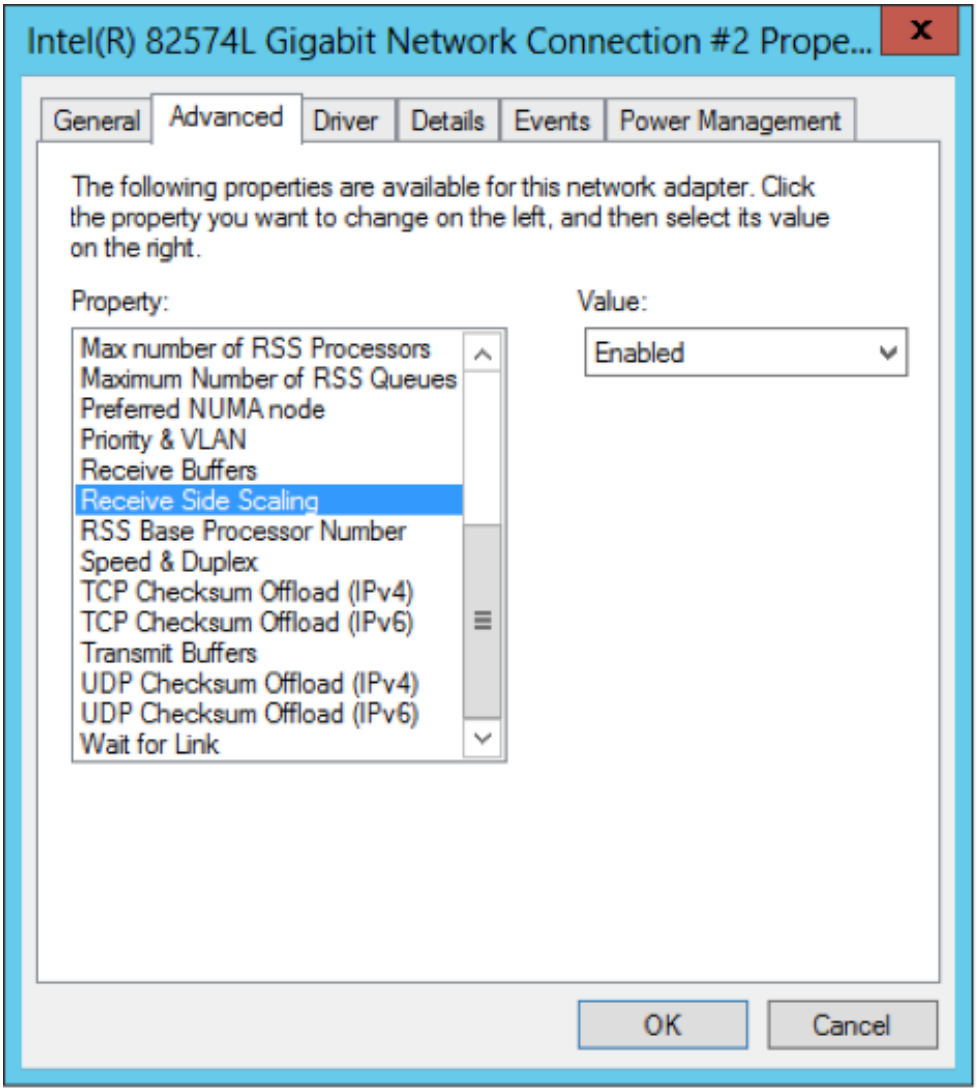
1. From Server Manager/Local Server/Network Connections select the NIC that Delphix will be connecting to.



2. Select Properties and then Configure.



3. From the Property menu on the left, select **Receive Side Scaling**, confirm that it is set to **Enabled**, and select **OK** to close each of the open windows.



9.12.4.10.5 Validated sync environment

This topic describes additional requirements for SQL Server environments that will be used as targets for validated sync. You must configure a staging (Validated Sync) environment as a target, with a few additional requirements.

9.12.4.10.5.1 Requirements for SQL server validated sync target environments

Each SQL Server target environment used for validated sync must meet these requirements:

- Only standalone target environments can be used as validated sync target environments. Windows Failover Cluster target environments and SQL Server Failover Cluster instances cannot be used.
- The SQL Server instance must be the same version as the instance hosting the source database. For more information about compatibility between different versions of SQL Server, see SQL Server Support Matrix (see page 2052)

- The owner of the SQL Server instances on the target environment that are used for the staging databases must have SMB read access to the location containing the backup images of the source databases
- If the source database is backed up with third-party backup software like LiteSpeed or Red Gate SQL Backup Pro, you must install the backup software on both the source and the validated sync environment. For backup software compatibility requirements, see SQL Server Support Matrix (see page 2052)

9.12.4.10.5.2 Add the validated sync environment

The order is important. Add the validated sync environment as the first step in setting up the SQL Server topology.

1. From the machine that you want to use as a target, start a browser session and connect to the **Delphix Management** application**.**
2. Click **Manage**.
3. Select **Environments**.
4. Next to **Environments**, click the **Actions** menu and select **Add Environment**.
5. In the **Add Environment**wizard Host and Server tab, select:
   a. Host OS: **Windows**
   b. Host Type: **Target**.
   c. Server Type: **Standalone**.
6. Click **Next**.
7. In the Environment Settings tab click the download link for the **Delphix Connector Installer**. The Delphix Connector will download to your local machine.
8. On the Windows machine that you want to want to use as a target, run the Delphix Connector installer. Click **Next**to advance through each of the installation wizard screens.

   The installer will only run on 64-bit Windows systems. 32-bit systems are not supported.
   a. For **Connector Configuration**, make sure there is no firewall in your environment blocking traffic to the port on the target environment that the Delphix Connector service will listen to.
   b. For **Select Installation Folder**, either accept the default folder, or click **Browse** to select another.
   c. Click **Next** on the installer final 'Confirm Installation' dialog to complete the installation process and then **Close** to exit the Delphix Connector Install Program.
9. Return to the Delphix Management application.
10. Enter the **Environment Name**, **Host Address**, **Delphix Connector Port**, **OS Username**, and **OS Password** for the target environment.
11. To provide your own Oracle Java select the **Provide my own JDK** checkbox and click **Next**.
12. In the Java Development Kit tab enter the absolute path to your Oracle JDK and click **Next**.
13. Click **Submit.**

As the new environment is added, you will see two jobs running in the **Delphix Admin Job History**, one to **Create and Discover** an environment, and another to **Create** an environment. When the jobs are complete, you will see the new environment added to the list in the **Environments** panel.

9.12.4.10.6 Changing the hostname (IP address) or IQN of a SQL server target or staging host

This topic describes how to change the hostname (IP address) or iSCSI Qualified Name (IQN) of a Windows Target or Staging host.

By default, Windows servers generate an IQN based on the hostname assigned to it. Changing the hostname will change the host IQN as well. Because the Delphix Engine exports storage for dSources and VDBs to Windows hosts using iSCSI, changes to the Windows hostname must be made according to the following procedure. If you have set a non-default IQN on a Windows Target or Staging host, and want to change that IQN, you must follow these procedures.

> Changing the hostname or IQN of a Windows target or staging server requires that you modify the iSCSI Initiator configuration on the Windows host. Doing so incorrectly can cause failures in dSources, VDBs, or non-Delphix users of iSCSI on the Windows host.
>
> The instructions in this topic describe how to change the IQN using the iscsicli command-line utility. Because many people are less familiar with the iscsicli utility, the instructions also include information for using the iSCSI Initiator graphical user interface.
>
> Failing to carefully follow the steps below in sequence can cause availability issues for your dSources and VDBs. If you have questions about the following instructions, please contact Delphix Support for help[634].

1. Disable the dSources.
2. Disable the VDB's.

> If your Windows server has dSources or VDBs from more than one Delphix Engine, you will need to disable the dSources and VDBs on each Delphix Engine.

3. Remove any remaining persistent volumes from the Windows server. From the Server Manager\Tools**iSCSI Initiator** configuration tool, use the options available in the **Volumes and Devices** tab. or, Follow these steps to use the iscsicli command-line utility: List the persistent volumes.

```
PS C:\> iscsicli reportpersistentdevices
Microsoft iSCSI Initiator Version 6.1 Build 7601
```

634 https://support.delphix.com/

```
Persistent Volumes
"\\?\storage#volume#{bb38add1-d03f-11e1-8767-005056b37fe6}
#0000000008010000#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}"
"C:\Program Files\Delphix\DelphixConnector\564d6fbb-df9d-e90b-00f1-da37b17011d3-
staging-15\ARCHIVE\"
[...]

The operation completed successfully.
```

a. Volumes with a "normal" path correspond to mounted volumes. For example, `C:\Program Files\Delphix` is a normal path. If you see any normal paths in the output, be sure you have disabled all of the VDBs and dSources.

b. Volumes with a path beginning `\\?\` correspond to unmounted persistent volumes. Remove each of them:

```
PS C:\> iscsicli RemovePersistentDevice
"\\?\storage#volume#{bb38add1-d03f-11e1-8767-005056b37fe6}
#0000000008010000#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}"
```

c. Alternatively, if all of the persistent devices are for unmounted volumes, you can remove them all at once with this command:

```
PS C:\> iscsicli clearpersistentdevices
```

4. Remove all of the persistent targets. From the Server Manager\Tools**iSCSI Initiator** configuration tool, use the options available in the **Favorite Targets** tab. or,
   Follow these steps to use iscsicli command-line utility:

a. List persistent targets:

```
PS C:\> iscsicli ListPersistentTargets
```

Remove the appropriate persistent targets. Below is sample output listing the persistent targets:

```
PS C:\> iscsicli ListPersistentTargets
    [...]
    Target Name           : iqn.2008-07.com.delphix:02:02843619-12c4-e4d2-8041-
f5c56a647bc2
    Address and Socket    : 10.43.5.45 3260
    Session Type          : Data
    Initiator Name        : Root\ISCSIPRT\0000_0
    Port Number           : <Any Port>
    Security Flags        : 0x0
    Version              : 0
    Information Specified: 0x20
    Login Flags          : 0x0
```

```
    Username              :
    [...]
```

> Misleading Help for RemovePersistentTarget Command
> The help for iscsicli `RemovePersistentTarget` is misleading:

```
iscsicli RemovePersistentTarget <Initiator Name> <TargetName>
                                <Port Number>
                                <Target Portal Address>
                                <Target Portal Socket>
```

> `<Initiator Name>` and `<Target Name>` show up in the listing and should be taken directly from there. `<Port Number>` can be taken from the listing output, but a `*` should be used if `<Any Port>` is listed. `<Target Portal Address>` and `<Target Portal Socket>` are shortened to `Address` and `Socket` in the `ListPersistentTargets` output. The term `Socket` in both places is what is more typically referred to as a `port`.

c. Use the `RemovePersistentTarget` command to remove the target, as shown in this example:

```
PS C:\> iscsicli RemovePersistentTarget Root\ISCSIPRT\0000_0 iqn.2008-07.com.delphix:0
2:02843619-12c4-e4d2-8041-f5c56a647bc2 * 10.43.5.45 3260
```

5. Log out of any sessions.
   From the Server Manager\Tools**iSCSI Initiator** configuration tool, use the options available in the **Targets** tab to log out. Selected a connected session under **Discovered Targets**, and then click **Disconnect**.
   or, Follow these steps to use the iscsicli command-line utility: List the sessions.

```
PS C:\> iscsicli sessionlist
Session Id : fffffa8003fb0018-4000013700000001
Initiator Node Name : iqn.1991-05.com.microsoft:10-43-1-200.ad.delphix.com
Target Node Name : (null)
Target Name : iqn.2008-07.com.delphix:02:02843619-12c4-e4d2-8041-f5c56a647bc2
[...]
```

b. Log out from the target.

```
PS C:\> iscsicli logouttarget fffffa8003fb0018-4000013700000001
```

6. Change the hostname (IP address) or IQN in the Delphix engine.

   a. If you are changing the hostname, follow the instructions in the Microsoft TechNet[635] article "Rename the Computer."

> Note that if the computer is on a domain, you will need a domain administrator to perform the rename or re-add the computer to the domain depending on the version of Windows it is running.

   b. To use the Delphix Setup application, go to **Manage** → **Environments** screen and select the Target host. To modify the Host Address for the environment, use the Edit (Pencil) icon next to the **Attributes** panel. Update the Host Address to reflect the new IP address, and then use the **Save (Tick)** icon to confirm the change. This will automatically trigger an action to Refresh the Environment. The refresh process includes steps to configure the iSCSI Target and verify that it is reachable.

   c. If you are changing the IQN only, change it through the Microsoft iSCSI Initiator GUI following the instructions in the Microsoft iSCSI User Guide[636].

7. Wait for the computer to finish rebooting.

8. Verify the new IQN in the iSCSI initiator.

> If you are using the default IQN and have changed the hostname (IP address), the IQN should include the new hostname.

9. Refresh the environment on the Delphix Engine.Re-enable the dSources.
10. Re-enable the dSources.
11. Re-enable the VDBs.
12. Using the iscsi cli command-line utility, verify that the sessions on the Windows server are using the new IQN.

```
PS C:\> iscsicli sessionlist
Microsoft iSCSI Initiator Version 6.1 Build 7601

Total of 1 sessions
```

635 http://technet.microsoft.com/en-us/library/dd894434%28v=ws.10%29.aspx
636 https://learn.microsoft.com/en-us/windows-server/storage/iscsi/iscsi-target-server

```
Session Id : fffffa8003f77018-4000013700000004
Initiator Node Name : <NEW IQN>
[...]
```

9.12.5 Linking data sources and syncing data with SQL Server

Creating a dSource will ingest data from the source and create a dSource on the engine. The dSource is an object that the Continuous Data Engine uses to create and update virtual copies of your database. As a virtualized representation of your source data, it cannot be managed, manipulated, or examined by database tools.

This section covers the following topics:

- Linking a dSource from a SQL server: an overview (see page 2139)
- Linking data sources with SQL server (see page 2144)
- Staging push implementation for SQL server (see page 2145)
- Data management settings for SQL server data sources (see page 2158)
- Additional SQL server dSource topics (see page 2160)

9.12.5.1 Linking a dSource from a SQL server: An overview

When linking a dSource from a SQL Server source database, Delphix offers several different methods of capturing backup information:

- Delphix Managed Backups, where the Delphix Engine schedules and initiates the backups from the source database, and captures them
- SQL Server Managed Backups, where the SQL Server source database schedules and initiates backups and the Delphix Engine captures them
    - Full backups
    - Full or differential backups
    - Transaction log backups (with LogSync disabled)
    - Transaction log backups (with LogSync enabled)

Delphix Managed Backups are conceptually a lot simpler to explain, but they can be considered less desirable because they might be viewed as intrusive. SQL Server Managed Backups are explained in two sections:

- the initial load of the dSource from the source database
- subsequently keeping the dSource synchronized with the source database.

> Source databases cannot be in Read-Only mode at the time that a backup is taken. While the snapshot will succeed, attempts to provision VDBs from those snapshots may fail with an error during the provision process.

> Temporarily setting the database to read-write mode, and taking a new backup and snapshot while the database is in in this state, will allow VDBs to be provisioned successfully.

Below is a brief explanation of how these three different modes of operation work.

9.12.5.1.1 Delphix managed backup mode

When the **Enabled** checkbox for Delphix Managed Backups is selected, the Delphix Engine will initiate a COPY_ONLY full backup of the source database for the initial load of the dSource. Thereafter, the Delphix Engine will initiate COPY_ONLY full backups of the source database using the schedule specified by the selected SnapSync Policy. If SnapSync Policy is set to None, the Delphix Engine will not automatically initiate COPY_ONLY backups, but you can initiate them manually using the snapshot (camera) icon.

When the **Force Compression** checkbox for Delphix Managed Backups is selected and the backups are not compressed, the Delphix Engine will take a compressed copy-only full backup of the source database and this will take effect during the next scheduled Snapshot or SnapSync operation.

> COPY_ONLY backup files are written to the Delphix storage that has been mounted on the staging server. Delphix does not require space on the source or staging servers to hold the backup files.

You must also select the Staging Environment and the SQL Server instance onto which the backups will be restored.

9.12.5.1.2 SQL server managed backup modes

9.12.5.1.2.1 Initial load of the dSource

When the Delphix Managed Backups option is left unchecked (*default*), the Delphix Engine will initiate a backup only when the user selects to initiate a COPY_ONLY full backup that the Delphix Engine will use to keep the dSource in sync with the source database.

For the initial load of the dSource, you can choose one of the following:

- have the Delphix Engine initiate a COPY_ONLY full backup
- use the most-recent existing full or differential backup (*default*)
- use a specific existing full or differential backup identified by its `backup_set_uuid`

> Simple Recovery Model

If the source database is using a simple recovery model, using a new full COPY_ONLY backup initiated by the Delphix Engine is not supported for the initial load of a dSource.

After the initial load, you need to select the Backup Paths and tell the Delphix Engine where to look for backups of the source database.

- If **Autodiscover** is selected, the exact path used to take the backup from the source database will be determined by querying by the source database instance. If this option is used, the Delphix Engine should take backups to a UNC path (Windows file share) so that they are accessible to the Staging Server.
- If custom paths are specified, the Delphix Engine will query the source database instance, identify the filename of the source backup, and then recursively search the specified Backup Paths for this filename. These paths should also be UNC paths (Windows file share) which are accessible to the Staging Server.

The path used to restore backups must be readable by the Windows server hosting the staging instance**,** using the staging environment's configured Environment User**.**

The Delphix Engine supports source database backups that SQL Server creates natively, as well as backups created by Quest/Netvault LiteSpeed, Red Gate SQL Backup Pro, Veritas NetBackup, and Commvault. For more information, see the topic SQL Server Support Matrix. (see page 2052)

Once you have decided how the dSource will be initially loaded, select the staging environment and the SQL Server instance onto which the backups will be restored. NetBackup and Commvault backups are not on local storage and therefore the Backup Paths will just be ignored. See Linking a dSource from a NetBackup SQL Server Backup (see page 2167) and Linking a dSource from a Commvault SQL Server Backup (see page 2162) for more information.

The staging instance opens the backup file for reading and may hold a lock on it when restoring the backup on the staging database. A new source database backup, initiated with the **Append to the existing backup set** option, may fail as SQL Server will not be able to open the locked backup file to append a new backup to it.

9.12.5.1.3 Keeping the dSource synchronized with the source database

Next, specify how the Delphix Engine will capture subsequent backups of the source database.

The selected Validated Sync mode determines how often the Delphix Engine will check for new backups, and which type of backups it will check for. You can always force synchronization with the source database by enabling **Validated Sync Mode** from the **Data Management** tab available under the **Configuration** tab for the selected dSource.

9.12.5.1.4 Validated sync and logSync

When you link a source database into the Delphix Engine, a staging database will still be required if the Validated Sync is not enabled, as described in Overview of Setting Up SQL Server Environments (see page 2092). In this process, the Delphix Engine continuously monitors the source database for new full and differential backups if the source database is using a **simple** recovery model, or transaction log backups if using a **full** recovery model. This will also depend on the selected backup mode. When it detects a new backup, it restores that backup to the staging database with the storage residing in Delphix. The result is a TimeFlow with consistent points from which you can provision a VDB, also known as snapshots.

Snapshots accumulate over time. To view a snapshot:

1. From the **Datasets** panel, click the **group** containing the dSource.
2. Select **dSource**.
3. Click the **TimeFlow** tab.

Each snapshot is displayed and includes some information about the captured database along with Snapshot database change number (SCN for Oracle and LSN for SQL Server). You can scroll through these cards to select the one you want, or you can enter a date and time to search for a specific snapshot.

9.12.5.1.5 Summary of validated sync modes

This table summarizes each mode of Validated Sync, displaying how often the Delphix Engine will poll to check for new backup files when it creates snapshots for the dSource, and whether point-in-time restores for provisioning and refreshing virtual databases (VDBs) is possible or not.

Validated Sync Mode	Polling Interval	Snapshot for each FULL backup	Snapshot for each DIFF backup	Snapshot for each TLOG backup	Allows Point-in-time Restores	Notes
Delphix Managed Backups	N/A	N	N	N	N	Takes COPY_ONLY full backups according to SnapSync schedule. For more information on this option, see Delphix as a Backup Solution to SQL Server. (see page 2118)

Transaction log backups (LogSync DISABLED)	1-minute	N	N	Y	N	Log backups are not collected if: • There are gaps in the sequence of log backups (a break in the "log chain") • The available log backups do not include any changes since the last successful Delphix snapshot
Transaction log backups (LogSync ENABLED)	1-minute	N	N	Y	Y	Log files consume additional space on the Delphix Engine and are managed according to the defined retention policy for logs. For NetBackup and Commvault backups, Point-in-time restores are not supported.
Full or differential backups	1-minute	Y	Y	N	N	
Full backups	1-minute	Y	N	N	N	
None	Manual only	N	N	N	N	Only retrieves backups when you initiate a manual snapshot.

Timeflow cards

The Delphix Engine will create a Timeflow card for each backup it restores to the staging server. For example:

- A database in Full backups Validated Sync mode, and daily backup configured on the dSource, would receive one Timeflow card per day
- A database in Full or differential backups sync mode, with one daily backup and two differential backups per day, would receive three Timeflow cards per day

- A database in Transaction log backups sync mode, with a log backup every 15 minutes, would receive 96 TimeFlow cards per day

9.12.5.2 Linking data sources with SQL server

Linking a dSource will ingest data from the source and create a dSource object on the engine. The dSource is an object that the Continuous Data Engine uses to create and update virtual copies of your database. As a virtualized representation of your source data, it cannot be managed, manipulated, or examined by database tools.

For an overview of all dSource related actions, please Managing Data Sources and Syncing Data. (see page 1040)

When linking a dSource from a SQL Server source database, Delphix offers several different methods of capturing backup information:

- SQL Server Managed Backups, where the SQL Server source database schedules and initiates backups and the Delphix Engine captures them
  - Full backups
  - Full or differential backups
  - Transaction log backups (with LogSync disabled)
  - Transaction log backups (with LogSync enabled)
- Delphix Managed Backups, where the Delphix Engine schedules and initiates the backups from the source database, and captures them.

Source databases cannot be in Read-Only mode at the time that a backup is taken. While the snapshot will succeed, attempts to provision VDBs from those snapshots may fail with an error during the provision process.

Temporarily setting the database to read-write mode, and taking a new backup and snapshot while the database is in in this state, will allow VDBs to be provisioned successfully.

9.12.5.2.1 Procedure

1. Login to the **Delphix Management** application.
2. Navigate to **Manage > Datasets.**
3. Click the plus icon and select **Add dSource**.
4. In the **Add dSource** wizard, select the source database with the correct environment user-specified.

5. Select user type for source database authentication and enter the login credentials. Enter username and password for Database user or Domain (Windows) user. For Environment User, select a source environment user from the dropdown list and click **Next**.
6. Enter a name and select a group for your dSource.
   Adding a dSource to a dataset group lets you set Delphix Domain user permissions for that database and its objects, such as snapshots. See the topics under Users and Groups (see page 674) for more information.
7. Select the **Data Management** settings needed. For more information, Data Management Settings for SQL Server Data Sources. (see page 2158)
8. Select the Staging environment and SQL Instance that will be used to manage the staging database used for validated sync of the dSource.
9. Select any policies for the new dSource.
10. Enter any scripts that should be run on the **Hooks** page.
11. Review the dSource Configuration and Data Management information, and then click **Submit**.

9.12.5.3 Staging push implementation for SQL server

This topic provides implementation details of Staging Push for SQL Server. Staging Push eliminates the need to access the customer's production environment and hence the dependency on it. For end-users with various architectural requirements such as an unsupported backup appliance, complex ingestion requirements, and supporting alternate replication methods, Staging Push increases flexibility by ensuring that Delphix can work with most of these unique requirements while maintaining Delphix standard product support.

Staging Push gives end-users control over some Staging DB processes so that the nuanced, staging-based ingestion can be orchestrated externally. It will give control of the staging database to the end-user to pull from any backup provider (as a part of this, you'll be responsible for keeping the Staging DB in sync with the production database). Staging database files will be stored on Delphix Storage. Delphix will still be the one snapshotting the underlying data files, and gathering the metadata required to provision from the snapshot.

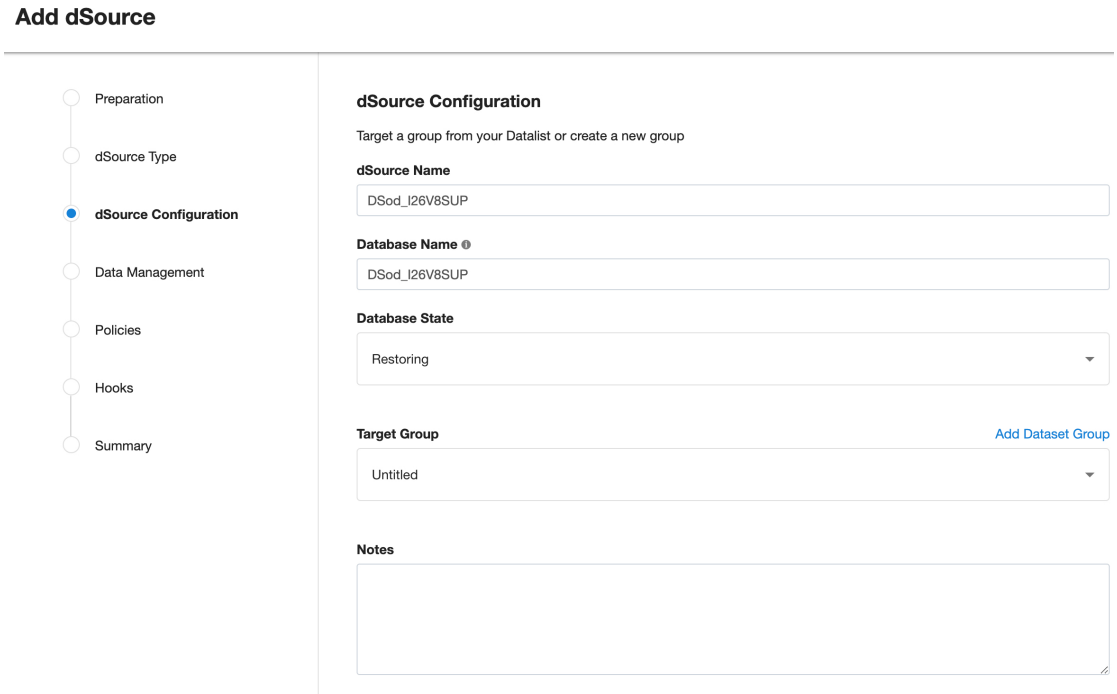
> Source databases cannot be in Read-Only mode at the time that a backup is taken. While the snapshot will succeed, attempts to provision VDBs from those snapshots may fail with an error during the provision process.
>
> Temporarily setting the database to read-write mode, and taking a new backup and snapshot while the database is in in this state, will allow VDBs to be provisioned successfully.

The below steps show how to create a dSource using the Staging Push mechanism.

9.12.5.3.1 Procedure

1. Login to the **Delphix Management** application.

2. Navigate to **Manage > Datasets.**
3. Click the plus icon and select **Add dSource**.
4. On the Preparation tab, click **Next**.
5. From the **dSource Type** tab, select the **MSSQL Staging Push** option and click **Next**.
6. From the **dSource Configuration** tab, enter the following dSource configurations, also shown in the screenshot below, and click **Next**.
   a. **dSource Name** - This name will be available on the Delphix Engine interface.
   b. **Database Name** - This is the name of the staging database that is created on the staging host after linking. It is suggested to use the word "staging" while assigning a name to the staging database as this will help to distinguish it from other end-user databases.
   c. **Database State** - Select one of the following: **Restoring** or **Online**.
      i. **Restoring:** Select this option when you do not need to read or write data on the staging database. You can update the staging database by restoring backups on it.
      ii. **Online:** Select this option when the staging DB is required in online mode so that you can use the database seamlessly. The most common use case for online mode is Azure Managed PaaS database 'Azure SQL Databases'.
   d. **Target Group** - Select the required target group from the dropdown list.
   e. **Notes** - Enter notes if there are any.



7. From the **Data Management** tab, select the staging environment and repository details. The staging environment is where the staging database will be hosted and a repository is a container for the SourceConfigs objects. Each environment in Delphix can contain any number of SQL Server instances and each SQL Server instance can contain any number of databases.

8. Select any policy for the new dSource. SnapSync policy is used as a default policy for taking snapshots. For more details on SnapSync policy, see Policies for Scheduled Jobs[637].[638]
9. Enter any script that should be run on the **Hooks** page. For staging push dSources, a restore backup script can be added as a part of the hooks script to automate backup restore. Refer to the below section for more details.
10. Review the Staging Push dSource Configuration and Data Management information, and then click **Submit**.

9.12.5.3.2 Importing Azure bacpac file

After creating staging push dsource in online mode and before taking a snapshot, if you want to align the staging database with the source database automatically, you can do so by providing the script in the hook.

You can also manually import an Azure SQL database bacpac file on the staging host.

9.12.5.3.2.1 Prerequisites

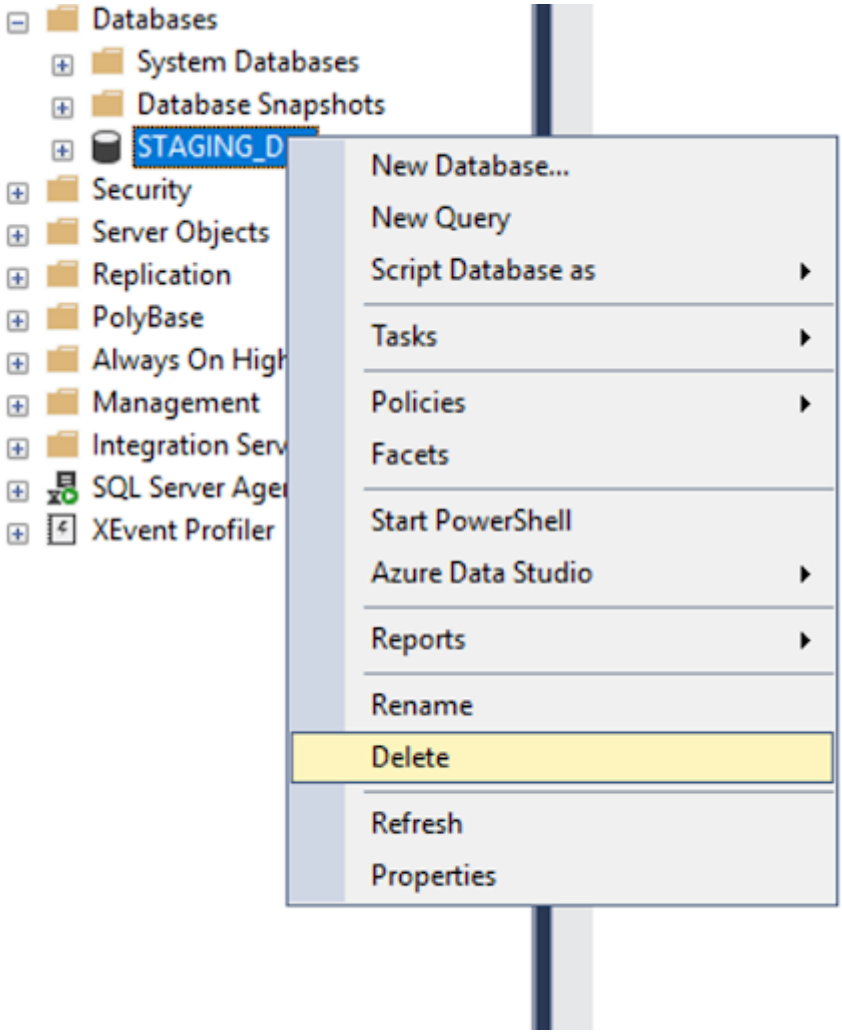
Before importing the bacpac file make sure that all the database files are present at the mounting location.

The mounting location is available as a mounting base under the Configuration tab of the newly created dSource. i.e. staging push dsource needs to be created in online mode it will create an empty dsource along with the mount location.

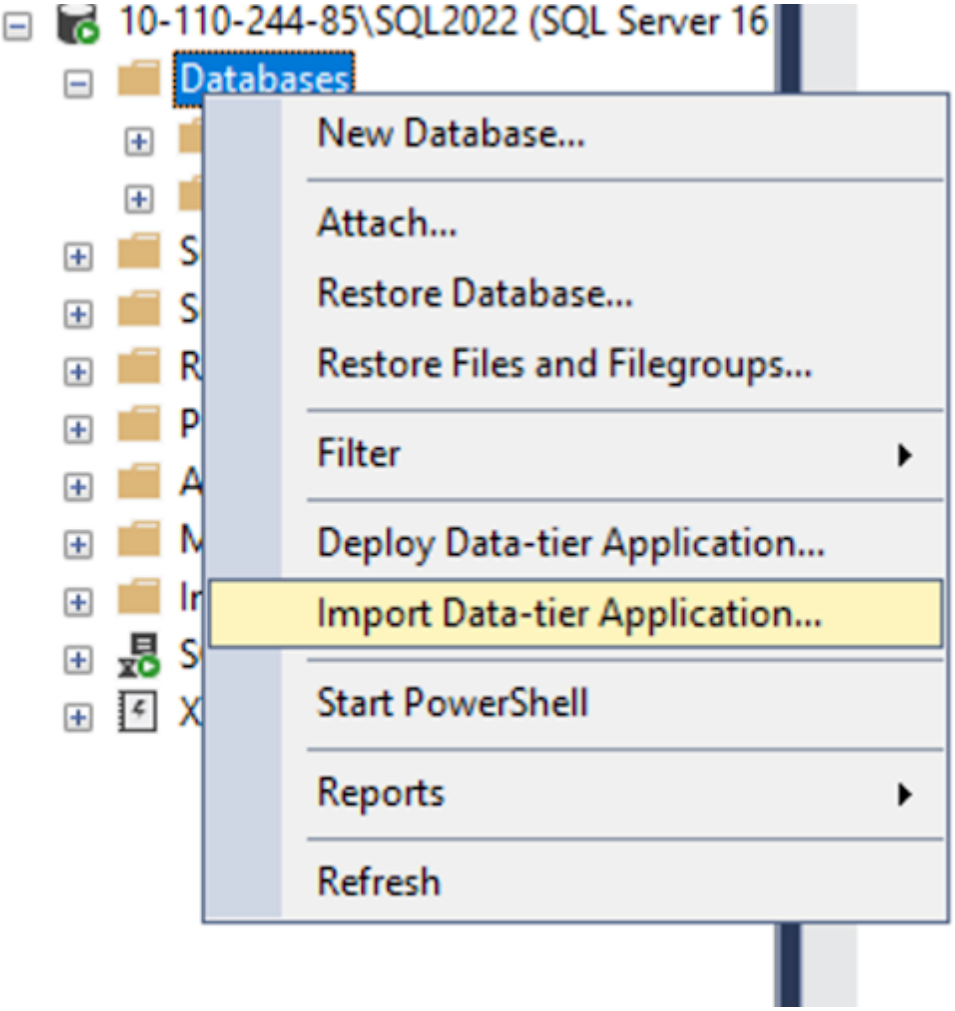
9.12.5.3.2.2 Procedure

1. Extract an Azure SQL backup i.e. bacpac file.
2. Delete the existing staging database from SSMS.
    a. Do not delete dSource on Delphix Continuous Data Engine.

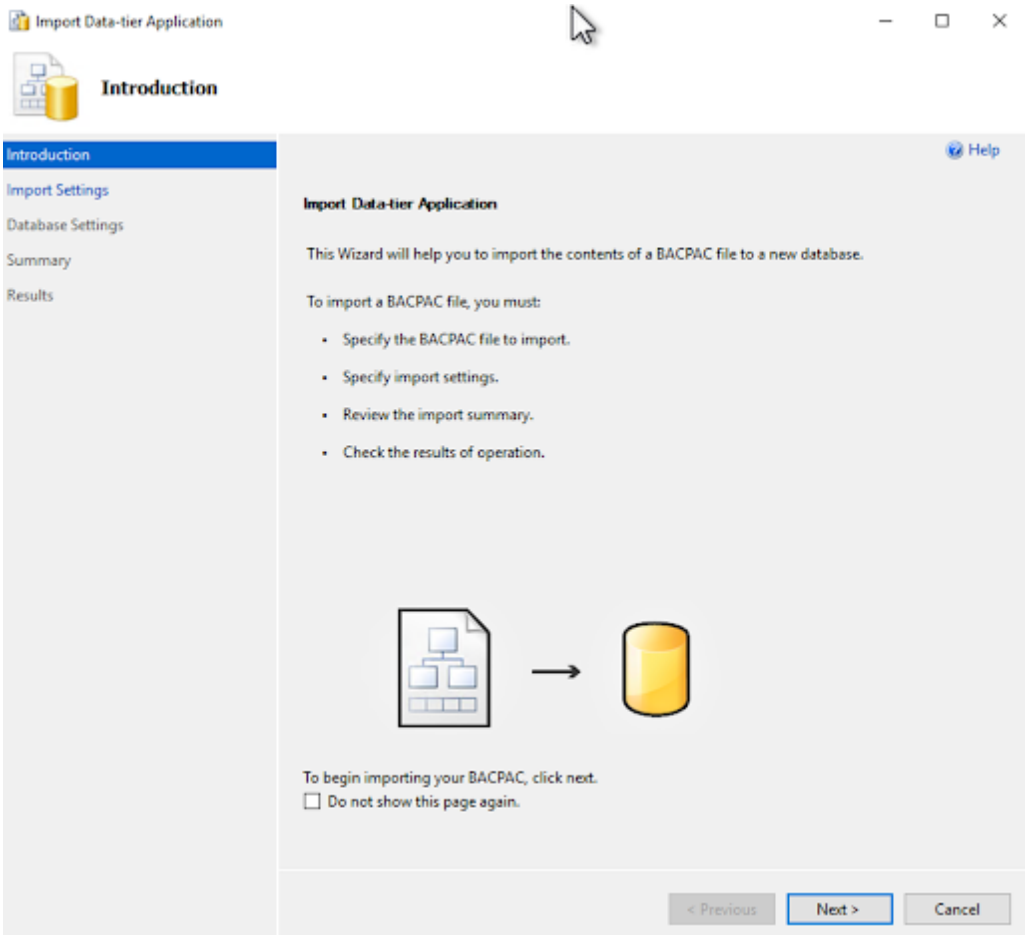
637 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/6230312/Policies+for+scheduled+jobs
638 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/policies-for-scheduled-jobs



3. Once the staging database gets deleted, follow the steps mentioned below:
    a. Create Staging DB for dSource
        i. On Staging SQL Server Instance, right-click on **Databases** & choose **Import Data-tier Application**.



ii. Follow the instructions to import the BACPAC file on the staging server, and hit **Next**.



iii. Choose the option to **Import from Windows Azure**, provide the required details, choose the required bacpac file, and hit **Next**.

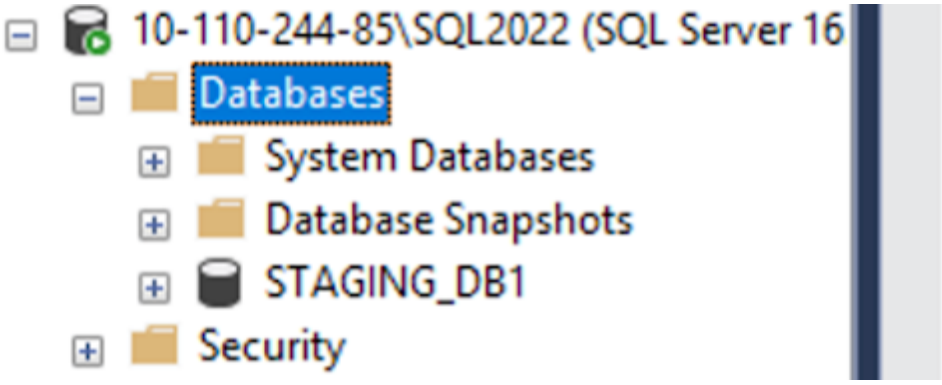
iv. Provide the database name, and under SQL Server Settings, point **Data file path** & **Log file path** to the staging mount created by Delphix, during staging push online dSource creation.

```
<Delphix Connector Path>\<Engine UUID>-Staging-<Container
ID>\DATA\DB
```

v. Verify the settings and click **Finish**.

vi. Wait for the operation to complete.

vii. Once the operation completes, verify the staging database exists on SQL server instance.



9.12.5.3.3 Restoring backups

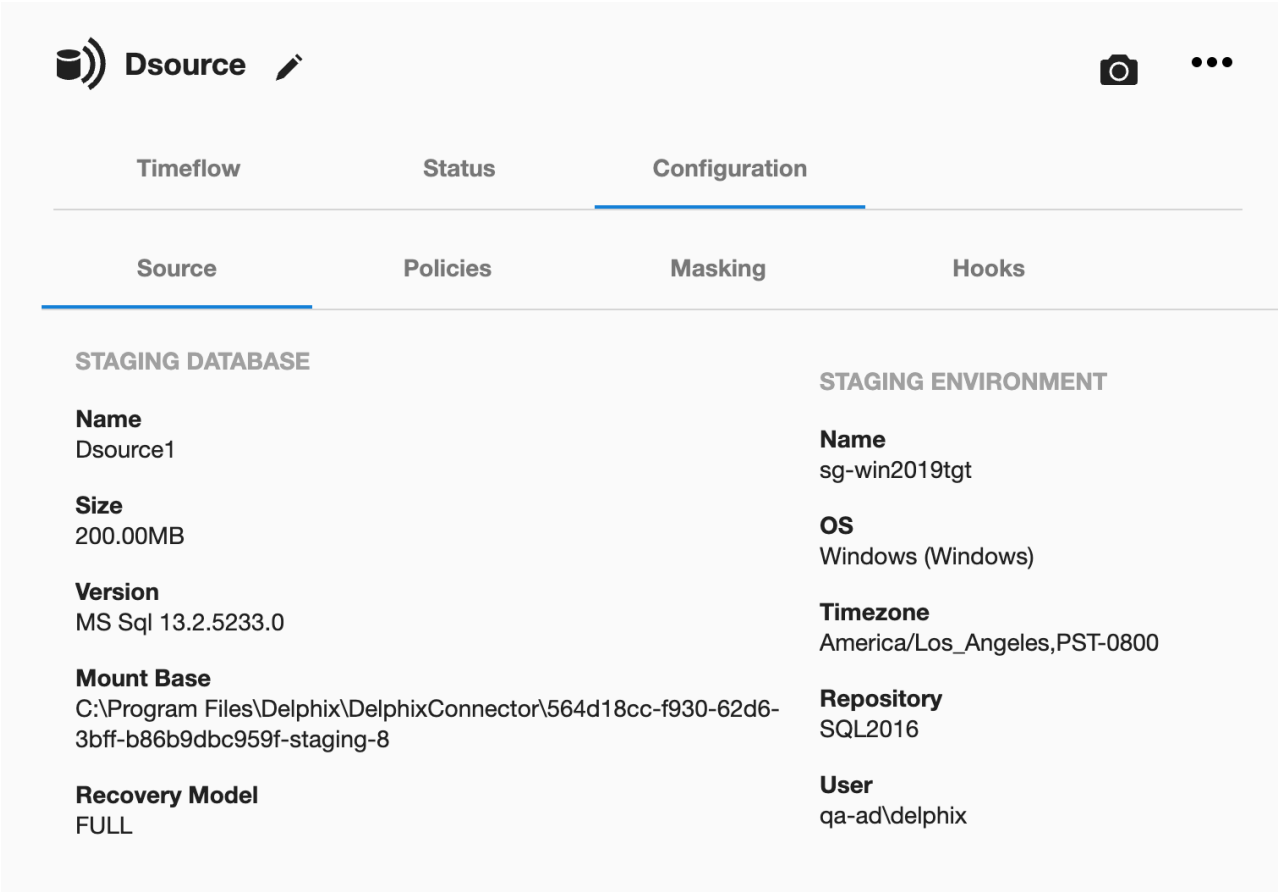
After linking and before taking a snapshot, if the user wants to align the staging database with the source database automatically they can do so by providing the restore script in the hook, as shown below.
They can also manually restore a database backup on the staging host.

> First backup restoration needs to be a full backup with the REPLACE keyword.

9.12.5.3.3.1 Prerequisites

> When restoring the backup, make sure that all the database files are present at the mounting location.

The mounting location is available as **Mount Base** under the **Configuration** tab of the newly created dSource.



9.12.5.3.3.2 Procedure

Manually restore a backup on the staging host or provide the restore script in the hook script. A sample pre-sync hook for native backup is shown below. It also lists the following keywords.

- NORECOVERY - Keeps the database in restoring state.
- REPLACE - Overwrites existing database created during linking with whichever source database that is in the backup set and is getting restored.
- MOVE - Restores the data and log file to the specified locations.
- BACKUP_FILE_LOCATION - The location of the backup file of the source database from where the backup is restored and it should be accessible from the staging host. For example, \ \10-43-89-18\Backup\sp.bak.

Sample hook to illustrate restore script usage for restoring backups

```
#
# Copyright (c) 2022 by Delphix. All rights reserved.
#
#Set-Variable UTF8_CODEPAGE 65001 -option readonly
#Set-Variable UTF8_ENCODING "System.Text.UTF8Encoding" -option readonly
# Uncomment the following log to turn on debugging
# Set-PSDebug -Trace 2;
```

```
$DSOURCE_HOST = $env:STAGING_INSTANCE_HOST
$DSOURCE_PORT = $env:STAGING_INSTANCE_PORT
$DSOURCE_INSTANCE = $env:STAGING_INSTANCE_NAME
$CONNECT_STRING = "$DSOURCE_HOST\$DSOURCE_INSTANCE,$DSOURCE_PORT"
$DSOURCE_NAME = "$env:STAGING_DATABASE_NAME"
$DataDbFilePath = "$env:STAGING_MOUNT_BASE"
$SQL_SCRIPT= "RESTORE DATABASE $DSOURCE_NAME FROM DISK = '\\BACKUP_FILE_LOCATION'
WITH NORECOVERY, REPLACE, MOVE N'<Source_db_name>' TO '$DataDbFilePath\DATA\db\
$DSOURCE_NAME.mdf', MOVE N'<Source_db_name>_log' TO '$DataDbFilePath\DATA\db\" +
$DSOURCE_NAME + "_log.ldf'"
function die {
    Write-Error "Error: $($args[0])"

    # run exit handler, if defined
    if (Get-Command -type Function -name atExit 2> $null) {
        atExit
    }
    exit 1
}

function verifySuccess {
    if (!$?) {
        die "$($args[0])"
    }
}

### Restore database.

echo $SQL_SCRIPT
Sqlcmd  -b -S $CONNECT_STRING -U sa -P <PWD>  -Q $SQL_SCRIPT
verifySuccess "Failed to restore backup"

# Uncomment the following line to turn off debugging
# Set-PSDebug -Trace 0;

exit 0
```

9.12.5.3.4 Snapshot

The first snapshot is created as a part of dSource creation and contains data and log files within the Delphix created mount point.

> The first snapshot created is of an empty database and does not contain any source database's data. (Unless a source backup was restored via a pre-sync hook. In that case, the initial snapshot will be that of source backup and won't be empty).

9.12.5.3.4.1 Prerequisites

- The database should be present on the staging host.
- The DATA directory should be mounted.
- The DB files should be present on Delphix mounted DB directory. For example, `C:\Program Files\Delphix\DelphixConnector\ec2197b2-e0c6-48d2-bd14-265e6fa9b5ab-staging-1\DATA\db`
- The database should be in Restoring state.
- No other restore operation should be in progress on the staging database.

9.12.5.3.4.2 SnapSync criteria

- Delphix tries to fetch the last restored backup. If no backup is found, the snapshot is skipped with a warning.
- If a snapshot already exists for the backup in the current timeflow, the snapshot is skipped with a warning.

> Snapshots display the Staging Host timezone, as opposed to Linked dSources, where snapshots display the Source Host timezone. Here, the Staging timezone is displayed for Staging Push as we don't have the Source host to fetch the timezone information. This functioning might change in the future.

Perform the following steps to take a snapshot:

1. Login to the **Delphix Management** application.
2. Click **Manage** and select **Datasets** from the dropdown list.
3. Select the dSource created using the Staging Push mechanism to Snapshot.
4. Click the **Camera** icon.
5. From the Snapshot dialog box, select **Yes**.
6. Navigate to the **Timeflow** tab and click **View: All snapshots** to verify the Snapshot you just created. You can now proceed to provision the VDB using the snapshot.

9.12.5.3.5 Disabling and enabling the dSource

When a disabled dSource is later enabled, it will resume communication and incremental data updates from the staging database according to the original policies and data management configurations that you set.

9.12.5.3.5.1 Procedure

Disabling a dSource will stop further operations on the Delphix Engine related to the staging dSource.

1. Login to the Delphix Management application.
2. Click **Manage**.
3. Select **Datasets**.
4. Select the **dSource** you want to disable.
5. In the upper right-hand corner, from the **Actions** menu (...) select **Disable.**
6. In the Disable dialog select **Disable**.

When you are ready to enable the dSource again, from the Actions menu (...) select **Enable**, and the dSource will continue to function as it did previously.

Keep a note of the following:

- If you want to import a new .bacpac file after disabling and enabling it, you will have to manually delete the existing staging database and then import the .bacpac file. Alternatively, you can unlink and relink the dSource with a different staging database name, and follow the same procedure as creating a staging database for dSource.
- During enable, if there exists a database with the same staging database name as the user entered and it is not Delphix managed, or if it is but Delphix Engine is unable to drop it, the enable operation will not be terminated.
- Storage mounting will be attempted for the dSource and a new job event `SKIP_ATTACH_OPERATION` will be displayed. This means the job is updated with a warning and attach operation is skipped. The enable operation will be marked as successful.
  Note : You can drop the database that is outside Delphix or that wasn't dropped and try again the disable/enable option.
- Delphix will attempt to mount the staging database's storage during the enable job. Even if the attach job fails, the storage will remain mounted (if Delphix was able to successfully mount it) and the enable job will be marked as successful.

9.12.5.3.6 Unlink(Detach)/link(attach) a dSource

Each dSource contains metadata that associates it with the staging database, as well as the data it has ingested from the staging database in the form of snapshots up to that point. It is possible to detach, or unlink, a dSource from its staging database.

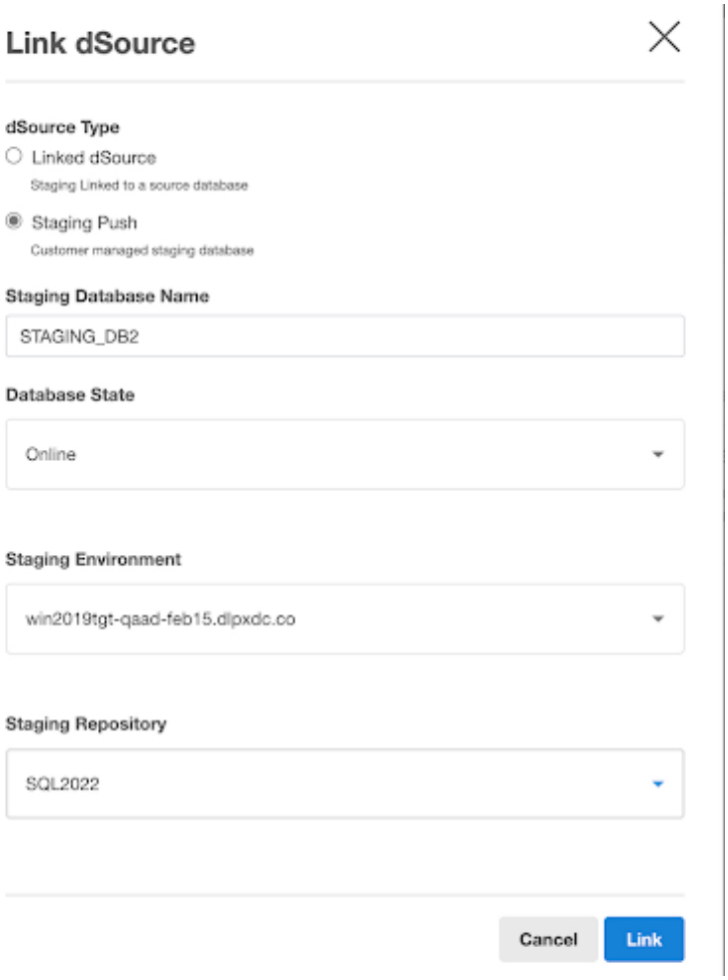
- A detached dSources can still be used to provision a virtual database (VDB).
- You can re-link the staging push dSource with a different staging database name than before. In that case, the staging database will be created with the new name provided. However, DB file names will remain the same as before.
- (Applicable for online mode only) Delphix Continuous Data Engine supports converting existing Linked dSource to Staging push dSource and vice versa.
- If a database changes from a supported state to an unsupported state, a fault will be generated.

9.12.5.3.6.1 Unlinking or detaching a dSource

1. Login to the **Delphix Admin** application.
2. Click **Manage**.
3. Select **My Datasets**.
4. Select the **database** you want to unlink.
5. From the Actions ... tab, click **Unlink.**
6. Click **Yes** to confirm.

9.12.5.3.6.2 Attaching a previously detached dSource

1. Login to the **Delphix Admin** application.
2. Click **Manage**.
3. Select **My Datasets**.
4. Select the **database** you want to link.
5. From the Actions ... tab, click **Link dSource**. Select **Staging Push** and enter **staging database Name**, select **Database State**, **Staging Environment** and **Staging Repository**.



6. Click **Link to confirm.**

Keep a note of the following:

- While unlink/link database state conversion is not supported. The attach job will be successful, but you will not be able to enable the dSource, as you will get the database state changed exception. Therefore, you should always provide the same database state that was given while creating the dSource.
- If staging push dsource is created in the online mode, then the staging database states other than Online such as Restoring, ReadOnly, StandBy, and Emergency are not supported. If a database is in any of these unsupported states, dSource will go into the cannot monitor state on the Delphix Continuous Data Engine interface, and sync operation will fail.

9.12.5.3.7 Limitations

1. Restore backups fails on staging database due to mount issues. When a critical threshold is enforced due to Delphix storage and later resumed successfully, a transactional log will not be applicable on the staging database directly. A generic warning will be displayed for the resume job only for staging

push dSources on the Delphix interface that after the mount, the user might face issues in restoring a transaction log. SQL server error when restoring the Tlog after resume:

`Msg 4320, Level 16, State 3, Line 13`

The file "`SourceDB4_log`" was not fully restored by a database or file restore. The entire file must be successfully restored before applying this backup set.

`Msg 3013, Level 16, State 1, Line 13`

RESTORE LOG is terminating abnormally.

RESOLUTION: Full or differential backup will be successfully restored.

2. Attach operation fails during the Enable operation.
   If the attach operation fails for staging push dSource, any exception is consumed and the enable operation is marked as a success.

   For example, if DB files were changed for the staging database but a snapshot was not taken on the Delphix engine and disable was performed directly, the enable will succeed with an attach failure that will be shown as a warning. The previous DB files will be available on the mount path.

   RESOLUTION: The customer can apply a full restore at this point to bring the database in restoring mode before trying to run a sync.
3. Point-in-Time provisioning is not supported.
4. The sub-directory structure for DB files is not supported.
   For example, If a staging database is restored with DB files inside another folder within the DB directory (`C:\Program Files\Delphix\DelphixConnector\ec2197b2-e0c6-48d2-bd14-265e6fa9b5ab-staging-1\DATA\db\folder1\sp.mdf, C:\Program Files\Delphix\DelphixConnector\ec2197b2-e0c6-48d2-bd14-265e6fa9b5ab-staging-1\DATA\db\folder1\folder2\sp_log.ldf`), then operations such as to enable, relink, VDB provisioning, and export will fail.
5. The validated sync process is not supported for Staging Push dSources.
6. If staging push dsource is created in the restoring mode, then the staging database states other than Restoring such as ReadOnly, StandBy, and Online are not supported. If a database is in any of these unsupported states, dSource will go into the Cannot Monitor state on the Delphix Engine interface, and Sync operation will fail.
7. We have limited the online dSources to follow a single timeflow, i.e. new timeflows will not be created when the recovery fork guide is changed. After restoring a full backup, you will have to delete old snapshots before using the new snapshot for provisioning.

9.12.5.4 Data management settings for SQL server data sources

Each dSource has its own data management settings, which can be configured during the linking workflow as well as in the configuration page for that dSource.

You can configure data management settings to improve overall performance and match the needs of your specific server and data environment.

The following settings are available for SQL Server data sources:

Setting	Explanation
Managed Backups	When enabled, the Delphix engine will take full backups of the database, per the dSource's SnapSync policy, and validated sync will be disabled. Existing backups cannot be used to synchronize the dSource when backups are managed by Delphix.
Recovery Model	The current recovery model of the source database. Three recovery models exist simple, full, and bulk-logged. This is not configurable from Delphix but can be changed within the source's database settings.
Initial Load	1. Delphix will take a copy-only full backup of your source database. 2. Use the most recent full or differential backup (default). 3. Use a specific full or differential backup.
Backup Paths	These are the locations where Delphix will be looking for backups for ingestion. Select Autodiscover to have the Delphix Engine automatically locate the backups by querying the msdb database in the SQL instance. Otherwise, for each path, please specify the top level of a particular backup path since the Delphix Engine will recursively search for the backup file in all subdirectories beneath the specified path.
Validated Sync Mode	Determines the types of backups validated sync will use to generate snapshots. 1. Transaction log backups. a. LogSync adds log files from the source database to the dSource, allowing you to provision a virtual database (VDB) from a specific point in time or LSN for SQL Server databases. 2. Full or Differential backups. 3. Full backups. 4. None.
Staging Environment	This environment will host the staging database used for validated sync.

Setting	Explanation
Repository	A repository is a container for the SourceConfigs objects. Each environment can contain any number of repositories, and repositories can contain any number of source configurations. A repository typically corresponds to a database installation. Whereas source configurations typically correspond to the databases.
Encryption Key	The encryption key to be used when restoring encrypted backups. If the source database is backed up using LiteSpeed or RedGate with password-protected encryption, you can supply the encryption key that the Delphix Engine should use to restore those backups.
NetBackup Ingestion	Enables ingestion from a NetBackup source.
Commvault Ingestion	Enables ingestion from a Commvault source.

9.12.5.5 Additional SQL Server dSource topics

This section covers the following topics:

- Changing the staging target environment for a SQL Server dSource (see page 2160)
- Upgrading a dSource after a SQL Server upgrade (see page 2161)
- Linking a dSource from a commvault SQL Server backup (see page 2162)
- Linking a dSource from a netBackup SQL Server backup (see page 2167)
- Restoring SQL backups stored in Azure cloud storage (see page 2174)
- Working with SQL Server snapshots (see page 2177)
- Detaching and re-attaching SQL Server dSources (see page 2178)

9.12.5.5.1 Changing the staging target environment for a SQL Server dSource

This topic describes how to change the staging target environment for a SQL Server dSource.

9.12.5.5.1.1 Prerequisites

The dSource for the staging database must be disabled before you can change the staging target environment. Follow the steps in *Enabling and Disabling SQL Server dSources* in Managing Data Sources and Syncing Data (see page 1040) to disable the dSource.

9.12.5.5.1.2 Procedure

1. Click **Manage**.
2. Select **Datasets**.

3. Select the **dSource** for which you want to change the staging target environment.
4. Click the **Configuration** tab to view the **Staging Environment**.
5. Click the **Pencil** icon next to **Staging Environment**.
6. Edit the target server and the SQL Server instance on the server to use for staging.
7. Click the **Check** icon to save your changes.

9.12.5.5.2 Upgrading a dSource after a SQL Server upgrade

This topic describes how to upgrade dSources after a SQL Server database upgrade.

There are two ways to upgrade a Source database:

- Perform an Upgrade installation of SQL Server, upgrading the Source and Staging SQL Server instances in-place
- Perform a fresh installation of the new SQL Server version, and migrating the databases to the new instance

The steps required to support this in the Delphix Engine are different depending on method chosen. The required steps are outlined in the sections below.

9.12.5.5.2.1 In-place SQL server upgrade

1. **Disable** all dSources on the instance being upgraded
2. Following Microsoft's procedures, perform an upgrade of the Source SQL Server instance
3. Following Microsoft's procedures, perform an upgrade of the Staging SQL Server instance to the same version as the Source
4. Refresh the Source and Staging environments in the GUI
5. **Enable** the dSources being upgraded

9.12.5.5.2.2 Migrate databases to newer instance

Prerequisites for database migration

- Record the configuration data for each dSource being upgraded, including the Database User, Database Password (if applicable), and Validated Sync configuration. This will be needed to re-link the dSource.
- Following Microsoft's procedures, install a new Source SQL Server instance with the new SQL Server version.
- Ensure that a Staging SQL Server instance is available, running the same SQL Server version as the new Source SQL Server instance.
- **Add** or **Refresh** the Environment containing the new SQL Server instance(s), using the steps in Adding a SQL Server Source Environment (see page 2110) and Adding a SQL Server Standalone Target Environment. (see page 2111)

> The refresh/rediscover operations do not affect the operations of any dSources or VDBs on the environment.

Migration steps

1. Navigate to the **Manage** → **Datasets** screen.
2. For each dSource being upgraded:
   a. Select **Unlink dSource** from the **Actions menu (...)**
3. Following Microsoft's procedures, migrate the Source database to the new Source SQL Server instance.
4. Commence a new **Full** database backup of each upgraded Source database
5. From the **Manage** → **Environments** screen, **Refresh** the new Source Environment. The migrated databases should be detected and visible from the Environment's **Databases** tab.
6. Navigate to the **Manage** → **Datasets** screen.
7. For each dSource that was upgraded:
   a. Select **Link dSource** from the **Actions menu (...)**
   b. Locate the upgraded Source database using the Source Environment, **Installation and Database** drop-down boxes
   c. Select a compatible Staging Environment and Staging Repository
   d. Configure the **Database Authentication** using values that are appropriate for the new Source database
   e. Click **Link** to begin linking the dSource
   f. Reconfigure any dSource settings using the dataset's **Configuration** tab
   g. Use the **Snapshot** button to take a snapshot using the upgraded database backup

9.12.5.5.3 Linking a dSource from a commvault SQL Server backup

9.12.5.5.3.1 Customer requirements:

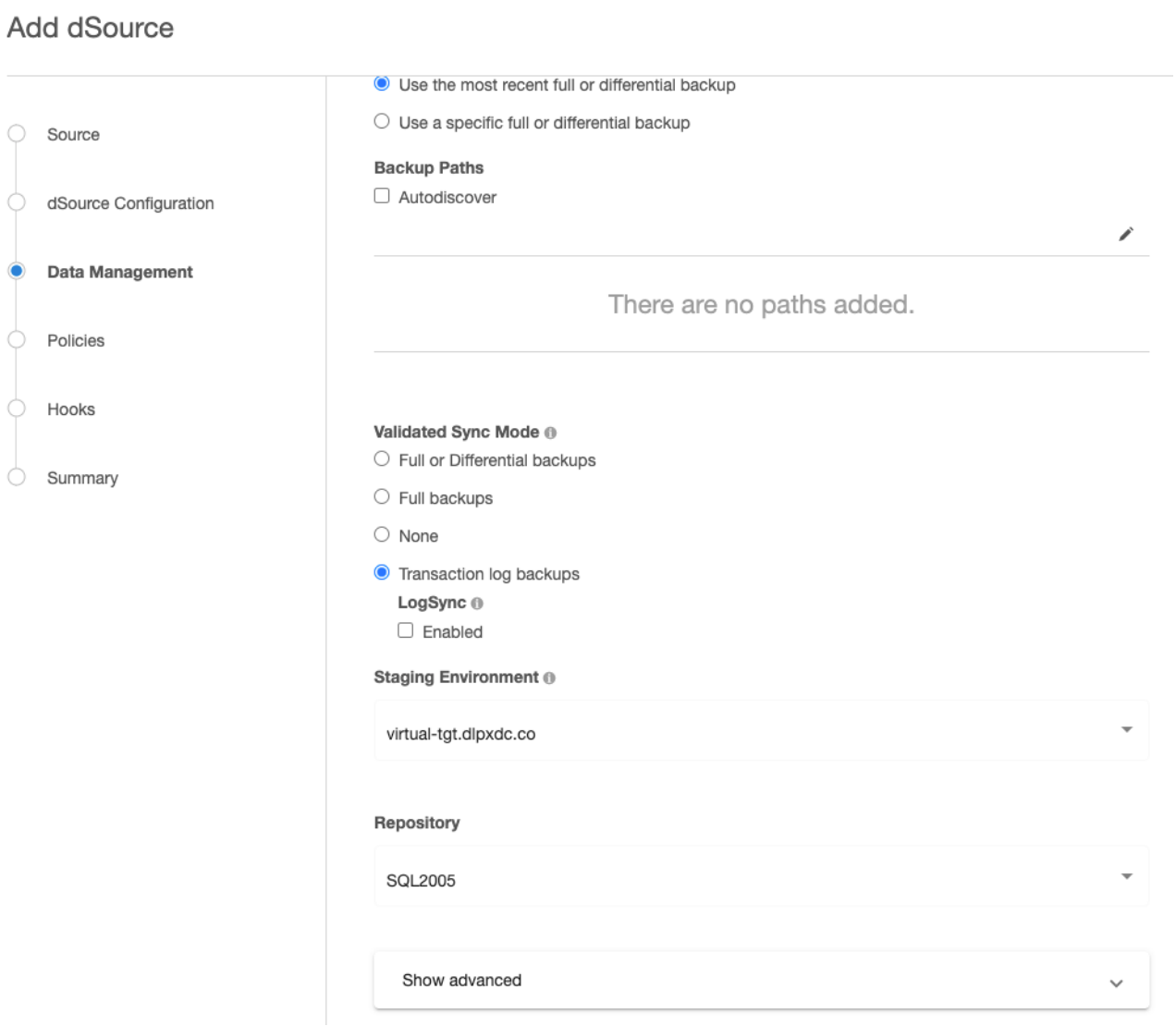
- Delphix currently supports Commvault v11. The version of Commvault SQL Agent on the staging environment must be the same as that on the source.
- The TCP port 8415 must be open from the staging host to the Delphix Engine.
- If the dSource is backed up with Commvault, the source and staging environments must each have the SQL Agent installed.
- Both SQL Agents (on source and staging) should be registered with Commvault Server mentioned during linking.
- The install path's 'Base' directory of Commvault SQL Server Agent on the staging host, must be part of the PATH environment variable as we need to access the Commvault CLI. This is typically located at <Commvault install path>\Base.

    - There are two PATH environment variables. One is for the current logged in user and the other is a global System variable. Since the Delphix Connector runs under the "Network Service" account and spawns the Commvault commands, changes need to be done to the System variable as well.
    - After the Commvault binaries are added to both the system and user PATH environment variables, the Delphix Connector service must be restarted in order for the new process to reflect the changes made to the PATH environment variable.
    - After making the changes to the PATH, login as the Delphix operating system user and try running "qlogin -sso -gt" to confirm the user can execute Commvault commands and authenticate to the Commvault server.
- Since SSO is used from Commvault CLI to login to the CommServe server from the staging environment, SSO should be enabled.
- Active Directory domain should be configured and configured staging environment user should be given required permissions to restore the database on staging client.
- In the CommServe server, the staging client should be configured with a user who has required permissions to restore the database on the staging environment. User account configuration can be done as explained in Commvault documentation at User Account Configuration for the SQL Server Agent[639].
- Only transaction logs, incremental, and database full backups are currently supported.
- For linking a dSource from a Commvault SQL Server backup, Delphix environment user should be part of the SQL Server sysadmin group. See, Staging database login requirements section of Overview of requirements for SQL Server environments (see page 0) for more details.

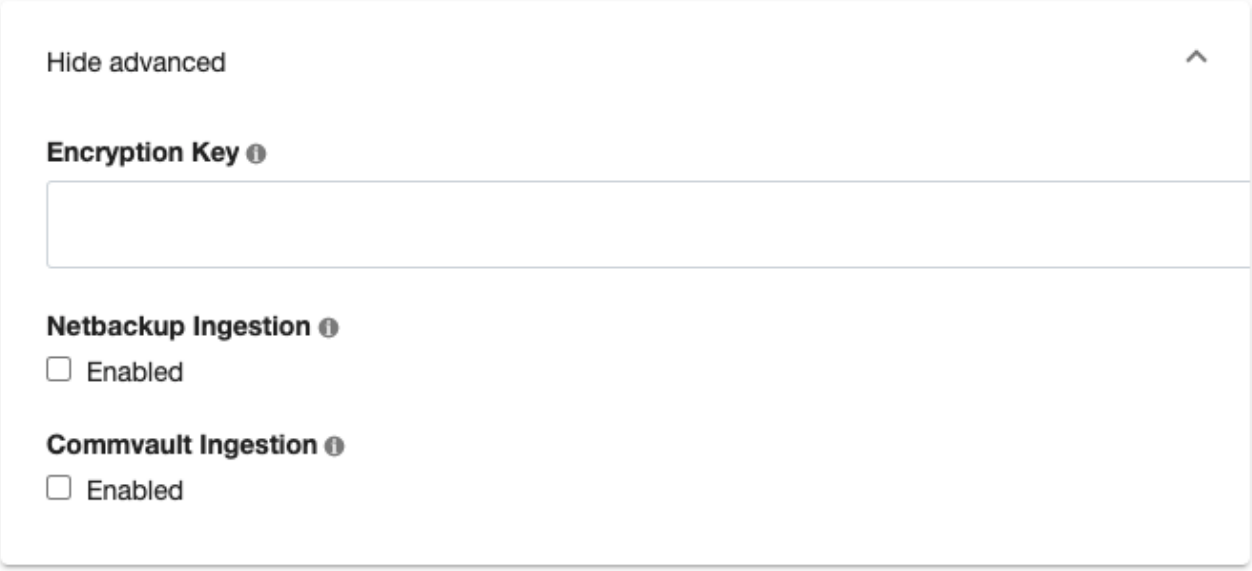
9.12.5.5.3.2 Linking with commvault backups:

To link a dSource and use Commvault, follow the steps to add the environments as earlier, and make sure all the requirements listed above are met. Going through the linking wizard on the **Data Management** page, select **Show advanced**.

639 http://documentation.commvault.com/commvault/v11/article?p=18202.htm



Enable **Commvault Ingestion** and input the CommServe hostname and, source and staging client names.



We validate whether the source and staging clients exist in the CommServe server and the Commvault SQL Agent on staging client is registered to the CommServe server. To run the validation commands following login command with SSO option is used to login into CommServe Server:

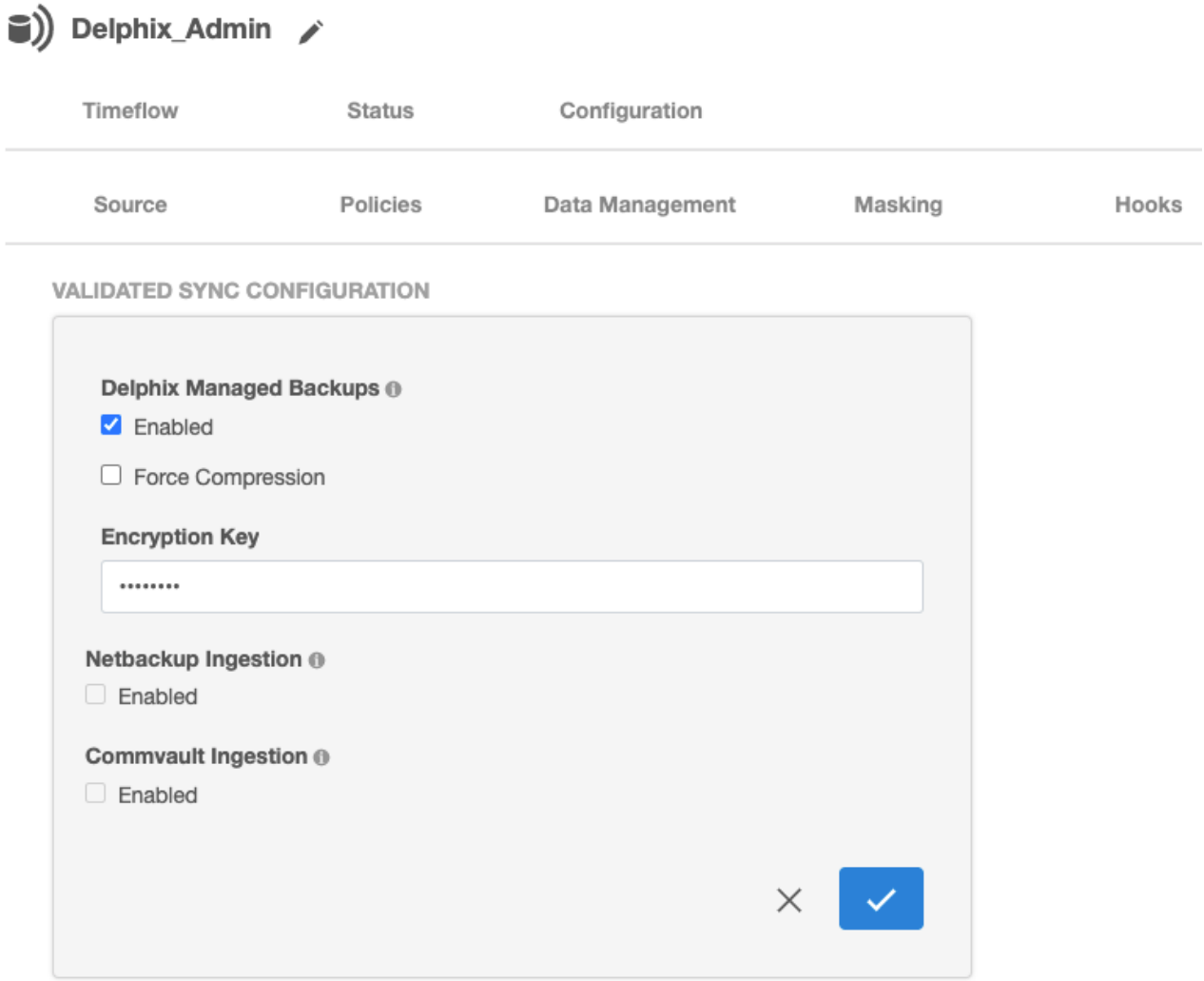
```
qlogin -sso -gt
```

9.12.5.5.3.3 LogSync for SQL server dSources

Logsync (point-in-time provisioning) is currently not supported for Commvault transaction logs. However, LogSync can still be enabled if Commvault ingestion is enabled. LogSync for backups taken with other backup providers that support LogSync will work as before.

9.12.5.5.3.4 Enabling commvault for previously created dSources

On an already created dSource go to **Configure** > **Data Management** and click edit (pencil button). Enable Commvault Ingestion and add all the configuration as necessary.



9.12.5.5.3.5 Linking an availability group database with commvault

Linking with an Availability Group (AG) source works similar to as described above, however, the source client name provided in the Commvault configuration should be the MSSQL AG Client name as in the CommServe server.

9.12.5.5.3.6 General notes/troubleshooting

- Commvault backups are not on local storage and therefore while ingesting backups taken using Commvault, Backup Paths provided in configuration will be ignored.

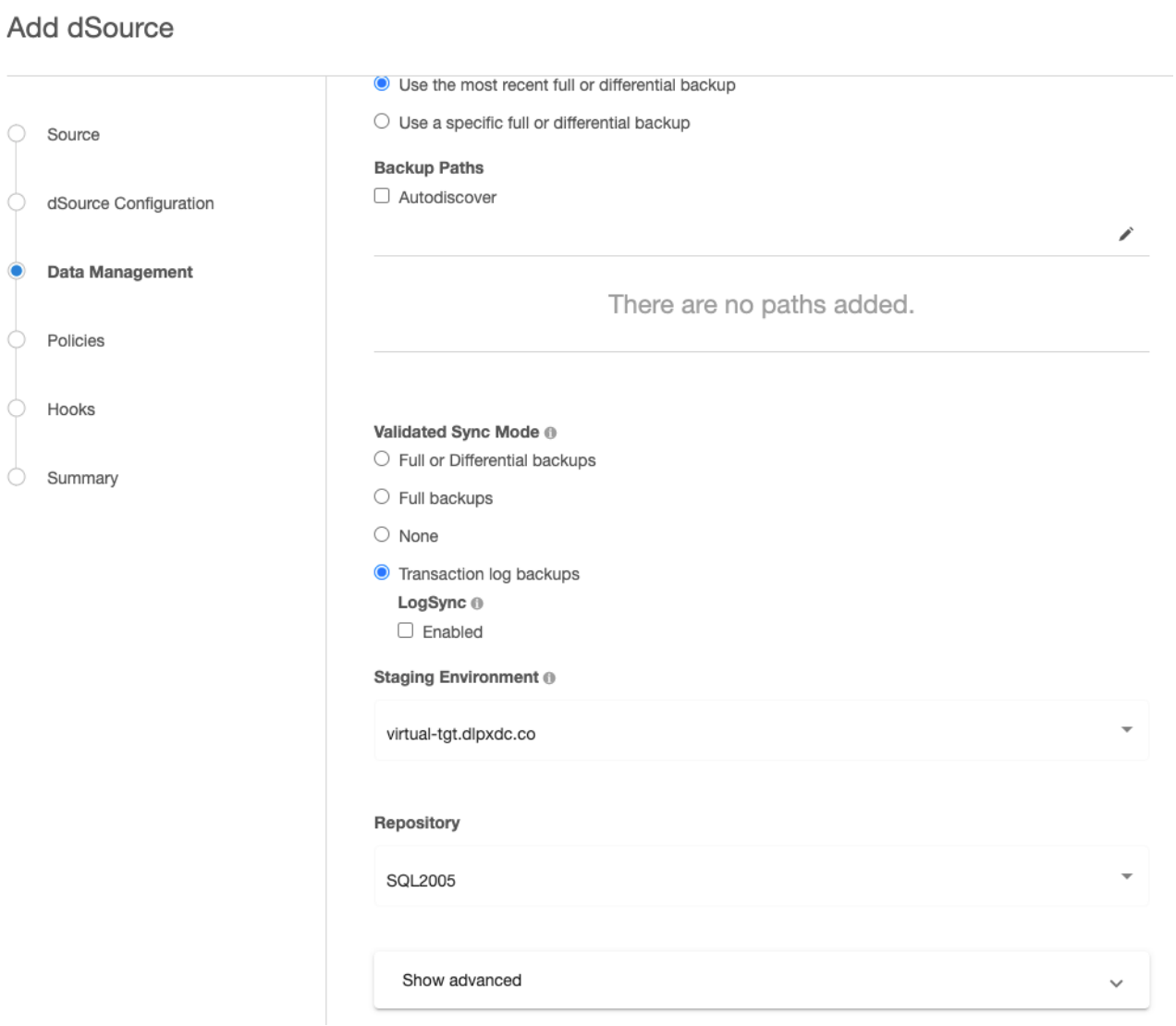
9.12.5.5.4 Linking a dSource from a netBackup SQL Server backup

9.12.5.5.4.1 Customer requirements

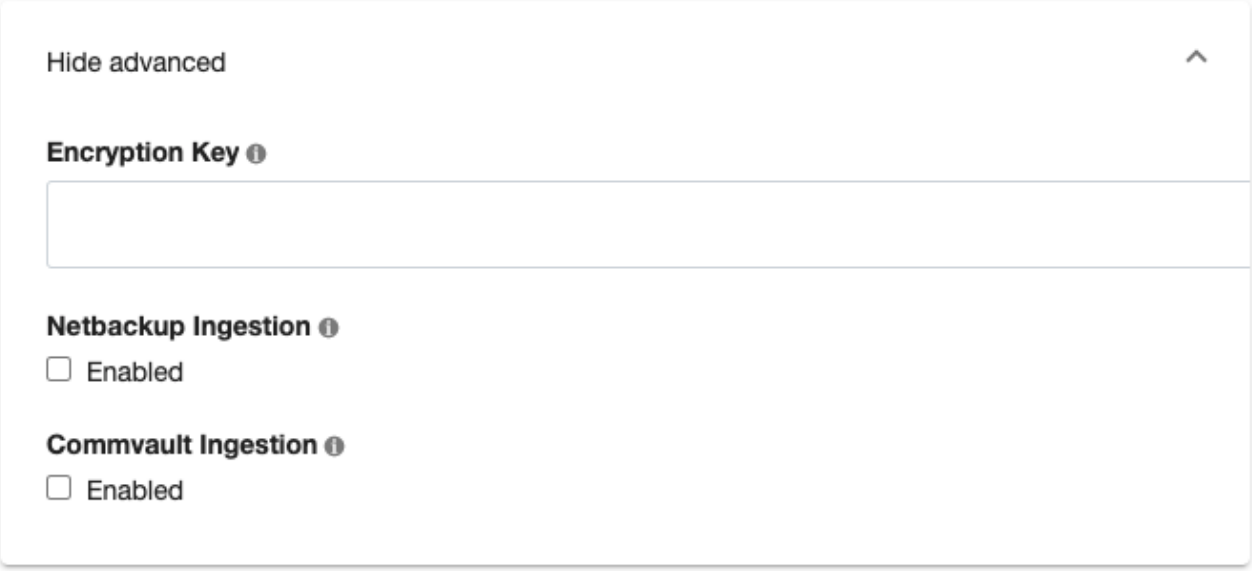
- The version of NetBackup client on the staging environment must be the same as that on the source.
- The TCP port 8415 must be open from the staging host to the Delphix Engine
- Backups must be taken via an MS-SQL-Server type policy with an INSTANCE client list type. (value is 15, and 1 respectively) Only transaction logs, incrementals, and database full backups are currently supported.
- The master server and source client servers' clocks must be within a minute of each other (timezones can be different).
- Any backups that Delphix needs to ingest for a dSource must be taken to one NetBackup master server using one NetBackup SQL Server client. A multiple master server setup for one dSource is not supported.
- If the dSource is backed up with NetBackup, the source and staging environments must each have the NetBackup client installed.
- Both clients (on source and staging) during installation must be setup with the master server and the source and staging instances must be registered.
- The install path's bin directory of the SQL Server NetBackup client on the staging host must be part of the system PATH as we need access to dbbackex.exe and bplist.exe. This is typically located at
- If you have modified the PATH, then please restart the Delphix connector service otherwise the linking process won't be able to pick the changed system PATH environment variable.
- Configure redirected restores on the master server between the source NetBackup Client and the staging NetBackup Client
    - https://www.veritas.com/support/en_US/doc/17221771-126559330-0/v113535700-126559330

9.12.5.5.4.2 Linking with netBackup backups

To link a dSource and use NetBackup, follow the steps to add the environments as before, and make sure all requirements listed above are met. Going through the linking wizard on the **Data Management** page, select **Show advanced**.



Enable **Netbackup Ingestion** and input the master and source client names.



We use these names to query bplist using these options:

> When validating the master and client servers we expect that at least one NetBackup MS-SQL-Server type backup had been taken with this pair in the last two days. Validation will fail otherwise.

9.12.5.5.4.3 Using NetBackup config parameters or templates

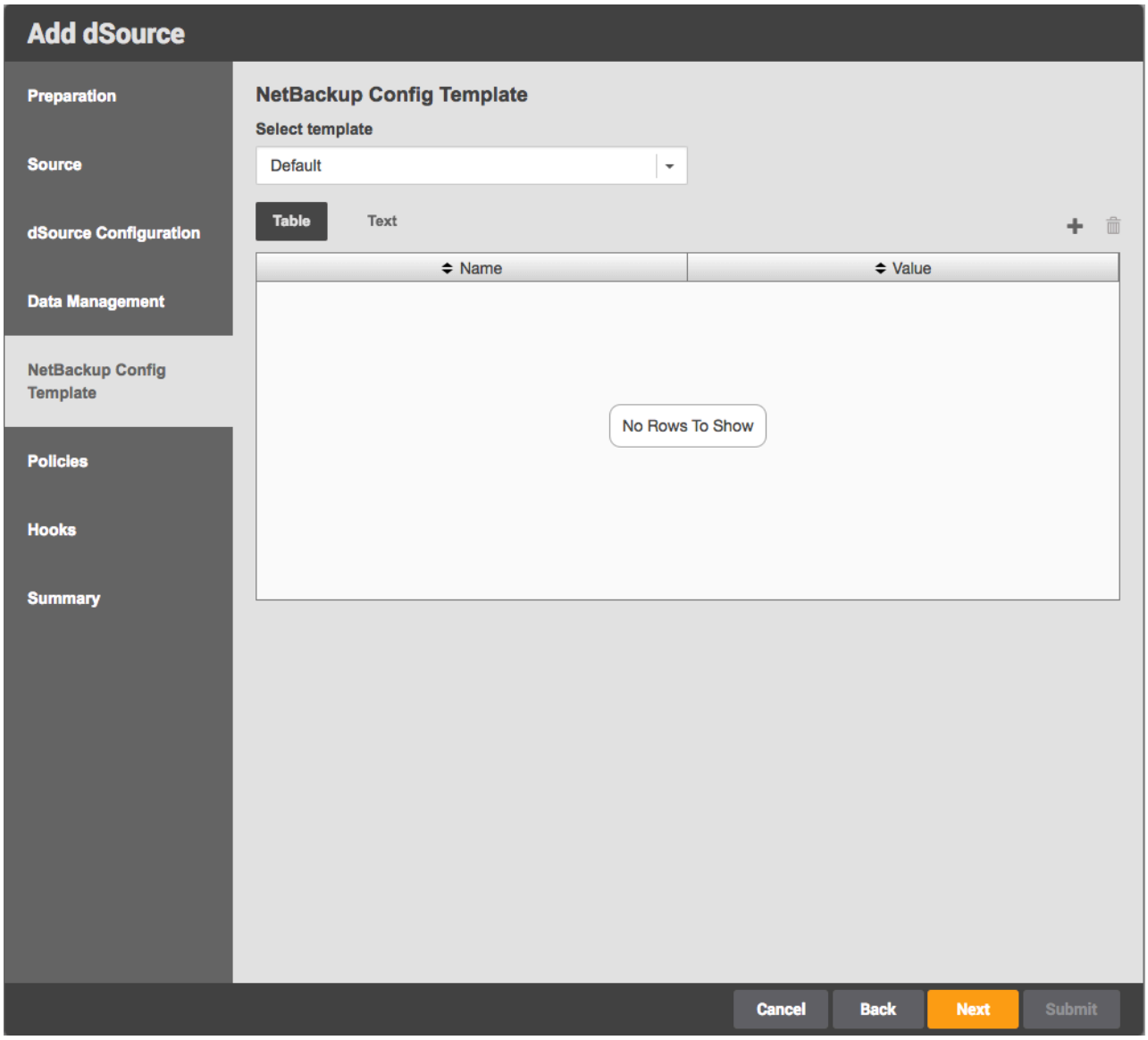
When restoring we create a batch file based on information we find on the backup. Batch files can be customized with non-default and additional options using config templates.

https://www.veritas.com/content/support/en_US/doc/123947690-126579517-0/id-SF930853806-126579517 explains which parameters in the batch file can be edited outside of these blacklisted keywords:

- ALTCLIENT
- BROWSECLIENT
- DATABASE
- DUMPOPTION
- ENABLESERVICEBROKER
- ENDOPER
- MOVE
- NBIMAGE
- NBSERVER
- OBJECTNAME
- OBJECTYPE
- RECOVERED STATE
- RESTOREBEFOREMARK

- RESTOREBEFOREMARK AFTERTIME
- RESTOREOPTION
- RESTOREPAGES
- PARTIAL
- PAGE Any key that is used outside of what is documented may cause the restore to fail with unknown errors.

If the **Configure NetBackup Config Template** is checked, the page after **Data Management** becomes **NetBackup Config Template**.



This page allows this dSource's specific config parameters to be updated, either by adding rows to the table or inputting them as text input (key=value).

> For config params/templates, the 'value' will be injected into the script as is so if the value of the parameter needs to be a String, quotes should be included with 'value'.

If creating a new set of config parameters while linking each dSource seems unnecessary, a config template can be specified. However, that is only possible via the CLI. To do that, log onto the CLI using the Delphix admin user/password and go to **database** > **templates**. Make sure to create the new template with sourceType set to MSSqlLinkedSource.

Example:

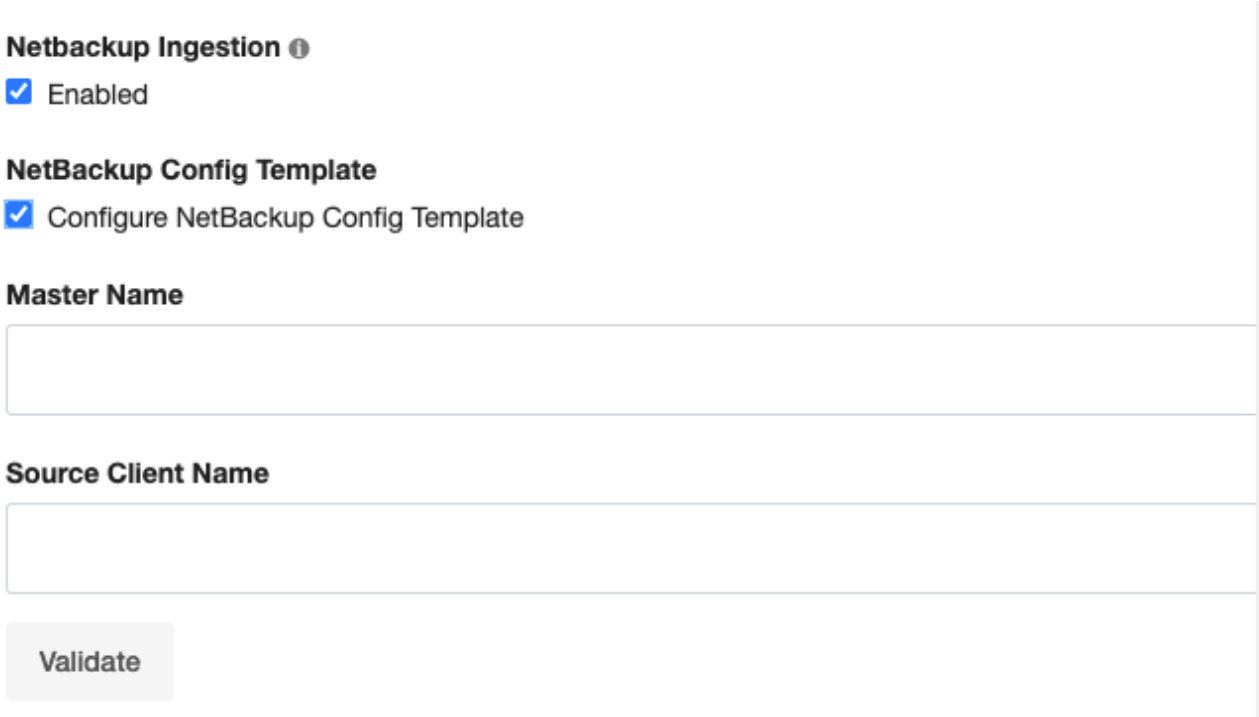
Then when linking, select the template created. (In this screenshot it would be **NetBackup**) If any edits are made to a selected template while linking, this will create a new set of config parameters for the specific dSource and will not edit the actual template. Config parameters added to a specific dSource will be ignored if the dSource has a config template selected.

9.12.5.5.4.4 LogSync for SQL server dSources

LogSync can still be enabled if NetBackup ingestion is enabled however Point-in-Time provisioning is currently not supported for NetBackup transaction logs. LogSync for backups taken with other backup providers that support LogSync will work as before.

9.12.5.5.4.5 Enabling NetBackup for previously created dSources

On an already created dSource go to **Configure** > **Data Management** and click edit (pencil button). Enable NetBackup Ingestion and add all the configuration as necessary.

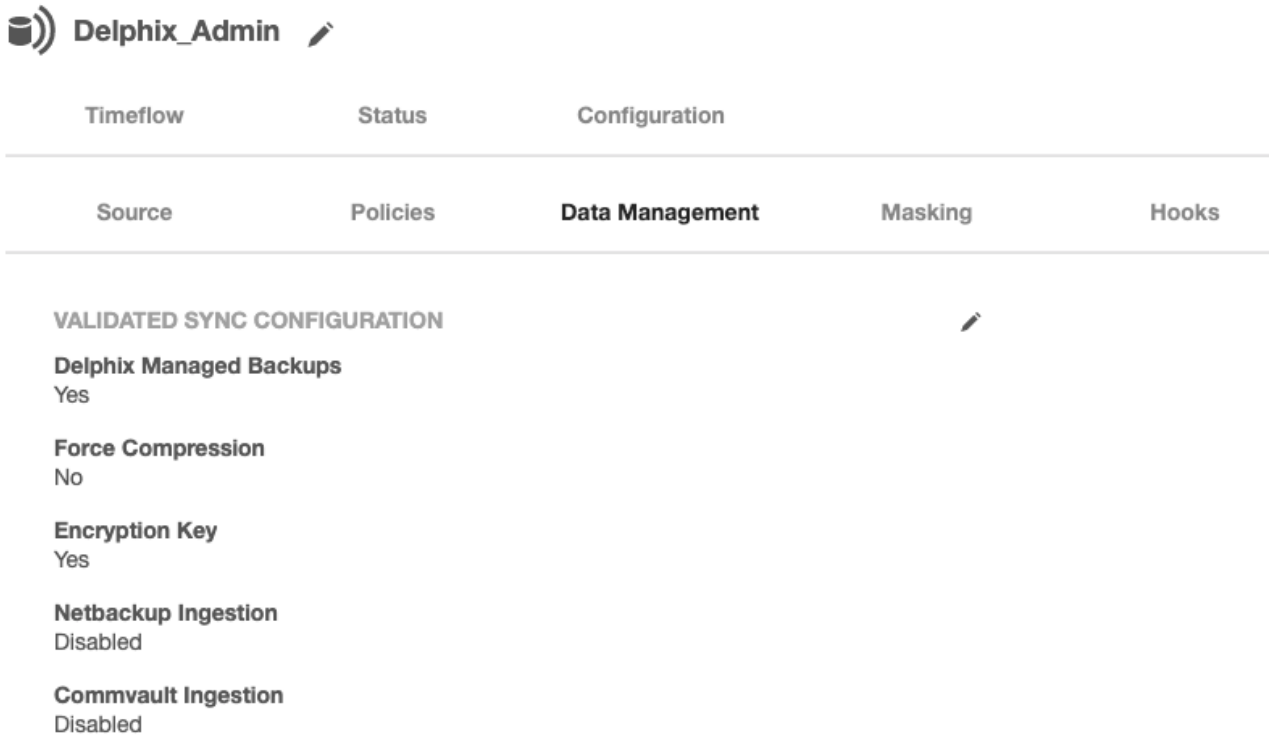


If **NetBackup Config Template** is set to **Default** then the params will be empty. Go to the CLI to update the config params.

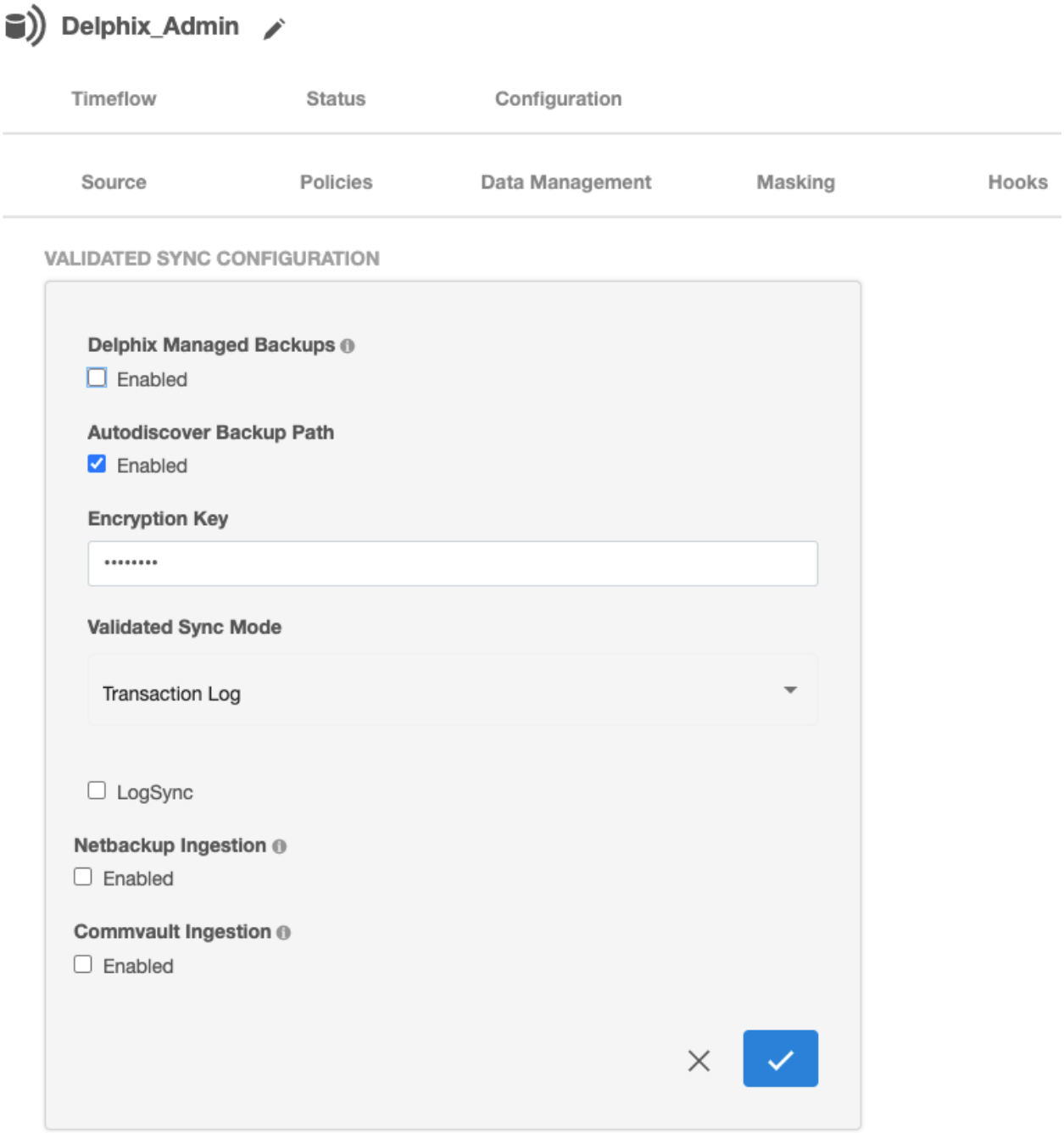
```
Password:
ip-10-110-238-136> cd source
ip-10-110-238-136 source> select Biscuit
ip-10-110-238-136 source 'Biscuit'> update
ip-10-110-238-136 source 'Biscuit' update *> edit mssqlNetbackupConfig
ip-10-110-238-136 source 'Biscuit' update mssqlNetbackupConfig *> set configParams.BLOCK_SIZE=6
ip-10-110-238-136 source 'Biscuit' update mssqlNetbackupConfig *> ls
Properties
    type: MSSqlNetbackupConfig
    configParams:
        BLOCK_SIZE: 6 (*)
    configTemplate: (unset)
    masterName: nbu-80-master.delphix.com
    sourceClientName: ln-win2012src.dlpxdc.co
ip-10-110-238-136 source 'Biscuit' update mssqlNetbackupConfig *> commit
```

Like above, to create a config template, create the template first so that during this update that specific template can be selected in the drop-down.

Delphix Managed Backups must be disabled before NetBackup can be enabled.
To disable Delphix Managed Backups for a specific dSource, select that specific dataset and go to **Data Management** tab under the **Configuration** tab.



Click the edit button and uncheck this feature. Then update all new inputs.



9.12.5.5.4.6 Linking an availability group database with NetBackup

Linking with an AG source works similarly however the source client name inputted to the NetBackup config should be the windows cluster name rather than the client name that takes the backup.

9.12.5.5.4.7 General notes/troubleshooting

- If a restore fails and the staging environment's NetBackup client job log shows no attempted restores, it is likely that you need to log onto the environment using the staging user provided to Delphix. This article explains that the user profile is created on the first login and that NetBackup requires it to restore. https://www.veritas.com/support/en_US/article.100032299
- Through observation, we've noticed that the block size of the restore batch file must be equal to or less than what was used when backing up the dump. We default to the blocksize to what the backup was taken with but the config param/template can include BLOCK_SIZE to change. We won't prevent this but it's up to the user to set it correctly. Editing this field is generally not recommended for this reason.
- NetBackup backups are not on local storage and therefore the Backup Paths will just be ignored.

9.12.5.5.5 Restoring SQL backups stored in Azure cloud storage

9.12.5.5.5.1 Need for supporting SQL backups stored in Azure cloud storage

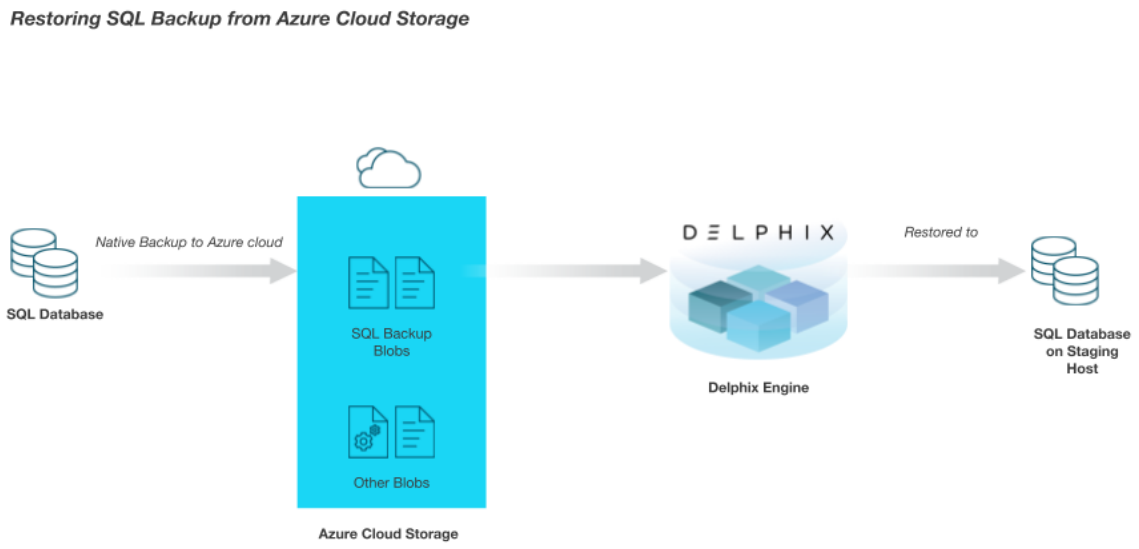
Currently, you can restore the SQL backups that are stored locally on the staging host, or on the network path, or on the backup servers with third-party vendors such as Commvault or Netbackup. At the same time, there are users that are having their native SQL backups on Azure Cloud Storage.

Delphix now supports restoring native SQL backups from Azure Cloud Storage. This enables users who are moving to Azure to use direct backups from the Azure Storage containers instead of third-party vendors.

These backups support the following.

- This is supported by SQL Server 2016 and above
- All backup modes
    - Full backups
    - Differential
    - Transaction log backups Logsync (point-in-time provisioning) is currently not supported. However, LogSync can still be enabled if Azure backup is being ingested. LogSync for SQL native backups that are present on Disk will work as before.
- Striped backups All backup files are to be entirely present on the Azure Cloud and no part of a backup should be present outside of Azure Cloud.
- Validated Sync
- Azure Storage authentication mode This backup solution uses a shared access signature token authentication method to authorize access to the blob data.

9.12.5.5.5.2 Workflow architecture



9.12.5.5.5.3 Workflow Steps

- **User takes backup**: You need to take the SQL native backup directly to the Azure Cloud using SQL Backup to URL[640]. Performance can be improved by enabling COMPRESSION while taking the backup. If the backup size is large, it should be striped into multiple files.

  ```
  BACKUP DATABASE [SourceDB4]
  to URL = 'https://idea1201.blob.core.windows.net/ideal1201container/source.bak'
  ```

- **Authentication to authorize access to the blob data**:
  You must create a SQL Credential[641] on the staging host for the following cases.
  - Create using SAS authentication token as Access Key authentication is currently not supported.
  - Create for each Azure container where the backup files are expected to be stored.
  - Create on each SQL instance where restore has to be performed.

640 https://docs.microsoft.com/en-us/sql/relational-databases/backup-restore/sql-server-backup-to-url?view=sql-server-ver15

641 https://docs.microsoft.com/en-us/sql/relational-databases/backup-restore/sql-server-backup-to-url?view=sql-server-ver15#credential

```
CREATE CREDENTIAL [ https://idea1201.blob.core.windows.net/
ideal1201container] (this is the Azure Container URL)
WITH IDENTITY = 'SHARED ACCESS SIGNATURE',
SECRET =
'sv=2019-12-12&ss=bfqt&srt=c&sp=rwdlacupx&se=2021-01-19T14:18:04Z&st=202
1-01-19T06:18:04Z&spr=https&sig=DBlZnu0VQTaXUwY9IgBEqNbSk';
(provide the SAS token here)
```

- **Run queries on Staging host**: Delphix Engine uses this credential to run the following queries Restore[642], Restore Headeronly[643], and Restore Filelistonly[644] on the staging host.
- Irrespective of whether auto-discovery is selected or not, if an Azure backup is being ingested, its location will be fetched from the `msdb.dbo.backupmediafamily` table on the source host and it will be restored on the staging host. Hence, it is necessary that the backup files are present on the same blob URL where the backup was originally taken. This is different from the usual native backups to the disk.

> In SQL native disk backup,
>
> - If auto-discovery is on, the backup path is fetched from `msdb.dbo.backupmediafamily` table on the source host.
> - If auto-discovery is off, Delphix Engine uses the custom path(s) specified by the user.
>
> If any native Azure backup is found, Delphix Engine will always try to restore it.
>
> - This is similar to the user taking backups using multiple backup vendors.
> - In that case, Delphix Engine tries to restore all backups, irrespective of the backup vendor.

9.12.5.5.5.4 Unsupported features

The following features and functionalities are currently not supported.

- Third-party backup vendors
- Point-in-time provisioning
- Access key authentication method for Azure backups
- Support for SQL Server versions below SQL 2016
- Moving backup files across Azure containers
- Moving a backup from disk to cloud
- Striped backups - Backup files that are partially present on Azure Cloud and partially on another device.
- Managed identity

642 https://docs.microsoft.com/en-us/sql/t-sql/statements/restore-statements-transact-sql?view=sql-server-2016#Azure_Blob
643 https://docs.microsoft.com/en-us/sql/t-sql/statements/restore-statements-headeronly-transact-sql
644 https://docs.microsoft.com/en-us/sql/t-sql/statements/restore-statements-filelistonly-transact-sql

9.12.5.5.6 Working with SQL Server snapshots

This section lists the steps to take a snapshot and delete the same.

Taking a snapshot creates a new snapshot entry in the Oracle dSource's Timeflow. You can use either **Snapshot (Default)** or **Snapshot with Parameters** option for taking the snapshot.

Snapshot (Default)

Perform the following steps to take a snapshot:

1. Login to the **Delphix Management** application.
2. Click **Manage** and select **Datasets** from the dropdown list.
3. Select the dSource you want to Snapshot.
4. Click the **Camera** icon. Alternatively, click the arrow next to the Camera icon and select **Snapshot (default)**.
5. From the Snapshot dialog box, select **Yes**.
6. Navigate to the **Timeflow** tab and click **View: All snapshots** to verify the Snapshot you just created.
7. To delete the snapshot, select the snapshot you just created, and from the Actions menu (...), select **Delete Snapshot**.
8. From the **Delete Snapshot** dialog box, select **Delete**.
9. Navigate to the **Timeflow** tab and click **View: All snapshots** to verify the snapshot you just deleted.

Snapshot with Parameters

Perform the following steps to take a snapshot:

1. Login to the **Delphix Management** application.
2. Click **Manage** and select **Datasets** from the dropdown list.
3. Select the dSource you want to Snapshot.
4. Click the arrow next to the Camera icon and select the **Snapshot with Params...** option.
5. From the Snapshot dialog box, select one of the following:
    a. **Force Full Backup** - If you select this option, then the Delphix Engine will perform an incremental backup by default. You must select this option only when a full backup is required. Full and Incremental backups consume the same space on the Delphix Engine.
    b. **Double Sync** - Selecting this option will perform a SnapSync operation as normal. After the first SnapSync is successful, the Engine will immediately perform a second SnapSync without waiting for the Log Files required for the first SnapSync to be made consistent. This is most useful when performing the initial SnapSync (or when "Force Full Backup" is selected) on a very large database that would lead to a large number of archive logs being required to make the SnapSync consistent. Provisioning from a SnapSync that requires excessive recovery is typically time-consuming.

   c. **Do Not Resume** - If a failure is encountered during the initial SnapSync, the Delphix Engine can resume the SnapSync at a later date. This option will cause the engine to not resume, but rather to start the initial SnapSync over again.

6. Navigate to the **Timeflow** tab and click **View: All snapshots** to verify the Snapshot you just created.
7. To delete the snapshot, select the snapshot you just created, and from the Actions menu (...), select **Delete Snapshot**.
8. From the **Delete Snapshot** dialog box, select **Delete**.
9. Navigate to the **Timeflow** tab and click **View: All snapshots** to verify the snapshot you just deleted.

9.12.5.5.7 Detaching and re-attaching SQL Server dSources

Each dSource contains metadata that associates it with the source database, as well as the data it has ingested from the source database in the form of snapshots up to that point. It is possible to detach, or unlink, a dSource from its source database. This breaks the association with the source database without affecting the data within the Delphix Engine. Detached dSources and their source databases have these properties:

- A detached dSources can still be used to provision a virtual database (VDB).
- You can re-link the source database as a different dSource.

9.12.5.5.7.1 Prerequisites

A dSource can only be attached to a new data source once it has been unlinked.

9.12.5.5.7.2 Detaching a dSource

1. Login to the **Delphix Management** application as a user with **OWNER** privileges on the dSource, group, or domain.
2. Click **Manage**.
3. Select **Databases**.
4. Select the **database** you want to unlink or delete.
5. From the **Actions** menu (...) select **Unlink.** A warning message will appear.
6. Click **Unlink** to confirm.

> Rebuilding Source Databases and Using VDBs
> In situations where you want to rebuild a source database but retain the existing dSource, you will need to detach the original dSource and create a new one from the rebuilt data source.
>
> 1. Detach the dSource as described in the procedure on this page.

2. You cannot attach a dSource with the same name as a dSource that is already attached. If you intend to give the new dSource the same name as the original one, rename the detached dSource.
    a. At the top of the **Configuration** tab, next to the dSource's name, click the **Edit** (pencil) icon.
    b. After renaming the dSource, click the green **checkmark**.
3. Create the new dSource from the rebuilt database.

You will now be able to provision VDBs from both the detached dSource and the newly created one, but the detached dSource will only represent the state of the source database prior to being detached.

9.12.5.5.7.3 Attaching a previously detached dSource

You can only re-attach databases that represent the same physical database.

1. Login to the **Delphix Management** application as a user with **OWNER** privileges on the dSource, group, or domain.
2. Click **Manage**.
3. Select **Databases**.
4. Select the database you want to re-attach.
5. From the **Actions** menu (...) select **Link dSource**.
6. Select the environment and database that represents the source to which you would like to re-attach.
7. Select an appropriate staging environment, and choose the credentials that should be used to authenticate to the source database.
8. Click **Link** to link the dSource to the new source database.
9. Once linking is complete, use the **Configuration > Data Managemen**t tab to set up an appropriate **Validated Sync** configuration.

9.12.5.5.7.4 Attaching a previously detached dSource using CLI

You can only re-attach databases that represent the same physical database.

1. Login to the **Delphix CLI** as delphix_admin or a user with OWNER privileges on the dSource, group, or domain.
2. Select the dSource by name using `database select <dSource Name>.` .
3. Run the `attachSource` command.
4. Set the source config you want to attach to, using `set source.config=<Source Database Unique Name>` . Source configs are named by their database unique name.

5. Set any other source configuration operations as you would for a normal link operation.
6. Run the `commit` command.

9.12.6 Provisioning and managing virtual databases with SQL Server

Virtual databases are a key data management concept for Delphix. In order to create or provision a virtual database, you will need a linked dSource from a source host and a compatible target environment. From a dSource, you can select a snapshot or point in time to create a VDB. SQL Server VDBs each have their own configuration settings.

This section covers the following topics:

- Overview of SQL server virtual databases (see page 2180)
- Provisioning a SQL server VDB (see page 2183)
- V2P with a SQL server VDB (see page 2187)
- Additional SQL server VDB topics (see page 2188)
- CDC support in SQL server (see page 2195)
- MSSQL V2P file mapping (see page 2197)

9.12.6.1 Overview of SQL Server virtual databases

Virtual databases are a key data management concept for Delphix, explained in Provisioning and Managing Virtual Databases (see page 1046). In order to create or provision a virtual database, you will need a linked dSource from a source host and a compatible target environment, as described in the overview for Managing Environments and Hosts (see page 1017) and Overview of Requirements for SQL Server. (see page 2067)

From a dSource, you can select a snapshot or point in time to create a VDB. SQL Server VDBs each have their own configuration settings as described in Configuration Settings for SQL Server Virtual Databases below. This document describes the steps to provisioning VDBs with SQL Server.

9.12.6.1.1 Procedure

1. In the Datasets panel on the left-hand side, click the group containing the dSource or VDB from which you want to provision and select the dSource or VDB from the provided list.
2. From the TimeFlow tab, select a snapshot to provision from. To provision from a specific point in time from dSources with LogSync enabled, use the open LogSync button.
3. Click to open the Provision VDB wizard, and select a compatible Target Environment for the new SQL Server VDB
4. On the Target Configuration page, you may customize the VDB. For a list of available configuration options, see Configuration Settings for SQL Server Virtual Databases (see page 2181) below.
5. Select a Snapshot Policy for the VDB.
6. If the VDB should be masked during provisioning, enable Masked Provisioning by selecting an option on the Masking page
7. Enter any operations that should be run on the Hooks page.

8. Review the VDB Configuration and Summary, and then click Submit.

When provisioning starts, you can review the progress of the job by selecting the VDB and clicking on the Status tab, or by selecting System and viewing the Jobs page.

Alternatively, you can see this in the Actions Sidebar. When provisioning is complete, the VDB will be included in the group you designated and listed in the Datasets panel.

9.12.6.1.2 Configuration settings for SQL server virtual databases

Each VDB has its own data management settings, found during the provisioning workflow as well as in the configuration page for that VDB. When you create a SQL ServerVDB, Delphix copies most configuration settings from the dSource and uses them to create the VDB. However, you can customize these with the following settings:

Setting	Explanation
Recovery Model	The current recovery model of the source database. By default, this value is set to SIMPLE. You must set it explicitly during provisioning.
Auto VDB Restart	Enabling this option will automatically restart this VDB whenever its target host is rebooted.
Change Data Capture (CDC)	Indicate whether this virtual source should be enabled for CDC.

9.12.6.2 Configuration settings for SQL Server virtual databases

Database configuration settings for a SQL Server virtual database can be provided as a key-value pair. Currently, there is support for the below-mentioned configuration parameter(s). These parameter(s) can be added in a configuration template and then applied to virtual databases.

The following parameters are available for SQL Server VDBs:

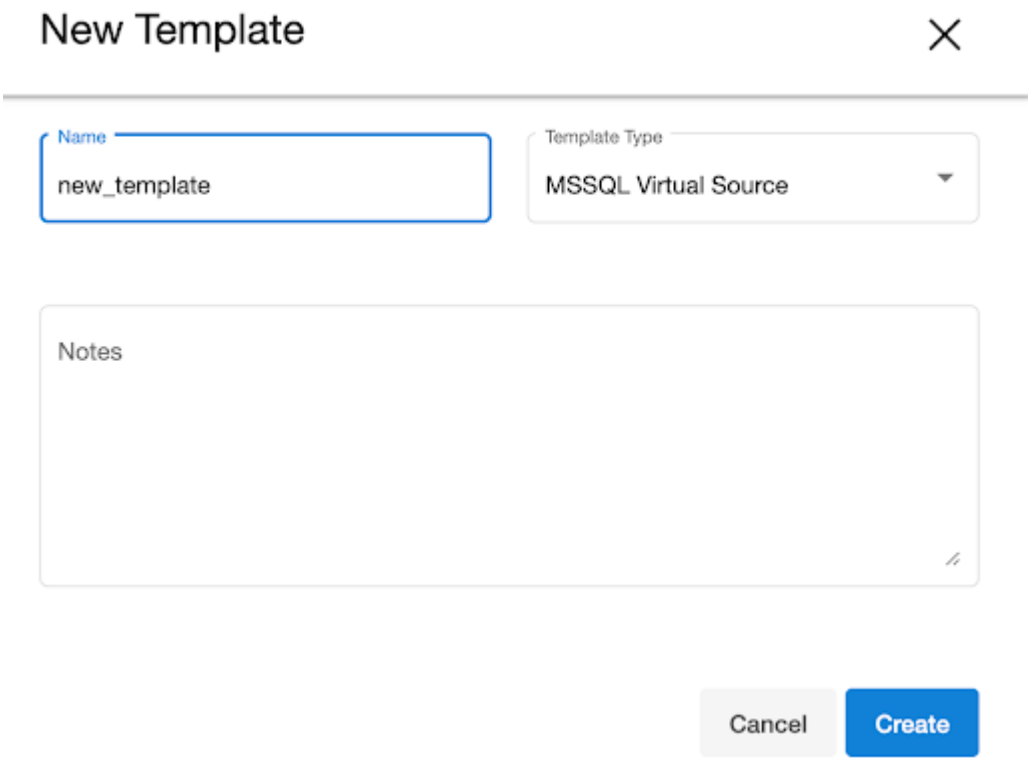
VDB Configuration Parameters	Value	Explanation
READ_COMMITED_SNAPSHOT	ON/OFF	Enables the READ_COMMITTED_SNAPSHOT isolation database option on the virtual database. By default, this option is set to OFF.

9.12.6.2.1 VDB config templates

A VDB config template is a list of database configuration parameter names and values that you can save on the Delphix Engine to use at a later time.

9.12.6.2.1.1 Creating a VDB config template via GUI

1. Log into the **Delphix Management** application as an engine administrator.
2. Click **Manage**.
3. Select **VDB Config Templates**.
4. Click the + icon next to the **VDB Config Temp...** and select **New Template** to create a new template.
5. In the **New Template** dialog window, enter the name for the new template, the parameters that you want to provide, and select the template type from the available options.

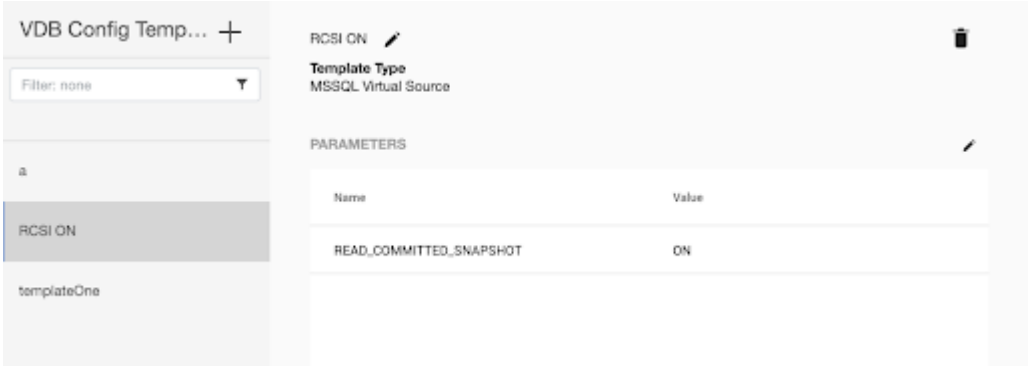


6. Click **Create**.

9.12.6.2.1.2 Updating a VDB config template via GUI

1. Log into the **Delphix Management** application as an engine administrator.
2. Click **Manage**.
3. Select **VDB Config Templates**.
4. Select the template from the left-side pane that you need to update.
5. Click on the **pencil** icons next to the parameters to edit an existing VDB template.

6. To add a new parameter, click the **plus** icon, and enter a name and value of the parameter.



7. Click the button to save the changes or click the button to discard the changes that you made.

You can apply a VDB Config Template to a VDB during the provisioning process, which copies the values from the template. Any subsequent changes to the template will be reflected in the VDB when that VDB is refreshed/rewinded. During provisioning, you can specify configuration parameters directly or copy them from a VDB Configuration Template. Once set, the Delphix Engine will use these parameters whenever the VDB is refreshed/rewinded, even if you change the original template.

9.12.6.3 Provisioning a SQL Server VDB

This topic describes how to provision a virtual database (VDB) from a SQL Server dSource.

9.12.6.3.1 Prerequisites

- You must have already linked a dSource from a source database, as described in Linking a SQL Server dSource (see page 2139) or have already created a VDB from which you want to provision another VDB.
- You must have already set up Windows target environments and installed the Delphix Connector on them, as described in Adding a SQL Server Standalone Target Environment. (see page 2111)
- Make sure that you have the required privileges on the target environment.
- If you are provisioning to a different target environment than the one where the staging database has been set up, make sure that the two environments have compatible operating systems. For more information on the staging database and the validated sync process.
- If using Change Data Capture (CDC):
  - The SQL Server instance on which the VDB is being provisioned or exported must support CDC.
  - The *SQL Server Agent* for the instance must be running otherwise adding a CDC capture and cleanup jobs will fail.

9.12.6.3.2 Procedure

> When provisioning or refreshing a SQL Server VDBs, both the DB_CHAINING and TRUSTWORTHY database parameters will be disabled (even if they were enabled on the dSource).

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Datasets**.
4. Select a **dSource**.
5. Click **Timeflow** tab.
6. Next to a snapshot select the

   Provision VDB icon. The **Provision VDB** panel opens.
7. Select a **target environment**.
8. On the **Target Configuration** page, specify a **Mount Path.**
   a. Linux and Unix hosts, this mount path must be the full path and not include symlinks.
9. On the **Configuration**page, do the following:
   a. Under **Target Group**, select a Target Group for the VDB.
   b. Enter a database name.
   c. Under Recovery Model, click the drop-down list to select a recovery model. You can choose one of the following:
      i. Simple - This option is recommended and selected by default. It allows SQL Server to automatically mark parts of its transaction log file for re-use if they are not in use.
      ii. Full - This option lets you take the responsibility for taking backups and log backups of the VDB to an external location.
      iii. Bulk Logged - This option lets you take the responsibility for taking backups and log backups of the VDB to an external location.

d. Specify any **Pre Scripts** or **Post Scripts** that should be used during the provisioning process. For CDC users, it is recommended that VDB Post Start hooks be configured to automatically start these jobs on successful provisioning

> It is advised to use a Post Start hook as it is executed in all VDB operations like provision, refresh, rewind, disable/enable. Also, CDC jobs get deleted whenever we disable a VDB. The Post Start hook will add the CDC jobs again on enabling the VDB.

e. Under VDB configuration, enable **Auto VDB Restart** to allow the Delphix Engine to automatically restart the VDB when it detects the target host reboot.

f. To enable **Change Data Capture (CDC)**, (see page 2183)select the Enable checkbox.

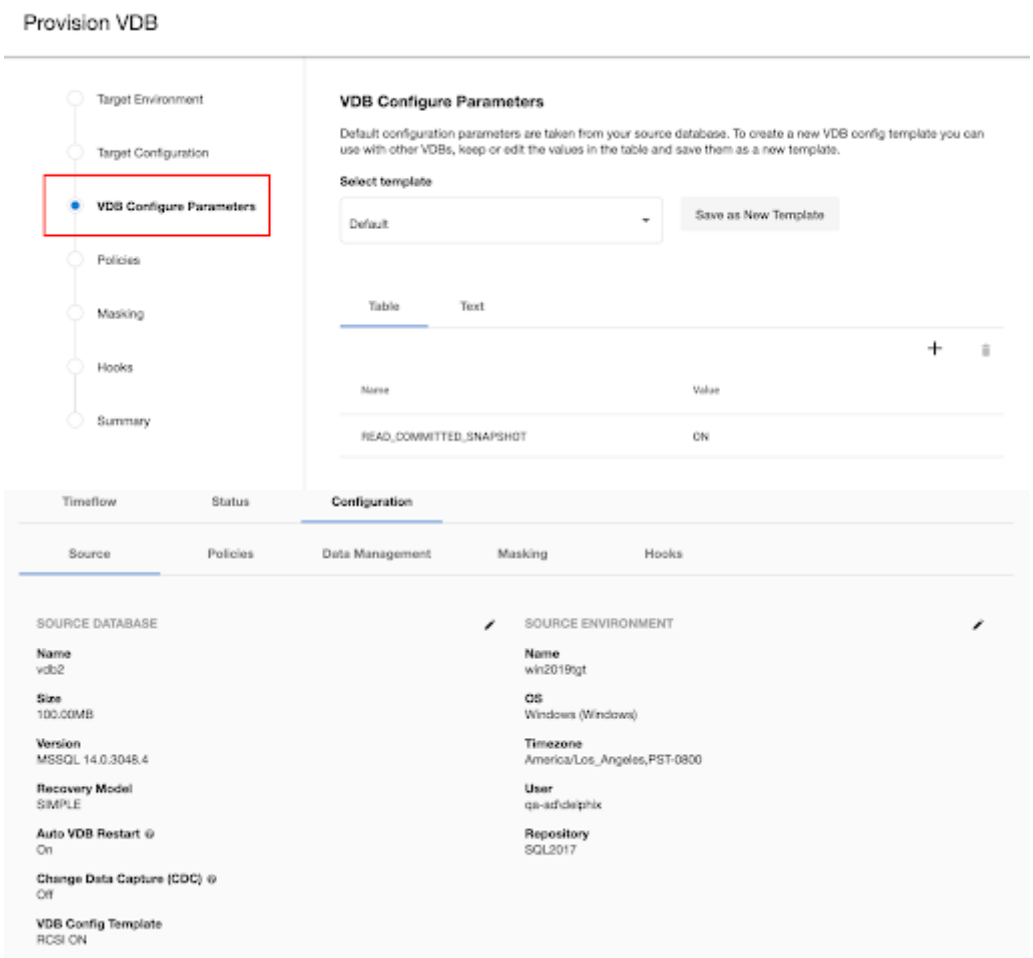
g. To enable VDB configuration parameters, select the **VDB Configuration Parameters** checkbox. This step displays a new page to either select an existing template or set configuration parameters.

> READ_COMMITTED_SNAPSHOT is the only parameter that can be defined here, and the allowed values are OFF and ON.

To do so, perform the following steps on the **VDB Configure Parameters** page:

a. Click the plus icon to add a new key-value pair as a new template for the configuration parameters.

b. OR, From the **Select Template** dropdown list, select an existing configuration template to be applied on the VDB. Applied VDB Config Template will be displayed under the **Configuration** tab in the dataset and will remain editable.



h. Click **Next**.

CDC documentation

For more information see:

- About Change Data Capture[645]
- Enable and Disable Change Data Capture[646]

10. Under **Policies**, select a **Snapshot Policy** for the VDB and click **Next**.
11. Under **Masking**, select **Mask this VDB** checkbox to mask your data during provisioning and then select one of the following masking options:
    a. Select an existing masking job
    b. Masking Job is not currently available for the selected data type, please mask using script(s) instead. If you select this option, you should define a Configure Clone script in the Hooks step to mask the dataset.
12. Under **Hooks**, specify any Hooks to be used during the provisioning process. For more information, see Hooks for SQL Server. (see page 2202)
13. Click **Next**.
14. The final summary tab will enable you to review your configurations.
15. Click **Submit**.

When provisioning starts, the VDB will appear in the **Datasets** panel. Select the VDB and navigate to the **Status** tab to see the progress of the job. When provisioning is complete, you can see more information on the **Configuration** tab.

You can select a SQL Server instance that has a higher version than the source database and the VDB will be automatically upgraded. For more information about compatibility between different versions of SQL Server, see SQL Server Support Matrix. (see page 2052)

9.12.6.3.2.1 Provisioning by snapshot or logSync

When provisioning by snapshot, you can provision to the start of any particular snapshot, either by time or LSN.

645 https://docs.microsoft.com/en-us/sql/relational-databases/track-changes/about-change-data-capture-sql-server?view=sql-server-ver15

646 https://docs.microsoft.com/en-us/sql/relational-databases/track-changes/enable-and-disable-change-data-capture-sql-server?view=sql-server-ver15

> You can take a new snapshot of the dSource and provision from it by clicking the **Camera** icon. Provisioning By Snapshot

Provisioning By Snapshot	Description
Provision by Time	You can provision to the start of any snapshot by selecting that snapshot card from the **TimeFlow** tab, or by selecting and entering a value in the time entry fields. The values you enter will snap to the beginning of the nearest snapshot.
Provision by LSN	You can use **Provision by LSN** control to open the LSN entry field. Here, you can type or paste in the LSN to which you want to provision. After entering a value, it will "snap" to the start of the closest appropriate snapshot.

9.12.6.4 V2P with a SQL Server VDB

This topic describes how to perform the Virtual to Physical (V2P) process with a SQL Server virtual database (VDB).

9.12.6.4.1 Procedure

1. Log into the **Delphix Management** application.
2. Click **Manage** and select **Datasets**.
3. Select the dSource or VDB you want to export.
4. Select the snapshot of the dSource or VDB state you want to export.
5. If you want to export the state of the database from a specific point in time, select the **LogSync** icon and then select the point in time from which you want to create the export.

6. From the actions menu (...) select **Virtual to Physical**.
7. Select the target environment.
8. Enter the **Target Directory** for the export. The target environment should have been added to Delphix previously and should meet all target host requirements, see Overview of Requirements for SQL Server Environments (see page 2067) for more information on user requirements for target environments.
   The target directory you enter here must exist in the target environment and the Delphix operating system user listed under the environment must have permission to write to it.

9. Select the checkbox to enable or disable the option **Open database after Recovery**.
   If you select to **disable** this option, you can use the scripts that are created in the target environment to manually recover the database at a later time. See Manually Recovering a Database after V2P (see page 1039) for more information.
10. Click **Show Advanced** to customize the target directory layout or to enable Change Data Capture (CDC) on the exported database. See Customizing Target Directory Structure for Database Export[647] for more information.
11. Click **Next**.
12. Select whether you want to have an email sent to you when the export process completes, and then click **Submit**.

9.12.6.4.2 Post-requisites

If you have selected disable for **Open database after Recovery**, then follow the instructions in Manually Recovering a Database after V2P (see page 1039) to complete the V2P process.

9.12.6.5 Additional SQL Server VDB topics

This section covers the following topics:

- Renaming a SQL Server VDB via CLI (see page 2188)
- Upgrading SQL Server VDBs (see page 2190)
- Extended properties for SQL Server VDBs (see page 2190)
- File permissions for SQL Server VDBs (see page 2191)
- Deleting a SQL Server VDB (see page 2192)

9.12.6.5.1 Renaming a SQL Server VDB via CLI

This topic describes how to change the database name on the SQL Server Instance for VDB through the Delphix CLI.

> Database name on SQL Server vs. VDB name on Delphix
> The database name is what you would see the SQL Server instance on the Target in SQL Server Management Studio or sys.databases. It is also the database name in the **Configuration** tab in Delphix.
> The name of the VDB is an internal name within Delphix Engine and does not need to be the same as the database name on Target. This is found on the **Status** tab of Delphx.

647 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/6262948/Customizing+target+directory+structure+for+database+export

9.12.6.5.1.1 Prerequisites

- The VDB must be running on the target environment.
- The SQL Server instance on the target environment where the VDB is located must be up and reachable.

9.12.6.5.1.2 Procedure

1. Select the **source** associated with the VDB and disable the VDB.

```
delphix> source "vexample"
delphix source 'vexample'> disable
delphix source 'vexample' disable *> commit
```

2. Select the source **config** associated with the source.

```
delphix source "vexample" > get config
  vexample
delphix source "vexample" > /sourceconfig "vexample"
delphix sourceconfig "vexample" >
```

3. Update the **databaseName** to the new name.

```
delphix sourceconfig "vexample" > update
delphix sourceconfig "vexample" update *> set databaseName=newDBName
delphix sourceconfig "vexample" update *> commit
delphix sourceconfig "vexample" >cd
```

4. Enable the VDB.

```
delphix> source "vexample"
delphix source 'vexample'> enable
delphix source 'vexample' enable *> commit
```

9.12.6.5.2 Upgrading SQL Server VDBs

This topic describes how to upgrade a SQL Server VDB to a higher version of SQL Server instance.

9.12.6.5.2.1 Procedure for VDB in-place upgrade

1. Remove any VDB Refresh Policy assigned to the VDB.
2. Upgrade the target SQL Server instance.
3. Refresh the target environment.

9.12.6.5.2.2 Procedure to upgrade a VDB to a new SQL instance

1. Refresh all environments.
2. Login to the **Delphix Management** application.
3. Click **Manage**.
4. Select **Datasets**.
5. Select the **VDB** to be upgraded.
6. From the **Actions** menu (...) select **Disable**.
7. Click **Disable** to confirm.
8. From the **Actions** menu (...) select **upgrade**. The **Upgrade Database** window will open.
9. Select the **SQL Server instance** to which you want the VDB to upgrade.
10. Click **OK**.
11. Enable the VDB. (See the *Enabling and Disabling SQL Server VDBs section in Provisioning and Managing Virtual Databases*[648].[649])
12. Repeat steps 5 to 12 for each VDB you want to upgrade.

9.12.6.5.3 Extended properties for SQL Server VDBs

9.12.6.5.3.1 Extended properties and how to view them

This topic describes extended properties on VDBs that you can use to track the origin of VDBs through SQL Server tools on target servers.

These are the extended properties:

648 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/6750977/Provisioning+and+managing+virtual+databases
649 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/getting-started-provisioning-and-managing-virtual-databases

Property	Description
`dlpx_server_name`	Address of the Delphix Engine hosting the VDB
`dlpx_server_uuid`	UUID of the Delphix Engine hosting the VDB
`dlpx_source_id`	Internal reference of the VDB

You can find these properties in the following locations:

- For a VDB using the SQL Server Management Studio tool: under the Extended Properties page of the Properties window
- Using the `sp_dlpx_vdbinfo` stored procedure.
    - To install and run this stored procedure, run the SQL code contained in `<Delphix Connector install path>\etc\sp_dlpx_vdbinfo.sql.`

9.12.6.5.4 File permissions for SQL Server VDBs

When provisioning a VDB, the Delphix Engine modifies the "access control lists" (ACLs) of database and log files to help prevent unintentional data loss through file deletion. Files could be deleted, for example, if there is an attempt to DROP a VDB directly through SQL Server management studio or other native SQL Server tools.

The Delphix Engine updates each database and log file ACL to include a deny-delete "access control entry" (ACE) for the user account running the SQL Server instance.

You can still drop VDBs directly through SQL Server tools. However, a warning message will be displayed, and the files will remain on the volume that the Delphix Engine exports. This file deletion prevention also applies to attempts to remove files from a database using the ALTER DATABASE .. REMOVE FILE command.

If a VDB is inadvertently dropped, you can reattach the database using SQL Server tools.

If you attempt to delete a database or log file and then try to add a file of the same name, this may fail because the original file was prevented from being deleted by the deny-delete ACE.

If you intend to delete the files from the volume that the Delphix Engine provides, you can change the ACLs on the files using the icacls command:

```
icacls <file> /remove <SQL Server instance account>:deny(D)
```

Accounts other than the SQL Server instance account will not be prevented from deleting the VDB database and log files.

9.12.6.5.5 Deleting a SQL Server VDB

9.12.6.5.5.1 Procedure

> Deleting a VDB is an unrecoverable operation. Proceed only if you want to permanently destroy the unique data that was created in the VDB.

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Datasets**.
4. Click the **VDB** that you want to delete.
5. From the Actions menu (...) select **Delete.**
6. If stopping or starting the VDB requires particular credentials for the target environment other than those of the default environment user:
   a. Check **Provide Privileged Credentials**.
   b. Enter the **username** and **password**.
   c. Click **Validate Credentials**.
7. Click **Delete** to confirm that you want to delete the VDB.

If the VDB was currently active, the Delphix Engine will shut it down, unmount all filesystems from the target environment, and finally delete the VDB itself.

9.12.6.5.5.2 Using Force Delete

Deleting unused or outdated objects should be a regular part of Delphix Engine administration. This is especially important to prevent low space errors, which can cause the Delphix Engine to stop. The Delphix Engine holds a maximum of 400 objects.

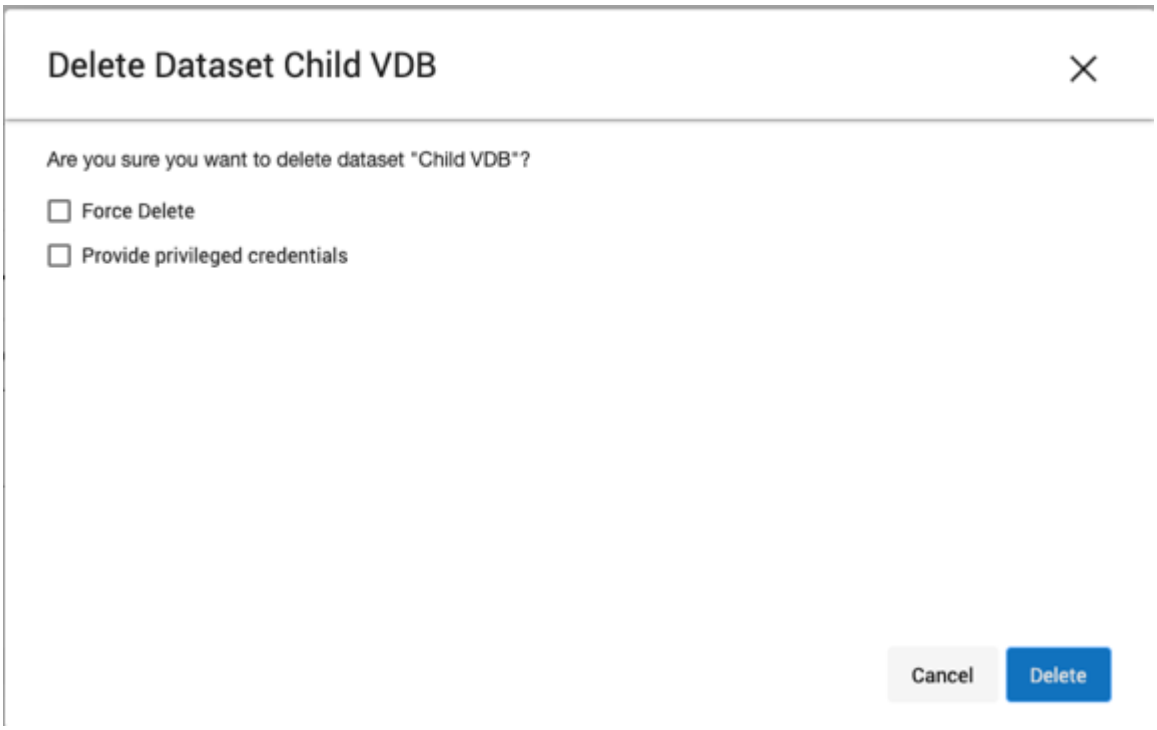
Force Delete can be used when:

- The staging host/container is not accessible via Delphix Engine anymore.
- The Delphix Connector service is not running (applicable to SQL Server only).
- The normal delete is throwing error and not working.
- When the target or staging host is decommissioned.

> Force Delete should not be used as an alternative to Delete in normal circumstances.

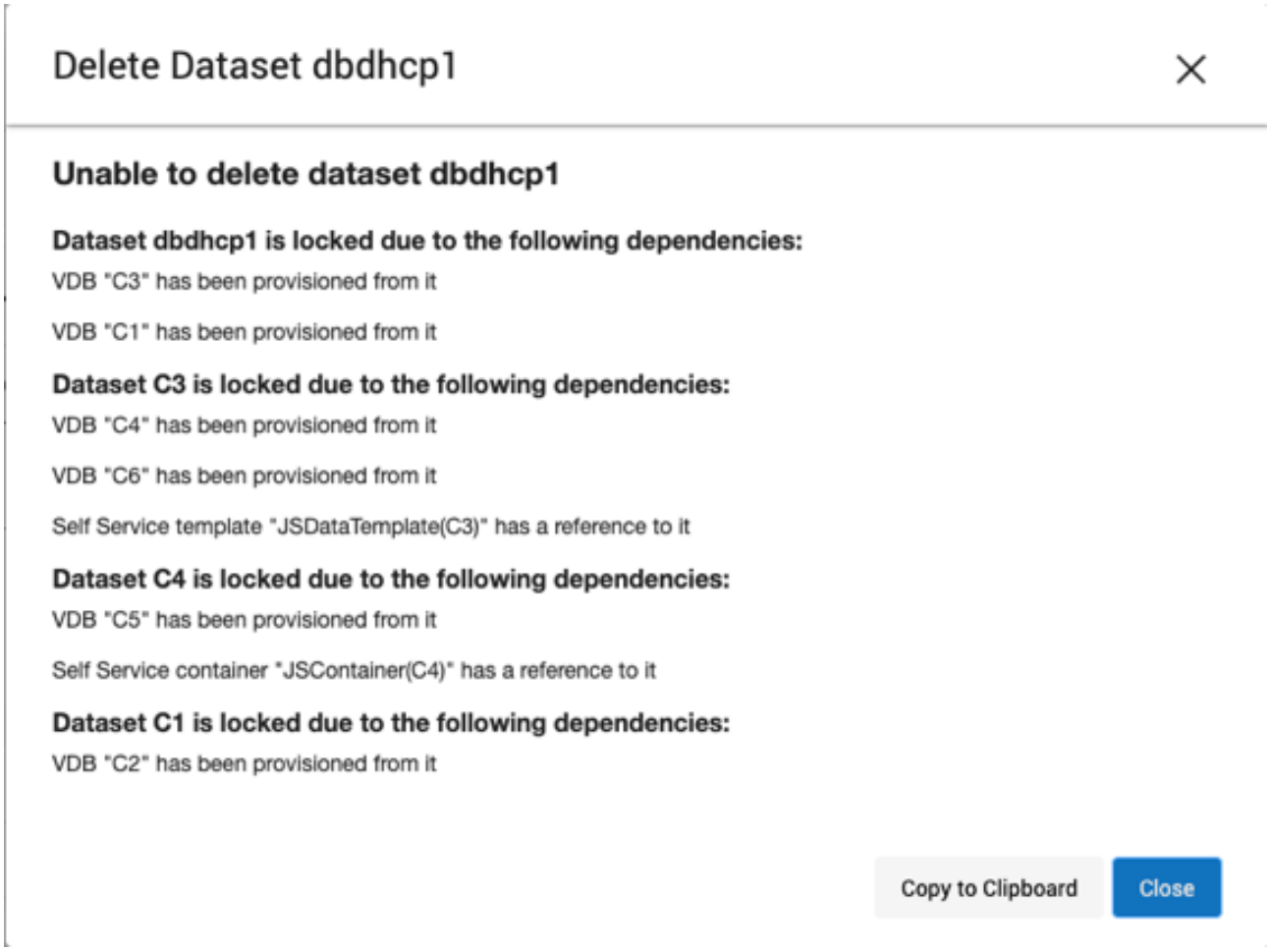
Procedure

1. Log into the **Delphix Management** application.
2. Select **Resources > Storage Capacity**.
3. Next to the object you want to delete select the **Trash can**.
4. In the Delete dialog select **Force Delete**. Oracle users will have the option to provide additional credentials.



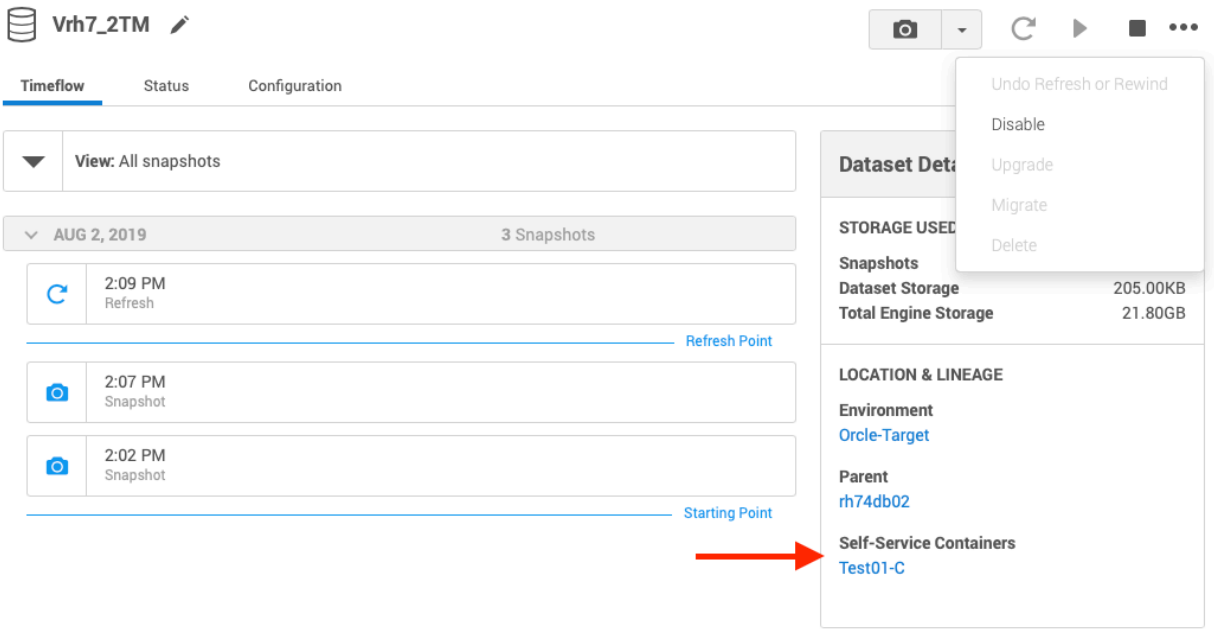
5. Click Delete.

> Dependencies
> If there are dependencies on the snapshot, you will not be able to delete the snapshot free space; the dependencies rely on the data associated with the snapshot. These items are displayed with a lock icon next to the name.



9.12.6.5.5.3 Deleting a VDB associated with a self-service container

As shown below if a VDB is associated with a Self-Service container, the delete option will not show.



Users will need to first delete the container and then delete the VDB. Refer to Delphix Self-Service Data Container Activities. (see page 2452)

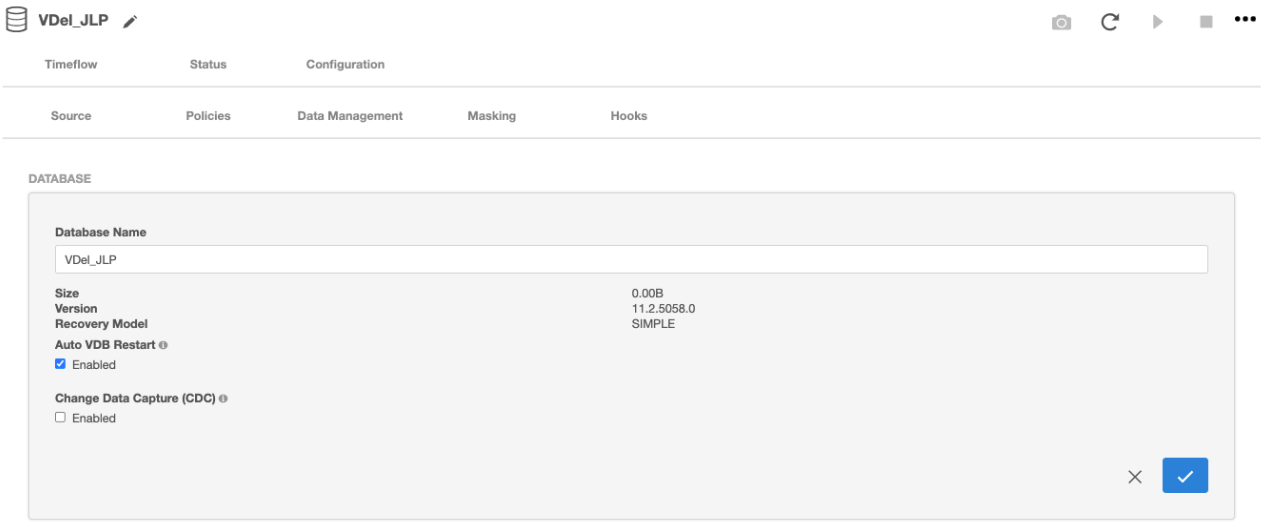
9.12.6.6 CDC support in SQL Server

9.12.6.6.1 Provisioning CDC enabled virtual databases

Please see Provisioning a SQL Server VDB. (see page 2183)

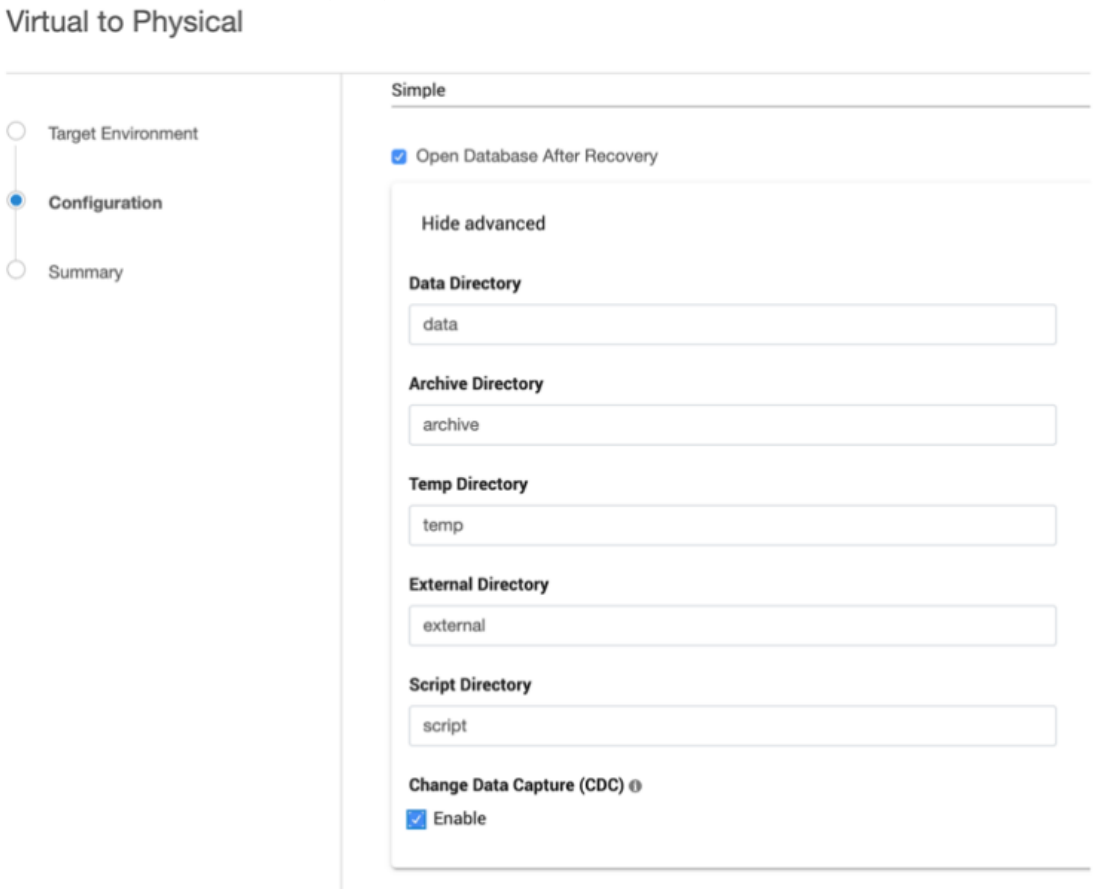
9.12.6.6.2 Viewing or updating CDC your configuration

CDC configuration used while provisioning would be saved and used for subsequent Refresh and Rewind operations on the VDB. This configuration can be viewed/updated after the VDB has provisioned, in the 'Source' tab under the **Configuration'**section of the VDB.



9.12.6.6.3 Exporting CDC enabled physical databases

1. In the Configuration page of the wizard, open the Advanced section and Select the **'Enable'** option under 'Change Data Capture (CDC)



2. CDC capture and cleanup jobs have to be added and CDC metadata has to be upgraded (if exporting is done from a lower database version to SQL2016 and above), manually on exported databases.

9.12.6.6.4 General notes/troubleshooting

- Make sure the SQL Server instance on which the VDB is being provisioned or exported, supports CDC.
- Make sure 'SQL Server Agent' for the instance is running otherwise adding CDC capture and cleanup jobs will fail.

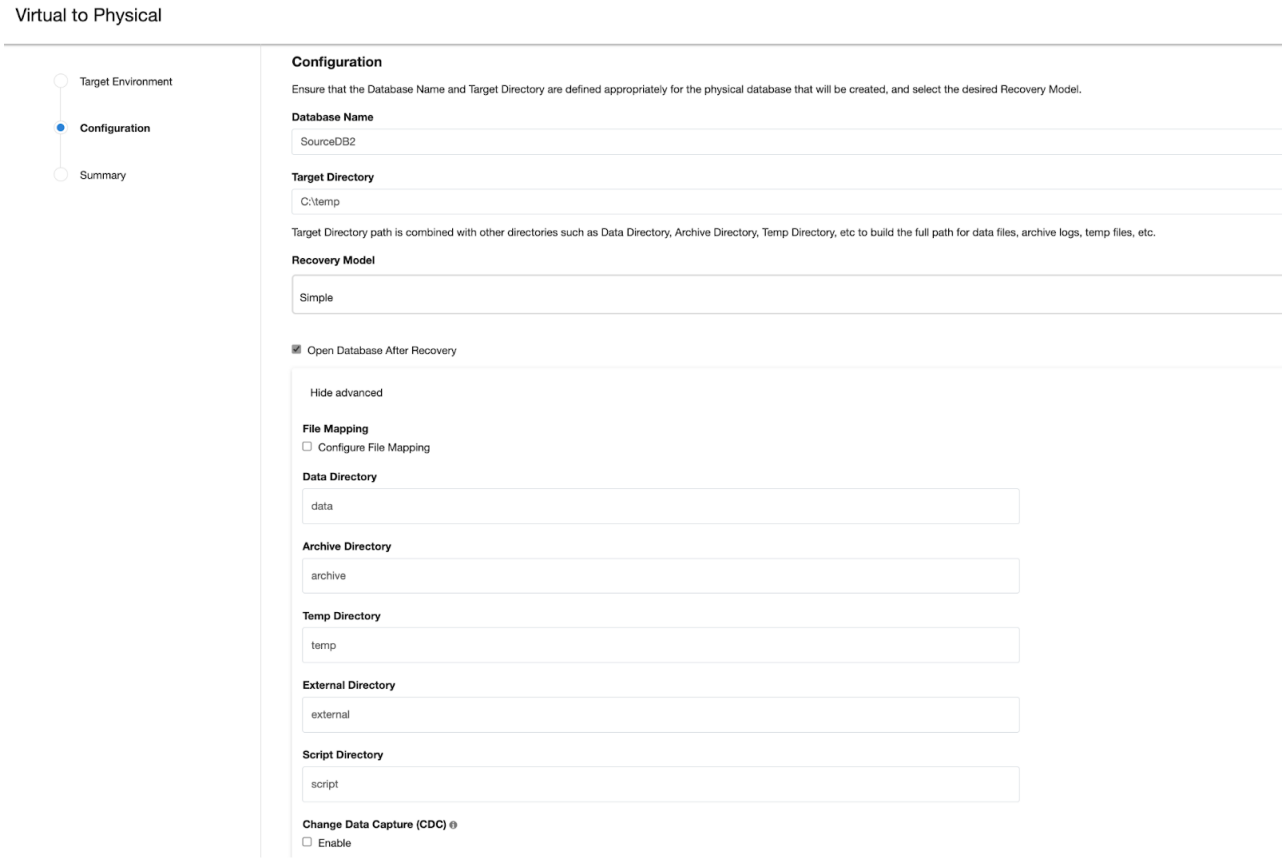
9.12.6.7 MSSQL V2P file mapping

9.12.6.7.1 Introduction

This article describes how to customize file path mappings when performing a Virtual to Physical (V2P) operation for MSSQL databases. During the V2P process, it could be required to create mappings between the files and directories existing on the staging host, as well as files and directories created on the target. For example, putting all of the transaction log files that exist on the staging environment into a folder on the target machine.

The **Configuration** section of the process has name fields that can be specified for the database and directories, as shown in the screenshot below.

- Database Name: V2PDatabase
- Target Directory: C:\temp
- Data Directory: data
- Archive Directory: archive
- Script Directory: script
- Temp Directory: temp
- External Directory: external



When working with File Mappings, the data file names affect everything that follows. These data files, including log files and File Streams folders, inherit file names from the dSources and VDBs by default (e.g., SourceDB2.mdf, SourceDB2_log.ldf, File_Stream, etc.). Users have control over the data files that are part of the dSource and VDB snapshots.

Archive log files go directly into the **C:\temp\archive** directory. If provided, the V2P process automatically appends the target directory and data directory to the data file names, as shown in an example list below.

- C:\temp\data\SourceDB2.mdf
- C:\temp\data\SourceDB2_log.ldf
- C:\temp\data\File_Stream

9.12.6.7.1.1 Pattern matching

Pattern Matching rules can be used to create full path names for data files and control files. These rules have take this format: **source-regex-expression-KEY ? target-replacement-VALUE**. Multiple rules can be used and are applied successively. In addition, multiple rules with the same source key are allowed.

9.12.6.7.2 File mapping options

9.12.6.7.2.1 Example 1

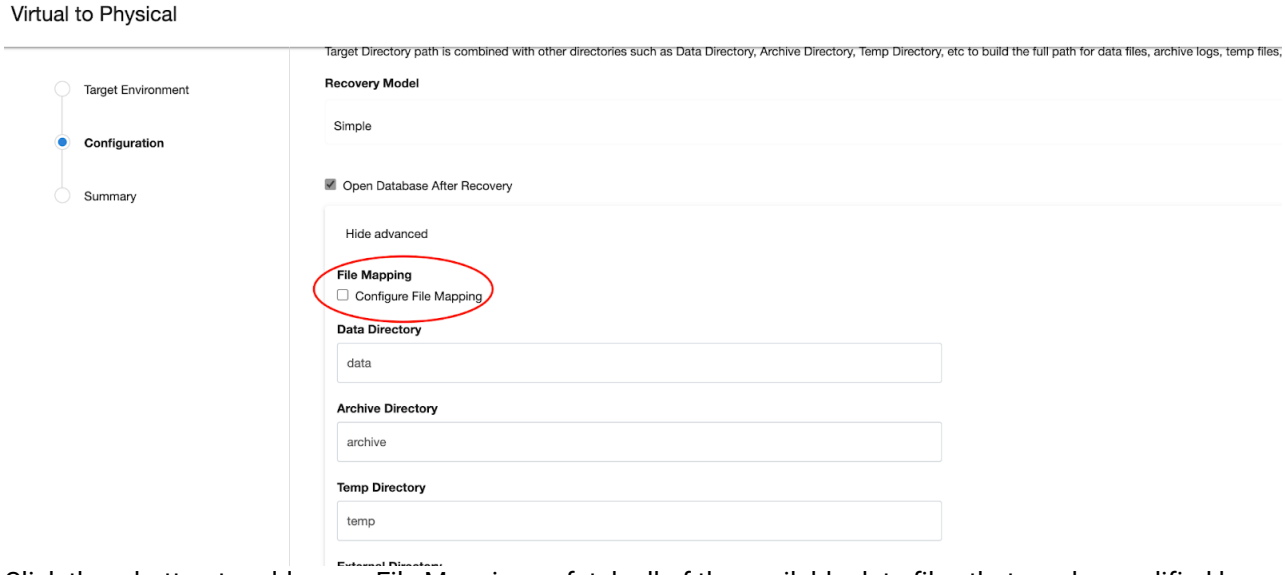
For this example, the ultimate goal is to perform a V2P operation that has the following data file File Mappings:

- C:\temp\SourceDB2\data\SourceDB2.mdf
- D:\temp\SourceDB2\logFiles\SourceDB2_log.ldf
- E:\temp\SourceDB2\fileStream\SourceDB2_File_Stream

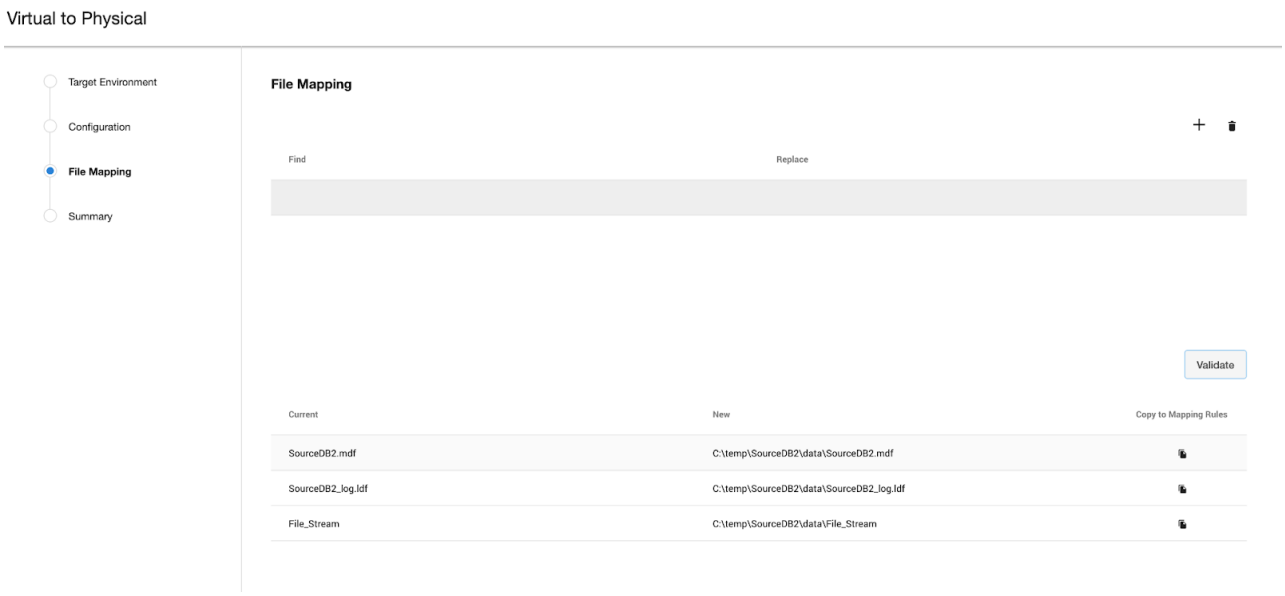
The default behavior can be changed to modify the target directory, modify the data directory, or modify the individual file location

- Modify the target directory.
- Modify the data directory.
- Modify the individual file location by using File Mapping.

To modify the location of each file, select **Configure File Mapping** in the **Advanced** tab of the Configuration page.

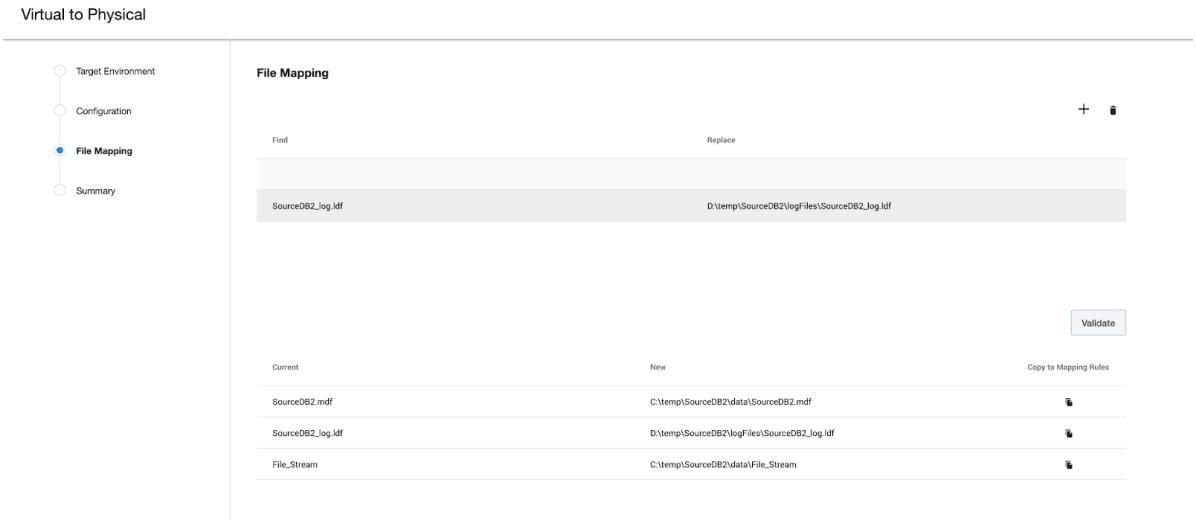


Click the + button to add a new File Mapping or fetch all of the available data files that can be modified by simply clicking **Validate** with an **empty** File Mapping list, as shown in the screenshot below.

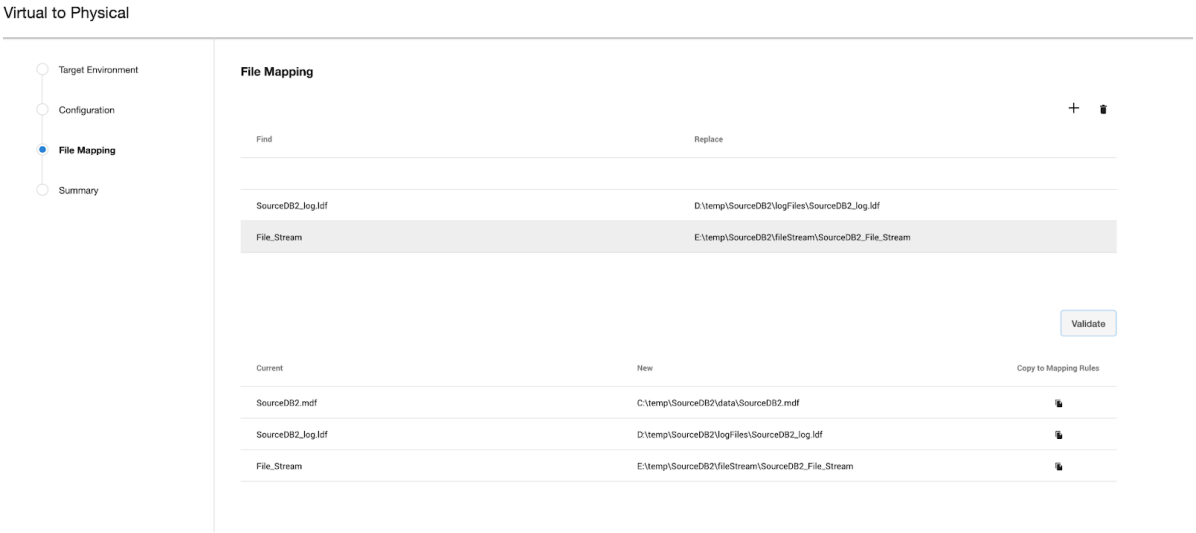


When modifying the file location for the received files, the **Copy to Mapping Rules** option copies the current (source location) and new (target location) to File Mapping, or the values can be entered manually to Find and Replace. The steps below are to follow.

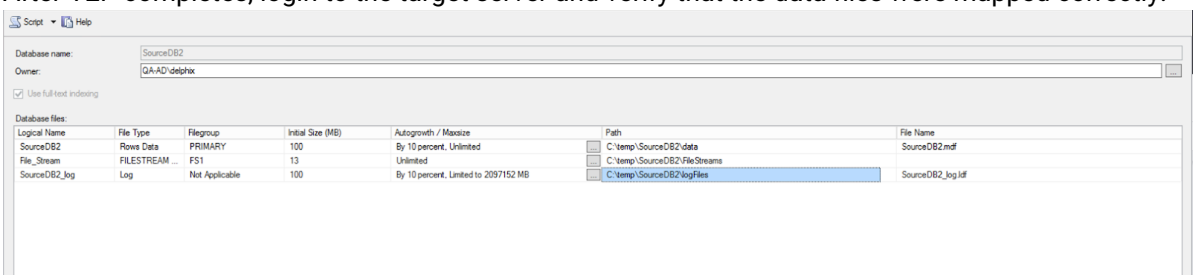
1. Since the new (target location) of the SourceDB2.mdf would appended to `C:\temp\SourceDB2\data\SourceDB2.mdf` (a combination of **C:\temp\SourceDB2** (target directory), **data** (data directory), and **SourceDB2.mdf** (file name)), applying File Mapping is not necessary.
2. Copy the SourceDB2_log.ldf file using the **Copy to Mapping Rules** option.
3. Place the SourceDB2_log.ldf file into the `D:\temp\SourceDB2\logFiles\SourceDB2_log.ldf` location.
4. Configure **Replace** to `D:\temp\SourceDB2\logFiles\SourceDB2_log.ldf` and then select **Validate** to see the results.



5. Similarly, to move **File_Stream** to `E:\temp\SourceDB2\fileStream\SourceDB2_File_Stream`, provide the mapping as shown below.



6. Select **Validate** between each new entry, in order to verify that data files are being mapped as expected.
7. The File Mappings build upon one another, so all the provided File Mapping Rules are applied sequentially to each source file in order to generate a target file location.
8. Once all the files are located as desired, select **Next** to continue the provision process.
9. The **Summary** page will show the modifications to **Target Directory** and **Database Name** directories, and will show that **Customized File Mapping** was defined.
10. After V2P completes, login to the target server and verify that the data files were mapped correctly.

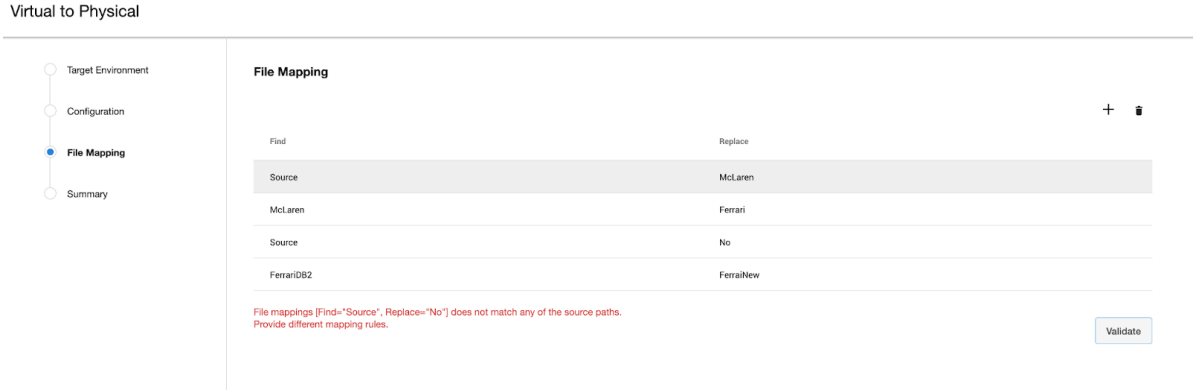


9.12.6.7.2.2 Example 2

In this example, several rules are applied to the source file path for **SourceDB2.mdf**. Note that the rules are applied in sequential order, as shown in the screenshot below.

1. Applying the rule **Source?McLaren** results in: **C:\temp\data\McLarenDB2.mdf**

2. Applying the rule **McLaren?Ferrari** results in: **C:\temp\data\FerrariDB2.mdf**
3. Applying the rule **FerrariDB2?FerraiNew** results in: **C:\temp\data\FerraiNew.mdf**
4. Applying the rule **Source?no** results in an error, because **Source** is no longer found in the pathname.



During the pattern matching process, various errors can be generated. Some of these errors are described below.

1. **No match for specified mapping rules** When none of the rules match a source file
2. **Invalid regex pattern specified for path mapping** An invalid regex rule mapping error
3. **File Extension Mismatch** A modified file extension or folder to file conversion error
4. **Duplicate target Paths** When the same target path is created for multiple source files

The java.regex.util class article[650] (redirect to docs.oracle.com[651]) shows the regular expression syntax and constructs recognized by the Delphix Engine pattern-matching operations.

> As "replace" takes an String value to create a valid windows path, characters like `?`, `<`, `>`, `|`, `*`, `"` are not supported.

9.12.7 Hooks for SQL Server

9.12.7.1 Overview

This topic describes the use of hook operations with dSources created from SQL Server source databases and virtual databases (VDBs) that are created from SQL Server dSources or other VDBs.

Hooks are Windows Powershell code executed on:

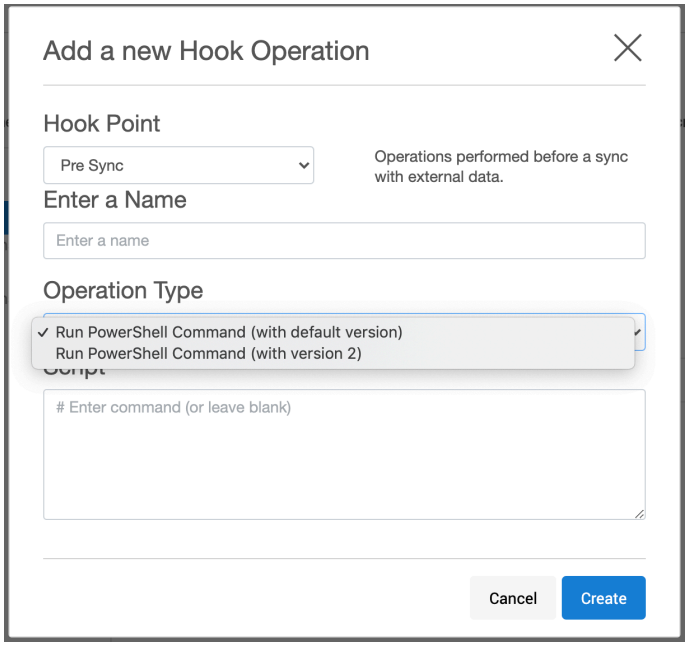
1. The staging target host before or after the manual snapshot of a dSource.

650 https://docs.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html
651 http://docs.oracle.com/

2. The VDB target host before and after the provision, refresh, rewind, snapshot, start or stop of a VDB.

Powershell version

While creating a hook, a user can provide the PowerShell version (default version or version 2) in the field "Operation Type" and this PowerShell version will be used to execute the hook script. Here, the default version is the version of PowerShell installed on the target host.



Hooks can be specified in the wizard used during the creation of a VDB, or modified afterward by navigating to the Configuration > Hooks tab. Hooks can also be set using the Delphix command-line interface (CLI) or REST Web API.

Each hook operation represents a user-configurable action that the Delphix virtualization engine will execute. You can configure the custom hook code to fail if they encounter an unexpected error. The failure of a hook operation will cause the enclosing operation to fail.

The Windows environment user for the dSource or VDB runs the "Powershell" executable, which runs the specified PowerShell script on the Staging or VDB Target host. The Delphix Engine captures and logs all output of the script and displays it if a failure occurs.

The intent of hook operations are customization of the data contents or configuration of a dataset while it is being manipulated. Actions performed by hooks effectively become an integrated part of the sync operations of a dSource or the provision, refresh, rewind, snapshot, start, or stop actions for that VDB.

Hooks are mainly used for pre- and post-provisioning operations. For example, you can use hooks to:

- Back up test data before refresh and rewind
- Back up data after provisioning
- Reset configuration settings from production to non-production settings after provisioning
- Create logins for dev/qa users who do not have privileges on production databases

- Sync logins on the target that are cloned from the production database
- Back up configuration data from the database

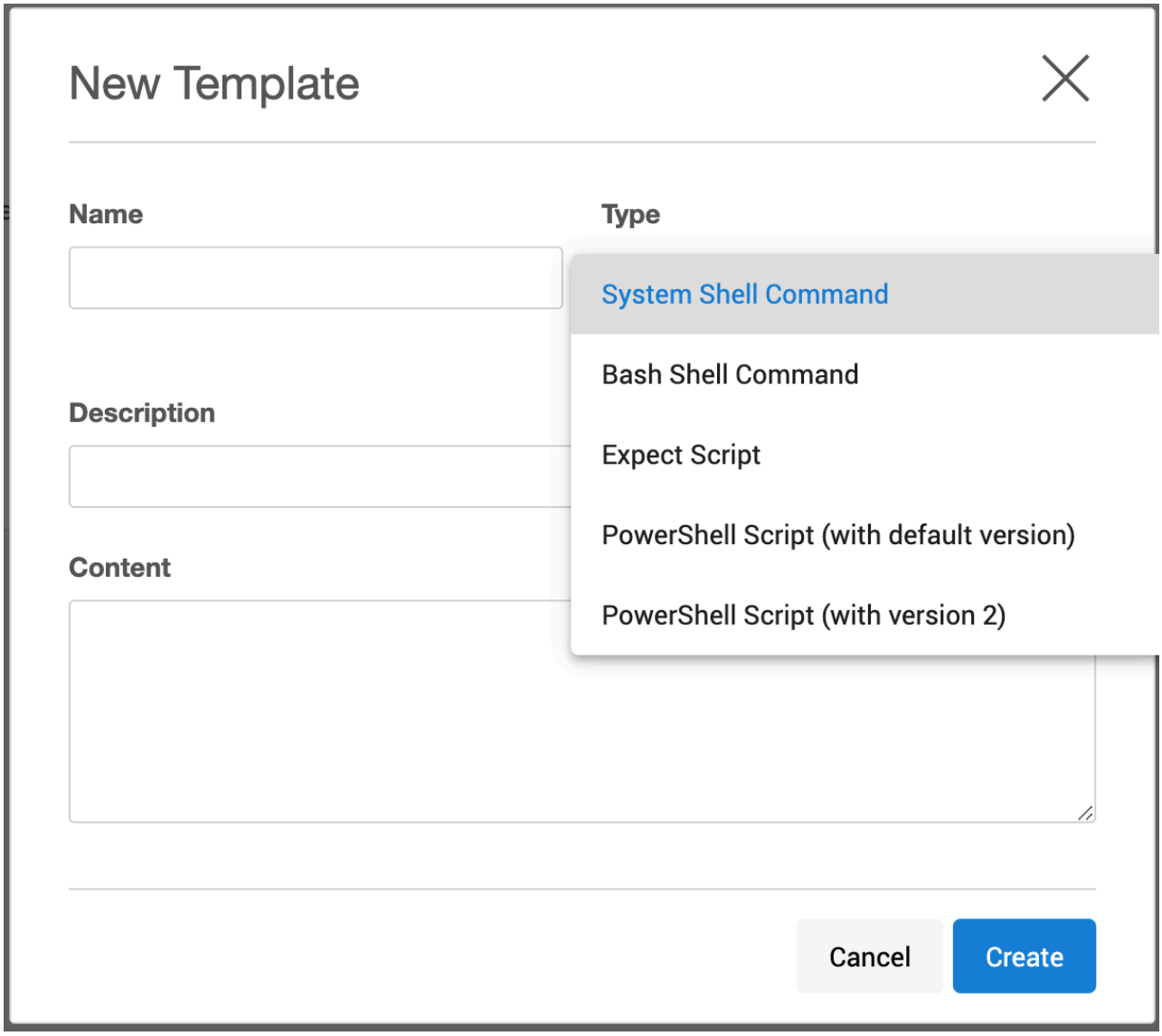
For more information on Hook Operations, see SQL server hook operation notes (see page 2209)

9.12.7.2 Hook operation templates

You can use operation templates to store commonly used operations, which allows you to avoid repeated code entry when an operation is applicable to more than a single hook, dSource, or virtual dataset. You can manage templates through the Delphix Management application.

You can also create templates from existing hooks by exporting the hooks in the Delphix Management application.

While creating a hook template, the user can provide the PowerShell version (default version or version 2) in the field "Type" and this PowerShell version will be used to execute the hook's script created from this template. Here, the default version is the version of PowerShell installed on the target host.



The existing template's PowerShell version can be changed by using Delphix CLI only because UI currently does not support this feature.

9.12.7.3 Windows environment variables

When a hook is executed, Delphix will set specific Windows environment variables to provide context, such as the name of the current host, the name of the SQL Server instance and port, and the name of the database. For more information, see SQL server hook operation notes (see page 2209)

9.12.7.4 Python script to migrate hooks from PowerShell version 2 to host's default powerShell version.

9.12.7.4.1 Overview

With 6.0.3.0, the Delphix Engine will use the default PowerShell version installed on the host (hereinafter referred as default PowerShell) to perform all its operations, also the new hooks and hook templates can be created using the default PowerShell. The existing hooks and hook templates on the engine will continue functioning using PowerShell version 2, and there will be an option to migrate them to use default PowerShell.

The article intends to introduce a Python script that can be used to migrate all the hooks and hook templates on the engine to default PowerShell. The motivation behind writing the script is to save the manual effort required in doing the migration via the UI or CLI.

migrate_mssql_hooks_and_...[652].

9.12.7.4.2 Requirements for running the script

The requirements for running the script are as follows, they are the same as running any Python script in general.

1. The machine where the script is run should have Python installed.

652 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149531033?preview=%2F149531033%2F149531052%2Fmigrate_mssql_hooks_and_hook_templates.py

   a. Relevant link: https://www.python.org/downloads/
   b. The script is supported for both Python 2 and Python 3 release.

2. The machine where the script is run should have the `delphixpy` Python package installed.
   a. Relevant link: https://pypi.org/project/delphixpy/
   b. python.org [653]documentation on installing packages and creating Python environments: Installing packages using pip and virtual environments.[654]
   c. If the package is already installed, it should be upgraded to the latest version, the minimum `delphixpy` version required to run the script is 1.11.3.0.

3. The Delphix Engine should be accessible from the machine where the script is run since the script makes API requests to the Delphix Engine to perform the migration,
   a. The easiest way to verify the same is to use the **ping** command.

9.12.7.4.3 Functionalities of the script

1. The script can migrate the hooks and hook templates in the Delphix Engine to run with default PowerShell.
2. The script can also migrate the hooks and hook templates back to PowerShell version 2.
3. It is possible to migrate only the hooks or only the hook templates.
4. If **INSTALLEDPOWERSHELL** feature flag is enabled on the Delphix Engine, the script will disable the same.

9.12.7.4.4 The script parameters and usage

Script parameter	Description	Type	Possible values	Default value
`--help`	Displays the description and usage details of the parameters for the script on the terminal	Optional	Not applicable	Not applicable
`--engine-addr`	The Delphix Engine host address.	Required	Not applicable	Not applicable
`--sys-admin-usr`	The username for System Administrator user to log into the Delphix Engine.	Required	Not applicable	Not applicable

653 https://packaging.python.org/
654 https://packaging.python.org/guides/installing-using-pip-and-virtual-environments/

Script parameter	Description	Type	Possible values	Default value
`--sys-admin-pwd`	The password for System Administrator user to log into the Delphix Engine.	Required	Not applicable	Not applicable
`--admin-usr`	The username for Engine Administrator user to log into the Delphix Engine.	Required	Not applicable	Not applicable
`--admin-pwd`	The password for Engine Administrator user to log into the Delphix Engine.	Required	Not applicable	Not applicable
`--hook-ps-version`	Migrates all the hooks to PowerShell Version two.	Optional	default, ps2	default
`--hook-templ-ps-version`	Migrates all the hook templates to PowerShell Version two.	Optional	default, ps2	default
`--migrate-only`	Migrates only the hook or only the hook templates. Input "hooks" for migrating only the hooks, whereas input "templates" for migrating only the hook templates	Optional	hooks, templates, hooks-and-templates	hooks-and-templates
`--debug`	In case there is an error executing the script, prints the Python stack trace required for debugging.	Optional	Not applicable	Not applicable

9.12.7.4.5 Command examples

1. Displaying the description and usage details of the parameters for the script on the terminal

```
python <path to the script> --help
```

2. An example command for migrating all the hook and hook templates to default PowerShell version:

```
python <path to the script> --engine-addr engine.delphix.com --sys-admin-usr
sysadmin --sys-admin-pwd sysadmin --admin-usr admin --admin-pwd delphix
```

3. An example command for migrating all the hook and hook templates to PowerShell version 2:

```
python <path to the script> --engine-addr engine.delphix.com --sys-admin-usr
sysadmin --sys-admin-pwd sysadmin --admin-usr admin --admin-pwd delphix --hook-
ps-version ps2 --hook-templ-ps-version ps2
```

Please note that this will migrate all the hooks and hook templates in the Delphix Engine to PowerShell version two, the script does not have the functionality to perform the migration for specific hooks or hook templates.

4. An example command for migrating only the hook templates:

```
python <path to the script> --engine-addr engine.delphix.com --sys-admin-usr
sysadmin --sys-admin-pwd sysadmin --admin-usr admin --admin-pwd delphix --
migrate-only templates
```

5. An example command for running the script in debug mode:

```
python <path to the script> --engine-addr engine.delphix.com --sys-admin-usr
sysadmin --sys-admin-pwd sysadmin --admin-usr admin --admin-pwd delphix --debug
```

9.12.7.4.6 Common errors while running the script:

1. If the Delphix engine is not accessible from the machine where the script is run, the connectivity test run by the script prior to running the migration will fail, and the following error will be thrown:

```
Error occurred while connecting to the engine via the given Delphix System
Administrator user:
[Errno 8] nodename nor servname provided, or not known

Error occurred while connecting to the engine via the given Engine
Administrator user:
[Errno 8] nodename nor servname provided, or not known
```

The connectivity test run by the script prior to running the migration will fail if the value of the required parameters provided is not correct, and the following error will be thrown:

```
Error occurred while connecting to the engine via the given Engine
Administrator user:
```

```
HTTP status was 401 when doing POST '{"username": "admin", "password": "blah",
"type": "LoginRequest", "target": "DOMAIN"}' to '/resources/json/delphix/login':
 {"type":"ErrorResult","status":"ERROR","error":{"type":"APIError","details":"I
nvalid username or password.","action":"Try with a different set of
credentials.","id":"exception.webservices.login.failed","commandOutput":null,"d
iagnoses":[]}}
```

2. If for some reason, the script execution stops in between before migrating all the hooks and hook templates, it's perfectly fine to run the script again; each time the script is run, the hooks and hook templates will be migrated to the specified PowerShell version.

9.12.7.5 SQL Server hook operation notes

9.12.7.5.1 SQL server clusters

When linking from, or provisioning to cluster environments, hook operations will not run once on each node in the cluster. Instead, the Delphix Engine always runs all hooks on the instance primary node.

9.12.7.5.1.1 Run powershell operation

The RunPowershell operation executes a PowerShell script on a Windows environment. The environment user runs this shell command from their home directory. The Delphix Engine captures and logs all output of the script. If the script fails, the output is displayed in the Delphix Management application and command-line interface (CLI) to aid in debugging.

If successful, the script must exit with an exit code of `0`. All other exit codes will be treated as an operation failure.

Example of a run powershell Operation

You can input the full command contents into the Run powershell operation.

```
$removedir = $Env:DIRECTORY_TO_REMOVE

if ((Test-Path $removedir) -And (Get-Item $removedir) -is [System.IO.DirectoryInfo])
{
    Remove-Item -Recurse -Force $removedir
} else {
    exit 1
}
exit 0
```

9.12.7.5.2 SQL server environment variables

Operations that run user-provided scripts have access to environment variables. For operations associated with specific dSources or virtual databases (VDBs), the Delphix Engine will always set environment variables so that the user-provided operations can use them to access the dSource or VDB.

9.12.7.5.2.1 dSource environment variables

Environment Variables	Description
`SOURCE_INSTANCE_HOST`	The hostname of linked instance for the dSource
`SOURCE_INSTANCE_PORT`	Port of linked instance for the dSource
`SOURCE_INSTANCE_NAME`	Name of linked instance for the dSource
`SOURCE_DATABASE_NAME`	Name of database linked for the dSource

9.12.7.5.2.2 Staging variables

We have the following environment variables applicable to Staging Push dSources.

Environment Variables	Description
`STAGING_INSTANCE_HOST`	The hostname of the staging instance
`STAGING_INSTANCE_PORT`	Port number of the staging instance
`STAGING_INSTANCE_NAME`	Name of the staging instance
`STAGING_DATABASE_NAME`	Name of the staging database
`STAGING_MOUNT_BASE`	Mount path for the staging push dSource

Environment Variables	Description
`STAGING_DATA_DB_FILE_PATH`	Filepath of the staging database

9.12.7.5.2.3 VDB environment variables

Environment Variables	Description
`VDB_INSTANCE_HOST`	The hostname of linked instance for the VDB
`VDB_INSTANCE_PORT`	Port of linked instance for the VDB
`VDB_INSTANCE_NAME`	Name of linked instance for the VDB
`VDB_DATABASE_NAME`	Name of database linked for the VDB

9.12.7.5.3 Error handling for SQL server powerShell scripts

If a pre-script or post-script encounters an unrecoverable error during execution, the Delphix Engine expects the script to return with a non-zero exit code or the error will not be detected. The **Powershell -File** prefix and **exit $LASTEXITCODE** suffix are required to pass the script's exit code up to the layer calling the script.

> Delphix does not perform error checking on PowerShell hook scripts. The script should perform error checking and logging, and return a non-zero exit code to indicate the script's failure. Failure to return a non-zero exit code when appropriate means that Delphix will think the hook script succeeded and mark the VDB provision/refresh/rewind job as a success, when it should be seen as a failure. This is especially important when masking data is part of the hook – the VDB should not be released to users when the hook failed to mask data.

PowerShell gives you a few ways to handle errors in your scripts:

- Set `$ ErrorActionPreference`. This only applies to PowerShell Cmdlets. For scripts or other executables such as `sqlcmd`, PowerShell will return with exit code 0 even if there is an error, regardless of the value of `$ErrorActionPreference`. The allowable values for `$ErrorActionPreference` are:

    - `Continue` (default) – Continue even if there is an error
    - `SilentlyContinue` – Acts like Continue with the exception that errors are not displayed
    - `Inquire` – Prompts the user in case of error
    - `Stop` : Stops execution after the first error
- Use exception handling by using traps and try/catch blocks to detect errors and return with non-zero exit codes
- Use custom error handling that can be invoked after launching each command in the script to correctly detect errors. The following example shows how you can use the function verifySuccess to detect whether the previous command failed, and if it did print, print an error message and return with an exit code of **1**.

```
function die {
    Write-Error "Error: $($args[0])"
    exit 1
}
function verifySuccess {
    if (!$?) {
        die "$($args[0])"
    }
}
Write-Output "I'd rather be in Hawaii"
verifySuccess "WRITE_OUTPUT_FAILED"
& C:\Program Files\Delphix\scripts\myscript.ps1
verifySuccess "MY_SCRIPT_FAILED"
```

9.12.7.6 Using pre- and post-scripts with SQL Server dSources

9.12.7.6.1 Overview

This topic describes the use of pre- and post-scripts with dSources that are created from SQL Server source databases.

Pre-scripts and post-scripts are Windows PowerShell code executed on the Staging Target host before and after a SnapSync of a dSource. You can specify pre- and post-scripts in the wizard for creating a dSource, or you can modify them afterward by navigating to the Configuration > Standard tab. You can also set pre- and post-scripts using the Delphix command-line interface (CLI) or REST Web API.

The Delphix Engine executes a pre-script on the Staging Target host prior to the SnapSync of a dSource. If it is an initial snapshot, or manual snapshot by selecting the snapshot button on the GUI, the pre/post scripts do not get executed for dSources. Since the dSource resides within a restoring database in the SQL instance on the Staging Target host, the script can perform queries on the instance if a database account is available.

Hooks do allow for a pre/post snapshot hook, but the pre/post scripts do not

A post-script is executed after the SnapSync on a dSource completes. If the post-script fails, the provision, refresh, or rewind operation will also fail with an error message, and a fault will be created on the dSource.

> Pre/Post hooks allow for pre/post snapshot hooks for which there is no equivalent in the old pre/post scripts. Pre/Post scripts are run during validated sync, for which there is no equivalent hook operation today.
> Pre- and post-scripts are supported for backward compatibility with older versions of the Delphix Engine.

9.12.7.6.2 Associating scripts with a dSource

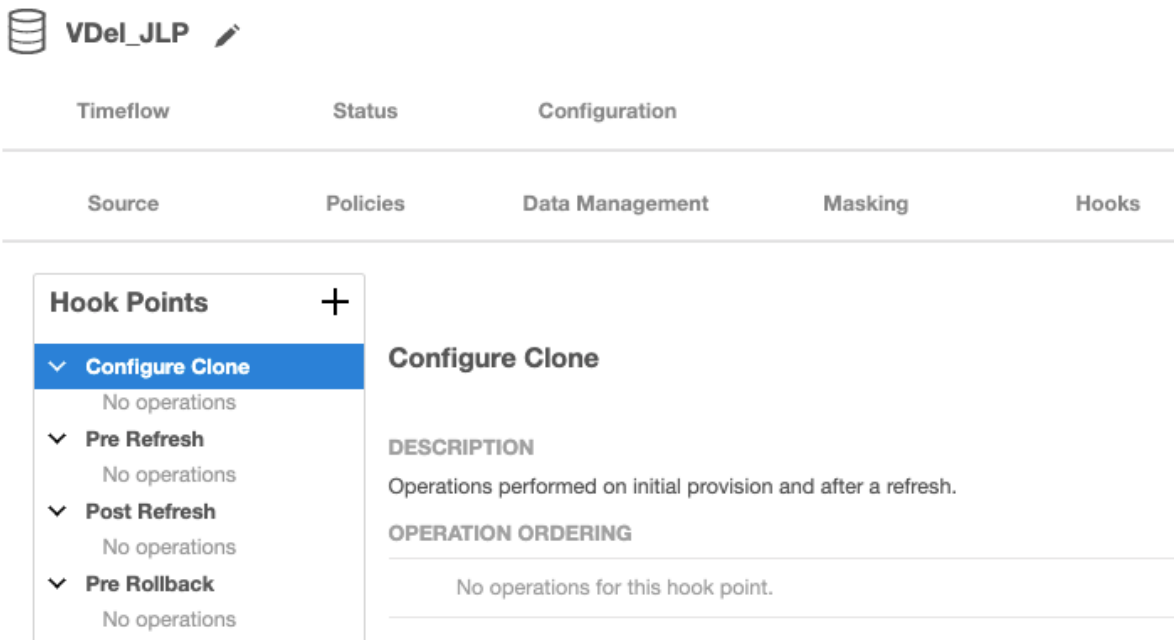
Pre- and Post-scripts can be associated with a dSource in one of two ways:

9.12.7.6.2.1 During the dSource linking process

1. Login to the **Delphix Management** application.
2. In the top menu bar, click **Manage**.
3. Select **Datasets**.
4. Select the **Plus** icon and then select **Add dSource**.
5. In the **Hooks** tab of the **Add dSource** wizard, select the Plus icon to add new **Pre Script** and **PostScript** hooks.
6. Enter the calling syntax of the Windows PowerShell script into either or both of the appropriate fields.
   a. The calling environment is used during linking, as shown in the **Environment Details** panel of the Delphix environment (**Manage** > **Environments**)
   b. Four (4) environment variables will be populated with the name of the Delphix dSource, the SQL Server instance name and port, and the SQL Server database
   c. name.

9.12.7.6.2.2 After linking, using the configuration tab of the datasets details page

1. Login to the **Delphix Management** application.
2. In the top menu bar, click **Manage**.
3. Select **Datasets** to display the SQL Server dSources and VDBs.
4. Select a SQL Server **dSource** from the listed **Datasets** in the left-hand navigation bar.
5. Click the **Configuration** panel, select the Hooks sub-tab.
6. By selecting a Plus icon you can create hooks from a template or create a new hook.



7. Enter the calling syntax of the Windows Powershell script into either or both of the appropriate fields
   a. The calling environment is that of the **primary Environment User account**, as shown in the **Environment Details** panel of the Delphix environment (**Manage** > **Environments**)
   b. Four (4) environment variables will be populated with the name of the VDB, the SQL Server instance name and port, and the SQL Server database name.
   c. Select the **Create** to accept the change.

9.12.7.6.3 Execution context for SQL server scripts

For dSources, pre- and post-scripts are executed in the context of the staging host user that was provided when linking.

9.12.7.7 Using pre- and post-scripts with SQL Server VDBs

9.12.7.7.1 Overview

This topic describes the use of pre- and post-scripts with virtual databases (VDBs) that are created from SQL Server dSources.

Pre-scripts and post-scripts are Windows PowerShell code executed on the VDB target host before and after the provision, refresh, or rewind of a VDB. You can specify pre- and post-scripts in the wizard for creating a VDB, or you can modify them afterward by navigating to the Configuration > Standard tab. You can also set pre- and post-scripts using the Delphix command-line interface (CLI) or REST Web API.

The intent of these scripts is a customization of the data contents or configuration of a VDB while it is being manipulated. Actions performed by pre-scripts and post-scripts effectively become an integrated part of the provision, refresh, or rewind actions for that VDB.

The pre-script executes during the initial provision of a VDB. During refresh and rewind, the PowerShell script referenced in a pre-script is executed after the VDB has been stopped and unmounted, but before the new VDB is mounted. If the pre-script fails, the refresh or rewind operation will also fail with an error message.

During provision, refresh, and rewind, the PowerShell script referenced in a post-script is executed after the Delphix engine has mounted and started the VDB. If the post-script fails, the provision, refresh, or rewind operation will also fail with an error message, and a fault will be created on the VDB.

You can use a pre-script to capture configuration file settings, but not the contents of the soon-to-be recreated VDB; a pre-script executes too late to access the VDB which has already been shut down and unmounted. This makes pre-script functionality much less useful than hook operations like Pre-Refresh.

You can use a post-script to run data transformation operations on newly-provisioned, newly-refreshed, or newly-rewound VDBs. These operations include data masking and setting non-production account passwords in place of cloned production passwords.

> Pre- and post-scripts are an older customization mechanism for SQL Server virtual databases. They have been replaced by hook operations, which have been the standard customization mechanism on all other data platforms.
>
> Pre- and post-scripts are supported for backward compatibility with older versions of the Delphix Engine. Delphix encourages everyone to use Hooks for customizing SQL Server VDBs for future implementations, if possible.

9.12.7.7.2 Associating scripts with a VDB

Pre- and Post-scripts can be associated with a VDB in one of two ways:

9.12.7.7.2.1 During the VDB provisioning process

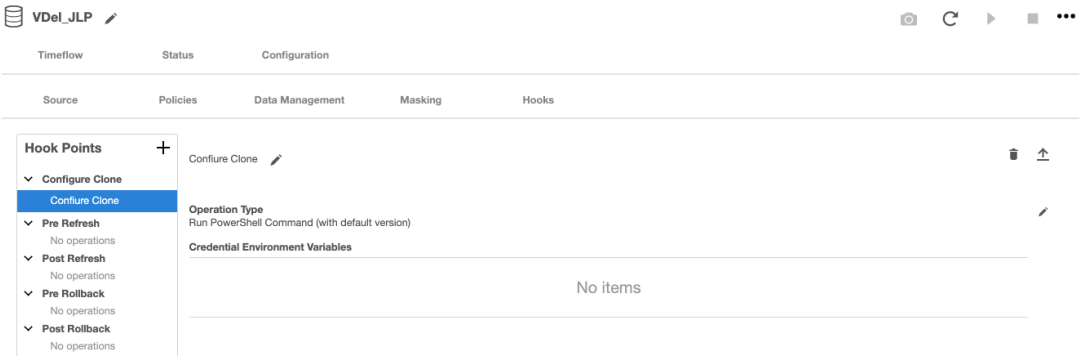
1. Login to the **Delphix Management** application/
2. In the top menu bar, click **Manage**.
3. Select **Datasets** to display the SQL Server dSources and VDBs.
4. From the listed **Datasets** in the left-hand navigation bar, select a SQL Server **dSource** or **VDB**.
5. Click the **TimeFlow**.
6. Click **Provision**.
7. In the first **Target Environment** step of the **Provision VDB** wizard, there are fields for **Pre Script** and **Post Script**.
8. Enter the calling syntax of the Windows PowerShell script into either or both of the appropriate fields.
    a. The calling environment is that of the **primary Environment User account**, as shown in the **Environment Details** panel of the Delphix environment (**Manage** > **Environments**)

b. Four (4) environment variables will be populated with the name of the VDB, the SQL Server instance name and port, and the SQL Server database name.

9.12.7.7.2.2 After provisioning, using the configuration tab of the Datasets details page

1. In the **Datasets** panel, click the virtual dataset.
2. Click the **Configuration** tab.
3. Within the **Configuration** tab, click the **Hooks** tab.
4. Select the **hook** to edit.
   a. Click the **Plus** icon to add a new operation.
   b. Select the **type of operation** or click

      to load a hook operation template.
   c. Click the **text area** and edit the contents of the operation.
   d. To remove an operation from the list, click the **Trash** icon on the operation.
   e. When you have set all hook operations, click the **checkmark** to save the changes.The current operations at this hook will be displayed. To edit this list of operations, click the **Pencil** icon in the top right-hand corner of the tab.



9.12.7.7.3 Execution context for SQL server scripts

For VDBs, pre- and post-scripts are executed in the context of the environment user that was selected during the VDB provision and not the primary environment user. The primary environment user can change over time, all VDB operations are done using the user that was initially selected.

9.12.7.7.4 Error handling for SQL server powerShell scripts

If a pre-script or post-script encounters an unrecoverable error during execution, the Delphix Engine expects the script to return with a non-zero exit code. Otherwise, the error will not be detected.

PowerShell gives you a few ways to handle errors in your scripts:

- Set `undefinedErrorActionPreference`. The allowable values for `$ErrorActionPreference` are:

    - `Continue` (default) – Continue even if there is an error
    - `SilentlyContinue` – Acts like Continue with the exception that errors are not displayed
    - `Inquire` – Prompts the user in case of error
    - `Stop` : Stops execution after the first error
- Use exception handling by using traps and try/catch blocks to detect errors and return with non-zero exit codes
- Use custom error handling that can be invoked after launching each command in the script to correctly detect errors. The following example shows how you can use the function verifySuccess to detect whether the previous command failed, and if it did print, print an error message and return with an exit code of **1**.

```
function die {
    Write-Error "Error: $($args[0])"
    exit 1
}
function verifySuccess {
    if (!$?) {
        die "$($args[0])"
    }
}
Write-Output "I'd rather be in Hawaii"
verifySuccess "WRITE_OUTPUT_FAILED"
& C:\Program Files\Delphix\scripts\myscript.ps1
verifySuccess "MY_SCRIPT_FAILED"
```

9.12.8 Using hostChecker to validate target database servers

9.12.8.1 Prerequisites

Make sure that your source and target environments meet the requirements.

9.12.8.2 Procedure to validate target environments

1. Verify with your System Administrator that the Delphix Connector has been installed in all Target environments.
2. Login to the Windows target host using the **Windows user account** that the System Administrator configured as a Delphix target user.
3. Open **Windows Powershell** using the **Run as Administrator** option.
4. Execute the **host checker script**by running:

```
<Delphix Connector installation folder>\etc\dlpx-host-checker.ps1
```

5. Select a path where a report file will be saved, such as **C:\temp\delphix-host-checker-report.txt**.
6. Select the **default** option of **Target Host**.
7. Read the output of the checks.
8. The error or warning messages will explain any possible problems and how to address them. Resolve the issues that the HostChecker describes. Do not be surprised or undo your work if more errors appear the next time you run HostChecker; the error you just fixed may have been masking other problems.
9. Repeat steps 4–7 until all the checks return no errors or warnings.

9.12.8.3 Tests run

Test	SQL Server Source	SQL Server Target	Description
Check Powershell Version	X	X	Verifies that Powershell 2.0 or greater is installed
Check OS User Privileges	X	X	For target hosts, verifies that the operating system (OS) user has administrative rights. For source hosts, verifies that the OS user can successfully perform remote registry access from the target host to the source host.
Check host settings	X	X	Verifies that the Delphix Engine can discover host environment details from the Windows registry.
Check SQL Server instance discovery	X	X	Verifies that the Delphix Engine can discover SQL Server instances.
Check SQL Server instance login permission	X	X	For target hosts, verifies that the Windows OS user can be used to log in to the SQL Server instances. For source hosts, verifies that the supplied SQL Server login credentials can be used to log in to the SQL Server instances.
Check database discovery	X	X	Verifies that the Delphix Engine can discover SQL Server databases.

9.12.8.4 Additional options

Run the following to view additional HostChecker options:

```
dlpx-host-checker.ps1 -?
```

9.13 Unstructured files and app data

This section contains the following topics:

- Getting started with unstructured files (see page 2219)
- Unstructured files support and requirements (see page 2219)
- Create an empty VDB for unstructured files in the Delphix Engine (see page 2251)
- Provisioning unstructured files as vFiles (see page 2252)
- Managing vFiles (see page 2254)
- vFiles best practices and common pitfalls (see page 2255)
- Delphix Engine plugin management (see page 2257)
- Unstructured files hook operation notes (see page 2263)

9.13.1 Getting started with unstructured files

The term “unstructured files" refers to data stored in a filesystem that is NOT usually accessed by a DBMS or similar software. Unstructured files can consist of anything from a simple directory to the root of a complex application like Oracle Enterprise Business Suite. Like with other data types, you can configure a dSource to sync periodically with a set of unstructured files external to the Delphix Engine. The dSource is a copy of these physical files stored on the Delphix Engine. On Unix platforms, dSources are created and periodically synced by an implementation of the rsync utility. On Windows, files are synced using the robocopy utility, which is distributed with Windows. dSources enable you to provision “vFiles,” which are virtual copies of data that are fully functional read-write copies of the source of the original file. You can mount vFiles across one target environment or many.

9.13.2 Unstructured files support and requirements

- vFiles matrix (see page 2220)
- Unstructured files environment requirements (see page 2226)

9.13.2.1 vFiles matrix

9.13.2.1.1 Unix environments

> Delphix Support Policies specifically list Major and Minor release coverage. If a minor release is listed as covered, then all patch releases under that minor release are certified.

Color	Supported?
Y	Yes

9.13.2.1.1.1 Red Hat Enterprise Linux (RHEL)

Supported OS Version	6.0+
	Delphix AppData Version (x86_64)
RHEL 5.5	Y
RHEL 5.6	Y
RHEL 5.7	Y
RHEL 5.8	Y
RHEL 5.9	Y
RHEL 5.10	Y
RHEL 5.11	Y
RHEL 6.0	Y

Supported OS Version	6.0+
RHEL 6.1	Y
RHEL 6.2	Y
RHEL 6.3	Y
RHEL 6.4	Y
RHEL 6.5	Y
RHEL 6.6	Y
RHEL 6.7	Y
RHEL 6.8	Y
RHEL 6.9	Y
RHEL 6.10	Y
RHEL 7.0	Y
RHEL 7.1	Y
RHEL 7.2	Y
RHEL 7.3	Y
RHEL 7.4	Y
RHEL 7.5	Y
RHEL 7.6	Y
RHEL 7.7	Y

Supported OS Version	6.0+
RHEL 7.8	Y
RHEL 7.9	Y 6.0.7+
RHEL 8.0	Y
RHEL 8.1	Y 6.0.3+
RHEL 8.2	Y 6.0.3+
RHEL 8.3	Y 6.0.8+
RHEL 8.4	Y 6.0.11+
RHEL 8.5	Y 6.0.14+
RHEL 8.6	Y 6.0.15+
RHEL 8.7	Y 13.0.0+
RHEL 8.8	Y 13.0.0+
RHEL 8.9	Y 19.0.0+
RHEL 9.0	Y 20.0.0+
RHEL 9.3	Y 20.0.0+

9.13.2.1.1.2 Oracle Enterprise Linux

Supported OS Version	6.0+
	Delphix AppData Version (x86_64)
Oracle Enterprise Linux 5.5	Y

Supported OS Version	6.0+
Oracle Enterprise Linux 5.6	Y
Oracle Enterprise Linux 5.7	Y
Oracle Enterprise Linux 5.8	Y
Oracle Enterprise Linux 5.9	Y
Oracle Enterprise Linux 5.10	Y
Oracle Enterprise Linux 5.11	Y
Oracle Enterprise Linux 6.0	Y
Oracle Enterprise Linux 6.1	Y
Oracle Enterprise Linux 6.2	Y
Oracle Enterprise Linux 6.3	Y
Oracle Enterprise Linux 6.4	Y
Oracle Enterprise Linux 6.5	Y
Oracle Enterprise Linux 6.6	Y
Oracle Enterprise Linux 6.7	Y
Oracle Enterprise Linux 6.8	Y
Oracle Enterprise Linux 6.9	Y
Oracle Enterprise Linux 6.10	Y
Oracle Enterprise Linux 7.0	Y

Supported OS Version	6.0+
Oracle Enterprise Linux 7.1	Y
Oracle Enterprise Linux 7.2	Y
Oracle Enterprise Linux 7.3	Y
Oracle Enterprise Linux 7.4	Y
Oracle Enterprise Linux 7.5	Y
Oracle Enterprise Linux 7.6	Y
Oracle Enterprise Linux 7.7	Y
Oracle Enterprise Linux 7.8	Y
Oracle Enterprise Linux 7.9	Y 6.0.7+
Oracle Enterprise Linux 8.0	Y
Oracle Enterprise Linux 8.1	Y
Oracle Enterprise Linux 8.2	Y 6.0.5+

9.13.2.1.2 SUSE Linux Enterprise Server (SLES)

Delphix AppData Version	6.0+
Supported OS Version (x86_64)	
SLES 11	Y
SLES 11 SP1	Y

9.13.2.1.3 Solaris

Delphix AppData Version	6.0+
Supported OS Version (SPARC x86_64)	
Solaris 10	Y
Solaris 11	Y

9.13.2.1.4 AIX

Delphix AppData Version	6.0+
Supported OS Version (Power)	
AIX 6.1	Y
AIX 7.1	Y
AIX 7.2	Y

9.13.2.1.5 HP-UX

Delphix AppData Version	6.0+
Supported OS Version (IA64)	
HP-UX 11i v3 (11.31)	Y

9.13.2.1.6 Windows environments

Delphix AppData Version	6.0+
Supported OS Version	
Windows Server 2012	Y
Windows Server 2012 R2	Y
Windows Server 2016	Y
Windows Server 2019	Y
Windows Server 2022	Y

9.13.2.2 Unstructured files environment requirements

This section contains the following topics:

- Unstructured files on unix environments (see page 2226)
- Unstructured files on windows environments (see page 2237)
- Linking unstructured files (see page 2249)

9.13.2.2.1 Unstructured files on unix environments

This section contains the following topics:

- Requirements for Unix environments (see page 2226)
- Network and connectivity requirements for Unix environments (see page 2229)
- Sudo privilege requirements for unstructured files on Unix (see page 2233)
- Sudo file configuration examples for unstructured files on Unix (see page 2233)
- Adding a Unix environment (see page 2236)

9.13.2.2.1.1 Requirements for Unix environments

This topic outlines the supported operating systems (OSs) for use on UNIX source and target environments.

Supported operating systems

Operating System	Version	Processor Family
Solaris	10, 11	SPARC x86_64
Red Hat Enterprise Linux	5.5 - 5.11 6.1 - 6.10 7.0 - 7.8 8.0	x86_64
Oracle Enterprise Linux	5.5 - 5.11 6.1 - 6.10 7.0 - 7.8 8.0	x86_64
SUSE Linux Enterprise Server	11, 11SP1	x86_64
AIX	6.1, 7.1, 7.2	Power
HP-UX	11i v3 (11.31)	IA64
Ubuntu	18, 20	X86_64

Delphix supports all 64-bit OS environments for source and target.

PHNE_37851 - resolves a known bug in HP-UX NFS client prior to HP-UX 11.31.

Additional source environment requirements

- There must be an operating system user with these privileges. For example, here in this section delphix_os is the primary user for the environment:
  - Ability to login to the source environment via SSH
- There must be a directory on the source environment where you can install the Delphix platform Toolkit – for example, `/var/opt/delphix/toolkit`.
  - The **delphix_os** user must own the directory
  - The directory must have permissions -rwxrwx--- (0770), but you can also use more permissive settings

    - The **delphix_os** user must have read and execute permissions on each directory in the path leading to the toolkit directory. For example, when the toolkit is stored in `/var/opt/delphix/toolkit`, the permissions on `/var`, `/var/opt`, and `/var/opt/delphix` should allow read and execute for "others," such as -rwxr-xr-x.
    - The directory should have 1.5GB of available storage: 400MB for the toolkit and 400MB for the set of logs generated by each client that runs out of the toolkit
- On a Solaris host, `gtar` must be installed. Delphix uses `gtar` to handle long file names when extracting the toolkit files into the toolkit directory on a Solaris host. The `gtar` binary should be installed in one of the following directories:
    - `/bin:/usr`
    - `/bin:/sbin:/usr`
    - `/sbin:/usr/contrib`
    - `/bin:/usr/sf`
    - `/bin:/opt/sfw`
    - `/bin:/opt/csw/bin`
- The Delphix Engine must be able to initiate an SSH connection to the source environment

Additional target environment requirements

- There must be an operating system user with these privileges. Here we use delphix_os as an example for the primary user for the environment:
    - Ability to login to the target environment via SSH
    - The following permissions are usually granted via sudo authorization of commands.
    - See Sudo Privilege Requirements (see page 2233) for further explanation of the commands and Sudo File Configuration Examples for Unstructured Files on Unix (see page 2233) for examples of the /etc/sudoers file on different operating systems.
        - The primary user for the target environment must have the ability to run mount, unmount, mkdir, and rmdir as a super-user on any directory mounted to by the Delphix Engine.
        - If the target host is an AIX system, permission to run the nfso command as a super-user
    - There must be a directory on the target environment where you can install the Delphix Engine Toolkit - for example, /var/opt/delphix/toolkit.
        - The **delphix_os** user must own the directory
        - The directory must have permissions -rwxrwx--- (0770), but you can also use more permissive settings
        - The **delphix_os** user must have read and execute permissions on each directory in the path leading to the toolkit directory. For example, when the toolkit is stored in `/var/`

`opt/delphix/toolkit`, the permissions on `/var`, `/var/opt`, and `/var/opt/delphix` should allow read and execute for "others," such as -rwxr-xr-x.

- The directory should have a total of at least 800MB of storage, plus 1MB of storage per vFile that will be provisioned to the target

- On a Solaris host, `gtar` must be installed. Delphix uses `gtar` to handle long file names when extracting the toolkit files into the toolkit directory on a Solaris host. The `gtar` binary should be installed in one of the following directories:
  - `/bin:/usr`
  - `/bin:/sbin:/usr`
  - `/sbin:/usr/contrib`
  - `/bin:/usr/sf`
  - `/bin:/opt/sfw`
  - `/bin:/opt/csw/bin`
- There must be a directory under which the mount points are created when provisioning a vFile to the target environment. The group associated with the directory must be the primary group of the delphix-os user. Group permissions for the directory should allow read, write, and execute by members of the group.
- The Delphix Engine must be able to initiate an SSH connection to the target environment.
- NFS client services must be running on the target environment.
  - Required packages on target hosts:
    i. Portmapper (rpcbind)
    ii. status daemon (rpc.statd)
    iii. NFS lock manager (rpc.lockd/lockmgr)
  - The Delphix Engine enables the use of NFSv3 by default for mounting target host filesystems, thus, the prerequisite packages to support NFSv3 client communication are required for normal operation. In addition, the required services to support NFS client communications (including file locking) must be running. These services are shown in the left column.
    - To enable NFSv4, which does not need to interact with those discrete services, See NFSv4 configuration (see page 746).[655]

9.13.2.2.1.2 Network and connectivity requirements for Unix environments

Port allocations specific to unstructured files

The Delphix Engine makes use of the following network ports for unstructured files dSources and vFiles:

Inbound to the Delphix Engine port allocation

655 http://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/nfsv4-configuration

Protocol	Port Number	Use
TCP	873	Rsync connections used for communication to rsyncd[656] during SnapSync
TCP/UDP	111	Remote Procedure Call (RPC) port mapper used for NFSv3 mounts Note: RPC calls in NFSv3 use additional fixed ports for supporting services (lockd, mountd and statd) seen below.
TCP	1110	NFS Server daemon status and NFS server daemon keep-alive (client info)
TCP	2049	NFS Server daemon from vFiles to the Delphix Engine
TCP	54043	NFSv3 mount daemon
TCP	54044	NFSv3 stat daemon (lock state notification service)
TCP	54045	NFSv3 lock daemon/manager
TCP	54046	Connections from Source and Target Environments to the Engine When NFS Encryption is enabled
UDP	33434 - 33464	Traceroute from source and target database servers to the Delphix Engine (optional)

Outbound from a source environment port allocation

Protocol	Port Numbers	Use
TCP	873	Rsync connections to rsyncd[657] used during SnapSync
TCP	xxxx	DSP connections used for monitoring and script management during SnapSync. Typically DSP runs on port 8415.

Inbound to a source environment port allocation

656 https://manpages.ubuntu.com/manpages/focal/man5/rsyncd.conf.5.html
657 https://manpages.ubuntu.com/manpages/focal/man5/rsyncd.conf.5.html

Protocol	Port Numbers	Use
TCP	22	SSH connections to the source environment

Outbound from a target environment port allocation

Protocol	Port Numbers	Use
TCP	873	Rsync connections to rsyncd[658] used during V2P
TCP	xxxx	DSP connections used for monitoring and script management. Typically DSP runs on port 8415.

Inbound to a target environment port allocation

Protocol	Port Numbers	Use
TCP	22	SSH connections to the target environment

General outbound from the Delphix Engine port allocation

Protocol	Port Numbers	Use
TCP	25	Connection to a local SMTP server for sending email
TCP/UDP	53	Connections to local DNS servers
UDP	123	Connection to an NTP server
UDP	162	Sending SNMP TRAP messages to an SNMP Manager
TCP	443	HTTPS connections from the Delphix Engine to the Delphix Support upload server

658 https://manpages.ubuntu.com/manpages/focal/man5/rsyncd.conf.5.html

Protocol	Port Numbers	Use
TCP/UDP	636	Secure connections to an LDAP server
TCP	8415	Connections to a Delphix replication target. See Configuring Replication (see page 2381).[659]
TCP	50001	Connections to source and target environments for network performance tests.

General inbound to the Delphix Engine port allocation

Protocol	Port Number	Use
TCP	22	SSH connections to the Delphix Engine
TCP	80	HTTP connections to the Delphix GUI
UDP	161	Messages from an SNMP Manager to the Delphix Engine
TCP	443	HTTPS connections to the Delphix Management Application
TCP	8415	Delphix Session Protocol connections from all DSP-based network services including Replication, SnapSync for Oracle, V2P, and the Delphix Connector.
TCP	50001	Connections from source and target environments for network performance tests via the Delphix CLI.

Firewalls and Intrusion Detection Systems (IDS)

Production databases on source environments (for dSources) are often separated from the non-production environment by firewalls. Firewalls can add milliseconds to the latency between servers. Accordingly, for best performance, there should be no firewalls between the Delphix Engine and the virtual database (VDB) target environments. If the Delphix Engine is separated from a source environment by a firewall, the firewall must be configured to permit network connections between the Delphix Engine and the source environments for the application protocols (ports) listed above.

659 http://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/configuring-replication

Intrusion detection systems (IDSs) should also be made permissive to the Delphix Engine deployment. IDSs should be made aware of the anticipated high volumes of data transfer between dSources and the Delphix Engine.

9.13.2.2.1.3 Sudo privilege requirements for unstructured files on Unix

This topic describes `sudo` file privilege configurations necessary for interacting with the Delphix Engine when virtualizing unstructured files on Unix Environments.

Sudo privilege rationale

Privilege	Sources	Targets	Rationale
`mkdir/rmdir`	Not Required	Optional	Delphix dynamically makes and removes directories under the provisioning directory during vFiles operations. This privilege is optional, provided the provisioning directory permissions allow the **delphix_os** user to make and remove directories.
`mount/umount`	Not Required	**Required**	Delphix dynamically mounts and unmounts directories under the provisioning directory during vFiles operations. This privilege is required because `mount` and `umount` are typically reserved for a super-user.
`nfso` (AIX only)	Not Required	**Required**	Delphix monitors NFS read and write sizes on an AIX target host. It uses the `nfso` command to query the sizes in order to optimize NFS performance for vFiles running on the target host. Only a super-user can issue the `nfso` command.

9.13.2.2.1.4 Sudo file configuration examples for unstructured files on Unix

This topic describes `sudo` file privilege configurations necessary for interacting with the Delphix Engine when virtualizing unstructured files on Unix Environments.

Considerations for sudo access and account locking

> The Delphix Engine tests its ability to run the `mount` command using `sudo` on the target environment by issuing the `sudo mount` command with no arguments. Many of the examples shown in this topic do not allow that. This causes a warning during environment discovery and monitoring but otherwise does not cause a problem. If your vFiles operations succeed, it is safe to ignore this warning.
> However, some users configure the security on the target environments to monitor `sudo` failures and lockout the offending account after some threshold. In those situations, the failure of the sudo commands might cause the **delphix_os** account to become locked. One workaround for this situation is to increase the threshold for locking out the user account. Another option is to modify `/etc/sudoers` to permit the **delphix_os** user to run the `mount` command without parameters.

Configuring `sudo` access on Solaris for unstructured files

On a Solaris SPARC target, `sudo` access to `mount`, `umount`, `mkdir`, and `rmdir` is required. In this customer example, super-user privileges are restricted to the virtual dataset mount directory /delphix and are further restricted to commands which mount data from a single Delphix Engine with IP address 100.245.235.12.

Delphix requires `umount -f` for emergency force unmounts on Solaris.

Additionally, sudo access to `ps` may be added to avoid warnings during discovery but is not required.

Example: Solaris /etc/sudoers entries for a Delphix Target for Unstructured Files

```
User_Alias DELPHIX_USER=delphix_os

Cmnd_Alias DELPHIX_CMDS= \
/usr/sbin/mount     100.245.235.12\:* /delphix/*, \
/usr/sbin/mount * 100.245.235.12\:* /delphix/*, \
/usr/sbin/umount    /delphix/*, \
/usr/sbin/umount *  /delphix/*, \
/usr/sbin/umount -f /delphix/*, \
/usr/bin/mkdir      /delphix/*, \
/usr/bin/mkdir -p   /delphix/*, \
/usr/bin/rmdir      /delphix/*
/usr/bin/ps
DELPHIX_USER ALL=(ALL) NOPASSWD: DELPHIX_CMDS
```

Configuring `sudo` access on Linux for unstructured files

On a Linux target, sudo access to `mount`, `umount`, `mkdir`, and `rmdir` is required. In this customer example, super-user privilege is restricted to the virtual database mount directory `/delphix`. Aliases are used to restrict the Delphix Engines which are allowed to run these commands.

Delphix requires `umount -lf` for emergency force unmounts on Linux.

Example: Linux /etc/sudoers file for a Delphix Target for Unstructured Files

```
Defaults:delphix_os !requiretty

Cmnd_Alias DELPHIX_ADMIN_CMDS= \
/bin/mount           /delphix/*, \
/bin/mount   *       /delphix/*, \
/bin/umount          /delphix/*, \
/bin/umount  *       /delphix/*, \
/bin/umount  -lf     /delphix/*, \
/bin/mkdir -p -m 755 /delphix/*, \
/bin/mkdir -p        /delphix/*, \
/bin/mkdir           /delphix/*, \
/bin/rmdir           /delphix/*
/bin/ps
Host_Alias DELPHIX_HOSTS=delphix001, delphix002
delphix_os DELPHIX_HOSTS=NOPASSWD:DELPHIX_ADMIN_CMDS
```

Configuring `sudo` access on AIX for unstructured files

In addition to sudo access to the `mount`, `umount`, `mkdir`, and `rmdir` commands on AIX target hosts, Delphix also requires sudo access to `nfso`. This is required on target hosts for Delphix to monitor the NFS read/write sizes configured on the AIX system. The super-user access level is needed to run the `nfso` command. This example does not restrict the Delphix Engine which is allowed to run these commands.

Delphix requires `umount -f` for emergency force unmounts on AIX.

Example: AIX /etc/sudoers File for a Delphix Target for Unstructured Files

```
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD: \
/bin/mount, \
/bin/umount, \
/bin/mkdir, \
/bin/rmdir, \
/usr/sbin/nfso, \
/usr/bin/ps
```

Configuring `sudo` access on HP-UX for unstructured files

On the HP-UX target, as with other operating systems, `sudo` access to `mount`, `umount`, `mkdir`, and `rmdir` is required. This example does not restrict the Delphix Engine which are allowed to run these commands.

Example: HP-UX /etc/sudoers file for a Delphix Target for Unstructured Files

```
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD:/sbin/mount, /sbin/umount, /bin/mkdir, /bin/rmdir, /bin/ps
```

9.13.2.2.1.5 Adding a Unix environment

This topic describes how to add a new Unix environment.

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Environments**.
4. Click the **Plus** icon next to **Environments**.
5. In the **Add Environment** dialog, select **Unix/Linux**.
6. Select **Standalone Host**.
7. Enter the **Host IP address**.
8. Enter an optional **Name** for the environment.
9. Enter the **SSH** port. The default value is **22**.
10. Enter a **Username** for the environment.
11. Select **Login Type**.
    a. Password - enter the OS password associated with the user in Step 10, or
    b. Public Key, or
    c. Password Vault - select from an existing Enterprise Password Vault

    Note:
    Using Public Key Authentication
    If you want to use public-key authentication for logging into your Unix-based environment, there are two options: use the engine's key pair or provide a key pair for this environment.
    To use the engine's key pair:
    i. Select **Public Key** for the **Login Type.**
    ii. Click **View Public Key**.
    iii. Copy the public key that is displayed, and append it to the end of your `~/.ssh/authorized_keys` file. If this file does not exist, you will need to create it.
        1. Run chmod 600 ~/.ssh/authorized_keys to allow only the file's owner to read and write to it (make sure the file is owned by the user).
        2. Run chmod 755 ~ to restrict access to the user's home directory so no other user may write to it.
        3. Run chmod 700 ~/.ssh so that others cannot write to it. The ~/.ssh directory cannot be writable by group or other users. Otherwise, authentication will fail.

    As an alternative, you can provide a key pair specific for this environment via the API or CLI.
12. For **Password Login**, click **Verify Credentials** to test the username and password.
13. Enter a **Toolkit Path**. The toolkit directory stores scripts used for Delphix Engine operations. It should have a persistent working directory rather than a temporary one.

14. Click **Submit**.

Post-Requisites

After you create the environment, you can view information about it by doing the following:

1. Click **Manage**.
2. Select **Environments**.
3. Select the **environment name**.

9.13.2.2.2 Unstructured files on windows environments

This section contains the following topics:

- Requirements for Windows environments (see page 2237)
- Windows iSCSI configuration requirements (see page 2240)
- Network and connectivity requirements for Windows environment (see page 2244)
- Adding a windows environment (see page 2247)
- Options for linking unstructured files on Windows environments (see page 2248)

9.13.2.2.2.1 Requirements for Windows environments

Supported operating systems

- Windows Server 2012, 2012 R2
- Windows Server 2016
- Windows Server 2019
- Windows Server 2022

> **Requires 64-Bit Windows**
>
> Delphix must install the Delphix Connector on all Windows hosts that Delphix will directly communicate with. This means all target hosts, and source or staging hosts. The Delphix Connector only supports 64-bit versions of Windows.
>
> See Options for linking unstructured files on Windows environments (see page 2248) for more information about source vs. staging hosts.

Additional source or staging environment requirements

- The Delphix Connector must be installed on the source or staging environment. You must have used the Delphix Connector to register this environment with the Delphix Engine.

- The `robocopy` utility must be installed on the source or staging Windows environment. `robocopy` is installed by default on Windows Server 2008, Windows Vista, Windows 7, and Windows 8. For other versions of Windows, it is available by downloading a resource kit from Microsoft.
- If using a staging environment, the source's files must be made available and readable to the environment user from the staging environment via a UNC path. For example, use Windows Sharing.

Additional target environment requirements

- The Delphix Connector must be installed on the target environment. You must have used the Delphix Connector to register this environment with the Delphix Engine.

Procedure for adding and installing the Delphix connector for Windows

All Windows environments that will communicate with Delphix must have the Delphix Connector installed. The instructions in this topic cover downloading Delphix Connector, running the Delphix Connector installer on the Windows machine, and then registering the environment with the Delphix Engine.

Procedure

Downloading the Delphix connector

> Delphix Connector software supplied by Delphix Engine versions before 4.2.4.0 required that the Windows machine had SQL Server installed. If you are using a Windows machine that does not have SQL Server installed, you must download the Delphix Connector from a Delphix Engine of version 4.2.4.0 or higher.

The Delphix Connector can be downloaded through the Delphix Engine Interface, or by directly accessing its URL.

Using the Delphix Engine interface

> A Flash player must be available on the Windows host in order to download Delphix Connector using the Delphix GUI.

1. From the Windows machine that you want to use, start a browser session and connect to the **Delphix Management** application using the delphix_admin login.
2. Click **Manage**.

3. Select **Environments**.
4. Next to **Environments**, click the **Plus** icon.
5. In the **Add Environment** dialog, select **Windows** in the operating system menu.
6. Select **Target**.
7. Select **Standalone**.
8. Click the download link for the **Delphix Connector Installer**. The Delphix Connector will download to your local machine.

Direct download

1. You can download the Delphix Connector directly by navigating to this URL: `http://<name of your Delphix Engine>/connector/DelphixConnectorInstaller.exe`

Installing Delphix connector

On the Windows machine that you want to want to use, run the Delphix Connector installer. Click **Next** to advance through each of the installation wizard screens.

> The installer will only run on 64-bit Windows systems. 32-bit systems are not supported.

1. For **Connector Configuration**, make sure there is no firewall in your environment blocking traffic to the port on the Windows environment that the Delphix Connector service will listen to.
2. For **Select Installation Folder**, either accept the default folder or click **Browse** to select another.
3. Click **Next** on the installer final **Confirm Installation** dialog to complete the installation process and then **Close** to exit the Delphix Connector Install Program.
4. Note: At this point, you can close the Delphix GUI dialog by clicking **Cancel**.

Registering environment with Delphix Engine

1. Return to the Delphix Management application.
2. Enter the **Environment Name**, **Host Address**, **Delphix Connector Port**, **OS Username**, and **OS Password** for the target environment.
3. To provide your own Oracle Java select the **Provide my own JDK** checkbox and click **Next**.
4. In the Java Development Kit tab, enter the absolute path to your Oracle JDK and click **Next**.
5. Click **Submit.**

As the new environment is added, you will see two jobs running in the **Delphix Admin Job History**, one to **Create and Discover** an environment, and another to **create** an environment. When the jobs are complete, you will see the new environment added to the list in the **Environments** panel.

Post-Requisites

- On the Windows machine, in the **Windows Start Menu**, go to **Services > Extended Services**, and make sure that the **Delphix Connector** service has a **Status** of **Started**, and that the **Startup Type** is **Automatic**.

9.13.2.2.2.2 Windows iSCSI configuration requirements

Windows iSCSI configuration requirements are split into two types. These requirements are needed on both staging and target servers.

1. iSCSI configuration required for operational stability.
2. Optional iSCSI parameters for performance improvement.

> When target environments are discovered, Delphix will configure the Microsoft iSCSI Initiator Service for Automatic startup.

iSCSI configuration required for operational stability

The following Microsoft iSCSI Initiator configuration parameters are required for the target and staging Hosts. For details about configuring registry settings, see How to Modify the Windows Registry[660].

> You must reboot the Windows server reboot after changing the iSCSI configuration parameters.

Registry Key	Registry Value	Type	Data
`HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\iSCSI\Discovery`	MaxRequestHoldTime	REG_DWORD	0x384 (900)

660 http://support.microsoft.com/kb/136393

Registry Key	Registry Value	Type	Data
`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Disk`	TimeOutValue	REG_DWORD	0x384 (900)
`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4D36E97B-E325-11CE-BFC1-08002BE10318}\<Instance Number>\Parameters`	MaxRequestHoldTime	REG_DWORD	0x12C (300)

These settings will improve operational stability for VDBs and staging databases. If these settings are not adjusted, SQL Server may raise errors if VDBs are accessed during a temporary infrastructure outage. Affected VDBs may need to be manually restarted using the Continuous Data Engine.

Delphix Knowledge Base article KB1251[661] includes scripts to validate or set registry parameters so that they meet current Delphix recommendations.

Optional iSCSI parameters for performance improvement

The following iSCSI Registry setting may improve SQL Server dSource and VDB performance on the staging and target hosts.

Registry Key	Registry Value	Type	Data
`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\<Interface GUID>`	TcpAckFrequency	REG_DWORD	0x1 (1)

This setting is recommended for storage networks in Microsoft's TechNet article iSCSI and the Nagle Algorithm[662], described in Microsoft's document TcpAckFrequency to control the TCP ACK behavior[663].

In some environments, adjusting this setting may not improve performance compared to Windows defaults. Modifications to this registry parameter should be tested in each environment, to confirm that this provides a performance improvement.

Delphix Engine validation for Windows iSCSI configuration

661 https://support.delphix.com/Continuous_Data_Engine_(formerly_Virtualization_Engine)/MSSQL_Server/Registry_Settings_for_Optimal_Database_Performance_and_Stability_(KBA1251)
662 https://social.technet.microsoft.com/wiki/contents/articles/7636.iscsi-and-the-nagle-algorithm.aspx
663 https://docs.microsoft.com/en-us/troubleshoot/windows-server/networking/registry-entry-control-tcp-acknowledgment-behavior

Delphix Engine validates the Windows iSCSI Configurations that are set on any supported windows staging and target host with the Delphix recommended configurations while performing the following operations:

1. Add environment operation
2. Refresh environment operation
3. Enable environment operation

Prerequisites

1. Supported if you are using Powershell 3.0 or above - If you are on Powershell version below 3.0, then the job will be updated with a warning that the Powershell version on your host is not supported for validating iSCSI parameters.
2. The below alerts are applicable only for staging or target Windows hosts.

Additional Information

1. Delphix Engine will only validate and will not alter any configuration in the user environment.
2. On update of registry values on the target host to match Delphix recommendations, the faults from the Delphix engine will only be resolved if any of the operations (environment add, refresh or enable) is performed. Delphix engine will not monitor the state of the target host in the background and hence any change will not be picked up unless an operation is triggered. So, the user needs to take action for the change to reflect in faults.
3. On a successful Delphix Engine upgrade, the latest default iSCSI recommendations will be used for validations.

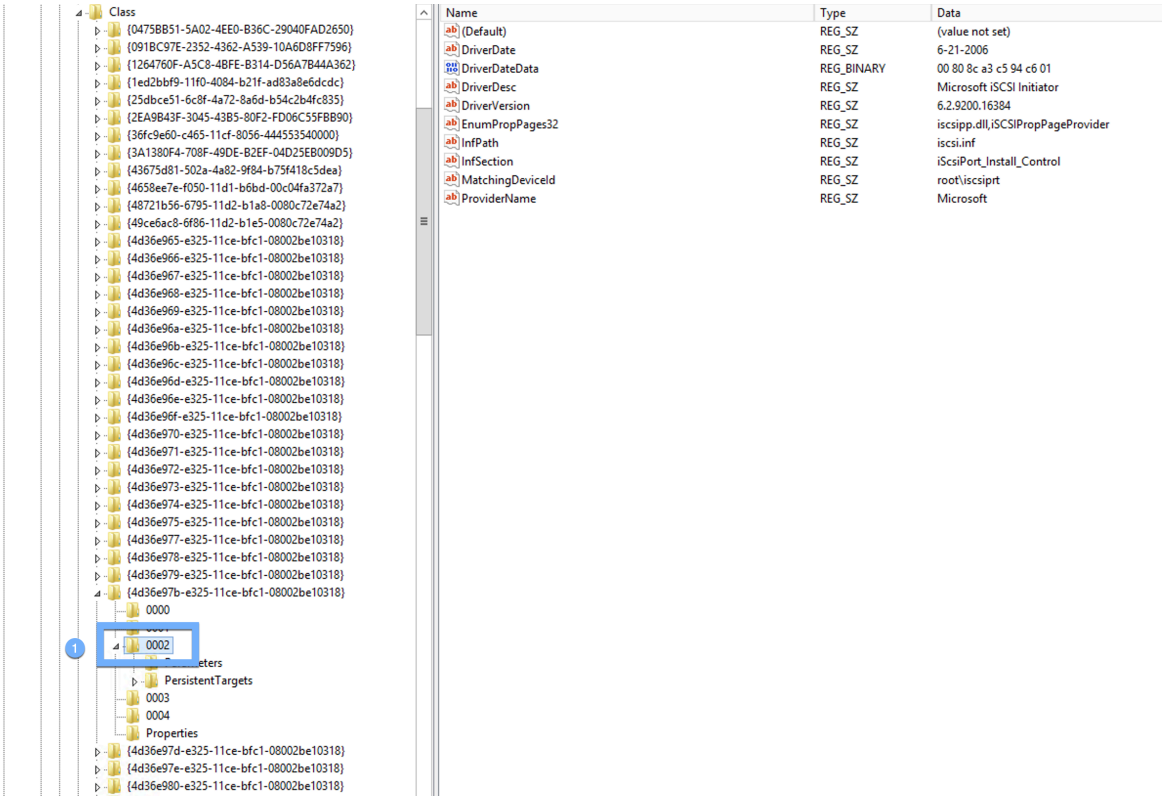
Troubleshooting

Type		Description
Fault(Severity = Warning)	`ENVIRONMENT_ISCSI_CONFIG_MISMATCH`	The single fault is thrown for all mismatched parameters
	`ISCSI_FETCH_CONFIG_PARAM_FAILURE`	The single fault is thrown for all parameters where we failed to fetch the value at the target host
Warning	`ISCSI_CONFIG_PARAM_TIMEOUT`	Job warning raised if we are unable to get the iSCSI parameters on the host within 5 minutes. No fault thrown at this point.

Type		Description
	`ISCSI_PS_VERSION_NOT_SUPPORTED`	Job warning raised if the PowerShell version is below 3 on the host side during the validation of the iSCSI parameters. No fault thrown at this point. The validation is skipped.
	`ENVIRONMENT_ISCSI_CONFIG_MISMATCH`	Job warning added for all mismatched parameters
	`ISCSI_FETCH_CONFIG_PARAM_FAILURE`	Job warning for all parameters where we failed to fetch the value at the target host

Identifying the instance number for iSCSI control class initiator drivers

1. From the Windows toolbar, click **Start** and select **Run** from the menu.
2. Type `regedit` in the **Open** field and click **OK.**
3. Go to the following registry key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4D36E97B-E325-11CE-BFC1-08002BE10318}\<Instance Number> where the value of <Instance Number> is the one that shows a DriverDesc value of Microsoft iSCSI Initiator. Under the registry key, locate and expand the plus (+) sign next to the instance number. In the example below, the value of the instance number is 0002.



1 Value of Instance Number

9.13.2.2.2.3 Network and connectivity requirements for Windows environment

Port allocations specific to unstructured files

The Delphix Engine makes use of the following network ports for unstructured files dSources and VDBs:

Outbound from the Delphix Engine

Protocol	Port Numbers	Use
TCP	xxxx	Delphix Connector connections to source and target environments. Typically, the Delphix Connector runs on port 9100.

Inbound to the Delphix Engine

Protocol	Port Number	Use
TCP	3260	iSCSI target daemon for connections from iSCSI initiators on the target environments to the Delphix Engine

Outbound from a source, staging, or target environment

Protocol	Port Numbers	Use
TCP	80	The Delphix Connector registers environments over HTTP
TCP	xxxx	DSP connections used for monitoring and script management. Typically, DSP runs on port 8415.

Inbound to a source, staging, or target environment

Protocol	Port Numbers	Use
TCP	xxxx	Delphix Connector connections to source environments. Typically, the Delphix Connector runs on port 9100.
TCP	445	If a staging environment is used, the staging server must be able to access SMB (Windows File Sharing) files on the source server.

General outbound from the Delphix Engine port allocation

Protocol	Port Numbers	Use
TCP	25	Connection to a local SMTP server for sending email
TCP/UDP	53	Connections to local DNS servers
UDP	123	Connection to an NTP server
UDP	162	Sending SNMP TRAP messages to an SNMP Manager

Protocol	Port Numbers	Use
TCP	443	HTTPS connections from the Delphix Engine to the Delphix Support upload server
TCP/UDP	636	Secure connections to an LDAP server
TCP	8415	Connections to a Delphix replication target. See Configuring Replication (see page 2381).
TCP	50001	Connections to source and target environments for network performance tests.

General inbound to the Delphix Engine port allocation

Protocol	Port Number	Use
TCP	22	SSH connections to the Delphix Engine
TCP	80	HTTP connections to the Delphix GUI
UDP	161	Messages from an SNMP Manager to the Delphix Engine
TCP	443	HTTPS connections to the Delphix Management Application
TCP	8415	Delphix Session Protocol connections from all DSP-based network services including Replication, SnapSync for Oracle, V2P, and the Delphix Connector.
TCP	50001	Connections from source and target environments for network performance tests via the Delphix CLI.

Firewalls and Intrusion Detection Systems (IDS)

Production databases on source environments (for dSources) are often separated from the non-production environment by firewalls. Firewalls can add milliseconds to the latency between servers. Accordingly, for best performance, there should be no firewalls between the Delphix Engine and the virtual database (VDB) target environments. If the Delphix Engine is separated from a source environment by a firewall, the firewall must be configured to permit network connections between the Delphix Engine and the source environments for the application protocols (ports) listed above.

Intrusion detection systems (IDSs) should also be made permissive to the Delphix Engine deployment. IDSs should be made aware of the anticipated high volumes of data transfer between dSources and the Delphix Engine.

9.13.2.2.2.4 Adding a windows environment

This topic describes how to add a Windows environment to the Delphix Engine for use with unstructured files.

All Windows source and target environments containing unstructured files must have the Delphix Connector installed to enable communication between the environment and the Delphix Engine. The instructions in this topic cover initiating the Add Target process in the Delphix Management application, running the Delphix Connector installer on the environment, and verifying that the environment has been added to the Delphix Engine.

Prerequisites

- Make sure your source and target environment meet the requirements described in Requirements for Windows Environments (see page 2237).[664]

Procedure

1. From the machine that you want to use, log in to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Environments**.
4. Next to **Environments**, click the **Actions** menu and select **Add Environment**.
5. In the **Add Environment** wizard Host and Server tab, select:
   a. Host OS: **Windows**
   b. Host Type: **Target**.
   c. Server Type: **Standalone**.
6. Click **Next**.
7. In the Environment Settings tab click the download link for the **Delphix Connector Installer**. The Delphix Connector will download to your local machine.
8. On the Windows machine that you want to want to use as a target, run the Delphix Connector installer. Click **Next** to advance through each of the installation wizard screens.

   The installer will only run on 64-bit Windows systems. 32-bit systems are not supported.
   a. For **Connector Configuration**, make sure there is no firewall in your environment blocking traffic to the port on the target environment that the Delphix Connector service will listen to.
   b. For **Select Installation Folder**, either accept the default folder or click **Browse** to select another.
   c. Click **Next** on the installer's final 'Confirm Installation' dialog to complete the installation process and then **Close** to exit the Delphix Connector Install Program.

664 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/requirements-for-windows-environments

9. Return to the Delphix Management application.
10. Enter the **Environment Name**, **Host Address**, **Delphix Connector Port**, **OS Username**, and **OS Password** for the target environment.
11. To provide your own Oracle Java select the **Provide my own JDK** checkbox and click **Next**.
12. In the Java Development Kit tab enter the absolute path to your Oracle JDK and click **Next**.
13. Click **Submit.**

As the new environment is added, you will see two jobs running in the **Delphix Admin Job History**, one to **Create and Discover** an environment, and another to **Create** an environment. When the jobs are complete, you will see the new environment added to the list in the **Environments** panel.

Post-requisites

1. On the Windows environment, in the **Windows Start Menu**, select **Services**.
2. Select **Extended Services**.
3. Make sure that the **Delphix Connector** service has a Status of **Started**.
4. Make sure that the **Startup Type** is **Automatic**.

9.13.2.2.2.5 Options for linking unstructured files on Windows environments

There are two techniques for linking a new dSource from files on a Windows source.

Direct communication with the source environment

The simplest technique is to have the Delphix Engine communicate directly with the source environment. This requires installing the Delphix Connector on the source machine. When linking, specify a local path on the source machine, such as C:\Files\MyData.

Using a staging environment

In some cases, it is not possible or desirable to install the Delphix Connector on the source environment. In those cases, you can install the Delphix Connector on a "staging environment." This is another Windows machine that will act as an intermediary between the Delphix Engine and the source environment. Files on the source must be accessible by the environment user from the staging environment via a UNC path. Specifically, the environment user is only required to have READ access to the path, its directories, and files so that the robocopy utility called from the staging host can function properly. For example, use Windows Sharing on the source machine. When linking, specify the UNC path to the files on the source – for example, \\MySource\MyData\

For more information on installing the Delphix connector, refer to [Installing the Delphix Connector Service on the Target Database Servers](#) (see page 2098).[665]

665 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/installing-the-delphix-connector-service-on-the-target-database-servers

9.13.2.2.3 Linking unstructured files

9.13.2.2.3.1 Prerequisites

- The source environment must meet the requirements outlined in Unstructured Files Environment Requirements (see page 2226).[666]
- The Delphix Engine must have access to an environment user. This user should have read permissions on all files to be cloned.

> **Unstructured Files on Cluster Environments**
> Unstructured files cannot be linked from, or provisioned to, any form of a cluster environment, such as an Oracle RAC environment. To link or provision unstructured files from a host that is part of a cluster, add the host as a standalone environment. Then link from, or provision to, this standalone host.

9.13.2.2.3.2 Procedure

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Environments**.
4. Select the **environment** containing the unstructured files you want to link.
5. Click the **Environment Details** tab.
6. If the environment user described in the Prerequisites section is not listed under **Environment Users**, add the user.
7. Click the **Databases** tab.
8. Scroll to the bottom of the page to view the **Unstructured Files** section.
9. Click the **Plus** icon on the right. This action displays a dialog box prompting for the **Database Name** and **Path**. Enter a name to help identify the files. The path is the absolute path to the directory on the environment server.
10. Click **Add** to save the configuration. After saving this configuration, add the dSource.
11. Click **Manage > Datasets**.
12. On the left-hand side, click the **plus** sign.
13. Select **Add dSource**. Alternatively, on the **Environment Management** screen, you can click **Add dSource** next to a dataset name to start the dSource creation process.

666 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/unstructured-files-environment-requirements

14. In the **Add dSource** wizard, select the source of the files.
15. Select the **Environment User** outlined in the Prerequisites section.
16. Click **Advanced**.
17. Enter **Paths to Exclude**. These paths are relative to the root path of the dataset home path and will not be linked by the Delphix Engine. This feature is most commonly used to exclude directories containing log files. Wildcard (*) *pattern matching is supported to exclude all the contents of a directory, without excluding the directory itself. For example, specifying* `/dir/` will exclude all contents of `/dir` but still link `/dir` as an empty directory. For PowerShell to escape a $ sign in a directory path please use the following / `before the dollar sign when adding it to the exclude paths, for example: /` $RECYCLE.BIN.
    Info:
    Retroactive Edits to Exclude Paths on Windows

    After creating a dSource, you can edit the set of **Paths to Exclude** from syncing at any time on the dSource's **Configuration** tab. For Unix environments, retroactively adding a path to exclude will result in the next SnapSync deleting the newly-excluded files. However, for Windows environments, retroactively adding a path to exclude will result in the next SnapSync ignoring newly-excluded files. Stale versions of these files will still exist in all future snapshots.
18. If you are linking files from a Unix environment, enter **Paths of Symlinks to Follow**. These paths are relative to the root path of the dataset home path and will be followed to gather additional files to copy.
    Info:

    Paths of Symlinks to Follow - Caveats
    - This feature can only be used to follow symlinks to directories. Symlinks to files will be ignored.
    - This feature is not available for files on Windows environments.
19. Click **Next**.
20. Enter a **dSource Name**.
21. Select a **Database Group** for the dSource.
22. Click **Next**. Adding a dSource to a database group enables you to set Delphix Domain user permissions for that dSource's objects, such as snapshots.
23. Select a **SnapSync** policy.
24. Click **Advanced** to edit retention policies.
25. Click **Next**.
26. Enter any operations that should be run at **Hooks** during the sync process (or any future sync processes).
27. Click **Next**.
28. Review the **dSource Configuration** and **Data Management** information.
29. Click **Submit**.

The Delphix Engine will initiate two jobs to create the dSource, **DB_Link,** and **DB_Sync**. You can monitor these jobs by clicking **Active Jobs** in the top menu bar, or by selecting **System** > **Event Viewer**. When the jobs have been completed successfully, the file's icon will change to a **dSource** icon on the **Environments** > **Databases** screen, and the dSource will be added to the list of **Datasets** under its assigned group.

> **dSource Information**
> After you have created a dSource, you can view information about it and make modifications to its policies and permissions by selecting it in the **Datasets** panel.

9.13.3 Create an empty VDB for unstructured files in the Delphix Engine

This topic describes the procedure for creating an empty VDB, used for unstructured files. The term "unstructured files" in Delphix refers to a dataset that acts as a directory of different files. An empty VDB for unstructured files is not a database and does not receive special treatment or processing by Delphix, it exists as a place for the files to be generated, tracked, and copied. Creating a VDB can be done with provisioning from an existing dataset (a dSource or another VDB) or as an empty VDB created and filled with data.

Creating an empty VDB places an initially-empty mount on target environments. It functions similar to a VDB created via provisioning except it cannot refresh. Refreshing a dataset means overwriting the dataset content with new data pulled from the dataset parent. If a new VDB is created from scratch, the newly-created dataset will not have a parent and cannot be refreshed. All other functions are identical, meaning that the new VDB for unstructured files can be provisioned from, rewound, take snapshots, and so on.

9.13.3.1 Prerequisites

The target environment must meet the requirements outlined in Unstructured Files Environment Requirements (see page 2226).[667]

> **Unstructured Files on Cluster Environments**
> You cannot create an empty VDB on any form of the cluster environment, such as an Oracle RAC environment. To create an empty VDB on a host that is part of a cluster, add the host as a standalone environment, then create the empty VDB on the standalone host.

9.13.3.2 Procedure

To create an empty VDB without provisioning:

1. Login to the **Delphix Management** application.
2. Click **Manage**.

667 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/unstructured-files-environment-requirements

3. Select**Datasets**.
4. Click the **plus** icon.
5. Select **Create Empty VDB**.
6. Click **Next** to go to the Target environment tab and do the following:
   a. Under Environment, select a target environment where your empty VDB will be placed.
   b. In the **Mount Path** field, enter the absolute path where the empty VDB/dataset will be mounted.
   c. Click **Next**.
7. On the **Configuration** tab, do the following:
   a. In the **Empty VDB name** field, enter a name for the VDB.
   b. Select a target group for the empty VDB.
   c. Select the **Enable auto-restart of the empty VDB** checkbox to allow VDB to be automatically restarted when the target host is rebooted.
   d. Click **Next**.
8. On the **Policies** tab, select a Snapshot Policy for the empty VDB.
9. (Optional) On the **Hooks** tab, select a hook point and then click + to add a script to run at that point. Click **Next**.
10. The **Summary** tab will enable you to review your configurations. Click **Submit**

After the operation completes, the empty VDB will appear in the Datasets panel.

9.13.4 Provisioning unstructured files as vFiles

9.13.4.1 Overview

This topic describes the process of provisioning to a set of unstructured files as vFiles.

9.13.4.2 Prerequisites

- You will need an unstructured files dSource, as described in Linking Unstructured Files (see page 2249), or an existing vFiles from which you want to provision another.
- The target environment must meet the requirements outlined in Unstructured Files Environment Requirements (see page 2226).[668]

Unstructured Files on Cluster Environments

668 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/unstructured-files-environment-requirements

> Unstructured files cannot be linked from, or provisioned to, any form of a cluster environment, such as an Oracle RAC environment. To link or provision unstructured files from a host that is part of a cluster, add the host as a standalone environment. Then, link from or provision to this standalone host.

9.13.4.3 Post provision/migration ownership rules

When a new vFile VDB is provisioned, the ownership is changed to match the Environment User anytime a VDB is enabled. This could cause conflict in the ownership of existing files in a case where the VDB has just been migrated to a new host – the VDB files on the new host will now be owned by the new Environment User.

9.13.4.4 Procedure

1. Login to the **Delphix Management** application.
2. Click **Manage**.
3. Select **Datasets**.
4. Select a **dSource** or **vFiles**.
5. Click the **TimeFlow** tab.
6. Select a **snapshot**.
7. Click **Provision**. The **Provision vFiles** panel will open, and the field **Mount Path** will auto-populate with the path to the files on the source environment.
8. Select a target environment. If you need to add a new target environment for the vFiles, click the **Plus** icon next to **Filter Target**to add an environment.
   Note: You can only target a Unix environment when provisioning from a Unix dSources or vFiles. You can only target a Windows environment when provisioning from a Windows dSources and vFiles.
9. If necessary, modify the **Mount Path**.
   - On Windows, this mount path must not be a UNC path. It must be a local drive letter and folder path. The UNC path will operate after the provisioning completes.
   - On Linux and Unix hosts, this mount path must be the full path and not include symlinks.
10. Click **Advanced**.
11. Enter **Additional Mount Points**. When it is mounted to the target environment, the vFiles will be mounted to any additional mount points you provide.
    Note: The **Shared Path** is a relative path dictating which portion of the vFiles should be available on the additional environments. To share the entirety of the vFiles, specify a **Shared Path** of /.
12. Select an Environment User to own the mounted files. If the files are being mounted to multiple environments, ensure this user is available across all environments.
13. Click **Next**.

14. Enter a vFiles **Name**.
15. Select a **Target Group** for the vFiles. If necessary, click the **Plus** icon to add a new group.
16. Select a **Snapshot Policy** for the vFiles. if necessary create a new policy.
17. Click **Next**.
18. Enable Auto VDB Restart to allow the VDB to be automatically restarted when the target host reboot is detected by Delphix.
19. Enter any operations that should be run as Hooks during the lifetime of the vFiles.
20. Click **Next**.
21. Click **Submit**.

When provision starts, the vFiles will appear in the Datasets panel. Select the vFile and navigate to the Status tab to see the progress of the job. When provisioning is complete, you can see more information on the Configuration tab.

9.13.5 Managing vFiles

9.13.5.1 Overview

This article is used to cover steps on adding an additional mount to an existing vFile and outlining post provision/migration ownership rules.

9.13.5.2 Adding an additional mount

1. Login to the **Delphix Management** application.
2. Select the **vFile** you want to edit.
3. From the Actions menu (**...**) select **Disable**.
4. In the **Configuration** tab select the **Source** sub-tab, a Pencil icon will appear on the right of the Additional Mount Points.
5. Click on the **Pencil** to edit the mount points.
6. In the **Additional Mount Points** window select the **Plus** icon to add additional mount points.
7. Click the **checkmark** to save.
8. From the Actions menu (**...**) **Enable** the vFile.

9.13.5.3 Post provision/migration ownership rules

When a new vFile VDB is provisioned, the ownership is changed to match the Environment User anytime a VDB is enabled. This could cause conflict in the ownership of existing files in a case where the VDB has just been migrated to a new host – the VDB files on the new host will now be owned by the new Environment User.

9.13.6 vFiles best practices and common pitfalls

9.13.6.1 Overview

This document is the implementation guide for the best practices of implementing data source integrations using the Delphix vFile functionality provided through the AppData toolkit. Since "unstructured files", or vFiles, typically is implemented for non-DBMS file types, this implementation will require a certain degree of configuration and scripting in order to function. This means that while you do get some of the main functionality of the Delphix Engine, there are also quite a few pitfalls that may deter customers from wanting this type of configuration. Using vFiles and scripts to implement data source ingestion is a workaround that should only be used if you and the customer are fully aware of the best practices and limitations as called out below.

9.13.6.2 Best practices and implementation

In order to be successful with this workaround, please keep in mind these best practices which will enable you to be successful in implementing a vFile configuration for data sources.

- Write a script to put the RDBMS into 'Backup' mode to prepare it for the next set of scripts Delphix will perform
- Create tablespaces in the RDBMS which dxtoolkit will take snapshots of while the database is in backup mode
- After any vFile provisioning job, script a recovery of the RDBMS instance

9.13.6.3 Common use cases

These use cases are the typical scenarios for which you would use this workaround.

9.13.6.3.1 Unsupported DBMS configuration

This functionality is most appropriate when a customer is using an unsupported data source and wants a quick MVP to demonstrate the value of Delphix. It is critical to understand the pitfalls of this implementation and what value propositions that this excludes so that the customer is aware of what capabilities our core integrations provide.

9.13.6.4 Common pitfalls

These pitfalls of functionality should be considered whenever implementing vFile scripted ingestion. Refer to the table for a quick reference of pitfalls.

Pitfall	Description
LogSync	Not Available

Pitfall	Description
Point in Time Recovery	Not Available
Clustering	Not Available
User Interface	Not Available
Production Downtime	Yes
Use of dxtoolkit	Requires Training

9.13.6.4.1 LogSync and Point in Time provisioning

Since these scripts are taking full backups of the database, there will not be logs available for us to sync with. Therefore, point in time provisioning is not available to any vFile implementation.

9.13.6.4.2 Manual operations

Currently, customers using this workaround have to perform the scripts manually. This creates operational overhead for customers, particularly their DBA team.

9.13.6.4.3 Clustering

Using vFiles is not available for clustered instances, and there is currently no workaround for this type of configuration.

9.13.6.4.4 Lack of Graphic User Interface (GUI)

Scripts perform all the core functionality that is typically carried out in the Delphix GUI. This implementation precludes the usage of the main GUI and is only accessible via the Command Line Interface (CLI).

9.13.6.4.5 Production downtime

As the database enters backup mode during the ingestion process, this workaround causes the production source to go down while Delphix ingests their data.

9.13.6.5 Understanding and implementing scripts and dxtoolkit

Implementing these scripts requires an understanding of dxtoolkit and the data source which you are trying to integrate with. Typically, we estimate the learning period of this to be about 1 - 2 months to fully understand how to use these tools and how to best implement the scripts for any given source.

9.13.6.6 Conclusion

Using vFile functionality with scripts is a quick way to demonstrate the value of the Delphix Engine easily. With a little scripting and Delphix knowledge, you can engineer ingestion with dxtoolkit to take snapshots of the entire database. However, there are several pitfalls that every customer should know about and understand before moving forward with this type of implementation. If you have any questions, please reach out to Product Management.

9.13.7 Delphix Engine plugin management

9.13.7.1 Plugin management

This section provides information on the data management plugin deployed on the Delphix Engine. Users can directly upload their Plugins to the Delphix Management application with a single click, thus eliminating the process of uploading the Plugin using the command. Plugin upload via the Delphix Management application helps users to build their Plugin library in a simple and easy-to-use manner.

The Plugin screen allows users to view a structured or formatted view of their selected Plugin and it provides a single click option for users to delete a Plugin.

> Only admin users are able to upload or delete plugins. Standard users are only able to view Plugin information.

9.13.7.2 Plugin types

There are two types of plugins the Delphix engine supports Lua/Python plugins and platform plugins. Because support for these types of plugins was written at different times, the upgrade and replication workflows will differ depending on what the plugin type is. To figure out what type of plugin this is, follow the instructions below:

1. Login to the **Delphix Management** application.
2. Select **Manage > Plugins**.
3. Select the plugin you want to check and look for the type field in the details section on the right.
4. If the type is Toolkit this means the plugin is a Lua plugin. If the type is Plugin, then the plugin is either a platform plugin or a python plugin.

9.13.7.3 Installation of plugin

9.13.7.3.1 Downloading a plugin from Delphix Downloads

9.13.7.3.1.1 Procedure

1. In the web browser, go to the Delphix download[669] site.
2. Login to the download site using email and password credentials.
3. Navigate to the required version number of Delphix Engine.
4. Go to the Plugins>Select the plugin to be downloaded> Plugin_<version_number>.zip folder
   Ex: For EBS, EBS> Toolkit_EBS_<version_number>for_<EBS12.1/12.2>.zip folder
5. Download the zip file, unzip it, and copy both the .json files at your preferred location.

9.13.7.3.2 Uploading a plugin to Delphix Engine

Plugins can be uploaded onto the Delphix Engine only by an admin.

9.13.7.3.2.1 Prerequisites

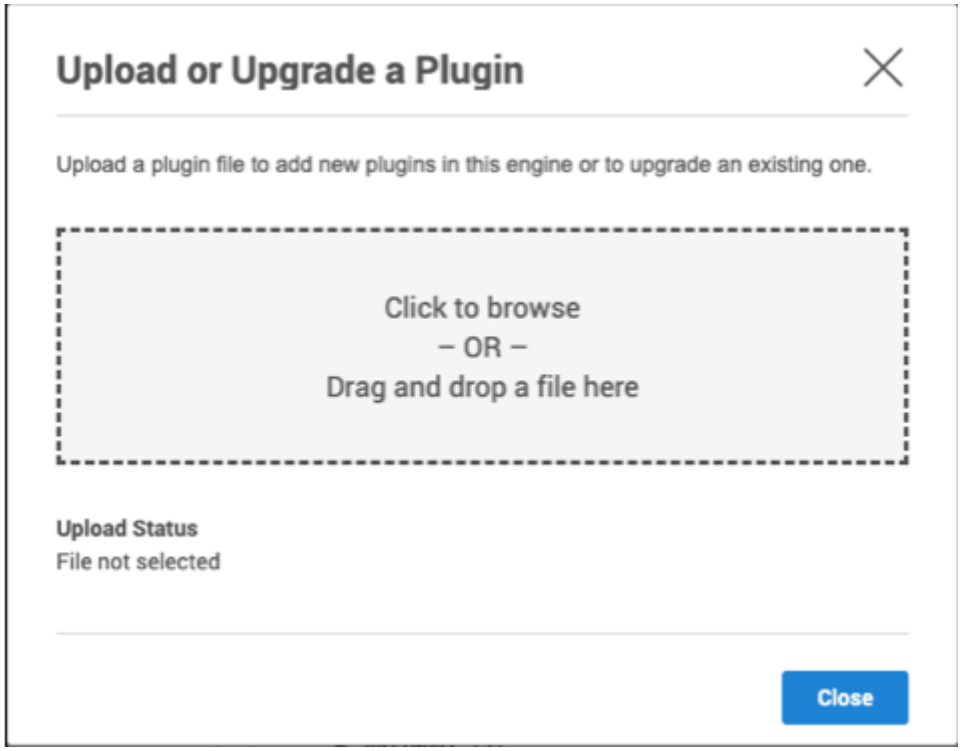
- Make sure you have access to a copy of the plugin from the computer you will be installing.
- Verify with the plugin author that the version of the plugin you have is compatible with your Delphix Engine version.

9.13.7.3.2.2 Procedure

To add or upload a new Plugin complete the instructions below:

1. Login to the **Delphix Management** application.
2. Select **Manage > Plugins**.
3. To add or upload a Plugin, click on the plus icon. This opens the **Upload or Upgrade Plugin** dialog.

669 https://download.delphix.com/



4. To start using the plugin, refresh target and source environments to discover any repositories.

9.13.7.4 Upgrading a plugin

When a new version of a plugin is released, the new plugin can only be uploaded onto the Delphix Engine by an Admin.

9.13.7.4.1 Prerequisites

- Make sure you have access to a copy of the plugin from the computer you will be installing.
- Verify with the plugin author that the new version of the plugin is compatible with your current version, and with your Delphix Engine version.
- If the already-installed plugin does not use the Lua language, then you don't have to disable any dSources or VDBs. However, be aware that you will not be able to perform any Delphix operations on them while the plugin is being upgraded.

9.13.7.4.2 Procedure

To upgrade a Plugin just follow the instructions above on how to upload a Plugin. The newly-uploaded plugin will be used to upgrade any already-present version(s) of the same plugin.

9.13.7.5 Notes on replication with objects administered by a plugin

Replication works the same way for objects created with plugins and objects created using natively supported data types. But, complications and problems can arise when a replicated plugin's version does not match the version of the installed plugin.

> **Replication Recommendation**
> As discussed in the section on Delphix Replication Overview (see page 2368), we highly recommend that replication target engines be used solely to hold replicated objects, and that failover is only done in disaster recovery situations. Or, at the least, make sure that the versions of replicated plugins match the versions of plugins that are already installed on the replication target engine.

There are specifically two operations done after replication that will be listed below.

9.13.7.5.1 Provisioning from replicated data sources or VDBs

Replica provisioning is only guaranteed to work when the source object's plugin is already also installed on the target engine with the identical version.

Installed Plugin Version	Result
None	Replica provisioning will fail.
Lower than replicated version	Replica provisioning will fail.
Same as replicated version	Replica provisioning will succeed.
Higher than replicated version	If the replicated plugin uses the Lua language, replica provisioning will fail. If the replicated plugin does not use the Lua language, replica provisioning may or may not succeed, depending on whether the installed plugin is compatible with the replicated version.

More general information on how to perform a replica provision can be found here (see page 2403).

9.13.7.5.2 Failing over a replica

As mentioned above, we strongly recommend that replication target engines do not have any non-replica objects. Although the replica failover process will always work, the failed-over objects may or may not be fully functional if there are non-replica plugin-administered objects already on the same engine. In some cases, you may have to delete some objects, and their associated plugins.

Installed Plugin Version	Result
None	The replicated plugin and its objects will be moved into the live namespace. All failed-over objects will be fully functional.
Lower than replicated version	The failed-over objects may become "inactive", and any Delphix operations run on them will fail. In this case, you can try to upgrade the existing plugin so that the version matches the replicated inactive plugin. This may or may not work, depending on whether there are any conflicts between any objects.
Same as replicated version	The failed-over objects will be fully functional.
Higher than replicated version	The failed-over objects will become "inactive", and any Delphix operations run on them will fail. In this case, you can try to upgrade the inactive plugin (see instructions below). Again, this may or may not work, depending on whether there are conflicts between any objects, and on whether the installed plugin knows how to interact with the inactive plugin.

More generic information on how to perform an actual failover can be found here[670].

9.13.7.6 Upgrading an inactive plugin

If a failed-over plugin becomes "inactive" (see details above), you may be able to activate the plugin by upgrading it to match the version of the installed plugin.

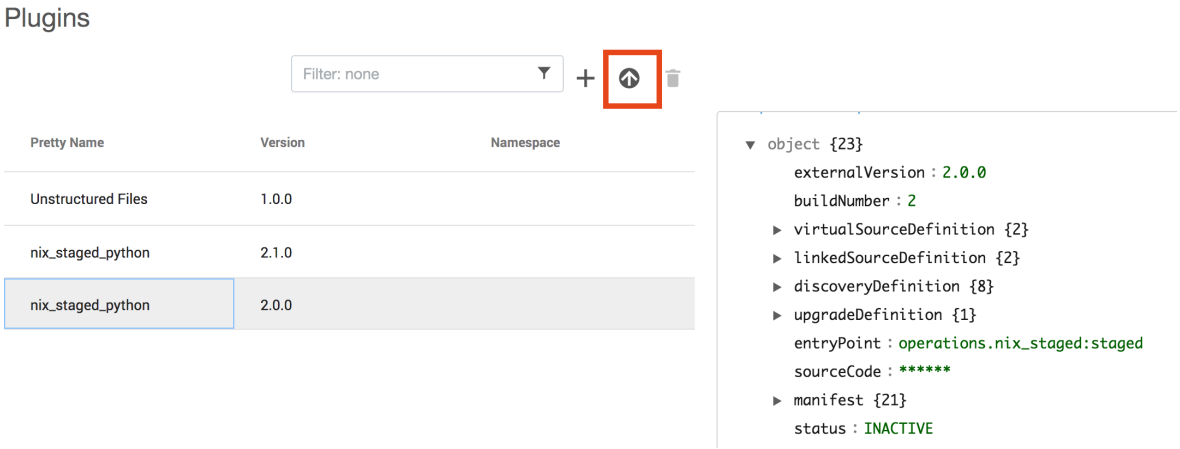
9.13.7.6.1 Procedure

To upgrade an inactive Plugin complete the instructions below:

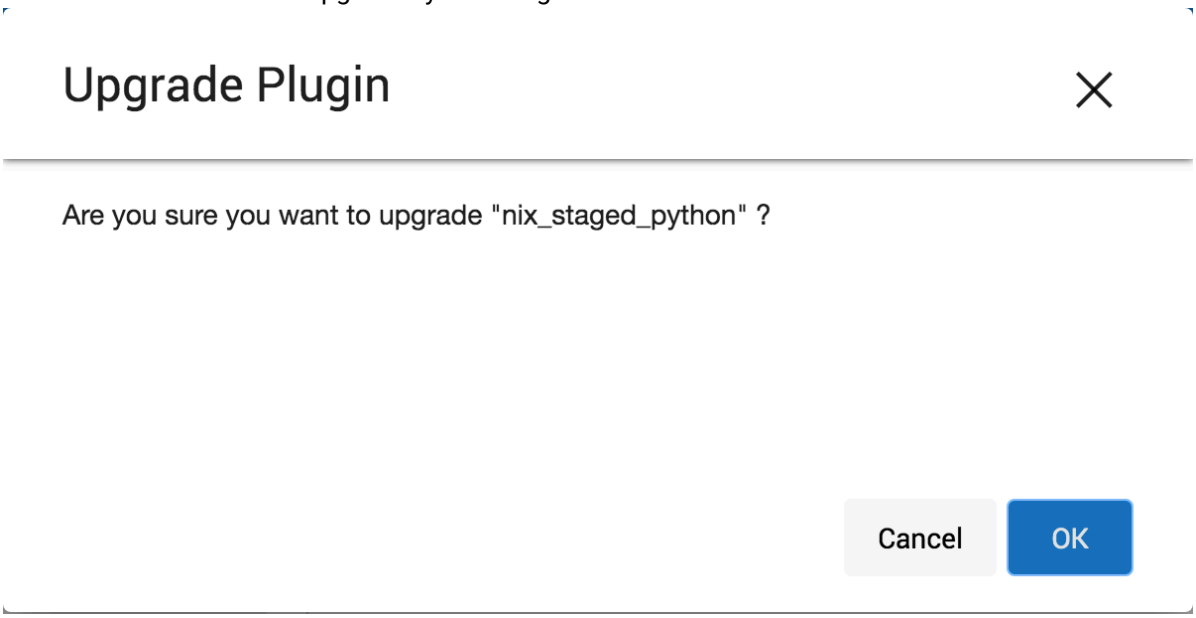
1. Login to the **Delphix Management** application.
2. Select **Manage > Plugins**.

670 http://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/controlled-failover

3. In the drop-down menu on the left, select the plugin that is inactive.
4. Click the upgrade button shown below, which is now clickable to activate the inactive plugins.



5. Make sure to confirm the upgrade by selecting OK.



6. After the upgrade is successful the inactive plugin of the lower version will not exist in the plugin list.

> If the inactive plugin is a higher version, the active plugin needs to be upgraded using the normal upgrade procedure instead.

9.13.8 Unstructured files hook operation notes

9.13.8.1 Shell Operations

9.13.8.1.1 RunCommand operation

The RunCommand operation runs a shell command on a Unix environment using whatever binary is available at `/bin/sh`. The environment user runs this shell command from their home directory. The Delphix Engine captures and logs all output from this command. If the script fails, the output is displayed in the Delphix Management application and command line interface (CLI) to aid in debugging.

If successful, the shell command must exit with an exit code of `0`. All other exit codes will be treated as an operation failure.

9.13.8.1.1.1 Examples of RunCommand operations

You can input the full command contents into the RunCommand operation.

```
remove_dir="$DIRECTORY_TO_REMOVE_ENVIRONMENT_VARIABLE"

if test -d "$remove_dir"; then
    rm -rf "$remove_dir" || exit 1
fi

exit 0
```

If a script already exists on the remote environment and is executable by the environment user, the RunCommand operation can execute this script directly.

```
/opt/app/oracle/product/10.2.0.5/db_1/dbs/myscript.sh "$ARG_ENVIRONMENT_VARIABLE"
"second argument in double quotes" 'third argument in single quotes'
```

9.13.8.1.2 RunBash operation

The RunBash operation runs a Bash command on a Unix environment using a `bash` binary provided by the Delphix Engine, unless it's a Linux environment, in which case it uses the system's native bash binary. The environment user runs this Bash command from their home directory. The Delphix Engine captures and logs all output from this command. If the script fails, the output is displayed in the Delphix Management application and command line interface (CLI) to aid in debugging.

If successful, the Bash command must exit with an exit code of `0`. All other exit codes will be treated as an operation failure.

9.13.8.1.2.1 Example of RunBash operations

You can input the full command contents into the RunBash operation.

```
remove_dir="$DIRECTORY_TO_REMOVE_ENVIRONMENT_VARIABLE"

# Bashisms are safe here!
if [[ -d "$remove_dir" ]]; then
    rm -rf "$remove_dir" || exit 1
fi

exit 0
```

9.13.8.1.3 Shell operation tips

9.13.8.1.3.1 Using `nohup`

You can use the `nohup` command and process backgrounding from resource in order to "detach" a process from the Delphix Engine. However, if you use `nohup` and process backgrounding, you MUST redirect `stdout` and `stderr.`

Unless you explicitly tell the shell to redirect `stdout` and `stderr` in your command or script, the Delphix Engine will keep its connection to the remote environment open while the process is writing to either `stdout` or `stderr` . Redirection ensures that the Delphix Engine will see no more output and thus not block waiting for the process to finish.

For example, imagine having your `RunCommand` operation background a long-running Python process. Below are the bad and good ways to do this.

Bad Examples

- `nohup python file.py & # no redirection`
- `nohup python file.py 2>&1 & # stdout is not redirected`
- `nohup python file.py 1>/dev/null & # stderr is not redirected`
- `nohup python file.py 2>/dev/null & # stdout is not redirected`

Good Examples

- `nohup python file.py 1>/dev/null 2>&1 & # both stdout and stderr redirected, Delphix Engine will not block`

9.13.8.2 Other operations

9.13.8.2.1 RunExpect operation

The RunExpect operation executes an Expect script on a Unix environment. The Expect utility provides a scripting language that makes it easy to automate interactions with programs which normally can only be used interactively, such as `ssh`. The Delphix Engine includes a platform-independent implementation of a subset of the full Expect functionality.

The script is run on the remote environment as the environment user from their home directory. The Delphix Engine captures and logs all output of the script. If the operation fails, the output is displayed in the Delphix Management application and CLI to aid in debugging.

If successful, the script must exit with an exit code of `0`. All other exit codes will be treated as an operation failure.

9.13.8.2.1.1 Example of a RunExpect Operation

Start an `ssh` session while interactively providing the user's password.

```
spawn ssh user@delphix.com
expect {
    -re {Password: } {
        send "${env(PASSWORD_ENVIRONMENT_VARIABLE)}\n"
    }
    timeout {
        puts "Timed out waiting for password prompt."
        exit 1
    }
}
exit 0
```

9.13.8.2.1.2 RunPowershell operation

The RunPowershell operation executes a PowerShell script on a Windows environment. The environment user runs this shell command from their home directory. The Delphix Engine captures and logs all output of the script. If the script fails, the output is displayed in the Delphix Management application and command line interface (CLI) to aid in debugging.

If successful, the script must exit with an exit code of `0`. All other exit codes will be treated as an operation failure.

Example of a RunPowershell operation

You can input the full command contents into the RunPowershell operation.

```
$removedir = $Env:DIRECTORY_TO_REMOVE

if ((Test-Path $removedir) -And (Get-Item $removedir) -is [System.IO.DirectoryInfo])
{
    Remove-Item -Recurse -Force $removedir
} else {
    exit 1
}
exit 0
```

9.13.8.2.2 Unstructured files environment variables

Operations that run user-provided scripts have access to environment variables. For operations associated with specific dSources or virtual databases (VDBs), the Delphix Engine will always set environment variables so that the user-provided operations can use them to access the dSource or VDB.

9.13.8.2.2.1 dSource environment variables

Environment Variable	Description
`DLPX_DATA_DIRECTORY`	Path where linked-staged database is mounted

9.13.8.2.2.2 VDB environment variables

Environment Variable	Description
`DLPX_DATA_DIRECTORY`	Path where virtual database is mounted

9.14 Yugabyte data sources

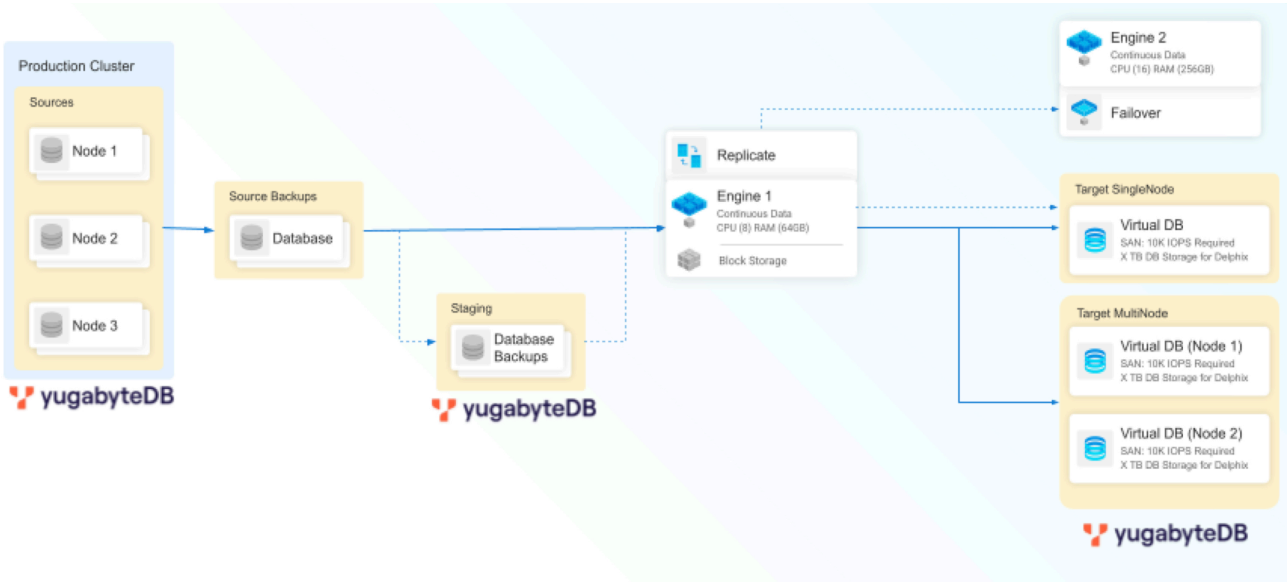
YugabyteDB data virtualization solution employs a staging architecture model (see page 1014). Within this framework, the source or production database backup is ingested into the Delphix Continuous Data Engine

through a dedicated staging environment. This ingested backup copy is subsequently denoted as a **dSource**. This dSource enables the creation and updates of virtual databases of YugabyteDB within the target environment.

The staging ingestion (resync process) ingests new backups from the provided NFS location. The backups are taken by the user leveraging the `yb_backup.py`.

This script takes a snapshot of the production database and exports the backup files to the given location.

The connector flow is depicted below:



The Yugabyte connector expects the backup location to be accessible by the staging host via NFS. It ingests the backup onto the mounted filesystem.

This tool can be leveraged to create a zero-touch staging cluster. The connector does not directly communicate/connect with the production (source) cluster.

The backups on the staging host are used to provision virtual copies of the database on the target host.

9.14.1 Yugabyte virtualization process

This topic serves as a quick guide that guides you through the first implementation of the YugabyteDB connector, providing high-level steps. Each step briefly describes the task and the expected result. Refer to the documentation links for more detailed information.

1. Begin by reviewing the general Delphix Continuous Data architecture diagrams (see page 2266). Identify existing or new YugabyteDB source, staging, and target environments, the underlying Operating Systems, and YugabyteDB database versions. This will give you a good understanding of the architectural strategy and confirm support.
2. Review the YugabyteDB support matrices (see page 2268). Verify that the YugabyteDB connector complies with all system prerequisites, including those of the Delphix Continuous Data Engine, staging/target environments, Operating Systems, and YugabyteDB database versions. Verifying this information is essential to align with the features and functionalities of the connector.

3. Install the YugabyteDB connector (see page 2276). The YugabyteDB connector establishes the connection between the Delphix Continuous Data Engine and the connector version. It is recommended to stay up-to-date with the latest version.
4. Create staging and target environments (see page 2268). The staging environment will host a database that Delphix Continuous Data Engine will stay in sync with the source environment database at an interval chosen by you. The target environment will provide the infrastructure to generate virtual database copies.
    a. Consult the source environment (see page 2268) and staging/target environment (see page 2268) requirements documentation for configuration details.
5. Discover staging and target environments (see page 2279). To enable the Delphix Continuous Data Engine to recognize the provided infrastructure, search for YugabyteDB installation, and then make it available for database linking and provisioning.
6. Linking the YugabyteDB database (see page 2283). This configures a dSource that enables the syncing mechanism between the source database, staging database, and Delphix Continuous Data Engine. Many ingestion methods differ between connectors and solve different requirements. It is best to identify the ingestion method that best suits your organization and infrastructure policies.
7. Provision virtual YugabyteDB databases (see page 2293). Administrators and self-service users can provision one or more copies of their virtual databases.

9.14.2 Yugabyte requirements and prerequisites

This section contains the following pages about Yugabyte requirements and prerequisites in Delphix Continuous Data Engine.

- Yugabyte support matrix (see page 2268)
- Additional Yugabyte staging prerequisites for ingestion using backups (see page 2270)
- Yugabyte network and connectivity requirements (see page 2271)
- Yugabyte host environment requirements (see page 2273)

9.14.2.1 Yugabyte support matrix

9.14.2.1.1 Platform vs database vs connector compatibility

Refer to the below support matrix to ensure compatibility between the platform, database, connector, and Delphix Continuous Data Engine version.

Color	Supported?
Y	**Yes**
N	**No**

9.14.2.1.2 OS and database compatibility

Supported OS Version	Supported DBMS Versions	
	Yugabyte 2.18.x	Yugabyte 2.20.x
CentOS 7.6	Y	Y
Windows	N	N

9.14.2.1.3 Delphix Continuous Data Engine and connector compatibility

Delphix Engine Versions	Supported Yugabyte Connector Versions	
	version 1.0.0	version 1.1.0
21.0.0.0	Y	Y
22.0.0.0	Y	Y

9.14.2.1.4 Database and Connector compatibility

Yugabyte Version	Supported Yugabyte Connector Versions	
	version 1.0.0	version 1.1.0
Yugabyte 2.18.x	Y	Y
Yugabyte 2.20.x	Y	Y

9.14.2.1.5 Requirements for source databases

There are no specific requirements for the source database, as the connector does not establish a connection with it.

9.14.2.1.6 Requirements for staging/target databases

The staging and target environments have the same set of requirements for YugabyteDB.

9.14.2.1.7 Limitations

- V2P is not supported.
- Point-in-time recovery is not supported. Provision/refresh/rewind time is based on snapshot only.

9.14.2.2 Additional Yugabyte staging prerequisites for ingestion using backups

The backup can exist on the staging server or can be accessed over NFS.

For performing a backup operation via `yb_backup.py` script, you can execute the following command:

```
python3 yb_backup.py --masters 10.23.1.2:7100 --remote_yb_admin_binary /home/
yugabyte/yugabyte-2.20.1.3/bin/yb-admin --remote_ysql_dump_binary /home/yugabyte/
yugabyte-2.20.1.3/postgres/bin/ysql_dump --remote_ysql_shell_binary /home/yugabyte/
yugabyte-2.20.1.3/postgres/bin/ysqlsh --ssh_user yugabyte --ssh_port 22 --
ssh_key_path /home/yugabyte/.ssh/id_rsa --storage_type nfs --nfs_storage_path /home/
yugabyte/database --backup_location /home/yugabyte/backup --no_auto_name --keyspace
ysql.yb_demo create
```

Script arguments:

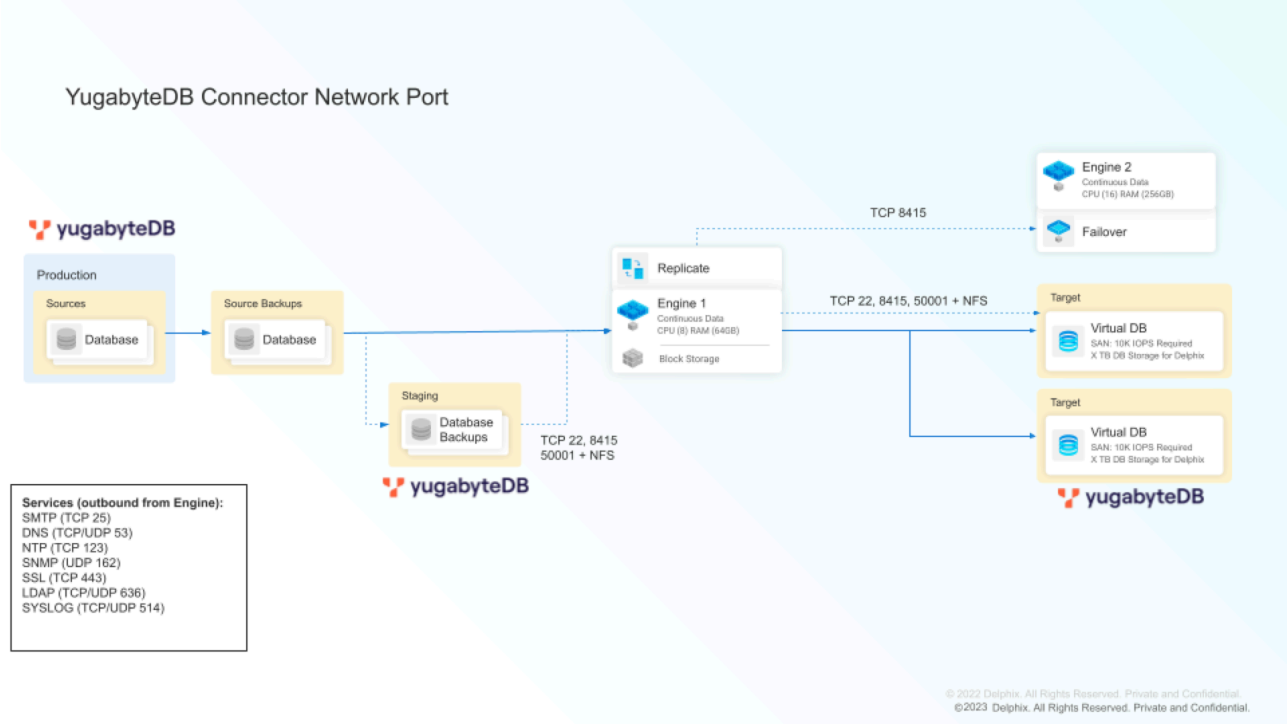
- masters: comma (“,”) separated list of <ip>:<port> for master servers of the production database
- remote_yb_admin_binary: yb_admin binary path on the master server
- remote_ysql_dump_binary: ysql_dump binary path on the master server
- remote_ysql_shell_binary: ysqlsh binary path on the master server
- ssh_user: OS user used to ssh into a remote host
- ssh_port: ssh port enabled on the remote host
- ssh_key_path: ssh keypair with a path on the remote host
- storage_type: use `nfs` to take backup on NFS location
- nfs_storage_path: NFS storage mount path
- backup_location: Location where backup is to be taken
- no_auto_name: Disable the automatic generation of a name under the given backup location.
- keyspace: Repeatable keyspace of the tables to backup or a target keyspace for the backup restoring.

This command reaches the production master server at `10.23.1.2:7100` and takes backup at location `/home/yugabyte/backup`. The database backed up is `yb_demo`.

9.14.2.3 Yugabyte network and connectivity requirements

Ensure that the requirements for Delphix Continuous Data Engine network connectivity (see page 566) are met before you proceed further. This page addresses YugabyteDB connector-specific network and connectivity requirements.

9.14.2.3.1 Ports for Yugabyte connector



9.14.2.3.1.1 SSH connectivity to staging/target host

1. Staging and target environments are required to have `sshd` running and configured for Delphix Continuous Data Engine to connect over `ssh`.
2. The Delphix Continuous Data Engine platform expects to maintain long-running, highly performant-SSH connections with the host environment.

9.14.2.3.2 Whitelisting inbound port for SSH connection to target host

9.14.2.3.2.1 Creating NFS mounts on the target host

The Delphix Continuous Data Engine shares its storage space with the target host using NFS mounts.

Protocol	Port Number	Use
TCP	22	SSH connection to the target database environment.
TCP/UDP	111	Remote Procedure Call (RPC) port mapper used for NFSv3 mounts.
TCP	1110	Network Status Monitor (NSM) client from target hosts to the Delphix Continuous Data Engine.
TCP	2049	NFS client from target hosts to the Delphix Continuous Data Engine (NFSv3 and NFSv4).
TCP	54043	Client mount daemon (NFSv3 only).
TCP	54044	Lock state notification service (NFSv3 only).
TCP	54045	Network Lock Manager (NLM) client from target hosts to Delphix Continuous Data Engine (NFSv3 only).
UDP	33434 - 33464	Traceroute from the target database server to the Delphix Continuous Data Engine (optional).

9.14.2.3.2.2 Connecting to YugabyteDB instances (VDBs)

Outgoing (From)	Incoming (To)	Protocol	Port Number	Use
End-user hosts	Target YugabyteDB instance	TCP	<target-port>	Connecting to the created database instance on the target host.

9.14.2.4 Yugabyte host environment requirements

Staging/target host requirements	Explanation
Host Compatibility Check The Operating System and architecture of the target host must match the source environment.	It is required that the source and the target environments should have identical OS and hardware configurations.
There must be a directory on the staging/target host where the Yugabyte connector support files can be installed (for example, `/var/tmp`) with the following properties: 1. The directory must be writable by the Operating System user mentioned above. 2. The directory must have at least 1.5 GB of available storage.	This path is called the “Toolkit Path” in the Environment creation UI page.
YugabyteDB Installation and Compatibility Check There must be an installation of YugabyteDB on the target host that is compatible with the installation of YugabyteDB on the source host.	The YugabyteDB binary version on the target must be identical to the source side.
There must be a mount point directory (for example, `/mnt`) that will be used as the base for mount points that are created when provisioning a VDB with the following properties: 1. The mount point directory must be writable by the operating system user mentioned above. 2. The mount point directory should be empty.	

Hostname and IP must be correctly set in `/etc/hosts` file.	The output of the hostname -i command should produce the correct result as the IP address of the server.
	For example:
	`[root@source ~]# hostname -i`
	`10.110.211.225`

Staging/target host requirements	Explanation
Required Utilities and Tools Ensure the expect utility is installed on the target host.	YugabyteDB connector runs password sensitive commands by making use of expect utility. Checking if the expect utility is installed with: `which expect`
Ensure the aws-cli utility is installed on the staging host. (Mandatory only in case of Dsource type: **aws_s3**)	YugabyteDB Connector runs aws-cli commands to list and copy backups from AWS S3 buckets. Checking if the aws-cli utility is installed with: `which aws`
Storage Requirement	1. The staging host must have enough storage space to hold the source backup file OR NFS mount source backup filesystem. 2. An empty folder on the staging host to hold the Delphix Toolkit (approximately 2GB free space).

9.14.2.4.1 Sudo privileges for the environment OS user

The Delphix Continuous Data Engine connects to the target host (via SSH) to majorly perform the following operations:

1. Delphix Continuous Data Engine dynamically mounts and unmounts directories for linking and provisioning operations. This privilege is required because mount and unmount are typically reserved for superusers.
2. Delphix Continuous Data Engine dynamically creates and removes directories for linking and provisioning operations.

The operations (1) and (2) will require sudo privileges for the OS user.

9.14.2.4.1.1 Configuring sudo access on target environments (as root user):

1. The sudo configuration exists in the `/etc/sudoers` file. The file can be edited directly or a utility called `visudo` can be used to add the privilege.

```
vi /etc/sudoers
```

```
# Or
visudo
```

2. Adding the privileges

```
Defaults:delphix_os !requiretty
delphix_os ALL=NOPASSWD: /bin/mount,/bin/umount,/bin/mkdir,/bin/rmdir, /bin/ps
delphix_os ALL=(Yugabyte) NOPASSWD: ALL
```

> The following examples are for illustrative purposes and the sudo file configuration options are subject to change.

9.14.2.4.1.2 Requiretty settings

Delphix Continuous Data Engine requires that the requiretty setting be disabled for all the Delphix Continuous Data Engine users with sudo privileges.

It is required to specify the NOPASSWD qualifier within the "sudo" configuration file. This ensures that the "sudo" command does not demand the entry of a password, even for the "display permissions" (i.e. "sudo -l") command.

- Additionally, Delphix OS user should be,
  - Regular o/s user - e.g., delphix_os.
  - Add OS user to Yugabyte OS group.
  - Must have access to backup files of the source database stored on the staging host.
  - Must have access to the path `create` and modify files in `/opt/yugabyte/var/lib/yugabyte`.
  - Empty folder on host to hold Delphix toolkit (approximate 2GB free space).
  - Empty folder on host to mount NFS filesystem. This is just an empty folder with no space requirements and acts as a base folder for NFS mounts.
  - Sudo privileges for mount and umount. See the sample below assuming `delphix_os` is used as a Delphix user.
- Ensure any login banners similar to the one in the image below must be disabled on the staging host.

```
[staging_host]$ su - delphix
*******************************************************
*
*              Welcome to the Linux Terminal
*
*-----------------------------------------------------
*
*  You are now logged in as: delphix
*
*  Take caution with your actions and be responsible!
*
*******************************************************
Password:
```

9.14.3 Yugabyte installation and upgrade

9.14.3.1 Prerequisites

- Install a Delphix Continuous Data Engine of a version compatible with the connector version of choice (see page 2268).
- Install Yugabyte binaries on source, staging, and target servers.

9.14.3.2 Downloading the connector

The Connector is provided as a zip file on the Delphix download site. Follow the below steps to download the connector:

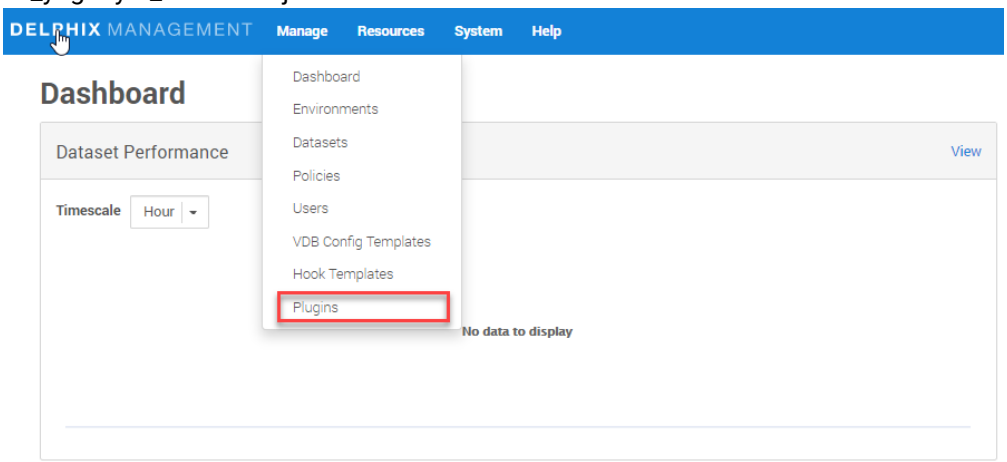
1. In the web browser, go to the Delphix download[671] site.
2. Log in to the download site using email and password credentials from the Delphix support portal.
3. Go to: **Delphix Product Releases** -> **Continuous Data Connectors** -> **Yugabyte** -> **v<version_number>** -> **sc_yugabyte_<version_number>.json** and download the connector(JSON file).

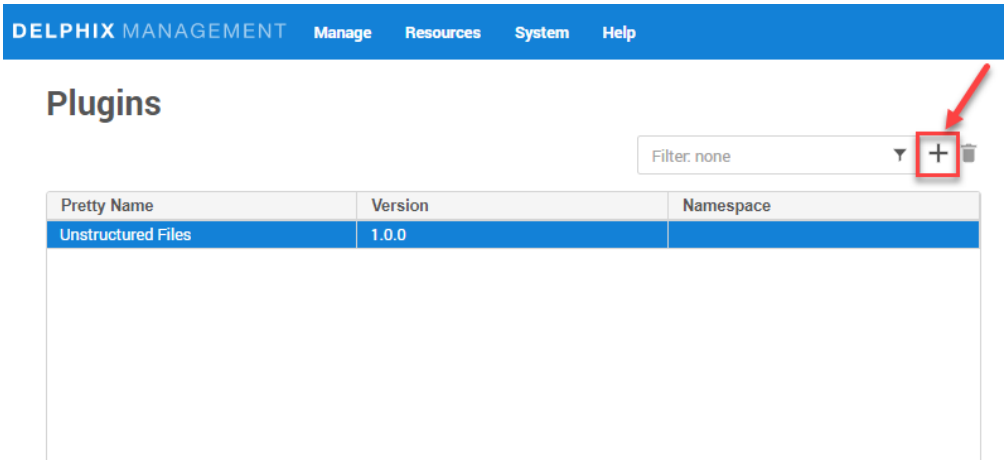
9.14.3.3 Installing the connector

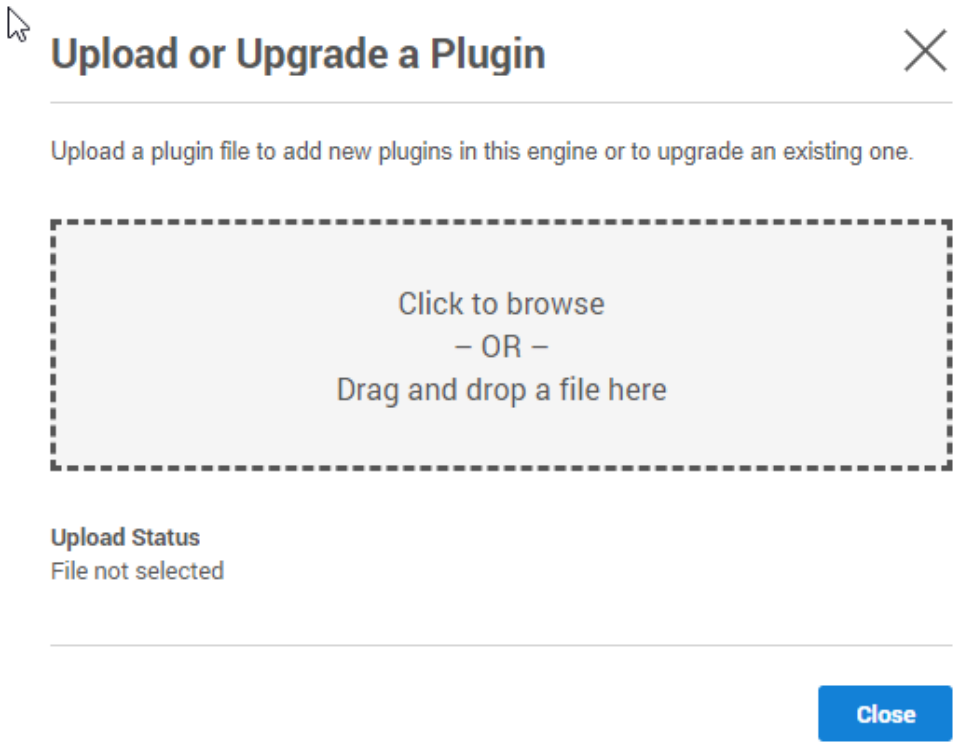
1. Login to the **Delphix Management** application.
2. Click **Manage > Plugins**.

671 https://download.delphix.com/

3. To upload a plugin, click the plus icon.
4. This step displays the **Upload or Upgrade a Plugin** dialog. Upload the <file_name>.json. For example, sc_yugabyte_<version>.json.







For more information, refer to the Continuous Data Engine connector management (see page 2257) section.

9.14.3.4 Upgrading the connector

The procedure to upload a higher version connector is the same as installing a new YugabyteDB connector. For more information, refer to the section “Upload connector to the Continuous Data”.

9.14.3.4.1 Instructions to upgrade a connector

1. Upload the higher version connector. For more information, refer to the Upload/Install connector in the Continuous Data section above.
2. Refresh all the corresponding environments.
3. Ensure connector discovery is completed successfully.
4. Take a new snapshot of the dSource and VDB.

While updating a connector, disable & enable of datasets is not required. Only refresh of environments is required.

9.14.3.5 Upgrading the Yugabyte version on the source/staging/target host

If the Yugabyte version at staging or target host is changed, the staging database and target database have to be recreated.

> If the existing staging database or target database is refreshed with upgraded installation, the database might not function properly.

9.14.4 Yugabyte connector operations

To successfully ingest and virtualize a YugabyteDB instance, you must perform the following connector operations in sequence:

1. Discovering Yugabyte installations (see page 2279)
2. Linking Yugabyte data sources (see page 2283)
3. Provisioning Yugabyte virtual databases (see page 2293)

The respective sections provide detailed information about prerequisites, limitations (if any), and the procedure to be followed for completing each of the operations.

9.14.4.1 Discovering Yugabyte installations

Environment discovery/refresh is a process that enables the YugabyteDB connector to determine Yugabyte installation details on a host. Database discovery is initiated during the environment setup process. Whenever there is any change (installing a new database home) to an already set-up environment in the Delphix Continuous Data Engine application, environment refresh must be performed.

9.14.4.1.1 Adding a Yugabyte environment

This topic describes the steps for adding a Yugabyte staging and target environment.

9.14.4.1.1.1 Prerequisites

- A Yugabyte staging environment must be set up with the requirements as described in Yugabyte requirements and prerequisites (see page 2268).
- Installation of the Yugabyte connector is required before the discovery operation.
- Install Yugabyte binaries on staging and target servers.
- The connector identifies the Yugabyte repository in the following cases:
    - Default installation in `/usr/bin` such that `yb-master` returns the binary path.

- Environment variable `$YUGABYTE_REPO` set on the staging/target host, which contains the binary path of `yb-master` on the host for the user with which the environment is added (in our case, delphix_os).
- Running `yb-master` or `yb-tserver` running on the host.

9.14.4.1.1.2 Procedure

Perform the following steps to add an environment(staging/target) to the Delphix Continuous Data Engine:

1. Login to the **Delphix Management** application.
2. Navigate to the **Manage > Environments** page.
3. Click the **Actions** ··· menu and select **Add Environment**.
4. From the **Add Environment** dialog window, select **Unix/Linux.**
5. Select **Standalone Host**.
6. Click **Next**.
7. Provide the name of the environment.
8. Provide the **Host IP address** or **host FQDN**.
9. Provide the **SSH port**. The default value is 22.
10. Provide an **Environment Username** for the environment.
11. Choose the **Login Type** that fits your setup.

- **Username and Password** – provide the environment username and password.
- **Username and Public Key** – provide the environment username.
- **Password Vault** – select from an existing Enterprise Password Vault.

Using Public Key Authentication
If you want to use public-key authentication for logging into your Unix-based environment, you have two options:

- Use the Delphix Continuous Data Engine's key pair. For more information on how to use the Delphix Continuous Data Engine's key pair, refer to the How to use the Continuous Data Engine's key-pair[672] section.
- Provide a key pair for this environment. As an alternative, a key pair specific for this environment can be provided via the API or CLI. For instructions, refer to CLI Cookbook: Setting up SSH key authentication for UNIX environment users (see page 2548).

672 https://docs.google.com/document/d/1WUkjGmv194W3WQ_UW5NxHjTFzjUPpF_ZZw0RdgvrN-8/edit#heading=h.shxgq976hoxd

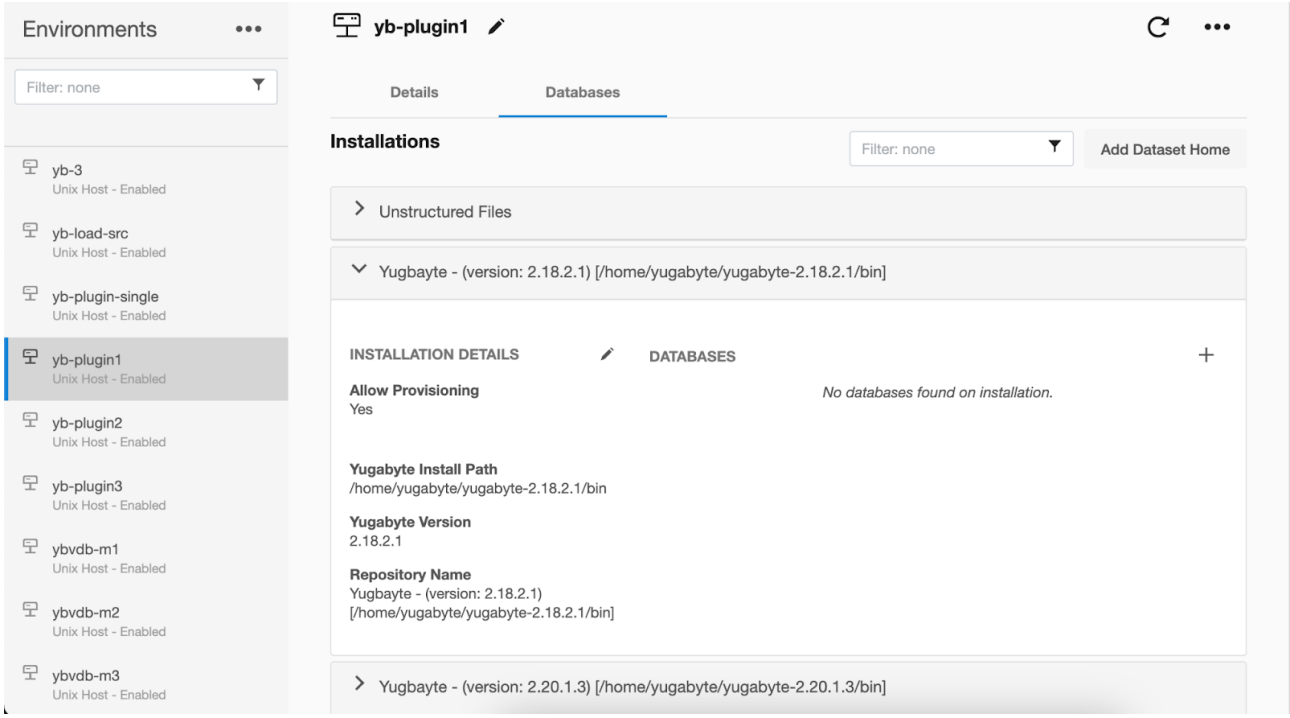
12. For Password, enter the password associated with the user in step 11.
13. If you want to use Public Key Encryption for logging into your environment:
    a. Select **Public Key** for the Login Type.
    b. Click **View Public Key**.
    c. Copy the public key that is displayed, and append it to the end of your `~/.ssh/authorized_keys` file. If this file does not exist, it must be created.

> The addition of the public key is a one-time requirement per user and per environment. Additionally, you have the option to add public key authentication to an environment user's profile via CLI (For more information, refer to the CLI Cookbook: Setting up SSH key authentication for UNIX environment users (see page 2548) topic).

14. For Password Login, click **Verify Credentials** to test the username and password.
15. Enter **Toolkit Path** (ensure that the toolkit path does not have any spaces).
16. To provide your own Oracle Java Development Kit, select the **Provide my own JDK** checkbox and click **Next**.
    a. On the **Java Development Kit** page, provide an absolute path to your Oracle JDK and click **Next**.
17. Click **Submit**.

After successfully adding and discovering the environment, you should be able to view the newly added environment listed under the Environments tab. If it does not appear, click the **refresh** icon in your browser.

After the environment has been successfully discovered if you wish to establish further links, add a source configuration to the previously discovered installation.



Perform the following steps to access the information about any environment you have created:

1. Navigate to **Manage** > **Environments**.
2. Select the environment name.

9.14.4.1.2 Adding an installation to a Yugabyte environment

If a new Yugabyte installation is added to the environment, then to reflect it into the Delphix Continuous Data Engine UI, it is required to refresh the environment. For more information, refer to the "Environment Refresh" and “Prerequisites” sections on the Environment management (see page 1017) page.

9.14.4.1.2.1 Add a source config

An environment contains repositories that are Yugabyte installations in the environment. Each environment may have any number of repositories associated with it.

A SourceConfig object defines the configuration of the dSource and is required to create a dSource. You can create any number of SourceConfig objects using a repository, that represents known database instances.

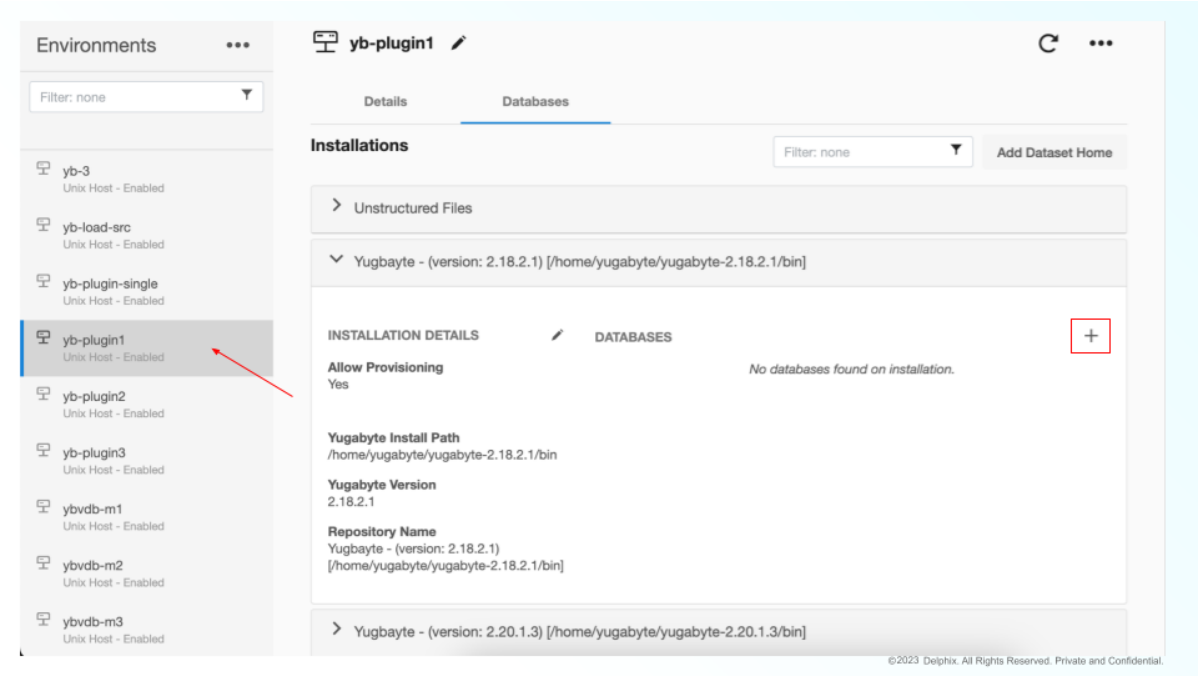
For the Yugabyte connector, the source config must be created manually.

Procedure

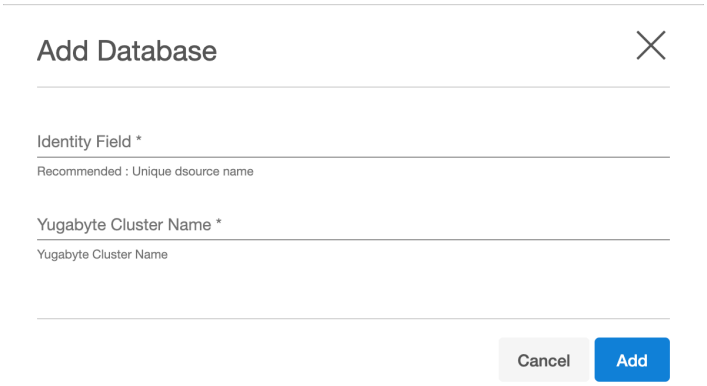
Perform the following steps to create a source config on the staging environment.

1. Login to the **Delphix Management** application.
2. Click **Manage**.

3. Select **Environments**.
4. Select the repository.
5. Click the plus icon.



6. Enter a unique name for the staging database in the identify field.
7. Enter a cluster name for the database in the **Yugabyte Cluster Name** field.
8. Click the **Add** button.



9.14.4.2 Linking Yugabyte data sources

The Delphix Continuous Data Engine provides the following ingestion method for linking the dSource from a Yugabyte source database.

- Linking a Yugabyte dSource (see page 2284)

9.14.4.2.1 Linking a Yugabyte dSource

Linking a data source will create a dSource object on the Delphix Continuous Data Engine and allow the Delphix Continuous Data Engine to ingest data from this source. The dSource is an object that the Delphix Continuous Data Engine uses to create and update virtual copies of your database.

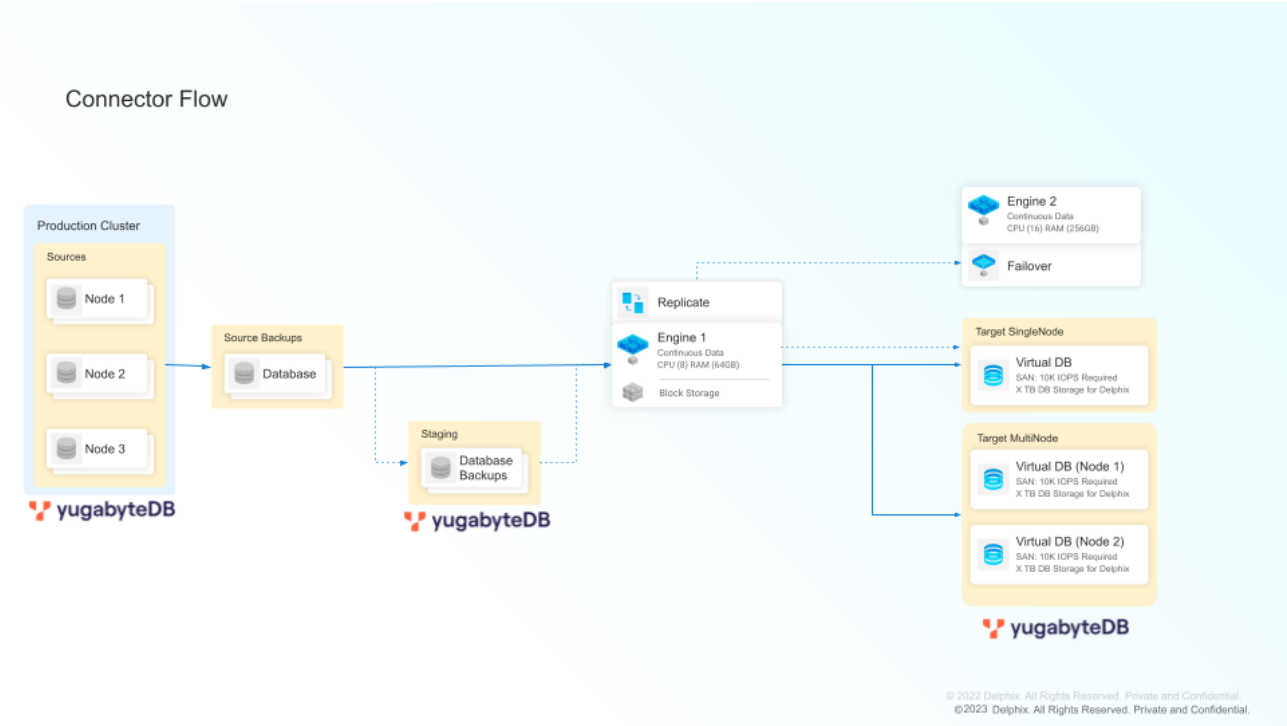
> An environment can be associated with only one dSource/VDB at a time.

9.14.4.2.1.1 Prerequisites

A staging environment must be created and the source database must be configured in the discovered repository.

9.14.4.2.1.2 Creating a dSource

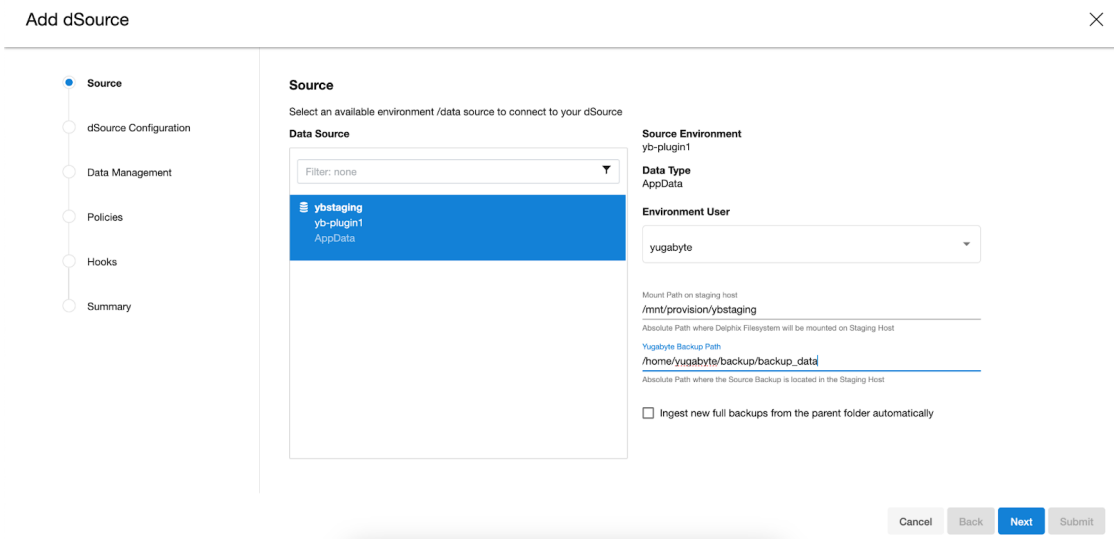
Below is the brief architecture of the Delphix Continuous Data Engine support for the Yugabyte database:



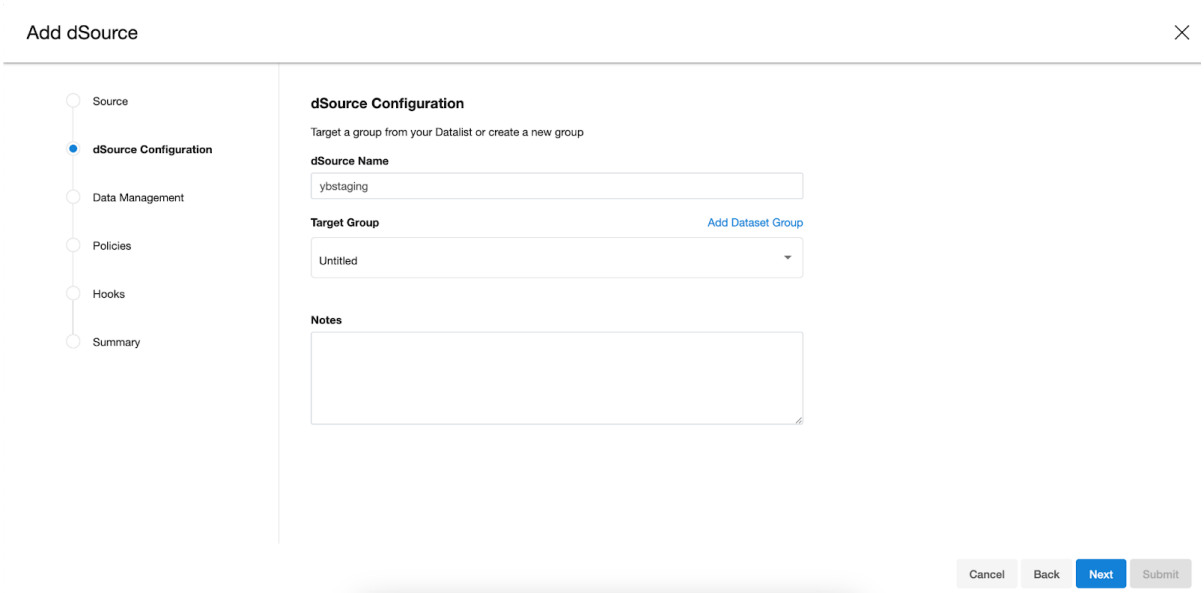
9.14.4.2.1.3 Creating a dSource with ingestion type (NFS)

1. Login to the **Delphix Management** application.
2. Click **Manage** > **Datasets**.

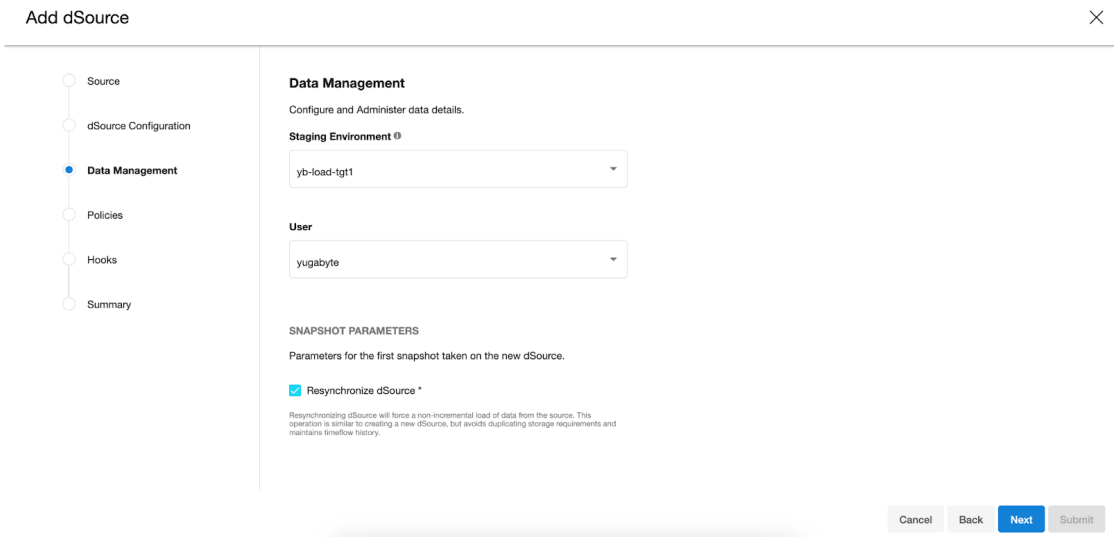
3. Select **Add dSource**.
4. In the **Add dSource** wizard, select the Yugabyte source configuration that is created on the staging host.
5. Enter the details for the mount path available on the staging host. This empty folder acts as a base for NFS mounts.
6. Enter details for source or production Yugabyte cluster backup. This should be an absolute Linux path.
7. Select the I**ngest new backups from the parent folder automatically** checkbox if it is required to ingest new backups on every snapshot.
    - If the source backup path provided is `/home/yugabyte/backup/20240310043200`, the connector will look for new backups in the folder `/home/yugabyte/backup` on every snapshot.
    - **Note:** If automatic ingestion of new backups is enabled, it is necessary for the backup folder to be named in DateTime format as: `%Y%m%d%H%M%S`.



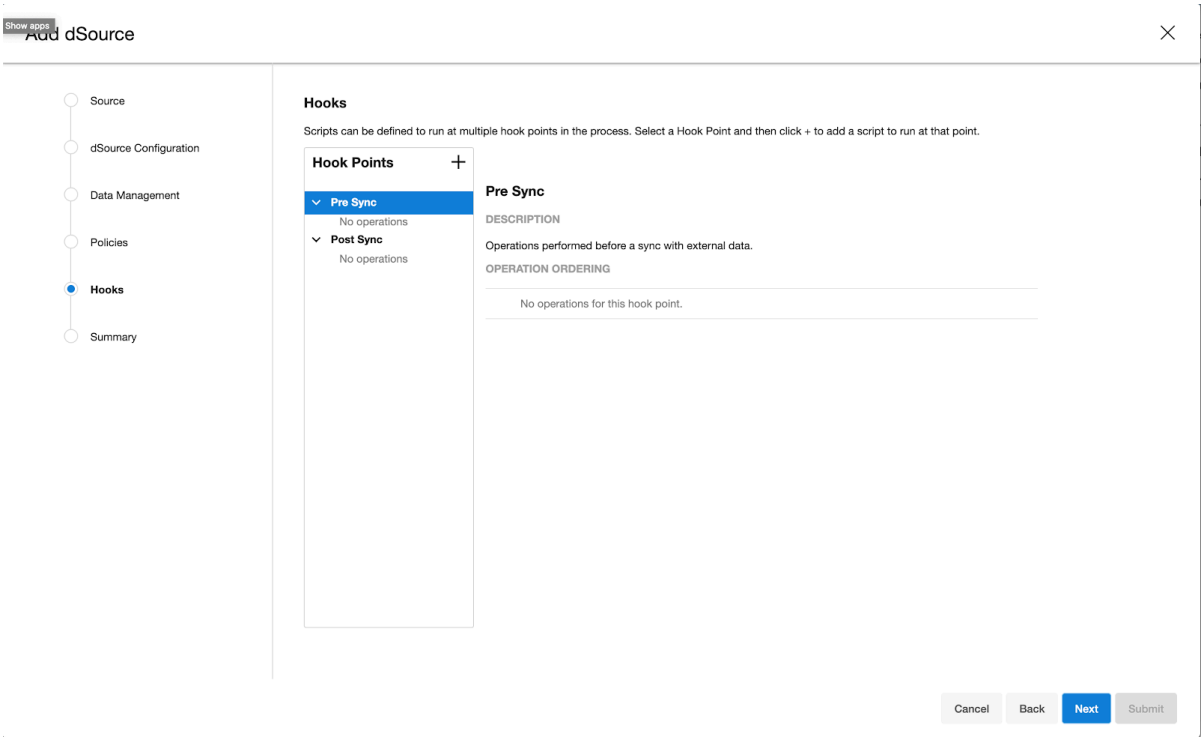
8. Provide a **dSource Name** and **Target group** on the dSource configuration page.

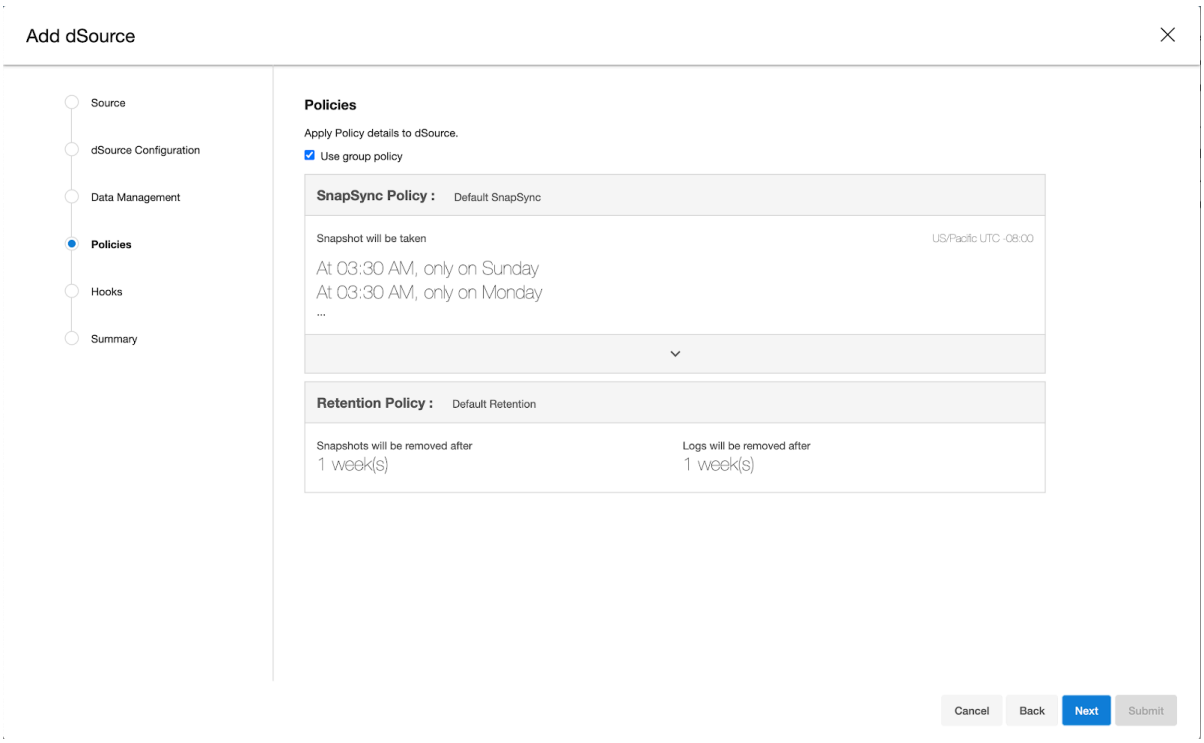


9. On the **Data management** page, select the following:
   - **Staging Environment:** This will be your staging host where the source config was created.
   - **User:** Database OS user with required privileges for linking the dataset.

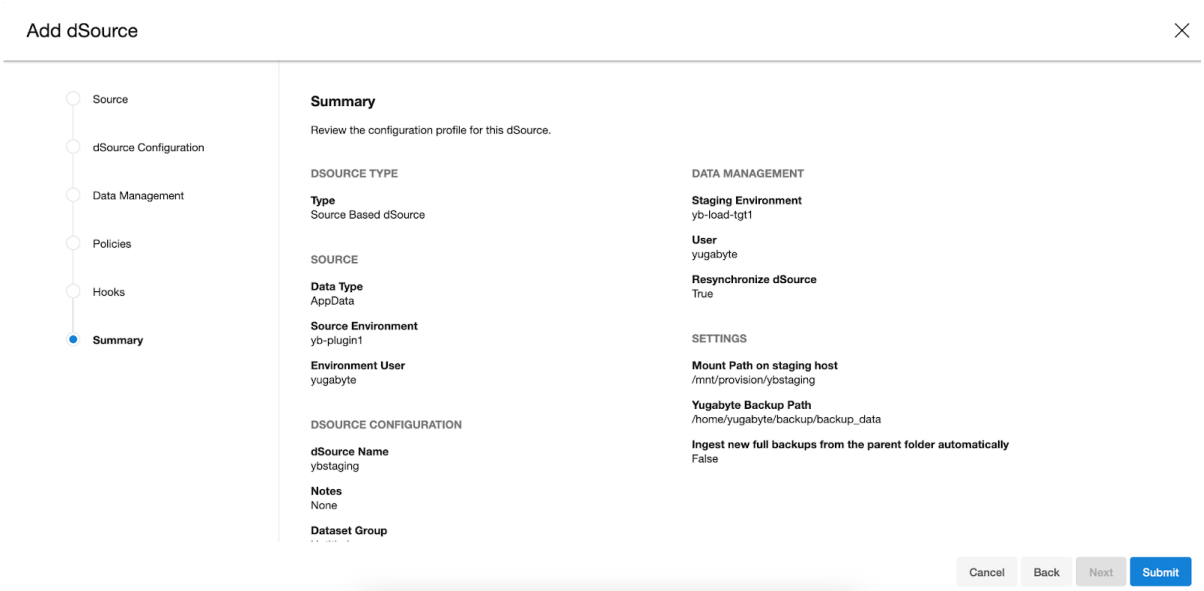


10. On the next screens, configure policies, hooks, and review the configuration, then click on the **Next** button to view the summary.

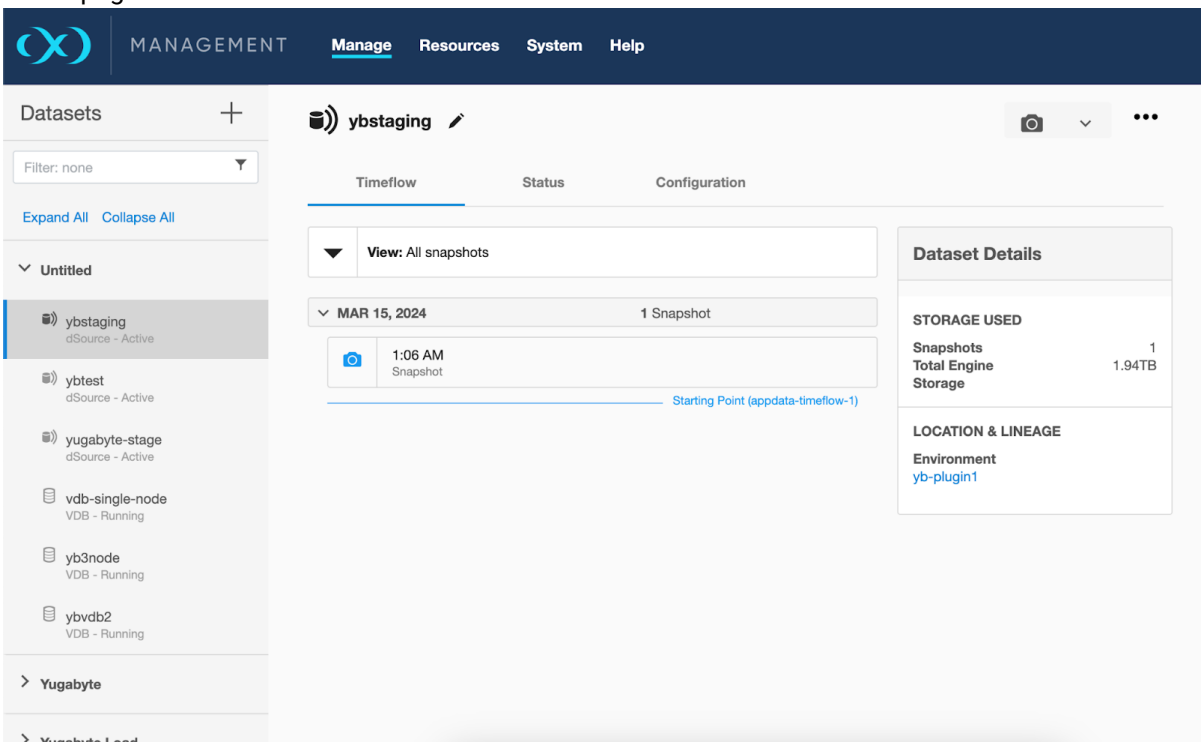




11. Click the **Submit** button which will initiate the linking process.



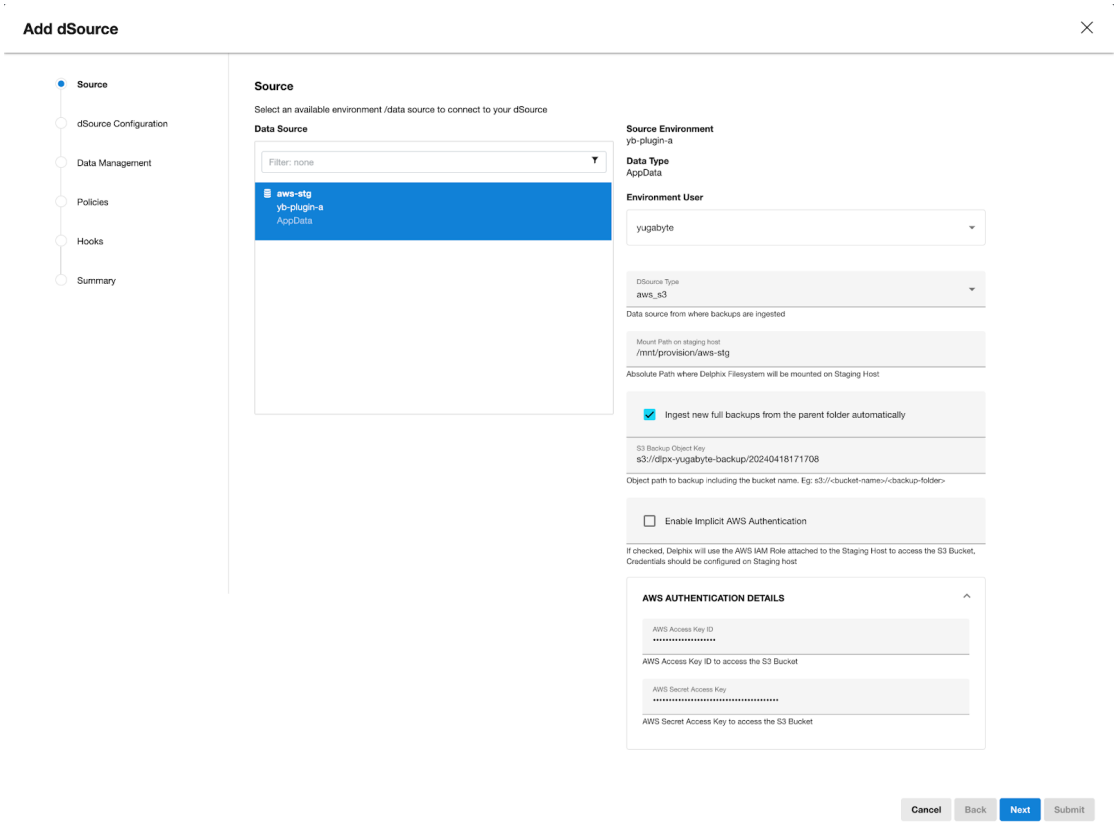
12. Once dSource is created successfully, you can review the datasets on **Manage** > **Datasets** > **dSource Name** page.



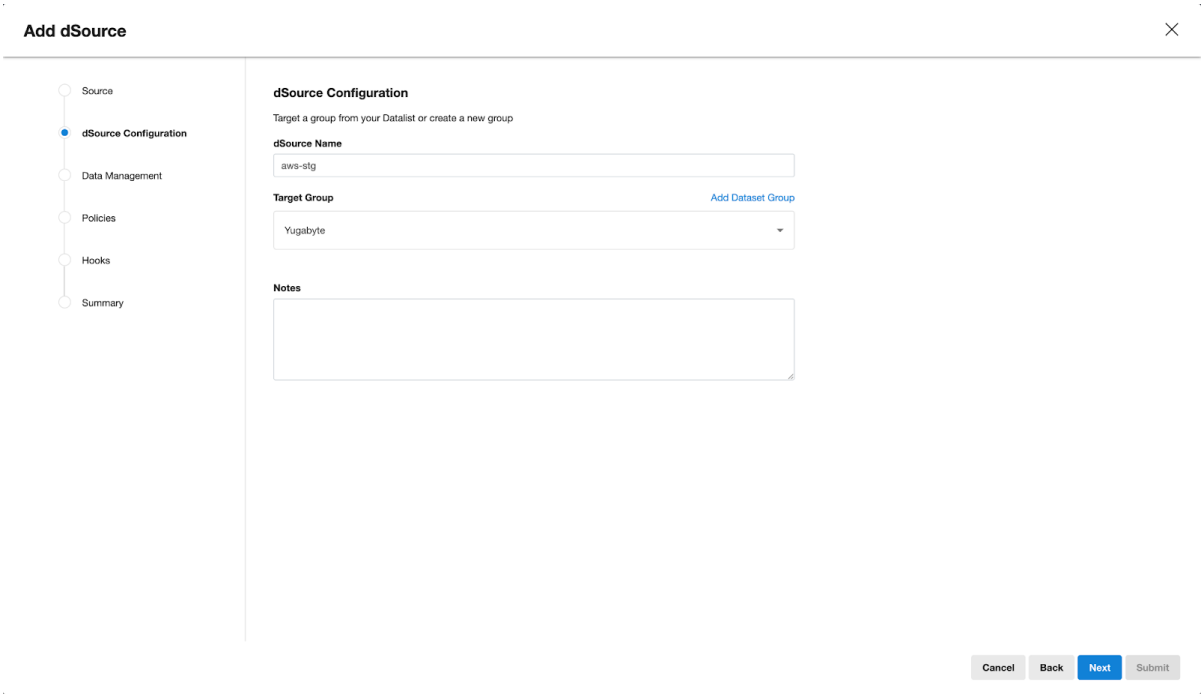
Creating dSource with ingestion type (AWS S3)

1. Login to the **Delphix Management** application.
2. Click **Manage** > **Datasets**.

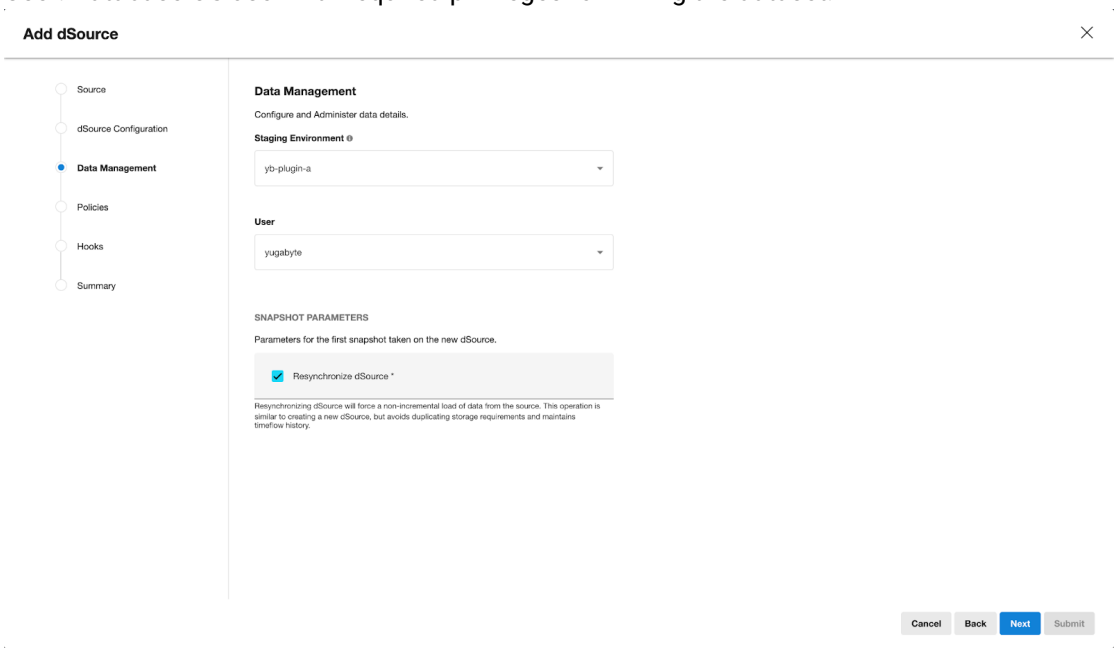
3. Select **Add dSource**.
4. In the Add dSource wizard, select the Yugabyte source configuration which is created on the staging host.
5. Enter the details for Mount Path available on the staging host. This empty folder acts as a base for NFS mounts.
6. Enter details for Source (or Production) **S3 Backup Object Key**. This should be an S3 Bucket path.
7. Check the box Ingest new backups from the parent folder automatically if it is required to ingest new backups on every snapshot.
    - If Source Backup S3 Bucket Path provided is s3://dlpx-yugabyte-backup/20240418171708, the connector will look for new backups in the folder s3://dlpx-yugabyte-backup/ on every snapshot.
    - **Note:** If automatic ingestion of new backups is enabled, it is necessary for the backup folder to be named in date time format as: ***%Y%m%d%H%M%S***



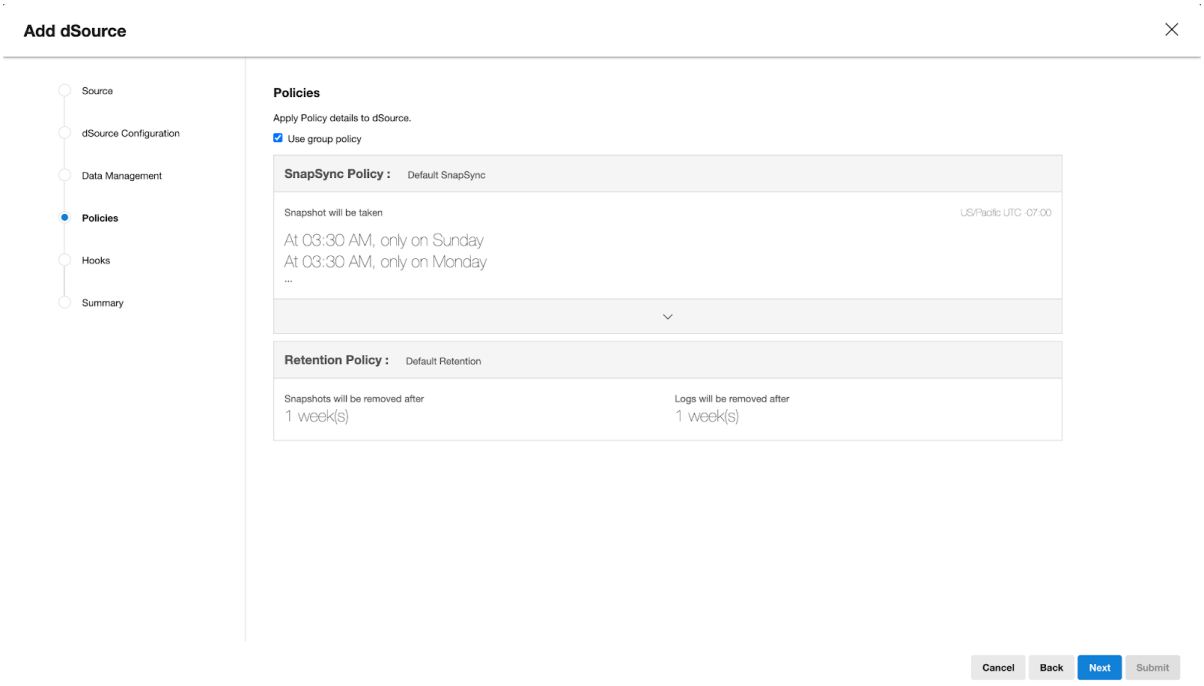
8. Provide the details for **dSource Name** and **Target group** on the dSource configuration page.

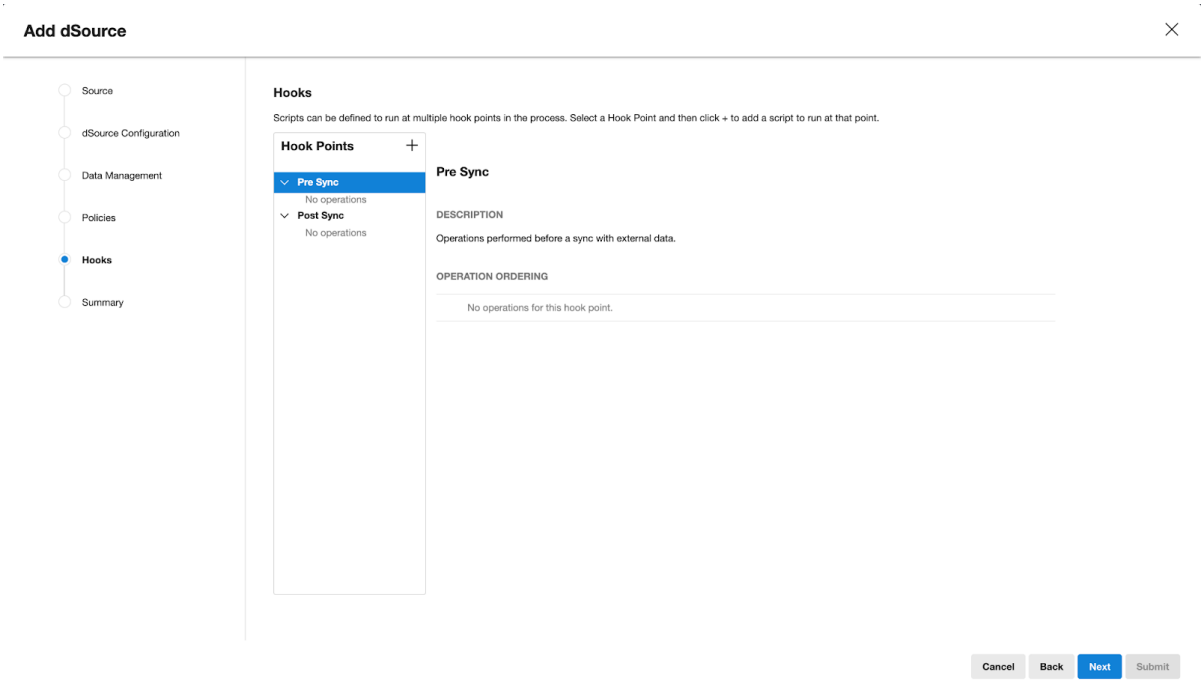


9. On the **Data management** page, select the following:
   - Staging Environment: This will be your staging host where source config was created.
   - User: Database OS user with required privileges for linking the dataset.

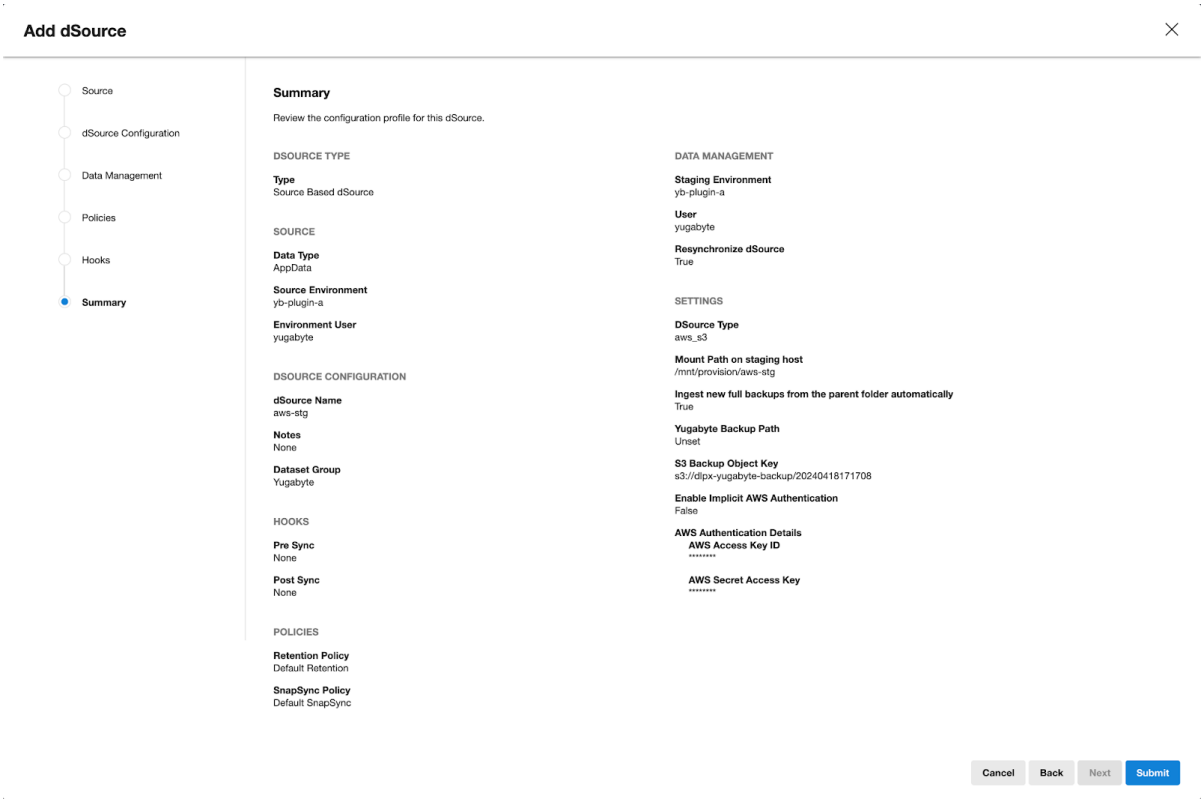


10. On the next screens, configure a policy, hooks, and review the configuration, then click on **Next** button to view the summary.

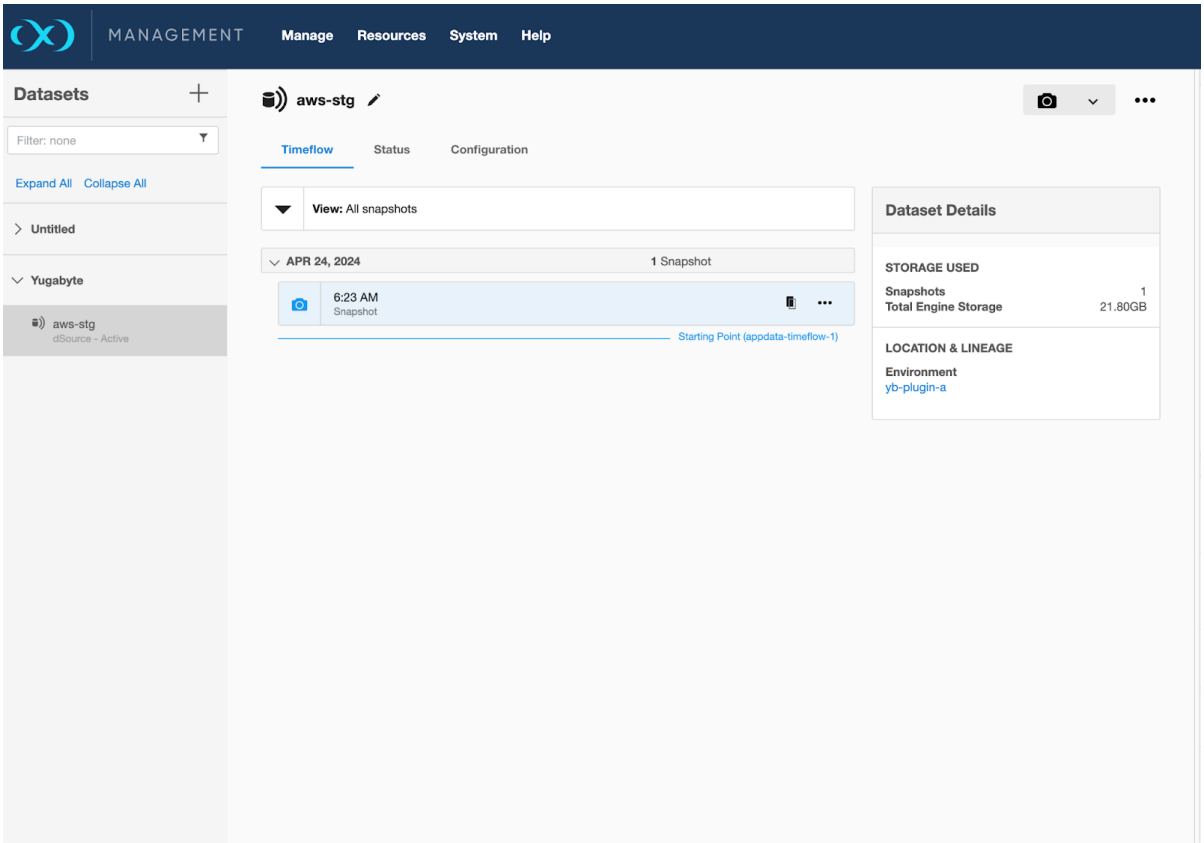




11. Click the **Submit** button which will initiate the linking process.



12. Once dSource is created successfully, you can review the datasets on **Manage** > **Datasets** > **dSource Name**.



9.14.4.3 Provisioning yugabyte virtual databases

To provision a virtual database (VDB), you can either use an existing dSource snapshot or use another VDB snapshot for VDB to VDB provisioning. A snapshot is a timeflow created in a dSource or VDB.

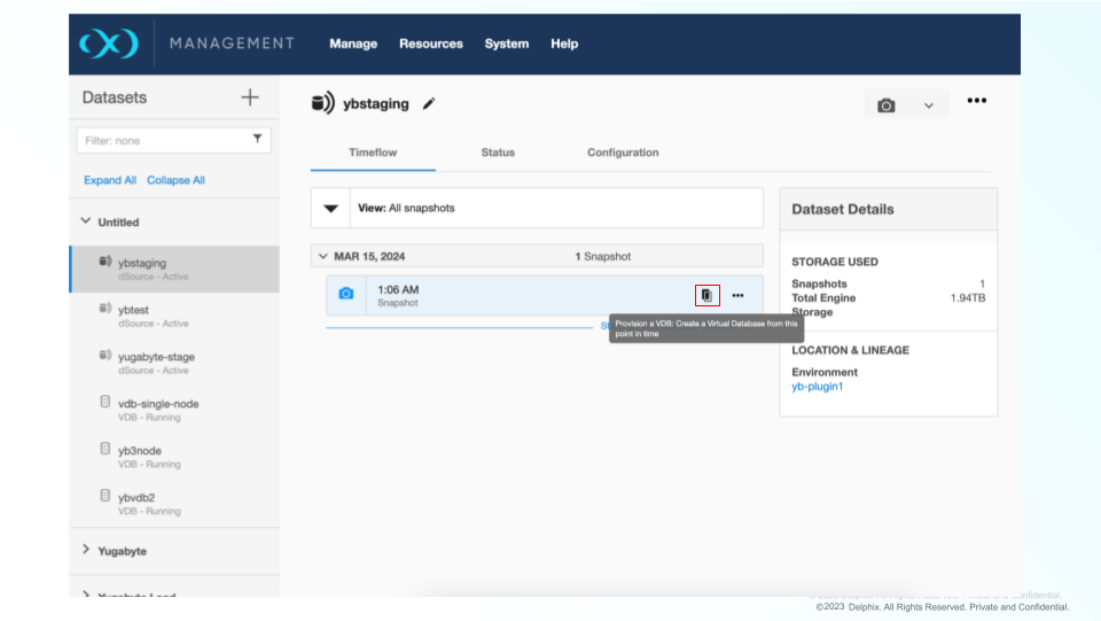
> An environment can be associated with only one dSource/VDB at a time.

9.14.4.3.1 Prerequisites

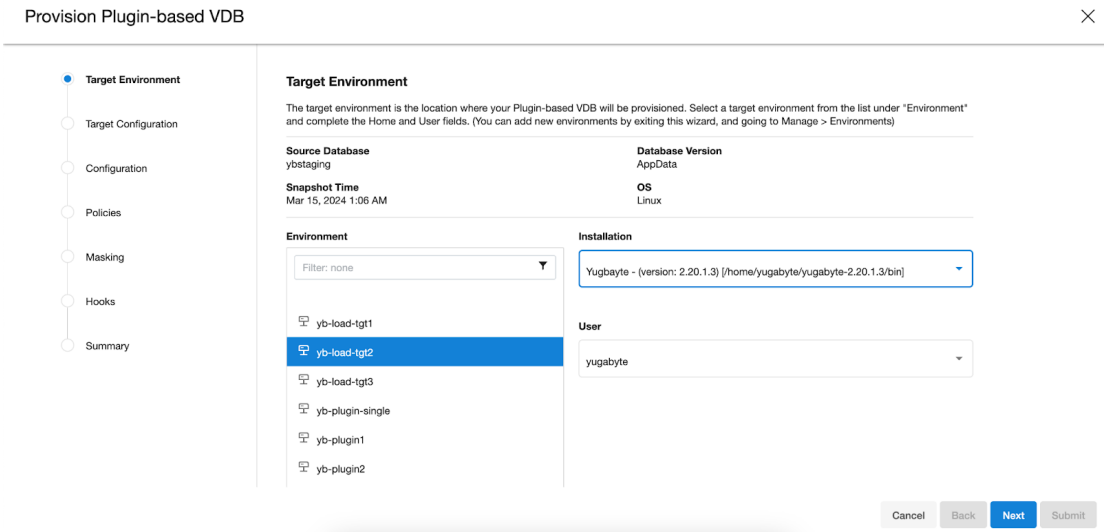
- You should either have linked a dSource from a staging instance following the instructions in "Linking a Yugabyte dSource (see page 2283)," or you should have already created a VDB from which you intend to provision another VDB.
- A Yugabyte target environment must be set up with the requirements as described in Yugabyte support and requirements (see page 2268).
- A Yugabyte binary with the same version as the source database must be installed on the target host.

9.14.4.3.2 Provisioning a Yugabyte VDB

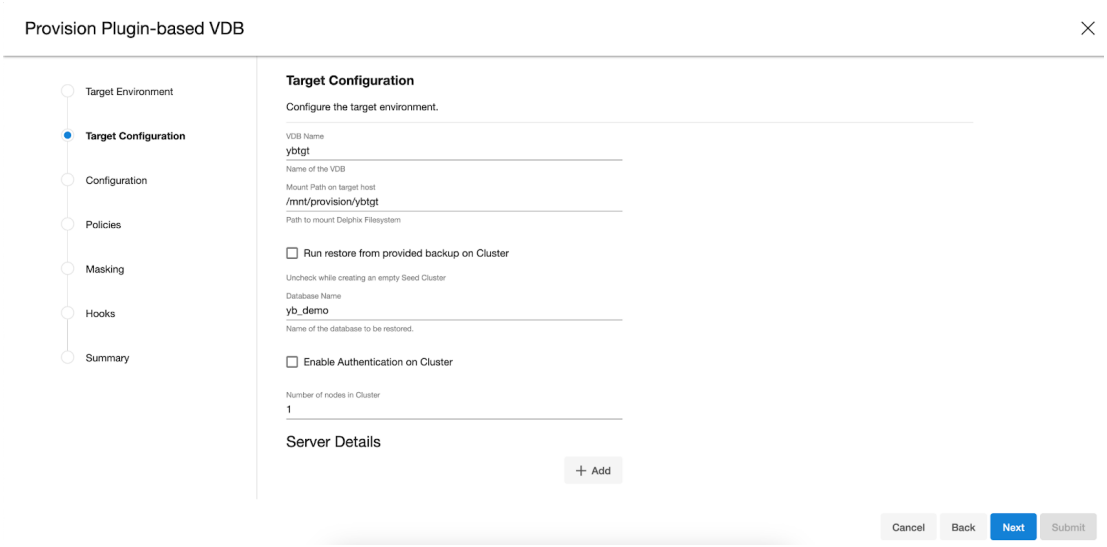
1. Navigate to **Manage**, and select **Datasets**.
2. Select the dataset and the snapshot that you want to provision from. Click the provision VDB icon to open the provision VDB wizard.



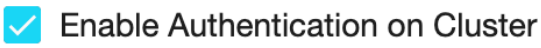
3. Select the target host from the dropdown on which VDB needs to be created.

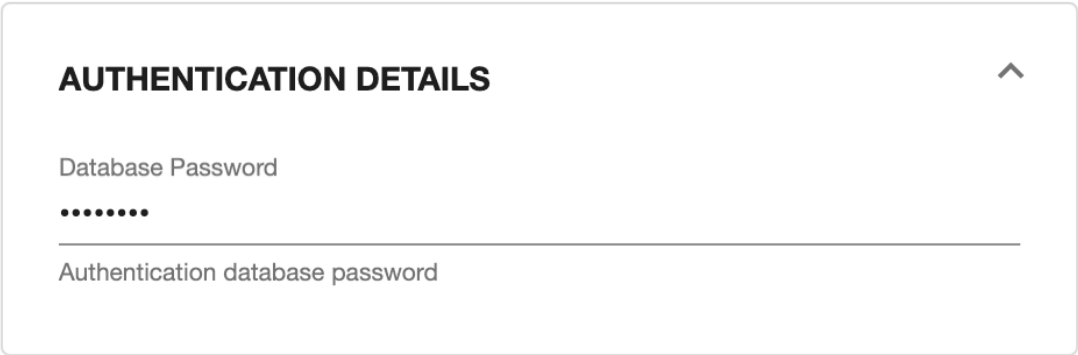


4. Enter the following values for the target configuration:

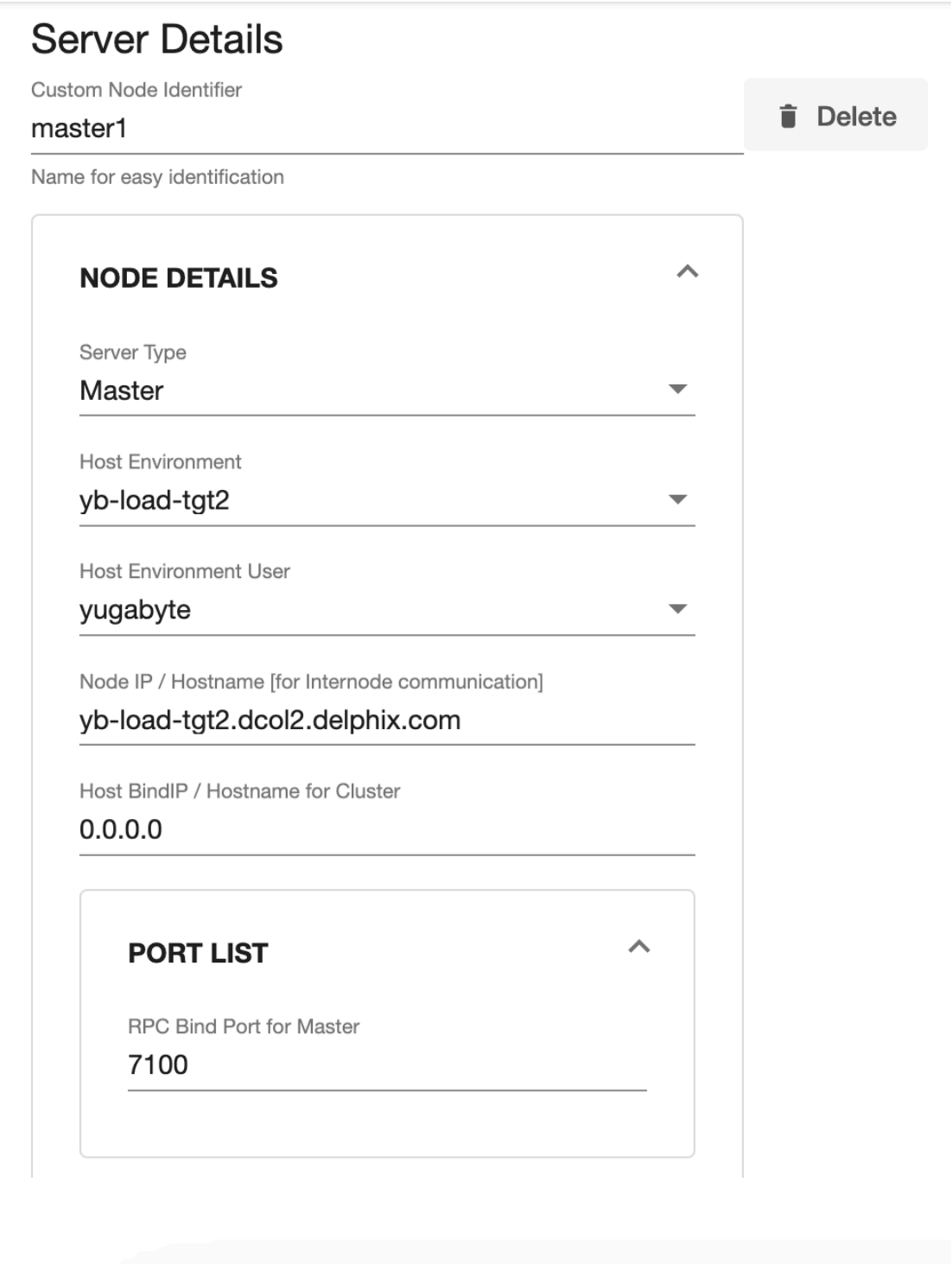


a. **VDB Name**: A unique database identifier.
b. **Mount Path**: NFS mount path where dSource snapshot will be mounted by Engine (ex. `/mnt/provision/targetdemo`)
c. **Run Restore from the provided backup:** Select the checkbox if the restore needs to be performed on the created cluster. Unselect the checkbox in case of a seed (empty) database.
d. **Database Name:** Database to be restored from backup.
e. **Enable Authentication:** Check to enable authentication on the created cluster.



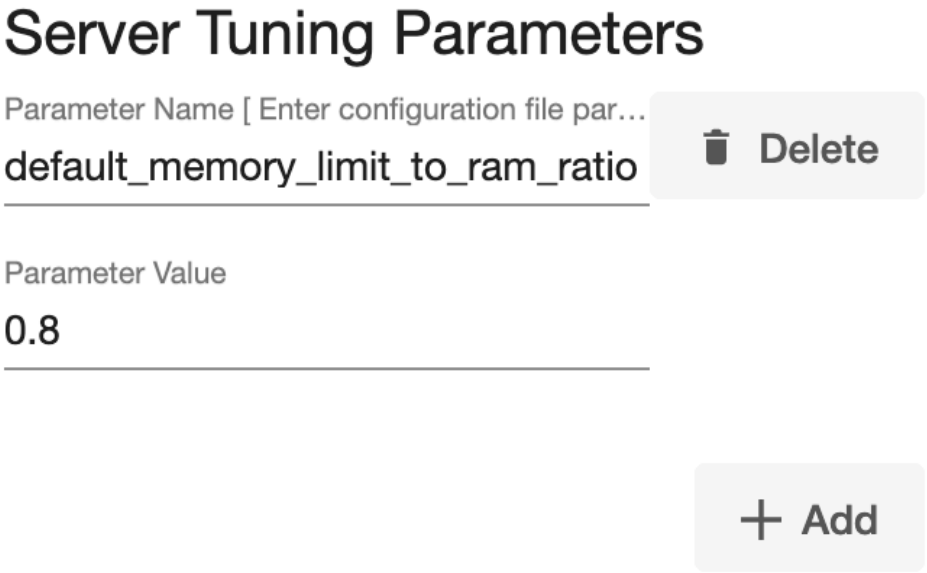


f. **Database Password:** This field will be available if authentication is enabled. Fill in the desired password corresponding to the Yugabyte user.
g. **Number of Nodes in Cluster:** Enter the number of master and tserver pairs in the cluster to be created.
h. **Server Details:** Enter each server (master / tserver) details.

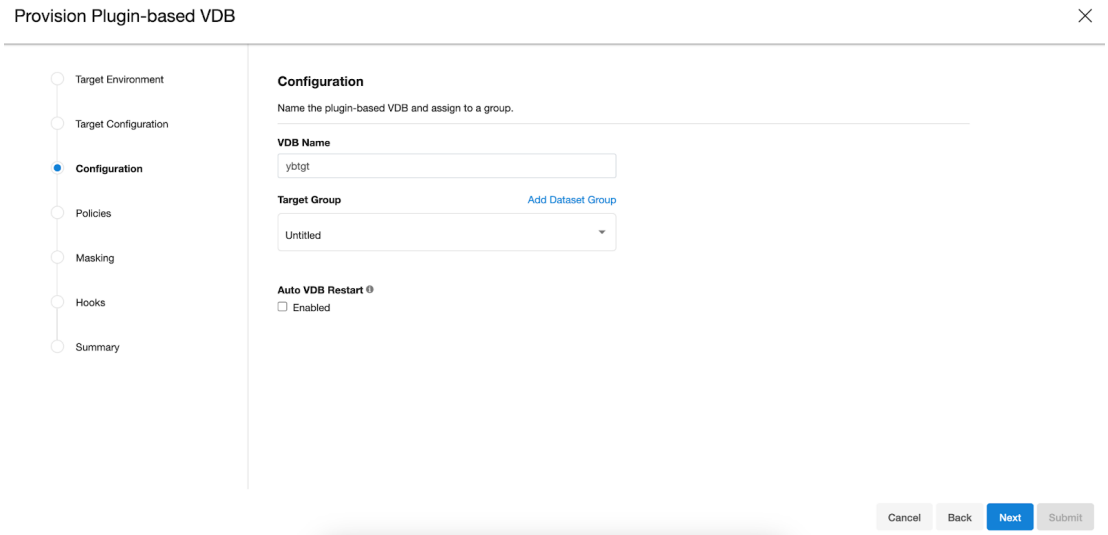


i. **Custom Node Identifier:** A custom identifier for easy identification for the user (not used by the connector).
ii. **Server Type:** Select the type of server: master or server
iii. **Host Environment:** Select the host on which the master/tserver has to be provisioned.
iv. **Host Environment User:** Select the user corresponding to the host.

v. **Node IP:** IP using which the node could be reached from other nodes. It needs to be a public IP specifically in the case of multinode architecture.
vi. **Host BindIP / Hostname for Cluster:** Listen to address for the node
vii. **Port List:** Specific ports required by the master/tserver for the server to be initiated.
viii. **Server Tuning Parameters:** Custom tuning parameters that are optional to the customer and are used to configure the instance.

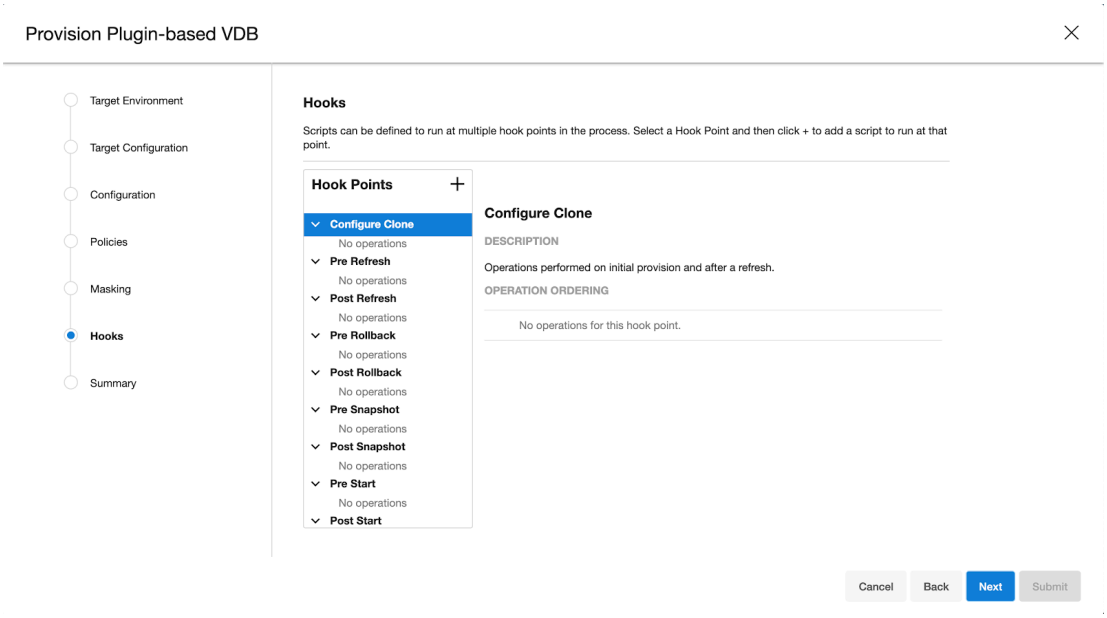


5. Provision connector-based VDB. Enter the following value:
   a. **VDB Name**: Delphix target cluster name
   b. **Target group**: Delphix target cluster group



6. Select a policy for VDB, and select **Next**.
7. Select masking for VDB if needed, and select **Next**.

8. Add hooks for VDB if needed, and select **Next**.

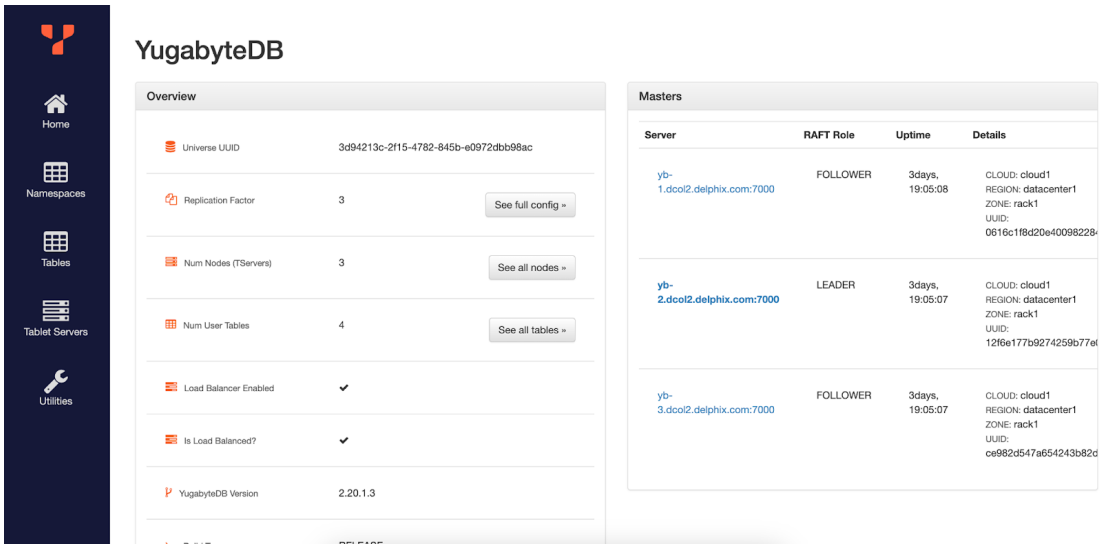


9. Preview the summary and select **Submit**.
10. Once the VDB is created successfully, you can review the datasets on **Manage** > **Datasets** > **vdb Name**.

9.14.4.3.2.1 Accessing Yugabyte target VDB cluster

Use an IP/hostname of the target environment and VDB port defined above to access the target cluster VDB. The admin user name and password are defined based on input from point 4.

An example screenshot of the Yugabyte console of the provisioned VDB can be found below:



9.14.5 Yugabyte troubleshooting

9.14.5.1 Logging

There are the following sets of logs for the Yugabyte connector.

1. The connector logs are part of the Delphix Continuous Data Engine. Refer to How to Retrieve Logs[673] to find out how to get the Delphix connector Logs.
2. Yugabyte Instance logs:
   a. YugabyteDB Instance logs are located at:
      i. `<mount_path>/<node_name>/yb-data/master/logs for master`
      ii. `<mount_path>/<node_name>/yb-data/tserver/logs for server`

9.14.6 Yugabyte FAQs

9.14.6.1 How to use the Delphix Continuous Data Engine key pair?

Perform the following steps to use the Delphix Continuous Data Engine key pair:

1. Select **Public Key** for the **Login Type.**
2. Click **View Public Key**.
3. Copy the public key that is displayed, and append it to the end of your `~/.ssh/authorized_keys` file. If this file does not exist, you must create it.
   a. Run `chmod 600 ~/.ssh/authorized_keys` to allow only the file's owner to read and write to it (make sure the file is owned by you).
   b. Run `chmod 755 ~` to restrict access to your home directory so that no other may write to it.
   c. Run `chmod 700 ~/.ssh` so that others cannot write to it. The `~/.ssh` directory is not writable by group or others. Otherwise, authentication will fail.

9.14.7 Yugabyte developers guide

This section includes the following CLI operations.

- Yugabyte sourceconfig operations (see page 2300)
- Yugabyte linking operations (see page 2301)
- Yugabyte provisioning operations (see page 2303)
- Yugabyte discovery operations (see page 2300)

673 https://developer.delphix.com/References/Logging/#how-to-retrieve-logs

9.14.7.1 Yugabyte discovery operations

9.14.7.1.1 Command Line Interface (CLI)

9.14.7.1.1.1 Discovery operations

To discover an environment via CLI, log in to the Delphix Continuous Data Engine using the admin user account. (Connection to Delphix Continuous Data Engine with FDQN " `dlpx-engine.delphix.com` ")

```
ssh admin@dlpx-engine.delphix.com
```

9.14.7.1.1.2 Discovering the environment

Run the following commands to discover the environment. (Adding environment with FQDN " `yb-load-tgt1.dcol2.delphix.com` ", hostname " `yb-load-tgt1` ", toolkit-path "M", os-user " `delphix_os` ", os-password " `delphix` ")

Run the following commands to discover the environment.

```
Example -
environment
create
set hostEnvironment.name=yb-load-tgt1.dcol2.delphix.com
set hostParameters.host.address=yb-load-tgt1
set hostParameters.host.toolkitPath=/var/tmp
set primaryUser.name=delphix_os
set primaryUser.credential.password=delphix
commit
```

9.14.7.2 Yugabyte sourceconfig operations

To create a source config and perform updates from the CLI, login to the Delphix Continuous Data Engine using the admin user account.

```
ssh admin@dlpx-engine.delphix.com
```

9.14.7.2.1 Add a source config creation via CLI

Run the following command to add the source config.

```
Example -
sourceconfig create
set type=AppDataStagedSourceConfig
set name=testscfg
set parameters="{ \
\"cluster_name\":\"Staging\", \
\"pretty_name\":\"testscfg\"
}"
set repository="yb-load-tgt1/Yugbayte - (version: 2.20.1.3) [/home/yugabyte/
yugabyte-2.20.1.3/bin]"
commit
```

9.14.7.2.2 Delete

Run the below command to delete sourceconfig.

```
Example -
sourceconfig create
select testscfg
delete
commit
```

9.14.7.3 Yugabyte linking operations

9.14.7.3.1 dSource creation via CLI

To create a dSource via CLI, you must create a source configuration first. For more information on creating a source configuration, refer to Add Source Config using CLI (see page 1041).

To create a dSource and perform operations via CLI, log in to the Delphix Continuous Data Engine using the admin user account.

```
ssh admin@dlpx-engine.delphix.com
```

9.14.7.3.2 Set default linking parameters

This instruction is common for all types of ingestions.

Run the below commands to set the default linking parameters. The below example uses `testscfg` source config.

```
Example -
database link
set type=LinkParameters
set linkData.type=AppDataStagedLinkData
set name=testds
set group=Untitled
set linkData.config=testscfg
set linkData.environmentUser=yb-load-tgt1/delphix_os
set linkData.stagingEnvironment=yb-load-tgt1
set linkData.stagingEnvironmentUser=yb-load-tgt1/delphix_os
set linkData.syncParameters.parameters={\"resync\":true}
```

9.14.7.3.3 Link parameters for ingestion using NFS backup:

Run the below commands to link parameters:

```
Example -
set linkData.parameters="{ \
\"mount_path\":\"/mnt/provision/testds\", \
\"backup_location\":\"/home/yugabyte/backup/20240310040000\", \
\"ingest_new_backups\":true \
}"
```

9.14.7.3.4 Commit

Run the below command to commit the change

```
commit;
```

9.14.7.3.5 Disable

Run the below command to disable the dSource.

```
Example -
source
select testdb
disable
commit
```

9.14.7.3.6 Enable

Run the below command to enable the dSource.

```
Example -
source
select testdb
enable
commit
```

9.14.7.4 Yugabyte provisioning operations

9.14.7.4.1 Provision operations via CLI

To create a VDB via CLI, you must first create a dSource. For more information on creating a dSource, refer to Link operation via CLI section (see page 2301).

To create a VDB and perform operations via CLI, log in to the Delphix Continuous Data Engine using the admin user account.

```
ssh admin@dlpx-engine.delphix.com
```

9.14.7.4.2 Provision

Run the below commands to provision a VDB. The below example uses `testdb` dSource.

```
Example -
database provision
set type=AppDataProvisionParameters
set container.name=testvdb
set container.group=Untitled
set source.allowAutoVDBRestartOnHostReboot=true
set sourceConfig.repository="yb-load-tgt2.delphix.com/delphix_os vFiles (15.2)"
set timeflowPointParameters.container=testdb
set sourceConfig.name=testscfgv
set source.parameters="{ \
\"vdb_name\":\"testvdb\", \
\"run_restore\":false, \
\"database_name\":\"yb_demo\", \
\"mount_path\":\"/mnt/provision/testvdb\", \
\"enable_authentication\":true, \
\"authentication_details.password\":\"delphixpwd\", \
\"num_nodes\":1, \
\"server_config\": [ \
    { \
        \"node_id\":\"master1\", \
        \"node_details.server_type\":\"Master\", \
        \"node_details.host_ref\":\"yb-load-tgt2\", \
        \"node_details.user_ref\":\"yb-load-tgt2/delphix_os\", \
```

```
        \"node_details.master_ip\":
\"yb-load-tgt2.dcol2.delphix.com\", \
        \"node_details.hostname\":\"0.0.0.0\", \
        \"node_details.port_list.master_rpc_port\":7100, \
        \"node_details.tuning_params\": [ \
            { \
                \"parameter_name\":
\"default_memory_limit_to_ram_ratio\", \
                \"parameter_value\":\"0.8\" \
            } \
        ] \
    }, \
    { \
        \"node_id\":\"tserver1\", \
        \"node_details.server_type\":\"TServer\", \
        \"node_details.host_ref\":\"yb-load-tgt2\", \
        \"node_details.user_ref\":\"yb-load-tgt2/delphix_os\", \
        \"node_details.hostname\":\"0.0.0.0\", \
        \"node_details.port_list.tserver_rpc_port\":9100, \
        \"node_details.port_list.tserver_pgsql_port\":5433, \
        \"node_details.port_list.tserver_csql_port\":9042, \
        \"node_details.tuning_params\": [ \
            { \
                \"parameter_name\":
\"default_memory_limit_to_ram_ratio\", \
                \"parameter_value\":\"0.8\" \
            } \
        ] \
    } \
}"
commit
```

9.14.7.4.3 Disable

Run the below commands to disable a dSource. You must select the same sourceConfig name that was given during VDB creation.

```
Example -
source
select testscfgv
disable
commit
```

9.14.7.4.4 Enable

Run the below command to enable a dSource.

```
Example -
source
select testscfgv
enable
commit
```

9.14.7.4.5 Stop

Run the below command to stop a dSource.

```
Example -
source
select testscfgv
stop
commit
```

9.14.7.4.6 Start

Run the below command to start a dSource.

```
Example -
source
select testscfgv
start
commit
```

10 Best practices

For optimal usage of Delphix, there are many different configurations and best practices you'll need to note to tune your engines for peak performance. This ranges from settings on the hypervisor that runs Delphix to the setting on your sources, virtual databases, and the operating systems on which they run. Find out more information based on each category below.

These pages are the starting point to begin your architecture planning for the best possible Delphix deployment. To function as a high-performance virtual appliance, we need to ensure the underlying infrastructure components are deployed in an ideal manner consistent with our best practices - these pages are to help you achieve that goal.

10.1 Introduction to Delphix architecture

Database virtualization with Delphix (see page 538)

10.2 Hypervisor and host

Best practices for hypervisor host and VM guest (see page 2327)

10.3 Network

Best practices for network configuration (see page 2331)

CLI Cookbook: network performance (see page 2740)

Optimal network architecture for the Delphix engine (see page 719)

Network operations using the Delphix session protocol (see page 722)

Network performance expectations and troubleshooting (see page 731)

10.4 Storage

Best practices for storage (see page 2348)

Optimal storage configuration parameters for the Delphix engine (see page 896)

Storage performance expectations and troubleshooting (see page 903)

10.5 Data protection

Best practices for Delphix engine data protection (see page 2334)

10.6 Source DB and OS settings

Best practices for source DB and OS settings (see page 2337)

10.7 Target DB and OS settings

Best practices for target DB and OS settings (see page 2338)

10.8 Staging target

Best practice for staging targets (see page 2344)

10.9 Architecture checklist (consolidated best practices)

Architecture Checklist (see page 2313)

10.10 Architecture checklist FAQ

Architecture checklist FAQ (see page 2313)

10.11 Architecture checklist FAQ

10.11.1 Architecture best practices overview FAQ

These questions relate to the standard processes our Solutions Architecture team executes when engaged for Architecture review of a customer's environment.

10.11.1.1 Why go through an architecture or sizing process?

To measure - when possible - the workload in your environment and calculate the number of engines needed

- To ensure best practices are applied so your engine(s) perform to the level expected
- To find and avoid or work around potential challenges in an environment
- To maximize your license value

10.11.1.2 Why does Delphix collect IO data?

The collection of IO data allows us to more accurately size the Delphix Engine rather than making recommendations based on assumptions or synthetic data. The data you provide for analyses is from the SOURCE and TARGET DB servers.

10.11.1.3 Why is the customer database and inventory (CDI) important?

The database information is typically the fundamental building block upon which we build our understanding of your future Delphix environment. We ask questions related to the database names, locations, platforms, and versions, as well as estimates of size, throughput, and changes.

The answers you provide to our infrastructure questions give us insight into potential challenges that may arise as we seek to integrate our application and best practices into your environment. Processes and software you use normally may not integrate optimally into our engine, and it's helpful to know this as soon as possible.

10.11.1.4 Why does Delphix use a virtual machine? Shouldn't all high-performance applications use physical infrastructure?

Using virtualization not only allows our customers to use their choice of commodity hardware, but it also allows us to focus on core features rather than hardware support.

10.11.2 Architecture best Practices for hypervisor host and VM guest - ESX FAQ

10.11.2.1 Why does Delphix recommend 8 vCPUs and 128GB of memory per 8 vCPUs?

8 vCPUs are not only a standard licensing block, but they are also key to meeting our 10Gbps single-engine throughput potential and help to sustain low latency for VDBs.

As with CPU, cache memory is required to drive peak loads on the Delphix Engine. More memory allows for more blocks to be read from the cache rather than going to less performant disks. Delphix stores cached data in a compressed format and only keeps a single copy of unique blocks in memory. These features give read performance across multiple VDBs provisioned from a single source dramatic improvements in speed, scalability, and memory utilization.

10.11.2.2 Why does Delphix request reservations for CPU and memory?

Delphix performance can be greatly impacted there is contention over CPU or RAM. Reservations allow the engine to explicitly control those resources and avoids the possibility of contention with other VMs, even when resources are overcommitted.

10.11.2.3 Why does Delphix request Hyper-Threading (HT) be disabled?

Hyper-Threading can have a positive impact on many, but not all applications. Delphix has a different execution profile that does not benefit from Hyper-Threading. Factors such as significant memory bandwidth requirements and a high level of parallelism which requires a high number of shared locks mean that HT and Delphix do not generally work well together.

10.11.2.4 Why does Delphix request 4 controllers, and why must the storage be identical between them?

To provide optimal storage performance, you must spread data equally over the maximum (4) virtual SCSI controllers. To provide consistent performance between each of the four controllers, you need to ensure storage is identical between them.

10.11.2.5 Why must virtual disks (VMDKs) be thick provisioned and eager-zeroed?

Thick provisioning and eager zero ensure performance is top-notch from the start with no hiccups from expanding virtual resources.

10.11.2.6 Why is 20% free space required?

While the ZFS file system has a lot of features leveraged by Delphix, it loses efficiency as space decreases. 20% is the minimum that must be available for best performance.

10.11.2.7 Why does Delphix request you reserve CPU and RAM for Hypervisor overhead?

Based on VMWare's resource management guide and our own experiences with high IO throughput. Note there is no specific mechanism to assign resources to the hypervisor, the only way to preserve overhead is by not allocating resources to guests.

10.11.2.8 Why does Delphix generally want VMWare HA enabled, but DRS disabled?

VMware HA (High Availability) addresses outages that occur when a physical host goes down or is completely offline, by migrating the guest(s) to another physical host and restarting them. There is no real downside, it simply brings unavailable servers back online.

VMware DRS (Distributed Resource Scheduler) is for load balancing host resources in a cluster. Because of high IO and best practices configuration for optimal performance, our engine is typically not a good candidate for relocation.

10.11.2.9 Why does Delphix request you set power management to High-Performance Mode?

This will ensure power management will never impact performance by entering into a lower power state (also known as c-state).

10.11.3 Architecture best practices for storage FAQ

10.11.3.1 Why does Delphix require 127GB of storage for the OS?

The system partition requires space to store and operate the OS, as well as application logs, upgrade and rollback images, and enough free space to store a kernel or application core dump should it be required.

10.11.3.2 Why does Delphix require our LUNS to be uniform and contain an equal quantity and capacity of VMDKs, yet thin provisioning is OK?

Because our engine leverages parallel reads, we need the storage capacity and quantity of disks they hold to be consistent. This allows the reads and writes to be evenly distributed, and minimizes the impact of potential utilization imbalances which would create a “long tail” of higher latency on a single controller, impacting the entire engine.

Data storage LUNS are generally formatted with a VMFS file system and have placed upon them a virtual disk (VMDK) which is thick provisioned and eager zeroed, so it would be a waste of time to thick provision the LUN also.

10.11.3.3 Why does Delphix require < 10ms latency (95th percentile) storage?

Storage latency is especially important in database environments. Average latency doesn’t give a complete picture of responsiveness, especially because Delphix leverages parallel reads; so inconsistent performance (e.g. good average latency but a “long tail”) can impact multiple operations. This is why Delphix has a focus on 95th percentile latency, and why we validate storage performance as the first step when a new engine is deployed.

For more information, you may find the following article helpful: The Cost of Latency[674].

10.11.3.4 Why does Delphix prefer to extend existing storage rather than simply add more while maintaining equal distribution?

While it is possible to add more storage and maintain the practice that "storage should be equal across controllers" – extending LUNS (then virtual disks) ensures that:

- We do not continue to fill disks which may be full
- Existing disks do not suffer a write performance penalty from low capacity

674 http://perspectives.mvdirona.com/2009/10/31/TheCostOfLatency.aspx

- Storage performance is consistent

10.11.4 Architecture best practices for network FAQ

10.11.4.1 Why does Delphix request 10GE Ethernet?

As a matter of physics and standards - 10 gigabit (Gb) Ethernet can sustain approximately 1 gigabyte (GB) per second of throughput. With all our best practices applied, a Delphix Engine can achieve very close to that line speed, allowing for optimal load, engine, and license utilization. Lower network speeds may be acceptable for low loads, while in some environments NIC teaming (e.g. LACP) may be required for top speeds.

10.11.4.2 Why does Delphix require < 1ms latency to TARGET servers and < 50ms to SOURCE servers?

Delphix leverages NFS and iSCSI (depending on platform) for live TARGET DB mounting over the network, so it's imperative that latency is as low as possible. Data coming from SOURCE servers is not generally as time-sensitive, so you need a minimum latency of < 50ms to ensure operational integrity.

10.11.4.3 Why does Delphix request Jumbo Frames?

Jumbo frames increase the Ethernet maximum transmission unit (MTU) from the default 1500 bytes to 9000 bytes. This has several effects such as decreasing CPU cycles by transferring fewer packets and increasing the engine throughput. You will find jumbo frames have a 10-20% real-world impact and are required (along with all other best practices) to handle peak loads of 800 - 1000MB/s on an 8 vCPU engine with a 10Gb network.

10.11.4.4 How does Delphix avoid communication impact with non-jumbo frame hosts when Jumbo Frames are enabled on the Delphix Host?

Path MTU Discovery[675] is the mechanism by which two hosts agree on the MTU leveraged for communication between them. This mechanism will ensure communication between both standard and Jumbo Frame enabled hosts works as expected.

10.11.4.5 When does Delphix recommend NIC teaming?

The Delphix Engine is capable of high throughput, but not every enterprise has sufficient network bandwidth to support it. Teaming is a less expensive way of increasing the bandwidth when compared to new hardware.

675 https://en.wikipedia.org/wiki/Path_MTU_Discovery

10.11.4.6 Why does Delphix recommend logical and physical and co-location?

The Delphix Engine leverages network connections extensively, so optimizing the latency whenever possible is very important - sometimes critical.

10.11.5 Architecture best practices for Delphix engine data protection FAQ

10.11.5.1 Why does Delphix recommend SAN snapshots or Delphix replication for backup?

There are a few possible methods for data protection of the Delphix Engine. Those methods are SAN snapshots, Delphix replication, and virtual machine snapshots (for very small engines only). Because the Delphix Engine is itself a backup of source environments, many customers simply plan to rebuild in the event of a disaster.

10.11.5.2 What is the DXToolkit, and how can it help?

The professional services team has created a Perl-based “DXToolkit” which can help export and import certain configuration data over web services. This toolkit can be leveraged to assist with would normally be a manual re-install outside of the above methods.

For further detail around data protection, please speak with your Delphix contact.

10.11.5.3 Can I use a VMware-based backup solution such as VEEAM to backup my Delphix Engine?

Yes, VMWare backup solutions are useful for backing up guest VMs. However, Delphix suggests that you only use this approach for Delphix Engines which have a smaller storage footprint (perhaps < 2 TB) and are less active.

Running this type of backup puts a load on the environment, which might adversely impact Delphix VM performance.

10.11.5.4 Can I use a VMware snapshot for backing up Delphix for a small window – for example, during an engine upgrade?

Yes. However, even though snapshots are instantaneous, they track changes separately from the base disks and can grow to consume as much space as the original.

Upgrades, in particular, can change substantial amounts of data.

If you lose physical disks, snapshots are useless because it needs them to make up the current state of VM.

A Delphix Engine is often allocated multiple terabytes of storage and is often very busy due to load aggregation from virtual databases on multiple target servers, so this approach may be challenging.

Snapshots cannot detect storage corruption.

10.11.5.5 Can I use a Storage snapshot solution to protect Delphix against Storage and Delphix corruption?

Yes. However, please note that the caveats which apply to VMWare snapshots will also apply here.

A specific concern related to storage layer snapshots is that you must create a consistency group that contains both the OS and Data disks.

10.11.5.6 Can I use RMAN to backup my VDBs just like a Physical database to provide extra protection?

You can backup Delphix VDBs using Oracle RMAN tools, but the recovery database would first require re-hydration of that VDB, which might take up equivalent production storage space.

Furthermore, that re-hydrated database needs to be brought into the Delphix framework as a dSource, after which you can provision a VDB to complete recovery. The whole process might take hours or days to recover.

The best approach is to use the VDB Snapshot capability to backup VDB frequently and then leverage Delphix Replication capability to protect underlying Delphix storage, which holds that VDB snapshot.

10.12 Architecture checklist

10.12.1 Overview

The Architecture Checklist is an overview combining the more important Architectural best practices into a single list. For more detail regarding some of the reasons for our best practices, review the Architecture Checklist FAQ.

10.12.2 Architecture best practices for hypervisor host and VM guest - ESX

10.12.2.1 Hypervisor

1. ESXi 6.x or 7.0 is recommended. ESXi 5.5 or earlier is no longer supported.
2. HyperThreading (HT) for Intel®-based servers (no HT on AMD CPUs).
    a. Disable HT in BIOS, on the ESXi Host, **and** disable HT Sharing[676] on the Delphix VM for consistency. This is our best practice, disable at all levels. Any other combination may result in non-deterministic performance.
    b. When HT cannot be turned off for both the Host and Delphix VM, it should be turned on at all levels, not run in a "mixed-mode".

676 https://stromasys.atlassian.net/wiki/display/KBP/VMware+ESX+-+Disabling+hyper+threading+for+a+specific+virtual+machine

        i. HT disablement at a guest level only can result in non-deterministic performance.
        ii. A dedicated ESXi host, cluster or DRA is recommended where consistent VDB performance is paramount.
    c. VM migration to a new host (e.g. VMware HA or vMotion) can create mismatched HT settings.

3. ESXi overhead (resources required for hypervisor cannot be reserved, they must be left unallocated).
    a. Memory Overhead - 10% of available RAM must not be allocated to guest VMs.
        - *Example: 256GB RAM, allocate 230GB to Delphix VM, leave 26GB for ESX*
    b. CPU Overhead - At least 2 cores (ideally 4) must not be allocated to guest VMs.
        - *Example: If 16 physical cores are available, allocate 12 to the virtual machines, leaving 4 for the hypervisor*
        - Why 4? Certain hypervisor functions require precedence over any virtualized system. If a hypervisor needs more CPU than the amount currently available, it can de-schedule all other virtual processes to ensure adequate CPU resources for the hypervisor. Ensuring the hypervisor will not have to de-schedule any running virtual processes (worlds) by setting aside and not over-subscribing CPUs for virtual functions will leave them available for hypervisor use.
    c. Even if the Delphix VM is the only VM on a host, the hypervisor is still active and essential; and still needs resources.

4. BIOS Power Management should be set to High Performance where ESXi controls power management.
    a. Can be impacted by VMware KB 1018206[677] - poor VM application performance caused by power management settings.
    b. Ensure that all BIOS managed C-States other than C0 are disabled if power management is hardware controlled.
    c. Ensure that all ACPI sleep states above S0 are disabled in the BIOS.
    d. Examples for popular server lines from Cisco, HP, Dell below. *Specific models will vary, use the appropriate spec sheet.*
        i. UCS: disable the Processor Power States, disable Power Technology, set Energy Performance to "Performance"
        ii. HP Proliant: set HP Power Regulator to HP "Static High Performance" mode
        iii. Dell: set BIOS System Profile to "Performance Optimized" mode

5. VMware HA can be enabled; VMware DRS is generally disabled.

6. Blade/Rack Server Firmware and ESXi Drivers should be updated to latest versions.

7. For Intel®-based servers with E5-2600 v2 processors.

8. Two typical server configurations:
    a. Blade Farm
    b. Rack Server Hyper-Converged configurations are possible for high performance.

677 http://kb.vmware.com/kb/1018206

10.12.2.2 Virtual Machine Guest

1. For VM machine settings, see Virtual Machine Requirements for VMware Platform. (see page 604)
2. VMWare Guest Specifications:
   a. Minimum: 8 vCPU x 64 GB Small: 8 vCPU x 128GB Medium: 16 vCPU x 256 GB Large: 24 vCPU x 512 GB
   b. Reserve 100% of RAM and CPU:
      - If the ESX host is dedicated to Delphix, CPU and RAM reservations are advised but not necessary, however, swap space will be required on the hypervisor to compensate for the lack of reserved RAM.
   c. Hyperthreading[678] - See the ESX host section at the top. Disable HT Sharing on VM, disable HT on ESX Host.
3. Assign single-core sockets for vCPUs in all cases. If there is a compelling reason to use multi-core CPUs, reference the following article from VMware which describes matching virtual multi-core sockets to the hardware ESX is running on. VMware Article on CoresPerSocket[679] Example: *ESXi Host has 2 socket x 18 core Intel Xeon, Delphix Engine wants 16 vCPU.* Configure Delphix VM with 2 Virtual Sockets, 8 Cores Per Socket to utilize hardware architecture.
4. Avoid placing other extremely active VMs on the same ESX host.
5. Monitoring - vSphere Threshold Alerts for CPU, Network, Memory, Capacity.
6. To set the number of vCPUs per virtual machine via the vSphere client, please see "*Virtual CPU Configuration"*in the Administration guide:

   ESXi 6.0[680]
   a. Delphix VM CPU Utilization[681] - Delphix KB article on what makes Delphix VMs similar to other resource-intensive applications
   b. Exchange on VMware[682]Best Practices - VMworld 2013 session
   c. ESXTOP Reference[683], Blog[684]
7. Ensure that the latest available VMware drivers and firmware versions are installed for HBAs, NICs and any other hardware components configured on the Delphix virtual machine. This is a critical step that can have a massive impact on the performance and robustness of our solution.

---

678 https://stromasys.atlassian.net/wiki/display/KBP/VMware+ESX+-+Disabling+hyper+threading+for+a+specific+virtual+machine
679 https://blogs.vmware.com/vsphere/2013/10/does-corespersocket-affect-performance.html
680 https://docs.vmware.com/en/VMware-vSphere/index.html
681 https://support.delphix.com/Delphix_Virtualization_Engine/Platforms/KBA1019_VMWare_and_Delphix_CPU_Utilization_Discrepancy_Explained
682 http://www.derekseaman.com/2013/08/vmword-2013-virtualize-microsoft-exchange-server.html
683 https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.monitoring.doc/GUID-A31249BF-B5DC-455B-AFC7-7D0BBD6E37B6.html
684 http://www.yellow-bricks.com/esxtop/

10.12.3 Architecture best practices for network configuration

1. For more information about network configuration refer to Network Performance Configuration Options. (see page 719)
2. Delphix Engine <===> Target Host:*Implement standard requirements for optimal NFS/ISCSI performance:*
    a. Optimal physical network topology:
        - Low latency: < 1ms for 8K packets.
        - Network adjacency: minimize network hops, co-locate in the same blade enclosure, co-locate on the same physical host.
        - Eliminate all Layer 3+ devices - firewalls, IDS, packet filters (Deep Packet Inspection - DPI).
        - Multiple switches can add latency and fragmentation and reordering issues will add significant latency.
    b. Optimal throughput:
        - 10GbE physical uplinks or higher.
        - Jumbo frames (typically MTU 9000) improves network efficiency for 8K packets: lower CPU, lower latency, greater throughput.
        - All devices end-to-end *must* be configured for the larger frame size including switches, routers, fabric interconnects, hypervisors and servers.
        - Traffic between two VMs on the same host is limited to ~16Gbps when leveraging built-in virtual networking.
    c. Optimal logical flow:
        - Disable QoS throttles limiting network utilization below line rate (e.g. HP Virtual Connect FlexFabric).
        - Consider a dedicated VLAN (with jumbo frames) for NFS/iSCSI traffic.
            - VMware KB-Configuring iSCSI port binding with multiple NICs in one vSwitch for VMware ESXi 6.0.x (2045040)[685]
            - VMware KB-Considerations for using software iSCSI port binding in ESX/ESXi (2038869)[686]
    d. NIC Teaming (at ESX layer) of multiple physical uplinks can provide additional throughput for higher workloads.
        i. Examples: 4 x 1Gb NICs support up to 400 MBPS IO, 2 x 10Gb NICs support up to 2 GBPS IO.

685 https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2045040
686 https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2038869

ii. VMware KB-1004088[687] has NIC teaming recommendations, including `route-based-on-IP-hash` policy.
iii. VMware KB-1001938[688] has host requirements for physical link aggregation (LACP, EtherChannel).
iv. VMware KB-1007371[689], popular blog post[690] details problems with NIC selection using `dest-IP` hash.

e. Fragmentation and dropped packets can result in excessive retransmissions of data, reducing throughput.
   i. On AIX, LSO (Large Send Offload) and LRO (Large Receive Offload) network features have caused many problems. The virtualization engine employs LSO only, but no LRO, AIX employs both. These features are enabled by default. The best practice is to disable LSO on the VE when the target is AIX.
      - LSO enabled on the Virtualization engine can result in dramatically limited or blocked transmit throughput (and increase re-transmission traffic).
      - When wishing to disable all, LSO should be disabled on the Delphix Virtualization Engine, while both LRO and LSO should be disabled on guest VM of AIX Target. The specific commands to disable LSO and LRO on AIX are not captured here and appear to be context/version-specific.

f. **Jumbo frames check via ping**

```
Delphix Engine
$ ping -D -s [Target_IP] 8000
"ICMP Fragmentation needed and DF set from gateway" indicates MTU < 8028

Linux
$ ping -M do -s 8000 [Delphix_Engine_IP]
"Frag needed and DF set (mtu = xxxx)" indicates MTU < 8028

MacOS
ping -D -s 8000 [Delphix_Engine_IP]

Note: "sudo sysctl -w net.inet.raw.maxdgram=16384" will increase the max
ICMP datagram size on Mac, allowing you to use -s 9000 on MacO

Windows
ping -f -l 8000 [Delphix_Engine_IP]

http://www.mylesgray.com/hardware/test-jumbo-frames-working/
```

g. Measure Network Bandwidth and Latency:

687 https://kb.vmware.com/s/article/1004088
688 https://kb.vmware.com/s/article/1001938
689 https://kb.vmware.com/s/article/1007371?sliceId=2&dialogID=137604103&docTypeID=DT_KB_1_1&stateId=0+0+137602450
690 https://blogs.vmware.com/kb/2013/03/troubleshooting-network-teaming-problems-with-ip-hash.html

        i. Latency in both directions should be < 1ms for an 8KB payload.
        ii. Network Hops should be minimized: `traceroute (Unix/Linux) / tracert (windows).`
        iii. Throughput in both directions: 50-100 MB/s on 1 GbE, 500-1000 MB/s on 10 GbE physical link.
        iv. Always use the CLI when possible.
    h. NIC should use Auto-negotiate on Ethernet with a minimum of 1000Mbps.
        - Hard setting speed/duplex will limit network throughput below the line rate.
3. **Delphix <===> Staging Server**(SQL Server, Sybase):

   Measure latency, bandwidth for transaction log restore performance
4. **Source <===> Delphix:** Measure latency, bandwidth to explain snapsync performance
5. **ESX host <===> ESX host**(ESX Cluster):
    a. This is because the entire memory footprint of the Delphix VM (more precisely, the entire address space of the ESX processes that comprise the VM) must be copied to the receiving ESX host, along with all changes to that address space as they happen. Our ZFS cache comprises the bulk of that memory and changes as I/O occurs, and that rate of change is governed by the network bandwidth available to the Delphix Engine.
    b. Attempting to live vMotion a DE with a 10Gb NIC over a 1Gb vMotion network is thus likely to fail.

10.12.4 Architecture best practices for storage

1. Configuration:
    a. Virtual Disks (VMDK) with spinning or tiered media must be thick provisioned + lazy zeroed.
        - For storage which is 100% SSD/EFD/Flash-based, continue to thick provision, however, eager zero is not necessary.
    b. VMDKs may be homed on VMFS or RDM storage, however, VMFS is much more common and generally preferred.
    c. Regardless of VMFS/RDM selection, physical LUNs must have uniform characteristics (RAID, spindle count, tier) and should be thin provisioned to save time.
    d. Storage allocated must be identical between each vSCSI controller.
    e. The supported maximum of four virtual SCSI controllers (PVSCSI (default) or LSI Logic Parallel) must be enabled. A mix of different types of SCSI controllers is not supported within the engine.
    f. Virtual Disks must be evenly distributed across the 4 virtual SCSI controllers. For example, 8 virtual disks should be configured as 2 disks per controller: SCSI (0:0), SCSI (0:1), SCSI (1:0), SCSI (1:1), SCSI (2:0), SCSI (2:1), SCSI (3:0), SCSI (3:1).
        - You don't need to account for the OS in the even distribution of disks across controllers, just pick one; the OS doesn't place a substantial load on the controller.
    g. To provision VMDK disks over 16TB in size, the vSphere web client must be used, the win32 client will return an error.

   h. As of 5.1.3, we require 127GB for the system disk. older versions defaulted to 150GB.
      - Due to our unified OVA for masking, the 127GB requirement also applies to Masking Engines
   i. VMDK for the Delphix Engine OS is often stored on the same VMFS volume as the Delphix VM definition file (aka VMX). In that case, the VMFS volume must have sufficient space to hold the Delphix VMX Configuration, the VDMK for the system disk, a swap/paging area if the memory reservation was not enabled for the Delphix Engine (or to suspend the system), and any related VMware logging.
   j. Set ESX Storage Multipathing IO optimization (KB 2072070[691]).
      i. Set multipathing policy to round-robin.
      ii. Set path switching IO operation limit to 1 (default 1000).
   k. Verify that the queue depth setting on the virtual disks used for VMware is appropriate based on the minimum of the HBA type or the underlying storage's combined queue depth for all hosts attached to the same controller
      - See this VMware KB article[692] for how to check/set the queue depth
2. Testing:
   a. Run Storage Performance Tool on the raw storage **before any engine configuration**. This is a one-time opportunity for each engine upon installation
   b. Required maximum storage latency is < 2ms for writes and < 10ms (95th percentile) for small random reads. Minimum passing grades: 4KB/8KB reads (B-), 1MB reads (A), 1KB/128KB writes (A).
   c. If working with the Delphix Professional Services team, we would expect to run additional baseline performance measurements via composite tools and scripts.
      - e.g. "Sanity Check" (Oracle) or "DiskSpd" (MSSQL)
3. Detail Discussion:
   a. Before beginning any discussion on storage performance, or at the beginning of the installation, collecting the following specs from your storage administrator will assist in understanding
   b. Vendor, Model (For example: EMC VMAX 40k, HP 3PAR StoreServ 7000)
   c. IO latency SLO (For example 5ms 99.999%)
   d. IOPS/GB SLO (For example: 0.68 IOPS/GB for EMC Gold-1)
   e. Cache type and size (For example FAST cache 768GB)
   f. Tier, #Pools; if auto – tiering; relocation schedule (For example: Gold/Silver/Auto/3 pools/etc)
   g. Pool detail: (#) drives, RPM, Type (For example: Pool1: (20) EFD, (30) 15k SAS, Pool 2: (40) 10k SATA)
   h. Connection (For example: 16Gb Fibre Channel, 10Gb iSCSI, 20Gb N

10.12.5 Architecture best practices for Delphix engine data protection

1. For protection against physical host failure - leverage VMware HA.

691 http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2072070
692 https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1267

2. For protection against storage failure - leverage Delphix replication.
3. For protection against administrative error - leverage storage snapshots.
4. For protection against site failure - leverage Delphix replication and/or Delphix Live Archive.
5. Infrastructure Backup of the Delphix VM: Must take a consistent group snapshot of all Delphix VM storage (system disk, VM configuration, database VMDK/RDMs) RTO, RPO are inferior compared to Delphix Replication. The granularity of restore is at the VM level: ***all-or-nothing.***
    a. Virtual Machine Backup: Create VM snapshot, backup (proxy server), remove the snapshot. Products use VMware APIs: NBU for VMware, TSM, Networker Limited to < 2TB because of time to backup, impact on running VM
    b. Storage Array Backup: Take a consistent storage snapshot, replicate to tape/VTL media server, remove the snapshot. Products include Hitachi Shadow Copy, EMC SnapCopy, HP Business/Snap Copy.

10.12.6 Architecture best practices for Masking

1. As of release 5.0, the virtualization and masking functions are combined into a single OVA and require additional consideration for installation and configuration. Additionally, support for remote Continuous Compliance Engine calls has been implemented and is supported in 5.0.4 and above.
2. Continuous Compliance Engines should continue to be deployed to hosts dedicated to that function.
    - Possible exceptions to this would be when any virtualization needed is extremely low and unlikely to be heavily impacted by masking requirements.
3. The Masking Engine is disabled by default.
    - The Delphix Engine will continue to remain running (but unused) on a Masking-only VM.
4. The standard configuration for a dedicated Continuous Compliance Engine:
    a. 8 vCPUs
    b. 16GB RAM minimum, 32 GB RAM or more recommended.
    c. 300 GB storage for the OS/system root disk is required for the OS (5.1.4 and greater).
    d. 50 GB storage for the data disk must be added during the initial configuration via the Engine Setup wizard. (the engine will not complete its setup without a separate data disk).
    e. If a bulk operation is used, allocate extra space equivalent to the size of all datasets (tables) that will be masked concurrently.
        i. As a rule of thumb: *Disk Space Required for Bulk = ((Total Database Size * .66) * .10) where Raw data = Total Database Size * .66*
        ii. 10% change is an estimate based on our experience for data we mask. Often it is lower but there are exceptions such as masking a data warehouse with a large fact table and a bunch of much smaller tables.
        iii. The VMDK for the engine OS is often stored on the same VMFS volume as the VM definition file (aka VMX). In that case, the VMFS volume must have sufficient space to hold the VMX Configuration, the VDMK for the system disk, and any VMWare logging.
        iv. Additional VMFS space for swap/paging is required if RAM reservations are not enabled. (The VM will not start if reservations are lacking and disk space is not available for swap)

CPU utilization

1. One vCPU per concurrent masking job is considered a best practice.
2. Dependent on the algorithms used: Some are calculations such as ones using AES encryption and others are lookups and tend to do more I/O.

Memory utilization

1. The Continuous Compliance Engine uses its memory to cache data. More memory will provide better performance. 1GB per masking job is considered a best practice.
2. Dependent on memory settings in the Masking Engine and JVMs. An increase in parallel workloads will require more memory. Data is either cached directly or using Kettle so the larger the lookups for algorithms the more memory required. This is the first thing to look at for performance issues.

Network and I/O

Continuous Compliance leverages the Target DB server and VDB for most of the workload. This means the masking engine can be I/O bound waiting for the DB server. As long as the masking engine can read the data faster than it can process it this is not an issue. Slow networks with numerous hops between the DB server and the Masking server can cause performance problems. Co-locating the masking server with the DB server is recommended in these cases.

Masking VDB tuning

1. Always start with the tuning recommendations for Target servers and VDBs first. If the VDB is not performing well, the performance of masking will suffer.
2. For Oracle, it is critical to select no archivelog mode and tune online redo log size at provision time.
3. For SQL Server, the VDB should be in SIMPLE recovery with an appropriate log file and TempDB sizes.

Backup of a continuous compliance engine

1. Virtual machine backups are recommended for versions of software in which masking runs in its own VM – in other words, the masking VM is separate from the VM(s) where virtualization takes place.
    a. If an engine is supporting both masking and virtualization, review data protection best practices.
    b. Although XML exports of inventories and environments do exist, they are incomplete. Do not rely on them.
2. In-Place (not On-the-Fly) Masking is the primary use case.

The following recommendations apply to the Continuous Compliance Engine versions 5.0.2 and earlier:

1. Jobs vs. Streams:
    a. If there are multiple tables to be masked concurrently, use multiple, separate jobs – one per table.
    b. Avoid multiple streams due to internal limitations.

> ⚠ 1. The default setting for streams is 20; set it to 1. This will force serialization (one table at a time) if the job contains multiple tables.

2. Use one update thread per job; this avoids block collisions/contention during the UPDATE phas.

> ⚠ The default setting is 4; set it to 1.

a. Identify ALL indexes, constraints, and triggers on columns being masked (and only on columns being masked).

b. Evaluate whether it is better to drop/mask/recreate for indexes, or disable/mask/re-enable for triggers and constraints – as compared to leaving in-place during masking. The best choice depends on the situation

3. For Oracle VDBs, use ROWID for SQL UPDATE of masked row value(s).

1. Edit the ruleset, select Edit All Logical Keys, enter ROWID as the logical key value; see Managing Rule Sets[693]. When a single, large, non-partitioned table must be masked by concurrent jobs (each masking a subset of the table) to shorten masking elapsed time, segregate jobs by database block/page to avoid contention and locking conflicts. In other words, each job masks a unique set of blocks; each block is masked exclusively by one job.
2. If a single table is being masked by multiple, concurrent Jobs, and Indexes/constraints/triggers must be dropped/recreated or disabled/enabled, these must be performed OUTSIDE of masking Jobs.
3. Pre-masking and post-masking steps must be created manually.
4. Scheduling of pre-script and post-script jobs must be devised. Plan to scheduled/execute externally.

10.12.7 Architecture best practices for source DB and OS settings

10.12.7.1 Oracle

1. ARCHIVELOG must be enabled: `select log_mode from v$database.`
2. FORCE LOGGING should be enabled to ensure VDBs are not missing data. When NOLOGGING redo is applied during provision, the resulting VDB will be missing changes. Tables with NOLOGGING changes will throw corruption errors when scanned.
3. Block Change Tracking should be enabled to minimize snapsync time.
4. Consult the documentation for Oracle Standby sources.

693 https://masking.delphix.com/docs/latest/managing-rule-sets

5. If the database is encrypted with Oracle TDE (Transparent Data Encryption) plan your Delphix storage requirements with the expectation of minimal compression. Customer Observation: space usage for a TDE dSource copy was 92% (2.44 TB) of the source database size (2.67 TB). A typical Oracle dSource copy for a non-TDE database consumes 40% of the source database size.

10.12.7.2 SQL server

1. FULL vs SIMPLE recovery mode trade-off.
2. The maximum size of an MSSQL database that can be linked is 256TB for Windows versions greater than 2003. The limit is 2TB for Windows 2003. (See Linking a SQL Server dSource (see page 2139))

10.12.8 Architecture best practices for target DB and OS settings

10.12.8.1 Target database application settings

1. Oracle:
    a. Provision with 3 x 5GB online redo logs (minimum) to avoid pause when transaction logs wraparound.
    b. A provision in NOARCHIVELOG mode to reduce transaction log IO. Masking, Test, QA VDBs rarely need point-in-time rewind
    c. Always check initialization parameters inherited from a parent, remove any expensive or irrelevant parameters.
        i. `DB_CACHE_SIZE, SGA_TARGET`: set based on the target system being compared to.
        ii. `FILESYSTEMIO_OPTIONS` to `SETALL`. Any other setting inherited from the source is probably wrong.
        iii. `DB_BLOCK_CHECKSUM, DB_BLOCK_CHECKING, DB_LOST_WRITE_PROTECT, DB_ULTRA_SAFE`: set to default values to minimize the impact.
        iv. `PARALLEL_DEGREE_POLICY` to `AUTO`, `PARALLEL_MAX_SERVERS` default, `PARALLEL_EXECUTION_MESSAGE_SIZE` to 32768 (maximum): improve PQ performance.
        v. `FAST_START_MTTR_TARGET:` drives steady write activity. Set based on the target system being compared to.
        vi. Consider non-durable commits for Masking, Test, QA, UAT: set `COMMIT_WAIT = NOWAIT, COMMIT_LOGGING = BATCH`
    d. Use Oracle Direct NFS (dNFS) for 11.2.0.4+ (**unstable** on older releases):
        i. Recommended documentation:
            - Configuration Examples and Troubleshooting blog[694] from Helmut Hutzler

694 http://www.hhutzler.de/blog/dnfs-direct-nfs-setup-for-11-2-0-4-3-node-rac-cluster-on-oel-6-4/

        - Sample oranfstab to leverage multiple network paths for Delphix VDB
    ii. Set DNFS_BATCH_SIZE[695] = 128 (default is 4096). This is a good starting point and sufficient for most workloads.
    iii. Tune TCP stack: set `tcp_adv_win_scale` = 2 due to workaround hard-coded Oracle dNFS TCP buffer size.
    iv. Check Alert Log, V$DNFS_SERVERS, V$DNFS_FILES, V$DNFS_STATS to verify proper working (sample here[696]).
e. Create AWR snapshots around a reference customer workload, generate an AWR report.
    i. AWR snap before/after workload: `SQL> exec DBMS_WORKLOAD_REPOSITORY.CREATE_SNAPSHOT();`
    ii. AWR report between the snaps: `SQL> @?/rdbms/admin/awrrpte`
f. Generate ASH report to diagnose bottlenecks while a workload is running.
    - `SQL> @?/rdbms/admin/ashrpt`
g. Run synthetic benchmark `sc-workload.`
h. Where `db file scattered read` (multiblock cached read) latency is high consult this Support KB: How to Mitigate Multi-Block Read Performance on Oracle 10g[697]
i. Improve distributed query performance by modifying dblinks to use local IPs instead of SCAN IPs.
j. NFS recommended mount options for Oracle RAC/SI: Oracle Support Note 359515.1[698].

10.12.8.2 Target Host OS Settings

1. Existing documentation on Target OS practices: Target Host OS and Database Configuration Options (see page 907)
2. HP-UX 11.31+
    - Async NFS Direct I/O[699]: HP-UX requires Oracle `disk_asynch_io` turned off for filesystems
3. IBM AIX:
    a. Consult IBM documentation on AIX TCP Tuning
    b. AIX TCP Tuning Prezo[700]
4. Windows:
    a. Anti-virus programs can impact both performance and operation. Delphix recommends anti-virus scanning exclude folders where Delphix files are maintained, in addition to the normal exclusions put in place for MSSQL operation.

---

695 http://docs.oracle.com/cd/E11882_01/server.112/e40402/initparams083.htm#REFRN10334
696 https://sort.veritas.com/public/documents/sfha/6.0/linux/productguides/html/virtualstore_admin/ch03s05.htm
697 https://support.delphix.com/Delphix_Virtualization_Engine/Oracle/How_to_Mitigate_Multi-Block_Read_Performance_on_Oracle_10g
698 https://support.oracle.com/rs?type=doc&id=359515.1
699 https://h20564.www2.hpe.com/hpsc/doc/public/display?docId=emr_na-c02444831&sp4ts.oid=3553037&lang=en-us&cc=us
700 http://www.circle4.com/movies/budapest/baixperfpt3.pdf

b. Delphix Connector (aka DX Connector):
   i. Plan 3-5GB for the Delphix Connector installation.
   ii. Windows does not yet have ssh, so Delphix developed the "DX Connector for Windows target host communication.
   iii. The connector must be installed on all Target Windows hosts.
   iv. The connector supports two modes – v1 and v2 both use the same application binaries.
   v. The connector v1 process is used to bootstrap the v2 process on a target. This opens a DSP session back to the Delphix Engine (The same thing is done via SSH on U*nix Targets)
   vi. v2 mode is required to enable SQL hooks
   vii. The connector can always be downloaded from a local Delphix Engine at: http://<delphix_engine>/connector/DelphixConnectorInstaller.msi.
   viii. The connector is backward compatible, so it is not always necessary to upgrade it during a Delphix upgrade.
c. iSCSI connections:
   i. Read the following for general awareness of iSCSI limits
   ii. In addition to the hard limits on iSCSI connections, consideration must be given to the RAM, CPU, and Network to provide sufficient resources for the load on any Target or Staging host.
   iii. To increase the iSCSI timeout on both Target and Staging hosts.
   iv. In certain circumstances, it's possible that iSCSI startup will not complete before the SQL Service attempts to start a database. In such circumstances, it can be helpful to ensure the SQL service depends on the iSCSI service.
      - Example: c:\> sc config "MSSQLServer" depend="Microsoft iSCSI Initiator Service"
   v. **Note** that any changes to iSCSI are system-wide and could potentially impact other applications also leveraging that feature.
d. Enable Receive Side Scaling (RSS) (see page 2130) on each network interface that Delphix will be connecting to.

10.12.9 Architecture best practices for Staging Targets

This host is called a "staging target" because it has much in common with other targets, such as the remote storage mount to the Delphix Virtualization Engine. Because it leverages remote storage over the network, the staging target only needs enough disk capacity for the OS, database application, and any relevant logs or tools. A staging target is a requirement on all platforms that Delphix supports except Oracle, but you can also use it for Oracle to replay logs leveraging validated sync. (see page 2133)

10.12.9.1 Memory and CPU

1. 32 GB RAM minimum
2. 4 vCPU minimum

10.12.9.2 General guidance for staging servers (Multi-platform)

1. Delphix recommends dedicated Staging servers for role/architecture separation. However, any Target server can be used as Staging.
2. In cases where the same server is used as both Staging and Target, we strongly recommend a dedicated instance/install for staging to avoid confusion.
3. Delphix recommends at least one Staging server per Delphix Engine to avoid the possibility of a single point of failure across multiple engines.
4. If a staging server is shared among multiple Delphix Engines, please ensure that a dedicated SQL Server Instance is created for each Delphix Engine.
5. Configuration / performance factors:
    a. Transaction log generation rate.
    b. Number of VDBs.

> Precise guidance on these items has not yet been defined. In general, if there is a heavy log generation rate and few VDBs, the ideal recommendation is to have at least 1 Staging Target per Delphix Engine.

10.12.9.3 Disk / Local storage

1. The only local storage needed is for the OS and application with default databases.
2. Storage for a staging database is provided from the Delphix Engine, which is mounted over the network similar to any Target host (NFS/iSCSI).
3. If the customer has a standard DB server build, their standard storage sizing is probably fine.
4. If a recommendation is still needed, suggest 30GB for OS and application and any tools needed.

10.12.9.4 Network requirements

1. The Staging Target engages in network data transfers between Staging and the Source backup shared location as well as between Staging and the Delphix Engine.
2. The Staging Target is also a Target server, and as such should have < 1ms latency to the Delphix Engine (and low latency to the Source backup, when possible).
3. If the change rate on the Source database(s) is > 1 Gb/sec, the recommended network bandwidth to support network transfers is 10 Gb/sec.

4. In cases where only 1 Gb/sec network bandwidth is available, segregation of each network is recommended to reduce network contention.
5. Ensure that the virtual NIC is using the standard vmxnet3 adapter and not Intel for VMWare based clients. Logical IO errors have been reported while using Intel instead of vmxnet3 adapter.

10.12.9.5 Windows and MSSQL Specific

1. An MSSQL Server Instance used for Staging should not be clustered.
2. Staging should not be hosted on Windows 2003 - extended support ended July 14, 2015. It is also the first Windows version with iSCSI support and is not ideal.
3. The SQL Server Instances hosted on the Staging Target should have a Maximum Memory set. Also ensure that at all times, at least 10% of total memory is available for OS operations.
4. Only system databases (Master/MSDB/Temp/MSDB) are kept on local storage. All other data is read/written to the Delphix Engine.
5. Windows iSCSI configuration and limits for v2p, target, and staging hosts.
6. Ensure that Receive Side Scaling (RSS) (see page 2130) is enabled on every network interface that Delphix will be connecting to.

10.13 Best practices for hypervisor host and VM guest

10.13.1 Hypervisor

1. ESXi 6.x[701] or 7.0 is recommended. ESXi 5.5 or earlier is no longer supported.
2. HyperThreading (HT) for Intel®-based servers (no HT on AMD CPUs).
   a. Disable HT in BIOS, on the ESXi Host, **and** disable HT sharing[702] on the Delphix VM for consistency. This is our best practice, disable at all levels.
      Any other combination may result in a non-deterministic performance.
   b. When HT cannot be turned off for both the Host and Delphix VM, it should be turned on at all levels, not run in a "mixed-mode".
      i. HT disablement at a guest level only can result in non-deterministic performance.
      ii. A dedicated ESXi host, cluster or DRA is recommended where consistent VDB performance is paramount.
   c. VM migration to a new host (e.g. VMware HA or vMotion) can create mismatched HT settings.
3. ESXi overhead (resources required for hypervisor cannot be reserved, they must be left unallocated).

701 https://www.vmware.com/files/pdf/techpaper/VMware-PerfBest-Practices-vSphere6-0.pdf
702 https://stromasys.atlassian.net/wiki/display/KBP/VMware+ESX+-+Disabling+hyper+threading+for+a+specific+virtual+machine

   a. Memory Overhead - 10% of available RAM must not be allocated to guest VMs.
      - *Example: 256GB RAM, allocate 230GB to Delphix VM, leave 26GB for ESX*
   b. CPU Overhead - At least 2 cores (ideally 4) must not be allocated to guest VMs.
      - *Example: If 16 physical cores are available, allocate 12 to the virtual machines, leaving 4 for the hypervisor*
      - Why 4? Certain hypervisor functions require precedence over any virtualized system. If a hypervisor needs more CPU than the amount currently available, it can de-schedule all other virtual processes to ensure adequate CPU resources for the hypervisor. Ensuring the hypervisor will not have to de-schedule any running virtual processes (worlds) by setting aside and not over-subscribing CPUs for virtual functions will leave them available for hypervisor use.
   c. Even if the Delphix VM is the only VM on a host, the hypervisor is still active and essential; and still needs resources.
4. BIOS Power Management should be set to High Performance where ESXi controls power management.
   a. It can be impacted by VMware KB 1018206[703] - poor VM application performance caused by power management settings.
   b. Ensure that all BIOS managed C-States other than C0 are disabled if power management is hardware controlled.
   c. Ensure that all ACPI sleep states above S0 are disabled in the BIOS.
   d. Examples for popular server lines from Cisco, HP, Dell below. *Specific models will vary, use the appropriate spec sheet.*
      i. UCS: disable the Processor Power States, disable Power Technology, set Energy Performance to "Performance"
      ii. HP Proliant: set HP Power Regulator to HP "Static High Performance" mode
      iii. Dell: set BIOS System Profile to "Performance Optimized" mode
5. VMware HA can be enabled; VMware DRS is generally disabled.
6. Blade/Rack Server Firmware and ESXi Drivers should be updated to the latest versions.
7. For Intel®-based servers with E5-2600 v2 processors.
8. Two typical server configurations:
   a. Blade Farm
   b. Rack Server
   Hyper-Converged configurations are possible for high performance.

10.13.2 Virtual machine guest

1. For VM machine settings, see Virtual machine requirements for VMware platform (see page 604)
2. VMWare Guest Specifications:

703 https://kb.vmware.com/kb/1018206

   a. Minimum: 8 vCPU x 64 GB
      Small: 8 vCPU x 128GB
      Medium: 16 vCPU x 256 GB
      Large: 24 vCPU x 512 GB
   b. Reserve 100% of RAM and CPU:
      - If the ESX host is dedicated to Delphix, CPU and RAM reservations are advised but not necessary, however, swap space will be required on the hypervisor to compensate for the lack of reserved RAM.
   c. Hyperthreading[704] - See the ESX host section at the top. Disable HT Sharing on VM, disable HT on ESX Host.

3. Assign single-core sockets for vCPUs in all cases. If there is a compelling reason to use multi-core vCPUs, reference the following article from VMware which describes matching virtual multi-core sockets to the hardware ESX is running on.
   VMware article on CoresPerSocket[705]
   Example: *ESXi Host has 2 socket x 18 core Intel Xeon, Delphix Engine wants 16 vCPU.*
   Configure Delphix VM with 2 Virtual Sockets, 8 Cores Per Socket to utilize hardware architecture.
4. Avoid placing other extremely active VMs on the same ESX host.
5. Monitoring - vSphere Threshold Alerts for CPU, Network, Memory, Capacity.
6. To set the number of vCPUs per virtual machine via the vSphere client, please see "*Virtual CPU Configuration"* in the Administration guide:

   ESXi 6.0[706]
   a. Delphix VM CPU utilization[707] - Delphix KB article on what makes Delphix VMs similar to other resource-intensive applications
   b. Exchange on VMware best practices[708] - VMworld 2013 session
   c. ESXTOP reference[709], blog[710]
7. Ensure that the latest available VMware drivers and firmware versions are installed for HBAs, NICs and any other hardware components configured on the Delphix virtual machine. This is a critical step that can have a massive impact on the performance and robustness of our solution.

---

704 https://stromasys.atlassian.net/wiki/display/KBP/VMware+ESX+-+Disabling+hyper+threading+for+a+specific+virtual+machine
705 https://blogs.vmware.com/vsphere/2013/10/does-corespersocket-affect-performance.html
706 https://docs.vmware.com/en/VMware-vSphere/index.html
707 https://support.delphix.com/Delphix_Virtualization_Engine/Platforms/KBA1019_VMWare_and_Delphix_CPU_Utilization_Discrepancy_Explained
708 https://www.derekseaman.com/2013/08/vmword-2013-virtualize-microsoft-exchange-server.html
709 https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.monitoring.doc/GUID-A31249BF-B5DC-455B-AFC7-7D0BBD6E37B6.html
710 https://www.yellow-bricks.com/esxtop/

10.13.3 Best practices hypervisor host and VM guest ESX FAQ

10.13.3.1 Frequently asked questions

10.13.3.1.1 Why does Delphix require a minimum of 8 vCPUs and recommend 128 GB per 8 vCPUs?

8 vCPUs are not only a standard licensing block, but they are also key to meeting our 10Gbps single-engine throughput potential and help to sustain low latency for VDBs.

As with CPU, cache memory is required to drive peak loads on the Delphix Engine. More memory allows for more blocks to be read from the cache rather than going to less performant disks. Delphix stores cached data in a compressed format and only keeps a single copy of unique blocks in memory. These features give read performance across multiple VDBs provisioned from a single source dramatic improvements in speed, scalability and memory utilization.

10.13.3.1.2 Why does Delphix request reservations for CPU and memory?

Delphix performance can be greatly impacted there is contention over CPU or RAM. Reservations allow the engine to explicitly control those resources and avoids the possibility of contention with other VMs, even when resources are overcommitted.

10.13.3.1.3 Why does Delphix request Hyper-Threading (HT) be disabled?

Hyper-Threading can have a positive impact on many, but not all applications. Delphix has a different execution profile that does not benefit from Hyper-Threading. Factors such as significant memory bandwidth requirements and a high level of parallelism which requires a high number of shared locks mean that HT and Delphix do not generally work well together.

10.13.3.1.4 Why does Delphix request 4 controllers, and why must the storage be identical between them?

To provide optimal storage performance, you must spread data equally over the maximum (4) virtual SCSI controllers. To provide consistent performance between each of the four controllers, you need to ensure storage is identical between them.

10.13.3.1.5 Why must virtual disks (VMDKs) be thick provisioned and eager-zeroed?

Thick provisioning and eager zero ensure performance is top-notch from the start with no hiccups from expanding virtual resources.

10.13.3.1.6 Why is 20% free space required?

While the ZFS file system has a lot of features leveraged by Delphix, it loses efficiency as space decreases. 20% is the minimum that must be available for best performance.

10.13.3.1.7 Why does Delphix request you reserve CPU and RAM for Hypervisor overhead?

Based on VMWare's resource management guide and our own experiences with high IO throughput. Note there is no specific mechanism to assign resources to the hypervisor, the only way to preserve overhead is by not allocating resources to guests.

10.13.3.1.8 Why does Delphix generally want VMWare HA enabled, but DRS disabled?

VMware HA (High Availability) addresses outages that occur when a physical host goes down or is completely offline, by migrating the guest(s) to another physical host and restarting them. There is no real downside, it simply brings unavailable servers back online.

VMware DRS (Distributed Resource Scheduler) is for load balancing host resources in a cluster. Because of high IO and best practices configuration for optimal performance, our engine is typically not a good candidate for relocation.

10.13.3.1.9 Why does Delphix request you set power management to High-Performance Mode?

This will ensure power management will never impact performance by entering into a lower power state (also known as c-state).

10.14 Best practices for network configuration

1. For more information about network configuration refer to Network performance configuration options (see page 719)
2. Delphix engine <===> Target Host:*Implement standard requirements for optimal NFS/ISCSI performance:*
    a. Optimal physical network topology:
        - Low latency: < 1ms for 8K packets.
        - Network adjacency: minimize network hops, co-locate in the same blade enclosure, co-locate on the same physical host.
        - Eliminate all Layer 3+ devices - firewalls, IDS, packet filters (Deep Packet Inspection - DPI).
        - Multiple switches can add latency and fragmentation and reordering issues will add significant latency.
    b. Optimal throughput:
        - 10GbE physical uplinks or higher.

- Jumbo frames (typically MTU 9000) improves network efficiency for 8K packets: lower CPU, lower latency, greater throughput.
- All devices end-to-end *must* be configured for the larger frame size including switches, routers, fabric interconnects, hypervisors and servers.
- Traffic between two VMs on the same host is limited to ~16Gbps when leveraging built-in virtual networking.

c. Optimal logical flow:

- Disable QoS throttles limiting network utilization below line rate (e.g. HP Virtual Connect FlexFabric).
- Consider a dedicated VLAN (with jumbo frames) for NFS/iSCSI traffic.
  - VMware KB-configuring iSCSI port binding with multiple NICs in one vSwitch for VMware ESXi 6.0.x (2045040)[711]
  - VMware KB-considerations for using software iSCSI port binding in ESX/ESXi (2038869)[712]

d. NIC Teaming (at ESX layer) of multiple physical uplinks can provide additional throughput for higher workloads.

i. Examples: 4 x 1Gb NICs support up to 400 MBPS IO, 2 x 10Gb NICs support up to 2 GBPS IO.

ii. VMware KB-1004088[713] has NIC teaming recommendations, including `route-based-on-IP-hash` policy.

iii. VMware KB-1001938[714] has host requirements for physical link aggregation (LACP, EtherChannel).

iv. VMware KB-1007371[715], popular blog post[716] details problems with NIC selection using `dest-IP` hash.

e. **Jumbo frames check via ping**

```
Delphix Engine
$ ping -D -s [Target_IP] 8000
"ICMP Fragmentation needed and DF set from gateway" indicates MTU < 8028

Linux
$ ping -M do -s 8000 [Delphix_Engine_IP]
"Frag needed and DF set (mtu = xxxx)" indicates MTU < 8028

MacOS
```

711 https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2045040
712 https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2038869
713 https://kb.vmware.com/s/article/1004088
714 https://kb.vmware.com/s/article/1001938
715 https://kb.vmware.com/s/article/1007371?sliceId=2&dialogID=137604103&docTypeID=DT_KB_1_1&stateId=0+0+137602450
716 https://blogs.vmware.com/kb/2013/03/troubleshooting-network-teaming-problems-with-ip-hash.html

```
ping -D -s 8000 [Delphix_Engine_IP]

Note: "sudo sysctl -w net.inet.raw.maxdgram=16384" will increase the max
ICMP datagram size on Mac, allowing you to use -s 9000 on MacO

Windows
ping -f -l 8000 [Delphix_Engine_IP]

http://www.mylesgray.com/hardware/test-jumbo-frames-working/
```

Copy

f. Measure network bandwidth and latency:
   i. Latency in both directions should be < 1ms for an 8KB payload.
   ii. Network Hops should be minimized: `traceroute (Unix/Linux) / tracert (windows).`
   iii. Throughput in both directions: 50-100 MB/s on 1 GbE, 500-1000 MB/s on 10 GbE physical link.
   iv. Always use the CLI when possible.

g. NIC should use Auto-negotiate on Ethernet with a minimum of 1000Mbps.
   - Hard setting speed/duplex will limit network throughput below the line rate.

3. **Delphix <===> Staging server**(SQL Server, Sybase):

   Measure latency, bandwidth for transaction log restore performance

4. **Source <===> Delphix:**
   Measure latency, bandwidth to explain snapsync performance

5. **ESX host <===> ESX host**(ESX Cluster):
   a. This is because the entire memory footprint of the Delphix VM (more precisely, the entire address space of the ESX processes that comprise the VM) must be copied to the receiving ESX host, along with all changes to that address space as they happen. Our ZFS cache comprises the bulk of that memory and changes as I/O occurs, and that rate of change is governed by the network bandwidth available to the Delphix Engine.
   b. Attempting to live vMotion a DE with a 10Gb NIC over a 1Gb vMotion network is thus likely to fail.

10.14.1 Best practices network FAQ

10.14.1.1 Frequently asked questions

10.14.1.1.1 Why does Delphix request 10GE Ethernet?

As a matter of physics and standards - 10 gigabit (Gb) Ethernet can sustain approximately 1 gigabyte (GB) per second of throughput. With all our best practices applied, a Delphix Engine can achieve very close to that line speed, allowing for optimal load, engine, and license utilization. Lower network speeds may be

acceptable for low loads, while in some environments NIC teaming (e.g. LACP) may be required for top speeds.

10.14.1.1.2 Why does Delphix require < 1ms latency to TARGET servers and < 50ms to SOURCE servers?

Delphix leverages NFS and iSCSI (depending on platform) for live TARGET DB mounting over the network, so it's imperative that latency is as low as possible. Data coming from SOURCE servers is not generally as time-sensitive, so you need a minimum latency of < 50ms to ensure operational integrity.

10.14.1.1.3 Why does Delphix request Jumbo Frames?

Jumbo frames increase the Ethernet maximum transmission unit (MTU) from the default 1500 bytes to 9000 bytes. This has several effects such as decreasing CPU cycles by transferring fewer packets and increasing the engine throughput. You will find jumbo frames have a 10-20% real-world impact and are required (along with all other best practices) to handle peak loads of 800 - 1000MB/s on an 8 vCPU engine with a 10Gb network.

10.14.1.1.4 How does Delphix avoid communication impact with non-jumbo frame hosts when Jumbo Frames are enabled on the Delphix Host?

Path MTU Discovery[717] is the mechanism by which two hosts agree on the MTU leveraged for communication between them. This mechanism will ensure communication between both standard and Jumbo Frame enabled hosts works as expected.

10.14.1.1.5 When does Delphix recommend NIC teaming?

The Delphix Engine is capable of high throughput, but not every enterprise has sufficient network bandwidth to support it. Teaming is a less expensive way of increasing the bandwidth when compared to new hardware.

10.14.1.1.6 Why does Delphix recommend logical and physical and co-location?

The Delphix Engine leverages network connections extensively, so optimizing the latency whenever possible is very important - sometimes critical.

10.15 Best practices for Delphix engine data protection

The following are recommended best practices for data protection in the Delphix Engine.

1. For protection against physical host failure - leverage VMware HA.
2. For protection against storage failure - leverage Delphix replication.
3. For protection against administrative error - leverage storage snapshots.
4. For protection against site failure - leverage Delphix replication and/or Delphix Live Archive.

717 https://en.wikipedia.org/wiki/Path_MTU_Discovery

5. Infrastructure Backup of the Delphix VM:
   Must take a consistent group snapshot of all Delphix VM storage (system disk, VM configuration, database VMDK/RDMs)
   RTO, RPO are inferior compared to Delphix Replication. The granularity of restore is at the VM level: ***all-or-nothing.***
   a. Virtual machine backup: Create VM snapshot, backup (proxy server), remove the snapshot.
      Products use VMware APIs: NBU for VMware, TSM, Networker
      Limited to < 2TB because of time to backup, impact on running VM
   b. Storage array backup: Take a consistent storage snapshot, replicate to tape/VTL media server, remove the snapshot.
      Products include Hitachi Shadow Copy, EMC SnapCopy, HP Business/Snap Copy.

10.15.1 Best practices Delphix engine data protection FAQ

10.15.1.1 Frequently Asked Questions

10.15.1.1.1 Why does Delphix recommend SAN snapshots or Delphix replication for backup?

There are a few possible methods for data protection of the Delphix Engine. Those methods are SAN snapshots, Delphix replication, and virtual machine snapshots (for very small engines only). Because the Delphix Engine is itself a backup of source environments, many customers simply plan to rebuild in the event of a disaster.

10.15.1.1.2 What is the DXToolkit, and how can it help?

The professional services team has created a Perl-based “DXToolkit” which can help export and import certain configuration data over web services. This toolkit can be leveraged to assist with would normally be a manual re-install outside of the above methods.

For further detail around data protection, please speak with your Delphix contact.

10.15.1.1.3 Can I use a VMware-based backup solution such as VEEAM to backup my Delphix Engine?

Yes, VMWare backup solutions are useful for backing up guest VMs. However, Delphix suggests that you only use this approach for Delphix Engines which have a smaller storage footprint (perhaps < 2 TB) and are less active.

Running this type of backup puts a load on the environment, which might adversely impact Delphix VM performance.

10.15.1.1.4 Can I use a VMware snapshot for backing up Delphix for a small window – for example, during an engine upgrade?

Yes. However, even though snapshots are instantaneous, they track changes separately from the base disks and can grow to consume as much space as the original.

> Upgrades, in particular, can change substantial amounts of data.
> If you lose physical disks, snapshots are useless because it needs them to make up the current state of VM.
> A Delphix Engine is often allocated multiple terabytes of storage and is often very busy due to load aggregation from virtual databases on multiple target servers, so this approach may be challenging.
> Snapshots cannot detect storage corruption.

10.15.1.1.5 Can I use a Storage snapshot solution to protect Delphix against Storage and Delphix corruption?

Yes. However, please note that the caveats which apply to VMWare snapshots will also apply here.

> A specific concern related to storage layer snapshots is that you must create a consistency group that contains both the OS and Data disks.

10.15.1.1.6 Can I use RMAN to backup my VDBs just like a Physical database to provide extra protection?

You can backup Delphix VDBs using Oracle RMAN tools, but the recovery database would first require re-hydration of that VDB, which might take up equivalent production storage space.

Furthermore, that re-hydrated database needs to be brought into the Delphix framework as a dSource, after which you can provision a VDB to complete recovery. The whole process might take hours or days to recover.

The best approach is to use the VDB Snapshot capability to backup VDB frequently and then leverage Delphix Replication capability to protect underlying Delphix storage, which holds that VDB snapshot.

10.16 Best Practices for source DB and OS settings

10.16.1 Best practices

10.16.1.1 Oracle

1. ARCHIVELOG must be enabled: `select log_mode from v$database.`
2. FORCE LOGGING should be enabled to ensure VDBs are not missing data. When NOLOGGING redo is applied during provision, the resulting VDB will be missing changes. Tables with NOLOGGING changes will throw corruption errors when scanned.
3. Block Change Tracking should be enabled to minimize snapsync time.
4. Consult the documentation for Oracle Standby sources.
5. If the database is encrypted with Oracle TDE (Transparent Data Encryption) plan your Delphix storage requirements with the expectation of minimal compression. Customer Observation: space usage for a TDE dSource copy was 92% (2.44 TB) of the source database size (2.67 TB). A typical Oracle dSource copy for a non-TDE database consumes 40% of the source database size.

10.16.1.2 SQL server

1. If using a FULL recovery model, configure your dSource to stay synchronized using Transaction Log backups. This will usually allow the dSource to stay in sync using much less network and disk IO than Full or Differential backups.
2. Ensure that the number of Virtual Log Files (VLFs) in the Source database is appropriate. Databases with hundreds or thousands of VLFs will experience slower provisioning times as SQL Server must do more work during recovery. The blog post A Busy/Accidental DBA's Guide to Managing VLFs[718] has helpful information here.
3. Review index maintenance and defragmentation operations on your Source database, to ensure that they are not running more often than necessary. Many maintenance operations are logged, and can result in extremely large log backups without much benefit to the database server. See Stop Worrying About SQL Server Fragmentation[719].
4. If configuring a dSource to use Full or Differential backups, ensure that Source database backups do not run at the same time as database maintenance or large batch operations. This can significantly increase the amount of time required to perform database recovery and will slow down the provisioning of VDBs.

718 http://adventuresinsql.com/2009/12/a-busyaccidental-dbas-guide-to-managing-vlfs/
719 https://www.brentozar.com/archive/2012/08/sql-server-index-fragmentation/

5. For best compatibility with the Delphix Engine, avoid taking log backups more frequently than every 10 minutes. The validated sync process requires time to detect and validate log backups before they can be applied to a dSource, and extremely high-frequency log backups can make it difficult for a dSource to stay in sync.
6. If you are using SQL Authentication for database logins, and you want Source database users to be mapped to database logins when provisioning VDBs, ensure that the SID of your database logins is the same between your Source and Target environments (the passwords can be different). For more details, see the Correcting Orphaned SQL Server Database Users (KBA1111)[720] KB article.
7. To maximize dSource performance, ensure that backups are being taken to a disk drive or network location that can be accessed by the Staging Server at high speed.

10.17 Best practices for target DB and OS settings

10.17.1 Target database application settings

1. Oracle:
    a. Provision with 3 x 5GB online redo logs (minimum) to avoid pause when transaction logs wraparound.
    b. Provision in NOARCHIVELOG mode to reduce transaction log IO. Masking, Test, QA VDBs rarely need point-in-time rewind
    c. Always check initialization parameters inherited from parent, remove any expensive or irrelevant parameters.
        i. `DB_CACHE_SIZE, SGA_TARGET` : set based on target system being compared to.
        ii. `FILESYSTEMIO_OPTIONS` to `SETALL`. Any other setting inherited from source is probably wrong.
        iii. `DB_BLOCK_CHECKSUM, DB_BLOCK_CHECKING, DB_LOST_WRITE_PROTECT, DB_ULTRA_SAFE` : set to default values to minimize impact.
        iv. `PARALLEL_DEGREE_POLICY` to `AUTO`, `PARALLEL_MAX_SERVERS` default, `PARALLEL_EXECUTION_MESSAGE_SIZE` to 32768 (maximum): improve PQ performance.
        v. `FAST_START_MTTR_TARGET:` drives steady write activity. Set based on target system being compared to.
        vi. Consider non-durable commits for Masking, Test, QA, UAT: set `COMMIT_WAIT = NOWAIT, COMMIT_LOGGING = BATCH`
    d. Use Oracle Direct NFS (dNFS) for 11.2.0.4+ (**unstable**on older releases):
        i. Recommended documentation:

720 https://support.delphix.com/Delphix_Virtualization_Engine/MSSQL_Server/Correcting_Orphaned_SQL_Server_Database_Users_(KBA1111)

- Oracle direct NFS configuration[721]
- Configuration examples and troubleshooting blog[722] from Helmut Hutzler
- Sample oranfstab to leverage multiple network paths for Delphix VDB

ii. Set DNFS_batch_size[723] = 128 (default is 4096). This is a good starting point and sufficient for most workloads.

iii. Tune TCP stack: set `tcp_adv_win_scale` = 2 due to workaround hard-coded Oracle dNFS TCP buffer size.

iv. Check Alert Log, V$DNFS_SERVERS, V$DNFS_FILES, V$DNFS_STATS to verify proper working (sample here[724]).

e. Create AWR snapshots around a reference customer workload, generate an AWR report.

i. AWR snap before/after workload: `SQL> exec DBMS_WORKLOAD_REPOSITORY.CREATE_SNAPSHOT();`

ii. AWR report between the snaps: `SQL> @?/rdbms/admin/awrrpte`

f. Generate ASH report to diagnose bottlenecks while a workload is running.

i. `SQL> @?/rdbms/admin/ashrpt`

g. Run synthetic benchmark `sc-workload.`

h. Where `db file scattered read` (multiblock cached read) latency is high consult this Support KB: How to mitigate multi-block read performance on Oracle 10g[725]

i. Improve distributed query performance by modifying dblinks to use local IPs instead of SCAN IPs.

j. NFS recommended mount options for Oracle RAC/SI: Oracle support note 359515.1[726]

10.17.2 Memory and CPU

- 32 GB RAM minimum
- 4 vCPU minimum

10.17.3 Windows and MSSQL specific

The SQL Server Instances hosted on the Targets should have a Maximum Memory set. Also ensure that at all times, at least 10% of total memory and at least 4GB of available memory is available for OS operations.

721 https://dbtut.com/index.php/2019/05/01/oracle-direct-nfs-dnfs/
722 https://www.hhutzler.de/blog/dnfs-direct-nfs-setup-for-11-2-0-4-3-node-rac-cluster-on-oel-6-4/
723 https://docs.oracle.com/cd/E11882_01/server.112/e40402/initparams083.htm#REFRN10334
724 https://sort.veritas.com/public/documents/sfha/6.0/linux/productguides/html/virtualstore_admin/ch03s05.htm
725 https://support.delphix.com/Delphix_Virtualization_Engine/Oracle/How_to_Mitigate_Multi-Block_Read_Performance_on_Oracle_10g
726 https://support.oracle.com/rs?type=doc&id=359515.1

10.17.4 Network requirements

Target server should have < 1ms latency to the Delphix Engine.

10.17.5 Target Host OS Settings

1. Existing documentation on Target OS practices: Target host configuration options for improved performance (see page 907)
2. HP-UX 11.31+
   - Async NFS direct I/O[727]: HP-UX requires Oracle `disk_asynch_io` turned off for filesystems
3. IBM AIX:
   a. Consult IBM documentation on AIX TCP Tuning
   b. AIX TCP tuning prezo[728]
4. Windows:
   a. Anti-virus programs can impact both performance and operation. Delphix recommends anti-virus scanning exclude folders where Delphix files are maintained, in addition to the normal exclusions put in place for MSSQL operation.
   b. Delphix Connector (aka DX Connector):
      i. Plan 3-5GB for the Delphix Connector installation.
      ii. Windows does not yet have ssh, so Delphix developed the "DX Connector for Windows target host communication.
      iii. The connector must be installed on all Target Windows hosts.
      iv. The connector supports two modes – v1 and v2 both use the same application binaries.
      v. The connector v1 process is used to bootstrap the v2 process on a target. This opens a DSP session back to the Delphix Engine (The same thing is done via SSH on U*nix Targets)
      vi. v2 mode is required to enable SQL hooks
      vii. The connector can always be downloaded from a local Delphix Engine at: http://<delphix_engine>/connector/DelphixConnectorInstaller.exe.
      viii. The connector is backwards compatible, so it is not always necessary to upgrade it during a Delphix upgrade.
   c. iSCSI connections:
      i. Read the following for general awareness of iSCSI limits
      ii. In addition to the hard limits on iSCSI connections, consideration must be given to the RAM, CPU and Network to provide sufficient resources for the load on any Target or Staging host.
      iii. To increase the iSCSI timeout on both Target and Staging hosts.

727 https://h20564.www2.hpe.com/hpsc/doc/public/display?docId=emr_na-c02444831&sp4ts.oid=3553037&lang=en-us&cc=us

728 https://www.circle4.com/movies/budapest/baixperfpt3.pdf

iv. In certain circumstances it's possible that iSCSI startup will not complete before the SQL Service attempts to start a database. In such circumstances, it can be helpful to ensure the SQL service depends on the iSCSI service.

- Example: c:\> sc config "MSSQLServer" depend="Microsoft iSCSI Initiator Service"

v. **Note** that any changes to iSCSI are system-wide and could potentially impact other applications also leveraging that feature.

d. Enable receive side scaling (RSS) (see page 2341) on each network interface that Delphix will be connecting to.

10.17.6 Receive side scaling (RSS) for windows staging target and targets

Enabling Receive Side Scaling (RSS) on a Windows Target and Staging Target can have a significant improvement in the overall IO throughput to the Delphix Engine and is a best practice. RSS enables network adapters to distribute the kernel-mode network processing load across multiple processor cores in multi-core computers. The distribution of this processing makes it possible to support higher network traffic loads than would be possible if only a single core were to be used.

More information on RSS can be found here[729].

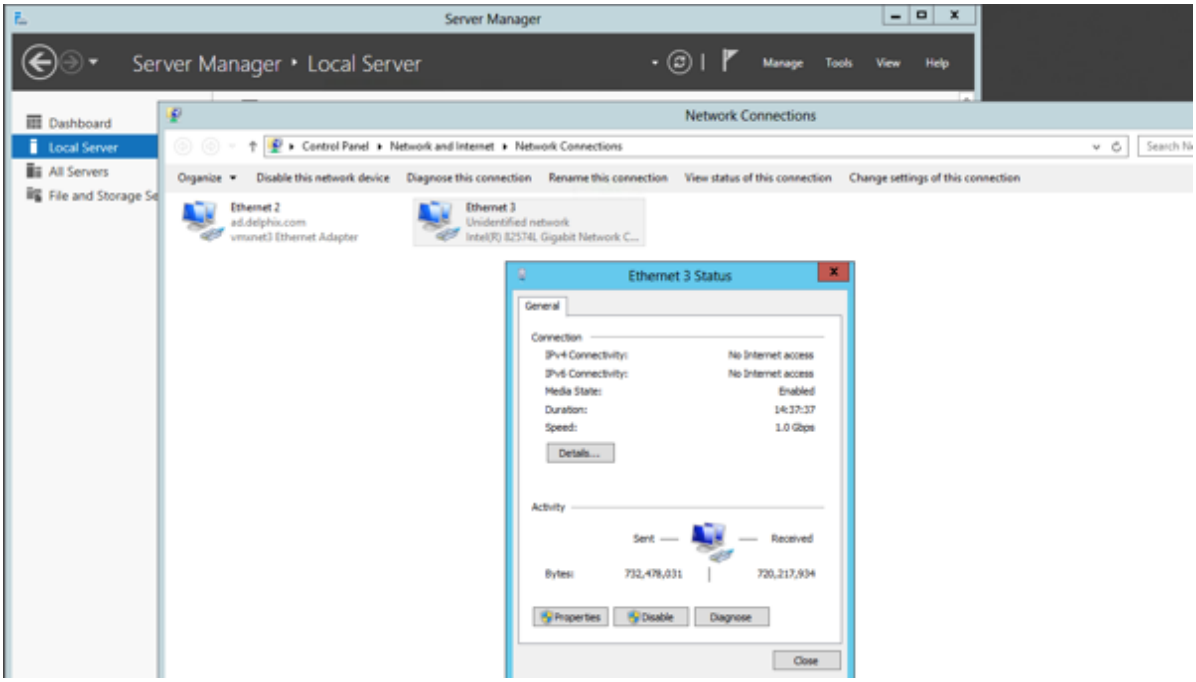
> Enabling RSS on the network interface will force the network service to restart and will cause a momentary loss of connectivity on that network interface.

> Because hyper-threaded CPUs on the same core processor share the same execution engine, the effect is not the same as having multiple core processors. For this reason, RSS does not use hyper-threaded processors.

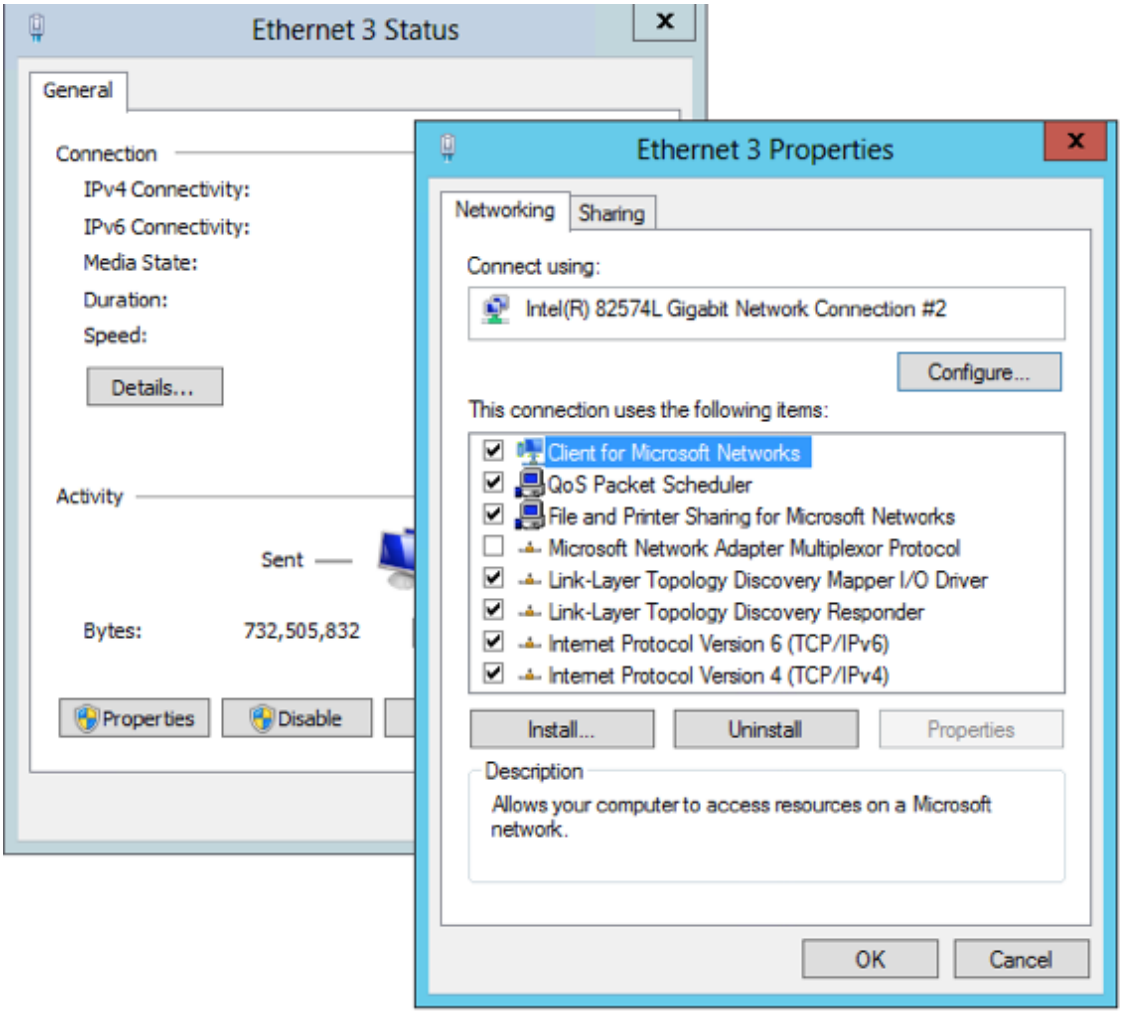
10.17.6.1 Steps to implement RSS on windows

1. From Server Manager/Local Server/Network Connections select the NIC that Delphix will be connecting to.

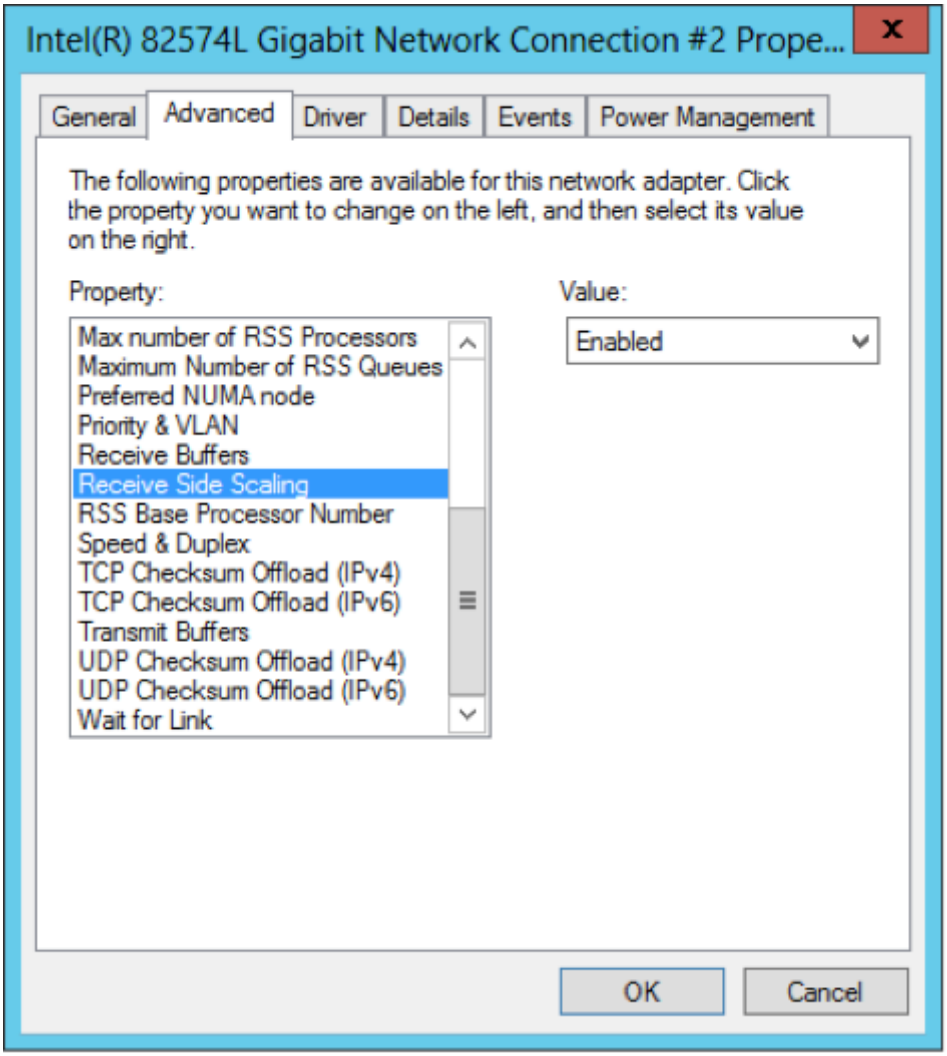
729 https://technet.microsoft.com/library/hh997036.aspx



2. Select Properties and then Configure.



3. From the Property menu on the left, select Receive Side Scaling and select 'OK' to close each of the open windows.



10.18 Best practices for staging targets

This host is called a "staging target" because it has much in common with other targets, such as the remote storage mount to the Continuous Data Engine. Because it leverages remote storage over the network, the staging target only needs enough disk capacity for the OS, database application, and any relevant logs or tools. A staging target is a requirement on all platforms that Delphix supports except Oracle, but you can also use it for Oracle to replay logs leveraging validated sync. (see page 2133)

10.18.1 Memory and CPU

1. 32 GB RAM minimum
2. 4 vCPU minimum

10.18.2 General guidance for staging servers (Multi-platform)

1. Delphix recommends dedicated Staging servers for role/architecture separation. However, any Target server can be used as Staging.
2. In cases where the same server is used as both Staging and Target, we strongly recommend a dedicated instance/install for staging to avoid confusion.
3. Delphix recommends at least one Staging server per Delphix Engine to avoid the possibility of a single point of failure across multiple engines.
4. If a staging server is shared among multiple Delphix Engines, please ensure that a dedicated SQL Server Instance is created for each Delphix Engine.
5. Configuration / performance factors:
    a. Transaction log generation rate.
    b. Number of VDBs.

> Precise guidance on these items has not yet been defined. In general, if there is a heavy log generation rate and few VDBs, the ideal recommendation is to have at least 1 Staging Target per Delphix Engine.

10.18.3 Disk / local storage

1. The only local storage needed is for the OS and application with default databases.
2. Storage for a staging database is provided from the Delphix Engine, which is mounted over the network similar to any Target host (NFS/iSCSI).
3. If the customer has a standard DB server build, their standard storage sizing is probably fine.
4. If a recommendation is still needed, suggest 30GB for OS and application and any tools needed.

10.18.4 Network requirements

1. The Staging Target engages in network data transfers between Staging and the Source backup shared location as well as between Staging and the Delphix Engine.
2. The Staging Target is also a Target server, and as such should have < 1ms latency to the Delphix Engine (and low latency to the Source backup, when possible).
3. If the change rate on the Source database(s) is > 1 Gb/sec, the recommended network bandwidth to support network transfers is 10 Gb/sec.

4. In cases where only 1 Gb/sec network bandwidth is available, segregation of each network is recommended to reduce network contention.
5. Ensure that the virtual NIC is using the standard vmxnet3 adapter and not Intel for VMWare based clients. Logical IO errors have been reported while using Intel instead of vmxnet3 adapter.

10.18.5 Windows and MSSQL specific

1. An MSSQL Server Instance used for Staging should not be clustered.
2. Staging should not be hosted on Windows 2003 - extended support ended July 14, 2015. It is also the first Windows version with iSCSI support and is not ideal.
3. The SQL Server Instances hosted on the Staging Target should have a Maximum Memory set. Also ensure that at all times, at least 10% of total memory is available for OS operations.
4. Only system databases (Master/MSDB/Temp/MSDB) are kept on local storage. All other data is read/written to the Delphix Engine.
5. Windows iSCSI configuration and limits for v2p, target and staging hosts.
6. Ensure that Receive Side Scaling (RSS) (see page 2130) is enabled on every network interface that Delphix will be connecting to.

Delphix Engine controls the number of concurrent restore operations that can run on a staging environment by the validated sync workers, which means we throttle the number of restore operations done by validated sync workers running for different staging databases on the staging environment, with five executing at a time and others waiting for their turn as per First Come First Serve scheduling. This is done to improve overall system performance by reducing resource contention, disk I/O, and network traffic. Also, note that this limit is per Delphix Engine which is connecting to the staging environment.

Following are the limiting factors which will come into play when looking at the performance of staging databases on a staging environment when a validated sync worker runs to keep up with the production databases:

- Backup generation frequency: With higher backup frequency, increased restore time will be seen as the pre-provisioning worker will keep ingesting previous backups while new backups are being generated.
- Staging database count: When multiple staging databases are hosted on the same server, the backup ingestion load on the staging host will increase. Additionally, if the frequency of backups is high, there will be a greater number of candidates (pre-provisioning workers) waiting in the queue.
- Number of VDB hosted on the server
- Multiple Delphix Engines connecting to the same staging environment will increase the number of parallel restore operations running on the staging environment and contribute to the performance.

Below are the troubleshooting steps for improving performance:

- Have dedicated Staging servers for role/architecture separation from VDB
- Add CPU/Memory
- Decrease backup frequency
- Introduce dedicated networks

Sharing an example of performance in a given setup for reference purposes.

> **Note**
> The below findings are from a non-production setup

10.18.5.1 Environment details

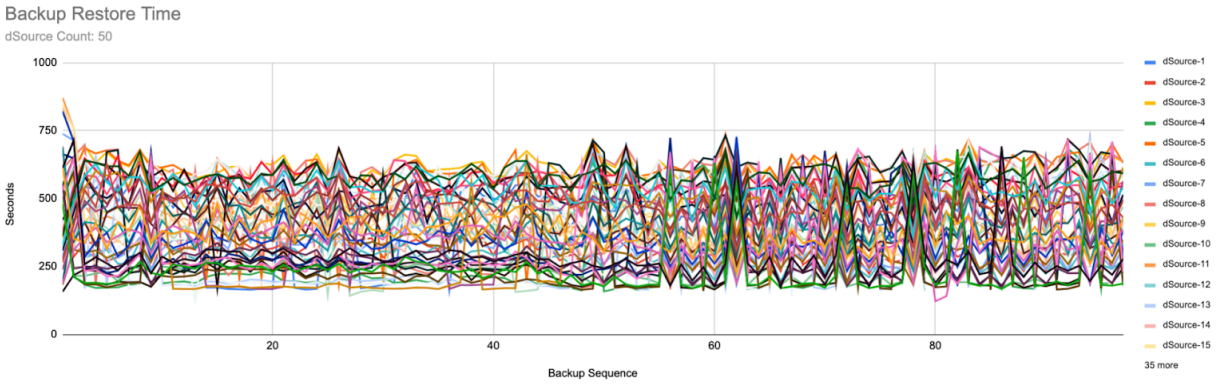
- Staging Host: 64 GB Memory, 8 vCPUs, ESXi version: 7.0.3
- Backup File Size: 200MB
- User for linking: Database user

10.18.5.2 Setup notes:

- No other operations were executed on the Delphix Engine other than pre-provisioning worker running.
- No virtual database existed on the staging host.
- The source servers and the Delphix Engine are all on the same on-prem data center.
- Only one Delphix Engine was connecting to the staging host.

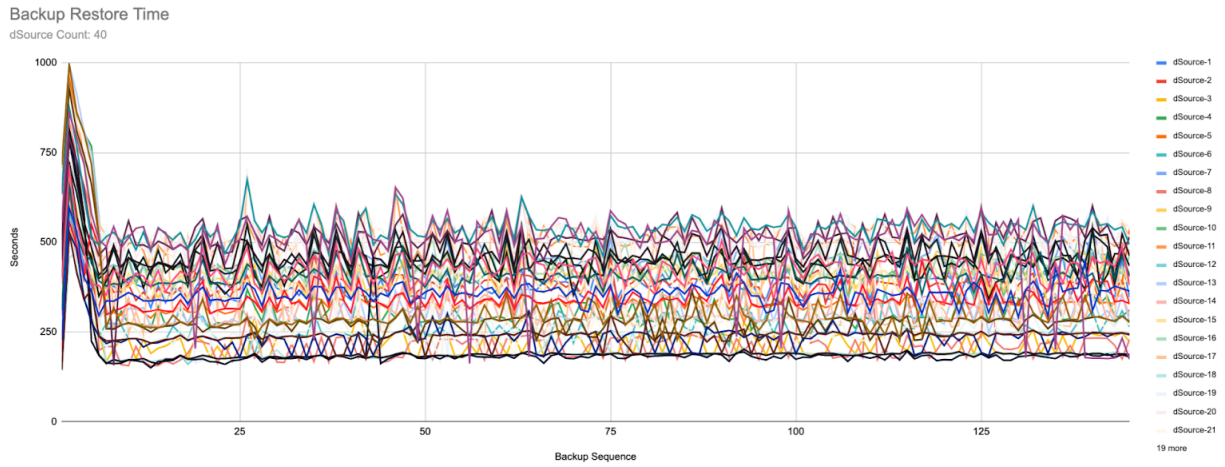
10.18.5.3 Performance Scenario 1

For a staging host with the above configuration supporting 50 staging databases on a single database instance, and with every dSource having a backup at the 15-minute interval, the time taken to restore these transaction logs stay under 13 minutes(< 750 seconds) on average and hence the staging databases keep up with the backups.



10.18.5.4 Performance Scenario 2

The same setup could support frequent backups, that is every 10 min, but required the staging databases to be reduced. For example, 40 staging databases on a single database instance could support backups every 10 minutes without causing any lag.



10.19 Best practices for storage

1. Configuration:
    a. Virtual Disks (VMDK) with spinning or tiered media must be thick provisioned + lazy zeroed.
        - For storage which is 100% SSD/EFD/Flash-based, continue to thick provision, however, eager zero is not necessary.
    b. VMDKs may be homed on VMFS or RDM storage, however, VMFS is much more common and generally preferred.
    c. Regardless of VMFS/RDM selection, physical LUNs must have uniform characteristics (RAID, spindle count, tier) and should be thin provisioned to save time.
    d. Storage allocated must be identical between each vSCSI controller.
    e. The supported maximum of four virtual SCSI controllers (PVSCSI (default) or LSI Logic Parallel) must be enabled. A mix of different types of SCSI controllers is not supported within the engine.
    f. Virtual Disks must be evenly distributed across the 4 virtual SCSI controllers. For example, 8 virtual disks should be configured as 2 disks per controller: SCSI (0:0), SCSI (0:1), SCSI (1:0), SCSI (1:1), SCSI (2:0), SCSI (2:1), SCSI (3:0), SCSI (3:1).
        - You don't need to account for the OS in the even distribution of disks across controllers, just pick one; the OS doesn't place a substantial load on the controller.
    g. To provision VMDK disks over 16TB in size, the vSphere web client must be used, the win32 client will return an error.
    h. Delphix requires 127GB for the system disk. older versions defaulted to 150GB.
        - Due to our unified OVA for masking, the 127GB requirement also applies to Masking Engines
    i. VMDK for the Delphix Engine OS is often stored on the same VMFS volume as the Delphix VM definition file (aka VMX). In that case, the VMFS volume must have sufficient space to hold the Delphix VMX Configuration, the VDMK for the system disk, a swap/paging area if the memory reservation was not enabled for the Delphix Engine (or to suspend the system), and any related VMware logging.

   j. Set ESX Storage Multipathing IO optimization (KB 2072070[730]).
      i. Set multipathing policy to round-robin.
      ii. Set path switching IO operation limit to 1 (default 1000).
   k. Verify that the queue depth setting on the virtual disks used for VMware is appropriate based on the minimum of the HBA type or the underlying storage's combined queue depth for all hosts attached to the same controller
      - See this VMware KB article[731] for how to check/set the queue depth

2. Testing:
   a. Run Storage Performance Tool on the raw storage **before any engine configuration**. This is a one-time opportunity for each engine upon installation
   b. Required maximum storage latency is < 2ms for writes and < 10ms (95th percentile) for small random reads. Minimum passing grades: 4KB/8KB reads (B-), 1MB reads (A), 1KB/128KB writes (A).
   c. If working with the Delphix Professional Services team, we would expect to run additional baseline performance measurements via composite tools and scripts.
      - e.g. "Sanity Check" (Oracle) or "DiskSpd" (MSSQL)

3. Detail Discussion:
   a. Before beginning any discussion on storage performance, or at the beginning of the installation, collecting the following specs from your storage administrator will assist in understanding
   b. Vendor, Model (For example: EMC VMAX 40k, HP 3PAR StoreServ 7000)
   c. IO latency SLO (For example 5ms 99.999%)
   d. IOPS/GB SLO (For example: 0.68 IOPS/GB for EMC Gold-1)
   e. Cache type and size (For example FAST cache 768GB)
   f. Tier, #Pools; if auto – tiering; relocation schedule (For example: Gold/Silver/Auto/3 pools/etc)
   g. Pool detail: (#) drives, RPM, Type (For example: Pool1: (20) EFD, (30) 15k SAS, Pool 2: (40) 10k SATA)
   h. Connection (For example: 16Gb Fibre Channel, 10Gb iSCSI, 20Gb NFS)
   i. Dedicated or Shared pool (how many applications/servers)

730 http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2072070
731 https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1267

11 Data backup and recovery solutions

Cloning an existing Delphix Engine by using hypervisor VM cloning features or by copying existing Engine storage to a new virtual machine is currently not supported. You can choose from the following methods for backup or duplication of an Engine.

11.1 Data backup and recovery solutions

Learn about the suite of Delphix Backup and Recovery Strategies

Delphix Continuous Vault (see page 2350)

Backup and Recovery Strategies for the Delphix Engine (see page 2359)

Replication (see page 2367)

Selective Data Distribution (see page 2408)

11.2 Delphix continuous vault

11.2.1 Overview

The Delphix Engine has a base feature set that is compelling as a data protection solution. An enhancement is being introduced that can further prevent snapshot and database loss in the event of a ransomware attack.

Delphix continuous vault for ransomware protection allows organizations to recover their application data access much faster than traditional backup solutions, in case of malicious attacks.

The Continuous Vault solution provides built-in protection against ransomware attacks by frequently refreshing production application data to a virtual database (dSource), from which you can recover applications almost instantly to any valid point in time prior to the encryption attack by leveraging Delphix VDB provisioning capabilities. The refresh interval between the production application and the Delphix dSource can be configured to be up to seconds, for business-critical applications (not applicable to all DB technologies).

There are two variants of Delphix Continuous Vault:

- **Replica Continuous Vault**, which replicates critical business DB data stored on Delphix Engines to a new target Engine called Replica Continuous Vault.
- **Single Engine Continuous Vault**, which protects critical business DB data stored on a Delphix Engine by preventing manual deletion of protected sources or snapshots.

Once securely stored on the Continuous Vault, the DB data can be used to recover business applications with very low RTO and RPO.

> Requires Technical Services Consult
> Delphix **requires** a Technical Services assessment prior to deployment and configuration of Continuous Vault. The process of configuring a Delphix Continuous Vault replication profile is simple; the assessment is required because each application has specific data protection and recovery requirements and we must ensure that Delphix can respond to them appropriately. To schedule an assessment, please contact your Customer Success Manager.

11.2.2 Replica continuous vault

The Replica Continuous Vault feature is available via CLI or via the Continuous Vault UI section of the Replication page. This UI provides functions for creating Continuous Vault replication profiles from scratch and converting existing profiles to be locked.

11.2.2.1 Advantages

The Replica Continuous Vault variant provides the following advantages:

- Creates a separation of responsibilities between the two Delphix Engines
  - One engine is used for regular Virtualization cases (ingestion, VDBs, SDD)
  - Another engine is used for ransomware protection.
- Creates a physical separation by allowing the admin to isolate and secure the locked Delphix Engine.
  - Only the DSP port has to be open for replication.
  - No ports are needed for JDBC, NFS, or SSH until VDBs need to be created to export data. This also prevents attack vectors related to any of those protocols.
- Can assist with making deployments and security reviews easier to pass since the locked Delphix Engine is isolated and has a single purpose to reduce potential attack vectors.

In the event of a ransomware attack on a primary engine source being compromised or corrupted, you can provision a VDB on the replica in the locked namespace of the replication target – similar to the normal replication namespace. This process can be further outlined in the Provisioning From Replicated Data Sources or VDBs (see page 2403) article. If a complete recovery of the primary engine is needed, please contact Delphix Support.

11.2.2.2 Implementation

This feature adds a property to the replication namespace and specifications called “locked”. Additional dSources, VDBs, groups, and domains can be added to locked replication specs, but data sources cannot be removed after doing so. Failover on the target Delphix Engine is not allowed if the namespace is part of the locked replication spec. The retention policy duration on a locked namespace can be modified as long as the duration is either being increased, or it is being decreased to a minimum of 100 days.

The time configuration on Delphix engines with Continuous Vault enabled cannot be changed. This is to prevent attempts at bypassing retention policies in order to try and delete snapshots on the target. Also, the

factory reset operation is forbdiden when at least one locked replication specification or namespace is present.

A fault is now generated on the Continuous Vault target for a locked namespace that has not received a successful replication update in 12 hours. Upon request, Delphix Support can change this value. New replication specs must also have automatic replication enabled and a satisfactory replication schedule.

11.2.2.3 CLI functions

Create a locked replication profile.

```
[user.hostname]> replication spec
[user.hostname] replication spec> create
[user.hostname] replication spec create *> set name=locked-spec-1
[user.hostname] replication spec create *> set objectSpecification.objects=Untitled/
dbname
[user.hostname] replication spec create *> set targetHost=example.delphix.com
[user.hostname] replication spec create *> set targetPrincipal=admin
[user.hostname] replication spec create *> set targetCredential.password=delphix
[user.hostname] replication spec create *> set lockedProfile=true
[user.hostname] replication spec create *> commit
    `REPLICATION_SPEC-3
[user.hostname] replication spec> select locked-spec-1
[user.hostname] replication spec 'locked-spec-1'> get
    type: ReplicationSpec
    name: locked-spec-1
    automaticReplication: false
    bandwidthLimit: 0
    description: (unset)
    encrypted: false
    lockedProfile: true <------------------------------ LOCKED
    numberOfConnections: 1
    objectSpecification:
        type: ReplicationList
        name: (unset)
        objects: Untitled/dbname
    reference: REPLICATION_SPEC-3
    runtime:
        type: ReplicationSpecRuntime
    schedule: (unset)
    tag: 5570be25-dbcf-48c3-b2d2-dd2c65eb98b7
    targetCredential:
        type: PasswordCredential
        password: ********
    targetHost: example.delphix.com
    targetPort: 8415
    targetPrincipal: admin
    useSystemSocksSetting: false
[user.hostname] replication spec 'locked-spec-1'> cd ..
[user.hostname] replication spec>
```

Lock an unlocked replication profile.

```
[user.hostname]> replication spec create
[user.hostname] replication spec create *> set name=locked-spec-2
[user.hostname] replication spec create *> set objectSpecification.objects=Untitled/
dbname
[user.hostname] replication spec create *> set targetHost=example.delphix.com
[user.hostname] replication spec create *> set targetPrincipal=admin
[user.hostname] replication spec create *> set targetCredential.password=delphix
[user.hostname] replication spec create *> commit
    `REPLICATION_SPEC-4
[user.hostname]> replication spec select locked-spec-2
[user.hostname] replication spec 'locked-spec-2'> get
    type: ReplicationSpec
    name: locked-spec-2
    automaticReplication: false
    bandwidthLimit: 0
    description: (unset)
    encrypted: false
    lockedProfile: false
    numberOfConnections: 1
    objectSpecification:
        type: ReplicationList
        name: (unset)
        objects: Untitled/dbname
    reference: REPLICATION_SPEC-4
    runtime:
        type: ReplicationSpecRuntime
    schedule: (unset)
    tag: e8608d05-0693-440d-8a2b-8c6cbfe06a62
    targetCredential:
        type: PasswordCredential
        password: ********
    targetHost: example.delphix.com
    targetPort: 8415
    targetPrincipal: admin
    useSystemSocksSetting: false
[user.hostname] replication spec 'locked-spec-2'> update
[user.hostname] replication spec 'locked-spec-2' update *> set lockedProfile=true
[user.hostname] replication spec 'locked-spec-2' update *> commit
[user.hostname] replication spec 'locked-spec-2'> get lockedProfile
    true
[user.hostname] replication spec 'locked-spec-2'>
```

Verify the locked status of a namespace.

```
[user.hostname]> namespace
[user.hostname] namespace> list
NAME
[user.hostname]-1
[user.hostname]-3
```

```
[user.hostname] namespace> select [user.hostname]-3
[user.hostname] namespace '[user.hostname]-3'> get
    type: Namespace
    name: [user.hostname]-3
    description: (unset)
    failedOver: false
    locked: true   <------------------------------ LOCKED
    namespaceType: REPLICATION
    reference: NAMESPACE-4
    secureNamespace: false
    tag: 5570be25-dbcf-48c3-b2d2-dd2c65eb98b7
[user.hostname] namespace '[user.hostname]-3'>
```

Verify that the namespace cannot be deleted or failed over.

```
[user.hostname] namespace '[user.hostname]-1'> delete
[user.hostname] namespace '[user.hostname]-1' delete *> commit
   Error: Namespace "[user.hostname]-1" is locked and cannot be deleted.
  Action: Cannot delete a locked namespace.
[user.hostname] namespace '[user.hostname]-1' delete *> discard
[user.hostname] namespace '[user.hostname]-1'> failover
[user.hostname] namespace '[user.hostname]-1' failover *> commit
   Error: Namespace "[user.hostname]-1" is locked and cannot be failed over.
  Action: Cannot failover a locked namespace.
[user.hostname] namespace '[user.hostname]-1' failover *> discard
```

Verify that the replication profile cannot be deleted or modified. Objects can still be added to the replication profile.

```
[user.hostname]> replication spec
[user.hostname] replication spec> select locked-spec-1
[user.hostname] replication spec 'locked-spec-1'> delete
[user.hostname] replication spec 'locked-spec-1' delete *> commit
   Error: The replication profile is locked and cannot be deleted.
  Action: Select an unlocked profile to delete.
[user.hostname] replication spec 'locked-spec-1' delete *> discard
[user.hostname] replication spec 'locked-spec-1'> update
[user.hostname] replication spec 'locked-spec-1' update *> set automaticReplication=t
rue
[user.hostname] replication spec 'locked-spec-1' update *> commit
   Error: The replication profile is locked and cannot be updated.
  Action: Select an unlocked profile to update.
[user.hostname] replication spec 'locked-spec-1' update *> discard
[user.hostname] replication spec 'locked-spec-1'> update
[user.hostname] replication spec 'locked-spec-1' update *> set
objectSpecification.objects=Untitled/dbname,Group:/Untitled
[user.hostname] replication spec 'locked-spec-1' update *> commit
[user.hostname] replication spec 'locked-spec-1'> update
[user.hostname] replication spec 'locked-spec-1' update *> set
objectSpecification.objects=Untitled/dbname
[user.hostname] replication spec 'locked-spec-1' update *> commit
```

```
   Error: Objects cannot be removed from a locked replication profile.
  Action: Select an unlocked profile to update.
[user.hostname] replication spec 'locked-spec-1' update *> discard
[user.hostname] replication spec 'locked-spec-1'>
```

Create a replica retention policy and apply it to the locked namespace.

```
[user.hostname]> policy
[user.hostname] policy> createAndApply
[user.hostname] policy createAndApply *> set policy.type=ReplicaRetentionPolicy
[user.hostname] policy createAndApply *> set policy.duration=6
[user.hostname] policy createAndApply *> set policy.durationUnit=YEAR
[user.hostname] policy createAndApply *> set target=Namespace:/[user.hostname]-1
[user.hostname] policy createAndApply *> set policy.name="Six Years"
[user.hostname] policy createAndApply *> get
    type: PolicyCreateAndApplyParameters
    policy:
        type: ReplicaRetentionPolicy (*)
        name: Six Years (*)
        customized: false
        duration: 6 (*)
        durationUnit: YEAR (*)
    target: Namespace:/[user.hostname]-1 (*)
[user.hostname] policy createAndApply *> commit
    `POLICY_REPLICA_RETENTION-30
[user.hostname] policy>
```

Verify that the replica retention policy cannot be deleted or modified.

```
[user.hostname] policy> select POLICY_REPLICA_RETENTION-30
[user.hostname] policy 'Six Years'> delete
[user.hostname] policy 'Six Years' delete *> commit
   Error: The replica retention policy "Six Years" could not be removed because the
target namespace "[user.hostname]-1" is locked.
[user.hostname] policy 'Six Years' delete *> discard
[user.hostname] policy 'Six Years'> update
[user.hostname] policy 'Six Years' update *> set duration=4
[user.hostname] policy 'Six Years' update *> commit
   Error: The replica retention policy "Six Years" could not be modified because the
target namespace "[user.hostname]-1" is locked.
[user.hostname] policy 'Six Years' update *> discard
[user.hostname] policy 'Six Years'> unapply
[user.hostname] policy 'Six Years' unapply *> set target=Namespace:/[user.hostname]-1
[user.hostname] policy 'Six Years' unapply *> commit
   Error: The replica retention policy "Six Years" could not be removed because the
target namespace "[user.hostname]-1" is locked.
[user.hostname] policy 'Six Years' unapply *> discard
[user.hostname] policy 'Six Years'>
```

11.2.3 Single engine continuous vault

This feature is available in versions 6.0.14.0 and above. CLI and UI functions are available for locking dSources.

11.2.3.1 Advantages

The Single Engine Continuous Vault provides effective protection against ransomware attacks in a standalone Delphix Engine. This option may be preferable for deployments where maintaining two separate engines is not architecturally necessary.

11.2.3.2 Implementation

This feature adds a “locked” property to sources. Once the locked property is enabled, the source cannot be removed. Locked sources are required to have a SnapSync policy defined that refreshes data at least once daily. Furthermore, an alert is raised if no new snapshot or log data is received in the last 12 hours. Upon request, Delphix Support can change this value.

To protect Continuous Vault application data from accidental deletion or malicious attack, snapshots of locked sources may not be manually deleted. These snapshots are managed by a retention policy that must be configured for locked sources. The retention policy duration can be modified as long as retention satisfies the minimum duration (100 days).

As with the Continuous Vault Replication implementation, the time configuration of a Single Engine Continuous Vault cannot be changed. This is to prevent attempts at bypassing retention policies in order to try and delete snapshots on the Continuous Vault. Also, the factory reset operation is forbidden when at least one locked source is present.

By default, locked sources are not required to have LogSync enabled. However, upon request, Delphix Support can configure an engine-wide setting that prevents LogSync from being disabled on a locked source and, optionally, requires LogSync to be enabled from the very beginning—before locking a source.

11.2.3.3 CLI functions

Locking a source.

```
sedv> source
sedv source> select src10
sedv source 'src10'> lock
sedv source 'src10' lock *> commit
sedv source 'src10'>
```

Verifying the locked status of a source.

```
sedv source 'src10'> ls
Properties
    type: OracleLinkedSource
    name: src10
```

```
    container: src10
    externalFilePath: (unset)
    linked: true
    locked: true     <-------------- LOCKED
    logCollectionEnabled: false
    operations:
...
```

Verify that a locked source cannot be disabled.

```
sedv source 'src10'> disable
sedv source 'src10' disable *> commit
   Error: The source "src10" cannot be disabled because it is locked.
  Action: Contact Delphix support.
sedv source 'src10' disable *> discard
```

Verify that source locking requires a SnapSync policy that refreshes data at least once daily.

```
# An insufficiently-frequent SnapSync policy: runs at 03:00 on Sundays
sedv policy 'snapsync_weekly'> ls
Properties
    type: SyncPolicy
    name: snapsync_weekly
    customized: false
    default: false
    effectiveType: DIRECT_APPLIED
    reference: POLICY_SYNC-7
    scheduleList:
        0:
            type: Schedule
            cronString: 0 0 3 ? * 1
            cutoffTime: 14400sec
    timezone:
        type: TimeZone
        id: America/New_York
        offset: 240
        offsetString: UTC -04:00

Operations
delete
update
apply
unapply


sedv> source
sedv source> select src10
sedv source 'src10'> lock
sedv source 'src10' lock *> commit
   Error: Insufficient or unrecognized day coverage in schedule that affects locked
sources.
```

```
  Action: Cover either all days of the month or all days of the week when locked
sources are affected. Check the documentation for examples.
```

Verify that source locking requires a retention policy that retains 100 days of snapshots.

```
# A one-month retention policy
sedv policy 'retention_one_month'> ls
Properties
    type: RetentionPolicy
    name: retention_one_month
    customized: false
    dataDuration: 1
    dataUnit: MONTH
    dayOfMonth: 1
    dayOfWeek: MONDAY
    dayOfYear: Jan 1
    default: false
    effectiveType: DIRECT_APPLIED
    logDuration: 1
    logUnit: MONTH
    numOfDaily: 0
    numOfMonthly: 0
    numOfWeekly: 0
    numOfYearly: 0
    reference: POLICY_RETENTION-8

Operations
delete
update
apply
unapply

sedv> source
sedv source> select src10
sedv source 'src10'> lock
sedv source 'src10' lock *> commit
   Error: The retention policy is less than the minimum "100" days required when
applied to locked sources.
  Action: Set retention parameters to preserve at least "100" days of data, and try
 again.
```

Verify that a snapshot from a locked source cannot be manually deleted.

```
sedv snapshot ''@2022-05-05T22:41:24.045Z''> delete
sedv snapshot ''@2022-05-05T22:41:24.045Z'' delete *> commit
   Error: The selected snapshot cannot be deleted because the source associated with
its container "Untitled/src10" is locked.
  Action: Wait for the snapshot to be automatically deleted based on its retention
policy, or Contact Delphix support.
```

11.2.4 Continuous vault alert system

In addition to the data-protection rules described in the previous sections, Continuous Vaults have a special alert (see page 784)system that notifies administrators about all events that can affect the ability to ingest and replicate locked data or even the ability to send such alerts.

There are two categories of Continuous Vault alerts: domain alerts and system alerts which are emailed to Engine Administrators (see page 685) and System administrators (see page 680), respectively. To receive alerts, an administrator must have an email address configured and SMTP (see page 817)must be enabled.

All Continuous Vault alerts are sent also via SNMP (see page 785), Syslog (see page 810)and Splunk (see page 835)/ Fluentd (see page 826)when those services are enabled and their configured severity levels allow for each alert level.

Continuous Vault domain alerts are generated for the following events:

1. Locking a source, replication specification or namespace (informational level).
2. All actions (see page 2707) on locked replication specifications and namespaces, such as changes in replication schedules, and all actions on locked sources and related objects that can affect them, such as changes in environment settings. All action alerts are audit-level alerts.
3. Deleting a Engine Administrator or changing their email address. These are warning-level alerts emailed to the addresses before the change takes effect.

Continuous Vault system alerts are generated for the following events:

1. Modifying, disabling or deleting services used for delivering alerts: SMTP, SNMP, Syslog and Splunk/ Fluentd. These are warning-level alerts sent using the service configurations before the change takes effect.
2. Creating or enabling such a service (informational level).
3. Deleting a System administrator or changing their email address. These are warning-level alerts emailed to the addresses before the change takes effect.

By default, system administrators are allowed to change these service configurations. Continuous Vaults only notify when those changes happen. However, upon request, Delphix Support can enable locking these services such that, once an administrator enables a service, that service cannot be changed (except for subsequent changes requested to Delphix Support).

11.3 Backup and recovery strategies for the Delphix engine

As a software virtual appliance, Delphix leverages features of the storage, hypervisor, and appliance infrastructure to provide for recovery in the event of failure. These topics walk through the process of evaluating requirements and defining a solution. This process depends on the requirements and features of the environment in which the Delphix Engine is deployed.

- Backup and recovery requirements (see page 2360)
- Backup solution implementation (see page 2361)
- Deployment architecture (see page 2363)
- Mapping requirements to solutions (see page 2365)

11.3.1 Backup and recovery requirements

This topic describes determining requirements for infrastructure failure modes and recovery.

Before devising a strategy, you must first have a set of requirements to evaluate possible solutions. What failures are you trying to protect against, and what are your recovery goals in the event of failure?

11.3.1.1 Failure points

Before devising a strategy, you must first have a set of requirements by which the resulting solution can be evaluated. What failures are you trying to protect against, and what are your recovery goals in the event of failure?

11.3.1.1.1 Physical server failure

The Delphix Engine runs within the VMware ESX hypervisor, which itself is running on a physical machine. Failure of that physical machine will affect the Delphix Engine, as well as any other virtual machines running on that server. The failure is isolated to that particular server and is not the result of a larger, site-wide failure.

- **Recommendation**: ESX Clustering

11.3.1.1.2 Storage failure

The Delphix Engine uses LUNs from a storage array provided through the VMware hypervisor. The storage array may have redundant disks and/or controllers to protect against single points of failure within the array. However, the Delphix Engine can still be affected by a failure of the entire array, the SAN path between the Delphix Engine and the array, or by a failure of the LUNs in the array that are assigned to the Delphix Engine.

- **Recommendation**: Replication

11.3.1.1.3 Site failure

When an entire site or datacenter goes down, all servers, storage, and infrastructure are lost. This will affect not only the Delphix Engine but any production databases and target servers in the datacenter.

- **Recommendation**: Replication

11.3.1.1.4 Administrative error

If an administrator mistakenly deletes a VDB or takes some other irreversible action, there is no method of recovery built into the Delphix Engine.

- **Recommendation**: Snapshots

11.3.1.2 Recovery objectives

Once infrastructure fails, some amount of work is required to restore the Delphix Engine to an operational state. Clients won't have access to the Delphix Engine during this time, and the point to which the system is recovered is dependent on the mechanism being used. These qualitative aspects of recovery can be captured by the following metrics. As these metrics are often directly associated with cost, it is important to think not just about the desired metrics, but also the minimum viable goals.

11.3.1.2.1 Recovery point objective (RPO)

The RPO is the acceptable amount of data that can be lost in the event of a failure. For example, if backups are taken once a day, then at most 24 hours of data will be lost if the system fails immediately before a regularly scheduled backup.

11.3.1.2.2 Recovery time objective (RTO)

The RTO is the time required to restore the system to an operational state after a failure. For example, recovery may require restoring data from a backup, followed by some number of manual steps to recreate the configuration in the new system. RTO is equivalent to the downtime experienced by clients.

11.3.1.2.3 Recovery time granularity (RTG)

The granularity of the recovery time is the specificity by which you can select a particular point in time from the past to restore the system. For example, VM snapshots taken every hour provide no way to restore to a point in time between those snapshots.

11.3.2 Backup solution implementation

This topic describes the tradeoffs involved with backup and recovery solutions.

With the exception of clustering, solutions can be implemented using features at both the storage and hypervisor layer. Choosing the right technology requires understanding both your requirements and what infrastructure is in use in your environment. The following sections outline some basic choices and the tradeoffs involved.

11.3.2.1 Clustering

VMware vSphere's high [732]availability provides the ability to have a VM configuration shared between multiple physical ESX servers. Once the storage has been configured on all physical servers, any server can run the Delphix Engine VM. This allows ESX clusters to survive physical server failure. In the event of failure, the VM is started on a different server and appears to clients as an unexpected reboot with non-zero but minimal downtime. Depending on the length of the outage, this may cause a short pause in I/O and database activity, but longer outages can trigger timeouts at the protocol and database layers that result in I/O and query errors. Such long outages are unlikely to occur in a properly configured environment.

732 http://www.vmware.com/products/vsphere/high-availability.html

Automatic detection of a failure in an HA environment does not work in all circumstances, and there are cases where the host, storage, or network can hang such that clients are deprived of access, but the systems continue to appear functional. In these cases, a manual failover of the systems may be required.

When configuring a cluster, it is important to provide standby infrastructure with equivalent resources and performance characteristics. Asymmetric performance capabilities can lead to poor performance in the event of a failover. In the worst case of an over-provisioned server, this can cause widespread workload failure and the inability to meet performance SLAs.

11.3.2.2 Snapshots

VMware provides storage-agnostic snapshots[733] that are managed through the VMware Snapshot Manager[734]. The use of VMware snapshots can, however, cause debilitating performance problems for write-heavy workloads due to the need to manage snapshot redo-log metadata. In order to provide an alternative snapshot implementation, while retaining the existing management infrastructure, VMware has created an API to allow storage vendors to supply their own snapshot implementation. This is only supported in ESX 5.1. Furthermore, the array must support the vStorage APIs[735]. Consult the VMware documentation[736] for supported storage solutions and the performance and management implications.

Storage-based snapshots, by virtue of being implemented natively in the storage array, typically do not suffer from such performance problems and are preferred over VMware snapshots when available. When managing storage-based snapshots, it is critical that all LUNs backing a single VM be part of the same consistency group. Consistency groups provide write order consistency across multiple LUNs and allow snapshots to be taken at the same point in time across the LUNs. This must include all VM configuration, system VMDKs, and VMDKs that hold the dSources and VDBs. Each storage vendor presents consistency groups in a different fashion; consult your storage vendor documentation for information on how to configure and manage snapshots across multiple LUNs.

In the event of a snapshot recovery becoming required, ensure that the Delphix Engine VM is powered off for the duration of the snapshot recovery. Failure to do so can lead to filesystem corruption as you're changing blocks underneath a running system.

11.3.2.3 Replication

Site Recovery Manager [737] (SRM) is a VMware product that provides replication and failover of virtual machines within a vSphere environment. It is primarily an orchestration framework, with the actual data replication performed by a native VMware implementation, or by the storage array through a storage replication adapter (SRA). A list of supported SRAs can be found in the VMware documentation[738]. There is some performance overhead in the native solution, but not of the same magnitude as the VMware snapshot impact. SRAs provide better performance but require that the same storage vendor be used as both source and target, and require resynchronization when migrating between storage vendors.

Storage-based replication can also be used in the absence of SRM, though this will require manual coordination when re-configuring and starting up VMs after failover. The VM configuration, as well as the storage configuration within ESX, will have to be recreated using the replicated storage.

The Delphix Engine also provides native replication within Delphix. This has the following benefits:

---

733 http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1015180
734 http://www.vmware.com/support/ws55/doc/ws_preserve_sshot_manager.html
735 http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1021976
736 http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1021976
737 http://www.vmware.com/products/site-recovery-manager/
738 http://www.vmware.com/support/srm/srm-storage-partners.html

- The target system is online and active
- VDBs can be provisioned on the target from replicated objects
- A subset of objects can be replicated
- On failover, the objects are started in a disabled state. This allows configuration to be adjusted to reflect the target environment prior to triggering policy-driven actions.
- Multiple sources can be replicated to a single target

Note that the Delphix Engine currently only replicates data objects (dSources and VDBs) and environments (source and target services). It does not replicate system configuration, such as users and policies. This provides more flexibility when mapping between disparate environments, but requires additional work when instantiating an identical copy of a system after failover.

11.3.2.4 Backup

There is a large ecosystem of storage and VM-based backup tools, each with its own particular advantages and limitations. VMware provides Data Protector[739], but there are size limitations [740] (linked to a maximum of 2TB of deduped data) that make it impractical for most Delphix Engine deployments. Most third-party backup products, such as Symantec NetBackup, EMC Networker, and IBM Tivoli Storage Manager, have solutions designed specifically for the backup of virtual machines. Because the Delphix Engine is packaged as an appliance, it is not possible to install third-party backup agents. However, any existing solution that can back up virtual machines without the need for an agent on the system should be applicable to Delphix as well. Check with your preferred backup vendor to understand what capabilities exist.

Some storage vendors also provide a native backup of LUNs. Backup at the storage layer reduces overhead by avoiding data movement across the network but loses some flexibility by not operating within the VMware infrastructure. For example, recreating the VM storage configuration from restored LUNs is a manual process when using storage-based recovery.

11.3.3 Deployment architecture

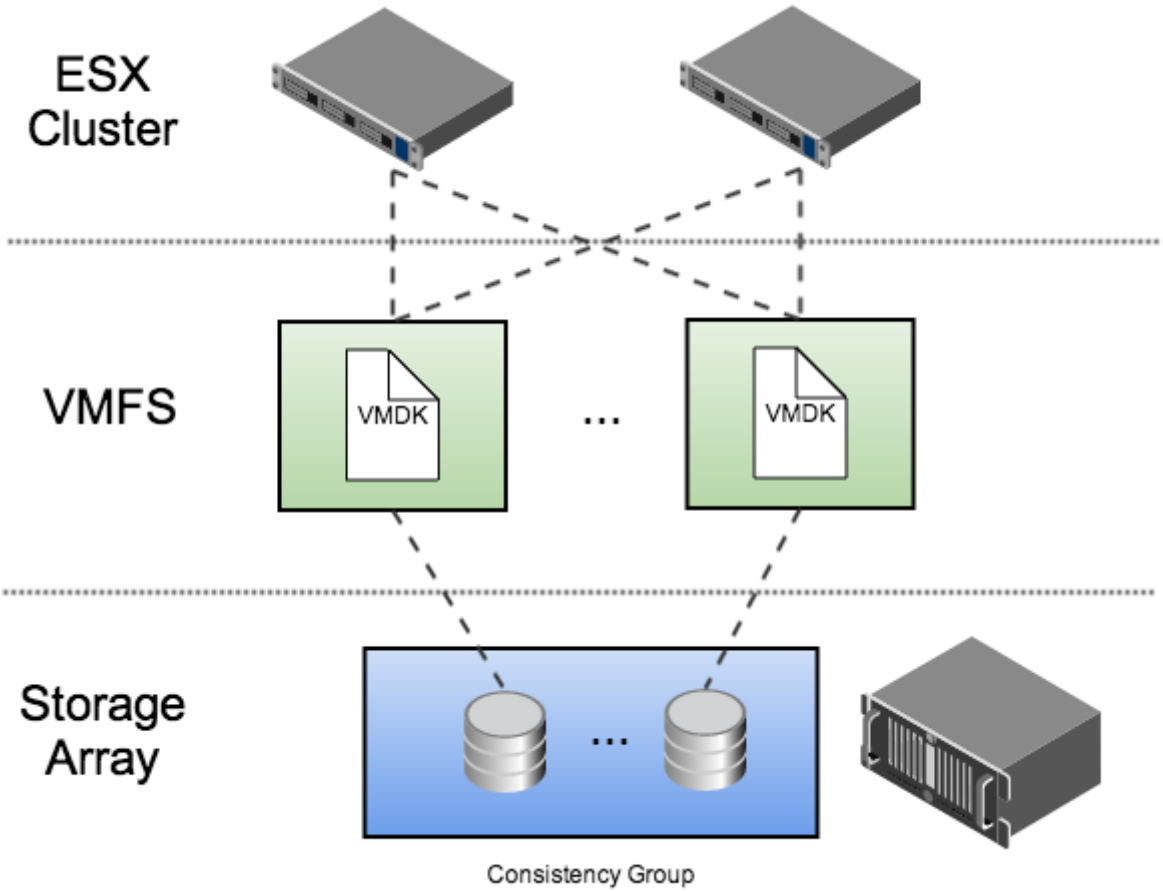
This topic describes components of the Delphix deployment architecture.

Delphix operates in a virtual environment with several core systems working in concert, each with its own set of capabilities. Understanding this architecture is critical in evaluating how solutions can be applied across the components, and the tradeoffs involved.

11.3.3.1 Architectural components

This diagram illustrates Delphix's recommended best practices for deploying the Delphix Engine in a VMware environment:

739 http://www.vmware.com/solutions/datacenter/business-continuity/data-protection.html
740 http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2016565



This architecture is designed to isolate I/O traffic to individual LUNs while using the most commonly deployed VMware components. In this example, each VMDK file is placed in a separate VMFS volume. Each volume is exported to every node in the ESX cluster, allowing the Delphix Engine to run on any physical host in the cluster.

11.3.3.2 Fault Recovery Features

Across the recommended deployment architecture there are three key components in play: Delphix Engine, VMware, and storage. Each of these provides different failure handling capabilities, which can be roughly grouped into the following areas.

11.3.3.2.1 Server Clustering

Clustering provides a standby server that can take over in the event of failure. A given clustering solution may or may not provide high availability guarantees, though all provide failover capabilities, provided that an identical passive system is available.

11.3.3.2.2 Snapshots

Snapshots preserve a point-in-time copy of data that can be used later for rollback or to create writable copies. Creating a snapshot is typically low cost in terms of space and time. Because they use the storage allocated to the array, snapshots restore quickly, but they do not protect against failures of the array.

11.3.3.2.3 Replication

Data replication works by sending a series of updates from one system to another in order to recreate the same data remotely. This stream can be synchronous, but due to performance considerations is typically asynchronous, where some data loss is acceptable. Replication has many of the same benefits of backup, in that the data is transferred to a different fault domain, but has superior recovery time given that the data is maintained within an online system. The main drawback of replication is that the data is always current - any logical data error in the primary system is also propagated to the remote target. The impact of such a failure is less when replication is combined with snapshots, as is often the case with continuous data protection (CDP) solutions.

11.3.3.2.4 Backup

Like snapshots, backup technologies preserve a point-in-time copy of a storage dataset, but then move that copy to offline storage. Depending on the system, both full and incremental backups may be supported, and the backup images may or may not be consistent. Backup has the advantage that the data itself is stored outside the original fault domain, but comes at high cost in terms of complexity, additional infrastructure, and recovery time.

11.3.4 Mapping requirements to solutions

This topic describes how to map from backup and recovery requirements to solutions.

With requirements and detailed knowledge of the deployment architecture, we can map to solutions tailored for the features provided by the underlying infrastructure.

11.3.4.1 Feature capabilities

Based on these failure points and recovery features, you can use the following table to map requirements to architectural components: VMware (V), Delphix (D), or storage (S). This can drive implementation based on infrastructure capabilities and recovery objectives.

	Fault Recovery Features			
Failure Point	**Clustering**	**Snapshots**	**Replication**	**Backup**
Server Failure	V	-	V S D	-
Storage Failure	-	-	V S D	V S

	Fault Recovery Features			
Site Failure	-	-	V S D	V S
Administrative Error	-	V S	-	V S

Recovery Point Objective (RPO)

Feature	Time	Description
Clustering	Zero	All changes committed to disk are automatically propagated to the new server. Any pending changes in memory are lost.
Replication	Near zero	Most solutions offer scheduled replication, but many can offer continuous replication with near-zero data loss.
Snapshots	Snapshot period (for example, one hour)	Given their relatively low cost, snapshots tend to be taken at a higher frequency than a traditional backup schedule.
Backup	Backup period (for example, one day)	Backup policies can be configured in a variety of ways, but even with incremental backups, most deployments operate no more frequently than once a day because of the cost of full backups, and the impact of incremental backups on recovery time.

11.3.4.1.1 Recovery time objective (RTO)

Feature	Time	Description
Clustering	Near zero	VM clustering with the Delphix Engine provides near-zero downtime in the event of failure, but clients may be briefly paused or interrupted.
Replication	15 minutes	The target side environment is kept in hot standby mode, so it is relatively quick to switch over to the target environment. Depending on the scope of the failure, however, some configuration information may need to be changed on the target side prior to enabling operation.

Feature	Time	Description
Snapshots	15 minutes	The Delphix Engine can be rolled back to a previous state. Changes made to systems external to the Delphix Engine (for example, deleting a VDB) can cause inconsistencies after rollback.
Backup	Hours or days	Restoring a full backup can be very time-consuming. In addition to having to read, transfer, and write all of the data, the same process will need to be run for each incremental backup to reach the objective point.

11.3.4.1.2 Recovery time granularity

Feature	Granularity	Description
Clustering	None	Only the current system state can be recovered.
Replication	None	Only the nearest replicated state can be recovered unless combined with snapshots.
Snapshots	Snapshot period (for example, one hour)	Determined by the snapshot schedule.
Backup	Backup period (for example, one day)	Determined by the backup schedule.

11.4 Delphix replication

This section covers the following topics:

- Replication overview (see page 2368)
- Forward Compatible Replication (FCR) (see page 2371)
- Replication concepts (see page 2375)
- Replication use cases (see page 2377)
- Configuring replication (see page 2381)
- Controlled failover (see page 2385)
- Uncontrolled failover (see page 2387)
- Managing replicated objects (see page 2395)

- Replicas and failover (see page 2397)
- Test-failover and failback (see page 2400)
- Provisioning from replicated data sources or VDBs (see page 2403)
- Replication user interface (see page 2403)

11.4.1 Replication overview

Delphix facilitates the replication of data objects across different Delphix Engines. The Delphix Engines can be running different Delphix versions, but with some limitations to older versions, as mentioned below.

11.4.1.1 Replication features

- **Data recovery and distribution**: If the source engine fails, the target engine can be activated to mirror the state of the source. This feature also enables geographical data distribution and remote VDB provisioning from replicated objects.
- **Replication scheduling**: While ad hoc replication is possible, it is commonly executed on a preset schedule. Only the changes made since the last update are transmitted in subsequent updates.
- **Versatile data management**: Delphix Engines (functioning as virtual appliances) offer robust data management capabilities that include backing up, restoring, replicating, and migrating various data objects such as groups, dSources, VDBs, Jet Stream data templates and containers, along with their dependencies.
- **Enhanced capabilities**: The Continuous Data Engine's native replication adds advanced options to include selective replication of data objects, consolidating multiple sources to a single target, and the ability to provision VDBs from replicated dSources and VDBs without interrupting updates.
- **Flexible configuration**: Replication is set up on the source engine, enabling the transfer of selected dSources and VDBs to a target engine. This process supports both manual and scheduled incremental updates. Detailed guidance for setting up replication can be found in the Configuring replication[741] page.
- **Provisioning from replicated data**: The replicated dSources and VDBs on the target engine can be used to create new VDBs. These can be refreshed with data from incremental replication updates, provided the original source objects remain intact. For more insights, refer to the Provisioning from replicated dSources or VDBs[742] page.
- **Disaster recovery and activation**: During replication, the replicated objects are stored in an inactive state on the target engine. In case of a disaster, a failover operation can be initiated to break the replication link and activate these objects. More information on this process is available under Replicas and failover[743].

741 https://cd.delphix.com/docs/19.0.0.0/configuring-replication
742 https://cd.delphix.com/docs/19.0.0.0/provisioning-from-replicated-data-sources-or-vdbs
743 https://cd.delphix.com/docs/19.0.0.0/replicas-and-failover

11.4.1.1.1 Limitations

- **No synchronous semantics**: Replication in Delphix does not offer synchronous semantics. This means there is a potential loss of data on the target engine, equivalent to the changes made since the last replication update, in the event of a failover.
- **Not ideal for rapid failover and failback**: In high-availability scenarios, it is advisable to utilize the capabilities of the underlying hypervisor or storage platform.
- **Older engine version compatibility**: Engine version 5.3.3 or above is required to allow for replication between engines running different Delphix versions, as elaborated in the Forward Compatible Replication (FCR) section.

Further guidance

For a detailed understanding of how Delphix Engine replication aligns with your data recovery needs, refer to the Backup and recovery strategies for the Delphix Engine. (see page 2359)

11.4.1.2 Replication details

When you select objects for replication, the engine will automatically include any dependencies, including parent objects, such as groups, and data dependencies such as VDB sources. This means that replicating a VDB will automatically include its group, the parent dSource, and the group of the dSource, as well as any environments associated with those databases. When replicating an entire engine, all environments will be included. When replicating a database or group, only those environments with the replicated databases are included.

Only database objects and their dependencies, as well as certain non-database objects, are copied as part of a backup or replication operation, including:

- dSources
- VDBs
- Groups
- Self-service (Jet Stream) Data Templates and Data Containers
- Environments
- Environment configuration (users, database instances, and installations)
- Delphix users, roles, permissions, and authorizations
- Policies
- Database configuration templates

The following objects are NOT copied as part of a backup or replication operation:

- Events and faults
- Job history
- System services settings, such as SMTP
- Hook templates
- Alert profiles

After failover, you must recreate these settings on the target.

On-premises Replication to Azure/OCI/GCP/Hyper-V
Replicating from on-premises engines with an underlying storage block size of 512B will experience disk usage inflation when replicating to target engines with different underlying block sizes.
Azure, GCP, Hyper-V, and OCI are known to have 4K block sizes and therefore will require extra disk capacity when receiving replication from an on-premises engine. This behavior is due to the fact that the underlying storage block size is different (512B vs. 4K) between the two Delphix Engines (one on-prem, one on Azure/OCI/GCP/Hyper-V), resulting in a lower compression rate on the replication target.
It is expected that 1.5-1.6x the amount of space is taken from objects on-premises in these cases.

11.4.1.3 Resumable replication

This feature allows for the continuation of large and time-intensive initial replications or incremental updates from a midpoint, rather than starting anew. This is crucial given that replications can fail due to various environmental or internal factors.

11.4.1.3.1 Real-world scenario

Imagine a scenario where a replication is set up between a source and a target, projected to take several weeks. If a power outage at the source's data center interrupts this process, the source machine, upon rebooting, will automatically reconnect to the target and resume replication from where it was interrupted. In the user interface (UI), the source will continue to show the ongoing replication job, while the target will display a new replication reception job, tracking the overall progress.

11.4.1.3.2 Applicability of resumable replication

Resumable replication is effectively utilized during source reboots, target reboots, and network partitions. It ensures the continuity and efficiency of the replication process in the face of unforeseen disruptions.

11.4.1.4 Replicating Delphix Self-Service templates

Templates can be replicated and accessed on the target engine via Delphix Self-Service (Jet Stream). Replicated templates can be replicated into the target space with or without their containers. On the new target engine, the newly created replicated template can be used to create new containers that are assigned to users. You cannot change the replicated template’s name or the names of the containers from which it was replicated.

Any containers that were replicated over with the template cannot be used to start, stop, etc until they are disconnected to their parent containers in the source engine during the failover operation.

11.4.1.5 Replication of non-data objects

As of 6.0.5.0, the replication of non-data objects is supported.

Non-data objects refer to:

- Delphix users, roles, permissions, and authorizations
- Policies
- Database configuration templates

These objects will not be presented as selectable objects when creating a replication spec, but will instead be passively included by association, the same way environments are. Replication of non-data objects will follow these rules:

- If the entire engine is replicated, all non-data objects will be included
- If specific data objects are replicated, all associated non-data objects will be included

For example, when a container is replicated, then policies that apply to that container, as well as users who have authorizations on that container, will all be included for replication.

Replicated users will be shown on the Received Replicas page on the Delphix Engine UI.

11.4.1.5.1 Automatic conflict resolution

Also as of 6.0.5.0, the automatic conflict resolution[744] option is chosen by default; it is the recommended way of handling the failover of non-data objects, as non-data objects are expected to cause collisions. Starting with version 6.0.17.0, automatic conflict resolution is always enabled and is no longer optional.

Automatic conflict resolution will resolve non-data object conflicts by the following rules:

- Users will be consolidated if both username and email match. Otherwise, the replicated user will be renamed.
- Roles, permissions, and authorizations will be consolidated
- Policies and database configuration templates will be renamed, except for "None" type policies, which will be consolidated. Following failover, replicated policies will continue to apply to the same replicated data objects.

11.4.2 Forward Compatible Replication (FCR)

11.4.2.1 Overview

The Delphix Continuous Data Engine supports the ability to replicate to a Delphix Target Engine running on a higher version. To do so, FCR has a few requirements and restrictions to consider:

- FCR is supported for replication jobs starting from a **source engine** running 5.3.0.0 and beyond.
- The **target engine** must be running 5.3.3.0 and beyond.
- As of 6.0.10.0, the FCR replication operation can not go beyond engine versions released more than 12 months apart.

744 https://documentation.delphix.com/continuous-data-11-0-0-0/docs/replicas-and-failover

- FCR supports replicating between major versions as long as those specific versions are no more than a year apart.

Examples of supported and not supported FCR configurations:

11.4.2.1.1 Supported

- 6.0.3.0 to 6.0.3.0 (same version)
- 6.0.3.1 to 6.0.3.1 (same version)
- 6.0.3.0 to 6.0.3.1 (higher patch version)
- 6.0.3.1 to 6.0.5.0
- 6.0.3.0 to 6.0.9.0
- 6.0.4.0 to 6.0.10.0

11.4.2.1.2 Not supported

- 5.2.5.0 to 6.4.0.0 (source version not compatible with FCR)
- 5.3.0.0 to 5.3.2.0 (target version not compatible with FCR)
- 6.0.9.0 to 6.0.8.0 (source version higher than the target)
- 6.0.0.0 to 6.0.10.0 (there is more than a year between release dates for those versions)

11.4.2.1.3 Exceptions

Some of the newer versions of 5.3.x are not compatible with the early versions of 6.0.x.0 (6.0.0.0 and 6.0.1.0).

- 5.3.7.0 and 5.3.7.1 are not compatible with 6.0.0.0
- 5.3.8.0 and 5.3.8.1 are not compatible with 6.0.0.0
- 5.3.9.0 is not compatible with 6.0.0.0
- 5.3.9.0 is not compatible with 6.0.1.0 and 6.0.1.1

11.4.2.2 Replication matrix

The following table lists all the Delphix Engine versions that a user can replicate to the required or the latest version.

When replicating from X.Y.X.* to X.Y.Z.* the target version has to be greater than or equal to the source version. The * convention is used to refer to patch releases and reduce the table size.

Replication supported	
From source version	**To target version**
5.3.0.*	5.3.0.* 5.3.3.* - 6.0.9.*

Replication supported	
5.3.2.*	5.3.2.* - 6.0.9.*
5.3.3.*	5.3.3.* - 6.0.9.*
5.3.4.*	5.3.4.* - 6.0.9.*
5.3.5.*	5.3.5.* - 6.0.9.*
5.3.6.*	5.3.6.* - 6.0.9.*
5.3.7.*	5.3.7.* - 5.3.9.* 6.0.1.* - 6.0.9.*
5.3.8.*	5.3.8.* - 5.3.9.* 6.0.1.* - 6.0.9.*
5.3.9.*	5.3.9.* 6.0.2.* - 6.0.9.*
6.0.0.*	6.0.0.* - 6.0.7.*
6.0.1.*	6.0.1.* - 6.0.7.*
6.0.2.*	6.0.2.* - 6.0.7.*
6.0.3.*	6.0.3.* - 6.0.7.*
6.0.4.*	6.0.4.* - 6.0.7.*
6.0.5.*	6.0.5.* - 6.0.11.*
6.0.6.*	6.0.6.* - 6.0.12.*
6.0.7.*	6.0.7.* - 6.0.13.*
6.0.8.*	6.0.8.* - 6.0.14.*

Replication supported	
6.0.9.*	6.0.9.* - 6.0.15.*
6.0.10.*	6.0.10.* - 6.0.16.*
6.0.11.*	6.0.11.* - 6.0.17.*
6.0.12.*	6.0.12.* - 7.0.0.*
6.0.13.*	6.0.13.* - 9.0.0.*
6.0.14.*	6.0.14.* - 11.0.0.*
6.0.15.*	6.0.15.* - 13.0.0.*
6.0.16.*	6.0.16.* - 15.0.0.*
6.0.17.*	6.0.17.* - 17.0.0.*
7.0.0.*	7.0.0.* - 19.0.0.*
8.0.0.*	8.0.0.* - 20.0.0.*
9.0.0.*	9.0.0.* - 21.0.0.*
10.0.0.*	10.0.0.* - 22.0.0.*
11.0.0.*	11.0.0.* - 23.0.0.*
12.0.0.*	12.0.0.* - 23.0.0.*
13.0.0.*	13.0.0.* - 23.0.0.*
14.0.0.*	14.0.0.* - 23.0.0.*
15.0.0.*	15.0.0.* - 23.0.0.*

Replication supported	
16.0.0.*	16.0.0.* - 23.0.0.*
17.0.0.*	17.0.0.* - 23.0.0.*
18.0.0.*	18.0.0.* - 23.0.0.*
19.0.0.*	19.0.0.* - 23.0.0.*
20.0.0.*	20.0.0.* - 23.0.0.*
21.0.0.*	21.0.0.* - 23.0.0.*
22.0.0.*	22.0.0.* - 23.0.0.*
23.0.0.*	23.0.0.*

11.4.3 Replication concepts

Delphix replication allows you to copy objects from one Engine (referred to as a source Engine) to another Engine (referred to as a target Engine).

Replication recreates objects from the source Engine onto the target Engine in a **replica**, also known as a **namespace**, that preserves object relationships and naming on the target Engine without interfering with its active objects. Objects within a replica are read-only and disabled until the replica is failed over, at which point they can be activated. VDBs and dSources within a replica can be used as the source for provisioning new VDBs. Below are key concepts for replication that will be explained in detail:

- **Received Replicas or Namespaces**: Once replication is complete, the target Engine will create a received replica, also known as a namespace. This is a copy of objects that are related via a grouping.
- **Failover and Conflict Resolution**: Certain objects may require changes to resolve conflicts prior to completing a replication failover. Names of replicated objects should be unique for replication to complete successfully
- **Enabling Databases and Environments**: Once replication takes place, you must enable the databases and environments on the target engine. This step ensures the configuration of these objects is correct and will ensure that replication has correctly copied the necessary object relationships from the source engine.

11.4.3.1 Received replicas or namespaces

A replica contains a set of replicated objects, such as dSources and VDBs. These objects are read-only and disabled while replication is ongoing. You may view replicated objects both in the Delphix Engine UI as well as the CLI. To view received replicas:

1. On the target Engine, navigate to 'System'. Then select Replication.
2. Under Received Replicas, select the replica.

On this screen, you can browse the contents of replicas, as well as failover or delete individual replicas.

Deleting or failing over a replica will break any link with the replication source. Subsequent incremental updates will fail, requiring the source to re-establish replication. Failover should only be triggered when no further updates from the source are possible, as in a disaster scenario. Various replication use cases are also described in Replication Use Cases. (see page 2377)

Multiple replicas can exist on the system at the same time. Active objects can exist in the system alongside replicas without interfering with replication updates. You can also use VDBs and dSources within a replica as a source when provisioning. For more information, see Provisioning from Replicated Data Sources or VDBs. (see page 2403)

11.4.3.2 Failover and conflict resolution

To activate the objects in a replica, you must first fail over the replica. This will disconnect the replication connection and move the objects to the live system, where they can be actively used.

During the replication failover, there may be objects that conflict between the source Engine and the target Engine. For example, 'groups' will conflict if the failing over group has the same name as a group in the live system, as well as environments, dSources, and VDBs. By default, active objects with conflicting names will cause an error at the time of failover. In these scenarios, you must rename the active objects before the failover operation can complete successfully.

Given that conflicting names prevent failover from succeeding, best practices in a disaster recovery situation are to leave the target system completely passive with no active objects until the time that failover is required.

Once a replica is failed over, the objects are active but will be automatically disabled. The next section describes enabling these objects after replication is complete.

11.4.3.3 Enabling databases and environments

Objects may refer to states (IP addresses, mount paths, etc.) that differ between the source and target Engines. Because of this, all databases and objects within a replica are automatically disabled after a failover. This allows the administrator to alter configuration prior to enabling databases and environments, without the system inadvertently connecting to invalid systems.

After failover is complete, you must explicitly enable all dSources, VDBs, and environments. If you need to change any configuration for the target environment, you can do so prior to enabling the objects. In the event that a failing-over environment is consolidated with a live system environment, it must be refreshed before all of its databases can be used.

11.4.4 Replication use cases

Replication allows you to move Delphix objects such as dSources and virtual databases (VDBs) between Delphix Engines. These topics describe how you can use replication to meet different use cases:

- **Replicating to the Public Cloud**: With Delphix replication you can send data from engines deployed on-premise to the public cloud. In this case, you may have a Delphix engine in the production zone to ingest data securely with replication set up to the public cloud for development access and rapid, on-demand testing.
- **Disaster Recovery**: In the event of a disaster, you may need to failover your engine. Delphix replication enables you to configure a failover Delphix Engine to preserve the data and Delphix objects from the source engine for disaster recovery.
- **Geographically Distributed Development**: Often, development teams access data from all over the world. In this scenario, you may want to replicate data to be local to the developers who require access. With Delphix replication, you can provide developers with local access to Delphix datasets.
- **Data Migration**: Delphix supports simple migration of data and resources between Delphix Engines. There may be cases when you need to consolidate workloads between different Delphix Engines. With replication, you can easily migrate data as needed.

With Delphix replication, you can achieve the ideal Delphix deployment:

1. Ingesting data in production with a Delphix Engine deployed within the production zone.
2. Masking the data in production using Delphix Masking.[745]
3. Replicating the masked data to a non-production environment, using Selective Data Distribution (SDD). SDD enables you to securely replicate masked data without compromising sensitive data from the source engine. For more information, read Selective Data Distribution Overview. (see page 2408)

For more information on a few examples of Replication use cases, view the sections below.

11.4.4.1 Replicating to the public cloud

Enterprises will usually have the separate infrastructure for production systems and non-production development environments. For example, in the hybrid-cloud model, on-premise data centers are used to allow the company maximum ownership of these systems, while the public cloud is leveraged by development and automated testing teams to accelerate software development.

11.4.4.2 Delphix engine deployed in the public cloud

- Once a Delphix Engine has been deployed in the public cloud (using a supported cloud platform as described in Deployment (see page 556)) you can begin replicating from any source engine to that target engine.
- This enables you to provide access to production data from a Delphix Engine deployed on-premise to a Delphine Engine deployed in the cloud platform of your choice.

745 https://delphixdocs.atlassian.net/wiki/spaces/CC/pages/9962775/Introduction+to+Delphix+Masking

11.4.4.3 Disaster recovery

Replication is often used to provide recovery in the event of a disaster, where a data center or site becomes completely unavailable. To prepare for this, you may configure a failover Delphix Engine, which we will refer to as the *replication target Engine.* This target Engine will regularly receive replication updates from the original, or source, Engine so that if the source ever becomes unavailable, the target can be activated immediately.

11.4.4.4 Configuration steps

- **Passive target engine**
  - For disaster recovery, the target engine should be kept in a passive state until the source engine is lost. This prevents failover conflicts that may occur during the replication process.
  - At this point, a failover is performed that breaks subsequent replication updates and activates objects so that you can manage them on the target side.
- **Same configuration for source and target engines**
  - Target hosts and systems should exist at the target that matches the configuration of those at the primary engine. For example, if the source engine has two Red Hat environments discovered with four Oracle databases, the target engine should have exactly two Red Hat environments and four Oracle databases as well.
  - The failover engine should be provisioned with identical resources as the primary engine. For example, both engines should have the same number and types of disks attached as storage.
  - Finally, both engines should have the same network and storage topologies.

11.4.4.5 Failover object management

Once a failover occurs, there are two scenarios that will affect how you manage replicated objects, which include dSources, VDBs, and Environments.

- **Failure of Infrastructure Running the Delphix Engine Only**
  - You can enable dSources and VDBs and reconnect to the original environments that existed on the Source Engine.
  - You can reconfigure environments on the target Engine prior to failover as well.
- **Failure of Infrastructure Running the Delphix Engine and Delphix-connected Environments**
  - Environments will then have to be adjusted to point to the new systems on the target side.
  - If there is not a 1:1 mapping, then you can migrate the VDBs to new systems on the target, and you can detach dSources and attach them to the standby system in the target environment.

Follow the best practices below to simplify failover and meet performance expectations in the event of a disaster:

- To the extent possible, the failover Engine should mirror the primary Engine

    - Maintain a 1:1 relationship between source and targets. All data-related objects such as dSources, VDBs, and Environments as well as their configurations such as users and policies.
- The target Engine should remain passive and not be actively used for other workloads

11.4.4.6 Geographically distributed development

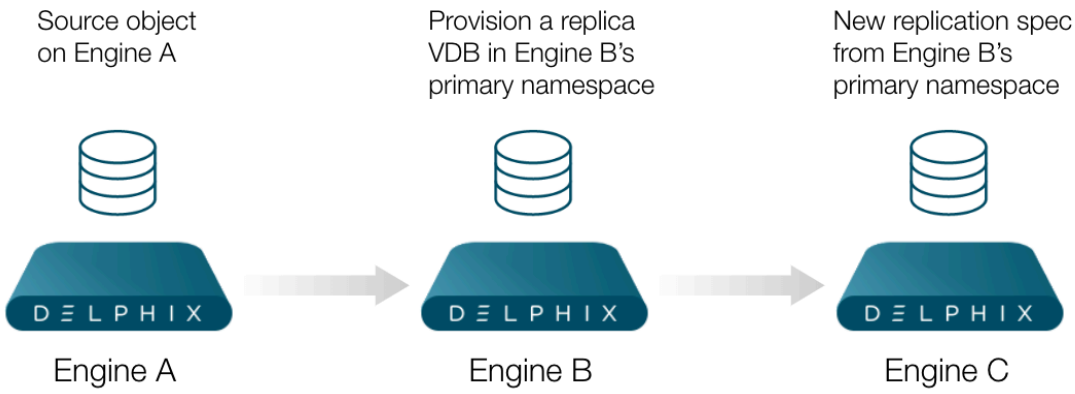
Organizations often have development teams distributed across different networks as well as different geographical locations. To improve performance or even meet security requirements, it may be necessary to replicate data from one location to another. The Delphix Engine allows for VDBs to be provisioned from replicated dSources and VDBs, as described in Provisioning from Replicated Data Sources or VDBs. (see page 2403)

This use case differs from **Disaster Recovery** as replication is never broken and failover is never performed. You can refresh remote VDBs as long as the parent objects continue to exist on the source. If they are deleted, then remote VDBs will continue to exist but cannot be refreshed or updated from their original source VDB.

11.4.4.6.1 Configuration steps

- **Avoid Heavy VDB Workloads on the Source**: Because each replication stream induces load on the source system:
    - Minimize the number of simultaneous replication updates. Each source engine can support replicating to multiple target engines, but you should try to keep simultaneous updates to under three target engines per source.
    - If possible, avoid heavy VDB workloads on the Source Engine
- **The permanence of Source Objects:** Provision only from sources that are effectively permanent, since replicated VDBs cannot be refreshed once the source is deleted. If you delete a source object you will need to re-replicate the VDBs if they need to be refreshed.
- **Additional Storage Capacity on the Target Engine:** Provision additional storage capacity on the target Engine
    - Remotely provisioned VDBs can consume shared storage (via NFS mounts) on the target even when the parent is deleted on the source

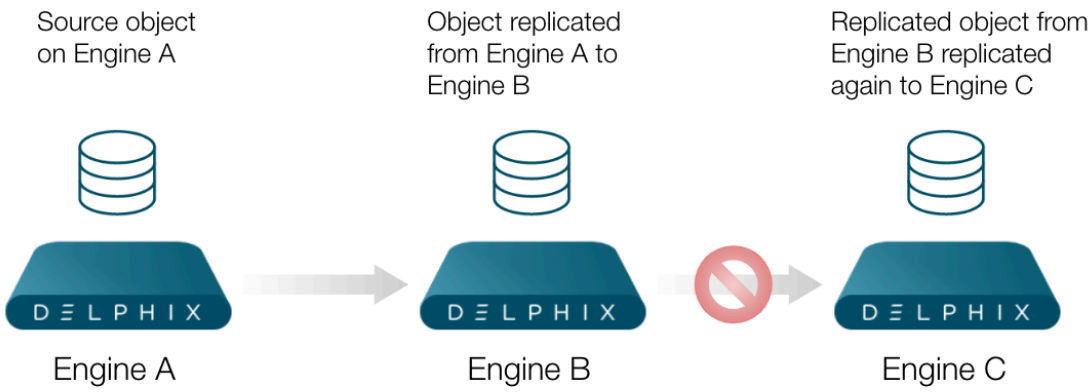
This use case supports more complex models such as 1-to-many and many-to-1. However, replication can only replicate objects that exist in the primary namespace. Consider a Delphix deployment with three Engines: Engine A, Engine B, and Engine C. The following workflow **is supported**:



1. **Engine A Replicates to Engine B**
2. **Provision a Replica VDB on Engine B**
3. **Engine B Replicates to Engine C**

In this scenario, you can replicate from Engine A to Engine B. Then, on Engine B provision a replica VDB into the primary namespace. Finally, add that object into a replication specification to replicate to Engine C. **Note**: Engine B can only replicate objects that exist in the primary (live) namespace to Engine C.

It is important to note the interim step of provisioning an on Engine B is required. For example, the following usage **is not allowed**:



1. **Engine A Replicates to Engine B**
2. **Engine B Replicates to Engine C**

You can replicate from Engine A to Engine B. But on Engine B, it is not possible, nor supported, to create a replication specification with objects in the replication namespace that are desired to be replicated to Engine C.

11.4.4.7 Data migration

You can use replication to perform a one-time migration of resources, such as virtual databases or environments, from one Delphix Engine to another. While the hypervisor provides tools to move virtual appliances between physical systems, there are times when migration is necessary, such as:

- Migrating between different physical storage
- Consolidating or distributing workloads across Delphix Engines

In these cases, you can use replication to copy a subset of objects across different topologies.

For migration, follow these best practices:

- Send full updates, followed by incremental updates, until the time required for incremental updates meets your downtime window
- Disable all objects to be migrated on the source, to ensure that they are not actively changing
- Send a final incremental update before failing over the target Engine
- After failover, delete any migrated objects on the source, or the entire Engine

11.4.5 Configuring replication

This topic describes how to configure data replication between Delphix Engines. Replication is configured with *Replication Profiles* that contain options such as the replication schedule, the hostname of the target Engine, and the selected objects that will be replicated.

11.4.5.1 Requirements

- **Version requirements**: the replication target can be on the same or newer version than the replication source.
- **Engine communication**: The target Delphix Engine must be reachable from the source Engine.
- **Storage allocation**: The target Delphix Engine must have sufficient free storage to receive the replicated data.
- **Privilege requirements**: The user in the replication profile must have administrative privileges on the source and the target engines.

11.4.5.2 Configuring the network

Delphix Replication uses a private network protocol to communicate between two Delphix Engines. You may specify a network interface to run replication by configuring routing to direct traffic over a particular interface.

The replication network protocol uses TCP port 8415. If there is a firewall between the source and target that is blocking this port, then there are two possible solutions:

1. Enable port 8415 on the firewall in order to allow connections to this port from the source to the target.

2. Replication can connect through a SOCKS proxy if one exists. Configure the SOCKS proxy address and port by connecting to the command-line interface (CLI) as a system administrator user and navigating to "service proxy" to update the SOCKS configuration.

> Port 1080
> SOCKS port 1080 is used by default but can be overridden

Replication can connect through a SOCKS proxy if one exists. Configure the SOCKS proxy address and port by connecting to the command-line interface (CLI) as a system administrator user and navigating to "service proxy" to update the SOCKS configuration. Example:

Example of a SOCKS Proxy

```
dlpx-engine> service proxy
dlpx-engine service proxy> update
dlpx-engine service proxy update *> set socks.enabled=true
dlpx-engine service proxy update *> set socks.host=10.2.3.4
dlpx-engine service proxy update *> set socks.username=someuser
dlpx-engine service proxy update *> set socks.password=somepassword
dlpx-engine service proxy update *> commit
dlpx-engine service proxy> get
    type: ProxyService
    https:
        type: ProxyConfiguration
        enabled: false
        host: (unset)
        password: (unset)
        port: 8080
        username: (unset)
    socks:
        type: ProxyConfiguration
        enabled: true
        host: 10.2.3.4
        password: ********
        port: 1080
        username: someuser
```

11.4.5.3 Configuring the replication source Delphix engine

1. On the source Delphix Engine, click **System**, then **Replication**.
2. In the left-hand navigation section, click **Create Profile**.
3. Enter the following required fields:
    a. Name of the replication profile
    b. The hostname or IP address for the target Delphix Engine.

11.4.5.4 Replication profile options

There are several configuration options for your replication profiles. These give you more granular control on options such as when replication will run, how much bandwidth it may use, and which objects are replicated. Details for each option are described below.

> The following configurable options are static and can not be configured at run-time. You can set these configurations while a replication spec is being executed, but the values will be applied only after the next execution.

- **Automatic replication**: A policy to automatically run replication. With this option, you can set up replication based on the schedule you need. Automatic replication allows you to define a policy to automatically run replication. By default, automatic replication is disabled, meaning that you must trigger replication updates manually. To enable automatic replication, click the Enabled checkbox. With this setting, you can enter the frequency and time for replication updates to the target Delphix Engine. Automatic replication uses Quartz, a job scheduling tool (http://www.quartz-scheduler.org/), for scheduling, which can be configured via the Advanced option.
- **Traffic Options**: Various traffic and bandwidth options are available. For example, you may want to enable encrypted traffic or limited bandwidth during replication updates.
  - **Encrypting traffic:** By default, replication streams are unencrypted, which provides maximum performance on a secure network. However, this setting allows you to encrypt traffic during replication.

    Note:
    Encrypting Replication Encrypting the replication stream will consume additional CPU resources and may limit the maximum bandwidth that can be achieved. During replication, the Delphix Engine will negotiate an SSL connection with its server peer to use TLS_AES_128_GCM_SHA256 as the cipher suite and TLSv1.3 as the protocol.
  - **Network connections**: Allows setting the number of underlying network connections that can be used by replication.
  - **Limiting bandwidth:** By default, replication will run at the maximum speed permitted by the underlying infrastructure. In some cases, particularly when a shared network is being used, replication can increase resource contention and may impact the performance of other operations. This option allows administrators to specify the maximum bandwidth that replication can consume.
- **Objects Being replicated**: Select the objects you wish to be replicated from the source engine to the target engine. In the right-hand column, under Objects Being Replicated, you can select the objects you want to replicate. Some selected objects may have dependencies – other objects that will be pulled into replication because they share data.

- This is not guaranteed to be the full set of dependent objects. The full set of objects and their dependents will be calculated at the time of replication.
- You can not configure a Replication Profile without selecting an object and the last object from a Replication Profile can not be removed. If you need to remove the last object from the Replication Profile, you must delete that Replication Profile.
- The object is removed from the replication target namespace only after a subsequent replication job is executed for the associated replication specification.
    a. Remove the object from the specification on the source
    b. Execute the replication specification
    c. The object is removed as a part of the replication job. The object is not removed while modifying the replication specification. To add back the object, the entire object needs to be sent again.

1. When replicating a group, all dSources and VDBs currently in the group, or added to the group at a later time, will be included.
2. If you select a Delphix Self-Service data template, all data containers created from that template will be included. Likewise, if you select a data container, its parent data template will be included.
3. Regardless of whether you select a VDB individually or as part of a group, the parent dSource or VDB (and any parents in its lineage) are automatically included.
    a. This is required because VDBs share data with their parent object.
    b. In addition, any environments containing database instances used as part of a replicated dSource or VDB are included as well.
4. When replicating individual VDBs, only those database instances and repositories required to represent the replicated VDBs are included. Other database instances that may be part of the environment, such as those for other VDBs, are not included.
5. Non-data objects (Delphix users, roles, permissions, authorizations, policies, database configuration templates) that are associated with selected objects will be automatically included during replication. They will not be shown on the selection interface.

11.4.5.5 Configuring the target Delphix engine

Additional configuration on the target engine is not needed. Replicated objects will appear in an alternate received replica (or namespace) that mirrors the original object layout.

To view replicated objects from the Delphix Engine:

1. Click System, then select Replication.
2. Look under Received Replicas. All replicated objects are read-only until the replica is failed over. For more information about managing replicas and how to activate a replica, see the topics Replicas and Failover (see page 2397) and Controlled Failover. (see page 2385)

You can create and manage objects on the target server without affecting subsequent updates, though this can cause conflicts on failover that require additional time to resolve. For disaster recovery use cases, it is recommended to keep the target passive and not create any local objects. This will avoid conflicts and guarantee a smooth failover operation.

Multiple sources can replicate to the same target, allowing for the flexible geographical distribution of data. This is not a recommended practice for disaster recovery, because it increases the probability of conflicts on failover and may oversubscribe resources on the target if multiple replicas are failed over and there is insufficient infrastructure to support the combined workloads.

11.4.6 Controlled failover

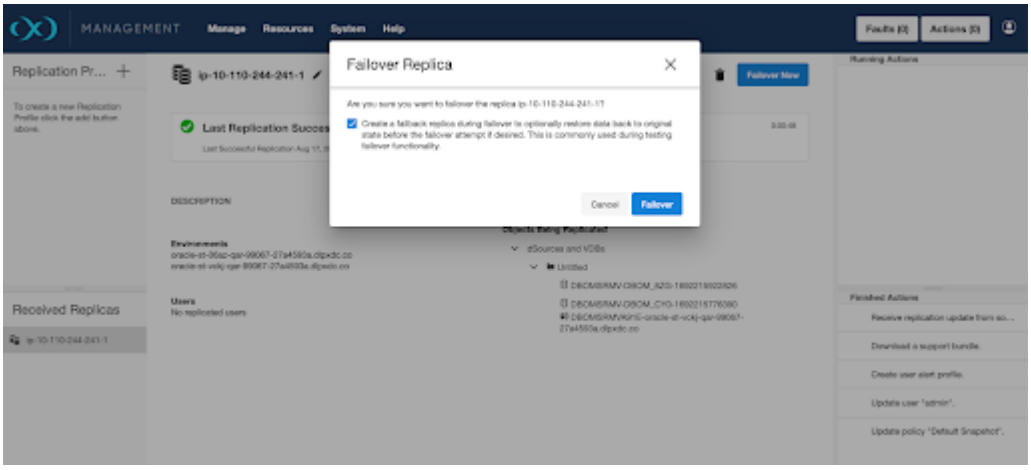
This topic describes the process of failing over a replica. Objects stored in a replica are read-only, and failing over a replica moves the replicated objects to the live system. After a failover, all of the objects will appear in the system as if they had been created locally.

11.4.6.1 Prerequisites

- A Delphix system that contains a replica is required.
- For an overview of what replicas are and what failover implies, see Replicas and Failover. (see page 2397)
- For more information on configuring replication, refer to the Configuring Replication (see page 2381) topics.

11.4.6.2 Procedure

1. Locate the replica to failover.
   a. Click **System**.
   b. Select **Replication**.
   In the **Received Replicas** section, you will see the list of replicas. Each replica has a default name which is the hostname of the source that sent the update. If you wish, you can customize these names. Each replica will list the databases and environments it contains.
   Note:
   If this replica is the result of a replication update, check to see whether or not the source Delphix appliance is still active. If it is still active, then disable any dSource or VDB that is part of the replica being failed over to ensure that only one instance is enabled. You can disable dSources and VDBs by selecting **Datasets**, finding the appropriate database, select **Disable**. After disabling the objects, navigate to **System > Replication** and click **Replicate Now** to get the most recent data to the target environment. If any scripts are accessing the engine, then you must repoint those scripts to the replication target environment to avoid VDBs from being started up in the old replication source environment.
2. Click **Failover** and a confirmation dialog will appear as below.To perform a test-failover see, Test-failover and Failback (see page 2400), confirm the dialog. For regular failover, uncheck the option and confirm.



This will pause while the replica is failed over.

3. When failover completes, the replica page will update.
4. Apply any configuration changes that are required to customize the objects for the system. This might include updating object states such as IP addresses, mount paths, or credentials. For more details, see the Replicas and Failover (see page 2397) topic.
5. Enable the environments that were failed over.
   a. Click **Manage**.
   b. Select **Environments**. The environments that were failed over will be disabled.
   c. From the Actions menu (...) select **Enable.**
6. Refresh any environments that were consolidated during failover.
   a. Click the **Refresh** icon for each affected environment.
7. Enable the dSources and VDBs that were failed over.
   a. Click **Manage**
   b. Select **Datasets**.
   c. Select the desired **database**.
8. From the Actions menu (...) select **Enable.**

> If the dSources and VDBs that were failed over belong to a plugin, depending on what version of the plugin existed on the target engine, the plugin may be in an inactive state. The plugin must be moved out of the inactive state before the object can be enabled. For more information see the Delphix Engine Plugin Management (see page 2257) topic.

11.4.6.3 After replication failover

Navigate to **System > Replication** and select the delete button to delete the replication specification.

A failover is similar to a migration operation for TDE-enabled vPDBs. If after the failover the original target hosts are still available, then the vPDB can be enabled on the failed over Delphix Engine without requiring

additional manual steps. If the vPDB is to be restored to a different target host, then the same manual steps are required for migration, as the vPDB will now be located on a new target and thus needs the artifact directory, parent Keystore, and merged Keystore available.

11.4.7 Uncontrolled failover

Before starting an uncontrolled failover, keep a note of the source and target engine's IP address and UUID. These details are available on the Delphix server setup application[746].

11.4.7.1 Uncontrolled failover procedure

1. On the source engine, create a replication specification for the dSources and VDBs by clicking **Replicate now**, to replicate these dSources/VDBs to the target engine.
2. To simulate a crash of the source engine, power off the source engine.
3. On the target engine, click on **Failover.**
   a. At this point, the target engine will become active and the replicated environments/dSources/VDBs will be seen in the default namespace of the target engine.

The following sections describe the procedures to handle an uncontrolled failover for each data platform:

- Oracle environment and data sources[747]
- SAP ASE environment and data sources[748]
- SQL server environment and data sources[749]

11.4.7.2 Oracle environment and data sources

After an uncontrolled failover, for Oracle SI and RAC complete the following procedure:

1. Enable dSources and VDBs on the target engine. dSources will be enabled but the VDBs/vPDBs will fail.
2. To enable VDBs, shutdown the database instance using **SHUTDOWN ABORT**.
3. Remove all the mounts from the target environment with sudo privilege.
   a. List all the mounts on the target environment using the `mount` command.
   b. Run `umount -lf <Delphix-mount>` to remove the mounts. For example: `umount -lf /mnt/provision/VDBOMSR8A1718_FPX/datafile`
4. Remove all the stale oracle processes of the database instance using `kill -p <process id>.`
   a. Run `ps -ef | grep “DATABASE INSTANCE` to get all the running processes associated with the database instance. For example `ps -ef | grep VDBOMSR8A1718_FPX`. This

746 https://cd.delphix.com/docs/19.0.0.0/determining-the-delphix-server-id-and-host-name
747 https://cd.delphix.com/docs/19.0.0.0/oracle-environment-and-data-sources
748 https://cd.delphix.com/docs/19.0.0.0/sap-ase-environment-and-data-sources
749 https://cd.delphix.com/docs/19.0.0.0/sql-server-environment-and-data-sources

command list all the running processes, e.g. **oracle 15236 9227 0 11:01 pts/0 00:00:00 grep --color=auto VDBOMSR8A1718_FPX**

b. Run `kill -p <process id` to kill all the running processes. For example: `kill -9 15236`

5. Now enabling VDBs and VPDBs will be successful.

11.4.7.3 SAP ASE environment and data sources

After an uncontrolled failover has been triggered, on the target/staging host, the SAP ASE dSource/VDBs mounts, corresponding to the source engine, will still be present. This can be checked using the following command on the target host:

mount | grep <source-engine-ip>
These mounts will be mounted in the user-provided toolkit directory (Example */work*) and these would be of the form **/ork/<Engine-UUID>-[staging/vdb]/[datafile/archive/temp**]. The following is an example for a dSource staging database mount **/work/4201763f-2f8d-1c8f-381a-effb180b0328-staging-2/datafile**.

All the mounts will be using the source engine's UUID.

Attempting to '**Enable**' or '**Start**' the ASE dSources/VDBs on the target engine while the NFS mounts are in this stale state, will cause the jobs to hang or timeout or result in a job failure.

To clear off these stale mounts, one of the following options can be used:

Manually unmount

1. Shutdown the concerned SAP ASE instances which host the dSource's staging database and VDBs.
2. List all the dSource/VDB mounts with a **mount | grep <source-engine-ip>.**
3. Remove all the mounts with a **umount -lf <Delphix-mount>** for example: **umount -lf /work/4201763f-2f8d-1c8f-381a-effb180b0328-staging-1/archive**, this might require sudo privileges. NOTE: Use the appropriate syntax for **umount** for your operating system. The above example is for Linux.
4. **Restart** the SAP ASE instances.

OR

Reboot the target/staging host

Reboot the target/staging host corresponding to the concerned dSources/VDBs. This option is probably the cleanest way of clearing the source engine's leftover dSource/VDB mounts, but should only be used if the concerned target/staging host(s) do not host any other databases that are not part of the uncontrolled failover.

Enabling Environments, dSources, and VDBs

1. Enable the source environment and then the target environment.
2. Enable the dSources and VDBs on the target engine.
3. Mounts will be moved to a new Delphix(Target Engine) and can be checked by using the following command: **mount | grep <target-engine-ip>** These mounts still use the source engine's UUID and not

the target engine's UUID, but validated/manual syncs on dSources and refreshes/rewinds on VDB operations will be successful.

11.4.7.4 SQL server environment and data sources

11.4.7.4.1 Enabling environments and dSources

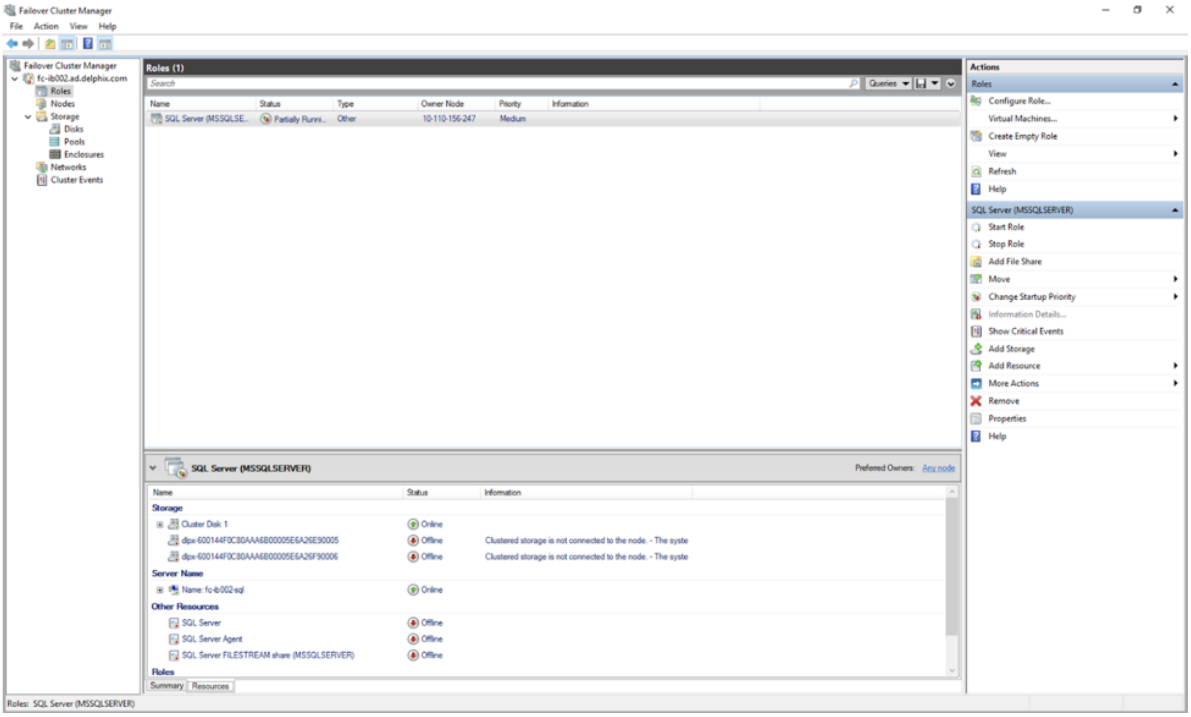
1. Enable the proxy(staging) environments followed by the source environments.
2. Enable dSources on the target engine.
3. Delete the staging database stale mounts from the connector directories. An example directory: **C:\ProgramFiles\Delphix\DelphixConnector\<Old Engine ID >-staging-<X>** where **X** is an arbitrary id representing the staging database.
4. Delete the staging database from the SQL instance on the proxy host using the following convention: <**Old Engine ID**>-**staging**-<**X**> where **X** is an arbitrary id representing the staging database.

11.4.7.4.2 Enabling VDBs on the standalone host

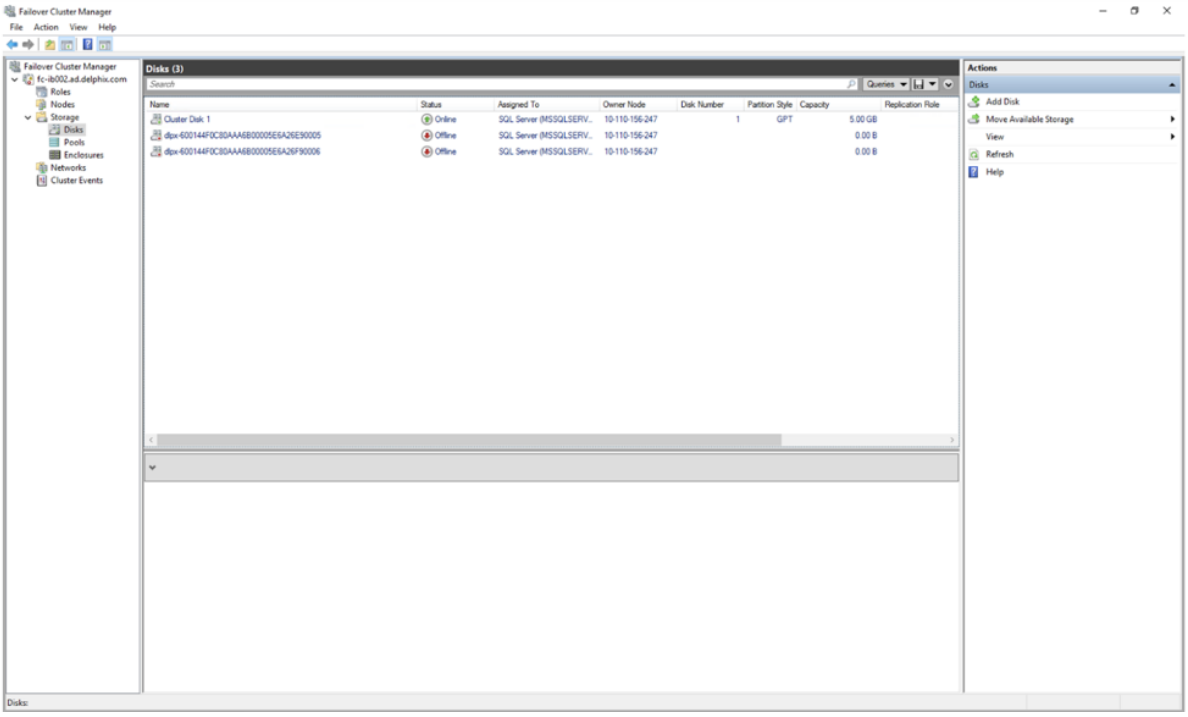
1. Delete the VDB directories stale mounts from: **C:\ProgramFiles\Delphix\DelphixConnector\\<Old Engine id>-vdb-<X>** where **X** is an arbitrary id representing the virtual database.
2. From the SQL instance delete VDB databases that are in recovery pending state.
3. Enable VDBs. New mounts will be created and a new engine ID will reflect on those mounts.

11.4.7.4.3 Enabling VDBs on failover cluster as target

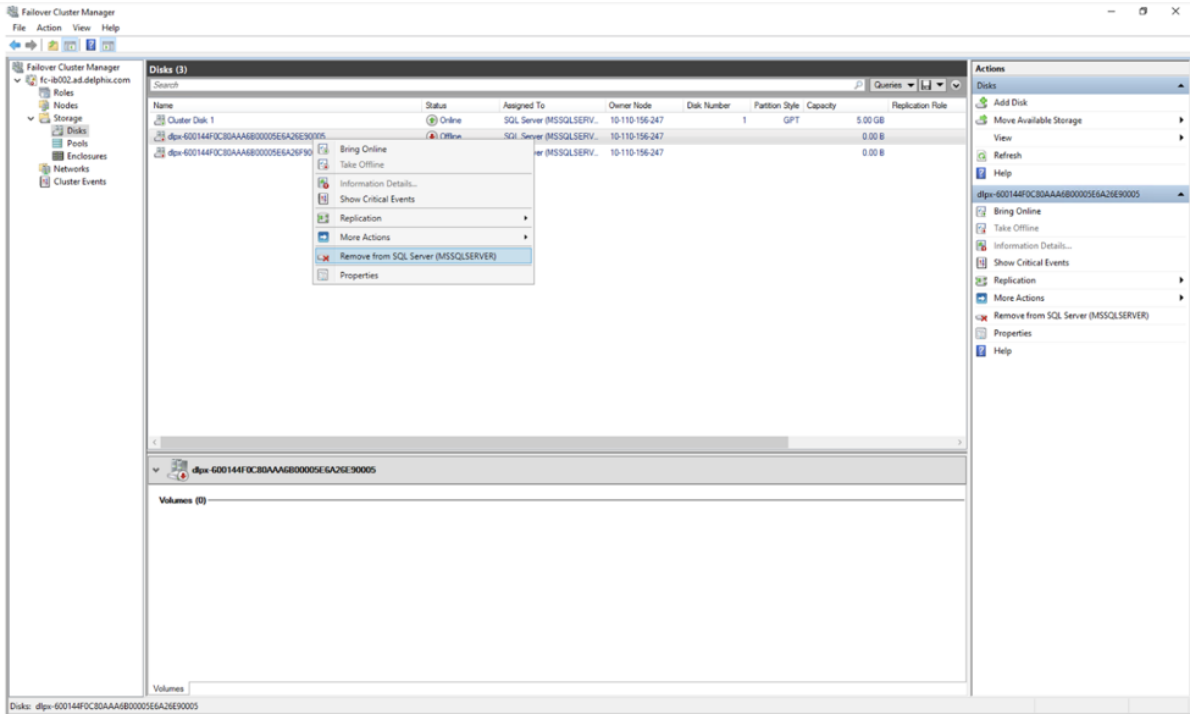
1. Once the source Engine is powered down login to **Failover cluster**.
2. Go to **Windows Administrative Tools** and select **Failover cluster manager**
3. On the left pane of the **Failover cluster manager** go to the **Roles**. As shown in the image below you will see the Failover cluster in the following state with the indicated status.



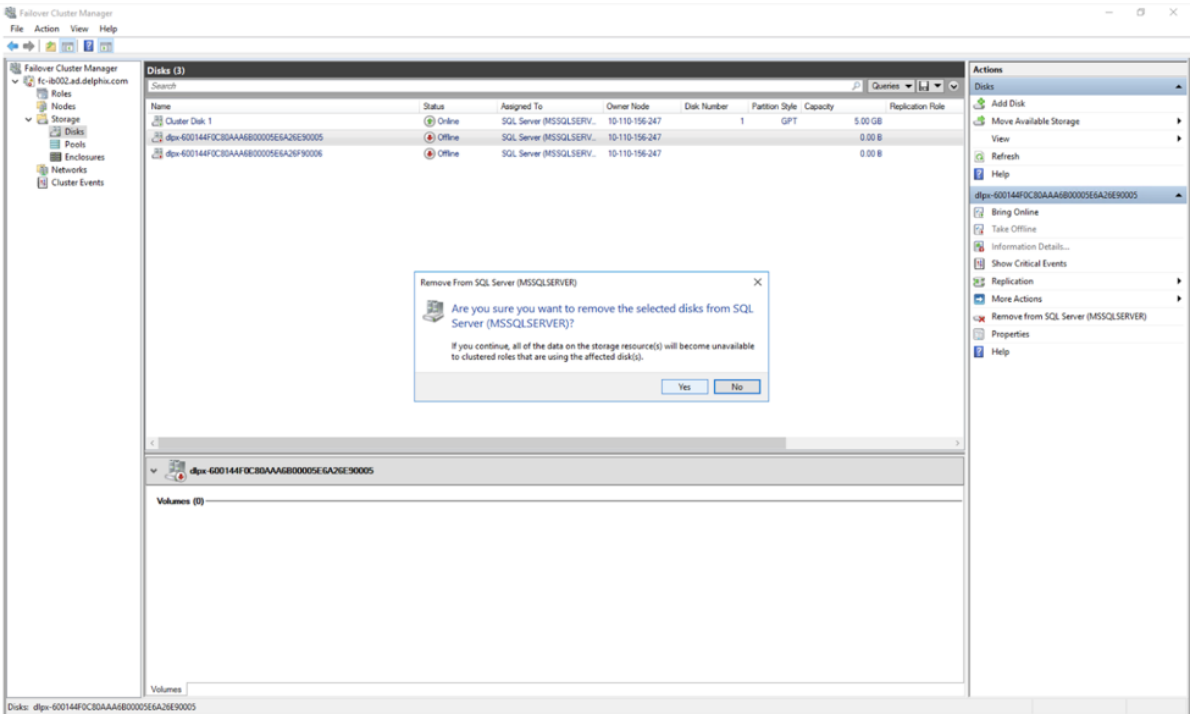
4. In order to fix the state go to the **Storage** section and select **Disks**. The **dlpx-XXXXXXXXX** mounts attached to the Cluster Disk will be in offline states. These mounts will be equal to the total number of VDBs provisioned on that cluster as the target.



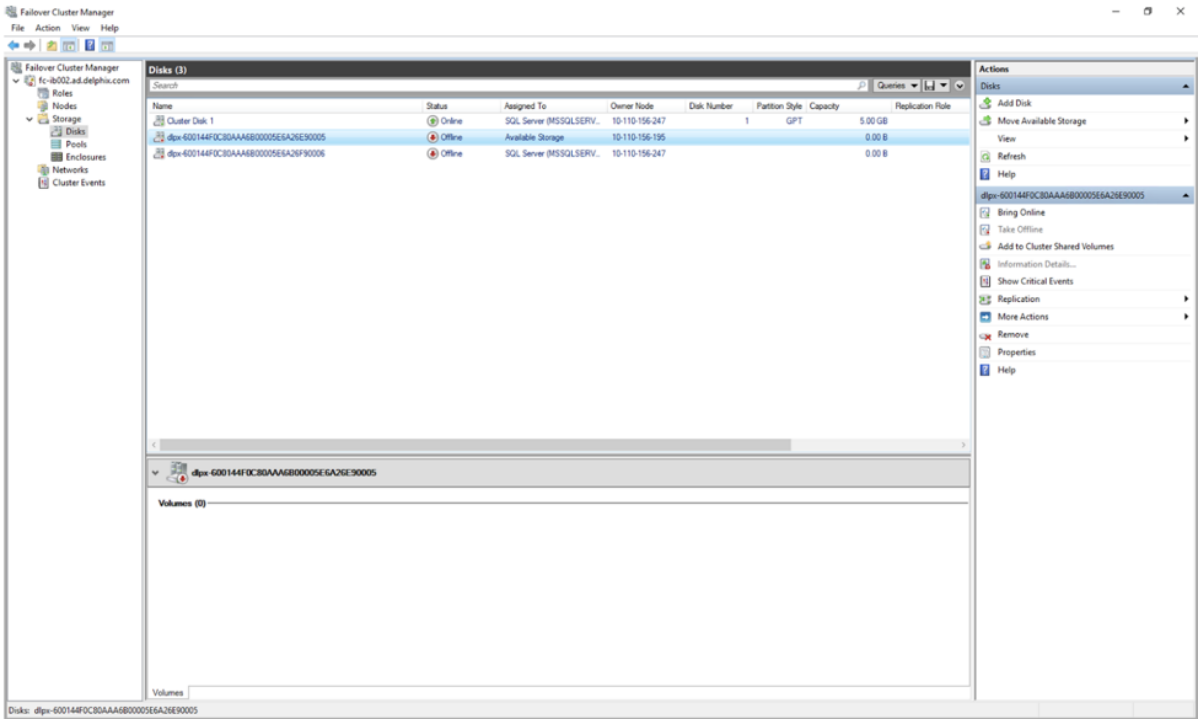
5. Right-click on the dlpx mount, and select **Remove from SQL Server (Instance Name)**.



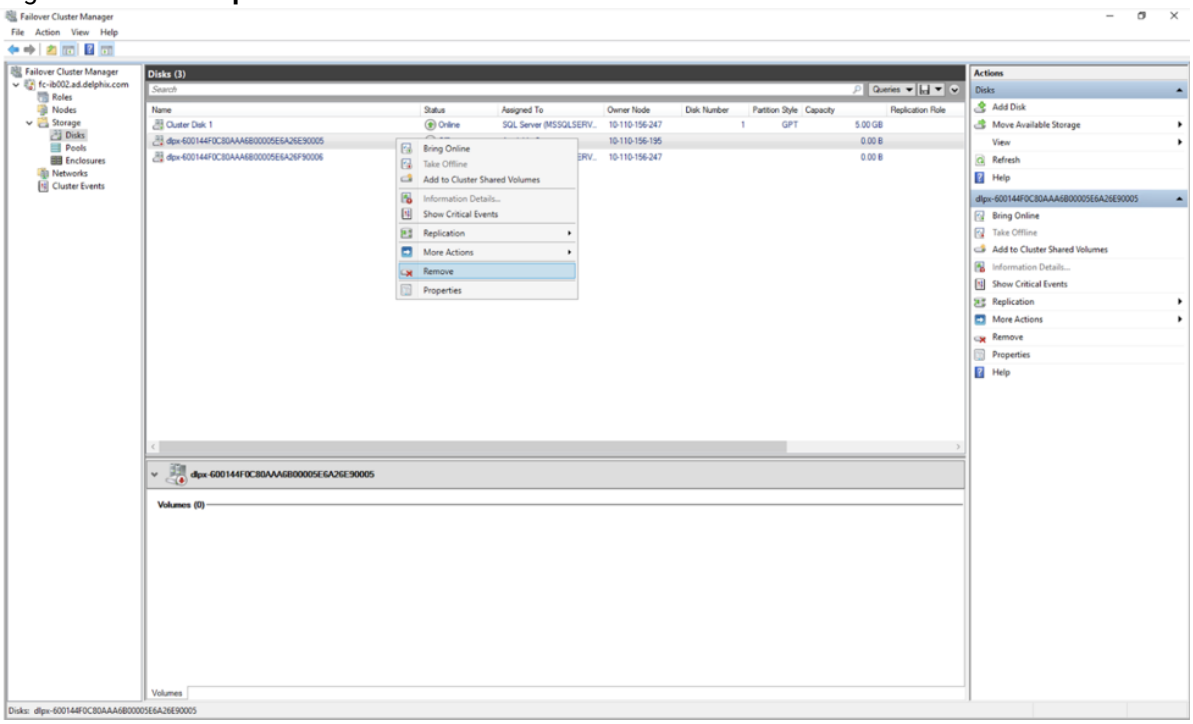
6. Click **Yes**.



In the **Assigned To** column, you will see **Available Storage**.

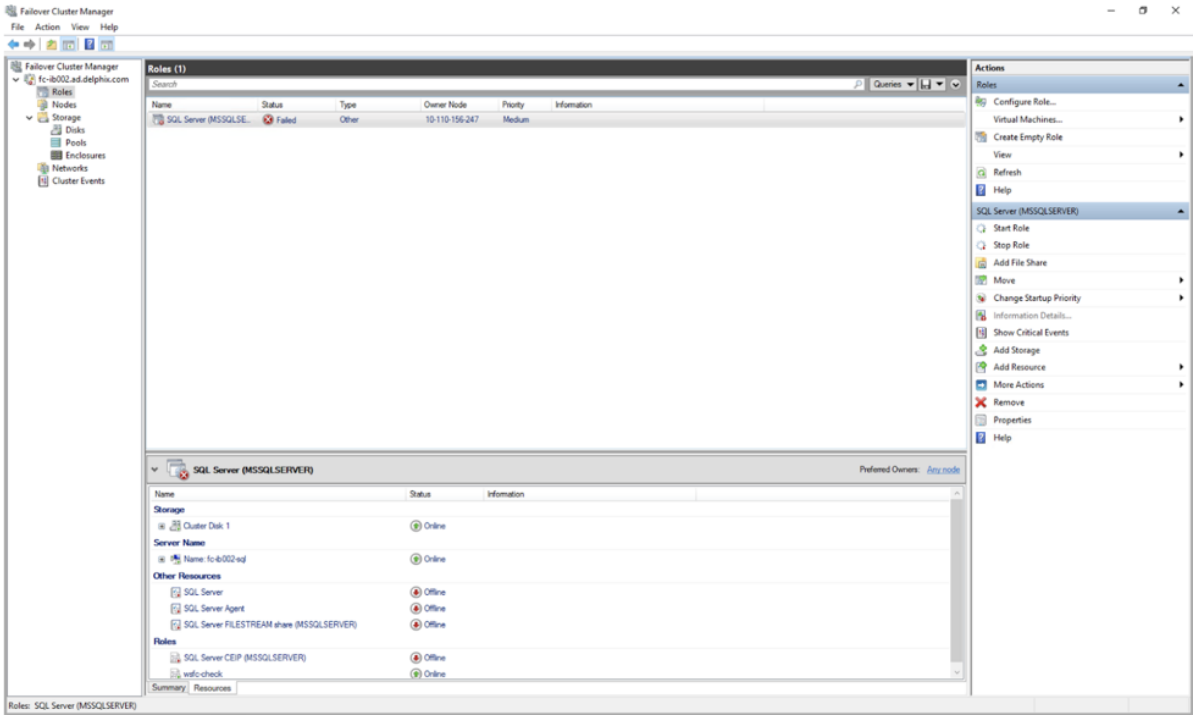


7. Right-click on the **dlpx mount** and click on **Remove**.

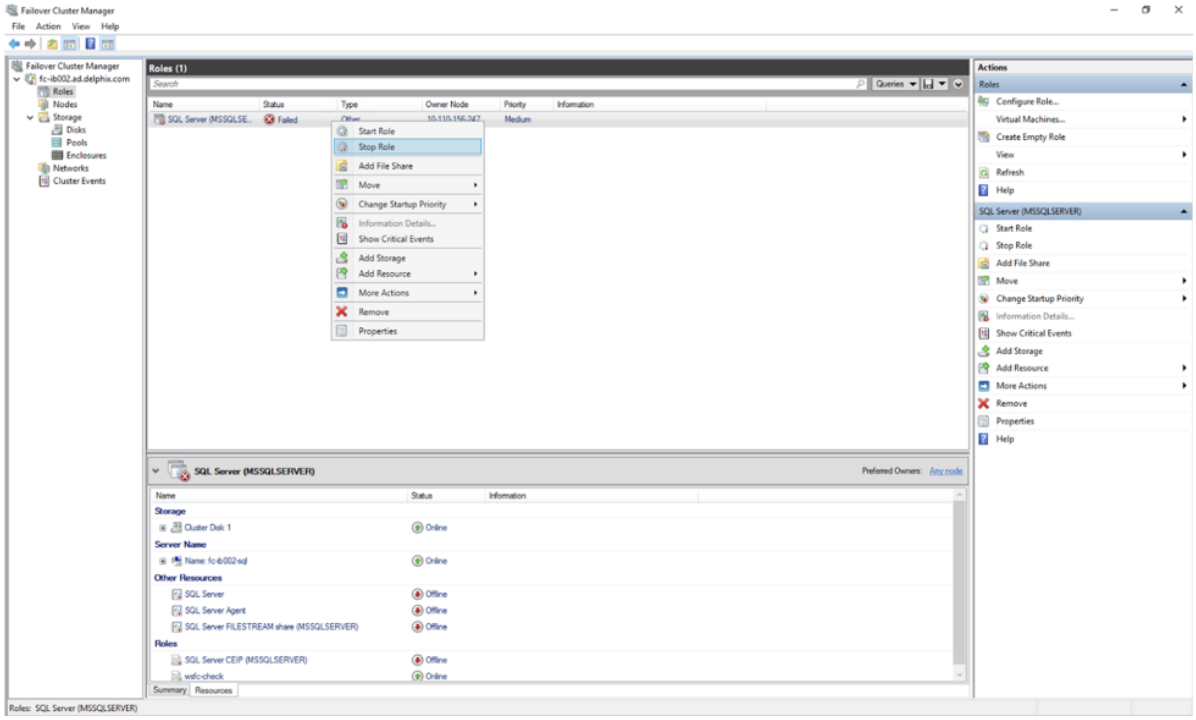


8. The dlpx mount will be deleted. Repeat the same process with other dlpx mounts assigned to **Available Storage**.

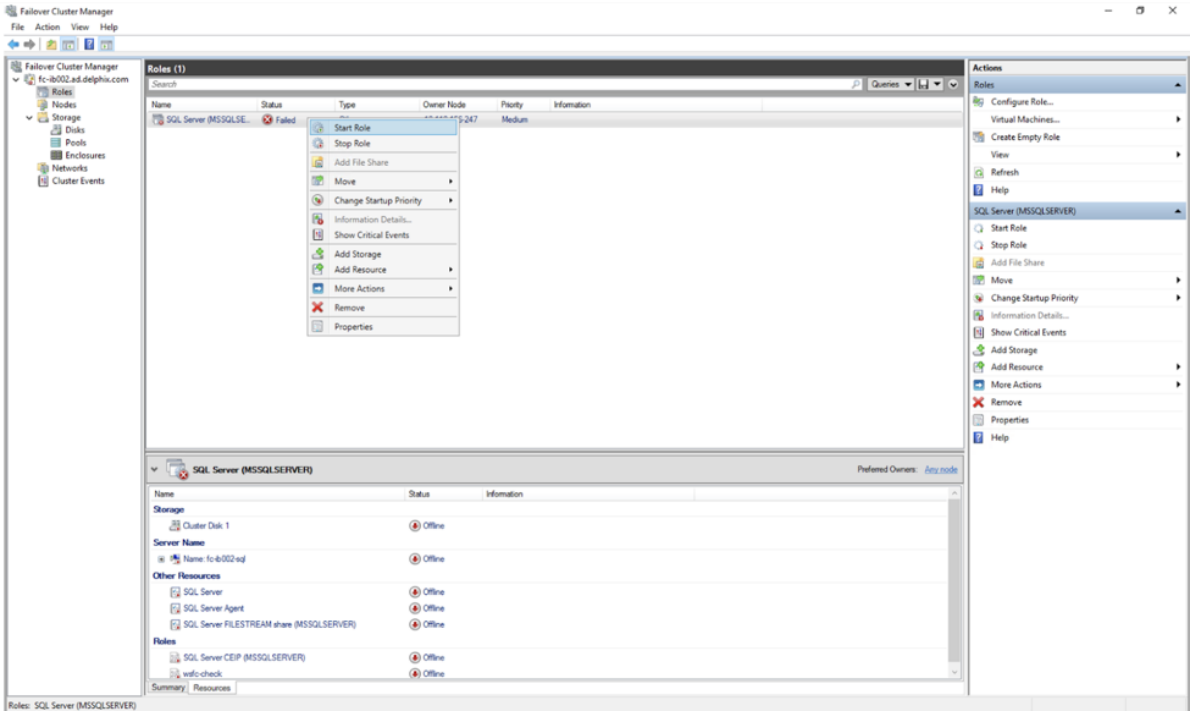
9. Under **Roles**, you would see that the **SQL Server service** is now in a **Failed** state.



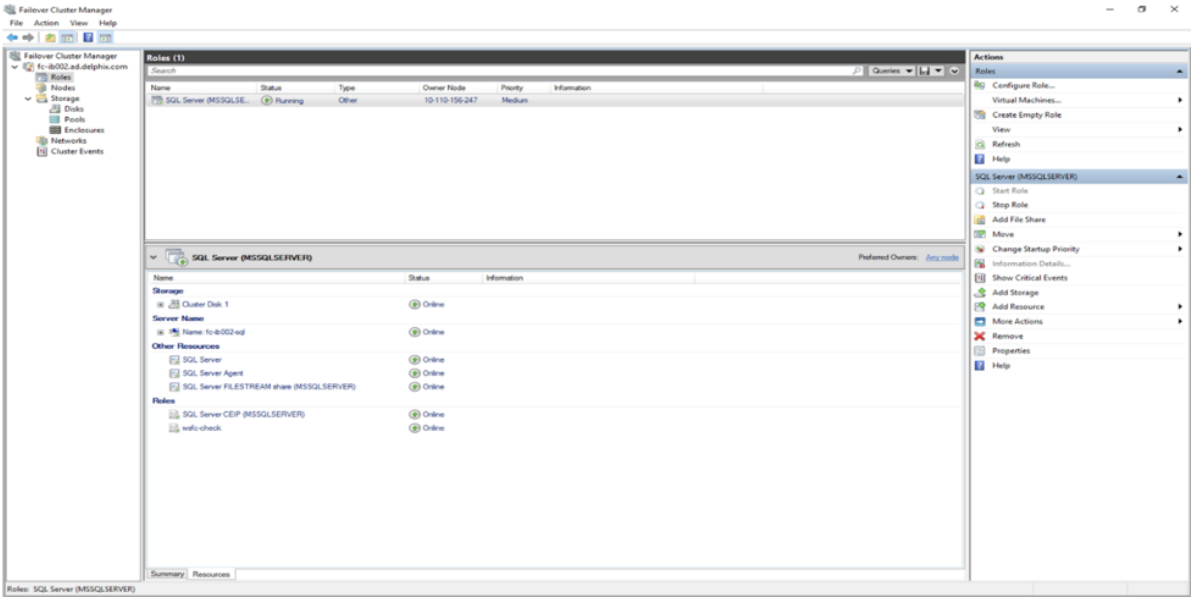
In order to fix it right-click on the service and click **Stop Role**.



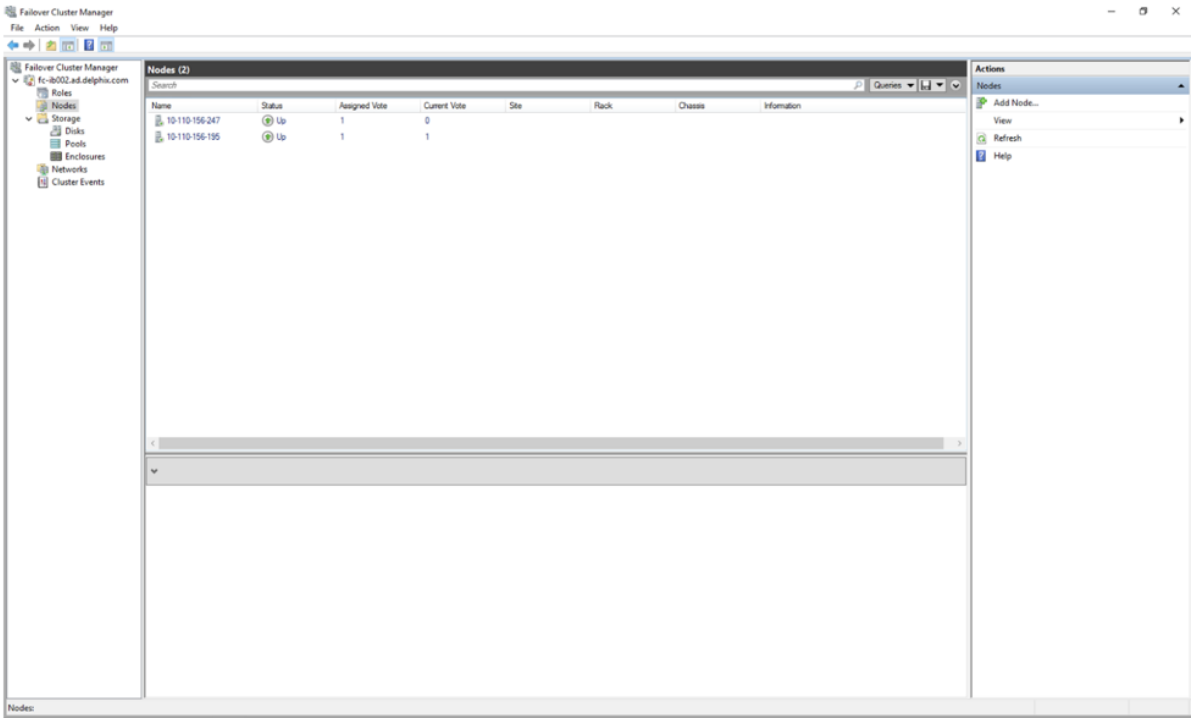
10. Once the Stop role is executed properly, right-click on the **SQL cluster service** and click on **Start Role**.



11. The SQL cluster service should now be online. Under the **Resources** tab the Storage, Server Name, Other Resources and Roles should have Online status. Ensure that the Disks and Nodes are also in an online state.



12. Make sure that both the nodes are in a healthy state.



11.4.8 Managing replicated objects

Once the failover process has completed the last step is to enable the objects on the Delphix Engine that received a replica. There are two major concepts:

1. **SSH Keys**: Public keys need to be re-published on the source, staging, and target environments for the replicated Delphix Engine, since the replication process does not automatically copy SSH keys.
2. **Enabling Replicated Objects**: Objects will need to be re-enabled once replication is complete. This ensures that replication has completed successfully and the configuration of each object is correct.

11.4.8.1 SSH keys

If you were previously using a password-less SSH key exchange on your source and/or target environments, you may need to publish the now-primary engine's public key for continued functionality.

1. Click **Manage**.
2. Select **Environments**.
3. Select **any environment**.
4. Under **Environment Users**, click the **edit** icon () next to the defined user.
5. Click **View Public Key** to display the public key for the Delphix Engine.
6. Highlight the public key string (starts with "ssh-rsa") and copy the key to your clipboard (Ctrl+C in Windows).

7. On each source and target host within your defined environments, paste the public key into the environment user's authorized_keys file, which is normally located in the user's ~/.ssh/ directory.

Once the public key has been copied to the hosts making up your environments, you are ready to enable the remaining objects.

11.4.8.2 Enabling environments

In order to begin using the environments and related dSources and VDBs, the environments must first be started.

1. Login to the Delphix Management application.
2. Click **Manage**.
3. Select **Environments**.
4. Select an **environment** that you want to enable.
5. From the Actions menu (...) select **Enable.** This will initiate jobs that will refresh and enable the environment.
6. Repeat these steps for each environment that you want to enable.

11.4.8.3 Enabling dSources

In order to again have actively syncing dSources, you must enable them.

1. Login to the Delphix Management application.
2. Click **Manage**.
3. Select **Datasets**.
4. Click the **dSource** you want to enable.
5. From the Actions menu (...) select **Enable.**
6. Click **Enable** to confirm.

11.4.8.4 Enabling VDBs

The final step in the failover would be to enable any needed VDBs.

1. Login to the Delphix Management application.
2. Click **Manage**.
3. Select **Datasets**.
4. Click the **VDB** you want to enable.
5. Click the **Configuration** tab.
6. From the Actions menu (...) select **Enable.**

7. Click **Enable** to confirm that you want to enable the VDB.

At this time the failover is complete. All objects previously running on the previous source Delphix Engine should now be running off of the target Delphix Engine on which the objects were failed over.

11.4.9 Replicas and failover

Replication recreates objects on the target system in a replica that preserves object relationships and naming on the target server without interfering with active objects on the system. Objects within a replica are read-only and disabled until a replica is failed over, at which point they can be activated. VDBs and dSources within a replica can be used as the source for provisioning new VDBs.

11.4.9.1 Replicas

A replica contains a set of replicated objects. These objects are read-only and disabled while replication is ongoing. To view replicated objects, look under **namespace** in the CLI. Or in the GUI:

1. Click **System**.
2. Select **Replication**.
3. Under **Received replicas**, select the replica.

On this screen, you can browse the contents of replicas, as well as failover or delete individual replicas. As described in the Replication overview (see page 2368) topic, databases (dSouces and VDBs) and environments are included within the replica.

Deleting or failing over a replica

Deleting or failing over a replica will sever any link with the replication source. Subsequent incremental updates will fail, requiring the source to re-establish replication. Failover should only be triggered when no further updates from the source are possible, as in a disaster scenario.

Multiple replicas can exist on the system at the same time. Active objects can exist in the system alongside replicas without interfering with replication updates. You can also use VDBs and dSources within a replica as a source when provisioning. For more information, see Provisioning from replicated data sources or VDBs (see page 2403)

11.4.9.2 Failover and conflict resolution

To activate the objects in a replica, you must first fail over the replica. This will sever replication and move the objects to the live system, where they can be manipulated in the same fashion as other objects on the system.

Objects that are failing over can conflict with objects in the live system. One reason for conflicts is identical names. For example, Groups will conflict if the failing over Group has the same name as a Group in the live system, as will most other objects including Environments, dSources, and VDBs.

Most of the object conflicts can be resolved automatically. Objects like Groups will be renamed to avoid conflict and objects like Environments will be merged and consolidated. After conflict resolution and successful failover, a report is presented with a list of objects that needed conflict resolution.

> Conflict resolution is not supported for objects like dSources and VDBs.

11.4.9.2.1 Conflicting plugins

Due to the potential for plugin version incompatibilities, if any of your replicated objects are plugin-based, we highly recommend that you leave the target system completely passive with no active objects until the time that failover is required. For more information, see the Delphix engine plugin management page.

When environments are consolidated, the Delphix engine maintains the existing environment configuration and privilege elevation profile; and merges replicated objects with them. Therefore, you must check and update the configured privilege elevation profiles when Environments are consolidated.

Once a replica is failed over, the objects are active but will be automatically disabled.

11.4.9.2.2 Manual conflict resolution

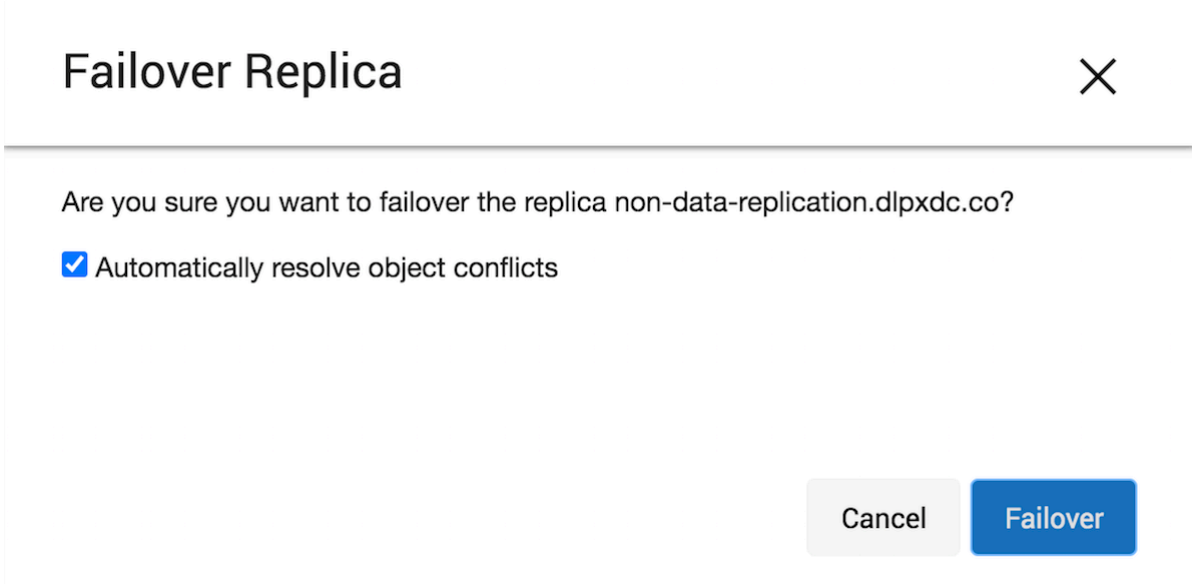
To resolve conflicts in Groups, you may either rename the conflicting group on the target replication engine or if the source replication engine is still available, rename on the source replication engine and send a replication update, before failover. For all other object types, because the replica objects are read-only, you must rename the active objects on the replication source engine and send a replication update before the failover operation can complete successfully.

11.4.9.2.3 Automatic conflict resolution

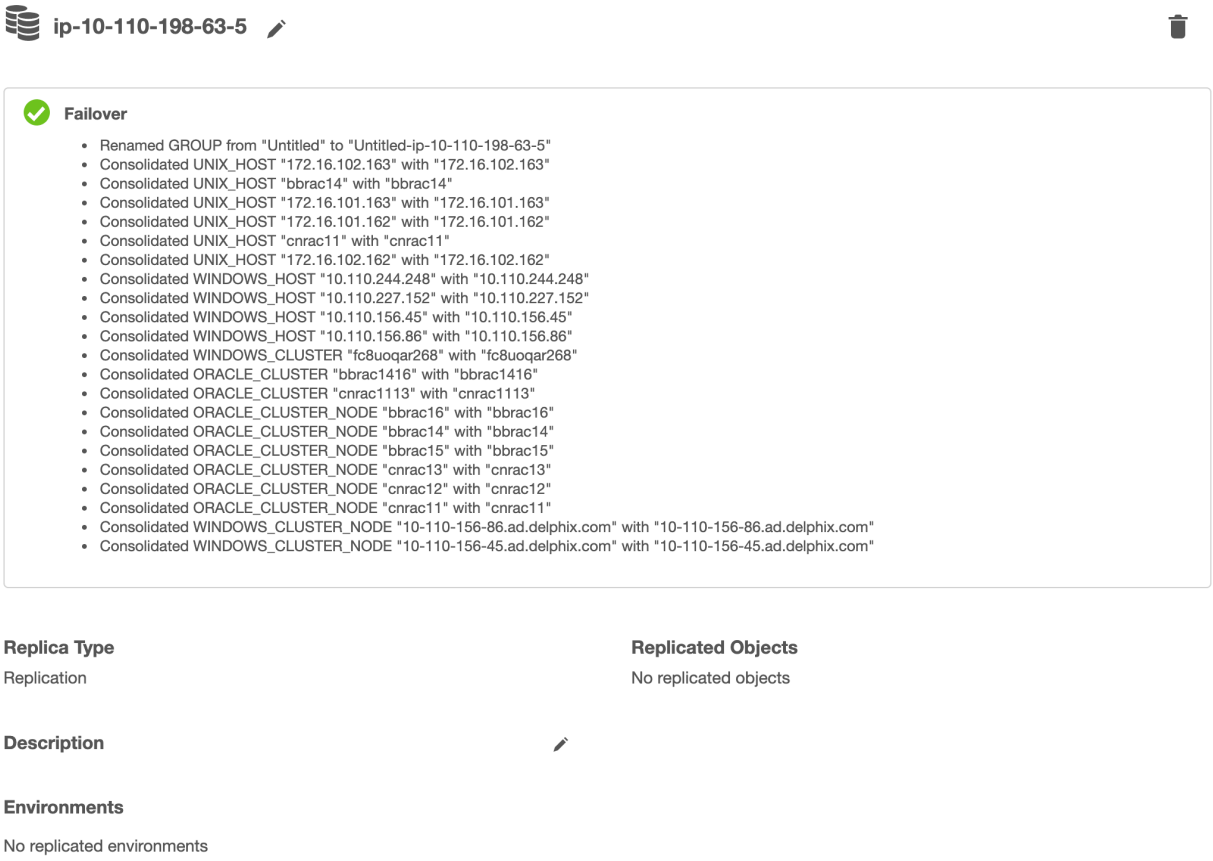
> Starting with 9.0.0.0, smart-failover a.k.a automatic conflict resolution is no longer optional. It is always on and there a GUI option is no longer presented. Starting 14.0.0.0, the enableFailback option is presented instead.

Starting with 6.0.2.0, most of the object conflicts can be resolved automatically. When you select "Automatically resolve object conflicts", replica objects like Groups will be renamed whereas objects like Environments will be merged and consolidated.

As of 6.0.5.0, the Automatic Conflict Resolution option will be chosen by default.



After automatic conflict resolution and successful failover, a report is presented with a list of objects that needed conflict resolution.



Automatic conflict resolution is not supported for objects like dSources and VDBs.

11.4.9.3 Enabling databases and environments

Objects may refer to states (IP addresses, mount paths, etc) that differ between the source and target system. Because of this, all databases and objects within a replica automatically start in the disabled state after a failover. This allows the administrator to alter configuration prior to enabling databases and environments, without the system inadvertently connecting to invalid systems.

After failover is complete, you must explicitly enable all dSources, VDBs, and environments. If you need to change any configuration for the target environment, you can do so prior to enabling the objects. In the event that a failing-over environment is consolidated with a live system environment, it must be refreshed before all of it's databases can be used.

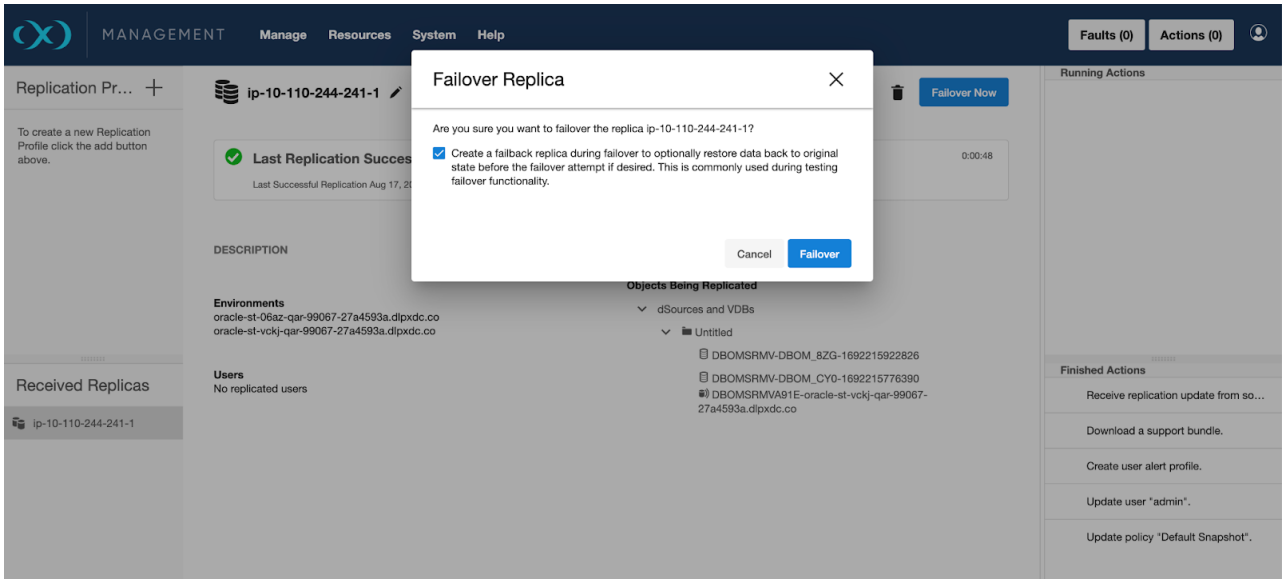
11.4.10 Test-failover and failback

11.4.10.1 Test-failover

Starting with 14.0.0.0, a test-failover can be performed to validate that the failover actually works when disaster strikes. The failover operation can be undone by doing a failback operation.

The failback operation takes the replication target back to the pre-failover state so that replication incremental can be received.

During failover, there is an option to save the state so that a failback can be performed. This is the `enableFailback` option and it is turned on by default.



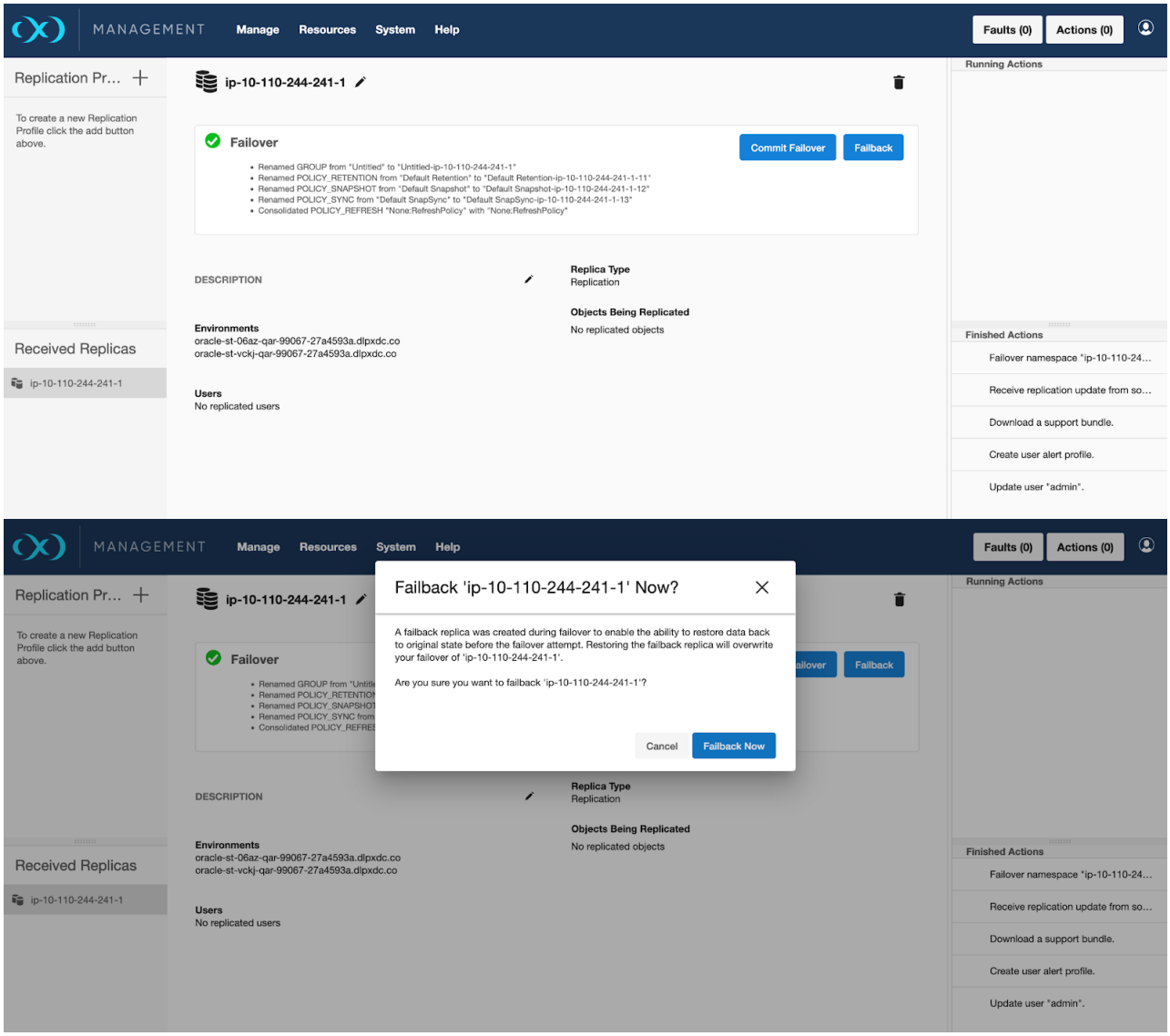
Once failover is complete, environments and datasets may be enabled and activated as mentioned in Controlled failover (see page 2385).

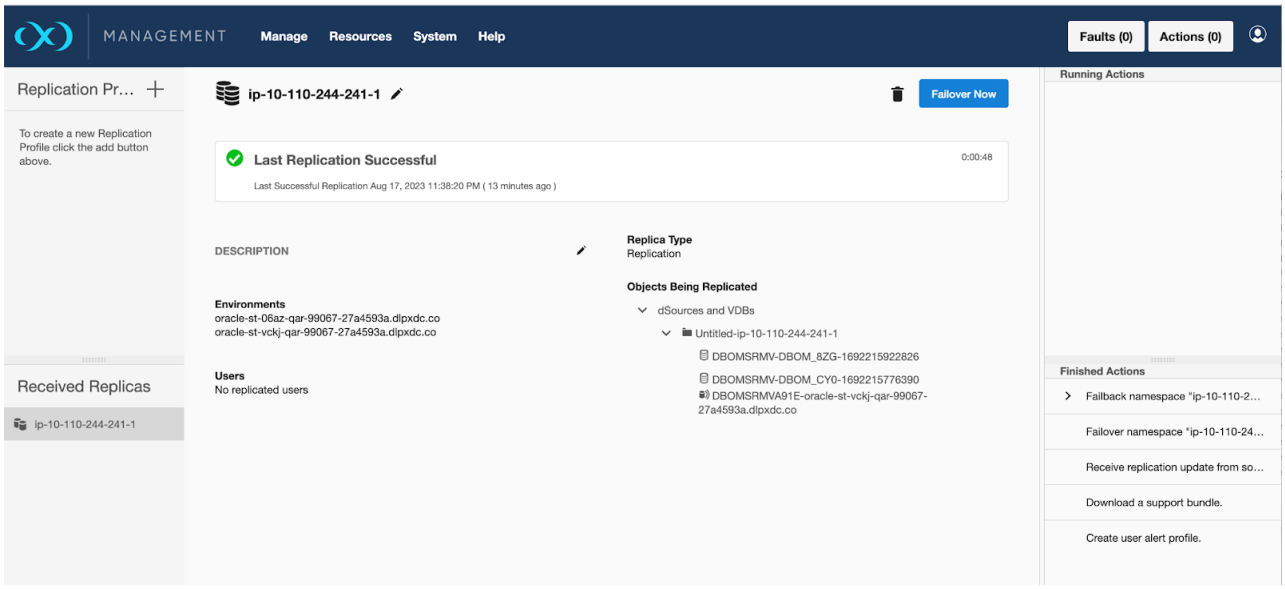
A test-failover does not place restrictions on the kinds of operations that may be performed on the failed over data. Environments and datasets may be enabled and used. New snapshots may be created. New VDBs may be provisioned from the failed-over datasets.

11.4.10.1.1 Failback

On failback, all newly provisioned child datasets will be destroyed. All new snapshots and refreshed time-flows will be deleted. The target-replication engine will thus be brought back to the state prior to failover by moving the objects into a read-only replica namespace so that the replication-source engine can continue replicating to the replica namespace.

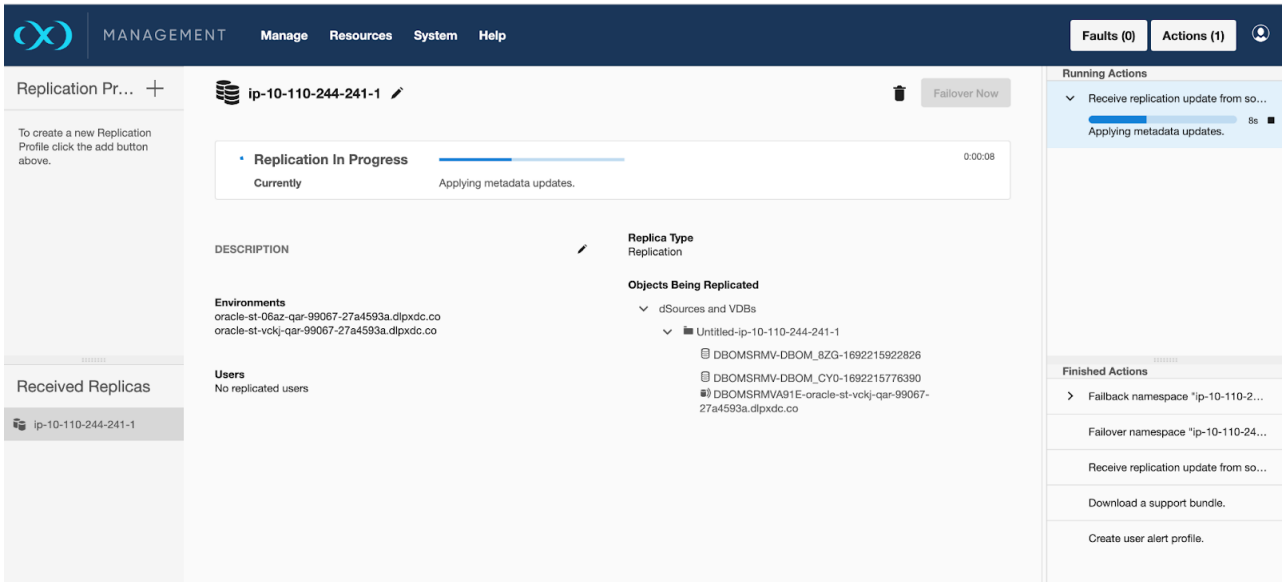
No changes are made to the replication-source engine during failback. No changes from the replication target are propagated back to the replication source. Thus failback can be thought of as an “undo failover” operation.





Full restoration of the **Received Replica** namespace requires a successful replication-receive from the replication-source engine after performing the failback operation. It is a best practice to do this right after a failback.

> The replication profile on the replication-source engine needs to be preserved for receiving updates to the *Received Replica* namespace.



11.4.10.1.2 Commit failover

If a failback is unnecessary, the test failover can be committed, making the failover permanent. This is similar to performing a failover with the enableFailback option off.

11.4.11 Provisioning from replicated data sources or VDBs

This topic describes how to provision from a replicated dSource or VDB. The process for provisioning from replicated objects is the same as the typical VDB provisioning process except for the need to first select the namespace containing the replicated object.

11.4.11.1 Prerequisites

You must have done the following:

- replicated a dSource or a VDB to the target host, as described in Replication overview (see page 2368)
- added a compatible target environment on the target host as described in Provisioning Oracle VDBs: an overview (see page 1515)
- installed on the target host any App Data plugin which the replicated objects depend on

11.4.11.2 Procedure

1. Login to the **Delphix Management** application for the target host.
2. Click **Manage**.
3. Select **Datasets**.
4. Select the **replica** that contains the dSource or VDB to be provisioned.
5. The provisioning process is now identical to the process for provisioning standard objects.If the dSource or VDB belongs to a plugin, the plugin must exist on the target engine to be successfully provisioned. The target's plugin's version must also be equal to or higher than the source's plugin's version. For more information see the Delphix engine plugin management (see page 2257) topic.

11.4.11.3 Post-requisites

Once the provisioning job has started, the user interface will automatically display the new VDB in the live system.

11.4.12 Replication user interface

The replication user interface helps users manage replication on both the source and the target. Replication consists of a profile-replica pair. You can view and edit the replication profile on the source engine, and view the replica on the target engine.

11.4.12.1 Sources for replication

The **Replication Profiles** section provides you with several configuration options for your replication profiles. This makes it possible to replicate objects from a single source to multiple targets. Each profile defines the set of data objects and the associated configuration between a single source and target.

11.4.12.2 Targets for replication

The **Received Replicas** section shows the set of all objects in the replica and allows you to initiate failover.

11.4.12.3 Canceling a replication job

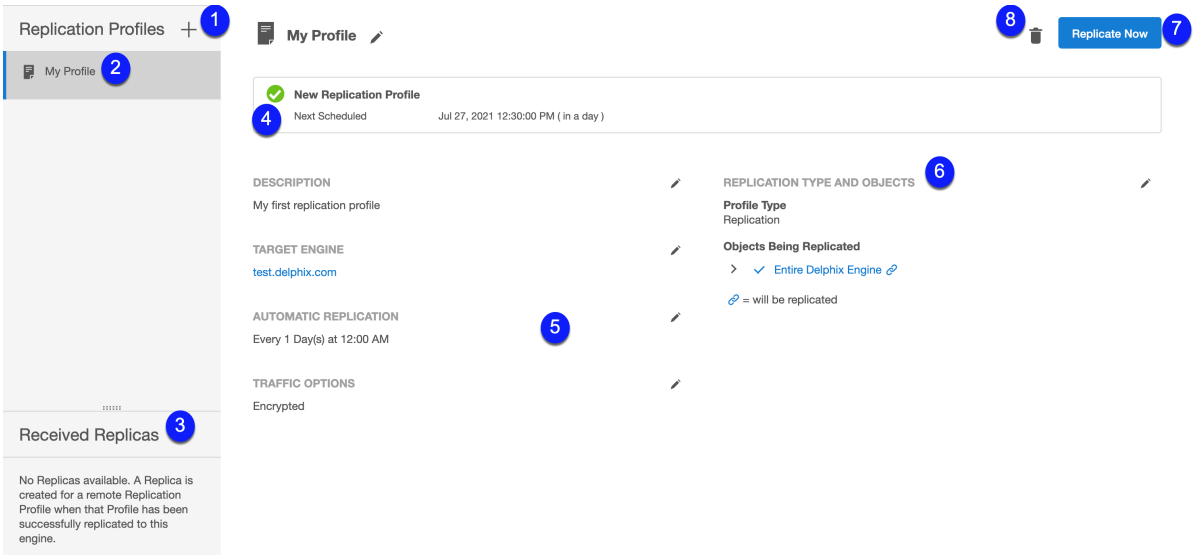
You must cancel the replication job at the source engine. Canceling the replication job at the target engine results in the re-starting of the receiving job at the target engine. If the source engine is down, then you can cancel the replication job at the target engine. If you re-start the source engine where the replication job was previously running, the source engine will attempt to restart the replication send job.

11.4.12.4 Replication UI

The Replication UI consists of the following sections.

Replication profiles

The screenshot and descriptions below illustrate the capabilities in the **Replication profiles** section.



1. **Create profile button**
   It allows you to configure a Replication Profile.
2. **Replication profiles list**
   Provides a list of existing replication profiles. Click a profile in this list to view its details.
3. **Received replicas list**
   Provides a list of all existing replicas on this Delphix Engine. Click a replica in this list to view its details.
4. **Status box**
   Shows the replication status of the selected profile. This includes the result of the most recent or

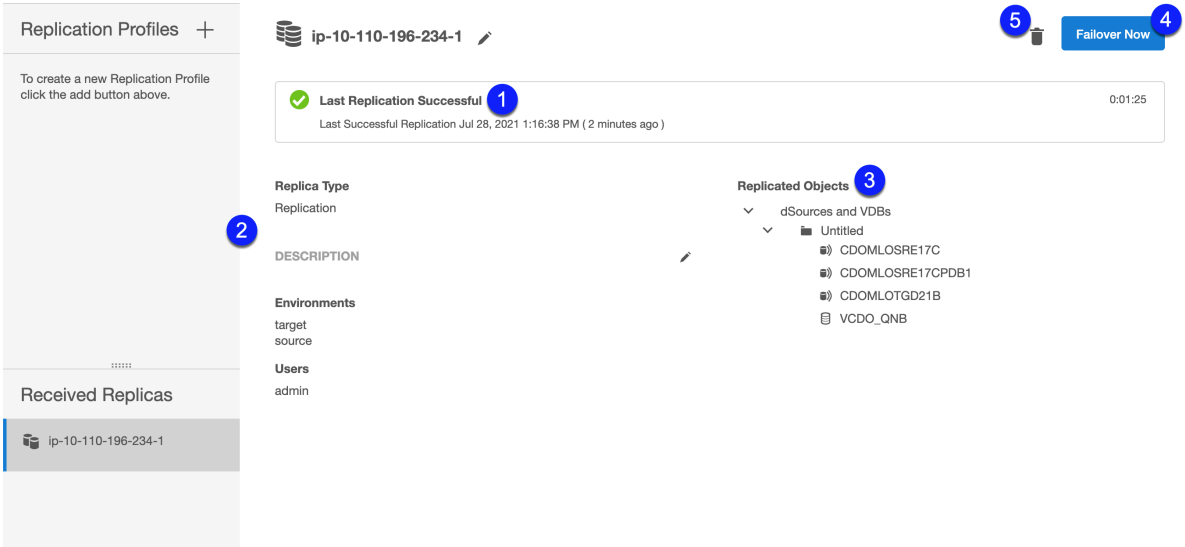
current replication event and statistics for the replication run (i.e. data transferred, duration, average throughput, etc.).

	This icon appears while a replication job is in progress.
	This icon appears after a successful replication job.
	This icon appears when a replication job has failed.
	This icon appears when a replication job was canceled.

5. **Configuration options**
   Additional configuration options for the selected replication profile.
   Description: Free text field for a profile description.
   Target Engine: The Delphix engine on the receiving end of this replication pair.
   Automatic Replication: If enabled, shows the frequency and time that regular replication will be run.
   Traffic Options: Summarizes the traffic options with which this profile has been configured.
6. **Objection selection tree**
   Shows all of the objects, such as groups, dSources, VDBs, and Self-service (JetStream) data layouts, that you have selected for replication in this replication profile. Selecting **Entire Delphix Engine** will cause all objects on the engine to be replicated, and thus the tree is collapsed.
7. **Replicate now button**
   Begins the replication process
8. **Delete button**
   Allows you to delete the current profile.

11.4.12.4.1 Received replicas

The screenshot and descriptions below provide more details of the functionality of this section.

1. 

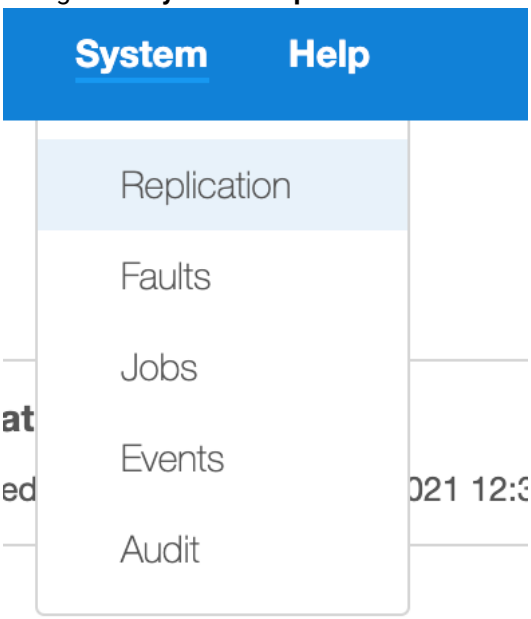
Status box
Similar to the **Replication profile** status box, this shows the most up-to-date status information for the replica on the target

2. **Replicated environments**
   Replicating a dSource or VDB will automatically replicate any environments associated with those objects. For more information, see Replication overview (see page 2368)
3. **Replicated objects tree**
   A read-only view of the objects in this replica.
4. **Failover button**
   Initiates a failover for this replica
5. **Delete button**
   Deletes this Replica on the target. This does not have an effect on the corresponding profile on the source engine.

11.4.12.5 Create new replication profile wizard

Perform the following steps to create a new replication profile.

1. Login to the **Delphix Management** application.
2. Navigate to **System** > **Replication**.



3. Next to **Replication Profiles**, Click the **plus** icon. The **Add Replication Profile** wizard screen appears.
4. Enter the **Name** of the replication profile and an optional **Description**, and click **Next**.
5. To specify the **Object**, select the **Profile Type** as **Replication Profile** and select from the **Objects Being Replicated** to a target engine. Click **Next**.

Note:
Selected Objects

a. Some selected objects may have dependencies – other objects that will be pulled into replication because they share data. Objects that will be replicated are confirmed with a blue chain link icon.
    i. Note that this is not guaranteed to be the full set of dependent objects, but rather is the best guess. The full set of objects and their dependents will be calculated at the time of replication.
b. When selecting objects, you can select the entire server (Entire Delphix Engine) or a set of groups, dSources, VDBs, and Jet Stream data layouts.
c. When replicating a group, all dSources and VDBs currently in the group, or added to the group at a later time, will be included.
d. If you select a Delphix Self-Service data template, all data containers created from that template will be included. Likewise, if you select a data container, its parent data template will be included.
e. If you select the entire server, all groups and Delphix Self-Service objects will be included.
f. Regardless of whether you select a VDB individually or as part of a group, the parent dSource or VDB (and any parents in its lineage) are automatically included. This is required because VDBs share data with their parent object. In addition, any environments containing database instances used as part of a replicated dSource or VDB are included as well.
g. When replicating individual VDBs, only those database instances and repositories required to represent the replicated VDBs are included. Other database instances that may be part of the environment, such as those for other VDBs, are not included.
h. Non-data objects (Delphix users, roles, permissions, authorizations, policies, database configuration templates) that are associated with selected objects will be automatically included during replication. They will not be shown on the selection interface.

6. For **Target Engine**, enter the **hostname** or **IP address** for the target Delphix Engine. Enter the **username** and **password** of a user who has Delphix Admin-level credentials on the target Delphix Engine. If the username or password changes on the target Delphix Engine, you must update these settings on the source Delphix Engine. Click **Next**.
7. By default, automatic replication is disabled and you must trigger replication updates manually. To **Schedule** automatic replication, click the **Enabled** checkbox.
   In the **Automatic Replication** field, enter the frequency and starting time for replication updates to the target Delphix Engine. Once you have entered and saved your replication settings, you will also see an option to trigger replication immediately with the **Replicate Now** button.
   Under **Traffic Options**, select whether you want to **Encrypt** traffic or **Limit bandwidth** during replication updates. Click **Next**.
8. The final summary tab will enable you to review your configurations. Click the **Back** button to go back and to change any of the configurations or click **Submit** to complete the wizard.

11.4.12.6 Viewing and editing an existing replication profile

Perform the following steps to view and edit existing replicas.

1. Login to the **Delphix Management** application.
2. Navigate to **System** > **Replication**.
3. Under the **Received Replicas** section, select a replica.
4. You can edit the **Name** and **Description** fields. All other fields are view-only.

Perform the following steps to edit the fields.

1. Click the **pencil** icon next to the corresponding field or group of fields to edit the fields.
2. To save the edits and/or selections, click the **Checkmark** icon.
3. To cancel the edits and/or selections, click the **cross** icon.

11.5 Selective data distribution

These topics describe concepts and procedures for using Selective Data Distribution (SDD) to replicate data from one Data Engine to another.

- SDD overview (see page 2408)
- SDD use cases (see page 2411)
- SDD user interface (see page 2413)
- Configuring SDD (see page 2415)
- SDD and failover (see page 2418)

11.5.1 Support

SDD is supported across all environments for each supported data source. The data source support matrices can be found below.

- Oracle matrix (see page 1384)
- Oracle EBS matrix (see page 1657)
- SQL Server matrix (see page 2052)
- PostgreSQL matrix (see page 1769)
- SAP ASE matrix (see page 1877)
- SAP HANA matrix (see page 1982)
- IBM Db2 matrix (see page 1153)

11.5.2 Selective data distribution overview

The selective data distribution technology permits the distribution of masked data between Delphix Engines without bringing over the unmasked parent source. These engines must be the identical versions of the Delphix Data as a Service Engine with the Delphix Masking Engine. Otherwise, they can be asymmetric in terms of engine configuration. You can provision VDBs from distributed masked objects, allowing for the geographical distribution of data and remote provisioning.

You can run selective data distribution ad hoc, but it is typically run according to a predefined schedule. After the initial update, each subsequent update sends only the changes incurred since the previous

update. Selective data distribution does not provide synchronous semantics, meaning that the data distributed to the target is only as current as of the most recent update.

11.5.2.1 Selective data distribution features

As virtual appliances, you can backup, restore, replicate, and migrate data objects between Delphix Engines using features of VMWare and the underlying storage infrastructure. In addition to the replication capabilities provided by this infrastructure, selective data distribution permits the distribution of masked data between Delphix Engines. The sources received on a target Delphix Engine do not include the original parent source, thereby making the original source inaccessible from the target.

Selective data distribution is configured on the source Delphix Engine. It first copies a subset of masked VDBs to a target Delphix Engine, then sends incremental updates either manually or according to a schedule. As illustrated below, sensitive data from the Production Data Center is brought into Delphix as a dSource. The Masking Engine then masks the dSource data as a VDB. Synchronously, DxFS redacts the sensitive data within that dSource before sending it across into the Non-Production datacenter using Delphix Replication. Using SDD capability, sensitive data is never exposed because it is protected both at the dSource and VDB layers.

You can use replicated masked VDBs to provision new VDBs on the target Delphix Engine. The provisioned VDBs contain the data in their masked parent and are therefore also considered masked. You can refresh these VDBs to snapshots sent as part of an incremental replication update, as long as you do not destroy the parent object on the replication source. For more information, see Provisioning from a Replicated Data Sources or VDBs. (see page 2403)

During replication, replicated masked VDBs are maintained in an alternate replica and are not active on the target side. The failover of a selective data distribution replica is not supported.

11.5.2.2 Selective data distribution details

When you select masked objects for selective data distribution, the engine will automatically include any dependencies, such as environments, associated with the VDB. The parent dSource and any parent VDBs are not included automatically. The data associated with parent objects are selectively included for disk space efficiency, but data in the parent dSource and VDBs that the masked VDB does not need are excluded.

During replication, the Delphix Engine will negotiate an SSL connection with its server peer to use SSL_RSA_WITH_RC4_128_MD5 as the cipher suite, and TLSv1 as the protocol.

Only database objects and their dependencies are copied as part of a selective data distribution operation, including:

- Masked VDBs
- Environments
- Environment configuration (users, database instances, and installations)

The following objects are not copied as part of a selective data distribution operation:

- Parent dSources of masked VDBs - The storage blocks for non-sensitive dSource data are sent from the source to the target replication host. This storage is displayed under Held Space, for more information see An Overview of Held Space. (see page 782)
- Groups of the parent dSources
- Users and roles
- Policies

- VDB (init.ora) configuration templates
- Events and faults
- Job history
- System services settings, such as SMTP

11.5.2.3 Resumable selective data distribution

A single selective data distribution instance can fail for a number of environmental and internal reasons. However, using the Resume feature, you can restart selective data distribution from an intermediate point; no data is retransmitted. Selective data distribution is resumable across machine reboot, stack restart, and network partitions. The resumable replication feature is fully automated and does not require or allow any user intervention.

For example, you can resume a large, time-consuming initial distribution or incremental update after it is interrupted. Suppose a selective data distribution profile has already been configured from a source to a target. A large, full send from the source begins that is expected to take weeks to complete. Halfway through, a power outage at the data center that houses the source causes the source machine to go down and only come back up after a few hours. On startup, the source will detect that a selective data distribution was ongoing, automatically re-contact the target, and resume the distribution where it left off. In the user interface (UI) on the source, the same selective data distribution send job will appear as active and continue to update its progress. However, in the UI of the target, a new distribution receives job will appear, although it will track its progress as a percentage of the entire replication.

Selective data distribution will not resume after failures that leave the source and target connected. For example, if a storage failure on the target, such as an out-of-space error, causes a distribution to fail, then the source and target remain connected. As a result, the Engine will discard state data associated with the failed Replication operation.

11.5.2.4 Selective data distribution restrictions

- Only masked VDBs can be added to a selective data distribution spec. You cannot add dSources, groups, or the entire domain.
- Only masked VDBs with a Snapshot Policy of None should be added to a selective data distribution spec.
- Unmasked VDBs cannot be added to a selective data distribution spec.
- VDBs that undergo selective data distribution and their children cannot be selectively redistributed to another target.
- You cannot go to the target engine and create a selective data distribution spec that includes VDBs that are present because of selective data distribution. However, you can replicate this data using a traditional replication spec.

Best practices for using Selective Data Distribution are described in the Selective Data Distribution Best Practices[750] knowledge base article.

11.5.2.5 Selective data distribution supported platforms

Selective Data Distribution supports the following platforms:

750 https://support.delphix.com/Delphix_Masking_Engine/Masking_Engine/Selective_Data_Distribution_Best_Practices

- Oracle
- Microsoft SQL Server
- SAP ASE (Sybase)
- Db2

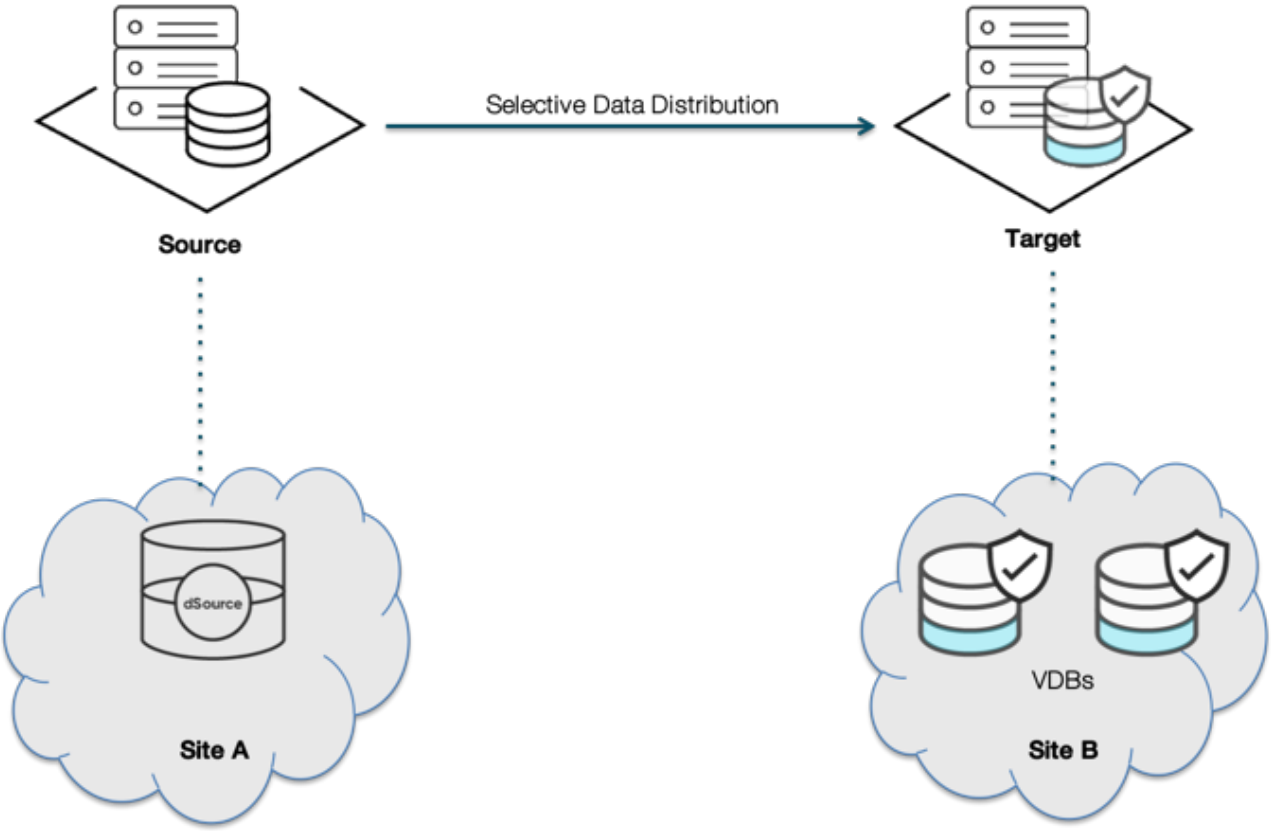
11.5.3 Selective data distribution use cases

Selective data distribution is a flexible tool that allows you to move dSources and virtual databases (VDBs) between Delphix Engines. Selective data distribution relies on the replication infrastructure to transmit masked data, as such selectively distributed data has much of the same capabilities as replicated data. These topics describe several different use cases for selective data distribution.

11.5.3.1 Geographically distributed development

The Delphix Engine allows you to provision VDBs from distributed dSources and VDBs, as described in Provisioning from Replicated Data Sources or VDBs (see page 2403). This allows dSources that are linked in a single central location to be geographically distributed so that developers can provision VDBs remotely without having to sync from the source database in multiple locations. Selective distribution ensures that masked VDBs are sent to the target without transmitting the original unmasked source. As such, the original unmasked source will not be accessible on the target.

Selective data distribution does not support failover. In this environment, the link between the source and the target is never broken, except when remote VDBs need to be preserved. You can refresh remote VDBs as long as the parent objects continue to exist on the source. If they are deleted, then remote VDBs will continue to function but cannot be refreshed.



Because there is no failover, this technology can support more complex topologies such as 1-to-many and many-to-1. Chained distribution (replicating from Site A -> Site B -> Site C) is not supported.

11.5.3.2 Best practices

For geographical distribution, follow these best practices:

- Because each replication stream induces load on the source system:
  - Minimize the number of simultaneous replication updates
  - If possible, avoid heavy VDB workloads on the source
- On the target, the provision only from sources that are effectively permanent. If a source is deleted remote VDBs can no longer be refreshed.
- Provision additional storage capacity on the target
  - Remotely provisioned VDBs can consume shared storage on the target even when the parent is deleted on the source

11.5.3.3 Migration

You cannot use selective data distribution for data migration. Full replication is needed for data migration.

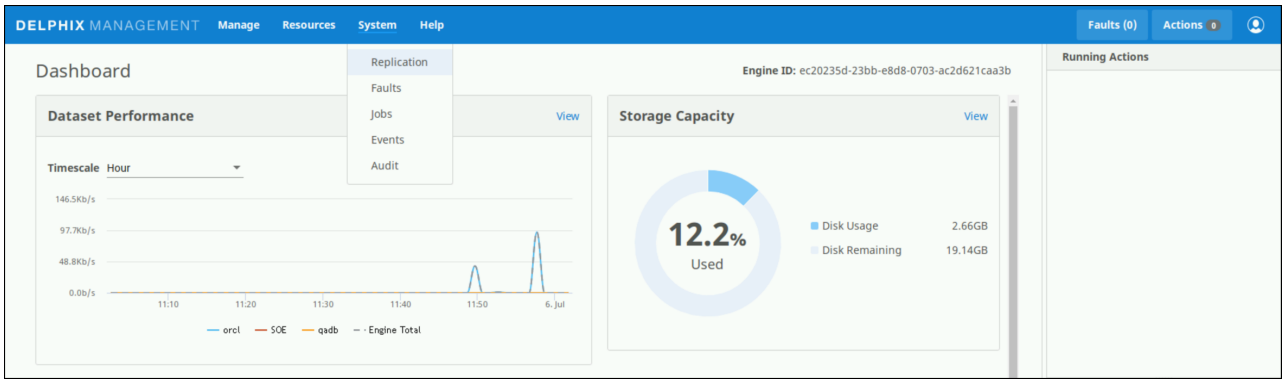
11.5.4 Selective data distribution user interface

11.5.4.1 Sources for selective data distribution

The **Replication Profile** section continues to handle Replication functions, but also allows for the configuration of **Selective Data Distribution (SDD)** profiles. This makes it possible to distribute objects from a single source to multiple targets. Each profile defines the set of data objects and the associated configuration between a single source and target.

11.5.4.2 Replication/selective data distribution section

This section is used to illustrate the capabilities in the **SDD Profiles** section. From the Delphix Management portal, navigate to **System** > **Replication**.



Select the plus icon to add a profile, this will open a new window for configurations. Below the **Replication Profiles** header and the plus icon is a list of existing Replication Profiles and SDD profiles. Select a profile in this list to view its details.

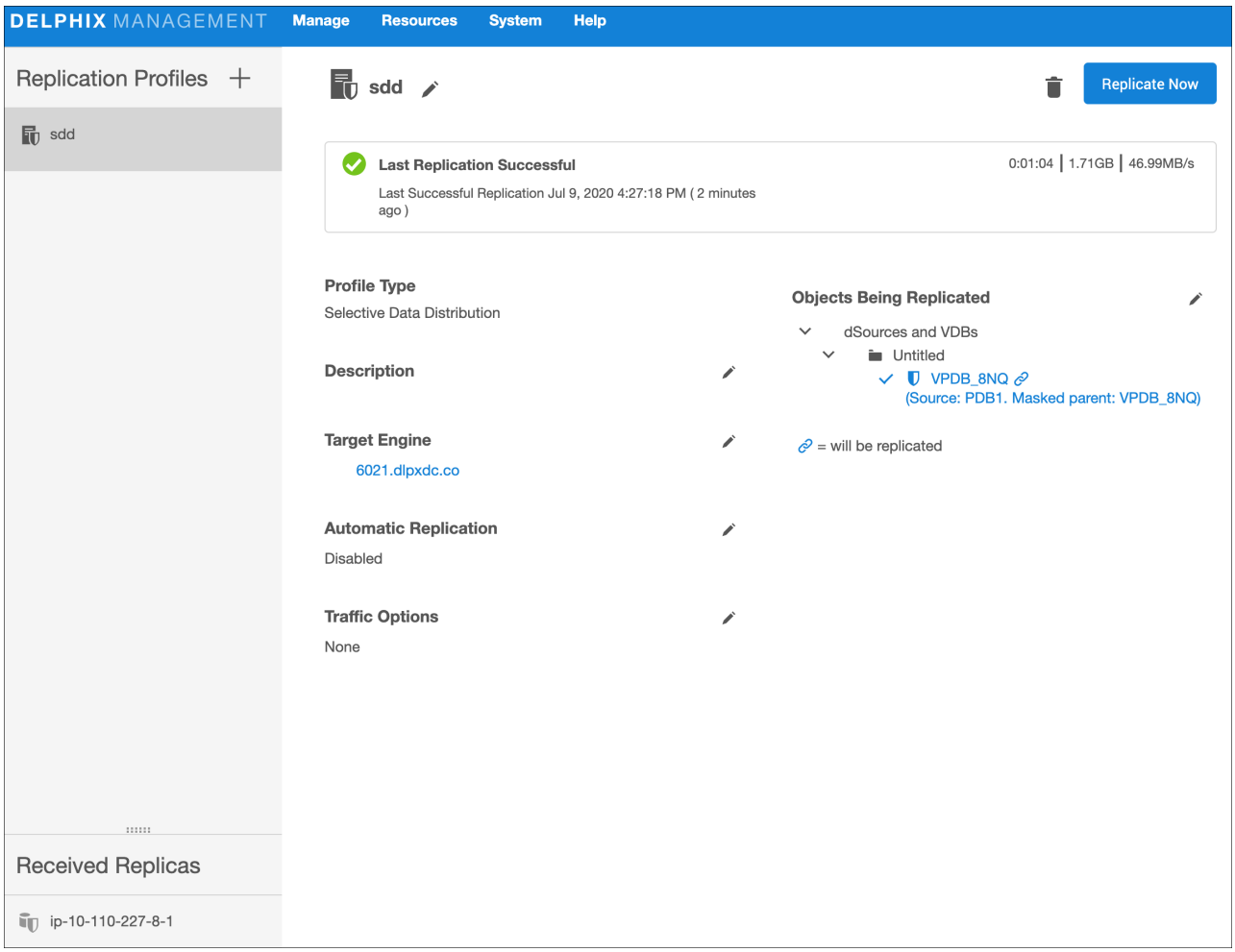
The **Received Replicas List** shows all replicas received to this Delphix Engine, including normal Replication and SDD replicas. Click a **replica** in this list to view its details.

A **Status Box** appears when a profile is selected from the list that shows the distribution status of the selected profile. This includes the result of the most recent or current distribution event and statistics for the distribution run, such as data transferred, duration, and average throughput. The **Profile Type** shows the type of the selected profile or replica.

There are several configuration options available for the selected profile or replica listed below the Profile Type. These sections, in addition to the profile name itself, can be modified by selecting the pencil icon.

- Description – Free text description of the profile
- Target Engine – The Delphix Engine on the receiving end of this data distribution pair
- Automatic Replication – If enabled, shows the frequency and time that regular distribution will be run
- Traffic Options – Summarizes the traffic options with which this profile has been configured

The **Objects Being Replicated** shows all of the masked objects selected for distribution in this SDD profile. The **Replicate Now** button initiates the distribution process and the **bin icon** to the left allows for current profile deletion.



11.5.4.3 Create new profile

1. In the left-hand navigation section, click the **+** icon.
2. Enter the **name** of the SDD profile and an optional **description**.
3. For **Type**, select **Selective Data Distribution**.
4. For **Target Engine**, enter the **hostname** or **IP address** for the target Delphix Engine.
5. Enter the **username** and **password** of a user who has Delphix Admin-level credentials on the target Delphix Engine. If the username and password change on the target Delphix Engine, these settings must be updated on the source Delphix Engine.

6. Automatic replication is disabled by default, which means the distribution updates must be triggered manually via the **Replicate Now** button. To enable automatic distribution, click the **Enabled** checkbox.
7. In the **Automatic Replication** field (if enabled), enter the **Frequency** and **Starting Time** for distribution updates to the target Delphix Engine.
   Note: Automatic replication uses Quartz for scheduling. Starting with Delphix version 4.2, the Quartz-formatted string is editable via the **Advanced** option. An example **Cron String** would be seen as: `0 0 0 * * ?`
8. Under **Traffic Options**, select to **Encrypt** traffic or **Limit bandwidth** during distribution updates.
9. In the right-hand column under **Objects Being Replicated**, select the **boxes** next to the objects to distribute. Only masked VDBs can be used and multiple masked VDBs cannot be added to the same profile.
10. Click **Create Profile** to submit the new profile. This saves the SDD profile details. Leaving the **Create** page prior to submitting the profile will result in the draft being discarded.

11.5.5 Configuring selective data distribution

This topic describes how to configure Selective Data Distribution between Delphix Engines. Selective data distribution relies on the replication infrastructure to transmit masked data, as such configuring selective data distribution is similar to configuring replication.

11.5.5.1 Prerequisites

- The replication target can be on the same or newer version than the replication source. Starting 6.0.10.0, replication operation can not go beyond engine versions released more than 12 months apart. For more information, see Replication Overview. (see page 2368)
- The target Delphix Engine must be reachable from the source engine.
- The target Delphix Engine must have sufficient free storage to receive the replicated data.
- The user must have administrative privileges on the source and the target engines.

11.5.5.2 Configuring the network

Replication and selective data distribution operate using a private network protocol between two Delphix Engines. Apart from standard network considerations for performance, no additional configuration is required for replication. Replication and selective data distribution can run over dedicated networks by configuring routing to direct traffic destined for the target IP address over a specific interface. The selective data distribution process can recover from transient network outages, but extended outages may cause the process to start from the previous update.

The selective data distribution network protocol uses TCP port 8415. If there is a firewall between the source and target that is blocking this port, then there are two possible solutions:

- Enable port 8415 on the firewall in order to allow connections to this port from the source to the target.

- Selective data distribution can connect through a SOCKS proxy if one exists. Configure the SOCKS proxy address and port by connecting to the command-line interface (CLI) as a system administrator and navigating to "service proxy" to update the socks configuration. Example:

```
dlpx-engine> service proxy
dlpx-engine service proxy> update
dlpx-engine service proxy update *> set socks.enabled=true
dlpx-engine service proxy update *> set socks.host=10.2.3.4
dlpx-engine service proxy update *> set socks.username=someuser
dlpx-engine service proxy update *> set socks.password=somepassword
dlpx-engine service proxy update *> commit
dlpx-engine service proxy> get
    type: ProxyService
    https:
        type: ProxyConfiguration
        enabled: false
        host: (unset)
        password: (unset)
        port: 8080
        username: (unset)
    socks:
        type: ProxyConfiguration
        enabled: true
        host: 10.2.3.4
        password: ********
        port: 1080
        username: someuser
```

Note that SOCKS port 1080 is used by default, but you can override it.

11.5.5.3 Configuring the source Delphix engine

1. On the source Delphix Engine, click **System**.
2. Select **Replication**.
3. In the left-hand navigation section, click **Create Profile**.
4. Enter the **name** of the replication profile and an optional **description**.
5. Select type **Selective Data Distribution**.
6. For **Target Engine**, enter the **hostname** or **IP address** for the target Delphix Engine.
7. Enter the **username** and **password** of a user who has Delphix Admin-level credentials on the target Delphix Engine. If the username and password change on the target Delphix Engine, you must update these settings on the source Delphix Engine.
8. By default, automatic replication is disabled, meaning that you must trigger replication updates manually. To enable automatic replication, click the **Enabled** checkbox.

9. In the **Automatic Replication** field, enter the **Frequency** and **Starting Time** for replication updates to the target Delphix Engine. Once you have entered and saved your replication settings, you will also see an option to trigger replication immediately with the **Replicate Now** button.
   Note:

   Quartz scheduling Automatic replication uses Quartz for scheduling. Starting with Delphix version 4.2, the Quartz-formatted string is editable via the **Advanced** option.

10. Under **Traffic Options**, select whether you want to **Encrypt** traffic or **Limit bandwidth** during replication updates. **Note**:

    Encrypting Traffic By default, replication streams are sent unencrypted. This provides maximum performance on a secure network. If the network is insecure, encryption can be enabled. Note that encrypting the replication stream will consume additional CPU resources and may limit the maximum bandwidth that can be achieved.

    Note:
    Limiting Bandwidth By default, replication will run at the maximum speed permitted by the underlying infrastructure. In some cases, particularly when a shared network is being used, replication can increase resource contention and may impact the performance of other operations. This option allows you to specify the maximum bandwidth that replication can consume.

11. In the right-hand column, under **Objects Being Replicated**, click the **checkboxes**next to the objects you want to replicate.

    Note:
    Selected Objects
    a. You can only select masked VDBs for selective data distribution.
    b. Only one masked VDB per parent dSource can be selected. Multiple sibling masked VDBs that share the same unmasked dSource cannot be selected.
    c. The parent dSource or VDB (and any parents in its lineage) are NOT automatically included. Some of the data from the parent may be included for disk space optimization. In addition, any environments containing database instances used as part of a replicated VDB are included as well.
    d. When replicating individual VDBs, only those database instances and repositories required to represent the replicated VDBs are included. Other database instances that may be part of the environment, such as those for other VDBs, are not included.

12. Click **Create Profile** to submit the new profile. This saves the replication profile details. If you leave the **Create** page prior to submitting the profile, the draft replication profile will be discarded.

Note:
Configuring Replication and Multiple Target Engines through the CLI
You can also configure replication on the Source Delphix Engine by using the replication spec in the command line interface. For more information, see the topics under CLI Cookbook: Replication.[751]

Note:

751 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/cli-cookbook-replication

Enabling Configuration of Multiple Replication Profile
Learn how to configure and use functionality for multiple replication profiles on the source using the replication profiles in the Replication User Interface.[752]

11.5.5.4 Configuring the target Delphix engine

No additional configuration on the target is needed. Selectively distributed objects will appear in an alternate replica that mirrors the original object layout. To view these replicas:

1. Click **System**.
2. Select **Replication**.
3. Review items listed under **Received Replicas**.

Alternatively, you can view replicas under **namespace** in the command-line interface (CLI). All replicated objects are read-only. For more information about managing replicas, see Selective Data Distribution and Failover. (see page 2418)

Multiple sources can replicate to the same target, allowing for the flexible geographical distribution of data. You can create and manage objects on the target server without affecting subsequent updates. However, you cannot use selective data distribution for disaster recovery.

11.5.6 Selective data distribution and failover

Selective data distribution recreates objects on the target system in a replica that preserves the object relationships and naming on the target server without interfering with active objects on the system. Objects within a replica are read-only and disabled. They cannot be failed over. However, you can use virtual databases (VDBs) within a replica as the source for provisioning new VDBs.

11.5.6.1 Replicas

A replica contains a set of replicated objects. These objects are read-only and disabled while replication is ongoing. To view replicated objects:

1. Click **System**.
2. Select **Replication**.
3. Under **Received Replicas**, select the replica. On this screen, you can browse the contents of replicas or delete individual replicas. As described in the Selective Data Distribution Overview, (see page 2408) VDBs and environments are included within the replica.

You can also view replicated objects under **namespace** in the command-line interface (CLI).

Deleting a replica will sever the link with the replication source. Subsequent incremental updates will fail, requiring the source to re-establish replication.

752 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/replication-user-interface

Multiple replicas can exist on the system at the same time. Active objects can exist in the system alongside replicas without interfering with replication updates. VDBs within a replica can also be used as a source when provisioning. For more information, see Provisioning from Replicated Data Sources or VDBs. (see page 2403)

11.5.6.2 Failover

Selective data distribution does not support the failover of replicas.

12 Delphix self-service

> **Important update**
> Delphix Self-Service has been deprecated and will reach its end of life in September 2025. Visit our community blog[753] for more information and guidance on transitioning to Data Control Tower's improved developer experience.

Delphix Self-Service eliminates data related bottlenecks by extending virtual data on demand to application development teams as a self-service.

Delphix self-service admin guide (see page 2420)

Delphix self-service data user guide (see page 2479)

12.1 Delphix self-service admin guide

Getting started with Delphix self-service (see page 2422)

Welcome to Delphix Self-Service The Delphix Engine has greatly improved the speed at which end users can get the data that they need. While end-users reap the benefits, they do not typically interact with the Delphix Engine directly, nor are they n... (see page 2422)

Updated on : 25 May 2023

Delphix self-service concepts (see page 2423)

Data Sources A data source can be a database, an application, or a set of unstructured files. Engine Administrators configure the Delphix Engine to link to data sources, which pulls in the data of these sources. The Delphix Engine will periodical... (see page 2423)

Updated on : 25 May 2023

Navigating the Delphix self-service admin interface (see page 2425)

The following screenshots provide a roadmap for how to navigate the primary screens and places a user will go within the Delphix Self-Service application. The application includes screens such as the Management Overview , Data Platform Management ... (see page 2425)

Updated on : 25 May 2023

753 https://community.delphix.com/blogs/nicholas-mathison/2024/02/08/delphix-end-of-support-for-delphix-self-service

Understanding data templates (see page 2431)

Data Templates: An Overview A data template represents a collection of data sources that you can provision to a user. A data source can be a dSource, VDB, or vFiles. These sources can be used in multiple data templates. Once you have created a dat... (see page 2431)

Updated on : 25 May 2023

Understanding how to manage data template details (see page 2439)

The Data Template Details Page In the Overview page, under the Templates , click the data template's name . This will direct you to the templates details page. You can use this page to view and configure details of an individual data temp... (see page 2439)

Updated on : 25 May 2023

Understanding data containers (see page 2443)

Delphix Self-Service Data Container Overview Data containers are provisioned from data templates by administrators and assigned to a user. A data container represents a socket that is capable of making any data within the data template accessible. ... (see page 2443)

Updated on : 25 May 2023

Delphix self-service data container activities (see page 2452)

Configuring Data Containers in Delphix Self-Service A data container is comprised of a set of virtual databases (VDBs), where each VDB is a direct child of the dSource, VDB, or vFiles in the data template's data sources. Delphix Self-Service does... (see page 2452)

Updated on : 25 May 2023

Using masked data sources with Delphix self-service (see page 2457)

SDD Overview You can now replicate masked data in a VDB directly to a target Delphix Engine without transmitting the unmasked data in its parent source. This is called selective data distribution (SDD). Although you can run selecti... (see page 2457)

Updated on : 25 May 2023

Understanding Delphix self-service user management (see page 2462)

User Management Activities This document describes the process of creating a user and assigning that user to a data container. It also provides an overview of the User Details page. Creating a User Follow the same process when creating a new us... (see page 2462)

Updated on : 25 May 2023

Working with multiple container owners (see page 2465)

Delphix Self-Service administrators can designate multiple users as owners of a single data container. These users all share access to the same data container which means actions taken by one user will impact all users on the same data container. Fo... (see page 2465)

Updated on : 25 May 2023

Understanding bookmarks (see page 2469)

Bookmarks Overview Bookmarks are a way to mark and name a particular moment of data on a timeline. You can restore the active branch's timeline to the moment of data marked with a bookmark. You can also share bookmarks with other users, which all... (see page 2469)

Updated on : 25 May 2023

Understanding Delphix self-service usage management (see page 2470)

Usage Management Dashboard Overview Data templates are comprised of dSources, virtual databases (VDBs), and vFiles. These data sources are controlled by the standard policies configured in the Management application of the Delphix Engine. ... (see page 2470)

Updated on : 25 May 2023

12.1.1 Getting started with delphix self-service

12.1.1.1 Welcome to Delphix self-service

The Delphix Engine has greatly improved the speed at which end users can get the data that they need. While end-users reap the benefits, they do not typically interact with the Delphix Engine directly, nor are they necessarily even aware that they are using it. End users most commonly file tickets for data management operations and wait for the tickets to be serviced by their IT organization. Delphix data management workflows allow database administrators (DBAs) to respond to these tickets much more quickly and reliably, but DBAs are often overloaded, and resolving high-priority issues takes precedence over requests from users. Requiring interactions between users and IT for every data operation is inefficient and can lead to unwanted delays.

The goal of Delphix Self-Service is to create a clear separation of IT infrastructure and data management. As with the current Delphix Engine, IT administrators and DBAs continue to control decisions about how resources such as virtual databases (VDBs) and vFiles are allocated. However, with Delphix Self-Service, administrators can also assign these resources directly to a user. A user has the ability to control what data their container presents, even though the details of the physical resources are hidden from them. This separation of roles empowers Delphix Self-Service users to get the data they need, when they need it while providing administrators with the controls to ensure resources are accounted for appropriately.

12.1.1.2 User roles and permissions

Delphix Self-Service has two types of users:

12.1.1.2.1 Admin user

Admin users have full access to all report data and can configure Delphix Self-Service, additionally, they can:

- use the Delphix Engine to add/delete users
- change tunable settings
- add/delete tags

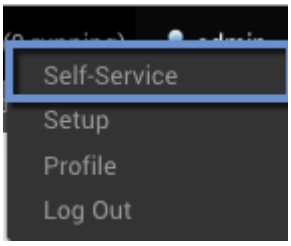
- create and assign data templates and containers

12.1.1.2.2 Data user

Data users have access to production data provided in a data container. The data container provides these users with a playground in which to work with data using the **Self-Service Toolbar**.

12.1.1.3 Login

1. Access Delphix Self-Service by opening a web browser and using the **IP address** or **DNS qualified hostname**.
2. Login with the **Delphix Admin User ID** and **Password** provided for you.
3. Upon successful login, you should be able to click on your username at the top right corner and select a Self-service view from the dropdown menu. This will open the Delphix Self-Service interface.



12.1.2 Delphix self-service concepts

12.1.2.1 Data sources

A data source can be a database, an application, or a set of unstructured files. Engine Administrators configure the Delphix Engine to link to data sources, which pulls in the data of these sources. The Delphix Engine will periodically pull in new changes to the data, based on a specific policy. This, in turn, begins building a custom timeline for each data source. Additionally, the Delphix Engine can rapidly provision new data sources that are space-efficient copies, allowing users to work in parallel without impacting each other.

12.1.2.2 Data templates

Data templates are the backbone of data containers. They are created by the Engine Administrator and consist of the data sources that users need in order to manage their data playground and their testing and/or development environments. Data templates serve as the parent for a set of data containers that the administrator assigns to users. Additionally, data templates enforce the boundaries for how data is shared. Data can only be shared directly with other users whose containers were created from the same parent data template.

12.1.2.3 Data containers

A data container allows data users to access and manage their data in powerful ways. Their data can consist of application binaries, supporting information, and even the entire database(s) that underlie it.

A data container allows users to:

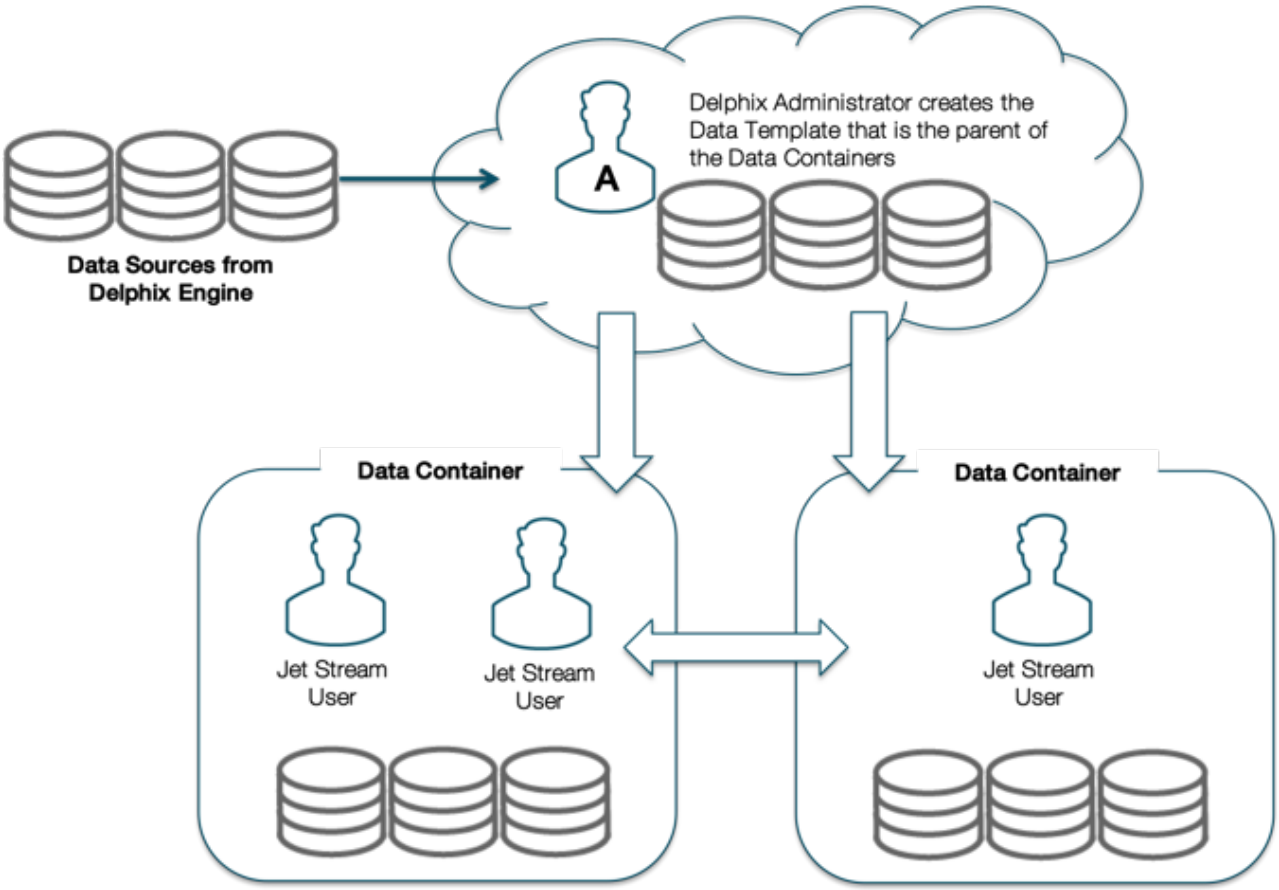
- Undo any changes to their application data in seconds or minutes
- Have immediate access to any version of their data over the course of their project
- Share their data with other people on their team, without needing to relinquish control of their own container
- Refresh their data from production data without waiting for an overworked DBA

A data container consists of one or more data sources, such as databases, application binaries, or other application data. The user controls the data made available by these data sources. Just like data sources in a template, changes that the user makes will be tracked, providing the user with their own data history.

The **Data Container Interface** lets users view the details and status of their data container and its associated data sources, as well as manipulating which data is in those sources. The **Data Container Interface** includes a section called the **Data Container Report Panel**, which displays details about each source, including the connection information needed to access it – for example, the java database connectivity (JDBC) string for a database. This connection of information is persistent and stable for the life of the data container, regardless of what data the resources are hosting.

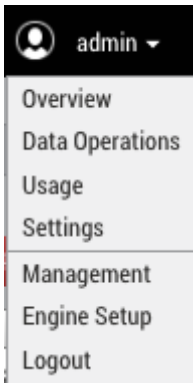
12.1.2.4 Delphix self-service data flow

The Delphix Self-Service data flow diagram below demonstrates how a Delphix Self-Service data user accesses data sources. Data sources are connected to a Delphix Engine, which is controlled by the Engine Administrator. The Engine Administrator will connect all data sources that developers and quality assurance (QA) teams need to a data template. This data template acts as a parent source to create the data containers that the administrator will assign to data users. Data sources flow from the Delphix Engine into a data template and downstream into a data container, where a data user or users will use the data sources to complete tasks. The data container acts as a self-contained testing environment and playground for the data user. Additionally, data users are able to set, bookmark, and share data points in their container with other data users of other data containers, as long as all the data containers were created from the same parent data template.



12.1.3 Navigating the Delphix self-service admin interface

The following screenshots provide a roadmap for how to navigate the primary screens and places a user will go within the Delphix Self-Service application. The application includes screens such as the **Management Overview**, **Data Platform Management**, **Users and Permissions**, and **Data User Management**.



12.1.3.1 Overview screen

The **Overview** screen is the homepage for Delphix Self-Service. To reach this page from the Management application as an admin user, select **Self-Service** in the user login drop-down.

12.1.3.1.1 Data management overview page

On this page, you can:

1. You can delete one or more templates at one time by selecting the checkboxes and clicking the **Delete** button.
    a. You cannot delete replicated templates or templates that have containers.
2. From this column, you can see which replica the template belongs to if it is a replicated template. You can sort by this column and see all templates from the same replica.
    a. This column lists replica names, not replication sources. You can view a list of replicas in the Management application at **System** > **Replication** > **Received Replicas**.
3. The number of containers associated with the template.
4. Shows the last time the template was updated.
5. Create a new data template.

12.1.3.1.2 Users screen

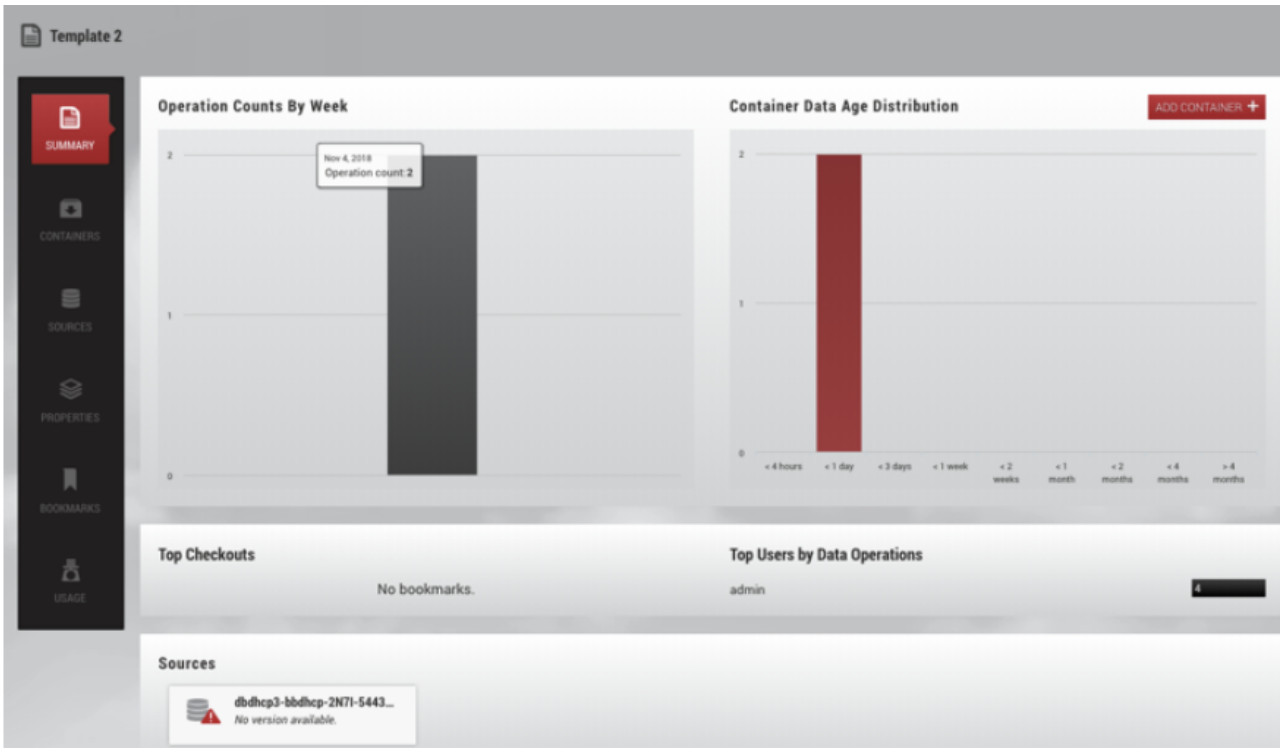
By selecting the Users Button you can view users, whom you can then assign to data containers that you create from existing data templates.

1. List of all configured users. You can click on a user to see their details, such as what containers they are assigned to.
2. The number of containers assigned to the user.
3. The Role column indicates whether a user is a **Jet Stream Only** user or a **Delphix Admin**.
4. Lists the associated email address for the user.
5. Select this link to add a new user or manage users.

12.1.3.2 Data template management

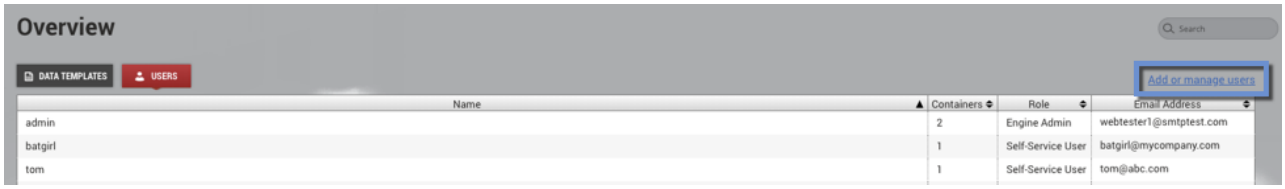
The **Data Template Management** page contains a view panel of 6 tiles on the left-hand side of the screen. Each tile reports on a variety of useful information, such as user activity, data sources, data capacity, specific details about data containers, and data templates. They also help you navigate to areas where you can

complete specific tasks, such as creating a new template or container, working with data timeflows, assigning users to containers, and bookmarking important points in time.

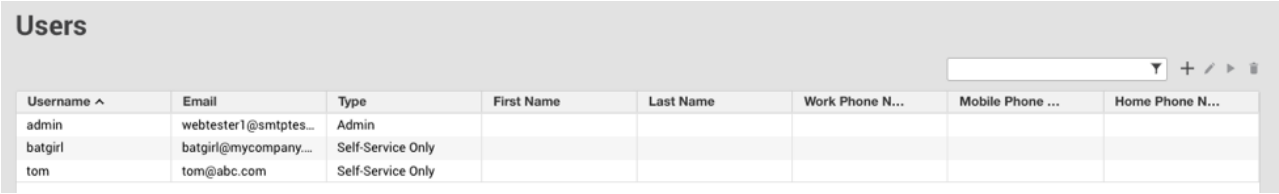


12.1.3.3 User roles in admin App

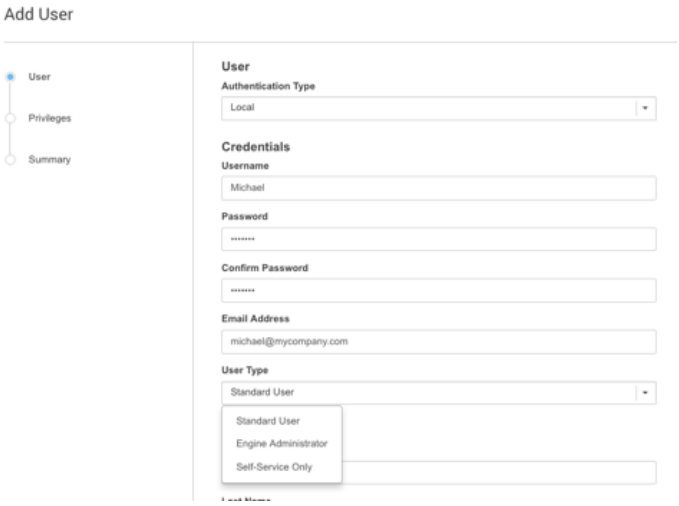
To add or manage user roles and permissions, select the **Add or manage users** link from the **Users Overview** screen.



The **Users** screen in the Management application will open. From this screen, you can add a new user, edit a user, suspend a user, or delete a user.



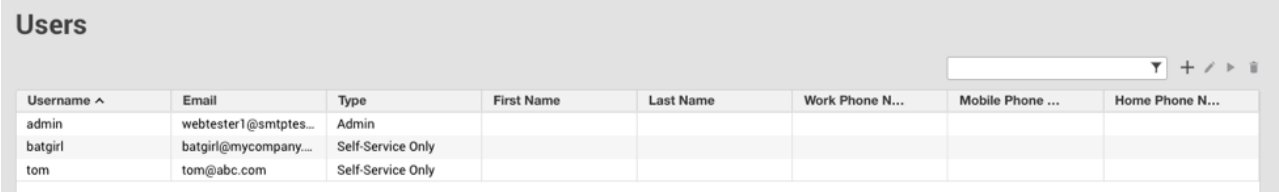
Selecting the **+** icon opens the **Add User** window. Use this window to add a user and select the user role and permissions.



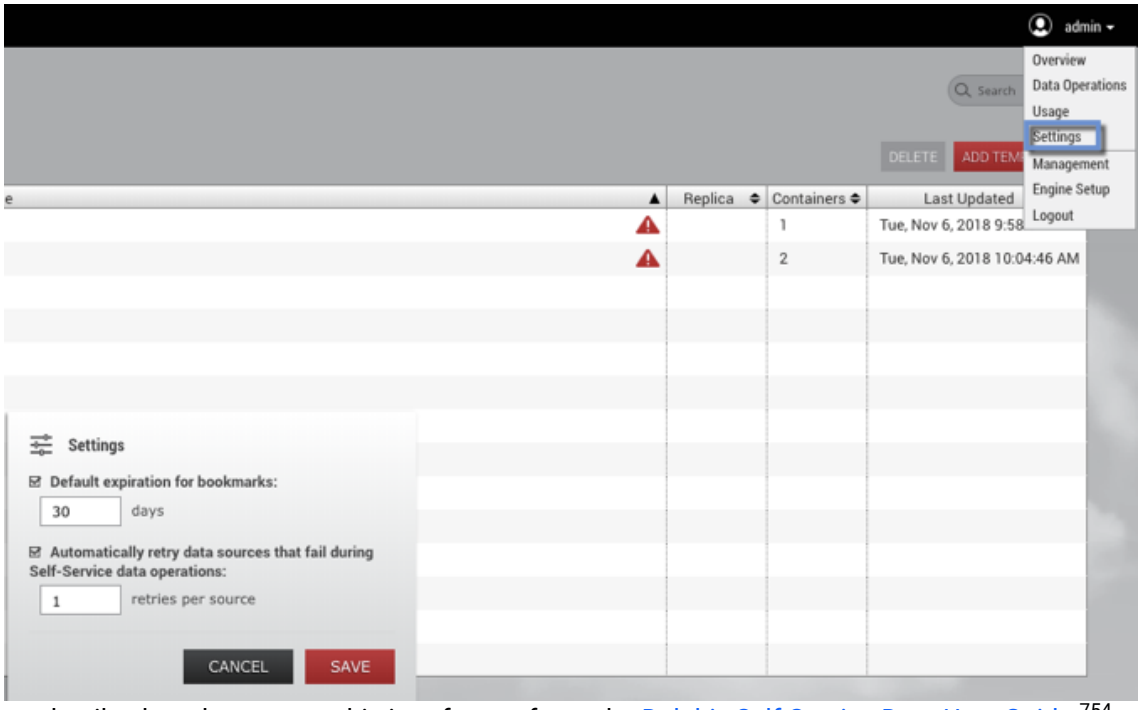
12.1.3.4 Data operations interface for Delphix administrators

The **Overview** page is the only interface to which Delphix Self-Service data users have access and with which they interact. The user interface is the environment in which a data user works with data in an assigned data container, using data sources from a data template.

As shown below, the **Data Operations** screen contains a list of available templates. Select a template to view details.



The following screen allows users to complete tasks using self-service operations.

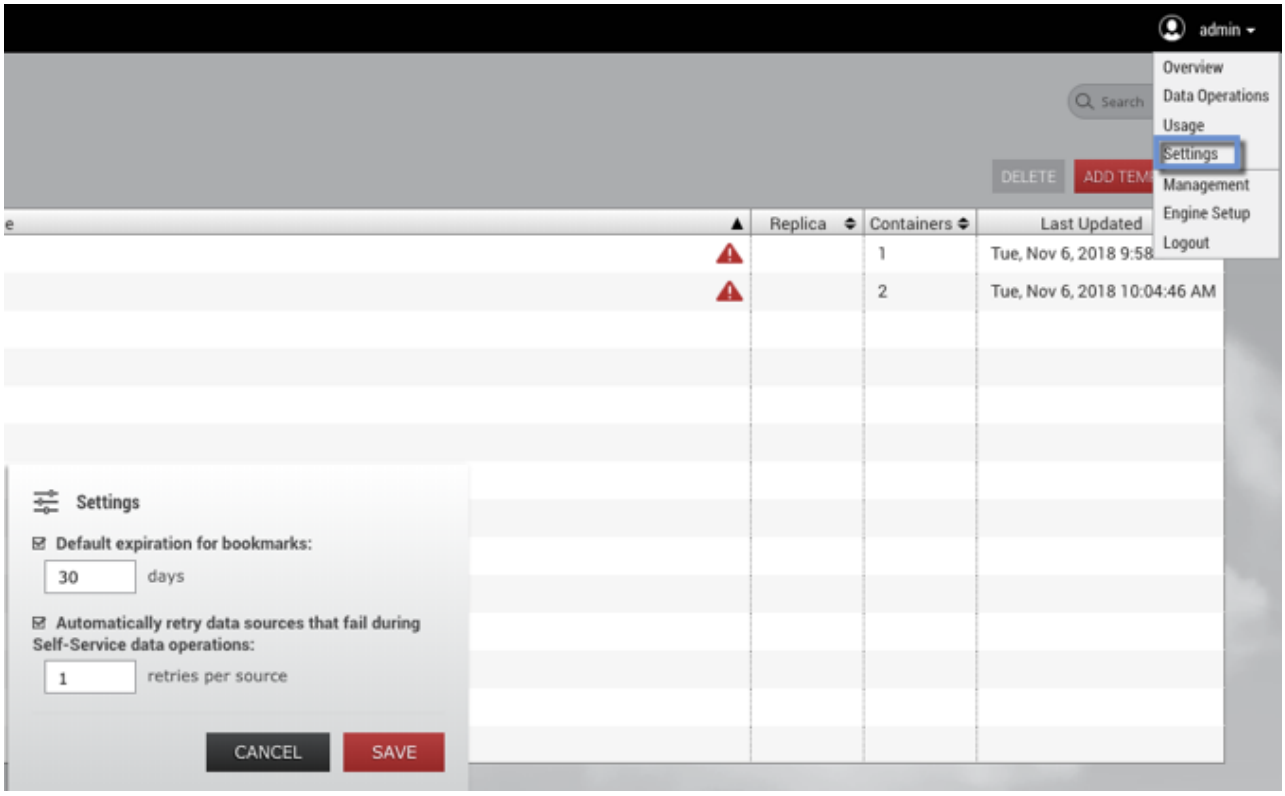


For more details about how to use this interface, refer to the Delphix Self-Service Data User Guide.[754]

12.1.3.5 Administrator settings

You can open the **Administrator Settings** dialog from the user drop-down menu. Here you can edit various settings that apply globally to all Delphix Self-Service users.

754 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/delphix-self-service-data-user-guide



12.1.3.5.1 Default bookmark expiration

You can set a value that controls the default expiration time, in days, for Bookmarks. This setting only applies to new bookmarks that are created through the Delphix Self-Service application, not the CLI or API. Note that this only controls the default selection; users can still disable expiration or pick a different date for a specific bookmark if they wish. This setting is disabled by default.

12.1.3.5.2 Automatic retry for data operations

To make operations on data containers more robust, Delphix Self-Service supports automatically retrying failed sources during data operations. You can specify a maximum number of retry attempts so that if an operation fails on any individual data source within a data container, it will be automatically retried until it succeeds, or until the retry limit is reached.

Automatic retry applies to any Delphix Self-Service operation that changes the data in the data container, such as Refresh, Restore, Reset, or Create Branch. This setting can be especially useful in scenarios where there are a large number of sources in a data container, and some sources fail to update the first time. If the reason for the failures was intermittent, an automatic retry may allow the sources that failed to succeed, and the operation can still complete successfully. The default number of retry attempts is **1**.

To change the number of automatic retries:

1. Click the user icon in the upper right-hand corner of the screen.
2. From the drop-down menu, select **Settings**.

3. In the field **Automatically retry data sources...**, enter a new maximum; alternatively, use the arrows to increase or decrease the number.
4. Click **Save**.

12.1.4 Understanding data templates

12.1.4.1 Data templates: an overview

A data template represents a collection of data sources that you can provision to a user. A data source can be a dSource, VDB, or vFiles. These sources can be used in multiple data templates. Once you have created a data template, the set of data sources associated with it is fixed; you cannot add data sources to an existing template, nor can you remove data sources from it. In addition to data sources, you can define the set of metadata that is relevant for a given template – for example, notes, descriptions, names for sources that are relevant to an end-user, and other configuration details. Once you have created the template, it provides a stencil for provisioning data containers. This, in turn, enables users to have self-service access to a space-efficient copy of the data sources defined in the data template.

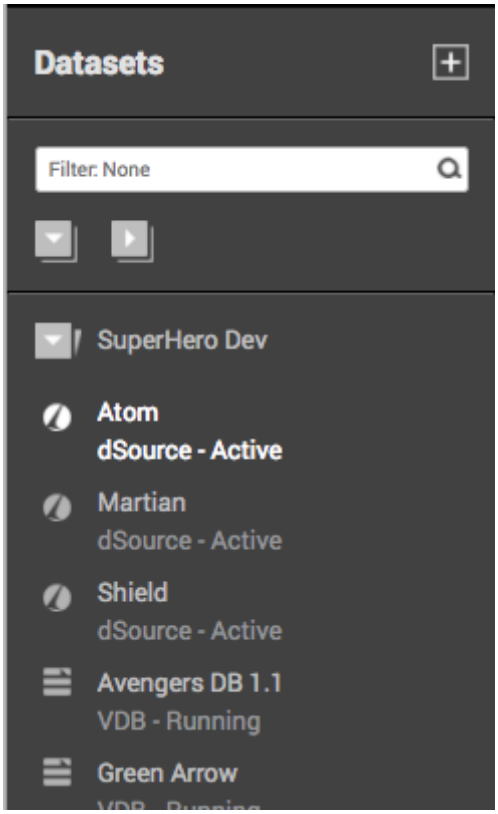
12.1.4.2 Data template activities

Data templates are managed by a Delphix admin. The admin can provision data containers from the data template and assign a data container to an end-user. The admin can also create bookmarks on the data template timeline in order to mark meaningful points in time.

When creating a data template, it is important to consider the set of users who will own data containers provisioned from it. In Delphix Self-Service, templates effectively define the boundaries of the data that users can share directly with each other. Only owners of data containers created from the same data template are able to share data using bookmarks.

12.1.4.2.1 Creating a Data template and adding data sources

A data template consists of an arbitrary set of dSources, virtual databases (VDBs), and vFiles. These are created and managed in the **Delphix Management** interface and can be used in Delphix Self-Service as data sources. You can use any data type supported by the Delphix Engine as a data source. For more information, refer to the Linking/Provisioning documentation for the standard Delphix Engine. The following is an example of the many kinds of data sources you can use to create a data template.



When adding data sources to the data template, it is important to consider whether there are any dependencies between them. For example, do data operations need to begin with a VDB (database) source before the same operation occurs on vFiles (application binary)? Or can data operations be performed in parallel with each of the data sources? The data source dependencies are by default synced together in parallel during any data operation, including starting the data container and its sources. When working with specific ordering constraints, such as with Oracle EBS, you can set up and configure the ordering sequence for each data source.

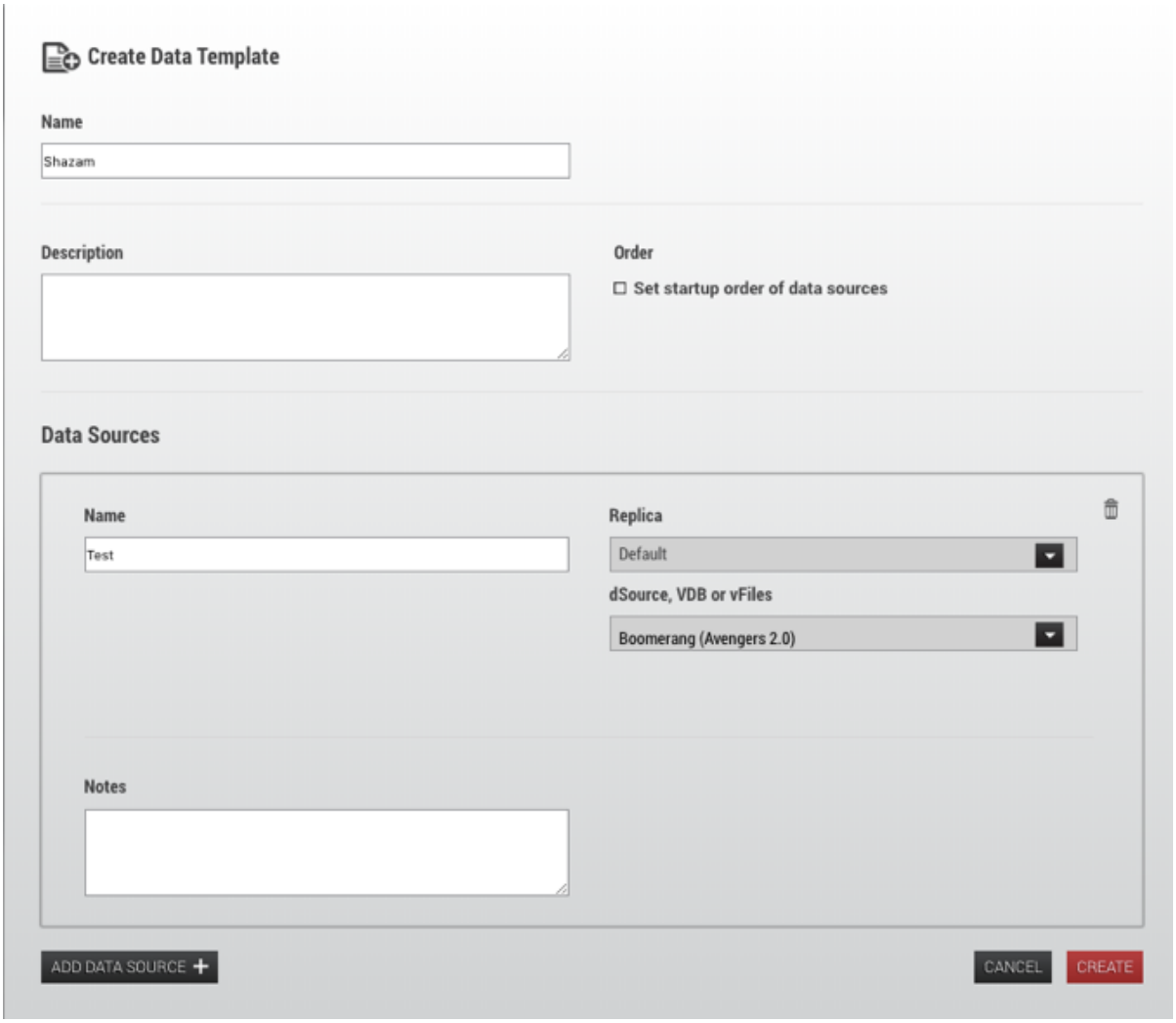
12.1.4.2.2 Procedure for creating a data template

To create a data template:

1. From the drop-down menu in the upper right-hand corner of the Delphix Management application, select **Self-Service**.
2. On the **Overview** page, click **Add Template**.

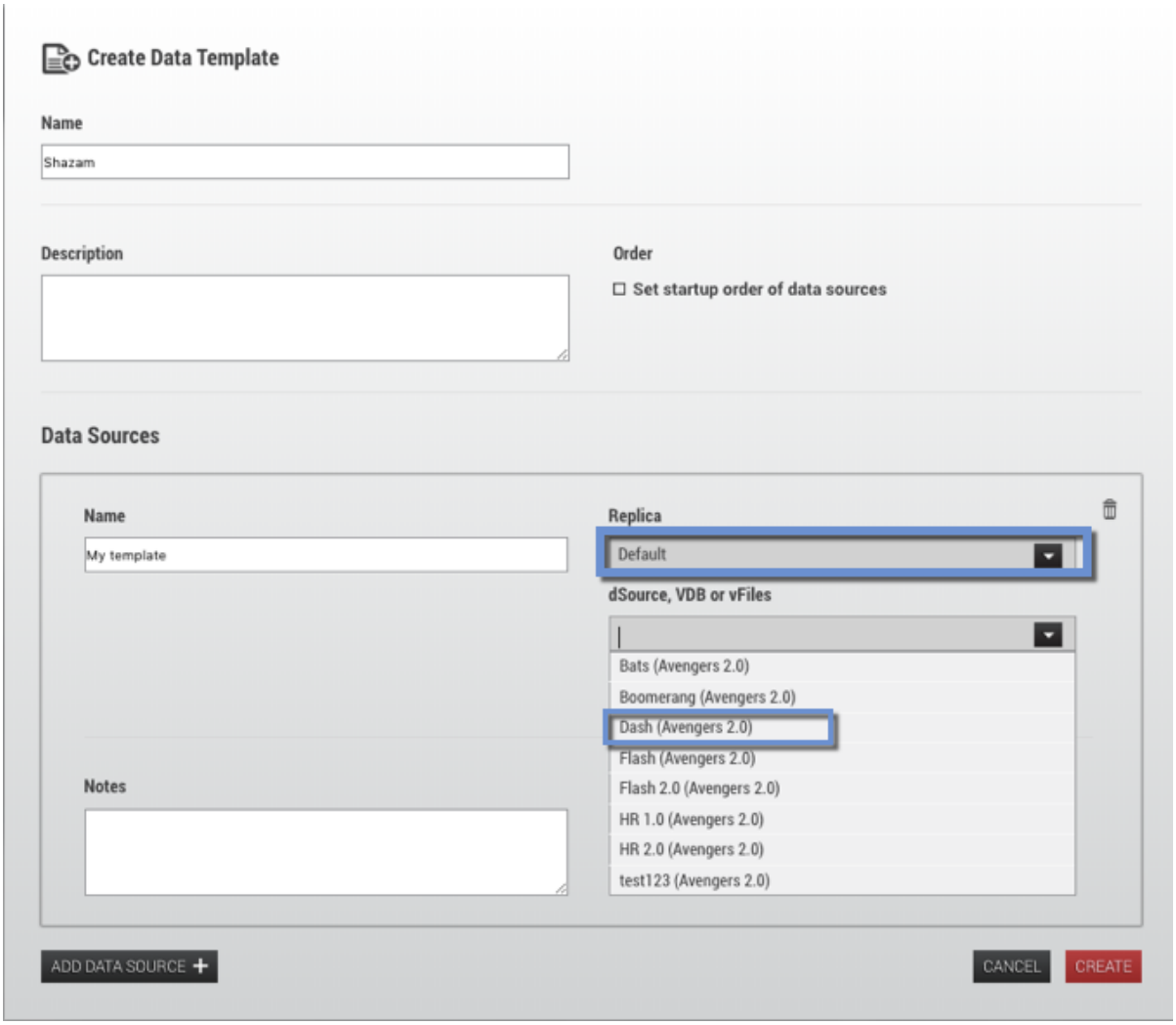
   ADD TEMPLATE +

   This will send you to the **Create Data Template** page.
3. Enter a **Name** for the data template.
4. Optionally, enter a **description** for the data template.
5. Click **Add Data Source** to add data sources to the template. Each data source name will include the name of the datasets group with which it is associated.



Create Data Template window with data source drop-down menu

To select a replica data source, first, select the name of the replica it belongs to. Then pick the replica masked VDB from the drop-down menu.



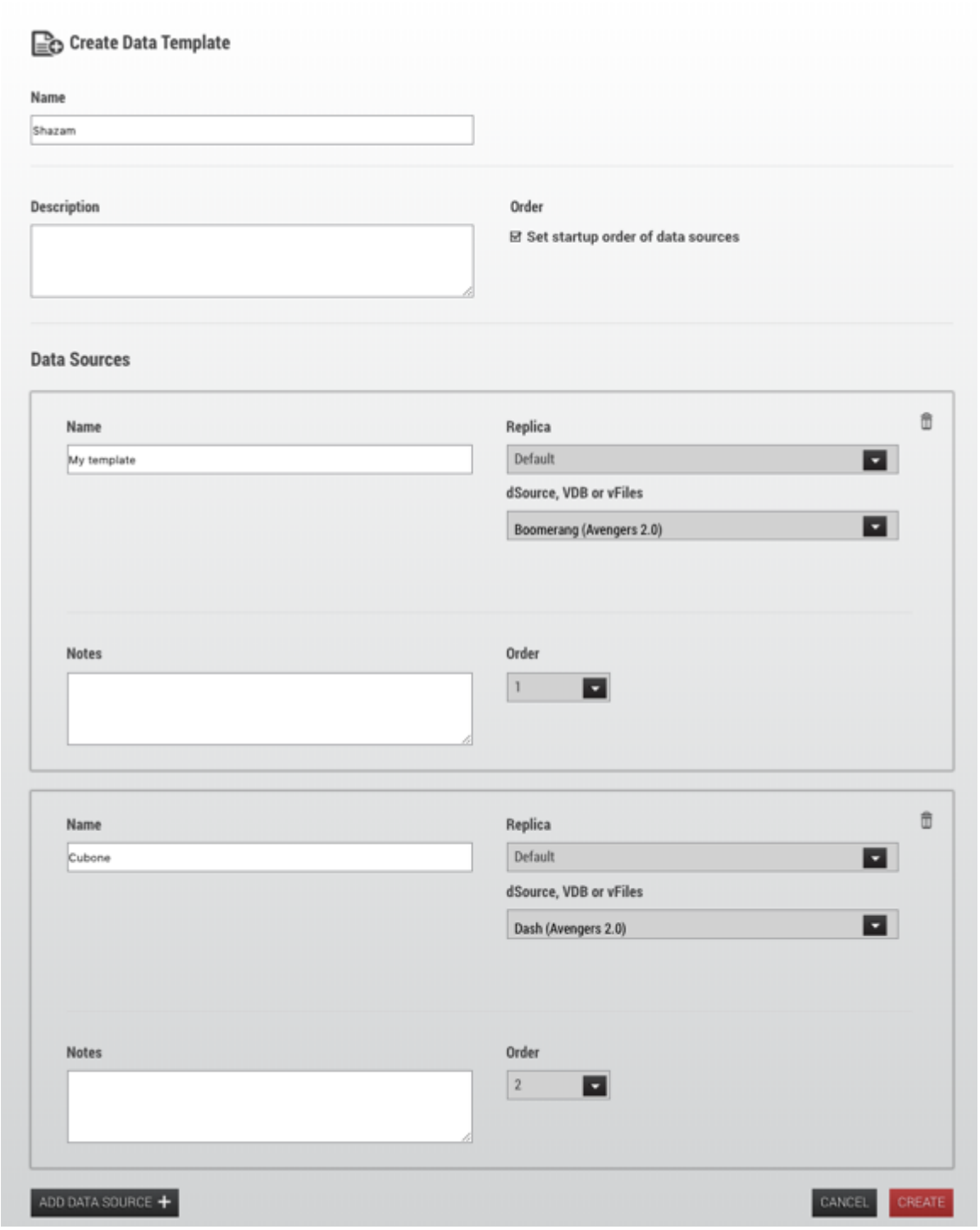
6. To set a startup order, select the Set startup order of data source checkbox, then from the drop-down select the order.
7. Select **Create**.

Default vs. Setting the Ordering of Data Sources to a Data Template
You have the option of setting the ordering of data sources to a data template. This option executes Delphix Self-Service operations sequentially on each data source (rather than in parallel), to ensure consistency among sources that need to be started/shut down in a particular order. You cannot change this setting after the data template has been created. **If you want the default behavior of running operations in parallel, do NOT select the box labeled "Set startup order of data sources".**
When your template has ordering constraints, as with Oracle EBS, you must set the startup order for each data source. Select the **Set startup order of data sources** box. The Delphix Engine will select the data source with order 1 as the first source started and the last one to be stopped. The data source with order 2 will be selected as the second source started, and this sequence will

continue until the last data source is selected and ordered. Note that it is not possible to have operations performed in parallel on a subset of data sources and sequentially on a different subset of data sources.

12.1.4.2.3 Configure the synchronization and consistent ordering of data sources

1. Select the **Set startup order of data sources** box.
2. Use the drop-down menu to select the **source** you want to include. The drop-down menu will display all dSources in the system and all VDBs and vFiles that are not already assigned to a Delphix Self-Service data container.
3. Enter a specific **name** for the data source.
4. Optionally, enter a **description** in the **Notes** section. Delphix Self-Service users see a copy of these notes in the data containers they own.
5. Click **Add Data Source** to continue to add and configure more data sources to the data template. You can remove data sources using the **Delete** icon.
6. By default, the **Order** of the sources will correspond to the order they are added. You can also edit the **Order** using the dropdown for each source.
7. Click **Create** to finish creating the data template.

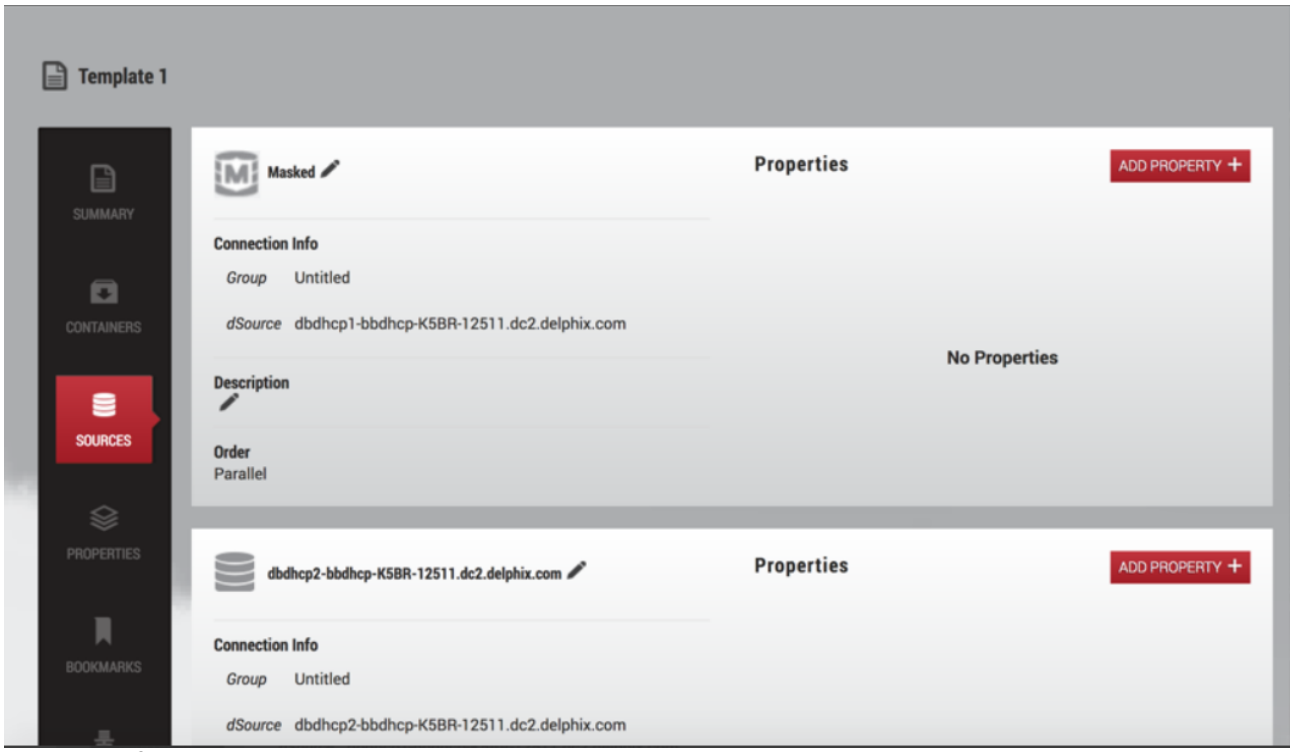


For Oracle EBS, the vFiles dbTechStack will have order 1, the Oracle database order 2, and the vFiles appsTier order 3. For more information about EBS, see the EBS documentation.

> Once you have created a template, you cannot change the set of data sources in it. Any VDBs or dSources being used as data sources in Delphix Self-Service will appear with a special badge in the Management application.

12.1.4.3 Viewing data templates

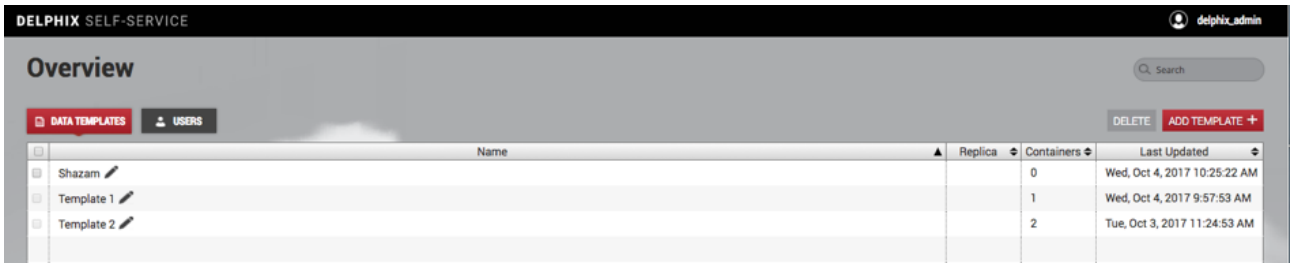
As the Delphix Admin user, you can view what sources have been included in a data template. You can distinguish the masked sources from the unmasked sources by referring to the corresponding data icons, as seen below.



Example of a template containing both masked and unmasked VDBs

12.1.4.3.1 Managing data templates

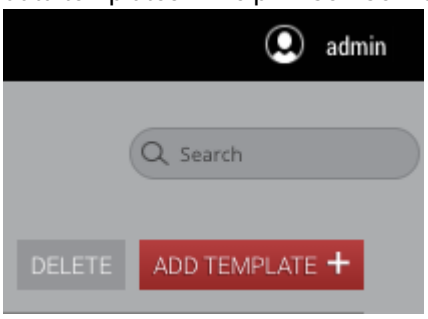
After you have created the data template, it will be visible from the **Overview** page under the **Data Templates** tab, which is the default tab.



Data Template Details in the Overview page

12.1.4.3.2 Notes

- Each line corresponds to a data template and contains high-level information about that data template. For example, the number of child data containers is visible in the **Containers** column.
- You can search, sort, and filter the list of data templates, making it easy to manage a large number of data templates in Delphix Self-Service.



12.1.4.3.3 Editing a data template's name

1. Click the **Edit** icon next to the data template name.
2. Enter the new **name**.
3. Click the **checkmark** icon to confirm changes.

12.1.4.3.4 Deleting a data template

1. Select the **data template** you want to delete.
2. Click the **Delete** button in the upper right-hand corner.

Data Containers

> If there are any data containers provisioned from the data template, you must remove them before you can delete the data template. See instructions in Data Container Activities[755].

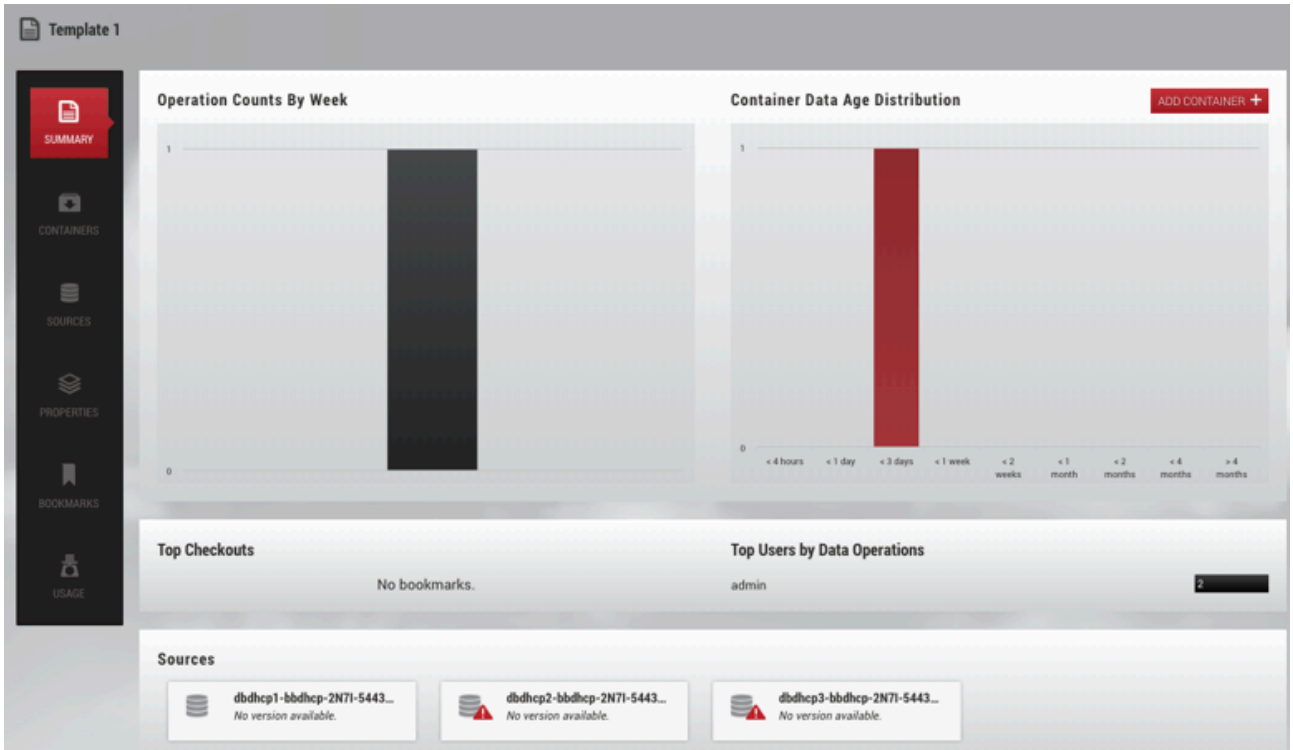
12.1.5 Understanding how to manage data template details

12.1.5.1 The data template details page

In the **Overview** page, under the **Templates**, click the data template's **name**. This will direct you to the templates details page. You can use this page to view and configure details of an individual data template. It consists of a number of tiles, described below.

12.1.5.1.1 Summary

Use this tile to get an overview of the data template and its child data containers.



6 Summary details for templates

Notes

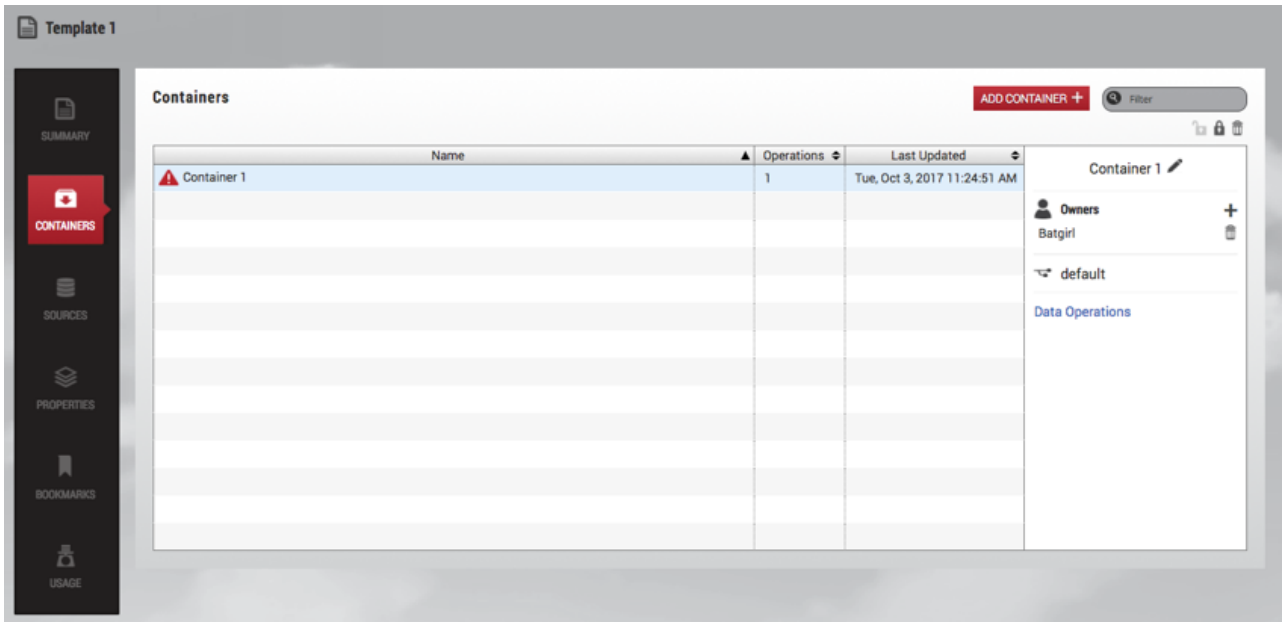
- The graphs labeled **Operation Counts By Week** and **Container Data Age Distribution** give a sense of the amount of activity in the data template over time

755 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/data-container-activities

- **Top Checkouts** shows at a glance which bookmarks have been used most frequently as part of a **Restore** or **Branch** operation
- **Top Users by Data Operations** shows at a glance which users are the most active

12.1.5.1.2 Containers

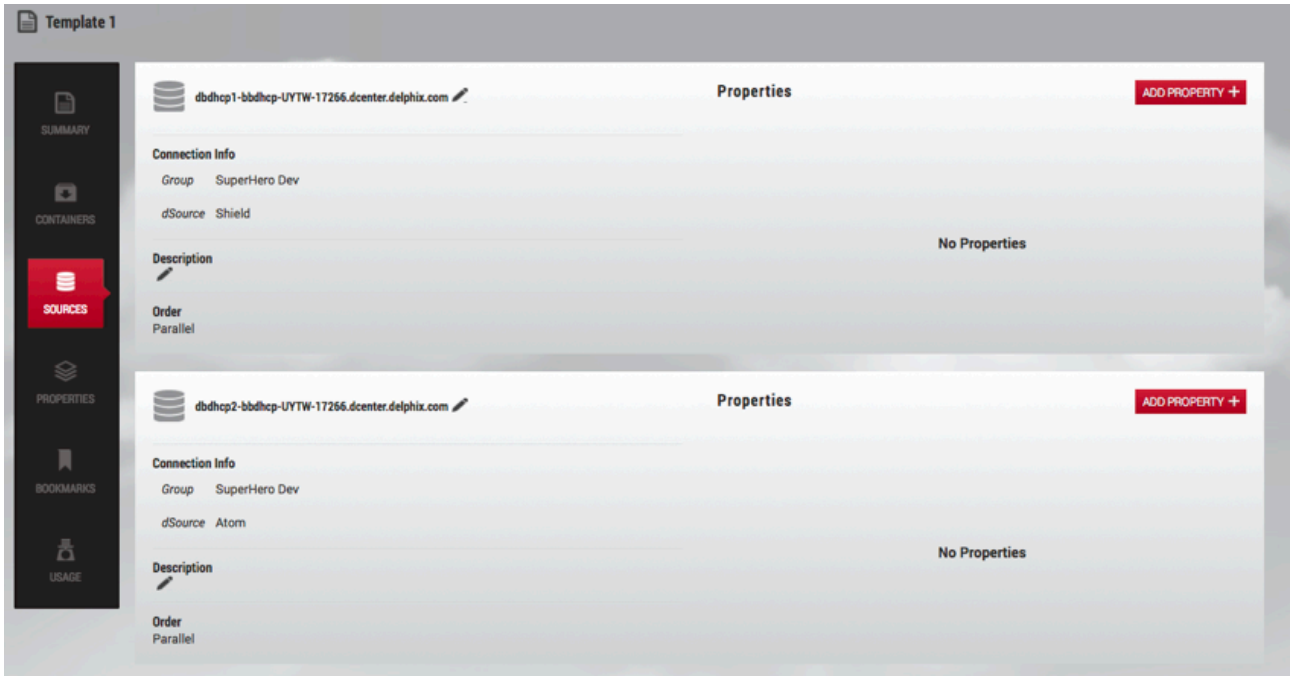
Use this tile to create, view, and delete child data containers from this data template.



7 Container details

12.1.5.1.3 Sources

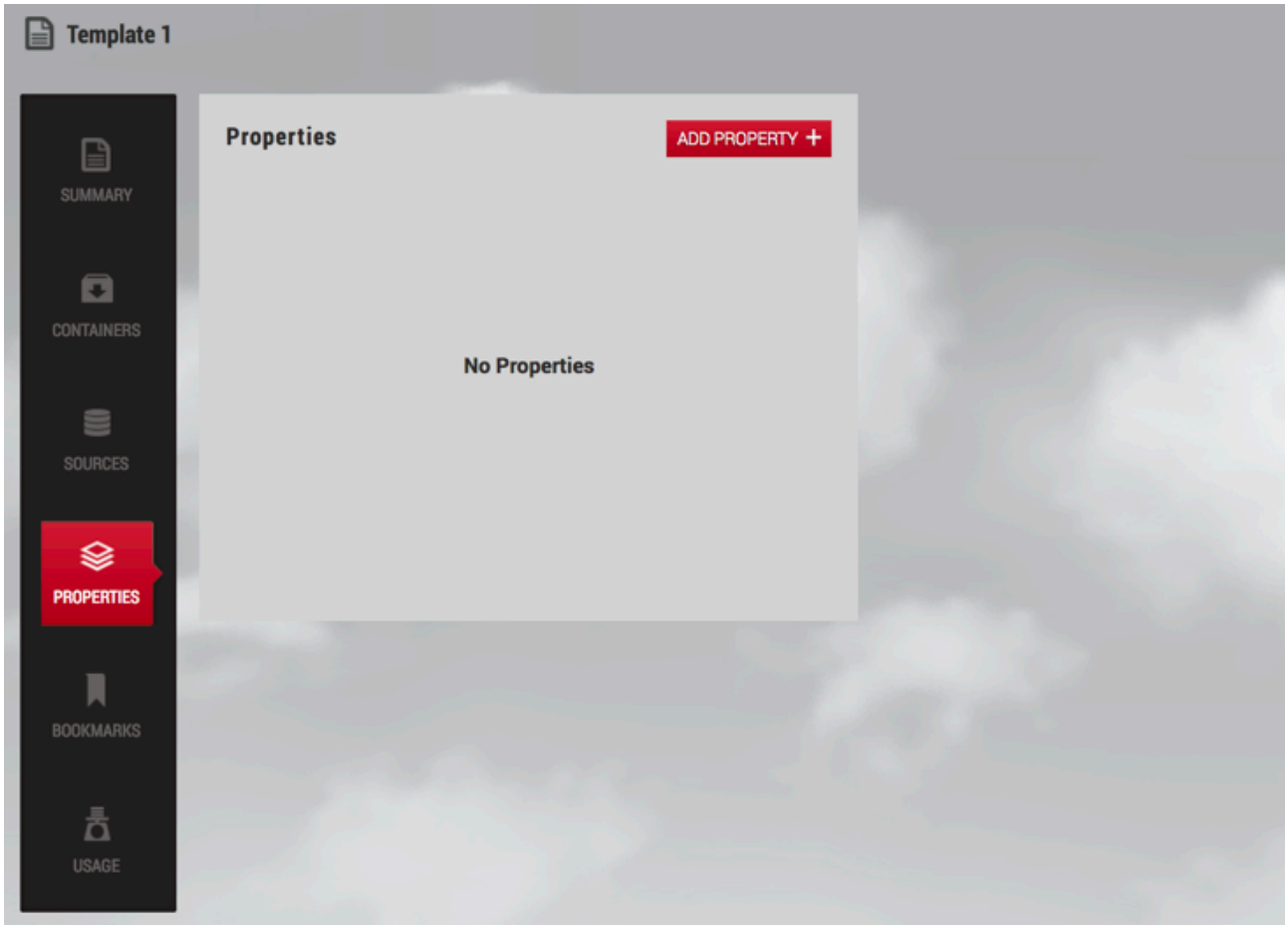
In this tile, you can view the data sources that this data template uses. Each data source has a user-visible name, a description, and a set of properties that consist of arbitrary key/value pairs. This information will be included in the data containers provisioned from this template.



8 Source details

12.1.5.1.4 Properties

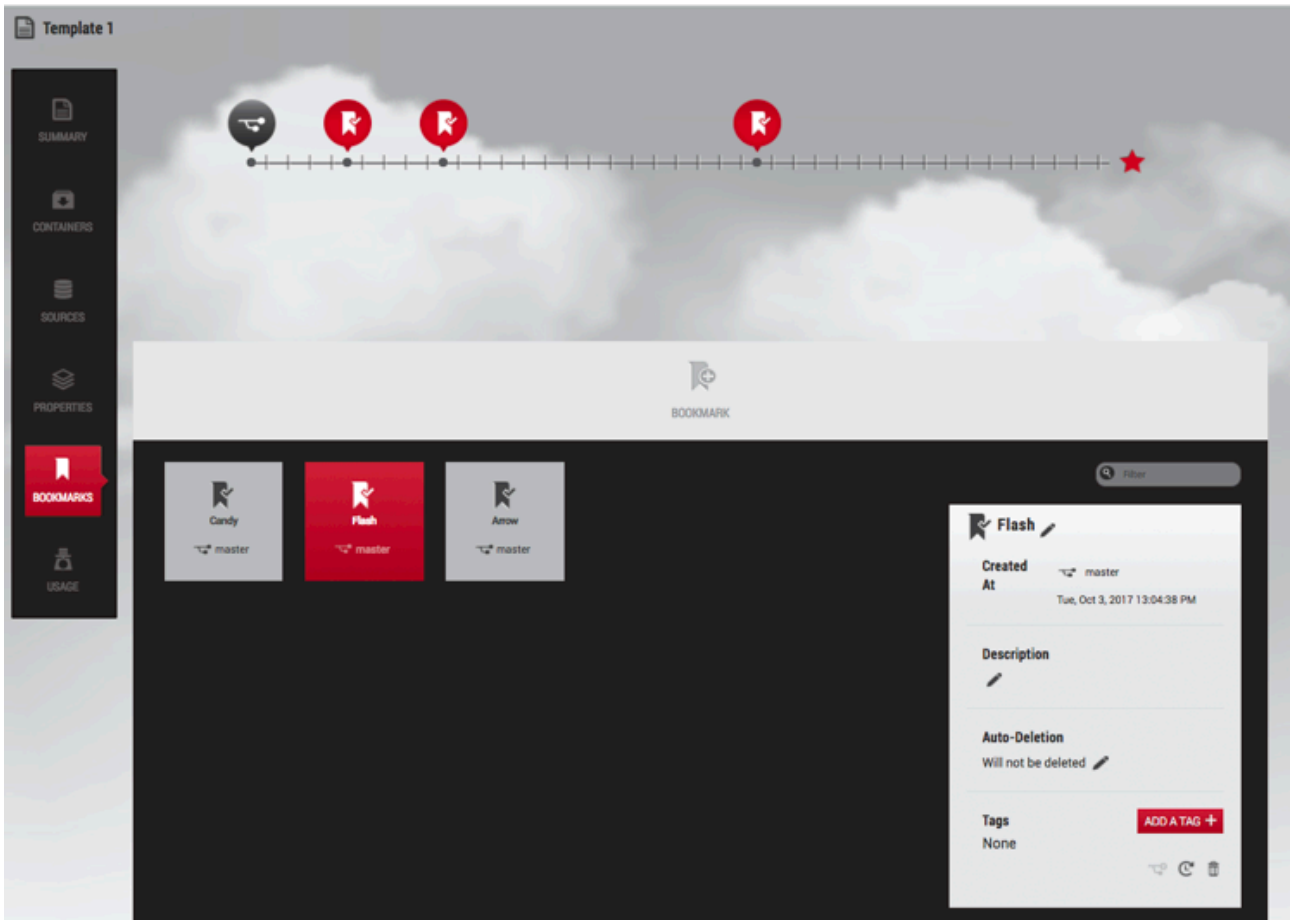
Use this tile to edit the data template's properties. Properties are arbitrary key/value pairs associated with the data template. These values will be propagated to all data containers provisioned from this template. This provides a way for you to annotate data templates and data containers with whatever information is relevant to their use case.



9 Properties details

12.1.5.1.5 Bookmarks

Use this tile to create and manage bookmarks on the data template. A bookmark represents a given point in time that is protected against retention. Bookmarks created on a data template are visible to all of the data containers provisioned from it. For more details, refer to the Bookmarks section in the Delphix Self-Service Data User Guide. (see page 2479)



10 Bookmark details

12.1.5.1.6 Capacity

Use this tile to get information about the storage associated with the data template and its child containers.

12.1.6 Understanding data containers

12.1.6.1 Delphix Self-Service data container overview

Data containers are provisioned from data templates by administrators and assigned to a user. A data container represents a socket that is capable of making any data within the data template accessible. The user controls what data they want to access.

Delphix Self-Service users have effectively been provisioned a set of "physical" resources, such as a database on a host that consumes some set of resources. A data container is comprised of a virtual database (VDB) or vFiles provisioned from each source in the data template from which it is created. The data container manages these VDBs, and the data operations performed on a data container will only impact these VDBs.

Data containers represent the separation between IT infrastructure and end-users. IT determines the set of VDBs or vFiles to allocate to a data container, and users determine the data that they want accessible in the containers allocated to them.

Data containers can be used to access any data within a single data template, but not across templates. Users have the ability to populate the data within their data container from any point in time on the data template, the data container's history, or shared bookmarks from other data containers.

Although operations are all accomplished by performing Timeflow operations on the underlying VDBs, the data containers hide the VDBs and their underlying properties from users. None of the data container operations require provisioning additional VDBs; everything is accomplished using the resources assigned when the data container is created.

Operation	Description
Refresh	This is the same basic concept as **Refresh** in VDBs. In Delphix Self-Service, **Refresh** will update the data on the active branch of a user's data container. The user will then have the latest data in the sources of the data template from which the container was provisioned.
Restore	**Restore** allows a Delphix Self-Service user to update the data on the active branch of their data container to one of the following: • Any point in time on the data container. • The data template from which the container was provisioned. • A bookmark. This operation effectively means, "take me to the data at this time."
Reset	**Reset** is a simplified version of **Restore**, built to support the notion of "undo." It allows a user to reset the state of their application container to the latest operation. This can be useful for testing workflows where, after each test, users want to reset the state of their environment.
Branch	A branch represents a logical timeline, effectively a task on which a user is working. Only one branch can be active at a time, but a user can use multiple branches to track logically separate tasks. Delphix Self-Service branches do not require the allocation of a new VDB; instead, they are comprised of a collection of TimeFlows within a VDB.
Activate	This allows the user to select which branch they want to be active. Only a single branch within a data container can be active at a time.
Bookmark	This creates a semantic name for a point in time and prevents this data from being removed by the retention policy. Bookmarks can be annotated with tags to make them easier to search for. In addition to tags, bookmarks allow a user to enter a description of what the bookmark represents.

Operation	Description
Share	Bookmarks can be shared, which allows them to be seen by users who own data containers that have been provisioned from the same data template. This allows users to share data, providing a way for other users to either restore their existing timeline or create a new branch from these shared points.

Once a data operation has been selected its progress can be viewed from the **Action** sidebar in the **Management** application. From the Action sidebar, users cannot cancel the enable, disable, or activate branch operations.

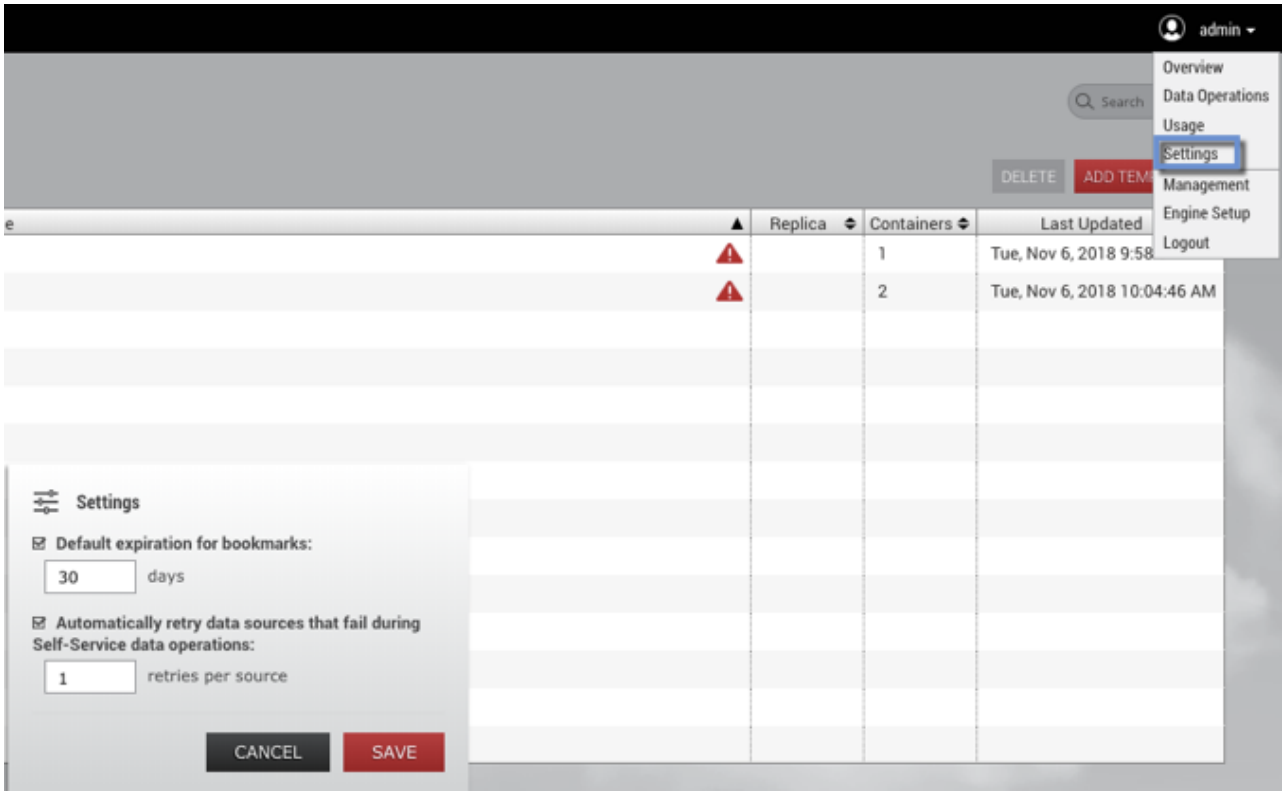
12.1.6.1.1 Automatic retry for data operations

To make operations on data containers more robust, Delphix Self-Service supports automatically retrying failed sources during data operations. You can specify a maximum number of retry attempts so that if an operation fails on any individual data source within a data container, it will be automatically retried until it succeeds, or until the retry limit is reached.

Automatic retry applies to any Delphix Self-Service operation that changes the data in the data container, such as Refresh, Restore, Reset, or Create Branch. This setting can be especially useful in scenarios where there are a large number of sources in a data container, and some sources fail to update the first time. If the reason for the failures was intermittent, an automatic retry may allow the sources that failed to succeed, and the operation can still complete successfully. The default number of retry attempts is **1**.

To change the number of automatic retries:

1. Click the user icon in the upper right-hand corner of the screen.
2. From the drop-down menu, select **Settings**.
3. In the field **Automatically retry data sources...**, enter a new maximum; alternatively, use the arrows to increase or decrease the number.
4. Click **Save**.



11 Settings in the user drop-down menu.

12.1.6.2 Delphix Self-Service data container recovery

12.1.6.2.1 Data containers consistency

Delphix Self-Service allows you to group multiple datasets in the same data container. This makes it easy for you to access entire applications such as PeopleSoft, including binaries and code.

If a data container represents an application, then there are likely to be dependencies between the application's datasets. For example, the vFiles data source containing the code will depend on a specific version of the database's schema. Therefore, it is important that all dataset sources are drawn from the same point in time. If they are, the data container is in a "consistent" state; if they are out of sync, or "inconsistent," errors will occur. For example, if the vFiles data source containing the code has been updated more recently than the database's schema, the dependency cannot work.

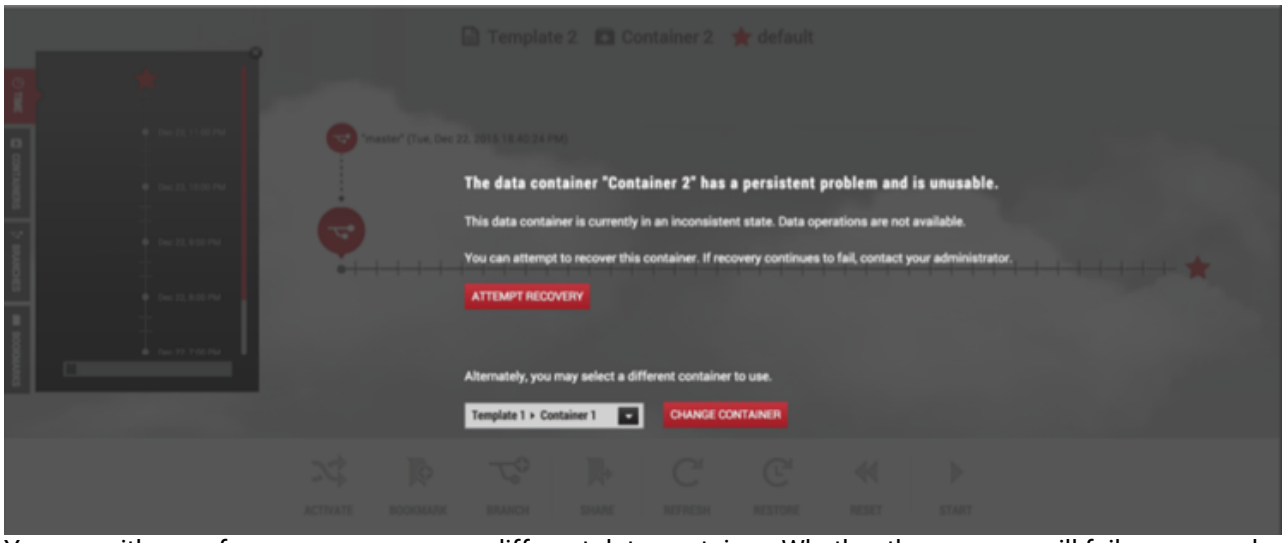
Delphix Self-Service currently has no way to determine whether the application is consistent. However, it attempts to minimize the chance that dataset sources are out of sync whenever it performs a data operation such as refresh, restore, or reset.

When performing a data operation, Delphix Self-Service attempts to snapshot all dataset sources from a point in time as close as possible to the desired time. If at least one of the data sources fails to go to the desired point, then Delphix Self-Service considers the data container to be in an inconsistent state.

The application as a whole may still be working, but Delphix Self-Service assumes that the failed dataset's data is not the correct version. To return to a consistent state, you must perform a recovery operation on the data container.

12.1.6.2.2 Data container recovery

Prior to performing any data operation, Delphix Self-Service takes snapshots of all datasets. Recovery is the process of rolling back a data container to a snapshot, thereby restoring it to a consistent state. When a failure occurs, you will see the following screen:



You can either perform recovery or use a different data container. Whether the recovery will fail or succeed depends on why the data operation failed in the first place.

If the problem was intermittent, such as a temporary network problem causing SSH failure, then performing recovery should work. If the problem is persistent – for example, the target host is out of space – then intervention is required; recovery will not succeed until you address the underlying root cause of the failure.

Admins can see the underlying failure in the **Actions** sidebar or the **Job History** dashboard.
The **Actions** sidebar is the preferred place to view the failure; it has a hierarchical display that makes diagnosing the failure more straightforward.

12.1.6.3 Preserving independent containers in Delphix Self-Service during replication

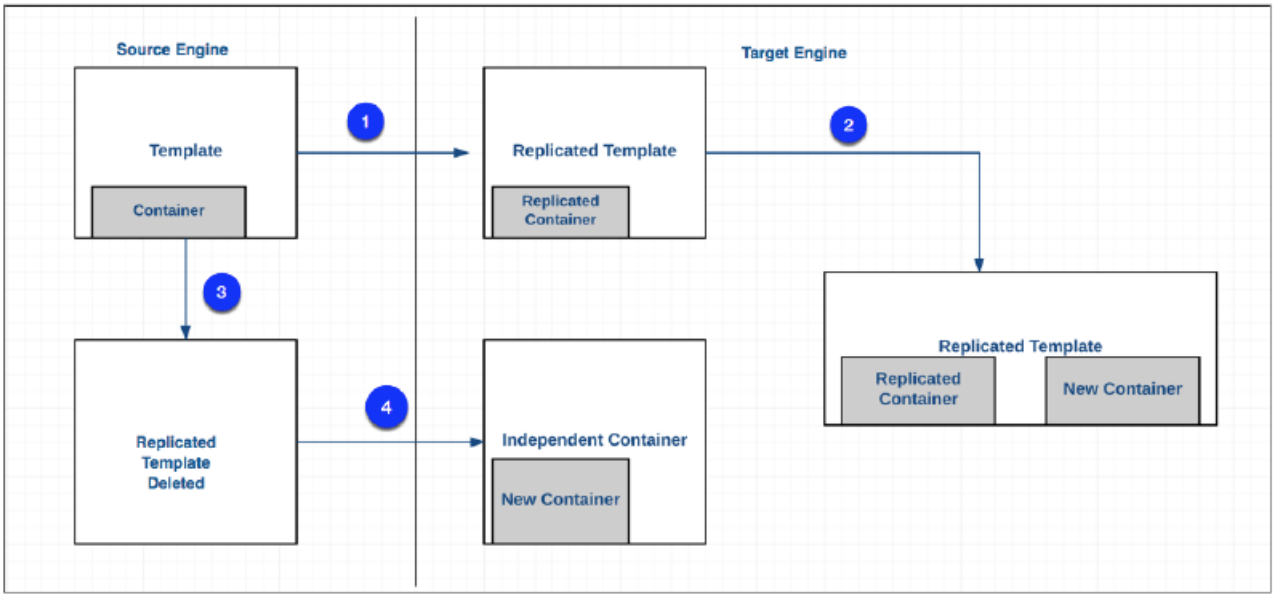
Replication is used for data backup and recovery as well as for managing and sharing data across remote data centers. Delphix Self-Service users can preserve their data after replication jobs.

In the past, if replication occurred on templates in containers, users would lose the data in their containers. Now, admin users can preserve containers to be used independently of replication jobs.

Independent containers behave in the same way as other containers, with two exceptions:

1. You cannot refresh them.
2. The bookmarks created on them cannot be shared, because they do not have a template reference.

The functional overview of independent containers seen below represents the flow of steps between the source engine and the target engine. A description of what is occurring between each of the steps appears below the diagram. See the corresponding number below the diagram to find out more details.



12 Functional overview of independent containers.

1. In Delphix Self-Service, you can create a template on the source engine and then replicate the template to the target engine.
2. The target engine, an admin can use the replicated template to create new containers and assign them to users. You cannot change the replicated template's name or the names of the containers with which it was replicated over.
3. Due to an update, the replicated template is deleted from the source engine
4. The deleted replicated template will be removed from the target engine. Any new container created in step 2 loses the reference to the deleted template and becomes an independent container.

12.1.6.4 Creating independent containers

12.1.6.4.1 Prerequisites

- The replication source and the replication target must be running identical versions of the Data Engine – for example, Data Engine version 5.1.
- The target Delphix Engine must be reachable from the source engine.
- The target Delphix Engine must have sufficient free storage to receive the replicated data.
- The user must have administrative privileges on the source and the target engines.

For more information, see Configuring Replication[756] and Understanding Data Templates.[757]

756 http://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/configuring-replication
757 http://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/understanding-data-templates

12.1.6.4.2 Limitations of this functionality

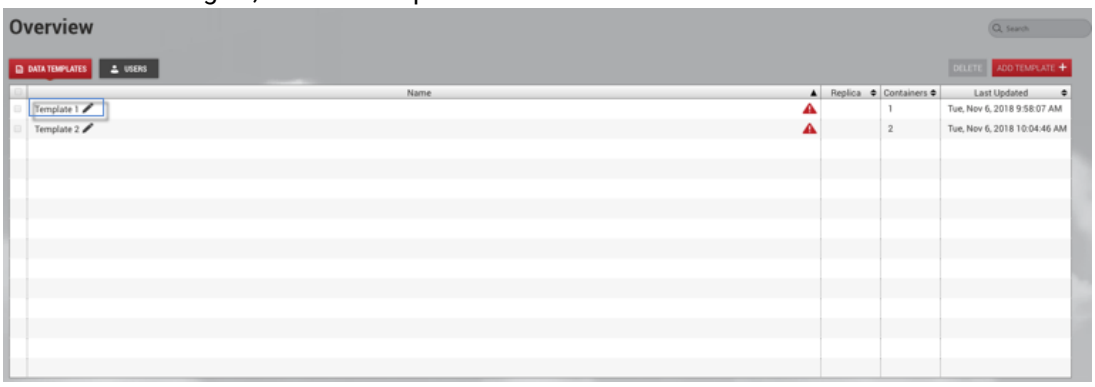
You can find independent containers in Delphix Self-Service on the target engine under the **Independent Containers** tab. They have the following characteristics:

- They cannot be refreshed, because they are no longer bound to a template.
- You can create bookmarks on them, but you cannot share those bookmarks because there is no common template.
- You can use them for branching, restoring, resetting, starting, and stopping.

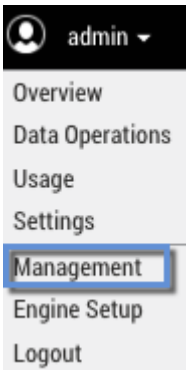
12.1.6.4.3 Procedure

To create an independent container, complete the following steps:

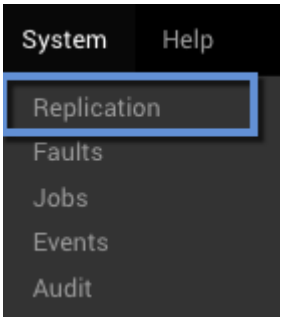
1. On the source engine, create a template with a container.



2. From the user drop-down menu, select **Management**.



3. From the **System** menu, select **Replication**.



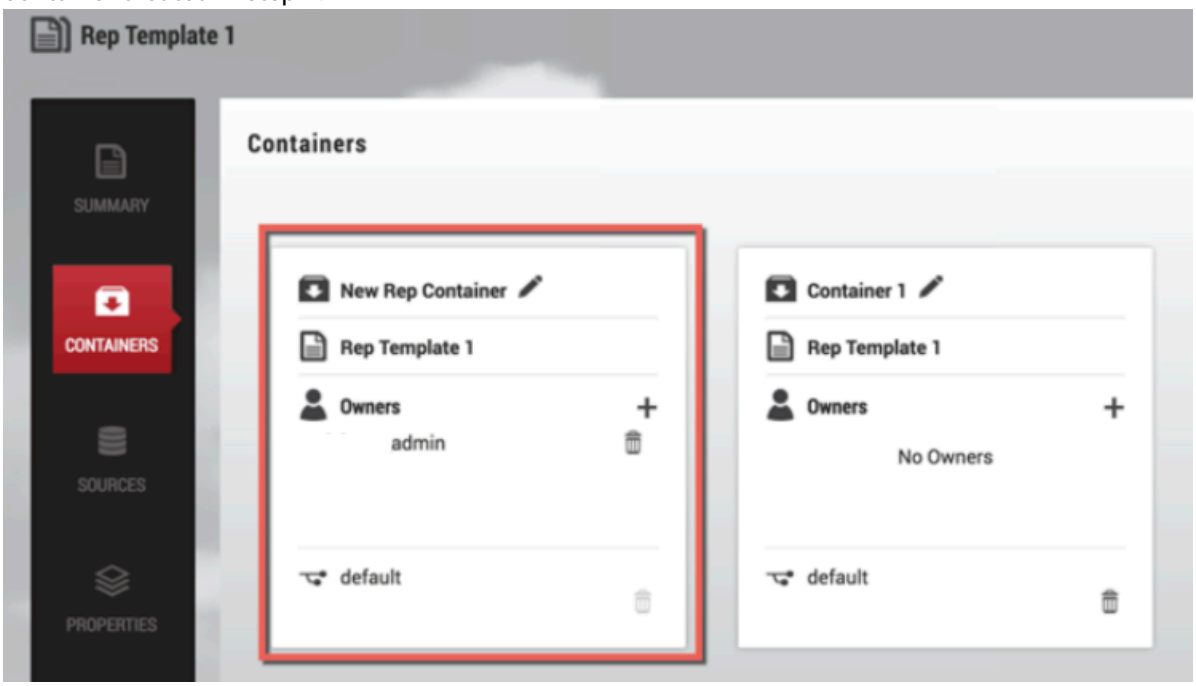
4. Next to **Replicated Profiles**, select the **plus** icon to **Create New Profile**.
5. Under **Objects Being Replicated**, select your Self-Service **template** and its associated **container**.
6. Enter your profile information.
7. Click **Create Profile**.

When replicating templates, you can select all, some, or none of their associated containers in the replication profile. This is done by selecting the checkbox next to the container's name in the **Create New Profile** window. When replicating a container, you must also replicate its associated template. Replicated objects cannot be modified on the target engine unless they are failed over, so you cannot modify the names of replicated data containers and templates.

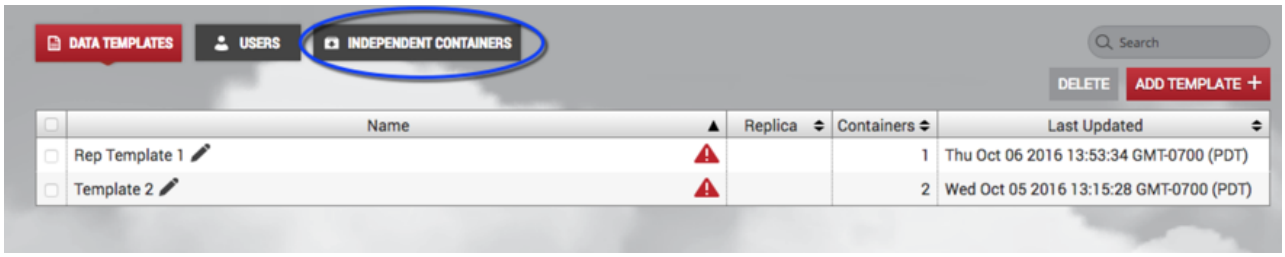
8. Once the profile has been created, click **Replicate Now**.
9. On the target engine, click the user menu.
10. Select **Self-Service**.
11. The replicated template will appear in Self-Service on the target engine. The replica name is displayed next to the template name. You can edit regular templates by clicking the pencil icon next to the template name.

12. Select the replicated template, then select **Containers**.

13. In the **Containers** window, click **Add Container**. In order to complete this action, you will need to ensure that there is data available from each data source in the template. This means that VDBs must have been provisioned from each replica dSource or VDB in the template. After the container is created, your replicated template should have the new container you just created and the original container created in step 1.



14. On the source engine in Self-Service, delete your template.
15. From the user menu, select **Management**.
16. From the **System** menu, select **Replication**.
17. Replicate your profile to create a new independent container.
18. On your target engine, select **Self-Service**.
19. The new container is created. To find it:
    a. Login to the target engine.
    b. Click the user menu.
    c. Select **Self-Service**.
    d. In the Overview page, select the **Independent Containers** tab.



12.1.7 Delphix self-service data container activities

12.1.7.1 Configuring data containers in Delphix self-service

A data container is comprised of a set of virtual databases (VDBs), where each VDB is a direct child of the dSource, VDB, or vFiles in the data template's data sources. Delphix Self-Service does not automatically provision VDBs when creating a data container; a Delphix admin must create the required VDBs using the **Management** application.

> Once a VDB or vFile is part of a Self-Service container, you cannot use the Management Service to rewind, refresh, or delete it. You can still use the Management Service to disable or enable it, take a snapshot of it, or provision a new VDB or vFile from it.

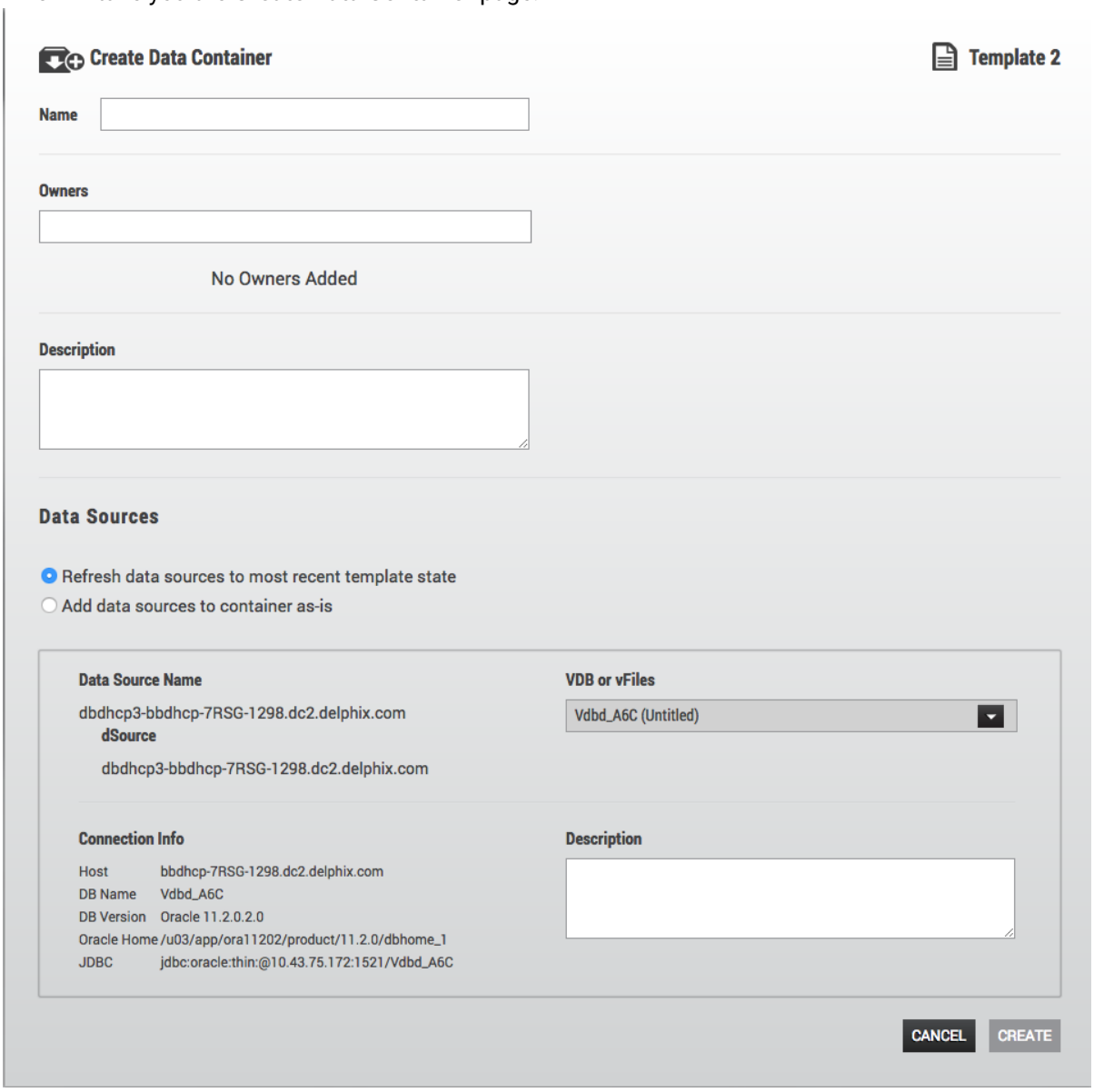
1. Select the **Overview** page.
2. Select a **template** from which you want to create the data container. This will take you to the **Data Template** page.

12.1.7.1.1 Add a data container

1. Click **Add Container** in the upper right-hand corner of the screen.

Details Panel and Dashboard

This will take you the **Create Data Container** page.

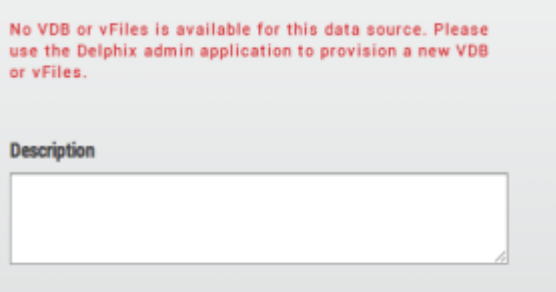


Data Container Dialog

2. **Optional**: Enter information about the data container, such as the **Name** and **Description**.
3. Once the data container has been createdSelect the **Owners** for the data container from the search box. It is acceptable to have multiple owners for each data container. Any Delphix administrator is able to manage all containers, so the owners should be end-users. For details, see Understanding Delphix Self-Service User Management. (see page 2462)

4. When a data container is created, you now have the option to:
   a. **Refresh data sources to most recent template state** – This option will refresh VDBs before adding them to the container. This is done to enforce that when multiple sources are used in a container, the sources are consistent.
   b. **Add data sources to container as-is** – This option will not refresh the data sources.
5. Select the **VDBs** to use for this container's data sources. The available VDBs have the following constraints:
   - They have been provisioned from the dSources/VDBs belonging to the parent data template
   - They are not already part of another data template or container

     If there are no VDBs that meet these constraints, you may see a message informing you that you do not have any compatible VDBs.

     VDB Warning Alert
6. Click **Create**.

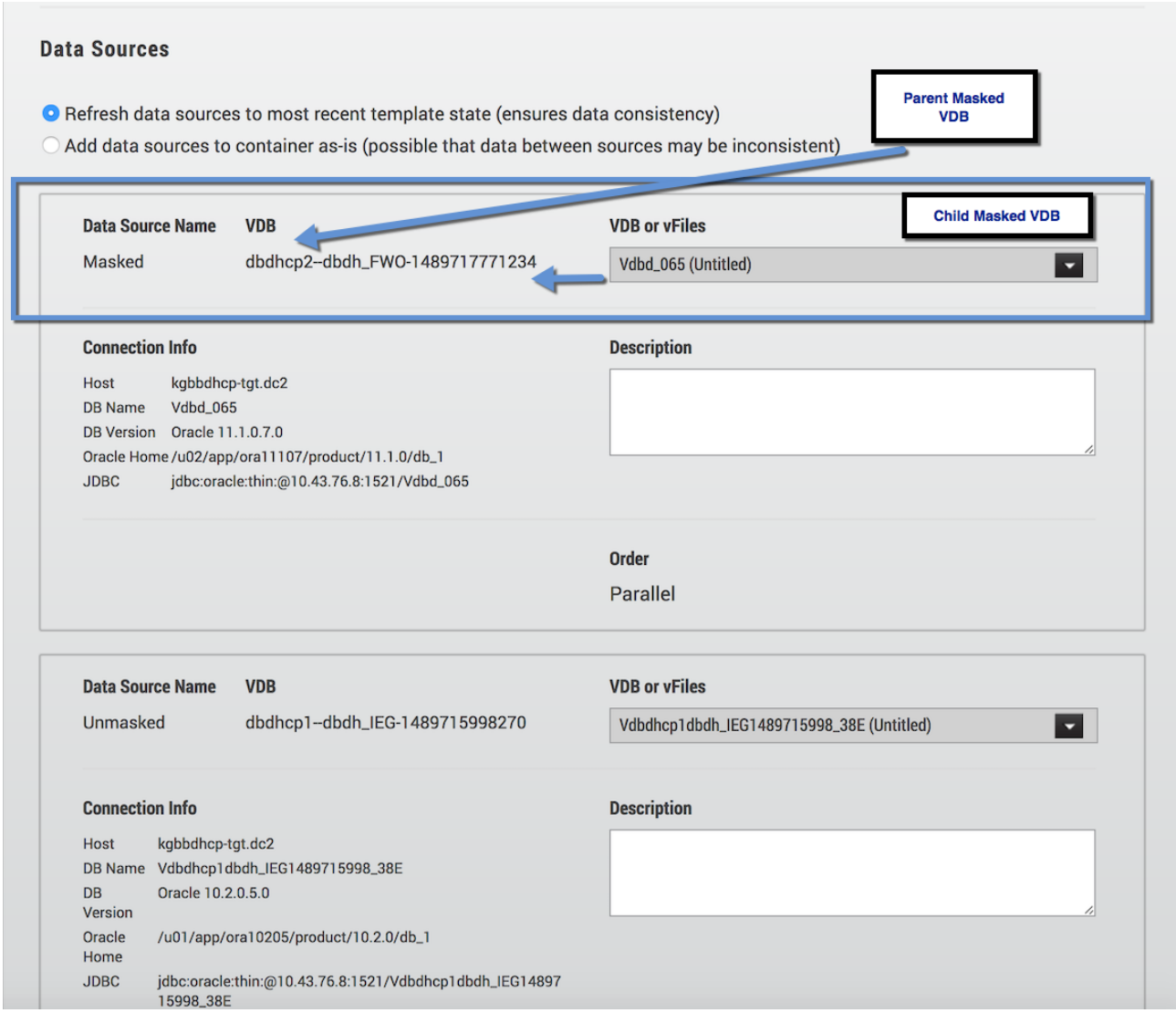
12.1.7.2 Selecting Masked Data Sources for Data Containers

12.1.7.2.1 Prerequisites

- Using Masked Data Sources with Delphix Self-Service (see page 2457)
- Selecting Masked Data Sources in Data Templates (see page 2457)

12.1.7.2.2 Procedure

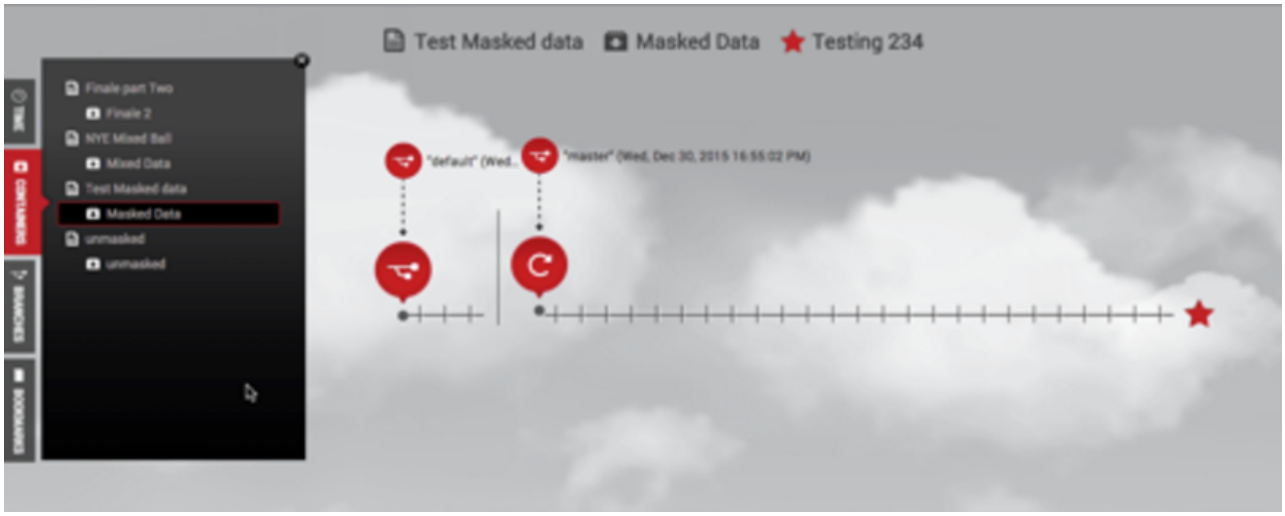
Once you have selected a child masked VDB for the data container, you can see the parent-child relationship as a masked source under data sources.



Masked Data Sources Parent/Child Relationship

As an admin user, you can select both masked and unmasked data sources in both Delphix Self-Service templates and data containers.

Delphix Self-Service users will not know whether the data in their containers and branches is masked or unmasked. All Delphix Self-Service functionality remains the same regardless of whether a data source is masked or unmasked.



The figure above is an example of a data container with masked data.

12.1.7.3 Delete a Data Container

All data sources (VDBs and vFiles) in a Data Container are not deleted as part of the Data Container deletion process.

When performing the **Delete Container** operation, you can check the **Delete associated VDBs and vFiles** box in the dialog window to delete the data sources associated with the container.

12.1.7.4 Data management operations

12.1.7.4.1 Start a data container

Starting a Data Container does the following:

- Starts the data sources, This means that each data source listed in the **Source Details** section of the **Data Container** page will start using CPU and network resources on the host system it is running.
- Puts a copy of the data from the active branch into those data sources.

On the **Self-Service Toolbar**, click **Start**.

12.1.7.4.2 Stop a data container

Stopping a data container does the following:

- If not already done, copies the current data in the data sources into the active branch of the data container
- Shuts down the data sources. This means each data source listed in the **Source Details** section of the **Data Container** page will stop using CPU and network resources on the host system.

On the **Self-Service Toolbar**, click **Stop**.

Other operations on the data container, such as Stop, Reset, and Refresh, must be performed from the **Data Management** page:

Data Management Interface Shortcut in Delphix Self-Service Data Template

12.1.7.4.3 Locking a Container

On the Self-Service Toolbar, click **Lock**.

Locking a data container does the following:

- You become the only user who can perform operations on it.
- For all other users, the container appears disabled.

12.1.7.4.4 Unlocking a container

Unlocking a data container does the following allows other Self-Service users to perform operations on that container.

This operation is only enabled if a container is currently locked. Only the user who locked a container or a Delphix Self-Service administrator can unlock it.

> In Delphix Engine version 8.0.0.0 (or later), if a Self-service data container contains an Oracle virtual PDB (vPDB) and the PDB's virtual container database (vCDB) contains more than one vPDB, the start of the container will fail with error 'failed to enable data container "<data container name>"' if the vCDB is not enabled or it is not running. Start the vCDB first (using Delphix Management UI) and then start the data container again.

12.1.8 Using masked data sources with Delphix self-service

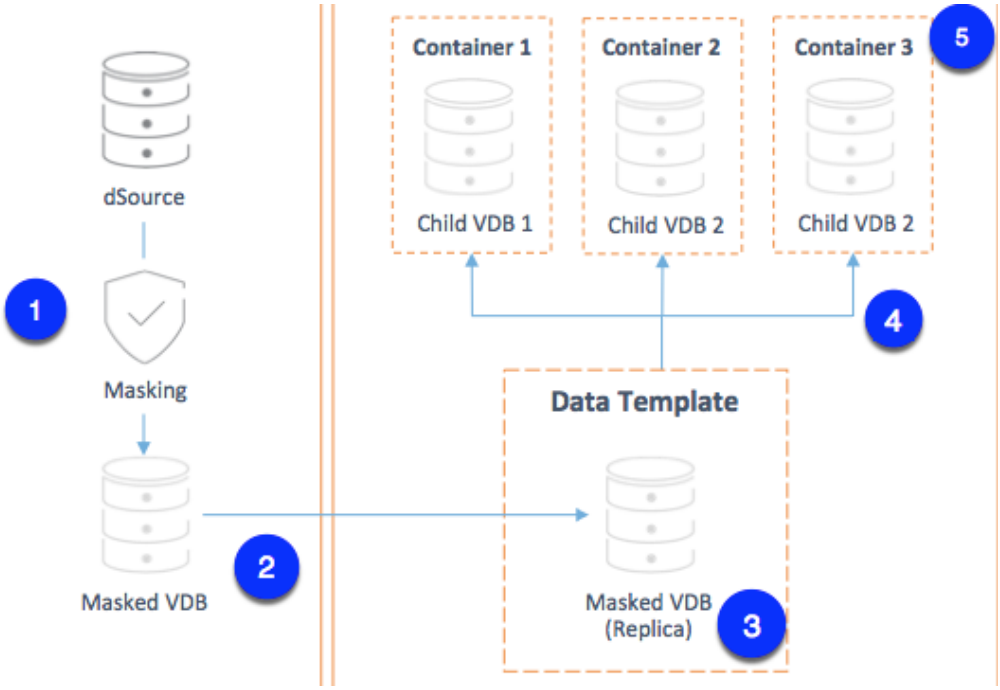
12.1.8.1 SDD Overview

You can now replicate masked data in a VDB directly to a target Delphix Engine without transmitting the unmasked data in its parent source. This is called selective data distribution (see page 2418) (SDD). Although you can run selective data distribution ad hoc, it is typically run according to a predefined schedule. In the current release, there are some best practices and limitations you should know about before using data from

SDD in Delphix Self-Service. This section covers the workflow for using SDD replicated data in templates and containers. It is aimed at administrators who are familiar with the process of creating a masked VDB, SDD replication and setting up objects.

12.1.8.2 Configuring Delphix Self-Service with Masked Data Sources

The diagram below illustrates the steps for using masked data sources in Delphix Self-Service.



12.1.8.2.1 Step 1: Provision a masked VDB on the source.

12.1.8.2.2 Step 2: Use SDD to replicate masked data to the target.

Use SDD to replicate your masked VDB to the target. The target VDB will be called the “replica masked VDB.” Replica Masked VDBs

To keep the replica masked VDB up to date, configure a refresh policy for how often it should refresh. The refresh policy should be related to the schedule for SDD updates from the source. Refreshing more frequently will result in the VDB being unavailable to Delphix Self-Service more often than needed.

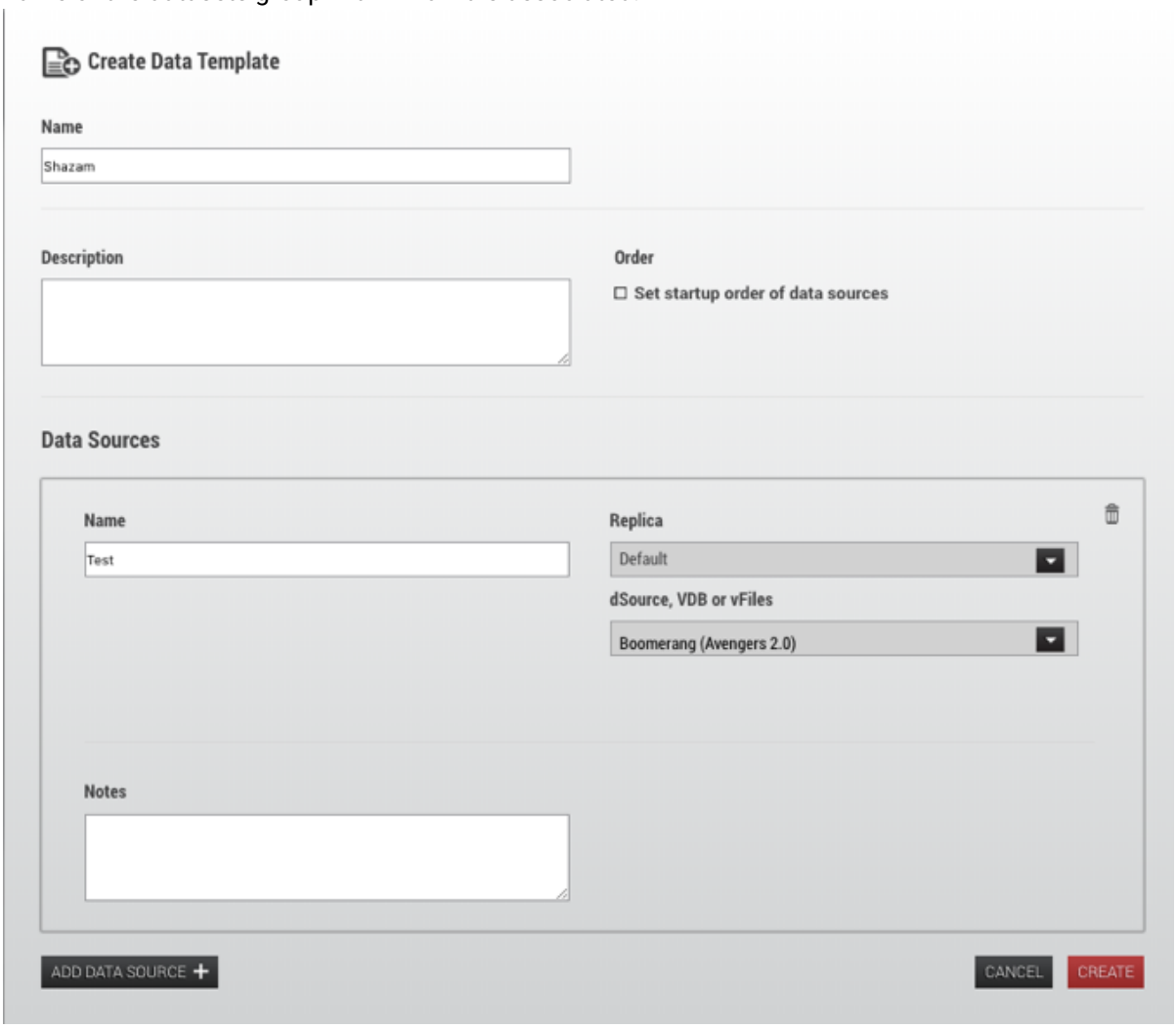
12.1.8.2.3 Step 3: Create a data template on the target.

To create a data template:

1. From the drop-down menu in the upper right-hand corner of the Delphix Management application, select **Self-Service**.
2. On the **Overview** page, click **Add Template**.

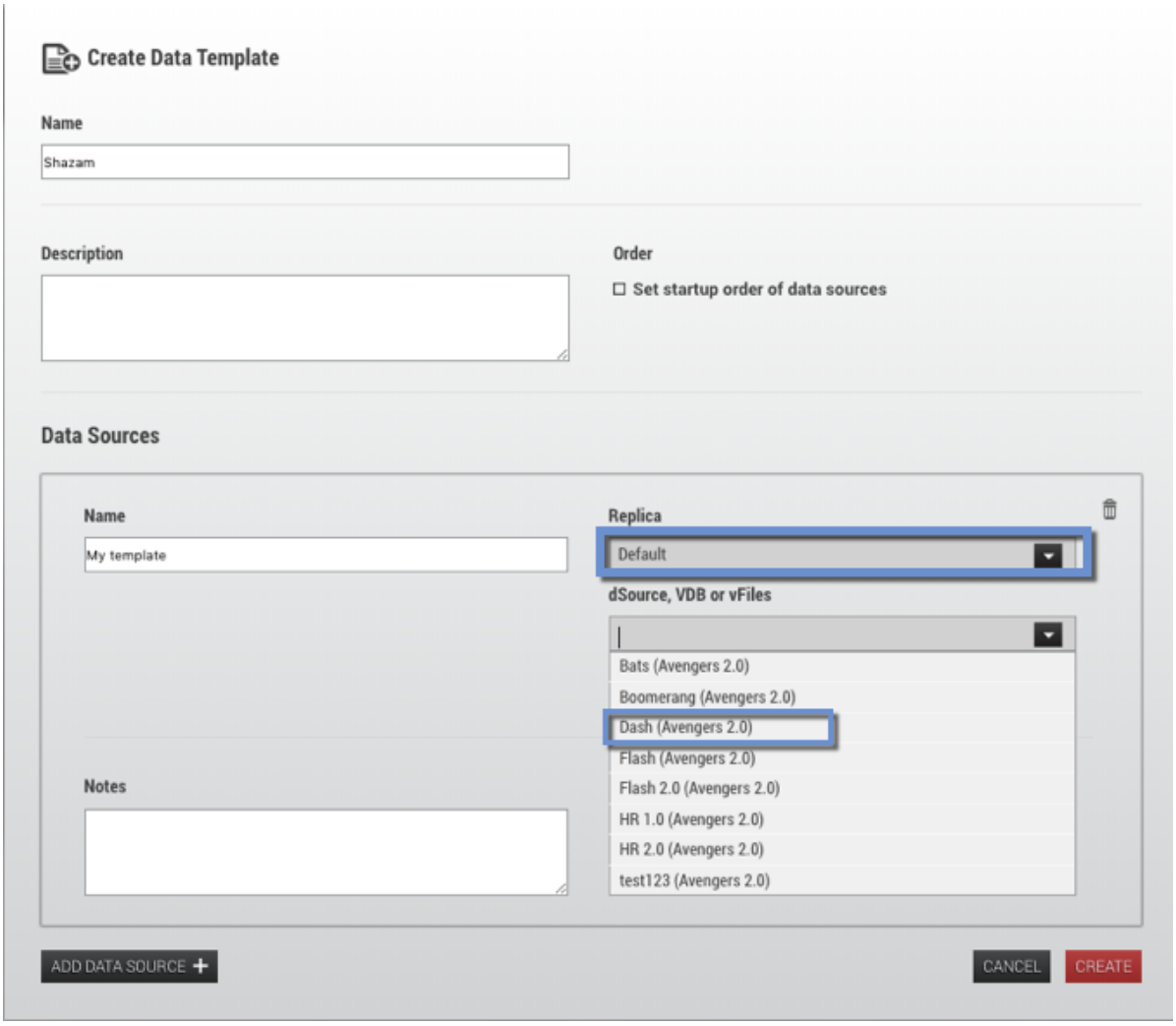
Add Data Template This will send you to the **Create Data Template** page.

3. Enter a **Name** for the data template.
4. Optionally, enter a **description** for the data template.
5. Click **Add Data Source** to add data sources to the template. Each data source name will include the name of the datasets group with which it is associated.



Create Data Template window with data source drop-down menu

To select a replica data source, first, select the name of the replica it belongs to. Then pick the replica masked VDB from the drop-down menu.



6. To set a startup order, select the Set startup order of data source checkbox, then from the drop-down select the order.
7. Select **Create**.

12.1.8.2.4 Step 4: Provision child VDBs from the replica masked VDB.

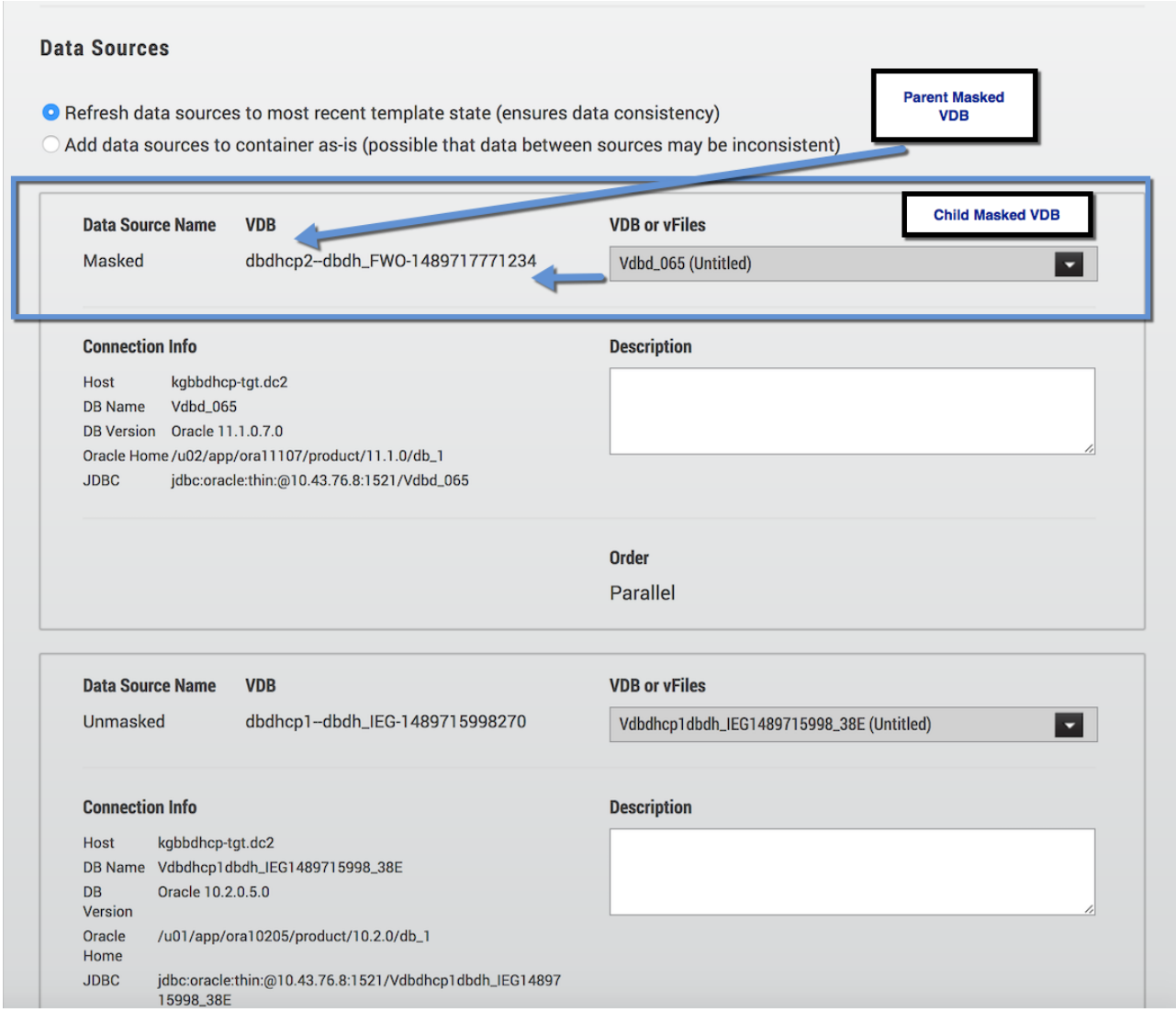
1. Login to the **Delphix Management** application for the target host.
2. Click **Manage**.
3. Select **Datasets**.
4. Select the **replica** that contains the dSource or VDB to be provisioned.
5. The provisioning process is now identical to the process for provisioning standard objects.

12.1.8.2.5 Step 5: Add data containers and select the child VDBs as data sources.

Follow the instructions to add a data container.[758]

Select a masked child VDB as a source for the container. As an admin user, you can select both masked and unmasked data sources in templates and data containers.

Once you select a child masked VDB for the data container, you can see the parent-child relationship as a masked source under data sources.



Masked Data Sources Parent/Child Relationship

12.1.8.3 Refreshing Masked VDBs in Delphix Self-Service Data Templates

Make sure that in step 4 above, you select the replica masked VDB as the source for the data template.

758 https://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/understanding-data-containers

In order for new data to be available in the template on the target, you must do the following:

1. Refresh the masked VDB on the source. This will re-run the masking job.
2. After the refresh completes, execute the SDD spec for the masked VDB.

Wait to refresh If you do not wait until the refresh is complete, unmasked data may be sent to the targ

SDD update
Although you can employ a policy to drive refreshes of the masked VDB, you cannot use that policy to drive the SDD update as well. You may need a combination of policies and scripts to automate the workflow.

12.1.8.4 Limitations

SDD Replication Profile

You cannot add data templates to an SDD replication profile. As a result, you must create the data template on the target. This is step 3 above.

12.1.9 Understanding Delphix self-service user management

12.1.9.1 User management activities

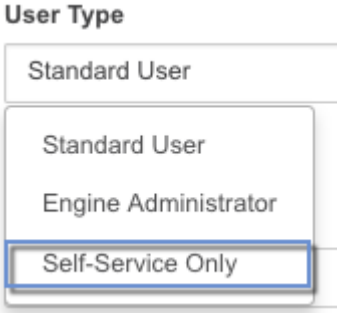
This document describes the process of creating a user and assigning that user to a data container. It also provides an overview of the **User Details** page.

Creating a User

Follow the same process when creating a new user or modifying an existing Delphix user. Delphix Self-Service users do not have access to the existing admin user interface, and they can only access the **Data Container** page for containers they own.

1. From the **Management** application, select **Manage**.
2. Select **Users**.
3. Click **+** to add a new user or to make an existing Delphix Engine user a Delphix Self-Service user, select the user from the list.
4. Enter the appropriate information.

5. From the **User Type** drop-down menu, select **Self-Service Only**.



6. Press **Submit**.

The user is now a Delphix Self-Service user! They can now login to the Delphix Self-Service user interface, and you can make them the owner of a data container.

Notes

- Users will only be able to access the **Data Management** page. They will not be able to access the other portions of the Delphix Self-Service interface, nor the **Management** application.
- A Delphix admin user cannot be made a Self-Service Only user. However, admins can still use Delphix Self-Service and own a data container. Admins are also able to manage all data containers.
- A user who owns one or more data containers cannot be deleted.
    - For the list of data containers that a given user owns, see **User Details**.
- You cannot revoke a user's Self-Service Only role if they own any data containers.
    - For the list of data containers that a given user owns, see **User Details**.

12.1.9.2 Assigning a user to a data container

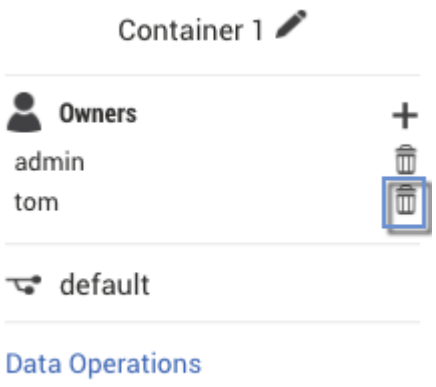
This section describes how to assign a user (created in the previous section) to a data container. Making a user the owner of a data container allows them to perform operations such as **Refresh** on that data container. Users cannot see or manipulate data containers that they do not own. You can either assign a user when creating a new data container or modify the owner of an existing data container.

12.1.9.2.1 Case 1: Assigning a user to a new data container

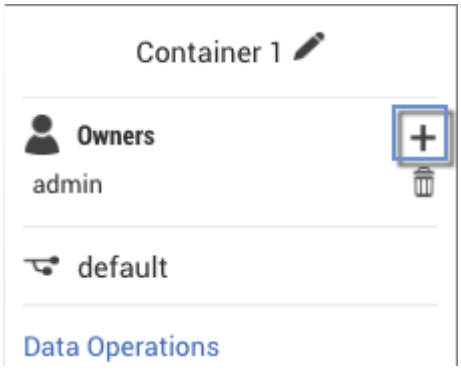
1. Navigate to the **Overview** page.
2. Select a **template** from which you want to create the data container. This will take you to the **Data Template** page.
3. In the upper right-hand corner of the screen, click **Add Container**.
4. Optional: Enter information about the data container, such as the **Name** and **Description**.
5. Select the **Owners** for the data container from the search box. It is acceptable to have multiple owners for each data container.
6. Select the **VDBs** to use for this container's data sources and Save.

12.1.9.2.2 Case 2: Changing the owner of an existing data container

1. On the **Management Overview** page, select the **data template** from which the data container was provisioned.
2. Click the **Containers** tile in the left-hand panel.
3. Select the **trash can** icon next to the username.



4. Click the **Plus** icon.



5. Select the desired **owner** from the drop-down list.

The user you selected is now the owner of the data container and can perform operations on that data container.

12.1.9.3 User details page

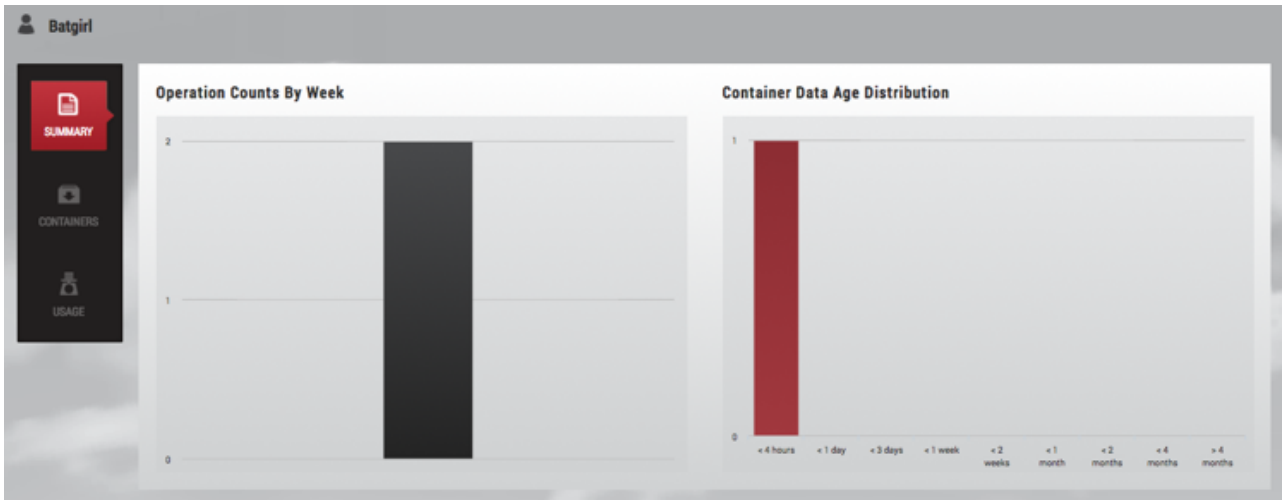
This section provides an overview of **Delphix Self-Service User Details**. This page displays graphs related to the user's Delphix Self-Service activity, as well as a list of all of the data containers that the user owns.

1. On the **Management Overview** page, click the **Users** tab.

2. Select the **name** of the desired user to go to their **User Details** page.

The following sections will display important information:

Section	Information displayed
Operation Counts By Week graph	Shows the aggregate of all Delphix Self-Service operations performed this user has performed on all of their containers.
Container Age Distribution graph	Shows the average time since a data operation was performed on all of the user's containers
Containers	Lists all containers that the user owns



User Details

12.1.10 Working with multiple container owners

Delphix Self-Service administrators can designate multiple users as owners of a single data container. These users all share access to the same data container, meaning actions taken by one user will impact all users on the same data container.

For example, if **user A** activates `branch X`, **user B** will also see `branch X` as the active branch. This ability for one user action to impact another user on the same containers creates new concerns for users sharing the same container.

As a result, more processes should be put into place to coordinate usage between users. Each team is different, but some effective strategies include:

- Designating a person to perform certain data operations.
- Saving your work with a bookmark or creating/working on a personal branch.
- Being aware of who is using your data container/data before performing operations.
- Locking a container to prevent others from performing any operations on it.

12.1.10.1 How many owners should a container ideally be shared between?

There is no technical limit built into the software, but it is best if a team of 5-10 users shares a single data container. In most cases, having fewer owners minimizes overhead and conflicting usage. One owner per container provides maximum productivity and minimal overhead, so this feature should only be used if your infrastructure or processes require that multiple users share a container. Additionally, Jet Stream-only users currently cannot see other users with whom they share the container.

12.1.10.2 How should users handle potentially disruptive operations?

If one user performs an operation on a data container, it will affect the other owners of that container. Additionally, each user has permission to perform the same operations on the data container; currently there are no fine-granularity permissions that limit the operations a user can perform. All operations are potentially disruptive, but the level of disruption varies by operation. If any of the following operations are performed at the same time, the second operation will fail due to a conflict when processing the job.

12.1.10.2.1 Conflicting operations

- Refresh
- Restore
- Reset
- Enable/Disable
- Create Branch
- Activate Branch
- Delete Branch
- Create Bookmark
- Delete Bookmark

If **user A** performs a destructive operation while **user B** is "using" the data container, the operation will destroy **user B**'s current state. Currently, the interface does not provide insight into whether the data container is in use by another user.

12.1.10.2.2 Destructive operations

- Refresh
- Restore
- Reset
- Enable/Disable
- Create Branch

- Activate Branch

12.1.10.2.3 Deleting objects

All owners can delete any bookmarks or branches in the container, regardless of who created them.

12.1.10.3 Coordinating users

Opportunity for disruption increases as more owners are sharing a single container. Sharing a container works best when users can communicate with each other, for example, they are part of the same team and working with the container at different times.

Additionally, these disruptions can be avoided when Delphix Self-Service users lock their containers to prevent other users from performing operations on them. Users cannot see the other users with whom they share the container. However, if a user locks a container, only that user will be able to use the container; it will appear disabled to others.

12.1.10.4 What operations could disrupt others using a container?

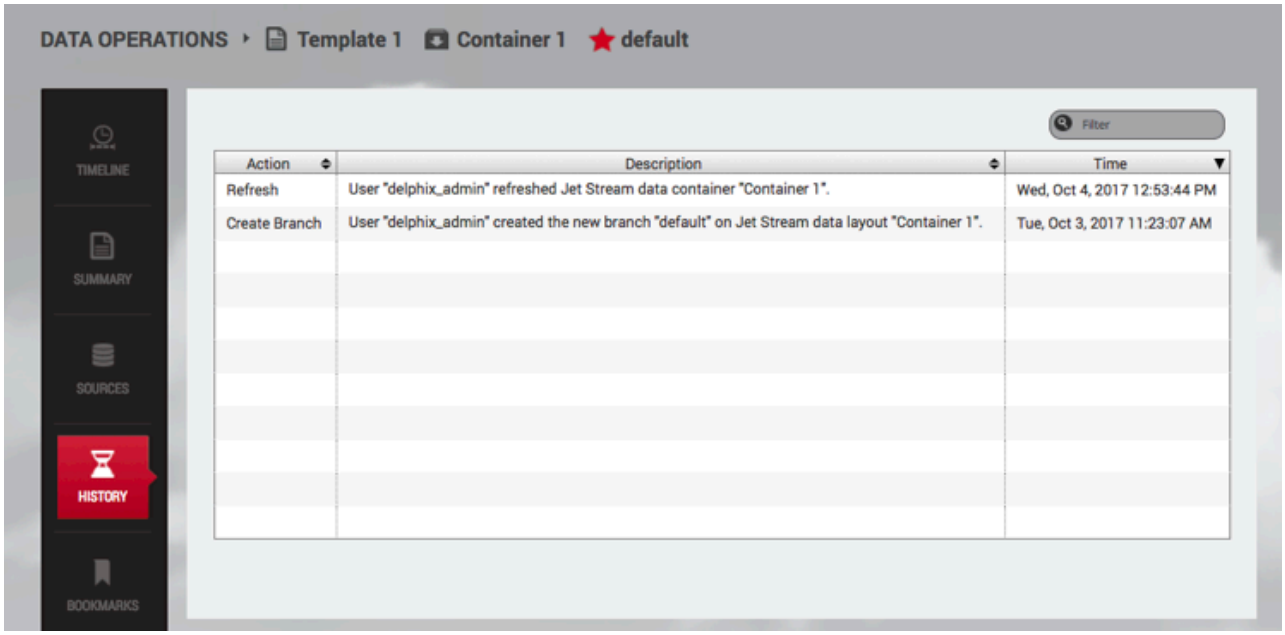
Potentially disruptive operations include:

- Refresh
- Switching active branches
- Deleting bookmarks
- Creating Branches
- Un-sharing bookmarks
- Restore
- Reset
- Staring/ stopping your container

12.1.10.5 Where can I see which user has performed what operation?

You can see which user has performed which action in the **History** tab of the **Data Operations** screen.

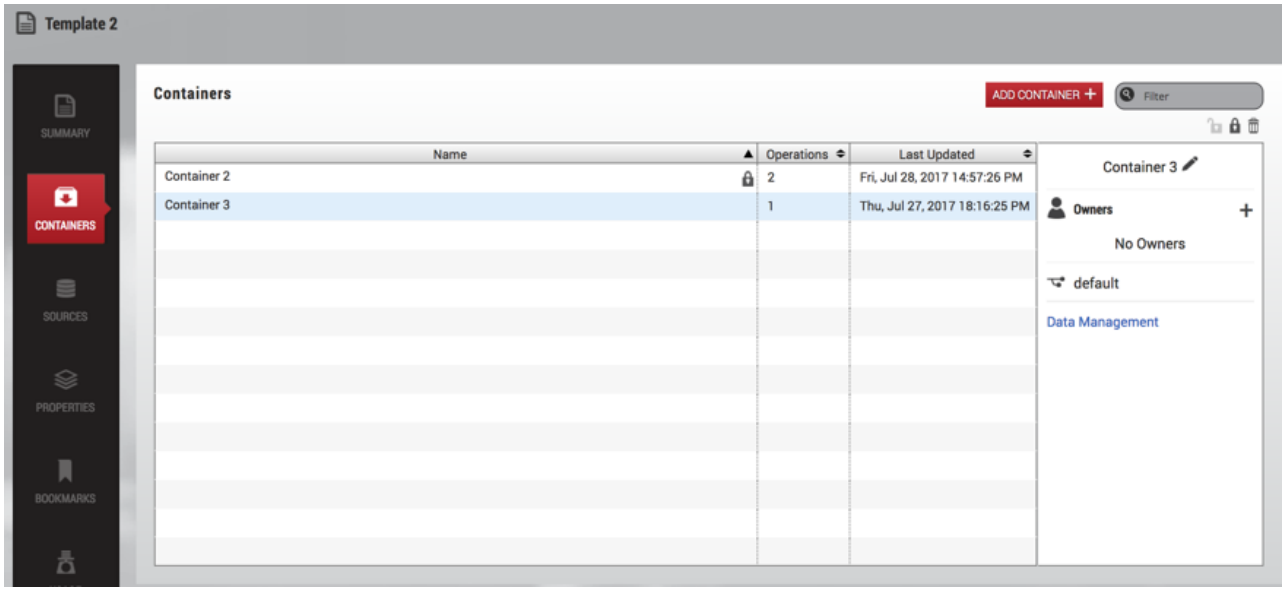
Be aware that operation counts in the template view are currently tabulated based on the container, not the user performing the operation.



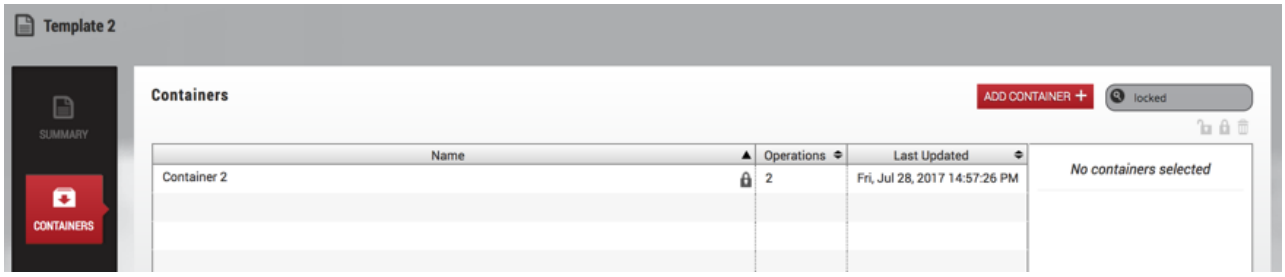
13 DATA OPERATIONS → HISTORY

12.1.10.6 Where can I see which containers are unlocked/locked?

Unlocked containers can be viewed by typing **unlocked** in the search filter.



You can view locked containers by typing **locked** in the search filter. To find containers locked by a specific user, type **locked by {username}** in the search filter.



12.1.11 Understanding bookmarks

12.1.11.1 Bookmarks overview

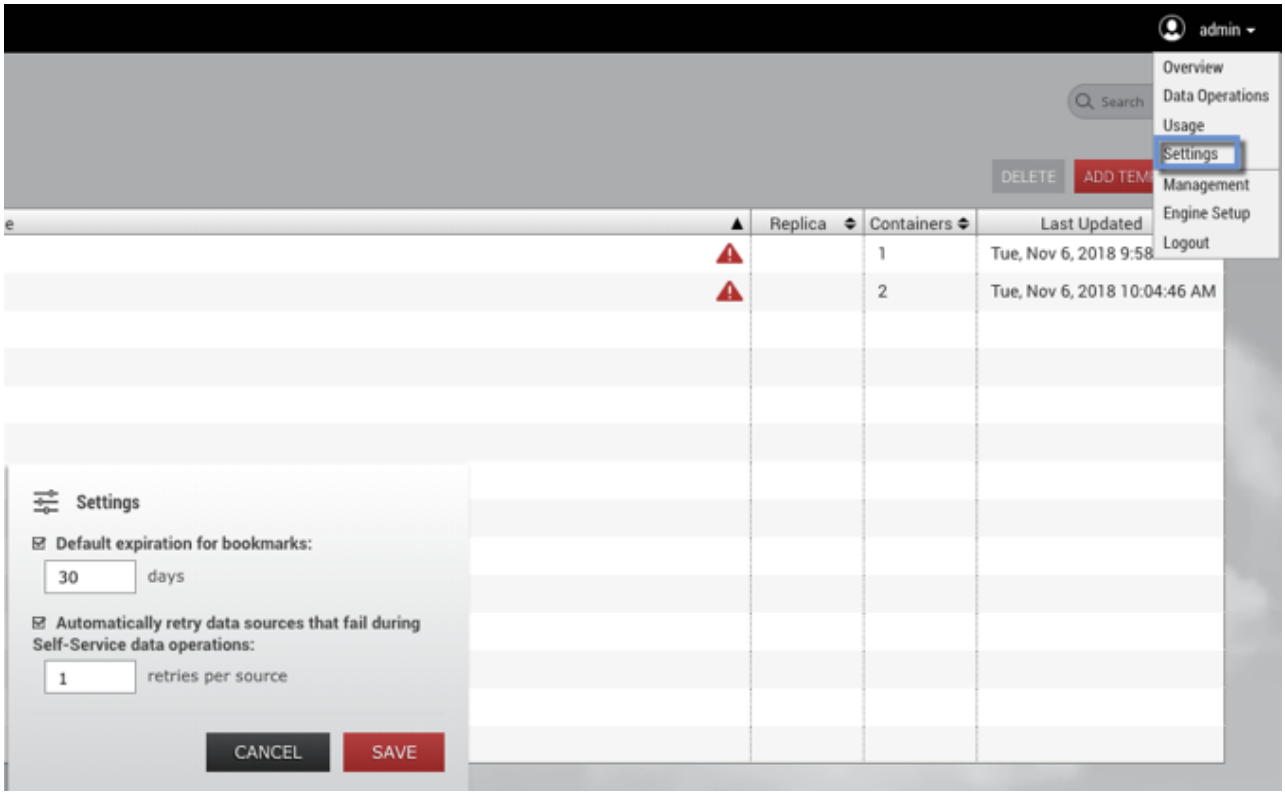
Bookmarks are a way to mark and name a particular moment of data on a timeline. You can restore the active branch's timeline to the moment of data marked with a bookmark. You can also share bookmarks with other users, which allows them to restore their own active branches to the moment of data in your container. The data represented by a bookmark is protected and will not be deleted until the bookmark is deleted. To help manage the space used by this data, users can set an optional expiration date for a bookmark. At the end of the set date, the bookmark will automatically be deleted. Once created, you can easily locate a bookmark through one of the bookmark viewers in the interface. To understand how to use bookmarks, please refer to the Delphix Self-Service Data User Guide. (see page 2479)

12.1.11.1.1 Using bookmarks in data templates

An admin user can create a bookmark on a template that will then be automatically shared to all containers created from that template. Additionally, an admin user can create a bookmark on the master template timeline with the point of time you are interested in. The bookmark will always be saved from retention policies, and a new branch can be created from this bookmark.

12.1.11.1.2 Default bookmark expiration

You can set a value that controls the default expiration time, in days, for Bookmarks. This setting only applies to new bookmarks that are created through the Delphix Self-Service application, not the CLI or API. Note that this only controls the default selection; users can still disable expiration or pick a different date for a specific bookmark if they wish. This setting is disabled by default.



12.1.12 Understanding Delphix self-service usage management

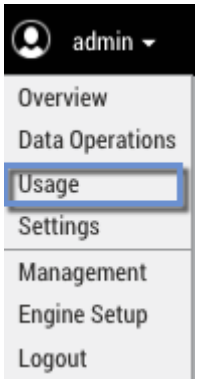
12.1.12.1 Usage management dashboard overview

Data templates are comprised of dSources, virtual databases (VDBs), and vFiles. These data sources are controlled by the standard policies configured in the **Management** application of the Delphix Engine. As with existing containers, space will be reclaimed by the retention policy over time. As retention cleans up historical data, users will no longer be able to use those points in time to restore or branch. In Delphix Self-Service, an admin can create a bookmark on the data template timeline, which will prevent retention from cleaning up the data that a bookmark references.

Data containers are comprised of VDBs provisioned from the sources defined in the data template. Similar to VDBs in the **Management** application, data containers' VDBs will share blocks with the source from which they are provisioned. This prevents the referenced data on the source from being cleaned up by retention. Retention for these VDBs is controlled by the standard Delphix Engine retention policies. As on templates, bookmarks in data containers will prevent storage from being reclaimed by retention. In addition, Self-Service will ensure that the latest data on each branch is never removed.

The **Usage** pages of the data templates and data containers provide information that can help you understand how storage is being used, how to reclaim space, and how much space you are able to reclaim.

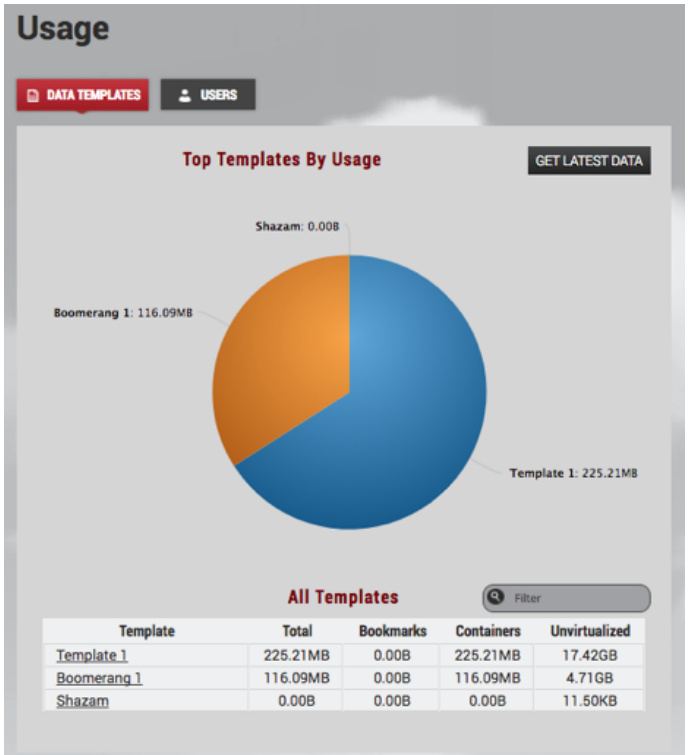
Usage Overview is a top-level page, along with the **Data Operations** and **Overview** pages. It contains the space usage breakdowns by data templates and users.



12.1.12.2 Template usage overview

The **Template Usage** page, seen in the image below, contains the usage breakdowns for data templates and users. The interface is interactive and allows you to visualize data by interacting with pie charts, bar graphs, and tables. The pie chart contains information about the top 10 space consumers; the table at the bottom contains information about all of the templates and/or users.

The table below the charts includes category fields. You can find corresponding descriptions by hovering over the names of the fields in the table:



The Template Usage page

Additionally, the table allows you to sort, navigate, and interact by clicking the field category of interest. For example, to sort the table, click a **column header** such as **Unvirtualized** and the table will sort by that category. To navigate to a particular data template or user, you can click either the **pie slice** or the **name** of the template/user in the table.

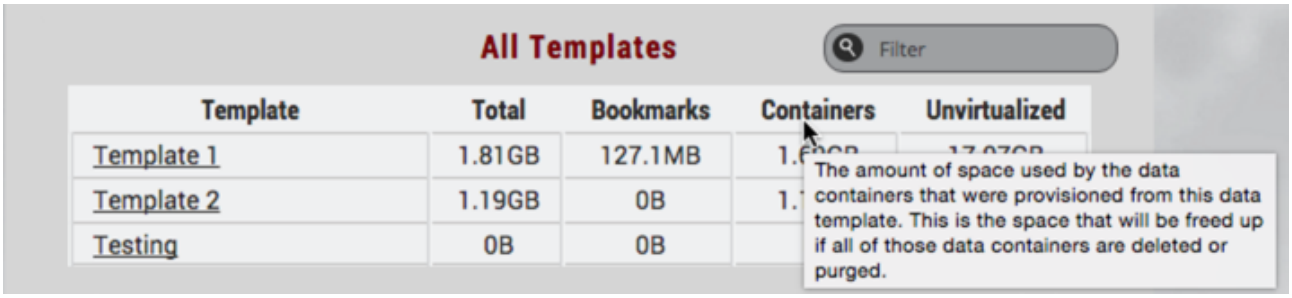


Table of templates/users

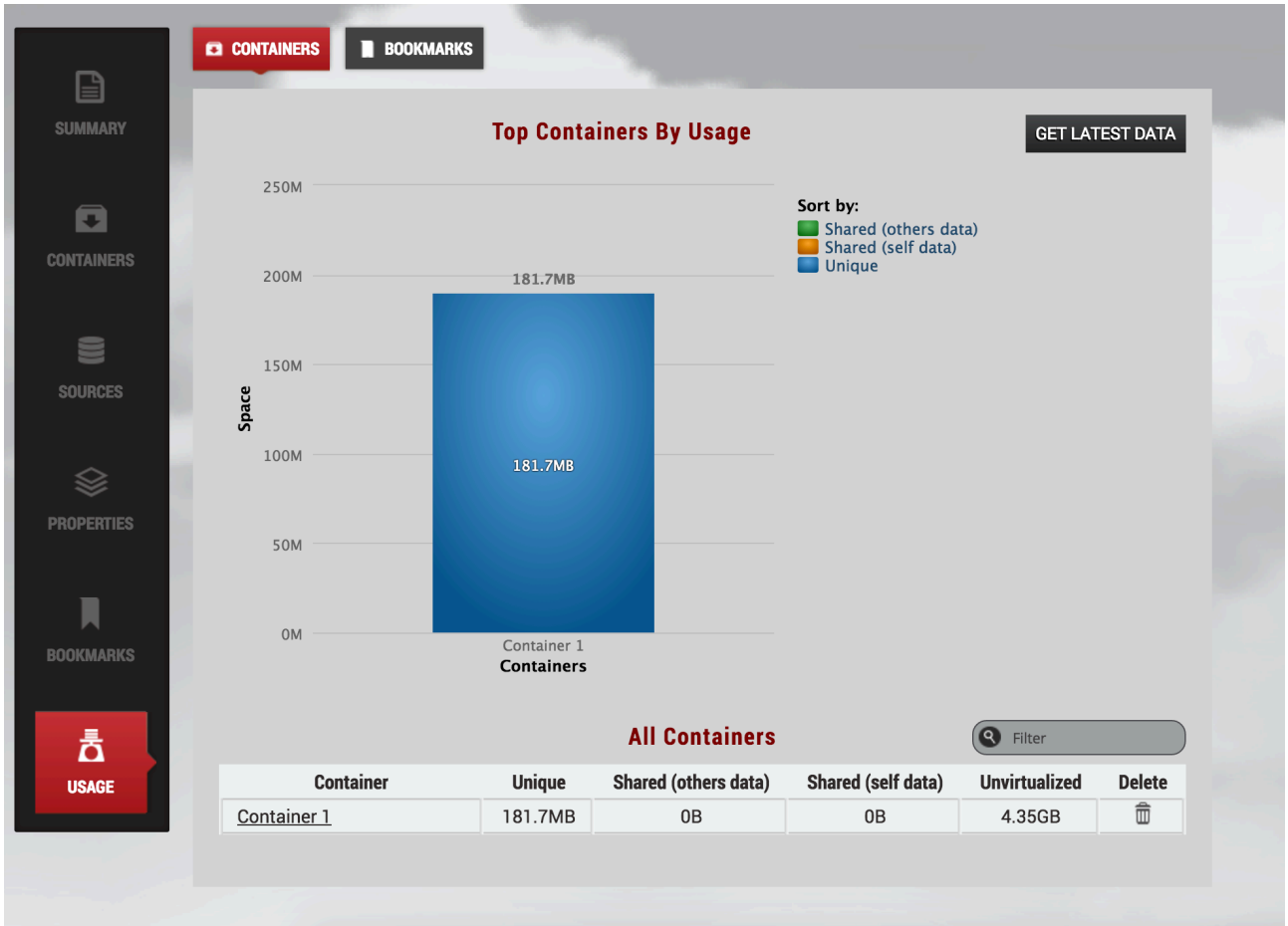
The field categories display the following information:

- **Total** – The sum of the space used by the data containers provisioned from this data template and by the bookmarks created on this template. This is the space that will be freed if you delete the template.
- **Containers** – The amount of space used by the data containers provisioned from this data template. This is the space that will be freed if you delete or purge all of the data containers.
- **Bookmarks** – The amount of space used by the bookmarks on this data template. This is the space that will be freed if you delete all bookmarks on the template.
- **Unvirtualized** – The amount of space that would be used by the data in this template and its child data containers without Continuous Data.

The pie chart and table graphs can help you analyze storage usage information.

12.1.12.3 Template usage details

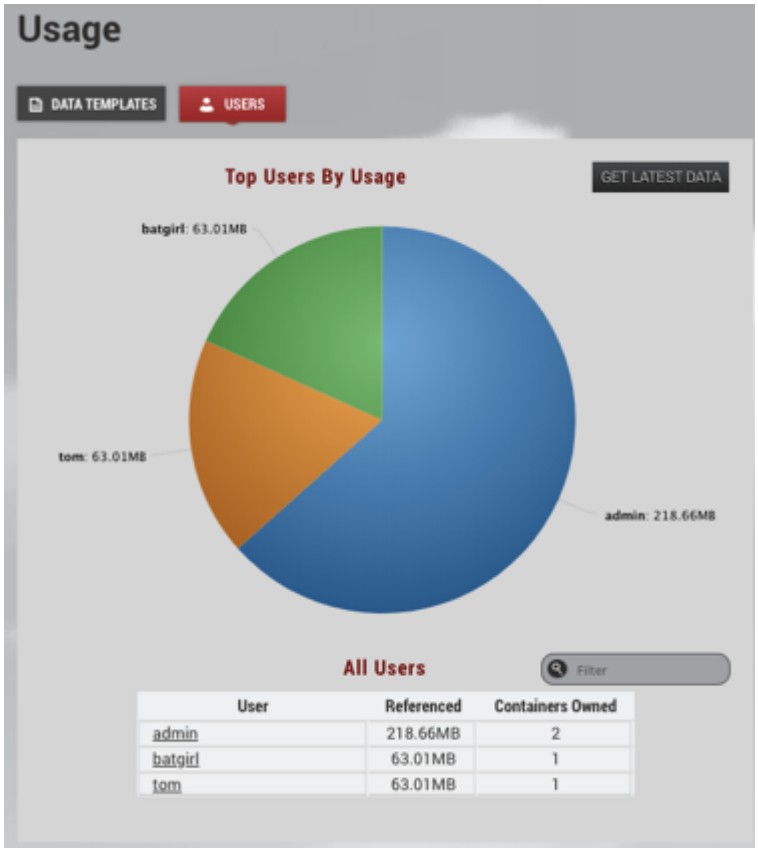
The **Usage** tile appears at the bottom of the Self-Service navigation sidebar, as seen in the image below. Usage summaries are available for templates, containers, and users. For example, when you click the **Usage** tile on the **Template Details** page, the usage details you interact with will be in the context of the selected data template.



The Usage tile

12.1.12.4 User usage overview

The **User Usage Overview** page provides graphical visualizations of space used by users assigned to data containers.



The field categories display the following information:

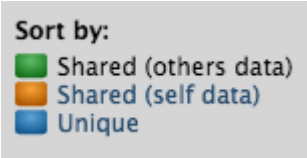
- **Referenced** – The amount of space used by data containers that are owned by this user. This excludes the space that this user is sharing with other users.
- **Containers Owned** – The number of data containers owned by this user

12.1.12.5 Template usage (Containers) overview

The **Template Usage Details** page, as seen below, shows the space used by data containers provisioned from the template and the bookmarks created on the template.

Container Usage

The stacked bar graph shows information about the top 10 space users. You can re-sort the graph based on the fields in the **Sort by** the legend on the top right-hand corner of the screen as seen in the image above. For example, if you want to know which data containers are sharing the most data with others, you can un-select **Shared (others data)** and **Unique** by clicking them in the legend.



The Sort by legend

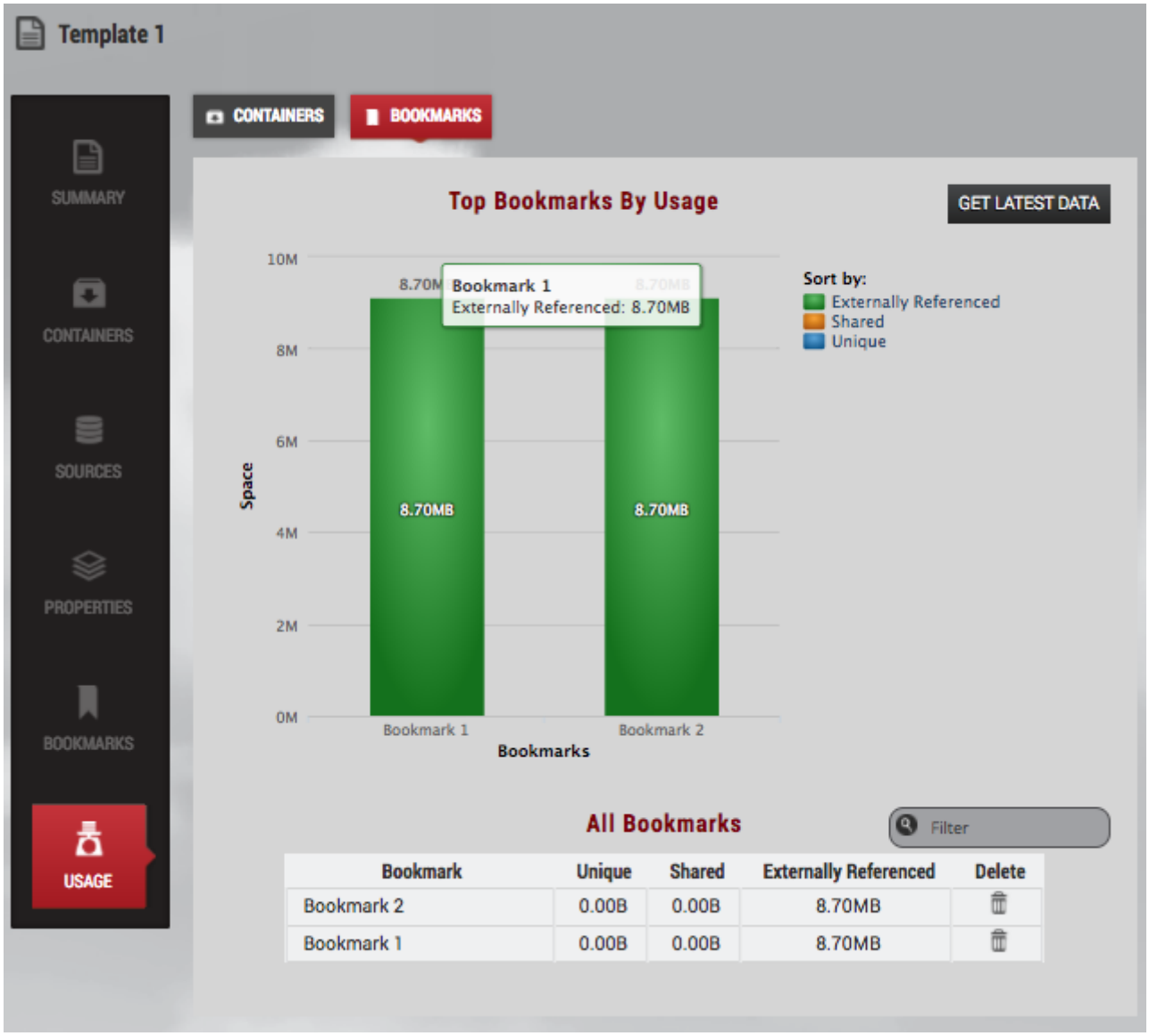
> Legend Items When the legend items are not selected, their corresponding colored boxes turn gray and the data is removed from the chart. The data and name will reappear when you re-select by clicking the grayed-out category you want.

The field categories display the following information:

- **Unique** – The amount of space that will be freed if you delete this data container. This assumes that it also deletes underlying data sources.
- **Shared (others data)** – The amount of space that cannot be freed on the parent data template (or sibling data containers) because it is also being referenced by this data container due to Restore or Create Branch operations. The snapshots on the template or sibling container are what use up space.
- **Shared (self data)** – The amount of space that cannot be freed on this data container because it is also being referenced by sibling data containers due to Restore or Create Branch operations, via shared bookmarks
- **Unvirtualized** – The amount of space that would be used by the data in this container without Continuous Data

12.1.12.6 Template usage (Bookmarks) overview

As shown in the image below, the **Template Usage Details** page provides the usage information about bookmarks created on a template. The primary categories of information include **Unique**, **Shared (others data)** and **Shared (self data)**.



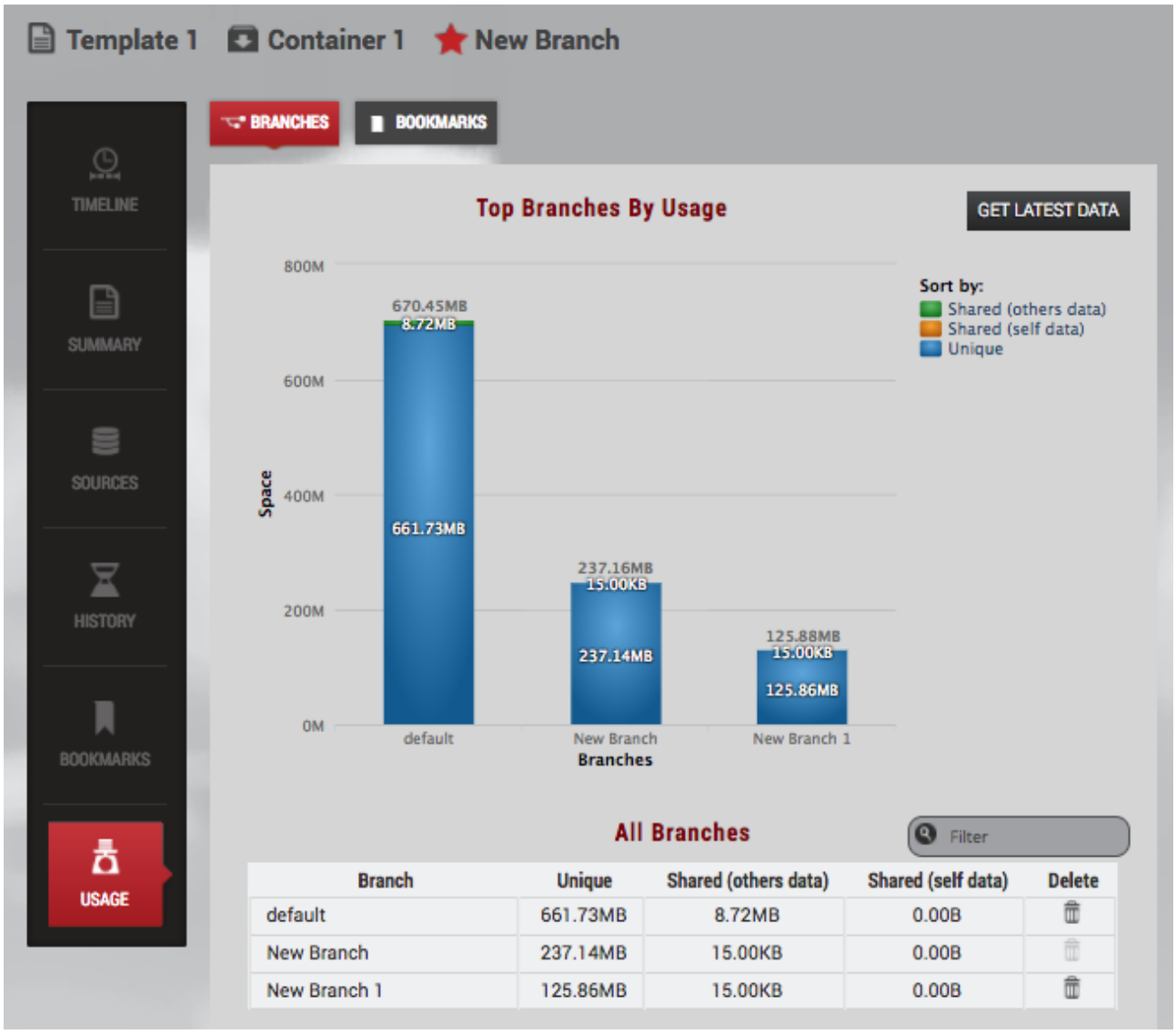
Template Usage (Bookmarks)

The field categories display the following information:

- **Unique** – The amount of space that will be freed if you delete this bookmark
- **Shared** – The amount of space referenced by this bookmark that cannot be freed by deleting this bookmark because it is also referenced by neighboring bookmarks or branches that have been created or restored from this bookmark
- **Externally Referenced** – The amount of space referenced by this bookmark that cannot be freed by deleting this bookmark because it is also being referenced outside of Self-Service – for example, by a retention policy.

12.1.12.7 Container usage (Branches) overview

The **Container Usage Details** page shows the usage information about the branches and bookmarks created on a container. The primary categories of information include **Unique**, **Shared (others data)**, and **Shared (self data)**.



The Container Usage Details page

The field categories display the following information:

- **Unique** – The amount of space that will be freed if you delete this branch
- **Shared (others data)** – The amount of space that cannot be freed on the parent data template or sibling branches because it is also being referenced by this branch due to Restore or Create Branch operations. The snapshots on the template or sibling container are what use up space.

- **Shared (self data)** – The amount of space that cannot be freed on this branch because it is also being referenced by sibling data containers due to Restore or Create Branch operations, via shared bookmarks.

12.2 Delphix self-service data user guide

Welcome to Delphix self-service (see page 2480)

Delphix Self-Service grants access to the data that users need, whenever they need it. Once users have been assigned a data container, they can control the data available within it. This means they can refresh to the latest production data, roll bac... (see page 2480)

Updated on : 25 May 2023

Delphix self-service data concepts (see page 2482)

Understanding Data Sources A data source can be a database, an application, or a set of unstructured files. Engine Administrators configure the Delphix Engine to link to data sources, which pulls in the data of these sources. The Delphix Engine ... (see page 2482)

Updated on : 25 May 2023

Delphix self-service user interface (see page 2484)

The User Interface is organized within a single web browser page. The screen serves as a data container report and management panel. Data Container Report Panel Data Container Workspace Data Container Report Panel The Data Container Re... (see page 2484)

Updated on : 25 May 2023

Understanding timelines and how to preserve data in a point in time (see page 2487)

Understanding Timelines Branch Timeline A branch timeline acts as a dynamic point-in-time interface for user actions within the branch. You can interact with the source data in the active branch by using both the branch timeline and icons along ... (see page 2487)

Updated on : 25 May 2023

Data container activities (see page 2500)

Getting Started Data containers can be shared between multiple users. In this situation, users should coordinate with their co-owners when performing data operations that could disrupt other users' workflow such as stopping or refreshing the data c... (see page 2500)

Updated on : 25 May 2023

Containers with multiple owners (see page 2511)

Delphix Self-Service administrators can designate multiple users as owners of a single data container. These users all share access to the same data container which means actions taken by one user will impact all users on the same data container. F... (see page 2511)

Updated on : 25 May 2023

Working with bookmarks in a data container (see page 2513)

Working with bookmarks is an easy way to share data with other users of any container created from the same template. By sharing with others, you can integrate testing, development, and QA needs. For example, in the past, if you found a bug you woul... (see page 2513)

Updated on : 25 May 2023

Understanding Delphix self-service usage (see page 2516)

Usage Management Dashboard Overview Data templates are comprised of dSources, virtual databases (VDBs), and vFiles. These data sources are controlled by the standard policies configured in the Management application of the Delphix Engine. As w... (see page 2516)

Updated on : 25 May 2023

12.2.1 Welcome to delphix self-service

Delphix Self-Service grants access to the data that users need, whenever they need it. Once users have been assigned a data container, they can control the data available within it. This means they can refresh to the latest production data, roll back to a previous point in the data container's timeline, and share data with another user without requiring any involvement from Information Technology or database administrators (DBAs). Delphix Self-service data management allows developers to be more productive while using fewer resources, dramatically improving operational efficiency.

12.2.1.1 User roles and permissions

Self-Service has two types of users:

12.2.1.2 Admin user

Admin users have full access to all report data and can configure Delphix Self-Service. Additionally, they can use the Delphix Engine to add/delete reports, add/delete users, change tunable settings, add/delete tags, and create and assign data templates and containers.

12.2.1.3 Data user

Data users have access to production data provided in a data container. The data container provides these users with a playground in which to work with data using the self-service toolbar.

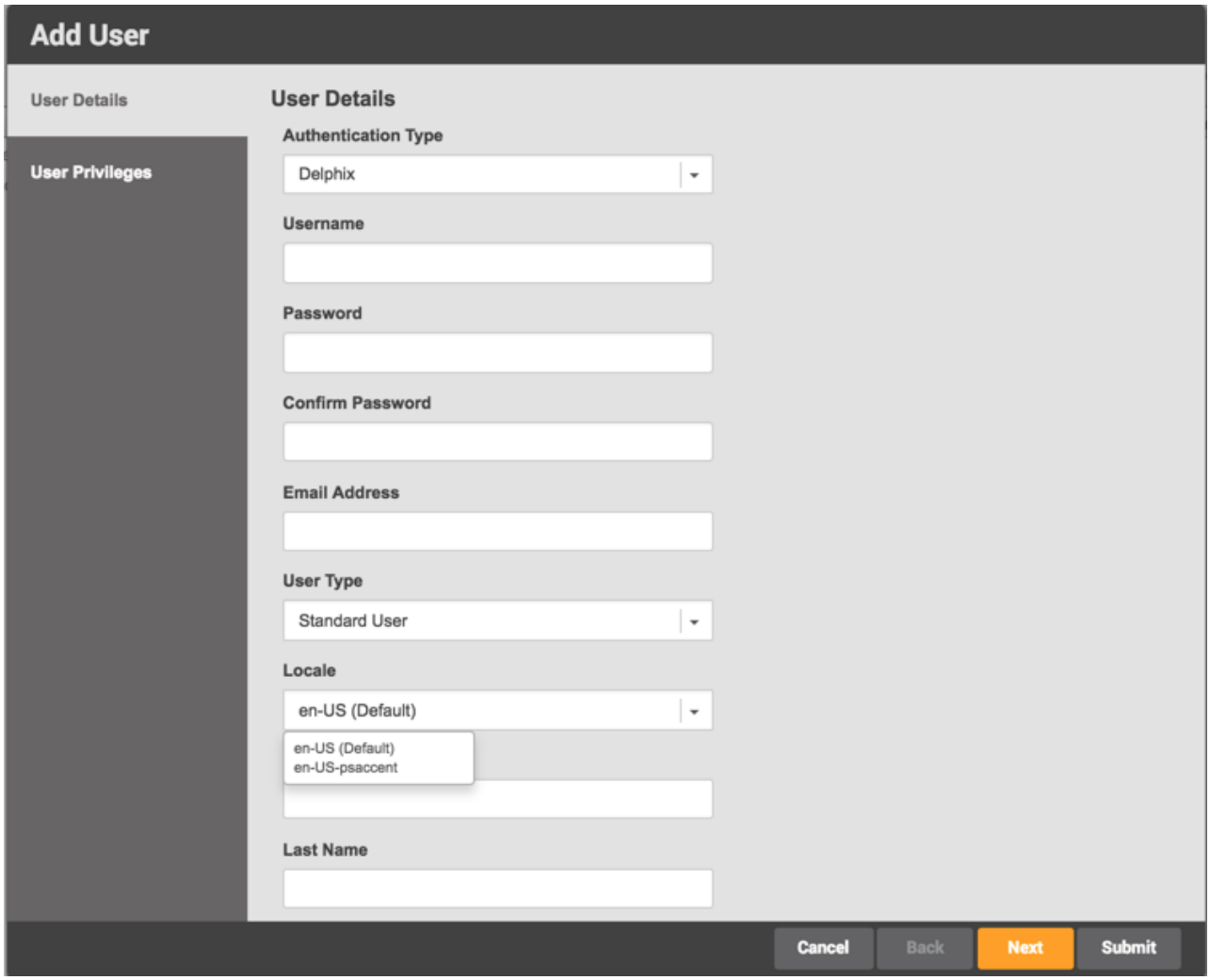
12.2.1.4 Login

1. Access Delphix Self-Service by opening a web browser using the **IP address** or **DNS qualified hostname**.
2. Login with the **User ID** and **Password** the Delphix Administrator has provided for you.

12.2.1.5 Changing your default locale

As a user, you can change your default locale by doing the following:

1. Click the **user login** icon in the upper right-hand corner of the screen.
2. Click the **Locale** drop-down menu.
3. Select the desired locale.



User Profile window

12.2.2 Delphix self-service data concepts

12.2.2.1 Understanding data sources

A data source can be a database, an application, or a set of unstructured files. Engine Administrators configure the Delphix Engine to link to data sources, which pulls in the data of these sources. The Delphix Engine will periodically pull in new changes to the data, based on a specific policy. This, in turn, begins building a custom timeline for each data source. Additionally, the Delphix Engine can rapidly provision new data sources that are space-efficient copies, allowing users to work in parallel without impacting each other.

12.2.2.2 Understanding data templates

Data templates are the backbone of data containers. They are created by the Engine Administrator and consist of the data sources that users need in order to manage their data playground and their testing and/or development environments. Data templates serve as the parent for a set of data containers that the administrator assigns to users. Additionally, data templates enforce the boundaries for how data is shared. Data can only be shared directly with other users whose containers were created from the same parent data template.

12.2.2.3 Understanding data containers

A data container allows data users to access and manage their data in powerful ways. Their data can consist of application binaries, supporting information, and even the entire database(s) that underlie it.

A data container allows users to:

- Undo any changes to their application data in seconds or minutes
- Have immediate access to any version of their data over the course of their project
- Share their data with other people on their team, without needing to relinquish control of their own container
- Refresh their data from production data without waiting for an overworked DBA

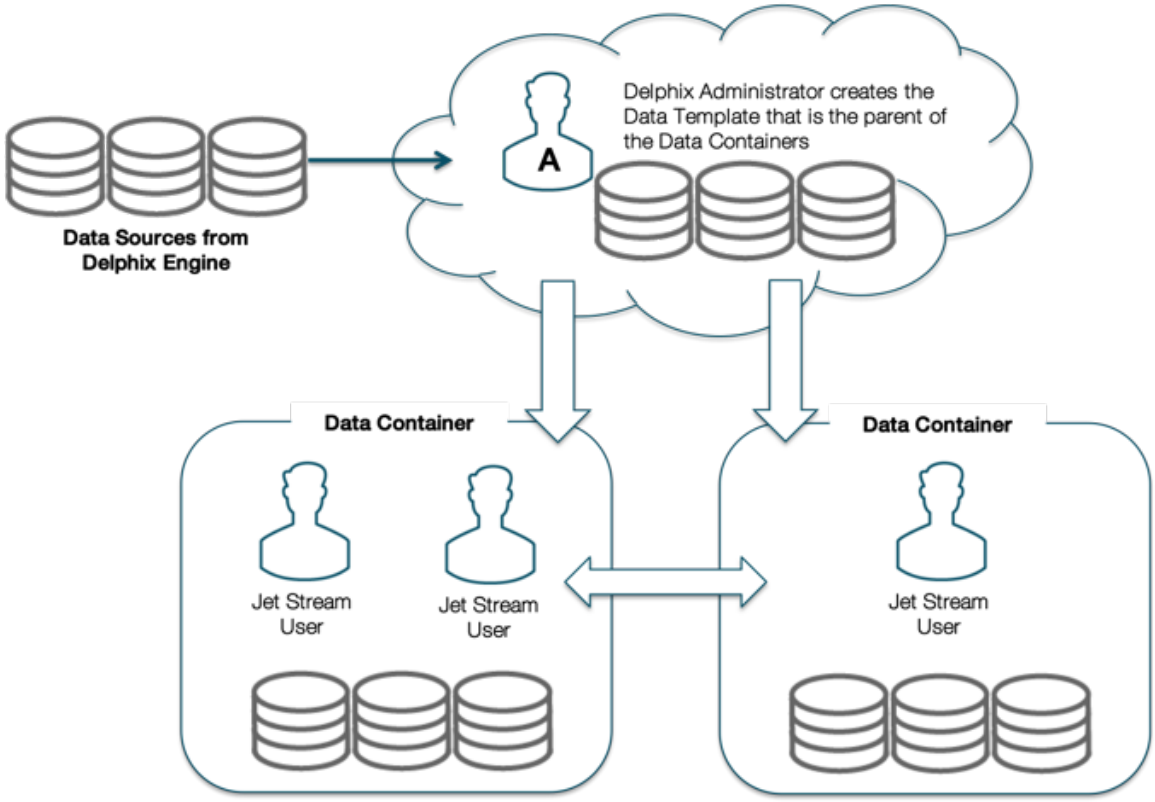
A data container consists of one or more data sources, such as databases, application binaries, or other application data. The user controls the data made available by these data sources. Just like data sources in a template, changes that the user makes will be tracked, providing the user with their own data history.

The **Data Container Interface** lets users view the details and status of their data container and its associated data sources, as well as manipulating which data is in those sources. The **Data Container Interface** includes a section called the **Data Container Report Panel**, which displays details about each source, including the connection information needed to access it – for example, the java database connectivity (JDBC) string for a database. This connection of information is persistent and stable for the life of the data container, regardless of what data the resources are hosting.

12.2.2.4 Data flow

The Delphix Self-Service data flow diagram below demonstrates how a Delphix Self-Service data user accesses data sources. Data sources are connected to a Delphix Engine, which is controlled by the Engine Administrator. The Engine Administrator will connect all data sources that developers and quality assurance

(QA) teams need to a data template. This data template acts as a parent source to create the data containers that the administrator will assign to data users. Data sources flow from the Delphix Engine into a data template and downstream into a data container, where a data user or users will use the data sources to complete tasks. The data container acts as a self-contained testing environment and playground for the data user. Additionally, data users are able to set, bookmark, and share data points in their container with other data users of other data containers, as long as all the data containers were created from the same parent data template.



Data Flow

12.2.2.5 Understanding branches

You can organize data in the data container into task-specific groupings, called "branches." For example, you can use a branch to group all the data you have used while addressing a particular bug, testing a new feature in an application, or exploring a business analytics scenario. By default, Delphix Self-Service automatically creates the first branch of source data for you when you login for the first time. You can view the default branch and any additional branches that you create over time by clicking the Branch tab. Additionally, to the right of the default branch, you will see an interconnected branch timeline unique to whichever branch is currently active. The illustration below displays both the default branch in the Branch tab of the Data Container View Panel and the default branch timeline.

Branch Tab

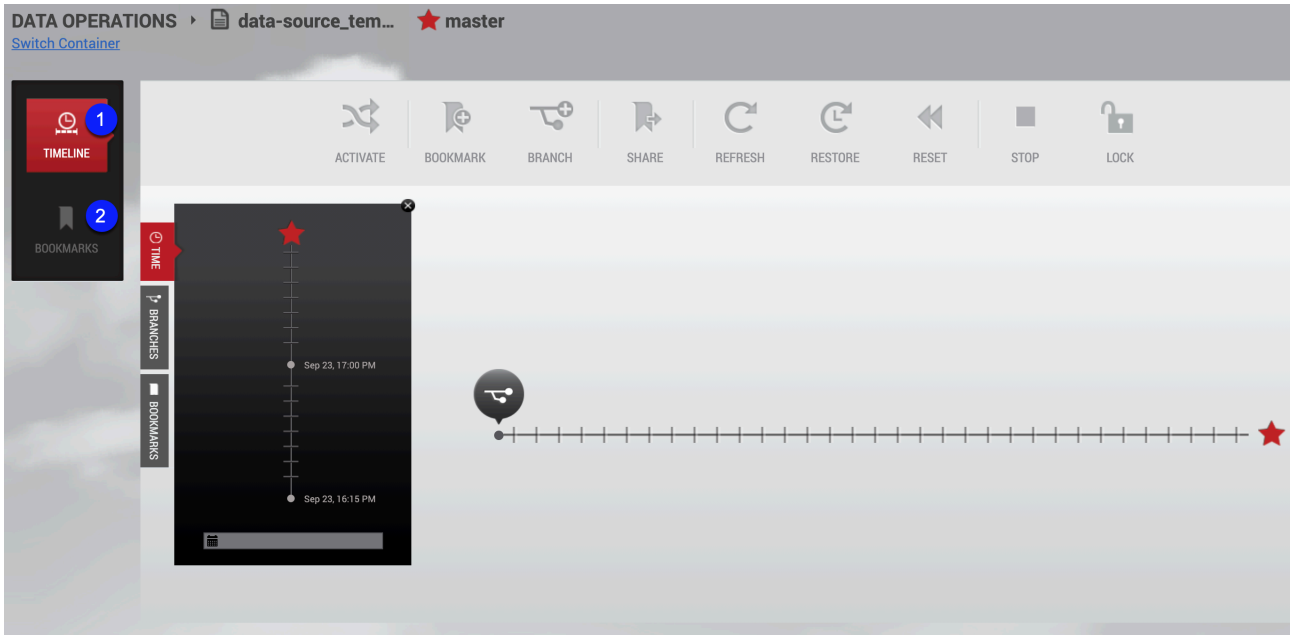
Branch View Panel and Branch Timeline

A branch is used to track a logical task and contains a timeline of the historical data for that task. One branch is the "active" branch, which means that it is the branch that is currently being updated with new data from the data sources. At any time, you can change which branch is active and thus change which data is in the associated data sources.

12.2.3 Delphix self-service user interface

The User Interface is organized within a single web browser page. The screen serves as a data container report and management panel.

12.2.3.1 Data container report panel



12.2.3.2 Data container report panel

The Data Container Report Panel for users consists of two tile buttons. They are summarized below as **Timeline** and **Bookmarks**.

12.2.3.3 Timeline

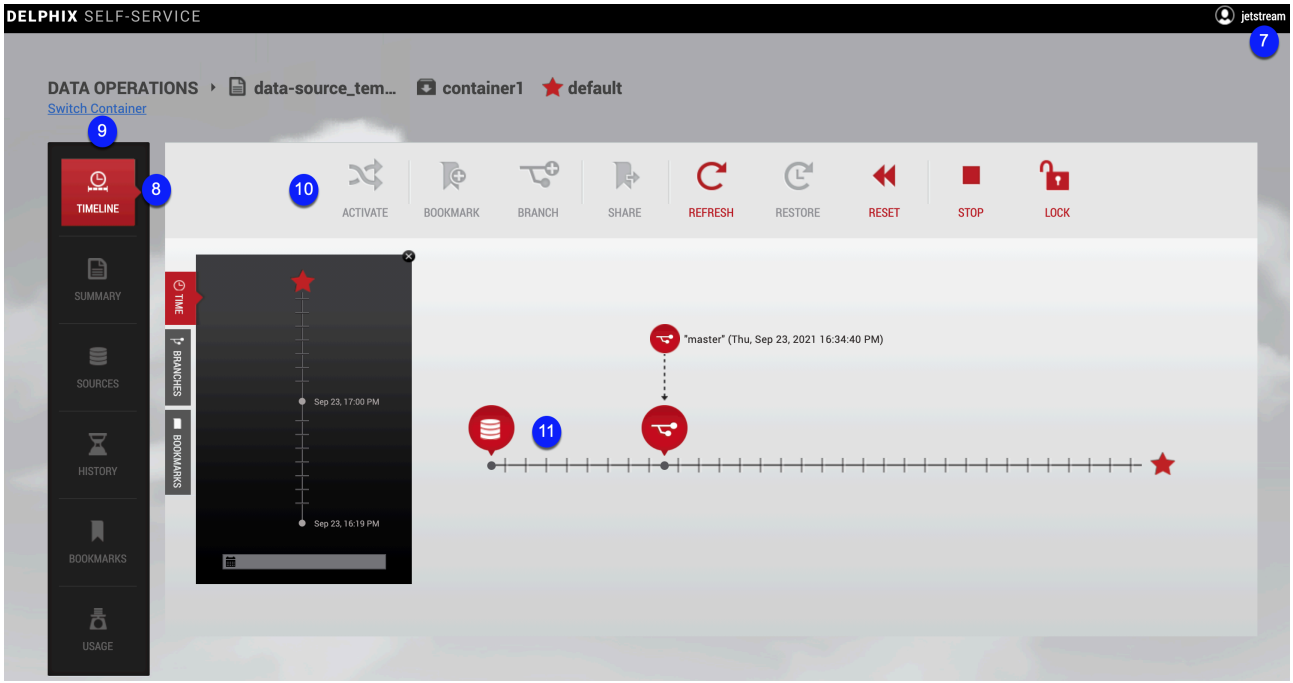
1. The **Timeline** tile allows you to view the timeline associated with a branch. Note that this only shows the timeline for a single branch. The branch timeline is how a user interacts with data in the container to mark, stamp, and perform tasks that occur at various points in time.

Bookmarks

2. The **Bookmarks** tile allows you to view and edit details about bookmarks within this data container and bookmarks accessible from it.

12.2.3.4 Data container workspace

The Data Container workspace is reached by selecting a template from the Management Overview page. Select a container and then select the Data Management link.



12.2.3.5 Data container workspace

The Data Container Workspace contains all the tools, actions, and view panels needed to begin using Self-Service features. For example, the workspace allows a user to view the history of their data on a branch, and to refresh, reset, and restore that data.

12.2.3.6 User login and settings drop down menu

7 The **user login** icon in the upper right-hand corner of the screen provides a drop-down menu with options to change your password and/or log out.

12.2.3.7 Data container view panel

8 The **Data Container View** Panel, found on the left-hand side of the screen, is divided into three tabular sections: time, branches, and bookmarks. These tabs allow you to find and select data that you are interested in. Based on user selections made in the view panel, the corresponding branch timeline can change.

12.2.3.8 Switch container

9 Allows you to switch between containers.

12.2.3.9 Data container self-service toolbar

10 The Data Container Self-Service Toolbar allows you to perform tasks and activities with data in the current container, by clicking on the following user action icons:

- **Activate** will make a branch active
- **Bookmark** will mark an interesting point of data on a branch timeline
- **Branch** will create a branch that supports one task. A branch is a group of data time segments called a "timeline."
- **Share** will share a bookmark with users of other data containers from the same template
- **Refresh** will refresh each source in the data container on a branch timeline to the latest data in the corresponding source of the data template.
- **Restore** will restore the data to a point in time from the template, the container, or a shared bookmark.
- **Reset** will reset to the last interesting moment of data time on the current data timeline
- **Stop** will stop a data container
- **Start** will start a data container
- **Lock** will lock a data container for the current user
- **Unlock** will unlock a data container.

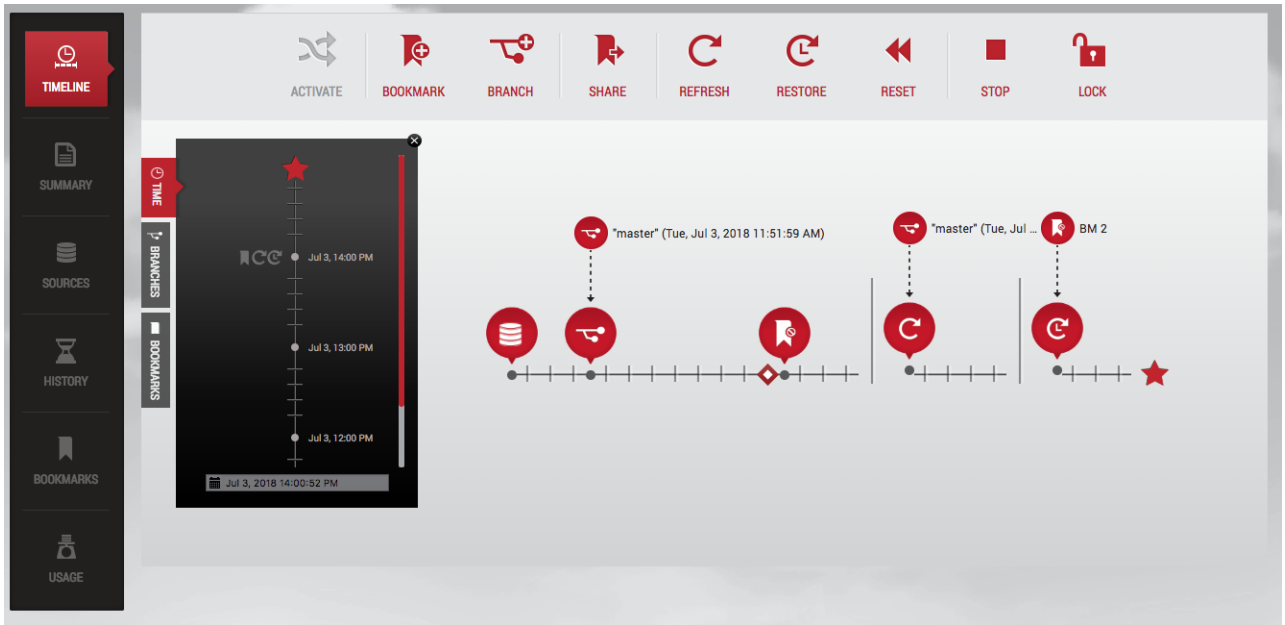
12.2.3.10 Branch timeline

11 Use this to view the timeline associated with a branch. Note that this only shows the timeline for a single branch. The branch timeline is how a user interacts with data in the container to mark, stamp, and perform tasks that occur at various points in time.

12.2.4 Understanding timelines and how to preserve data in a point in time

12.2.4.1 Branch timeline

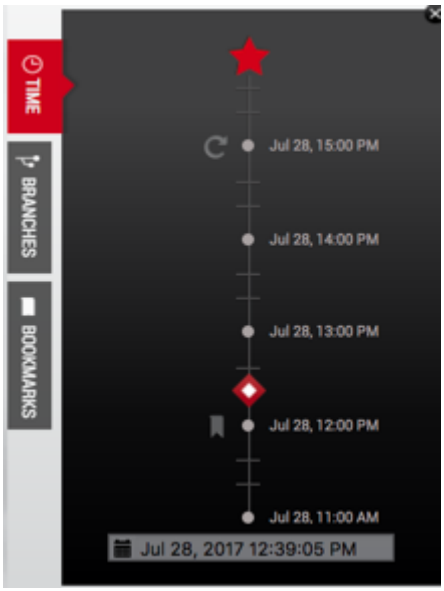
A branch timeline acts as a dynamic point-in-time interface for user actions within the branch. You can interact with the source data in the active branch by using both the branch timeline and icons along the **Self-Service Toolbar** at specific points in time. Common activities include re-setting data sources to run a test, refreshing the data container with the most current source data, and bookmarking data to share or track interesting moments of time along the branch timeline. Users work with one branch at a time to perform a series of actions related to a particular testing or debugging task such as data updates or starting and stopping data. The initial branch on your data container may include data from before the data sources were made into a container. As you work within your data container, you can create more branches overtime to run or complete separate tasks. Additionally, the data container tracks each branch and the corresponding actions you perform on the branches. To view the actions completed over the life of a branch, see the container timeline in the **Time** tab of the **Data Container View Panel**.



14 Branch with timeline segments over the life of the branch.

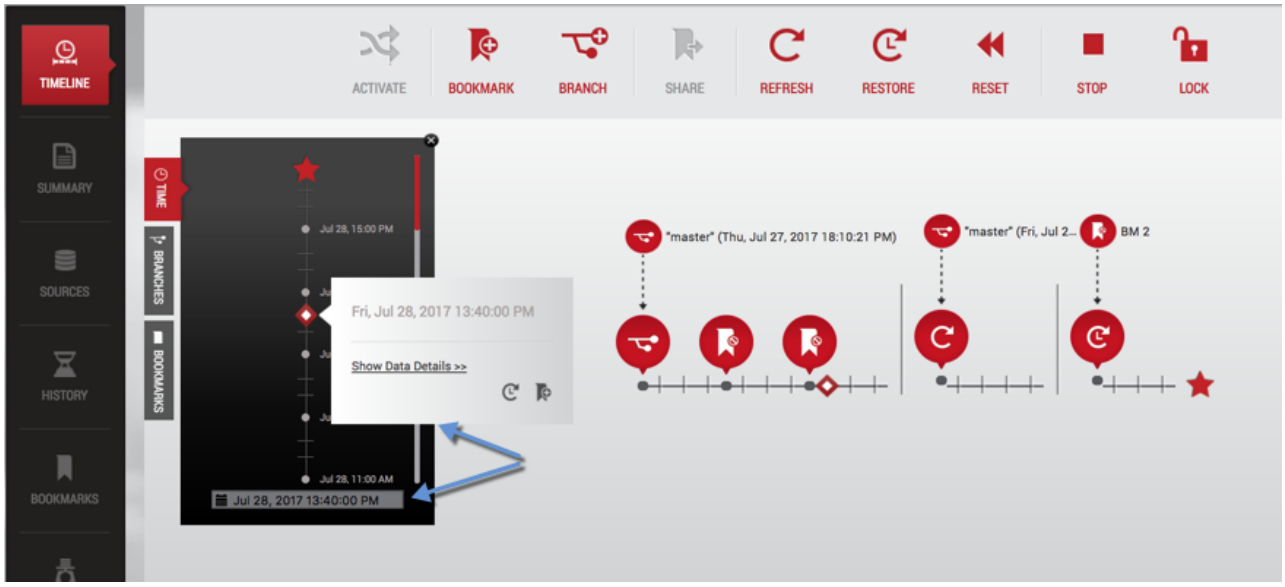
12.2.4.2 Container Timeline

The **Time** tab displays the data container's timeline, which acts as a wall clock of time. It shows continuous real-time across all branches and timeline segments. You can scroll up and down in the container timeline to find the point in time that interests you.



15 Time tab timeline.

Clicking a point in time in the container timeline will display the corresponding branch timeline capturing any actions performed on the branch. Additionally, should you need to select a time between tick-marks, you can use the **time input field** in the time selector on the left side of the screen.



16 Container timeline.

12.2.4.2.1 Selecting a Point in Time with the Time Selector

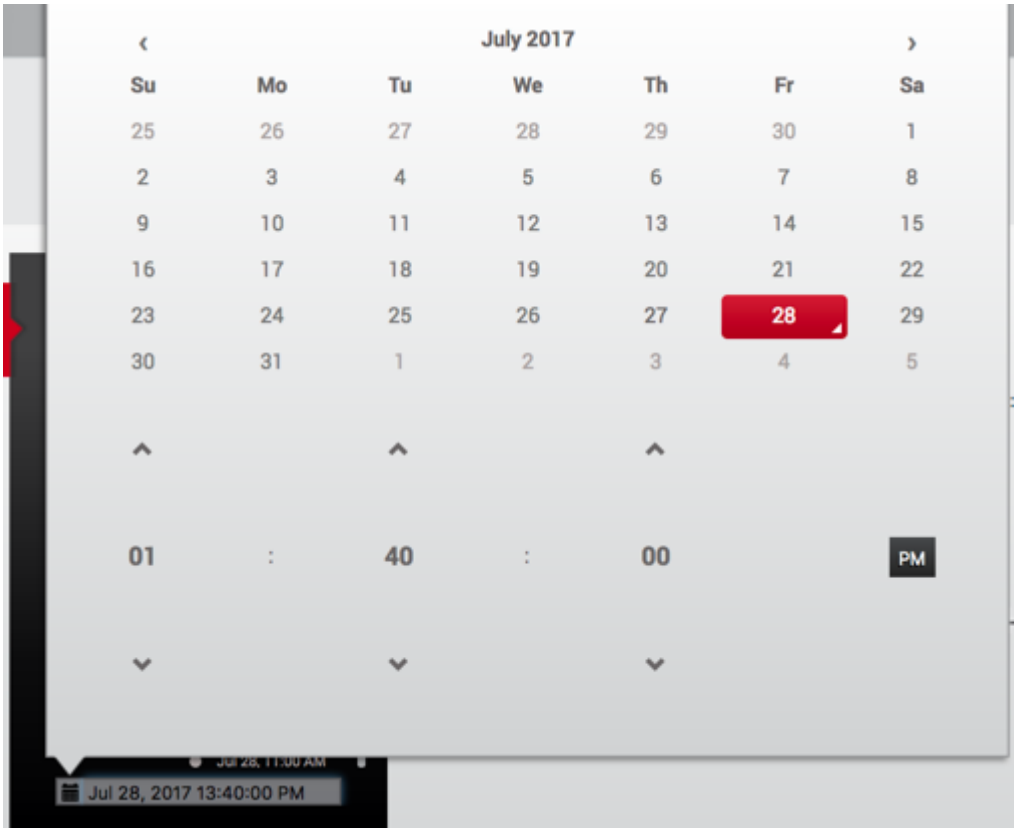
1. In the **time selector**, type in a **date** and **time** with the following format: Month/Day/Year Hour:Minute:Second{am|pm}. For example: 1/26/2015 1:14:13pm.
2. Press **Enter**.

The time input field will show the selected time. Now that you have entered the specific time you want, you can use the toolbar to select the data operation that you want to be performed at this point in time. Data operations can include Create Bookmark, Create Branch, and Restore.

> Invalid Time Value
> If you type in an invalid time value, or a time that is out of range, the value you typed in will revert to the previous default that existed before.

Selecting a Point in Time with the Time Selector Calendar

1. In the time selector, click the **calendar** icon to the left of the input field.
2. From the window that appears, click the **date** you want to use.
3. Select and **time** you want to use.



4. In the toolbar, click the button for the data operation that you want to perform at this point in time. Data operations can include Reset, Create Branch, and Create a Bookmark.

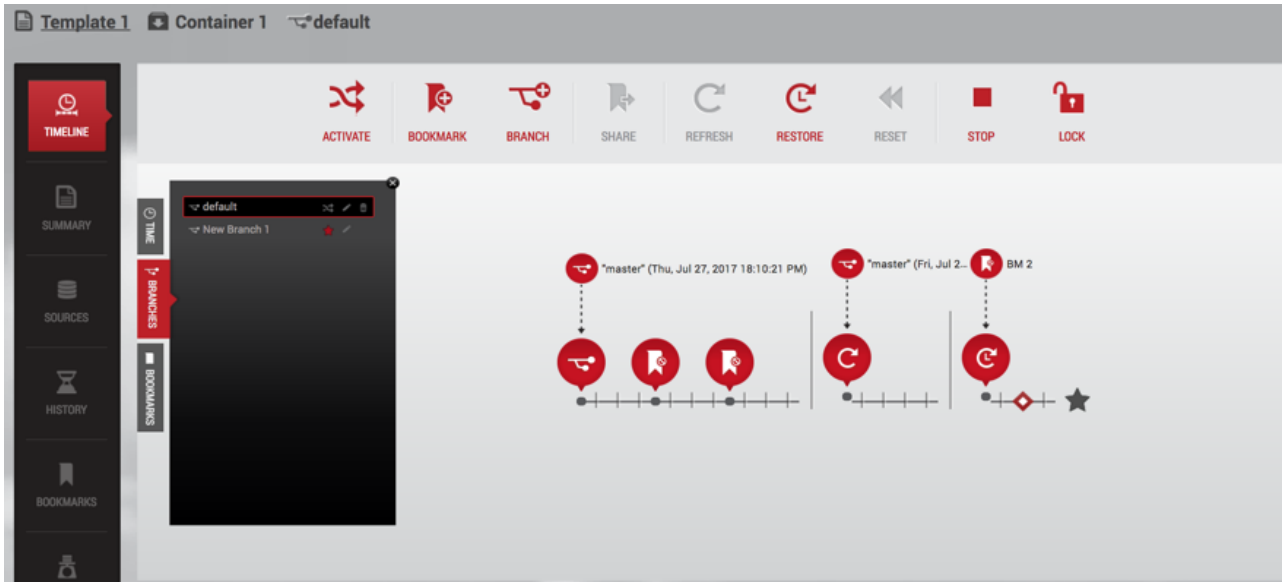
> Picking a date
> The flyout will not let you pick a date that is before the first point of data time in the container, or after the present moment.

12.2.4.3 Understanding the self-service toolbar

The **Self-Service Toolbar** contains self-service action icons that represent available actions a data user can perform. You can distinguish between available and unavailable icon actions by the use of color on the toolbar. Actions available to you will be red, and actions that are unavailable will be grey. All actions are dynamic, and availability will change based on how you use and work with data in both the branches and data container(s) that are assigned to you.

17 Self-Service toolbar.

For example, your options for actions on the **Self-Service Toolbar** can change if the branch of the branch timeline you are working with is activated. In the illustration below, the screen shows a user working in an active branch. Notice the bright red star at the end of the timeline. This indicates that the branch is active. Also, notice which actions are and are not available to the user on the **Self-Service Toolbar**.

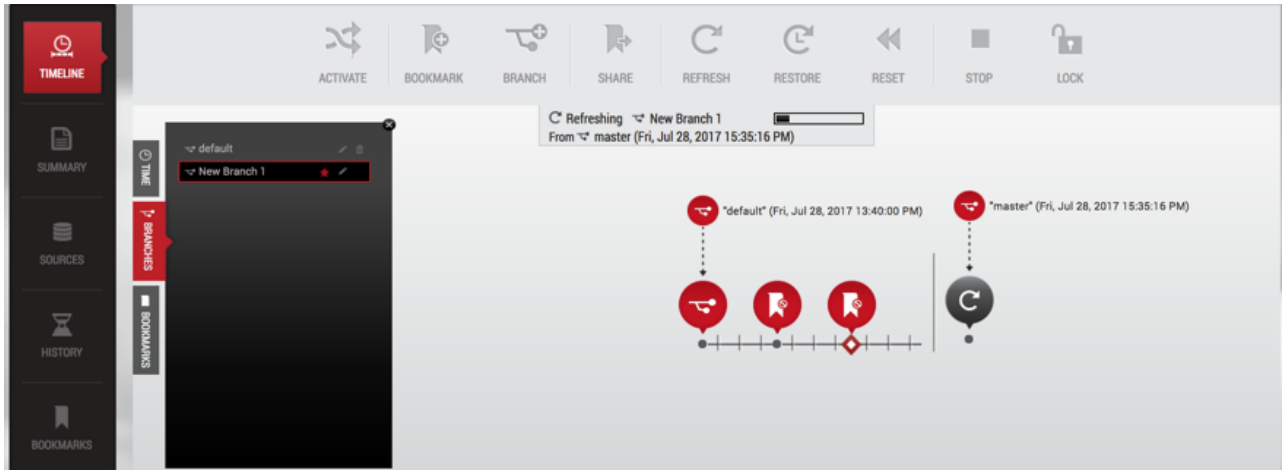


18 Self-Service toolbar with a point in time selected on an active branch timeline.

The **Self-Service Toolbar** is dynamic and will change based on tasks a user performs in Self-Service. These workflows will influence how and when self-service actions become available on the Self-Service Toolbar.

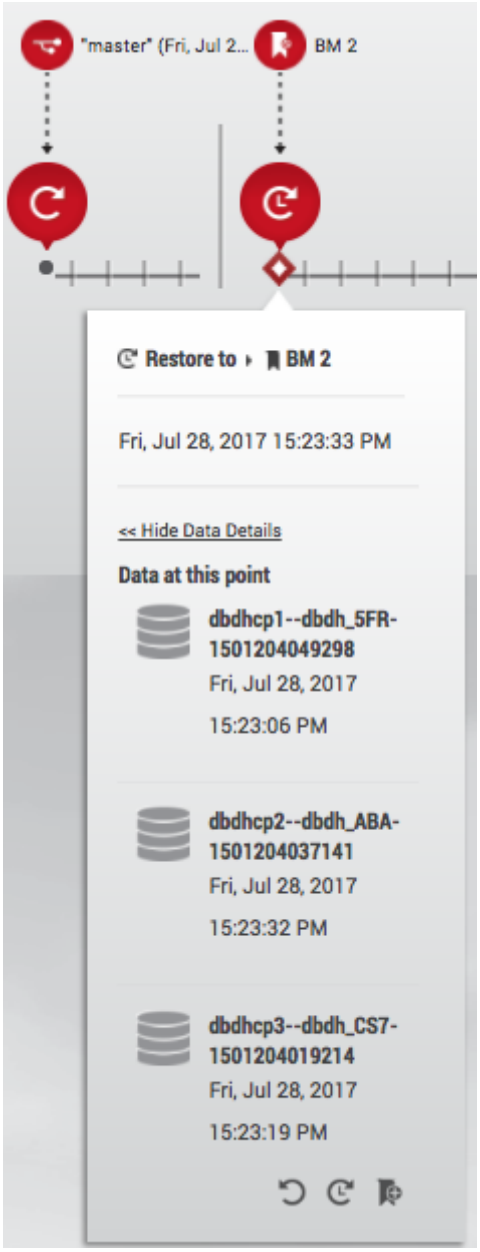
12.2.4.3.1 Branch timeline segments

A branch timeline with segments is a visual representation of actions taken on a branch timeline over a time span. The timeline segments represents data in time that is no longer contiguous once a user clicks **Create Branch**, **Refresh**, **Reset**, or **Restore** on the **Self-Service Toolbar**. A vertical bar between each of the segments appears to remind a Delphix Self-Service user that the data in one timeline segment is a completely new data start. In other words, while the data within one segment is logically contiguous, the data is never contiguous across segments. For example, the following image shows a timeline with multiple segments.



19 Segmented branch timeline.

As mentioned above, the branch timeline becomes segmented after you have performed a specific action or task, such as **Refresh**. Based on the action, two red bubbles will appear in the time segment. The top bubble indicates where the data used for this action came from – for example, the data template, a different branch, or a shared bookmark. The second red bubble appears on the timeline as the actual data stream in a point in time from the parent data. It appears because of actions such as **Refresh**, **Reset**, **Restore**, **Create Branch** and **Bookmark**. Clicking the second bubble will show you specific details of the action, such as the specifics of the action including its name, the time the action occurred, and the data sources used at a point in time. This is illustrated below.



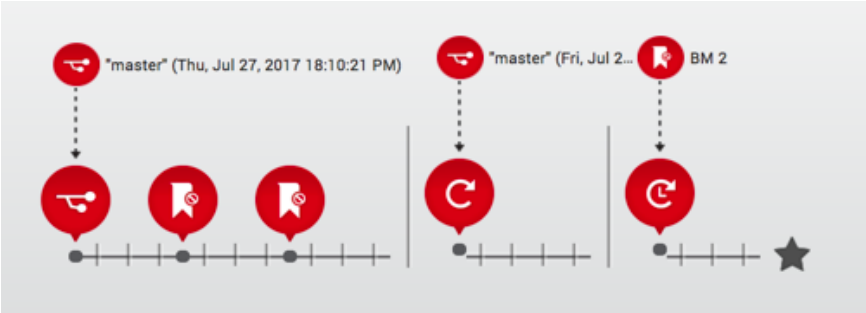
20 Parent data sources and child data sources.

12.2.4.3.2 Working with multiple branches and timelines

As you work in your data container, you can switch between branches to work on resolving a bug or to test a new application feature.

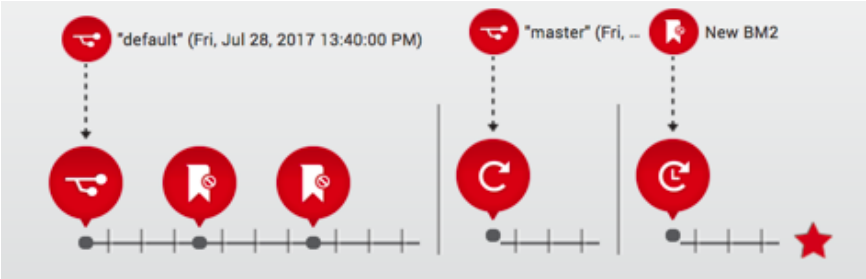
For example, consider what occurs on two different branches in a container:

- Branch one



21 Branch one timeline.

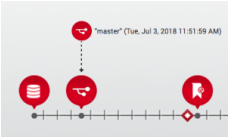
- Branch two



22 Branch two timeline.

A user may have actually worked with these branches in the following order over time:

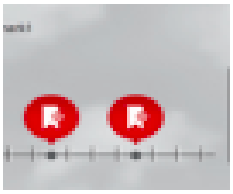
- Branch 1: Create a branch and use



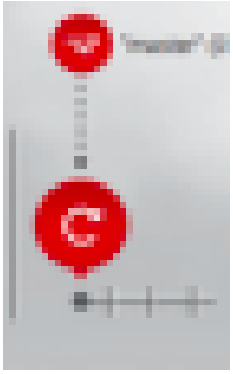
- Branch 2: Create another branch and use

- Branch 1: Activate branch, Restore the data source and use

- Branch 2: Activate branch and create bookmarks

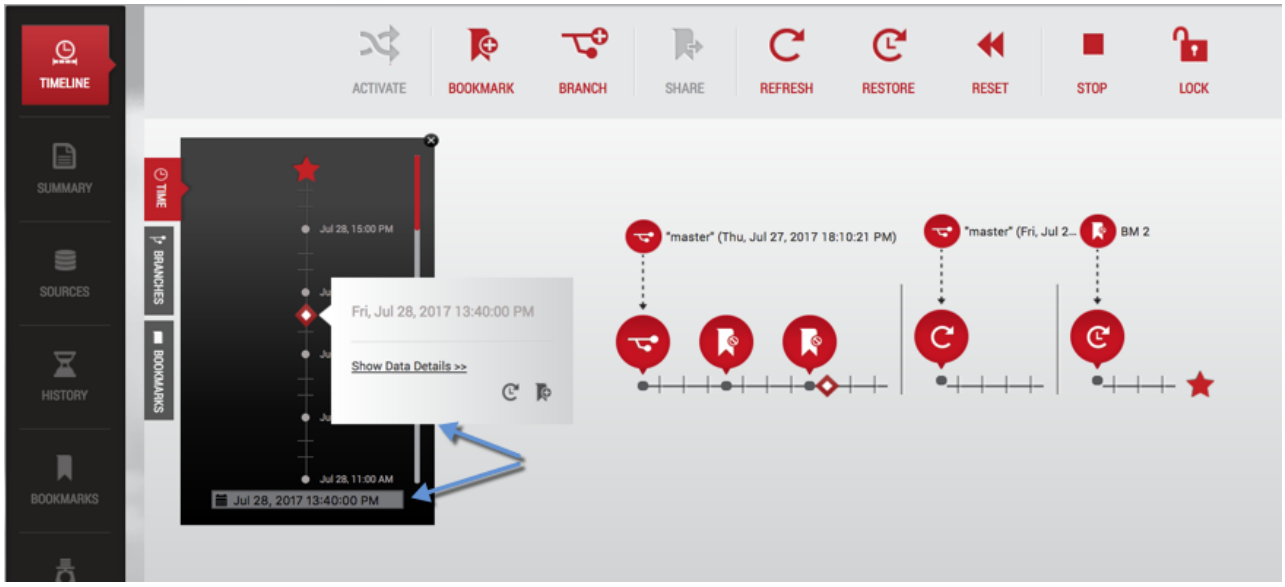


- Branch 2: Refresh the data source from a particular point in time



- Branch 2: Reset a branch to the last action (e.g., refresh) on the timeline, and use

In the above illustrations, an individual branch's timeline shows all actions performed on the branch while the branch was active. The active branch timeline can be interrupted and deactivated when you choose to perform actions such as switching to another branch, **Create Branch**, **Activate**, or **Stop** a data container. Additionally, you will only be able to view actions on a single branch at a time. A better way to manage multiple branches is to go to the **Time** tab in the **Data Container View Panel**. The **Time** tab allows you to access the **container timeline**, which becomes useful as you toggle back and forth between branches to complete different tasks. The **container timeline** allows you to view all the continuous data points of time, with all actions taken on all branches in a single data container.



23 Container timeline.

> Merging keystores requires several manual steps. Refer to Migrating a TDE-enabled vPDB[759] for more information.

12.2.4.4 Understanding how to preserve data in a point in time

The following illustration shows that on 8/27/14, at 9:33:09am, data was reset to the parent data branch (master) at 9:28:48am, capturing data points from 9:33:06am.

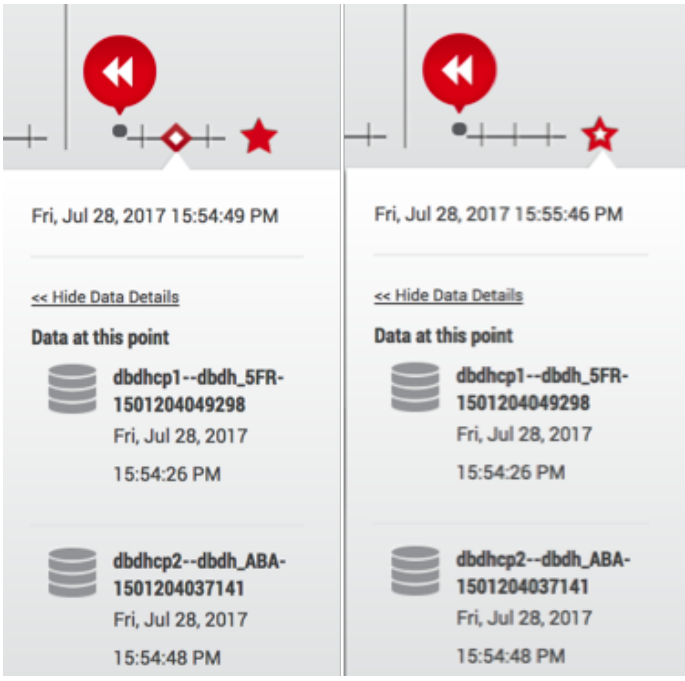
759 http://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/migrating-a-tde-enabled-vpdb

24 Preserve data in a point in time.

The black arrows above point to a tick, (representing a point in time) clicked on the branch timeline. This represents the time the **Reset** action was performed on the data container. The red arrows point to when time was captured in a data source using the **Reset** action on the branch timeline. When clicked, the reset bubble provides more details with a flyout, indicating where the data comes from and the time that the data represents. Additionally, the reset bubble detail flip card provides additional information about each data source. Specifically, the blue arrows point to the time used for each data source at this point in the data container.

> This does not show the time that was used for each source that pulled the data.

The time represented on the branch timeline varies based on many factors. For example, after you select a specific point in time on the branch timeline, the Delphix Engine will map that point to the closest usable point in time for each data source. Based on the properties of the underlying data sources, these times may be different. Not all data sources track changes at the same granularity, as illustrated below.



25 Point in time.

While a branch timeline can follow a continuous-time flow, the data sources being selected for each time segment may not be continuous.

12.2.4.5 Understanding bookmarks

Bookmarks are a way to mark and name a particular moment of data on a timeline that makes it easy to search and find events later. Once a point in time is selected, you can create a bookmark. You can select a specific point in time on the branch timeline, the Delphix Engine will then map that point to the closest usable point in time for each data source. Based on the properties of the underlying data sources, time may not be the same for every data source in the container. Not all data sources track changes at the same granularity. For more information, see Working with Bookmarks in a Data Container[760].

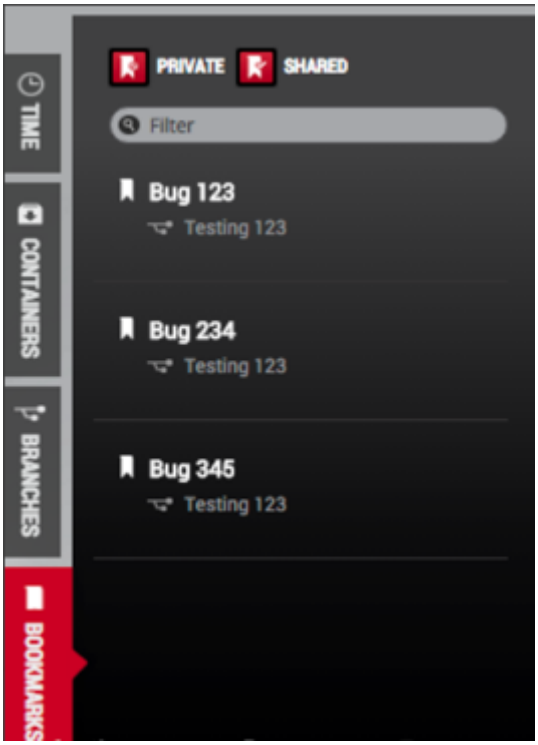
> **Note**
> A bookmark captures the latest point-in-time (not exceeding the bookmark timestamp) that is available while creating the bookmark. When you access the bookmark later, it does not get a closer time even if new logs are available.

Once created, you can easily locate a bookmark through one of the bookmark viewers in the interface.

760 http://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/working-with-bookmarks-in-a-data-container

12.2.4.6 Bookmarks tab in the data container view panel

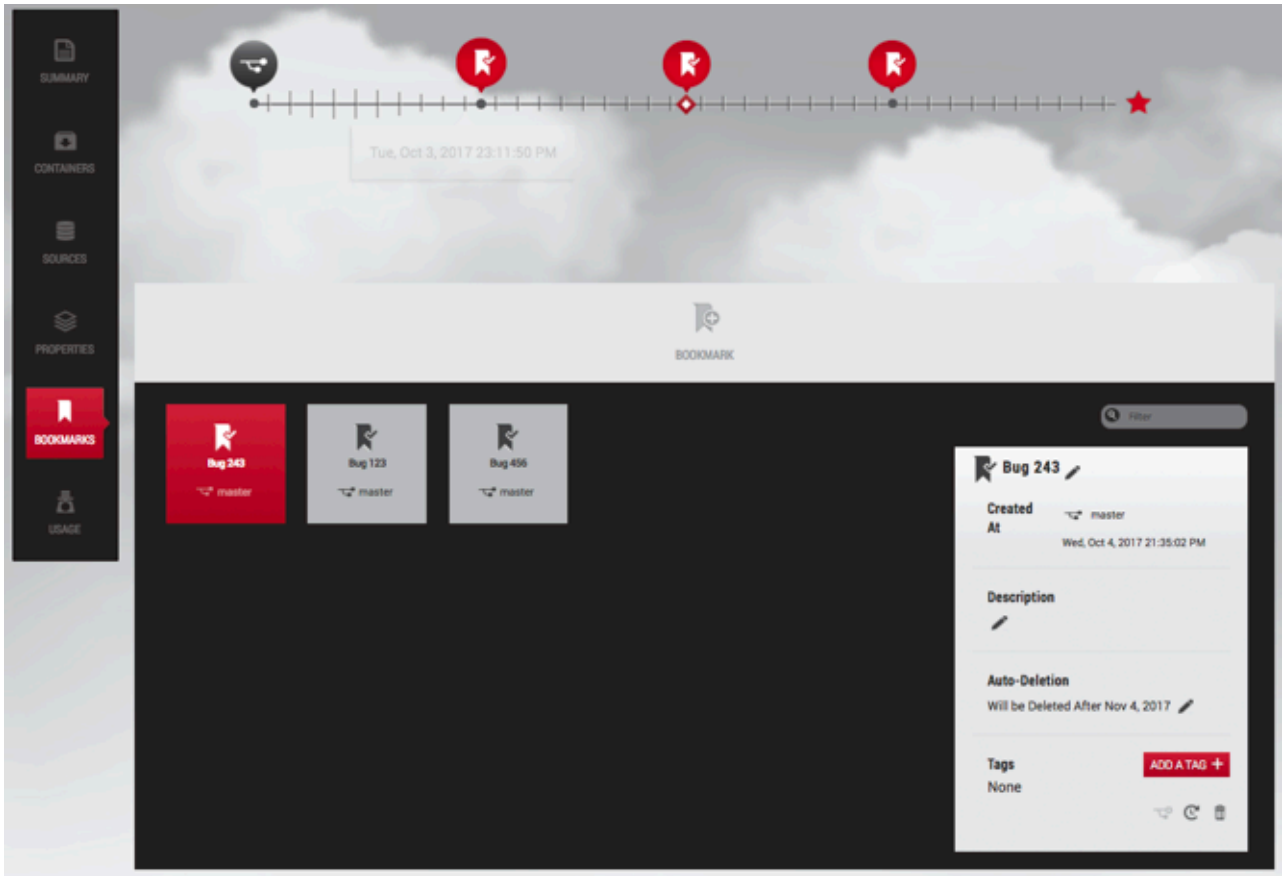
The **Bookmarks** tab is the third tab in the **Data Container View Panel** within the data container workspace of the Delphix Self-Service interface. It allows you to find a bookmark that is within your data container and view the branch where the bookmark has been placed.



26 Bookmark tab.

12.2.4.7 Bookmarks tile in the data container report Panel

The **Bookmarks** tile in the **Data Container Report Panel** allows you to see all bookmarks within your container and all bookmarks that other users have made available to you. Here you can also edit details about bookmarks, create new branches, and restore the active branch to the bookmark's point of data time.



27 Bookmark title.

12.2.4.8 Bookmark sharing permissions

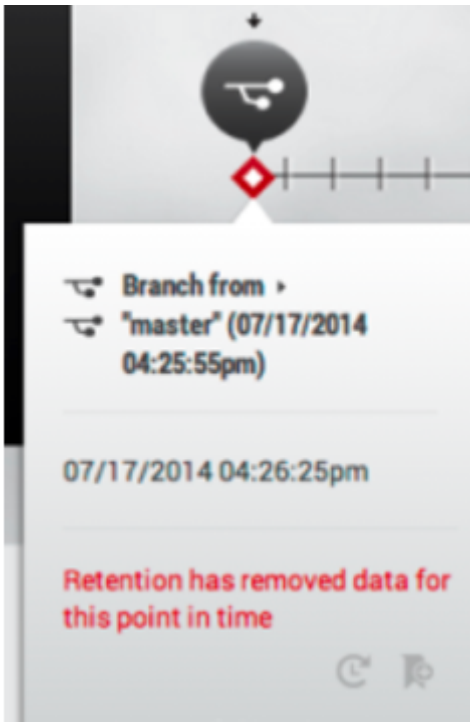
When you first create them, bookmarks are private to your data container, but you can share a bookmark with other data users. Bookmarks that other users have shared with you are called "available" bookmarks.
Your bookmarks will only be shared with data users in data containers created from the same data template. This is because all data containers created from the same data template have a compatible set of data sources.

12.2.4.9 Bookmark appearance

	A bookmark that is private.
	A bookmark you have shared.
	A bookmark that has been shared with you.

12.2.4.10 Data container storage and retention for branches and timelines

Bookmarks mark a moment of data. Delphix Self-Service will never automatically delete the data marked by a bookmark. However, Delphix Self-Service will automatically delete a bookmark with an expiration date set after it has expired. For more information on setting or removing an expiration date, see Data Container Activities[761]. Delphix Self-Service may delete data from any time in the past on your branches, depending on the retention policies configured by your administrator. If you select a moment of data that has been deleted, the flyout will indicate that retention has removed data for this point in time.



28 Data container retention.

12.2.5 Data container activities

12.2.5.1 Getting started

Data containers can be shared between multiple users. In this situation, users should coordinate with their co-owners when performing data operations that could disrupt other users' workflow such as stopping or refreshing the data container.

12.2.5.2 Activity One: how to start and stop a data container

Starting a data container does the following:

761 http://delphixdocs.atlassian.net/continuous-data-11-0-0-0/docs/data-container-activities

- Starts the data sources
  - This means that each data source listed in the **Source Details** section of the **Data Container** page will start using CPU and network resources on the target system it is running on
- Makes the data in the active branch available
  - Once the container has been started, the data represented by the active branch is available

Stopping a Data Container does the following:

- Shuts down the data sources
  - This means each data source listed in the **Source Details** section of the **Data Container** page will stop using CPU and network resources on the target system.

To start a data container, click **Start** on the **Self-Service Toolbar**.

To stop a data container, click **Stop** on the **Self-Service Toolbar**.

12.2.5.3 Working with a branch, a branch timeline, and the self-service toolbar

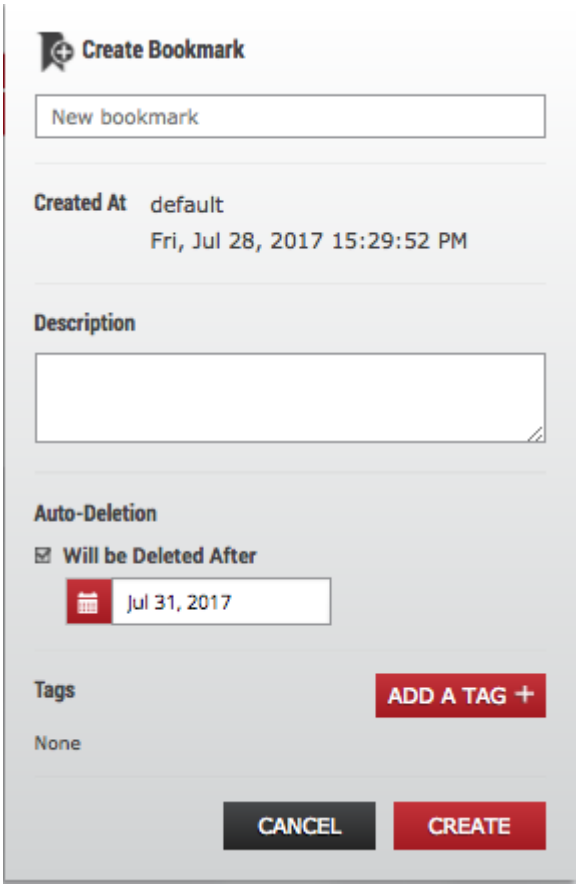
12.2.5.4 Activity Two: using reset from a bookmark to facilitate destructive testing

Reset is a data user workflow that is optimized to enable destructive testing. **Reset** automatically restores the data to the last operation conducted in the data container, which can include creating a bookmark, resetting, or restoring data. As an example, you can do a refresh and then get your data into a state required for testing. Once you are satisfied with the state of your data, you can create a bookmark, which will preserve the data at this point in time.

Afterward, you can then run destructive tests on the data. When you are done, you can click the **Reset** icon, which will automatically restore the state of the container to the last operation – in this case, the bookmark. This workflow ensures that each test has a clean copy of the data and is not impacted by the results of other tests. You only need to create a bookmark and click **Reset** on the **Self-Service Toolbar**.

12.2.5.4.1 Create a bookmark

1. Select a **Data Point** on a branch's timeline.
2. On the **Self-Service Toolbar**, click the **Bookmark** icon.
3. In the **Bookmark Window**, enter a new **name**.



Name a Created Bookmark

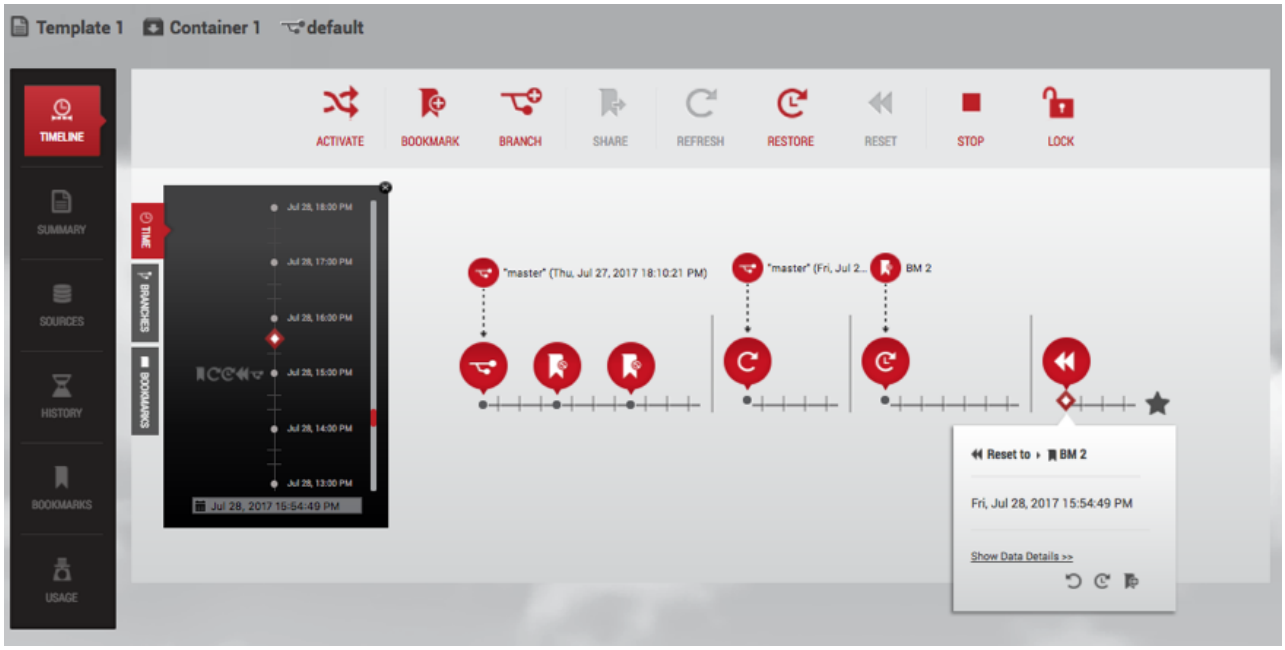
4. Optionally, fill in a **description.**
5. Optionally, set an expiration date. The bookmark will be automatically deleted at the end of this day.
6. Optionally, add one or more **tags**.
   You can use these to help filter a set of bookmarks.
7. Click **Create**.

After the bookmark has been created, you will see the **bookmark** icon appear on the timeline. When you click the **Reset** button, all data will be reset to that point in time.

12.2.5.4.2 Reset to data from a bookmark

Click the **Reset** icon.

This action reflects the moment of data marked by the closest operation bubble (**Refresh**, **Restore**, **Reset**, or **Bookmark**) into a new timeline segment on the active branch. It also copies the moment of data into the data sources.

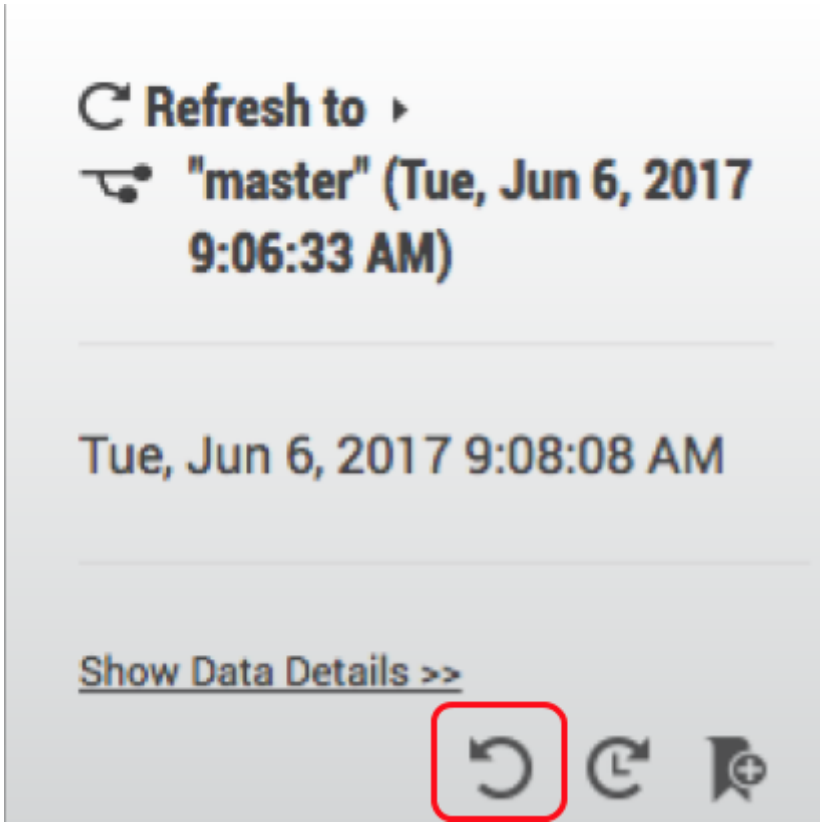


Update Data with Reset

12.2.5.4.3 Undo data

Click the **Undo Data** icon for an operation.

Undo the given operation. This is only valid for RESET, RESTORE, UNDO, and REFRESH operations.



12.2.5.5 Activity Three: using refresh to get the latest data from a data template

1. Start a new timeline segment with the most recent point of data from the data container's data template.
2. Click the **Refresh** icon.

Refresh creates a new timeline segment on the active branch. This refreshes each source in the data container to the latest data in the corresponding source of the data template.

Update Data with Refresh

12.2.5.6 Activity Four: using restore to return data back to a point in time

This starts a new timeline segment on the active branch with the selected point of data.

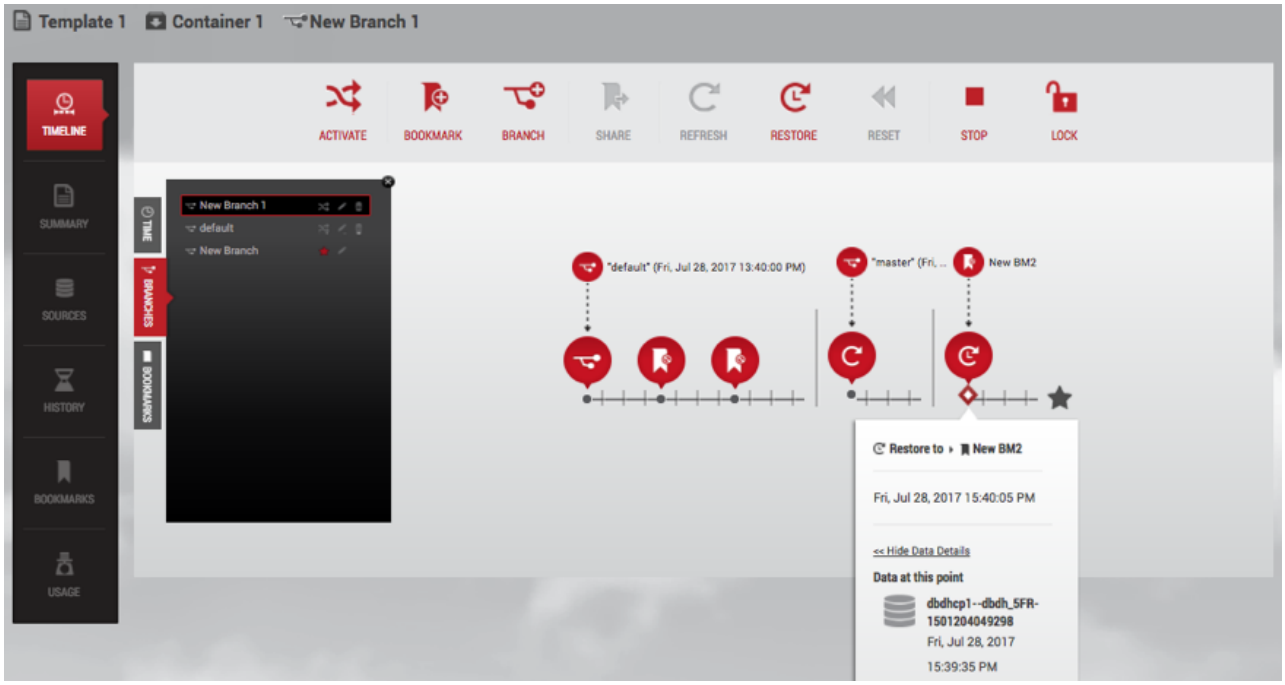
1. Select one of the following:
   a. A **point of data** on a timeline.
   b. A **bookmark** on a timeline.
   c. A **bookmark** under the **Bookmarks** tile in the **Data Container Report Panel**.
2. Click the **Restore** icon.

If you restore data back to a point in time on the data template master timeline, you will be asked which data container to restore into. It will then:

- Reflect the selected point of data into a new timeline segment on the active branch
- Copy the moment of data into the data sources

If the timeline segment on a branch timeline was created by a **Restore** operation, then the segment starts with the moment of data from the branch that was selected when the **Restore** operation was done. This is illustrated below.

> **Parent Branch**
> The parent branch for this segment can be the same branch of which this segment is a part. It is possible to restore the active branch from a point in time on the same branch.



Update Data with Restore

> **Source Branch**
> The source branch for this segment can be the same branch of which this segment is a part. It is possible to restore the active branch from a point in time on the same branch.

12.2.5.7 Activity Five: restoring to a point on the parent template

Data templates serve as the parent for a set of data containers, and as a data user, you have the flexibility to restore your container to any point on the template.

1. Choose the container tab and select the template by clicking on the template name located at the top of the screen.
2. Select one of the following:
   - A point of data on the timeline
   - A bookmark on the timeline
   - A bookmark under the Bookmarks tile in the Data Container Report Panel

3. Click the restore icon
4. A dialog will pop up. Use it to select the container you'd like to restore.

12.2.5.8 force option

forceOption is an API/CLI-only feature. Generally, if a source database is corrupted or otherwise prevents taking VDB snapshots, the Refresh, Reset, and Restore actions cannot be completed. With **forceOption**, you can bypass taking a pre-operation screenshot and proceed with the desired action.

Because **forceOption** does not take a snapshot of the VDB before refreshing/resetting/restoring, you cannot undo the operation afterward.

From the CLI:

1. Go to the Delphix Self-Service container endpoint.
2. Select the container.
3. Attempt the execute the operation.
4. Set **forceOption** parameter to **true**.
5. Commit the change.

The operation will now perform without taking a snapshot first.

12.2.5.9 Activity Six: create a new branch and switch between branches

Developers and QA teams can have multiple branches that can represent data from different points in time or different sources. You have many options for how you create a new branch. These include:

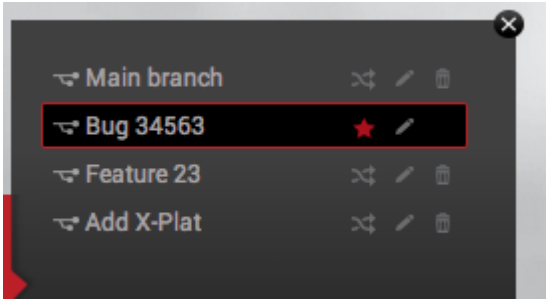
- A **point of data** time on a data timeline within the Delphix Self-Service data container, or
- A **bookmark** bubble on the timeline, or
- A **bookmark** in the **Bookmarks** tile in the **Data Container Report Panel**

1. Click the **Branch**

   icon to create a new branch.
2. Enter a **name** for the new branch.
3. Click **OK**.
4. On the **Self-Service Toolbar**, click the **Activate** icon.

If the inactive branch is not showing in the data container workspace:

1. Find the **branch** in the **Branch** tab.



Selection of Branches in Branch Tab

2. Click the **Activate**

icon.

3. After a moment, the branch will become active.

12.2.5.9.1 Active branch

Within a single data container, only one branch is active at any given time. The data located at the red star of the active branch's timeline is the newest copy of the data from the data container's data sources.
The active branch is distinguished by a red star, which appears at the far right of the timeline, alongside its name in the **Branch Name** area, and in the **Branch** tab.

Active branch	Inactive branch

12.2.5.10 Activity Seven: rename and/or delete a branch

12.2.5.10.1 Rename the Default Branch

1. Select the **Default Branch** in the **Branch** tab.
2. Click the **Pencil** icon to the right of the name.
3. Enter the **new name**.
4. Click the **Checkmark** icon.

12.2.5.10.2 Delete a created branch

1. Select the **branch** in the **Branch** tab.
2. Click the **Delete** icon to the right of the name.

3. Click **Delete** in the confirmation window that appears.

12.2.5.11 Activity Eight: restoring a data container to a consistent state with the recovery operation

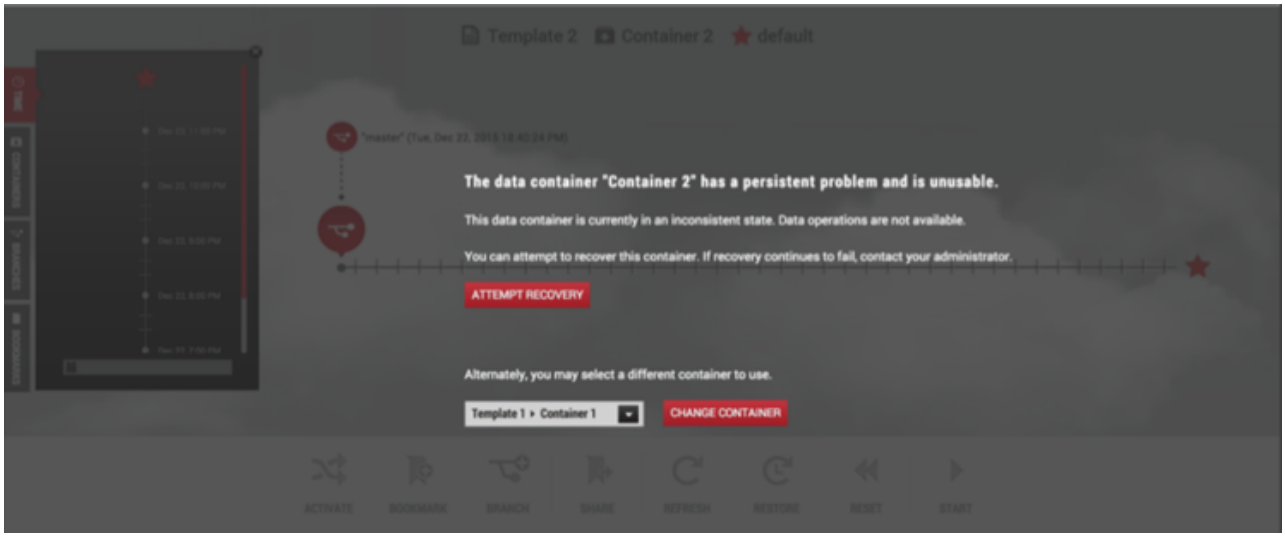
12.2.5.11.1 Data containers consistency

Delphix Self-Service allows you to group multiple datasets in the same data container. This makes it easy for you to access entire applications such as PeopleSoft, including binaries and code. If a data container represents an application, then there are likely to be dependencies between the application's datasets. For example, the vFiles data source containing the code will depend on a specific version of the database's schema. Therefore, it is important that all dataset sources are drawn from the same point in time. If they are, the data container is in a "consistent" state; if they are out of sync, or "inconsistent," errors will occur. For example, if the vFiles data source containing the code has been updated more recently than the database's schema, the dependency cannot work.

Delphix Self-Service currently has no way to determine whether the application is consistent. However, it attempts to minimize the chance that dataset sources are out of sync whenever it performs a data operation such as refresh, restore, or reset. When performing a data operation, Delphix Self-Service attempts to snapshot all dataset sources from a point in time as close as possible to the desired time. If at least one of the data sources fails to go to the desired point, then Delphix Self-Service considers the data container to be in an inconsistent state. The application as a whole may still be working, but Delphix Self-Service assumes that the failed dataset's data is not the correct version. To return to a consistent state, you must perform a recovery operation on the data container.

12.2.5.11.2 Data container recovery

Prior to performing any data operation, Delphix Self-Service takes snapshots of all datasets. Recovery is the process of rolling back a data container to a snapshot, thereby restoring it to a consistent state. When a failure occurs, you will see the following screen:

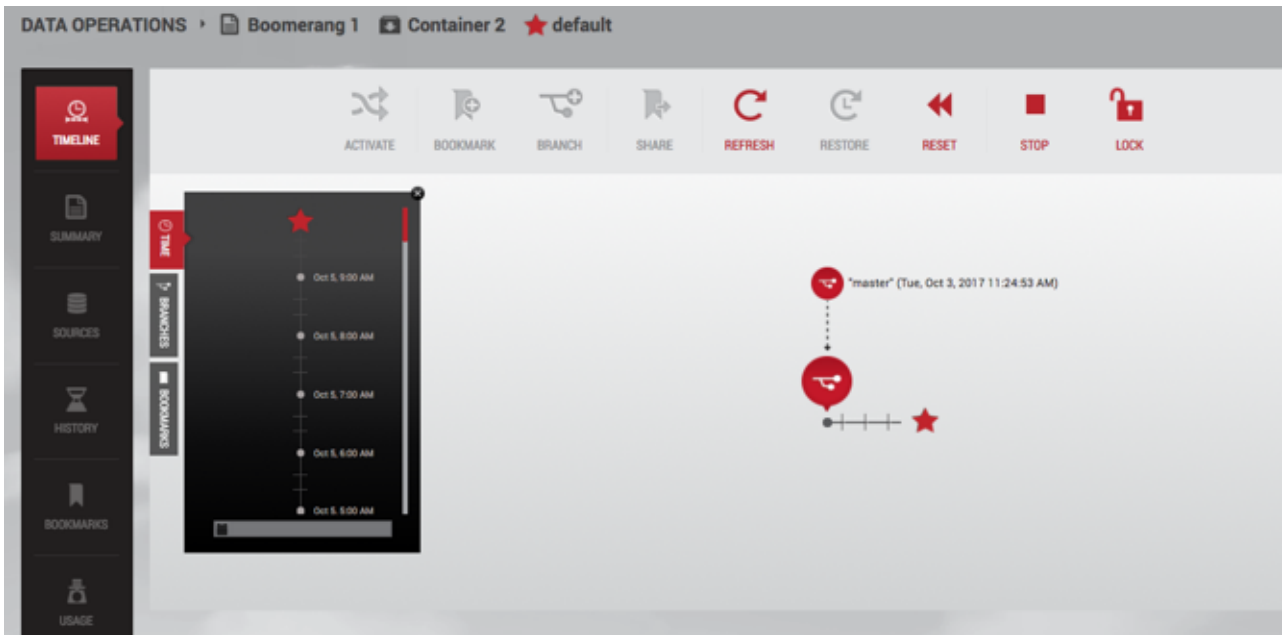


You can either perform recovery or use a different data container. Whether the recovery will fail or succeed depends on exactly why the data operation failed in the first place. If the problem was intermittent, such as a temporary network problem causing SSH failure, then performing recovery should work. If the problem is persistent – for example, the target host is out of space – then intervention is required; recovery will not succeed until you address the underlying root cause of the failure.

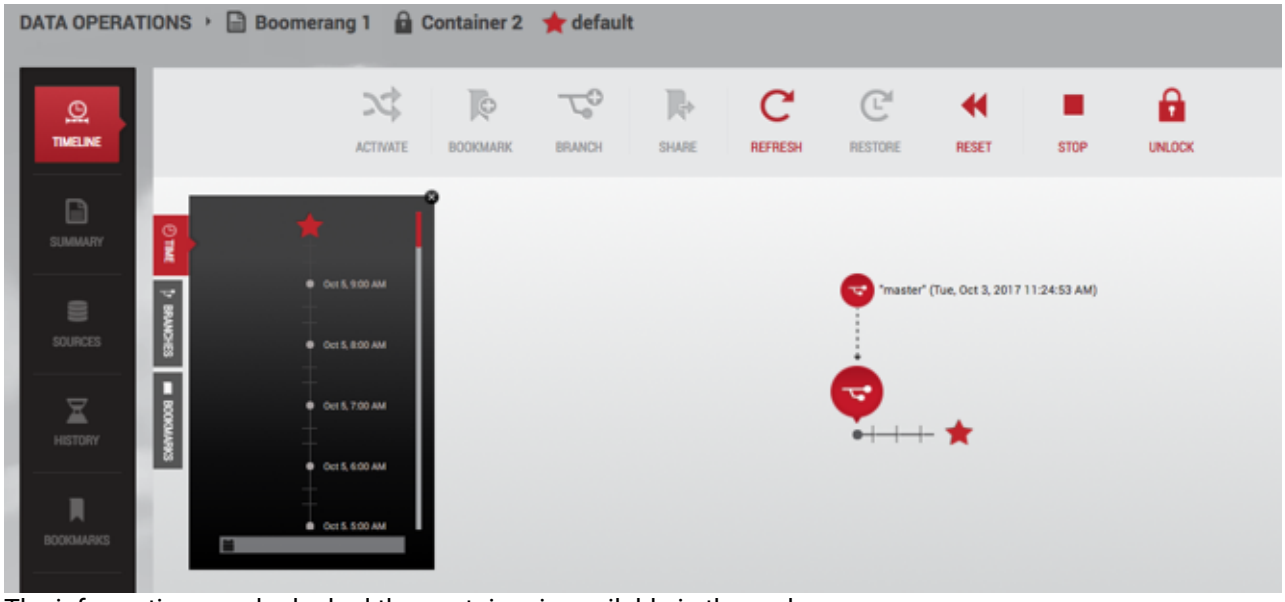
Admins can see the underlying failure in the **Actions** sidebar or the **Job History** dashboard.
The **Actions** sidebar is the preferred place to view the failure; it has a hierarchical display that makes diagnosing the failure more straightforward.

12.2.5.12 Activity Nine: working with container locks

Container owners have the ability to lock/unlock containers. By default, a container is unlocked, which means that all the container's owners can perform operations on it:



You can lock a container by clicking the **Lock** button in order to prevent other users from operating on it.



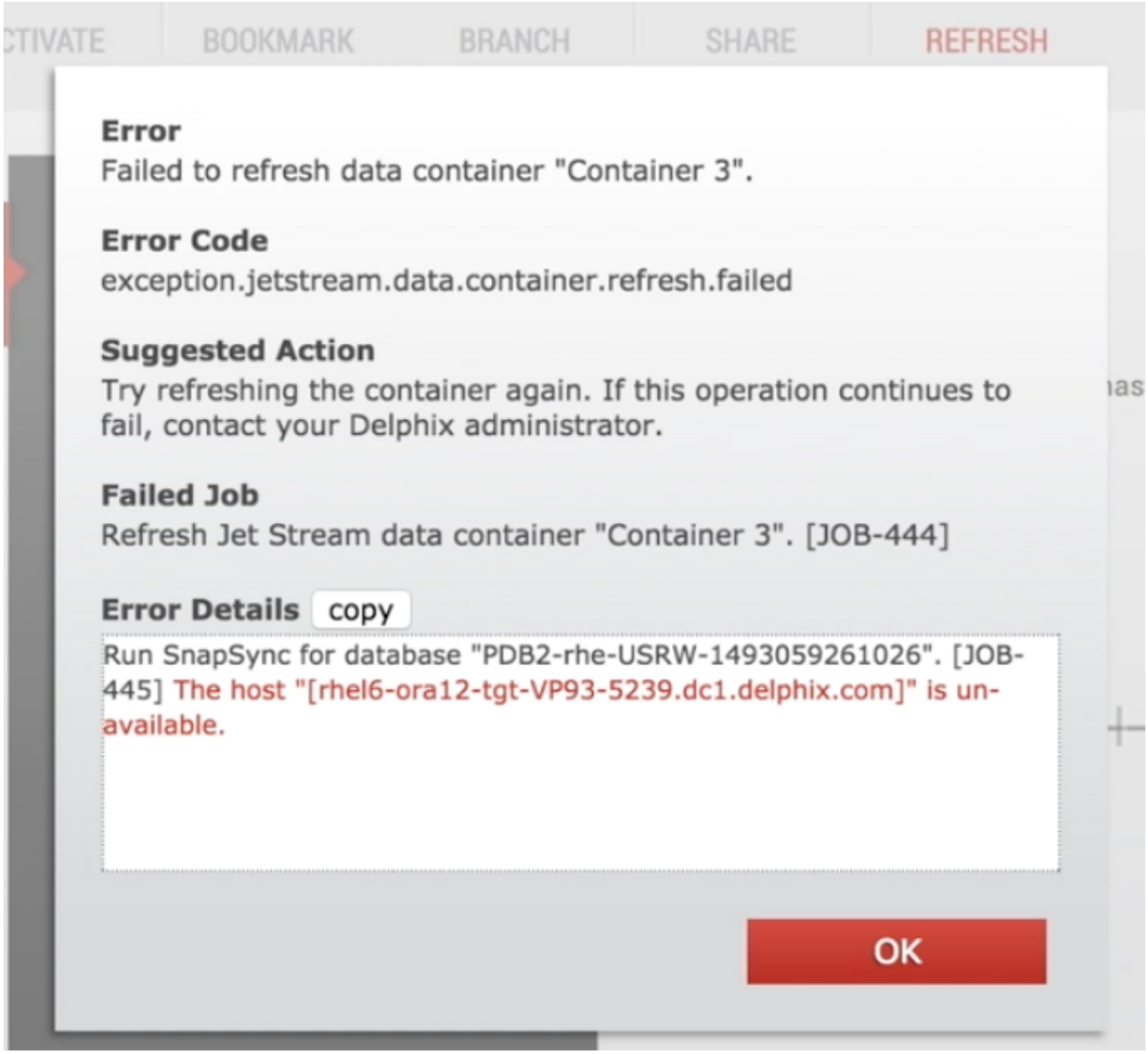
The information on who locked the container is available in three places:

- In the history below the timeline
- In the tooltip on the locked container's name in the container selector
- In the tooltip on the locked container's name in the container label above the timeline

12.2.5.13 Errors

If an action such as refresh fails, a dialog box will give you further information. The **Error Details** field will

The **copy** button enables you to copy and paste error details, which you can then send to your Delphix administrator for further assistance.



Error dialog

12.2.6 Containers with multiple owners

Delphix Self-Service administrators can designate multiple users as owners of a single data container. These users all share access to the same data container which means actions taken by one user will impact all users on the same data container. For example, if User A activates Branch X, User B will also see Branch X as the active branch. This ability to for one user's actions to impact another user on the same containers creates new concerns for users sharing the same container. As a result, more processes should be put into place in order to coordinate usage between users. Each team is different, but strategies include:

- designating a person to perform certain data operations
- saving your work with a bookmark or creating / working on a personal branch
- being aware of who is using your data container / data before performing operations

- locking a container to prevent others from performing any operations on it

12.2.6.1 How many owners should a container ideally be shared between?

There is no technical limit built into the software, but it is best if a team of 5-10 users shares a single data container. In most cases, having fewer owners minimizes overhead and conflicting usage. One owner per container provides maximum productivity and minimal overhead, so this feature should only be used if your infrastructure or processes require that multiple users share a container. Additionally, Jet Stream Only users currently cannot see other users with whom they share the container.

12.2.6.2 How should users handle potentially disruptive operations?

If one user performs an operation on a data container, it will affect the other owners of that container. Additionally, each user has permission to perform the same operations on the data container; currently there are no fine-granularity permissions that limit the operations a user can perform. All operations are potentially disruptive, but the level of disruption varies by operation. If any of the following operations are performed at the same time, the second operation will fail due to a conflict when processing the job.

12.2.6.3 Conflicting operations

- Refresh
- Restore
- Reset
- Enable/Disable
- Create Branch
- Activate Branch
- Delete Branch
- Create Bookmark
- Delete Bookmark

If User A performs a destructive operation while User B is "using" the data container, the operation will destroy User B’s current state. Currently, the interface does not provide insight into whether the data container is in use by another user.

12.2.6.4 Destructive operations

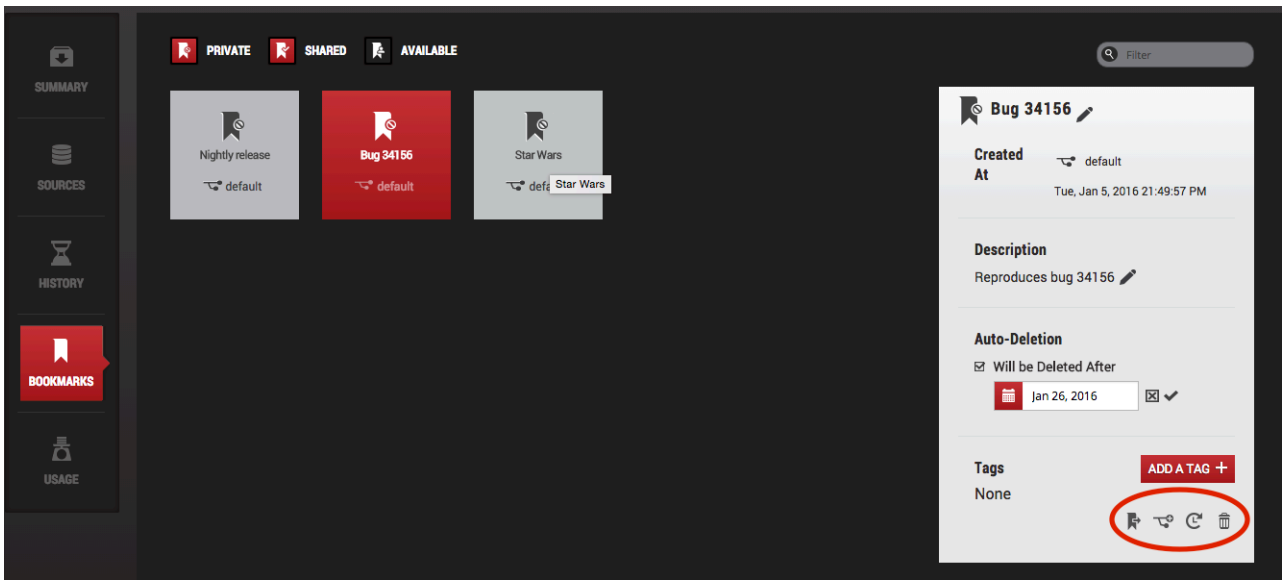
- Refresh
- Restore
- Reset
- Enable/Disable
- Create Branch
- Activate Branch

12.2.6.5 Deleting objects

All owners can delete any bookmarks or branches in the container, regardless of who created them.

12.2.7 Working with bookmarks in a data container

Working with bookmarks is an easy way to share data with other users of any container created from the same template. By sharing with others, you can integrate testing, development, and QA needs. For example, in the past, if you found a bug you would wait until it was fixed. But with bookmarks, you do not have to stop your work while someone tries to fix the problem. Sharing a bookmark allows users to work with data as they see fit. Bookmarks mark a moment of data. Delphix Self-Service will never automatically delete the data marked by that bookmark. However, Delphix Self-Service will automatically delete a bookmark with an expiration date set at the end of that day.



Bookmarks Management in the Data Report View Panel

12.2.7.1 Activity Nine: Share a bookmark with other Delphix self-service users

12.2.7.1.1 Share a bookmark

1. Select a **bookmark** by clicking one of the following:
   - The bookmark's **bubble** on the **branch timeline**.
   - The **Bookmarks** tab in the data container workspace.
   - The **Bookmarks** tile in the **Data Container Report Panel**.
2. Click the **Share**

icon.

You cannot share a bookmark that you or another user have already shared.

12.2.7.1.2 Unshare a bookmark

1. Select a **bookmark** by clicking one of the following:
   - The bookmark's **bubble** on the **branch timeline**.
   - The **Bookmarks** tab in the data container workspace.
   - The **Bookmarks** tile in the **Data Container Report Panel**.
2. Click the **Unshare**

   icon.

You cannot unshare a bookmark that is already private or a bookmark which someone else has shared.

12.2.7.1.3 Delete a bookmark

1. Select a **bookmark** by clicking one of the following:
   - The bookmark's **bubble** on the **branch timeline**.
   - The **Bookmarks** tile in the **Data Container Report Panel**.
2. Click the **Delete**

   icon.

12.2.7.2 Activity Ten: editing bookmarks

12.2.7.2.1 Rename a Bookmark

1. In the **Data Container Report Panel**, click the **Bookmarks** tile. A selection of bookmarks will appear based on whether you have chosen to view private, shared, and/or available bookmarks.
2. In the **detail bookmarks** window, click the **Edit** icon to the right of its name.
3. Enter the **new name** in the edit field.
4. Click the **checkmark** to the right of the field to accept and save the new name.

12.2.7.2.2 Edit the description of a bookmark

1. Select a **bookmark** by clicking the **Bookmarks** tile in the **Data Container Report Panel**.
2. Click the **Edit** icon to the right of its name.

12.2.7.2.3 Remove the expiration date of a bookmark

1. Select a **bookmark** by clicking the **Bookmarks** tile in the **Data Container Report Panel**.
2. To the right of the bookmark's name, click the **Edit** icon.
3. Uncheck the **Will be deleted after** checkbox.
4. Click the **checkmark** to the right of the date selector.

12.2.7.2.4 Set or update the expiration date of a Bookmark

1. In the **Data Container Report Panel**, click the **Bookmarks** tile.
2. To the right of the bookmark's name, click the **Edit** icon.
3. Check the **Will be deleted after** checkbox.
4. Use the date selector to pick a new date.
5. Click the **checkmark** to the right of the date selector.

12.2.7.3 Activity Eleven: filter and view bookmarks

12.2.7.3.1 View Only Your Created Bookmarks

In the **Bookmarks** tile in the **Data Container Report Panel**, bookmarks that belong to you are shown. To see only your own bookmarks:

1. In the **Data Container Report Panel**, click the **Bookmarks** tile.
2. De-select **Available**.

12.2.7.3.2 View Bookmarks You Have Shared with Others

1. In the **Data Container Report Panel**, click the **Bookmarks** tile.
2. De-select **Private**.
3. De-select **Available**.

Only your shared bookmarks will be shown.

12.2.7.3.3 View Bookmarks That Others Have Shared with You

1. In the **Data Container Report Panel**, click the **Bookmarks** tile.
2. De-select **Private**.
3. De-select **Shared**.
4. Select **Available**.

These are the bookmarks that have been shared with you.

12.2.7.3.4 Adding Tags To Your Bookmark

1. In the **Data Container Report Panel**, click the **Bookmarks** tile.
2. Select the **bookmark** to which you want to add tags.
3. Click **Add a Tag**.
4. Enter the **tag name**.
5. Click the **Accept** icon.

Your tags will be shown at the bottom of the **Bookmarks** tile in the **Data Container Report Panel**.

You can only add tags to bookmarks that you have created.

12.2.7.3.5 Finding Bookmarks

In either the **Bookmarks** tab in the data container workspace or the **Bookmarks** tile in the **Data Container Report Panel**:

- Type into the **Filter** field.

This will only show bookmarks that have names or tags that match the text you have entered.

12.2.8 Understanding Delphix self-service usage

12.2.8.1 Usage management Dashboard overview

Data templates are comprised of dSources, virtual databases (VDBs), and vFiles. These data sources are controlled by the standard policies configured in the **Management** application of the Delphix Engine. As with existing containers, space will be reclaimed by the retention policy over time. As retention cleans up historical data, users will no longer be able to use those points in time to restore or branch. In Delphix Self-Service, an admin can create a bookmark on the data template timeline, which will prevent retention from cleaning up the data that bookmark references.

Data containers are comprised of VDBs provisioned from the sources defined in the data template. Similar to VDBs in the **Management** application, data containers' VDBs will share blocks with the source from which they are provisioned. This prevents the referenced data on the source from being cleaned up by retention. Retention for these VDBs is controlled by the standard Delphix Engine retention policies. As on templates,

bookmarks in data containers will prevent storage from being reclaimed by retention. In addition, Self-Service will ensure that the latest data on each branch is never removed.

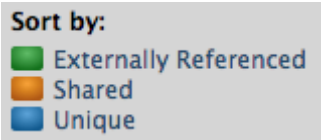
The **Usage** pages of the data templates and data containers provide information that can help you understand how storage is being used, how to reclaim space, and how much space you are able to reclaim.

Usage Overview is a top-level page, along with the **Data Operations** and **Overview** pages. It contains the space usage breakdowns by data templates and users.

12.2.8.2 Container usage overview

The **Usage Details** page, shows the space used by data containers provisioned from the template and the bookmarks created on the template.

The stacked bar graph shows information about the top 10 space users. You can re-sort the graph based on the fields in the **Sort by** legend on the top right-hand corner of the screen as seen in the image above. For example, if you want to know which data containers are sharing the most data with others, you can un-select **Shared (others data)** and **Unique** by clicking them in the legend.



When the legend items are not selected, their corresponding colored boxes turn gray and the data is removed from the chart. The data and name will reappear when you re-select by click on the preferred grayed-out category.

The field categories display the following information:

- **Unique** – The amount of space that will be freed if you delete this data container. This assumes that also delete underlying data sources.
- **Shared (others data)** – The amount of space that cannot be freed on the parent data template (or sibling data containers) because it is also being referenced by this data container due to Restore or Create Branch operations. The snapshots on the template or sibling container are what use up space.
- **Shared (self data)** – The amount of space that cannot be freed on this data container because it is also being referenced by sibling data containers due to Restore or Create Branch operations, via shared bookmarks
- **Unvirtualized** – The amount of space that would be used by the data in this container without Delphix virtualization

12.2.8.3 Bookmarks usage overview

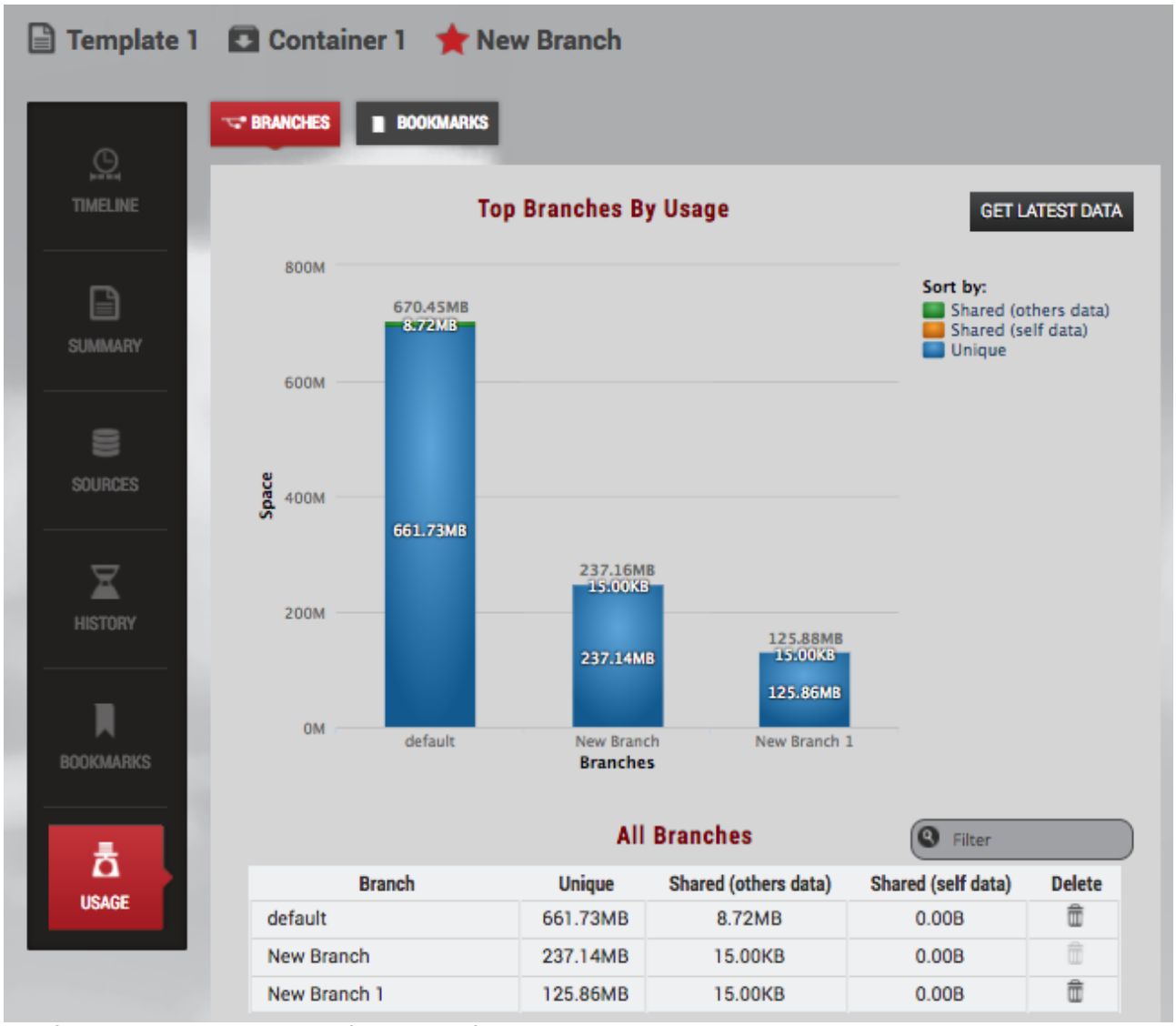
As shown in the image above, the **Container Usage** page provides the usage information about bookmarks created on a template. The primary categories of information include **Unique**, **Shared (others data)**, and **Shared (self data).**

The field categories display the following information:

- **Unique** – The amount of space that will be freed if you delete this bookmark
- **Shared** – The amount of space referenced by this bookmark that cannot be freed by deleting this bookmark because it is also referenced by neighboring bookmarks or branches that have been created or restored from this bookmark
- **Externally Referenced** – The amount of space referenced by this bookmark cannot be freed by deleting it because it is also being referenced outside of Self-Service – for example, by a retention policy.

12.2.8.4 Branches usage overview

As detailed in the image above, the **Container Usage Details** page shows the usage information about the branches and bookmarks created on a container. The primary categories of information include **Unique**, **Shared (others data)**, and **Shared (self data)**.



The field categories display the following information:

- **Unique** – The amount of space that will be freed if you delete this branch
- **Shared (others data)** – The amount of space that cannot be freed on the parent data template or sibling branches because it is also being referenced by this branch due to Restore or Create Branch operations. The snapshots on the template or sibling container are what use up the space.
- **Shared (self data)** – The amount of space that cannot be freed on this branch because it is also being referenced by sibling data containers due to Restore or Create Branch operations, via shared bookmarks.

13 Developer's guide

13.1 Developer's guide

Command line interface guide (see page 2521)

Web services API guide (see page 2752)

13.2 Command line interface guide

This section contains the following topics:

- Command line interface overview (see page 2521)
- Delphix objects (see page 2530)
- Command reference (see page 2535)
- CLI cookbook: common workflows, tasks, and examples (see page 2538)

13.2.1 Command line interface overview

This topic provides an overview of the Delphix Engine command-line interface and links to additional topics.

The Delphix Engine provides a native command-line interface (CLI) accessible over SSH. This CLI provides an interactive layer on top of the public web service APIs, and is intended for users that wish to automate interactions with the Delphix Engine, or simply prefer a text-based interface. All of the functionality available in the CLI is also available through the public stable web service APIs should more full-featured automation be required. For more information on automation using the web service APIs, see the Web service API guide (see page 2752).

The CLI has an internal help system and supports tab-completion to help guide users. Running the `help` command will display a list of valid commands and properties, if applicable. Specifying the command or property as an argument to `help` will display more specific information about that command or property. This guide serves as an overview of CLI operation and examples of some basic tasks, and is not a reference for all CLI commands or properties. As the CLI content is identical to the public web services, complete information about particular commands, properties, or other operations can be found in the API documentation delivered with each server instance, found at:

```
http://<server>/api
```

The API documentation is guaranteed to be consistent with the set of APIs exported by that particular server. All of the APIs used by the GUI will be supported by the CLI. While all the database and environment APIs are available, most of the system-oriented APIs (such as those required to do initial setup) will be made available in a later release.

- Connecting to the CLI (see page 2522)
- CLI contexts (see page 2524)
- Managing objects (see page 2525)

- Managing properties (see page 2526)
- Array properties (see page 2527)
- Untyped object properties (see page 2527)
- CLI automation (see page 2528)

13.2.1.1 Connecting to the CLI

This topic describes how to connect to the Delphix Engine command line interface.

> If the same username exists in both the SYSTEM and DOMAIN namespaces (for example a sysadmin and a domain user), when logging in via ssh it is necessary for the user to explicitly provide the namespace SYSTEM or DOMAIN. For example, a user whose name exists in both namespaces must ssh in as either "someone@SYSTEM" or "someone@DOMAIN". If the user tries to log in as just "someone", the engine will give an error:
>
> Ambiguous username. Add a @DOMAIN or @SYSTEM suffix to the username.

The CLI is available over SSH or the terminal console on any Delphix Engine version 3.0 or later. To connect, use any SSH client appropriate for your workstation environment and connect to the Delphix Engine by IP or hostname on the standard SSH port (22). Enter a username for either a domain or system user followed by the namespace appropriate to that user (either DOMAIN or SYSTEM). For example:

```
ssh admin@DOMAIN@delphix-server.example.com
```

```
ssh sysadmin@SYSTEM@delphix-server.example.com
```

> The default domain user created on Delphix Engines is now **admin** instead of delphix_admin. When engines created before 5.3.1 are upgraded to 5.3.1 or later they will retain their old username 'delphix_admin'. To avoid complications Delphix recommends creating users with an admin role and then Disabling delphix_admin.

At the prompt, enter your user password. Once connected, you will be placed at the CLI prompt:

```
delphix>
```

While both admin and sysadmin produce the same prompt once logged in, be aware that the two users have different menus and different functional areas.

13.2.1.1.1 Sysadmin menu

```
delphix> ls
Children
network
service
storage
system
user
Operations
version
Operations
version
delphix>
```

13.2.1.1.2 Delphix admin menu

```
delphix> ls
Children
alert
audit
authorization
connectivity
database
environment
fault
group
host
job
namespace
network
policy
replication
repository
service
session
snapshot
source
sourceconfig
system
timeflow
user
Operations
version
delphix>
```

Individual commands passed as arguments to the SSH client will be interpreted as if they had been read from the terminal. More complex scripts can be passed as input to the SSH command. When running SSH in non-interactive mode via these mechanisms, the command line prompt will be suppressed, as will terminal font decorations such as underline and bold.

The CLI is also available from the serial terminal console should the network be unavailable. Consult your VM platform documentation for information on how to connect to the terminal console. Once connected, log in using your Delphix user credentials just as you would over SSH.

If the management service is unavailable due to a software bug or other problem, the CLI can still be accessed as a system user provided that user is locally authenticated (not via LDAP) and has logged in at least once before. While in this state, only the `system` commands are available, including `restart`, which will attempt to restart the management service without rebooting the entire server. If this problem persists, please contact Delphix support.

The topic CLI cookbook: configuring Key-Based SSH Authentication for Automation (see page 2547) shows an example of how to connect to the CLI using SSH key exchange instead of the standard password-based authentication.

13.2.1.2 CLI contexts

This topic explains the concept of contexts within the Delphix Engine command-line interface.

The CLI is built on the concept of modal “contexts” that represent an administrative point for interacting with the web service APIs. These contexts can be divided into the following types:

Context	Description
Static children	These contexts exist for the purpose of navigating between points in the hierarchy, but have no properties of their own and do not correspond to any server-side object. The root context is an example of this, as are most of the top-level contexts such as `database` or `group`.
Object	These contexts represent an object on the server, either a specific object (such as databases) or system-wide state (such as SMTP configuration). These contexts have properties that can be retrieved via the get command.
Operation	These contexts represent a request to the server. Commands may or may not require input and may or may not change state on the server, but in all cases require an explicit commit operation to execute the command. When in a command context, the prompt includes a trailing asterisk (*) to indicate that `commit` or `discard` is required before exiting the context.

User can move between contents by typing the name of the context. To move to a previous context, the `up` or `back` commands can be used. In addition, the CLI supports UNIX-like aliases for `cd` and `ls`, allowing navigation similar to a UNIX filesystem. For more information on these commands, see the Command Reference (see page 2535) section.

13.2.1.3 Managing objects

This topic describes the use of objects in the Delphix Engine command-line interface and provides a list of the object management operations.

The Delphix Engine represents state through objects. These objects are typically managed through the following operations, covered in more detail in the Command Reference (see page 2535) topics.

The topic CLI cookbook: Changing the Default Group Name (see page 2561) illustrates the use of object management commands such as `list` and `get`.

Operation	Description
`list`	For a given object type (represented by a static context such as `database`), list the objects on the system, optionally constrained by some set of attributes. Some objects are global to the system and do not support this operation.
`select`	Select a particular object by name to get properties or perform an operation on the object. See the “Delphix Objects” section for more information on object naming.
`get`	Display all or some of the properties of an object after selecting it.
`update`	Enter a command context to change one or more properties of an object after selecting. Not all objects support this operation, and only properties that can be edited are shown when in the update command context.
`create`	Create a new instance of the object type from the root static context. Not all objects can be created in this simplified fashion. Databases, for example, are created through the `link` and `provision` commands.
`delete`	Deletes an object that has been selected. Not all objects can be deleted.

In contexts where there are multiple objects of a given type, the `list` command can be used to display available objects, and the select command can select an object for subsequent operation.

When listing objects, each context has its own set of default columns to display. The display option can be used to control what columns are displayed to the user. This is a comma-separated list of property names as they would be retrieved by the `get` command. It is possible to specify properties that do not exist in order to accommodate lists of objects of varying types and untyped objects.

The topic CLI cookbook: Listing Data Source Sizes (see page 2610) provide an example of using the `list` command.

13.2.1.4 Managing properties

This topic describes the use of properties in relation to objects in the Delphix Engine command-line interface.

Object properties are represented as a hierarchy of typed name/value pairs. The `get` command by itself will display the complete hierarchy for a particular object. This hierarchy is displayed with each nested object indented by an additional level. The set of available properties depends on the command context and may change if the type of an object is changed.

13.2.1.4.1 Property state

Properties are typically set to a specific value, but they can also be `unset`. Unset properties indicate there is no known value, either because it hasn't been provided yet, or it has been explicitly removed. Properties in this state are displayed via the following means:

- `(unset)` – The property is not currently set. It may never have been given a value or it may have been explicitly unset through the `unset` command.
- `(required)` – This has the same underlying semantics as `(unset)`, but indicates that the property **must** be set before the current command can be committed. Failure to do so will result in a validation error at the time the commit operation is attempted. Required properties are displayed in bold.

In addition, all objects have a default state when in command context. A property that has been modified is noted with an asterisk (*), and can be reverted to its default state through the `revert` command.

When updating properties, only those properties are sent to the server. The exception is arrays and untyped objects, covered in Array properties (see page 2527) and Untyped object properties (see page 2527). These objects are always sent in their entirety, so changing any one element will send the entire object.

13.2.1.4.2 Basic properties

Most properties are displayed and input as a string, though the underlying type may be more specific. The following are some of the basic types:

- **String** – An arbitrary string. This may be subject to additional validation (such as an IP address) that is enforced at the time the property is set.
- **Number** – An integer number.
- **Boolean** – Either "true" or "false".
- **Enumeration** – A string that must be chosen from a known set of options.

13.2.1.4.3 Nested properties

Some properties are in fact other objects and are represented as a nested set of properties. These properties can be manipulated in one of two ways: by specifying a dot-delimited name or changing the context via the `edit` command.

A dot (.) in a property name indicates that the portion to the left of the dot is the parent object name, and the portion to the right is a child of that object. For example, `sourcingPolicy.logsyncDisabled` denotes the `logsyncDisabled` property within the `sourcingPolicy` property. These dots can be arbitrarily nested. An alternative syntax of using brackets to enclose property names `(sourcingPolicy[logsyncDisabled])` is also supported for familiarity with other programming languages.

The `edit` command, in contrast, will change the current context such that all properties are relative to the specified object. This can be useful when changing many nested properties at once, or when the complete set of properties can be confusing to manage all at once.

The topic CLI cookbook: Disabling LogSync for a dSource (see page 2604) provides an example of manipulating nested properties.

13.2.1.5 Array properties

This topic describes the use of array properties in the Delphix Engine command-line interface.

Some Delphix objects represent properties as arrays. Arrays are effectively objects whose namespace is a contiguous set of integers. While they behave like objects and their properties can be referenced via the same object property notation, they differ in several key areas.

Arrays can be divided into two types: arrays of primitive types (strings, integers, etc.) and arrays of objects. Arrays of objects can be managed like other objects via nested property names and the `edit` command, but differ in the following respects:

- When an array element is `unset`, it removes the element from the array and shifts all other elements down to preserve the contiguous index space.
- New array elements can only be appended to the end of the array by specifying an index that is one more than the maximum index of the array.
- When displaying a property that is an array, if the length is greater than 3, then it is displayed only as "[…]". The complete contents of the array can be displayed by getting or editing that particular property.

Arrays of primitive types can be managed as arrays of objects, but also support an inline notation using comma-separated notation. This allows single-element arrays to be set as standard property, and for arrays of strings to be set on a single line instead of having to edit each element.

Regardless of the element type, arrays are sent as complete objects when updated. When any array element is changed and subsequently committed, the complete array is sent to the server. When a single array element is reverted, the entire contents of the array are reverted.

The topic CLI cookbook: Setting Multiple Addresses for a Target Host (see page 2592) provides an example of working with a property that is an array of strings.

13.2.1.6 Untyped object properties

This topic describes the use of the `type` field in the Delphix Engine command-line interface object model, and the use of untyped objects.,

Most Delphix objects are typed, meaning they have a `type` field that controls what properties are available and their types. Object types and their associated hierarchy are described in more detail in the topic Object Type Hierarchy (see page 2531) topic. In contrast, some properties are “untyped” objects, which means that there are no constraints on the property namespace, and all properties are plain strings. These objects are used for database configuration templates and other scenarios where the property namespace is unbounded or under the control of the user.

Untyped objects are always sent in their entirety when making updates. This means that when anyone value is changed and then committed, all values are sent. In addition, when reverting a single value within an untyped object, the entire parent object is reverted to its default state.

13.2.1.7 CLI automation

This topic describes using automation with both the Delphix Engine command-line interface (CLI) and the web service API.

All functionality is available in both because the CLI is built upon the web services API. The CLI enables you to create scripts for simple automation, and it is a useful aid in the development of more complex code that uses the web service API.

13.2.1.7.1 Using the CLI for simple scripts

For simple automation, you can build routines that make CLI calls through SSH.

This snippet lists all environment names. It leverages the SSH key exchange explained in CLI cookbook: Configuring Key-Based SSH Authentication for Automation (see page 2547) so that no password is required for the user named "automation".

```
DELPHIX_ENGINE=172.16.180.33
SSH_CMD="ssh automation@${DELPHIX_ENGINE}"


${SSH_CMD}  "cd host; list display=name"
```

13.2.1.7.2 Backward compatibility

Both the CLI and web services API are versioned to support backward compatibility. Future Delphix versions are guaranteed to support clients that explicitly set a version provided the major version identifier is compatible. For more information, see the Web Service API Guide (see page 2752). The CLI will always connect with the latest version, but the `version` command can be used to both display the current version and explicitly bind to a supported version.

Users building a stable set of scripts can run `version` to get the current version. Scripts can then run the `version <id>` command to guarantee that their scripts will be supported on future versions. For more information on the different API versions and how they map to Delphix versions, see the API Version Information (see page 2753) section.

13.2.1.7.3 Parsing CLI output

The default text output of the CLI is unstable. Any attempt to parse the output is certain to run into difficulties in repeatable results for unknown input, as well as instability as the text output is changed in subsequent releases. Column headings, column order, and the number of columns will change in subsequent releases.

You can specify a version in your scripts to counteract this, but you will not be able to take advantage of new features and fixes.

13.2.1.7.4 CLI as a development tool for complex automation

While the CLI is useful for simple automation tasks, it can be slow and overly complicated due to the many round trips needed to control the automation logic. For example, to disable all the environments for an engine, you could write a script which lists the environments and modifies each one:

```
DELPHIX_ENGINE=172.16.180.33
SSH_CMD="ssh automation@${DELPHIX_ENGINE}"
env_array=(`${SSH_CMD}  "version 1.5.0; cd environment; list display=name" | grep -v
NAME` )
for i in "${env_array[@]}"
do
  ${SSH_CMD} "version 1.5.0; cd environment; select $i; disable; commit"
done
```

This script works, but it will be slow on systems with many environments since each SSH command will start a new session.

The web service APIs are superior when performing many operations as a single logical unit. The web service APIs also provide substantially more data with a single call than what is shown in the CLI output, which can greatly simplify your code and avoid multiple round trips.

However, the input and output of web service API calls are JSON data, and it can be difficult to quickly determine what the input and output will look like.

For this reason, the CLI provides two options that can greatly assist you in the development of complex automation: **JSON Output** and **Tracing.**

`(setopt format=json)` changes the CLI to the output of all results to parseable JSON (javascript object notation). This is the fastest and easiest way to quickly see what the JSON output will look like when executed via the Web Service APIs. The JSON format has wide support in a variety of programming languages; see http://www.json.org for more information.

`(setopt trace=true)` will display the underlying HTTP calls being made with each operation and their JSON payload. This allows you to determine the GET and POST calls, and their JSON payloads, which perform the actions that you need to power your automation.

`(setopt format=text)` changes the CLI back into its regular output mode. `(setopt trace=false)` turns off the trace display.

> Using both options will show the JSON output twice

The fastest way to develop complex automation is to experiment with the CLI and copy the underlying API calls to a custom system for better control over behavior.

```
delphix421> setopt trace=true
delphix421> cd user
delphix421 user> create
delphix421 user create *> ls
Properties
    type: User
    name: (required)
    authenticationType: (unset)
    credential: (unset)
..... (Output Truncated) .....
    userType: DOMAIN
    workPhoneNumber: (unset)
delphix421 user create *> set name=Jose
delphix421 user create *> set authenticationType=NATIVE
delphix421 user create *> set credential.password=Password1
delphix421 user create *> commit;
=== POST /resources/json/delphix/user ===
{
    "type": "User",
    "name": "Jose",
    "authenticationType": "NATIVE",
    "credential": {
        "type": "PasswordCredential",
        "password": "Password1"
    }
}
=== RESPONSE ===
{
    "type": "OKResult",
    "status": "OK",
    "result": "USER-35",
    "job": null,
    "action": "ACTION-107"
}
```

Using the output above, you can see that to create a user you must use the URL "http://myengine/resources/json/delphix/user". You will use a POST command and pass a JSON payload which looks like the above. You will get a JSON response like the above, and can validate that the status is "OK".

13.2.2 Delphix objects

These topics describe the object model for the Delphix Engine command-line interface.

The Delphix object model is a flexible system for describing arbitrary hierarchies and relationships of objects. In order to enable the current and future functionality of the system, the relationship between objects is not always immediately obvious. The CLI is merely a veneer atop the web services layer to ensure that the full complement of functionality expressed by the API is always available, but this requires users to have some understanding of how objects are represented in the system.

This section covers the following topics:

- Object type hierarchy (see page 2531)
- Object names and references (see page 2531)
- Databases and environments (see page 2533)
- Asynchronous jobs (see page 2535)

13.2.2.1 Object type hierarchy

This topic describes the object type hierarchy for the Delphix Engine command-line interface.

All Delphix objects have an associated type. This type determines what properties are available for a particular object, the format of those properties, and controls how the system interprets objects and commands. The type hierarchy uses polymorphic inheritance to allow for common properties and behavior to be defined at a single point while permitting dramatically different types of objects to co-exist without requiring a completely separate API for each. For example, the `SourceConfig` object is the base type for all external database configurations, but it has children that include `OracleSIConfig` and `OracleRACConfig` types that refer to a single instance and RAC databases, respectively.

When specifying input types, the system will attempt to determine types appropriate for the current operation, but there are times when the type must be explicitly set, either because the operation supports multiple possible inputs, or the object can embed an abstract type. In these cases, it may be necessary to explicitly set the `type` property. Setting the type may change the set of visible properties and the resulting validation that is performed, but it will not affect any properties that are already set.

13.2.2.2 Object names and references

This topic describes the use of object names and references in the Delphix Engine command-line interface.

Most Delphix objects are persistent objects in that they have a well-known identity on the server and associated persistent state. The exceptions are objects used only as input to other operations, or global objects that have a persistent state but don't require any explicit identity since they always exist.

Persistent objects have both a name and a reference. The reference is the canonical identifier for the object and remains valid even if the object is renamed on the server. It is an opaque token that should never be interpreted by the client; the format may change in future releases though backward compatibility with current references will be maintained. All web service APIs operate using references. References can be used in the CLI when selecting objects, but given that they are a programmatically generated internal concept, they are difficult for most users to use.

The object name, on the other hand, is a much more convenient way to refer to objects but suffers from the fact that it is not guaranteed to be globally unique. When displaying or setting references, the CLI will convert to or from the 'canonical name' based on the type of the reference and the current set of objects on the system. The canonical name has the form:

```
<Type>:/<Parent>/<Object>@<Namespace>
```

The type, parent, and namespace are only included if the local object name conflicts with other objects on the system that would otherwise be valid for the given type specification. Not all objects have names relative to their parent; groups, environments, users, and many other objects are globally unique on the system. This “best fit” method is used both when displaying references as well as when setting properties that are references. If the given name potentially matches multiple objects when attempting to set a reference property, then an error is displayed that includes a list of possible names to clarify which object is being referred to. The conversion from reference to name on display only happens with text output format. When the output format is JSON, the raw content is displayed (including the local name) and it is up to the consumer to format names appropriately based on their semantics. The conversion from name to reference when setting properties always occurs. Consumers can use references, optionally prefixed with a backtick (`) character to signify they are references in the unlikely event that someone has created an object with the same name as a valid reference.

> Providing unique names for objects without the use of forward slashes ('/') and at signs ('@') will provide the simplest CLI experience when referencing objects.

Here are some scenarios for databases and groups and their resulting behavior:

No conflicting database name

The local name will be used when displaying references to the object, and can be used when setting references:

```
set container=example
```

Databases with the same name in different groups

The parent group name must be used when displaying references to the object and when setting references to the object:

```
set container=group1/example
```

Databases with the same name in different namespaces

The namespace name must be used when displaying references to the object and when setting references to the object:

```
set container=example@namespace
```

Objects of different types but with the same name

This conflict is exceptionally rare, as the reference context typically constrains the set of possible objects to be a single type, but there are cases (such as alerts, or policy targets) that can be applied to any object. In these cases, the type name must be included to uniquely identify the object:

```
set target=Container:/group1/example
```

In the event that one of the named components contains a slash (‘/’) or an at-sign ('@'), single quotes must be used to disambiguate the name from its parent or namespace.

13.2.2.3 Databases and environments

This topic describes the relationship between database container objects and environments in the Delphix Engine object model.

The core Delphix objects revolve around the notion of environments and databases, known at the API layer as **containers**. Understanding how these objects relate to each other is crucial to operating effectively within the CLI. This section provides an overview of these objects; for more information about a particular representation such as Oracle RAC, see the Web Service API Guide (see page 2752).

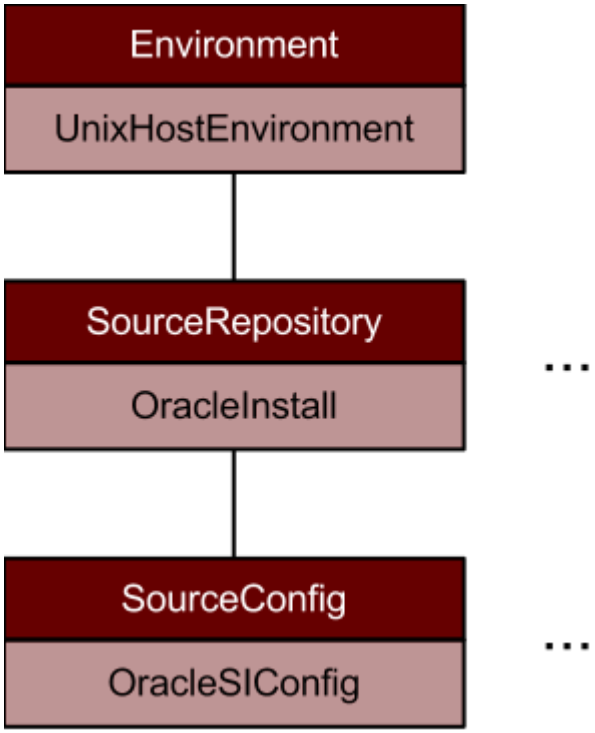
> In variations of a use case where a database propagates down to lower environment databases (for keeping consistent data), records created in the lower tables would likely be overwritten by the propagating database once they are synced. Delphix is keeping track of changes at the data block level on disk, which would render this result.
> As a solution, you could create a clone of the propagating database and then programmatically compare/merge the changes. You can also use a product like Oracle Goldengate, HVR, IBM Datastage, etc, to capture (extract) and propagate (replicate to lower) changes.

13.2.2.3.1 Environment components

- An **environment** is the root representation of the external state that manages database instances. An environment could be a single host (`UnixHostEnvironment`) or an Oracle cluster (`OracleClusterEnvironment`). Environments exist to contain repositories, and each environment may have any number of repositories associated with it.
- A **repository** is an entity that contains database instances. Repositories are typically installation directories (`OracleInstall`) within an environment. Within each repository is any number of `SourceConfig` objects, which represent known database instances.
- The **source config** exists independent of Delphix, and could represent a possible dSource (in which case there is no associated database object), or could be managed entirely by Delphix (for VDBs). The source config contains intrinsic properties of the database instance, while the source (described below) contains information specific to Delphix and only exists when the source config is linked to a dSource or VDB.

Most environment objects are created through the act of **discovery**. By specifying a host, Delphix will attempt to automatically discover all environments, repositories, and source configs. These objects can also be added manually after the fact in cases where discovery fails.

The environment hierarchy can be represented as depicted below. The generic type is listed in the top portion of each box, with an example of the Oracle single instance objects in the lower portion of each box. Each of these objects can contain multiple child objects with it.

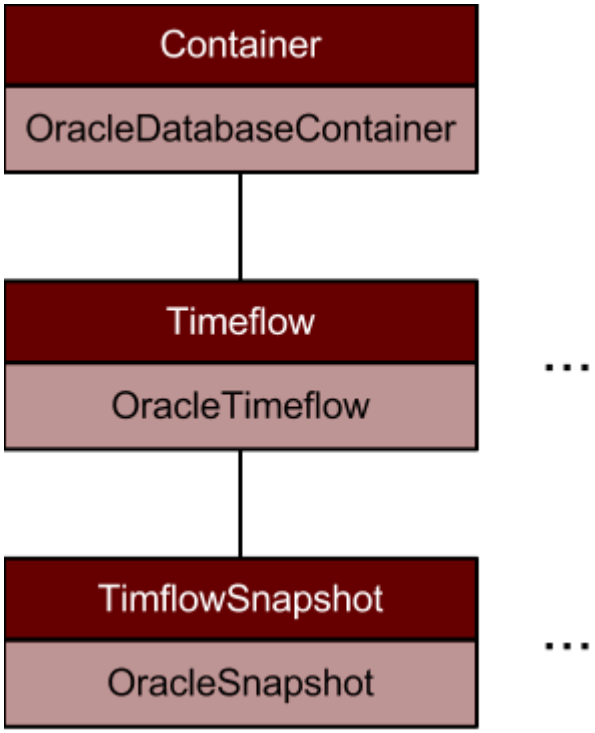


13.2.2.3.2 Database components

The core of all databases within Delphix is the container that holds all the physical data associated with the database, whether a dSource or VDB. Within each container is a **Timeflow**, which represents a single timeline of change within the database history. Currently, a container can only have one Timeflow, though this limitation may be relaxed in a future release.

Within a Timeflow are two important objects: `TimeflowSnapshot` objects and `TimeflowRange` objects. Timeflow **range** represents the provisionable ranges within the history of the Timeflow, while Timeflow **snapshot** represents a point at which a snapshot was taken and therefore is more likely to provision in a short amount of time.

The environment hierarchy can be represented as depicted below. Each container may be associated with a **source**, which is the Delphix representation of an external database when it is associated with a container, and contains information specific to managing that source. Not all source configs within an environment have a source associated with them (as is the case with linkable databases), but all sources must have a source config. Containers may have no sources associated with them if they are unlined; sources can be manually attached at a later point. Currently, each container can have at most once source associated with it, though this may change in a future release.



13.2.2.4 Asynchronous jobs

This topic describes conditions under which command-line interface operations may spawn jobs that run in the background and using the `wait` option to wait for job completion.

Not all operations can be performed in the context of a single web service API call. For cases where there is a long-running operation that cannot be executed quickly and transactionally, a job may be dispatched to do the remaining work in the background. For more information on jobs and their semantics, see the topic Viewing action status (see page 794). Within the CLI, any command can potentially result in an asynchronous operation. The default behavior is to wait for any such job to complete and display its progress in the CLI.

In the event that you do not want to wait for the operation to complete, the global wait option can be set `(setopt wait=false)`. If disabled, the CLI will display the reference to any job that was dispatched, but not wait for it to complete.

13.2.3 Command reference

These topics describe the core built-in commands within the CLI. It is not an exhaustive list of all commands in all contexts. For an object or type-specific commands, consult the API documentation.

This section covers the following topics:

- CLI help and display commands (see page 2536)
- CLI context commands (see page 2536)
- CLI object commands (see page 2537)
- CLI miscellaneous commands (see page 2537)

- CLI property commands (see page 2538)

13.2.3.1 CLI help and display commands

This topic describes help and display commands for the Delphix Engine command-line interface.

Command	Description
`children`	Display all statically defined children valid for the current context. These children can be targets of the `cd` command.
`commands`	Display all build in commands valid for this context.
`help`	Display all commands and properties valid for the current context. Specifying a command or property will provide more information about that command or object. When nested properties are present, only top-level properties are displayed by default, though specifying a particular property will display the entire hierarchy.
`ls`	Display children, commands, objects, and operations valid in the current context. Only those sections that are relevant in the current context are displayed.
`operations`	Display available context-specific operations. These operations require an explicit `commit` command to execute the operation, or `discard` to abort it.

13.2.3.2 CLI context commands

This topic describes context commands for the Delphix Engine command-line interface.

Command	Description
`back`	Return to the previous visited valid context. This history only tracks contexts that were actually visited, so running `database` “example” followed by back will return you to the root context, not the database (because the two were executed as part of one action and never actually visited). If a previous context was deleted or is no longer valid, this command will skip over it.
`cd`	Switch to the given child. This is identical to typing the name of the child itself, but also support UNIX-style directory structures, such as “/” and “..”. This allows for contexts to be chained such as `cd ../database/template`.

Command	Description
`history`	Display the history of input to the shell. The shell supports the ability to move back and forth in the history using the up and down arrows.
`up`	This is an alias for `cd ..` for the benefit of those less familiar with UNIX filesystem navigation. Unlike back, which only returns to the previous context only if it was visited, and may return to a child context, this command will always return to the immediate parent context.

13.2.3.3 CLI object commands

This topic describes object commands for the Delphix Engine Command Line interface.

Command	Description
`list`	List all objects of a particular type when in the appropriate root context. Different contexts may support different options to the list command to constrain the output; run `help list` to see possibilities.
`select`	Select an object by name within a list.

13.2.3.4 CLI miscellaneous commands

This topic describes miscellaneous commands for the Delphix Engine command-line interface.

Command	Description
`echo`	Print the input arguments.
`exit`	Exit from the current CLI session. This is equivalent to sending the EOF control character (typically Ctrl-D) or closing your client SSH application.
`getopt`	Get the current value of a global configuration option. The list of global options can be retrieved by running `help getopt`, but include options for controlling JSON output (`format`), tracing HTTP calls (`trace`), and enabling synchronous job semantics (`wait`).

Command	Description
`setopt`	Set the value of a global configuration option.
`version`	Display the current API version or bind to a particular version. See the CLI automation (see page 2528) section for more information.

13.2.3.5 CLI property commands

This topic describes property commands for the DelphixEngine command-line interface.

Command	Description
`commit`	When in operation context, commit the changes and execute the operation.
`discard`	When in operation context, discard any changes and abort the operation.
`edit`	Change the current context to be relative to a particular object property when in operation context.
`get`	Get all properties (with no arguments) or a particular property of the current object.
`revert`	Revert a particular property to its default value, either the value of the underlying object during an update or the default command input value.
`set`	Set the value of one or more properties. These properties can be specified as `name=value`, or as simply the property name. When only the property name is specified the CLI will prompt for the value to use, optionally obscuring the input if the property is a password.
`unset`	Clear the current value of a property. This is not the same as reverting the property, though this can have semantically identical behavior in the case that the default value is unset.

13.2.4 CLI cookbook: common workflows, tasks, and examples

This section contains the following topics:

- Authentication and users (see page 2539)
- CLI cookbook: system administration (see page 2555)

- CLI cookbook: hosts and environments (see page 2590)
- CLI cookbook: source databases and dSources (see page 2598)
- CLI cookbook: VDBs (see page 2618)
- CLI cookbook: replication (see page 2701)
- CLI cookbook: Delphix self-service actions (see page 2707)
- CLI cookbook: hooks and hook templates (see page 2738)
- CLI cookbook: network performance (see page 2740)
- Kerberos CLIs (see page 2745)

13.2.4.1 Authentication and users

These topics describe command-line interface procedures for authentication and managing users.

- CLI cookbook: configuring SAP ASE manual discovery (see page 2539)
- CLI cookbook: changing HTTP and HTTPS web connections (see page 2542)
- CLI cookbook: configuring key-based SSH authentication for automation (see page 2547)
- CLI cookbook: setting up SSH key authentication for UNIX environment users (see page 2548)
- CLI cookbook: configuring SSH host verification for UNIX environments (see page 2551)

13.2.4.1.1 CLI cookbook: configuring SAP ASE manual discovery

13.2.4.1.1.1 Overview

This topic describes how to use CLI commands to manually add ASE repositories to an SAP ASE environment. Discovery is the process by which the Delphix Engine identifies data sources and data dependencies on a remote environment. SAP ASE repository discovery is done automatically when an environment is added to the Delphix Engine or when an already added environment is refreshed. In some cases, automatic discovery does not discover all of the repositories in an SAP ASE environment. These repositories may be added using manual discovery.

> Unlike automatically discovered instances, manually discovered instances are not automatically deleted if the environment is refreshed when the instance is not running. The physical attributes such as ASE listener port, installation directory, and instance owner are not updated during an environment refresh either. If you change a physical attribute, you must manually update the repository.

To manually discover an SAP ASE repository you will need to:

- Add an SAP ASE environment
- Use CLI to manually discover a repository

13.2.4.1.1.2 Creating an SAP ASE environment

Please refer to Adding an SAP ASE Environment (see page 1905) for detailed steps.

13.2.4.1.1.3 Manually discover a repository

This example uses `sc-dev3.dc1` as the example environment.

1. Log into CLI and cd to the **repository** menu:

```
$ ssh admin@sc-dev3.dc1
Password:
sc-dev3.dc1> cd repository
sc-dev3.dc1 repository>
```

2. Add (manually discover) an SAP ASE repository instance:

> **Note:** The values used in the following code block are specific to the example instance we are adding.

3.

```
sc-dev3.dc1 repository> create
sc-dev3.dc1 repository create *> ls
Properties
    type: ASEInstance
    credentials: (unset)
    dbUser: (unset)
    dumpHistoryFile: (unset)
    environment: (required)
    installationPath: (required)
    instanceName: (required)
    instanceOwner: (required)
    isqlPath: (unset)
    linkingEnabled: true
    ports: (required)
    provisioningEnabled: true
    servicePrincipalName: (unset)
    staging: false
    version: (unset)
sc-dev3.dc1 repository create *> set credentials.type=PasswordCredential
sc-dev3.dc1 repository create *> set credentials.password=sybase
sc-dev3.dc1 repository create *> set dbUser=sa
```

```
sc-dev3.dc1 repository create *> set environment=rh610-ebf-ase-tgt-
dev3.dc3.delphix.com
sc-dev3.dc1 repository create *> set installationPath=/opt/sybase/15-7/sp139/
install
sc-dev3.dc1 repository create *> set instanceName=ASE157_TGT
sc-dev3.dc1 repository create *> set instanceOwner=sybase
sc-dev3.dc1 repository create *> set ports=5400
sc-dev3.dc1 repository create *> ls
Properties
    type: ASEInstance
    credentials:
        type: PasswordCredential (*)
        password: ******** (*)
    dbUser: sa (*)
    dumpHistoryFile: (unset)
    environment: rh610-ebf-ase-tgt-dev3.dc3.delphix.com (*)
    installationPath: /opt/sybase/15-7/sp139/install (*)
    instanceName: ASE157_TGT (*)
    instanceOwner: sybase (*)
    isqlPath: (unset)
    linkingEnabled: true
    ports: 5400 (*)
    provisioningEnabled: true
    servicePrincipalName: (unset)
    staging: false
    version: (unset)
sc-dev3.dc1 repository create *> commit
    `ASE_INSTANCE-3
sc-dev3.dc1 repository>
```

13.2.4.1.1.4 Updating a repository

Adding onto the above, the following example illustrates updating an SAP ASE instance's version after upgrading SAP ASE:

> ⚠ Take caution when setting the version string. Make sure it matches the output as displayed by the "select @@version" query all the way out to the patch level (PL). For example, "15.7 SP138" or "16.0 SP02 PL01".

```
sc-dev3.dc1> repository
sc-dev3.dc1 repository> select ASE157_TGT
sc-dev3.dc1 repository 'ASE157_TGT'> update
sc-dev3.dc1 repository 'ASE157_TGT' update *> set version="15.7 SP139"
sc-dev3.dc1 repository 'ASE157_TGT' update *> ls
Properties
    type: ASEInstance
    credentials:
```

```
        type: PasswordCredential
        password: ********
    dbUser: sa
    dumpHistoryFile: (unset)
    installationPath: /opt/sybase/15-7/sp139/install
    instanceOwner: sybase
    isqlPath: /opt/sybase/15-7/sp139/install/OCS-15_0/bin/isql_r64
    linkingEnabled: true
    ports: 5400
    provisioningEnabled: true
    servicePrincipalName: (unset)
    staging: false
    version: 15.7 SP139
sc-dev3.dc1 repository 'ASE157_TGT' update *> commit
sc-dev3.dc1 repository 'ASE157_TGT'>
```

13.2.4.1.2 CLI cookbook: changing HTTP and HTTPS web connections

By default, the Delphix Engine allows both HTTP and HTTPS web connections. The following steps provide instructions on how to change their configuration:

1. Via CLI, login to the Delphix Engine as a system administrator (sysadmin).
2. Go to "service > httpConnector".
3. Initiate an update.
4. Set "httpMode" to the desired value among:
    a. "BOTH": accepts HTTP and HTTPS connections (this is the default)
    b. "HTTPS_ONLY": accepts only HTTPS connections
    c. "HTTP_ONLY": accepts only HTTP connections
    d. "HTTP_REDIRECT": accepts HTTPS connections and redirects HTTP connections to HTTPS.
    e. "HTTP_REDIRECT_WITH_HSTS": redirect all requests made over HTTP to HTTPS and add Strict-Transport-Security Header to all responses.
5. Commit your Change. The Delphix web application will restart.

The following is an example of how to set the Delphix Engine to accept HTTPS connections and redirect HTTP connections to HTTPS.

```
cd /service/httpConnector
update
set httpMode=HTTP_REDIRECT
commit
```

13.2.4.1.3 CLI cookbook: replacing the HTTPS (HTTP secure) certificate

This topic explains how to replace the HTTPS (HTTP Secure) certificate used by the Delphix Virtualization Engine. There are two methods of replacing the certificate. The key difference between the two is whether Delphix or the user is providing the key pair (public and private key).

13.2.4.1.3.1 Delphix provided key pair

Use the following instructions to provide an HTTPS certificate chain for a key pair created by the Delphix Engine. Once the key pair is created users can download a Certificate Signing Request (CSR) to generate a signed certificate from the CA of their choice. This is done using the "create" operation in the "/service/tls/csr" API as seen below when using the CLI.

```
hostname.domainname> service tls csr
hostname.domainname service tls csr> create
hostname.domainname service tls csr create *> ls
Properties
   type: CertificateSigningRequestCreateParameters
   dname:
       type: X500DistinguishedNameComposite
       dname: (required)
  :
       type: EndEntityHttps
   forceReplace: false
   keyPair:
       type: RsaKeyPair
       keySize: 2048
       signatureAlgorithm: SHA256withRSA
```

The first key property is the dname. This will be used as the subject name of the CSR and resulting X.509 certificate unless it is changed when the certificate is signed. Delphix supports two different formats for dname:

- a composite string
- a list of fields

Use the composite string as follows:

```
hostname.domainname service tls csr create *> set dname.dname="CN=Delphix CA,
O=Delphix, C=US"
hostname.domainname service tls csr create *> ls
Properties
   type: CertificateSigningRequestCreateParameters
   dname:
       type: X500DistinguishedNameComposite (*)
       dname: CN=Delphix CA, O=Delphix, C=US (*)
```

Use the list of field formats as follows:

```
hostname.domainname service tls csr create *> set
dname.type=X500DistinguishedNameFields
hostname.domainname service tls csr create *> ls
Properties
   type: CertificateSigningRequestCreateParameters
   dname:
```

```
        type: X500DistinguishedNameFields (*)
        city: (unset)
        commonName: Delphix CA (*)
        country: US (*)
        organization: Delphix (*)
        organizationUnit: (unset)
        stateRegion: (unset)
```

The only required field is the commonName (CN).

The only currently supported type for endEntity is EndEntityHttps.

The next property is forceReplace. By default, this is false and means Delphix will not replace the active key pair and certificate with the newly generated keypair and self-signed certificate. If the user wants to replace the active key pair right away before the signed certificate has been created this can be set to true.

The final property keyPair impacts the generated key pair. When creating a new key pair the engine supports two algorithms:

- **RSA -** The supported signature algorithms are SHA256withRSA, SHA384withRSA, and SHA512withRSA . The valid key sizes range from 2048 to 4096.
- **ECDSA -** The supported signature algorithms are SHA256withECDSA, SHA384withECDSA, and SHA512withECDSA. The valid key sizes range from 256 to 571

Once the create operation has completed you can get the CSR in PEM format by selecting the CSR object and looking at the requestInPem property:


```
requestInPem: -----BEGIN CERTIFICATE REQUEST-----
MIIBezCB0gIBADAhMR8wHQYDVQQDExZiYmFrZXIuZGMyLmRlbHBoaXguY29tMIGn
MBAGByqGSM49AgEGBSuBBAAmA4GSAAQBU5WY9+GkCTFvbGHTNJDb/QM3t4YI/9S6
fhCJELx7SbJNti2n0l3mCePenyUuBY9m6BWvUQzlhawZG5YAJ9WdcM+IIPciiNsD
Xmw0eFH05z6yTLMnfBYYZKFbpu/dcK5V8WoltrIC7jTxg/k6jf/WeD+dmyIMQ0Z7
VmwnD6RsaAs7T5lajXkurwPfqQ5MnsmgADAKBggqhkjOPQQDAwOBlwAwgZMCSAGM
quqcnIAxIRDxQ+BzzSywNtozn5ihtfFxtTF/EW/ARBib2l9hq0pwHrIinnLvjW9u
avpAH1pkWHx1w0/O6W6DCZAPIIL3ugJHKsScJqsvaeZzVqJVfQt8g42cL9hKc7ic
HLhuAyMGQOXrEdLbOxtOH6SiExnyEv2Y9LHHYYgRafgGzOoA5tx+mrkr9J+zm8Y=
-----END CERTIFICATE REQUEST-----
```


Once the CSR has been signed and turned into an X.509 Certificate you can replace the certificate using the "service/tls/endEntityCertificate" API. To replace using the CSR method begin by setting the correct type of replace parameters as seen below:

```
hostname.domainname service tls endEntityCertificate> replace
hostname.domainname service tls endEntityCertificate replace *> set
type=EndEntityCertificateReplaceChainParameters
hostname.domainname service tls endEntityCertificate replace *> ls
Properties
    type: EndEntityCertificateReplaceChainParameters
    chain:
        type: PemCertificateChain
        chain: (required)
    endEntity:
        type: EndEntityHttps
```

The "chain" property must contain a list of the entire trust chain from the newly generated end-entity certificate to the root CA.

The CLI might not always interpret newline characters in PEM certificates correctly. Therefore, it is highly recommended to find and replace all newlines ('\n') with an empty string ('') prior to pasting the PEM certificate into the CLI.

To do this in the CLI first run:

```
hostname.domainname service tls endEntityCertificate replace *> edit chain.chain
Then `add` and `set contents` to the PEM certificate for each certificate in the
chain.
hostname.domainname service tls endEntityCertificate replace chain.chain *> add
```

When adding multiple certificates, use the command back after each add. After the final add, enter back and then commit.

The order in which the PEM certificates are added to the list does not matter.

13.2.4.1.3.2 Customer provided key pair

This section describes the steps to take if you are replacing the HTTPS with your own key pair and certificate.

1. To start, you need to add the key pair and full certificate chain as an entry in a file in JKS or PKCS #12 format.
2. Then, send a file upload request to the following endpoint:

```
hostname.domainname service tls endEntityCertificate
requestKeyPairAndCertChainUpload *> ls
Properties
   type: CertificateUploadParameters
   alias: alias_in_keystore (*)
   keypass: (unset)
   keystoreType: JKS
   storepass: ******** (*)
```

```
hostname.domainname service tls endEntityCertificate
requestKeyPairAndCertChainUpload *> commit
   type: FileUploadResult
   token: 8f4361c5-019c-4fee-9306-b7c85e977cf4
   url: /resources/json/delphix/data/upload
```

The **alias** field is where the key pair and certificate is saved in your JKS or PKCS #12 store.

The **keypass** field is the password for the given alias' key. If not set, it uses the keystore's password.

The **storepass** field is the keystore's password.

3. Then, establish a session from the host with the keystore to the Delphix Engine. Choose the location of the cookies, and determine the API version (command example uses 1.9.2):

```
curl -c <path/to/cookies> -X POST --data '{ "type": "APISession", "version":
{ "type": "APIVersion", "major": 1, "minor": 9, "micro": 2 } }' -H "Content-
Type: application/json" http://<delphix_engine_url>/resources/json/delphix/
session
```

4. Login to the Delphix Engine using the established session as a domain or system admin:

```
curl -b <path/to/cookies> -c <path/to/cookies> -X POST --data '{ "type":
"LoginRequest", "username": "sysadmin", "password": "sysadmin" }' -H "Content-
Type: application/json" http://<delphix_engine_url>/resources/json/delphix/
login
```

5. Send the file upload request with the location of your keystore and token from above:

```
curl -b <path/to/cookies> -X POST -F "file=@<path/to/keystore>" -F
"token=8f4361c5-019c-4fee-9306-b7c85e977cf4" http://<delphix_engine_url>/
resources/json/delphix/data/upload
```

6. You can now replace the HTTPS end-entity certificate with the keystore you have uploaded, identified by the token:

```
hostname.domainname service tls endEntityCertificate replace *> set
type=EndEntityCertificateReplaceKeystoreParameters
hostname.domainname service tls endEntityCertificate replace *> set
token=8f4361c5-019c-4fee-9306-b7c85e977cf4
hostname.domainname service tls endEntityCertificate replace *> ls
Properties
   type: EndEntityCertificateReplaceKeystoreParameters (*)
   endEntity:
       type: EndEntityHttps
   token: b0e889ff-847a-4d7d-bd17-c1292ddbb63e (*)
```

```
hostname.domainname service tls endEntityCertificate replace *> commit
```

13.2.4.1.4 CLI cookbook: configuring key-based SSH authentication for automation

This topic describes how to use CLI commands to configure individual users with SSH keys to allow for password-less authentication from a remote host to the CLI in an automated environment.

13.2.4.1.4.1 What is SSH Key-based authentication?

Secure Shell (SSH) is a connection method used to log into UNIX or Linux servers remotely. With Delphix, is it used to connect to the Delphix Command Line Interface (CLI) from a remote computer. This normally requires a password on each connection; however, it is possible to use Key-based Authentication to avoid the password requirement and allow the automation of Delphix commands.

Key-based Authentication relies on a public/private key pair generated on the client system. The private key allows access to any server acknowledging the matching public key as being authorized to login. In order to configure this, a public/private key pair must be created, and the resulting public key should be added to the Delphix server using the CLI.

> The default domain user created on Delphix Engines is now admin instead of delphix_admin. When engines created before 5.3.1 are upgraded to 5.3.1 or later they will retain their old username 'delphix_admin'. To avoid complications Delphix recommends creating users with an admin role and then Disabling delphix_admin.

13.2.4.1.4.2 Procedure

1. Consult your client documentation for information on generating a public/private key pair. The `ssh-keygen` program is typical on UNIX platforms. If you need details on ssh-keygen usage or have unique requirements (such as named RSA keys), see Third-Party SSH Key Generation Example (see page 2547). If you already have a public/private key pair generated on your system, you can skip to step 2.
2. Connect as the user you wish to configure or as a Engine Administrator.

> **Warning:** When you connect to the Delphix Engine with the CLI, you should specify the appropriate namespace (either DOMAIN or SYSTEM). See Connecting to the CLI (see page 2522) for more information.

3. Select the current user, or select a specific user if configuring another user as an administrator.

```
delphix> user current
```

4. Update the user and set the SSH key.

```
delphix user "admin"> update
delphix user "admin" update *> set publicKey="[PASTE KEY]"
delphix user "admin" update *> commit
delphix>
```

Note:
Avoid Newline Characters with Public Key Entry The public key value, which can be quite long, must be entered as a single string with no newlines. When copying and pasting the public key, be sure to avoid introducing any newline characters. For more information on how to manage multiple public keys for password-less user authentication on Delphix, please visit this Knowledge Base[762] article.

5. Verify you can authenticate through the Delphix CLI without a passphrase.

   Example Using Default SSH Key

```
ssh admin@DOMAIN@delphix-server.example.com
Last login: Thu Dec 13 22:16:28 2012 from 192.168.0.2
delphix>
```

   Example Using a Non-default SSH Key File Located at path/to/delphix_key

```
ssh -i path/to/delphix_key admin@DOMAIN@delphix-server.example.com
Last login: Thu Dec 13 22:16:28 2012 from 192.168.0.2
delphix>
```

13.2.4.1.5 CLI cookbook: setting up SSH key authentication for UNIX environment users

This topic describes adding public-key authentication for a UNIX environment user, thus allowing the Delphix server to connect to your UNIX Environments without an explicit password. This method uses the Delphix CLI in order to set up the environment user and gather SSH keys.

762 https://support.delphix.com/Delphix_Virtualization_Engine/Delphix_Admin/How_to_Manage_Multiple_Public_Keys_for_Passwordless_User_Authentication_on_a_Delphix_Engine_(KBA5897)

UNIX host environments (and Oracle cluster environments) can have users configured to use SSH-key-based authentication instead of the traditional password authentication method.

13.2.4.1.5.1 Prerequisites

- You must be able to log into the remote host (or all hosts of an Oracle cluster) and have write access to the `~/.ssh/authorized_keys` file within the desired user's home directory.

13.2.4.1.5.2 Option 1: system key

Within Delphix, there is a per-system SSH public key that can be placed into the `~/.ssh/authorized_keys` file of the remote user. Once this has been done, the Delphix environment user can be configured to use the private key instead of an explicit password. Note that it is also possible to configure an environment to use this system key in the Delphix Management application by navigating to **Manage** > **Environments** and selecting **Public Key** as the **Login Type** for the environment. For details, see Managing Environments (see page 1017).

1. Get the current system public key:

```
delphix> system get sshPublicKey
    ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAse1M7uJX44lVPBljhnxB6MZUTx8VF6cupaVATg120lQonIqx29l
P+Mwp0AWh7C983IDoYDo+AY7RXpcFP9nKksiJnGSGiK6wo9RIiqSnF1x/
VXNkTt2/67RVofoiui4W5fuxD4hOIvoTr47Bg1hh9L6nhP0tnUvS/
rusHFJ+ogxGHm46mwNlgUJUGmLTNao+W0YU693HRLukEch01t4k6olVGaC0eLjYlgBf0Z5XiIcBX6ZW
qVHAhwMinVjAvmfQhirAgCI7gYrd5/PwNl/DC8xyhWuxd2jgA7sSPeRqWY0JHt/
xcmdpIaPxTwtxQLKTnPxrFrQd+l4uf6LKxr5g7w== root@delphix
```

2. Add this key (starting with `ssh-rsa`) to the remote user's `~/.ssh/authorized_keys` file. You will need to get access to this file using an alternate authentication mechanism (such as logging in as the user with a password or logging in as an administrator). Depending on the target OS, you may need to do the following:
   a. If the directory does not exist:

```
$ mkdir ~/.ssh
```

   b. If creating the file or directory as an administrator:

```
# chown -R <username> <home>/.ssh
```

   c. If required by the host SSH configuration, ensure the directory is not world-readable:

```
$ chmod 600 ~/.ssh/authorized_keys
$ chmod 755 ~
```

3. Create or edit an environment user:

```
delphix> environment user create
```

4. Set the user environment and name:

```
delphix environment user create *> set environment=environment1
delphix environment user create *> set name=username
```

5. Set the user credential type to `SystemKeyCredential`:

```
delphix environment user create *> set credential.type=SystemKeyCredential
```

6. Commit the results:

```
delphix environment user create *> commit
```

13.2.4.1.5.3 Option 2: Per-environment key pair

Each environment user can also be configured to use an SSH key pair provided via the CLI or API.

1. Add the public key to the remote user's `~/.ssh/authorized_keys` file. You will need to get access to this file using an alternate authentication mechanism (such as logging in as the user with a password or logging in as an administrator). Depending on the target OS, you may need to do the following:
   a. If the directory does not exist:

   ```
   $ mkdir ~/.ssh
   ```

   b. If creating the file or directory as an administrator:

   ```
   # chown -R <username> <home>/.ssh
   ```

   c. If required by the host SSH configuration, ensure the directory is not world-readable:

   ```
   $ chmod 600 ~/.ssh/authorized_keys
   $ chmod 755 ~
   ```

2. Create or edit an environment user:

```
delphix> environment user create
```

3. Set the user environment and name:

```
delphix environment user create *> set environment=environment1
delphix environment user create *> set name=username
```

4. Set the user credential type to `KeyPairCredential`:

```
delphix environment user create *> set credential.type=KeyPairCredential
```

5. Set the private and public keys:

```
delphix environment user create *> set credential.privateKey="----BEGIN ..."
delphix environment user create *> set credential.publicKey="ssh-rsa AA..."
```

(these example values were trimmed for brevity)

6. Commit the results:

```
delphix environment user create *> commit
```

13.2.4.1.6 CLI cookbook: configuring SSH host verification for UNIX environments

This topic describes how to configure SSH host verification when authenticating to UNIX environments, which lets the Delphix server ensure it connects to the intended environment hosts. This method uses the Delphix CLI to set the SSH key or fingerprint of each host. Currently, it is only possible to configure SSH host verification via the CLI or the Web Service API.

When an SSH key or fingerprint is specified for an environment host, the Delphix server will use it when connecting to that host to verify that host. If the key or fingerprint does not match the information presented by that host, the Delphix server will close that connection and report the problem to the user.

The key types supported by the Delphix server are `RSA` (`ssh-rsa`), `DSA` (`ssh-dsa`), `ECDSA` (`ecdsa-sha2-nistp256`) and `ED25519` (`ssh-ed25519`). The fingerprint types supported are `SHA256` and `SHA512`; the `MD5` type is considered insecure and, therefore, is not supported.

13.2.4.1.6.1 Prerequisites

- To obtain the SSH public key or fingerprint of a host remotely from another machine, you must have the `ssh-keyscan` and `ssh-keygen` utilities.
- To obtain the SSH public key or fingerprint directly from a host, you must be able to log into that host.

13.2.4.1.6.2 Obtaining an SSH key or fingerprint

1. Remotely: List the SSH public keys of the host using the standard utility `ssh-keyscan` and choose one of them. For example:

```
$ ssh-keyscan example.environment.host # example.environment.host:22 SSH-2.0-
OpenSSH_7.4 example.environment.host ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDBsOAAokSRO67jI28syRmX0wY/fKIboLLu/
ofk6BzYLKtkMaK1QC78/6QlelIJUP5HdK8E7Um/iM1JMxry4h9Rl13onYOuJVZkDB9wnJiztSu/
Wl9Eqbt59TU1vGmp/4ulWS3PISl7bxs+l43HzsrjM4dTs2efQ7sLWoW86CDlL7Je4va65/
aopvifxKZeZkT0srB3L8VzHKw9+NJOumy1CI3DIBiICURJd4WZ10IH5TFUDRaUFAc/trzW1gvJY/
Whp892tPHekyP32hOZNIc7oDPx2boZauJVR6/
BHmKpmLlhkPpEqfZP8JW+JNsNnLr9BEmwJXaEpwnua1BUii8F ...
```

   where the key is the Base64-code string to the right of the key type. In this example, the RSA SSH public key is the string starting in "`AAAAB3Nza`" and ending in "`a1BUii8F`".

2. Alternatively, from the host: Log into the host and print the file contents of your public key of choice. For example:

```
$ cat /etc/ssh/ssh_host_rsa_key.pub ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDBsOAAokSRO67jI28syRmX0wY/fKIboLLu/
ofk6BzYLKtkMaK1QC78/6QlelIJUP5HdK8E7Um/iM1JMxry4h9Rl13onYOuJVZkDB9wnJiztSu/
Wl9Eqbt59TU1vGmp/4ulWS3PISl7bxs+l43HzsrjM4dTs2efQ7sLWoW86CDlL7Je4va65/
aopvifxKZeZkT0srB3L8VzHKw9+NJOumy1CI3DIBiICURJd4WZ10IH5TFUDRaUFAc/trzW1gvJY/
Whp892tPHekyP32hOZNIc7oDPx2boZauJVR6/
BHmKpmLlhkPpEqfZP8JW+JNsNnLr9BEmwJXaEpwnua1BUii8F
```

3. If a fingerprint is preferred, use `ssh-keygen` in conjunction with the above commands. For example, remotely:

```
$ ssh-keyscan example.environment.host | ssh-keygen -E sha256 -lf - ... 2048
 SHA256:8Cx8cBg/pSbkId3uu2vATeugkAXcm+Ruu9hu66OXEGI example.environment.host
(RSA) ...
```

   where the fingerprint is "`SHA256:8Cx8cBg/pSbkId3uu2vATeugkAXcm+Ruu9hu66OXEGI`" (the string between the key size and the hostname).
   Alternatively, from the host:

```
$ cat /etc/ssh/ssh_host_rsa_key.pub | ssh-keygen -E sha256 -lf - 2048
 SHA256:8Cx8cBg/pSbkId3uu2vATeugkAXcm+Ruu9hu66OXEGI user@environment.host (RSA)
```

13.2.4.1.6.3 Configuring SSH host verification during environment creation

The default SSH verification strategy is `SshAcceptAlways`, which always trusts the key or fingerprint presented by a remote host. The procedure to change this strategy to perform fingerprint-based host verification for single-host Unix environments is:

1. Set the new strategy to `SshVerifyFingerprint`:

```
delphix environment create *> edit hostParameters.host.sshVerificationStrategy
delphix environment create hostParameters.host.sshVerificationStrategy *> set
type=SshVerifyFingerprint
```

2. Set the key type and fingerprint type. For example:

```
delphix environment create hostParameters.host.sshVerificationStrategy *> set
keyType=RSA delphix environment create
hostParameters.host.sshVerificationStrategy *> set fingerprintType=SHA256
```

3. Set the fingerprint. For example:

```
delphix environment create hostParameters.host.sshVerificationStrategy *> set
fingerprint=SHA256:8Cx8cBg/pSbkId3uu2vATeugkAXcm+Ruu9hu66OXEGI
```

4. Alternatively, you can specify the key itself using the `SshVerifyRawKey` strategy. For example:

```
delphix environment create hostParameters.host.sshVerificationStrategy *> set
type=SshVerifyRawKey delphix environment create
hostParameters.host.sshVerificationStrategy *> set keyType=RSA delphix
environment create hostParameters.host.sshVerificationStrategy *> set
rawKey=AAAAB3NzaC1yc2EAAAADAQABAAABAQDBsOAAokSRO67jI28syRmX0wY/fKIboLLu/
ofk6BzYLKtkMaK1QC78/6QlelIJUP5HdK8E7Um/iM1JMxry4h9Rl13onYOuJVZkDB9wnJiztSu/
Wl9Eqbt59TU1vGmp/4ulWS3PISl7bxs+l43HzsrjM4dTs2efQ7sLWoW86CDlL7Je4va65/
aopvifxKZeZkT0srB3L8VzHKw9+NJOumy1CI3DIBiICURJd4WZ10IH5TFUDRaUFAc/trzW1gvJY/
Whp892tPHekyP32hOZNIc7oDPx2boZauJVR6/
BHmKpmLlhkPpEqfZP8JW+JNsNnLr9BEmwJXaEpwnua1BUii8F
```

5. When you are done specifying all other environment creation parameters, create the environment:

```
delphix environment create *> commit
```

6. If you are creating a Unix cluster, the procedure to start editing the SSH verification settings for the first node in that cluster is similar to the single-host case. For example:

```
delphix> environment create delphix environment create *> set
type=OracleClusterCreateParameters delphix environment create *> edit nodes
delphix environment create nodes *> add delphix environment create nodes 0 *>
edit delphix environment create nodes 0 *> edit
hostParameters.host.sshVerificationStrategy # configure SSH verification
settings
```

Note that only one node (host) can be specified and configured when creating a Unix cluster environment. The SSH verification settings for the remaining hosts can only be specified afterward by editing them via "`host select <hostname> update`", once the corresponding nodes have been discovered or added. See the next section.

13.2.4.1.6.4 Configuring SSH host verification for existing hosts

For any Unix environment host, whether it is single or part of a cluster, you can set up or change its configuration for SSH verification after the environment has been added by editing the host. For example:

```
delphix> host select example.environment.host update sshVerificationStrategy delphix
host 'example.environment.host' update sshVerificationStrategy *> edit
sshVerificationStrategy # configure SSH verification settings
```

13.2.4.1.6.5 Testing SSH host verification

It is possible to configure an SSH key or fingerprint when performing a connectivity test to a Unix host. This can be done without even creating an environment for that host. For example:

```
delphix> connectivity ssh # configure address and credentials ... delphix
connectivity ssh *> edit sshVerificationStrategy # configure SSH verification
settings
```

13.2.4.1.6.6 SSH host verification errors

When the Delphix server initiates an SSH connection to a host, if SSH host verification is configured (i.e. the verification strategy is not the default `SshAcceptAlways`), the server will first check the key presented by the host. Only if this check passes, the server will attempt to authenticate. Therefore, a host key verification failure will be reported as "Unrecognized key or fingerprint" to the user before any authentication failure. For example:

```
delphix connectivity ssh *> set credentials.password=<BAD PASSWORD> delphix
connectivity
```

13.2.4.2 CLI cookbook: system administration

These topics describe various system administration tasks that can be performed with the command-line interface, such as changing the name of the <default> group and setting up network connectivity.

This section covers the following topics:

- CLI cookbook: setting NFS version (see page 2555)
- CLI cookbook: configuring a second network interface (see page 2556)
- Removing IP addresses from a network interface (see page 2557)
- CLI cookbook: adding a static route (see page 2560)
- CLI cookbook: changing the default group name (see page 2561)
- CLI cookbook: how to change a Delphix user password (see page 2562)
- CLI cookbook: retrieve capacity information (see page 2565)
- CLI cookbook: view storage test results (see page 2566)
- CLI cookbook: how to change IP Address of Delphix engine (see page 2566)
- CLI cookbook: about alert notifications (see page 2568)
- CLI cookbook: obtaining CPU performance information using CLI (see page 2576)
- CLI cookbook: rebooting the Delphix engine via CLI (see page 2588)
- CLI cookbook: disabling user-click analytics (see page 2588)
- CLI cookbook: changing the API version (see page 2589)

13.2.4.2.1 CLI cookbook: setting NFS version

This topic describes how to configure the NFS version used for mounting VDBs.

13.2.4.2.1.1 Procedure

1. Log in to the Delphix Engine as the sysadmin user and switch to the Service NFS context. Then use the `ls` command to view the current configuration.

```
delphix service nfs> ls
Properties
    type: NfsConfig
    mountVersion: Automatic
```

2. Run the `update` command and configure the version (Options are Automatic, NFSv3, and NFSv4).

```
delphix service nfs> update
delphix service nfs update *> set mountVersion=NFSv4
```

3. Commit the operation.

```
delphix service nfs update *> commit
ip-10-110-212-129 service nfs> list
Properties
    type: NfsConfig
    mountVersion: NFSv4
```

13.2.4.2.2 CLI cookbook: configuring a second network interface

This topic describes how to configure a static IP address on a second network interface.

13.2.4.2.2.1 Procedure

1. Add a NIC to the Delphix virtual machine. The specific procedure will depend on the platform. For example, on VMware, a VMXNET3 virtual network adapter can be added dynamically from vSphere or other administrative interfaces, and the Delphix Engine will recognize the new NIC without a reboot. On other platforms, a reboot may be required for the Delphix Engine to recognize the new virtual hardware.
2. Log in to the Delphix Engine as the sysadmin user and switch to the network interface context. Then use the `list` command to view the available network interfaces, and select the new interface to be configured.

```
delphix network interface> list
NAME
vmxnet3s0
vmxnet3s1
delphix network interface> select vmxnet3s1
delphix network interface "vmxnet3s1"> get
    type: NetworkInterface
    name: vmxnet3s1
    addresses: (empty)
    device: vmxnet3s1
    macAddress: 0:c:29:e5:4c:c1
    mtu: 1500
    mtuRange: 60-9000
    reference: NETWORK_INTERFACE-vmxnet3s1
    state: DOWN
```

3. Run the `update` command and configure a static address.

```
delphix network interface "vmxnet3s1"> update
delphix network interface "vmxnet3s1" update *> edit addresses.0
```

```
delphix network interface "vmxnet3s1" update addresses.0 *> set address=10.1.2.
3/24
delphix network interface "vmxnet3s1" update addresses.0 *> get
    type: InterfaceAddress (*)
    address: 10.1.2.3/24 (*)
    addressType: STATIC (*)
```

4. `Commit` the operation.

```
delphix network interface "vmxnet3s1" update addresses.0 *> commit
delphix network interface "vmxnet3s1"> get
    type: NetworkInterface
    name: vmxnet3s1
    addresses:
        0:
            type: InterfaceAddress
            address: 10.1.2.3/24
            addressType: STATIC
            state: OK
    device: vmxnet3s1
    macAddress: 0:c:29:e5:4c:c1
    mtu: 1500
    mtuRange: 60-9000
    reference: NETWORK_INTERFACE-vmxnet3s1
    state: OK
```

13.2.4.2.3 Removing IP addresses from a network interface

This topic describes how to remove one or more IP addresses from a network interface.

13.2.4.2.3.1 Removing all IP addresses

Procedure

1. Log in to the Delphix Engine as the sysadmin user and switch to the network interface context. Then use the `list` command to view the available network interfaces, and select the interface.

```
delphix> network interface
delphix network interface> list
NAME    MACADDRESS         MTU
ens160  02:dc:02:00:7f:06  1500
delphix network interface> select ens160
delphix network interface 'ens160'> get
    type: NetworkInterface
    name: ens160
```

```
    addresses:
        0:
            type: InterfaceAddress
            address: 10.43.70.135/16
            addressType: DHCP
            enableSSH: true
            sessionInUse: false
            state: OK
    device: ens160
    macAddress: 02:dc:02:00:7f:06
    mtu: 1500
    mtuRange: 68-9000
    reference: NETWORK_INTERFACE-ens160
    state: OK
```

2. Run the `update` command and unset the addresses property to remove all addresses. Note that in this example, we only had one address to remove, but this procedure removes all addresses even if more than one were configured.

```
delphix network interface 'ens160'> update
delphix network interface 'ens160' update *> unset addresses
delphix network interface 'ens160' update *> get
    type: NetworkInterface
    name: ens160
    addresses: (unset) (*)
    mtu: 1500
```

3. `Commit` the operation.

```
delphix network interface 'ens160' update *> commit
delphix network interface 'ens160'> get
    type: NetworkInterface
    name: ens160
    addresses: (empty)
    device: ens160
    macAddress: 02:dc:02:00:7f:06
    mtu: 1500
    mtuRange: 68-9000
    reference: NETWORK_INTERFACE-ens160
    state: OK
```

13.2.4.2.3.2 Removing a specific IP address

1. Log in to the Delphix Engine as the sysadmin user and switch to the network interface context. Then use the `list` command to view the available network interfaces, and select the interface. Note that in this example, there are two addresses, a static address, and a DHCP address.

```
delphix> network interface
delphix network interface> list
NAME    MACADDRESS         MTU
ens160  02:dc:02:00:7f:06  1500
delphix network interface> select ens160
delphix network interface 'ens160'> get
    type: NetworkInterface
    name: ens160
    addresses:
        0:
            type: InterfaceAddress
            address: 10.11.12.13/24
            addressType: STATIC
            enableSSH: true
            sessionInUse: false
            state: OK
        1:
            type: InterfaceAddress
            address: 10.43.70.135/16
            addressType: DHCP
            enableSSH: true
            sessionInUse: false
            state: OK
    device: ens160
    macAddress: 02:dc:02:00:7f:06
    mtu: 1500
    mtuRange: 68-9000
    reference: NETWORK_INTERFACE-ens160
    state: OK
```

2. Run the `update` command and unset one of the addresses. In this example, only the static address is removed, retaining the DHCP address.

```
delphix network interface 'ens160'> update
delphix network interface 'ens160' update *> get
    type: NetworkInterface
    name: ens160
    addresses:
        0:
            type: InterfaceAddress
            address: 10.11.12.13/24
            addressType: STATIC
            enableSSH: true
        1:
            type: InterfaceAddress
            address: 10.43.70.135/16
            addressType: DHCP
            enableSSH: true
    mtu: 1500
```

```
delphix network interface 'ens160' update *> unset addresses.0
delphix network interface 'ens160' update *> get
    type: NetworkInterface
    name: ens160
    addresses:
        0:
            type: InterfaceAddress (*)
            address: 10.43.70.135/16 (*)
            addressType: DHCP (*)
            enableSSH: true (*)
    mtu: 1500
```

3. `Commit` the operation.

```
delphix network interface 'ens160' update *> commit
delphix network interface 'ens160'> get
    type: NetworkInterface
    name: ens160
    addresses:
        0:
            type: InterfaceAddress
            address: 10.43.70.135/16
            addressType: DHCP
            enableSSH: true
            sessionInUse: false
            state: OK
    device: ens160
    macAddress: 02:dc:02:00:7f:06
    mtu: 1500
    mtuRange: 68-9000
    reference: NETWORK_INTERFACE-ens160
    state: OK
```

13.2.4.2.4 CLI cookbook: adding a static route

This topic describes how to add a static route.

13.2.4.2.4.1 Procedure

1. Log in to the Delphix Engine as the sysadmin user and switch to the network route context.

```
delphix network route> list
DESTINATION      GATEWAY       OUTINTERFACE  PROTOCOL
default          172.16.0.1    ens192        DHCP
10.1.2.0/24      -             ens224        KERNEL
```

```
172.16.0.0/24     -              ens192         KERNEL
```

2. Run the `create` command to add a new route.

```
delphix network route> create
delphix network route create *> set destination=192.168.11.0/24
delphix network route create *> set gateway=10.1.2.1
delphix network route create *> get
    type: NetworkRoute
    destination: 192.168.11.0/24 (*)
    gateway: 10.1.2.1 (*)
    outInterface: (unset)
```

3. Optional outInterface Property Setting the `outInterface` property is optional, as the system will automatically determine the output interface based on the gateway address provided, as shown below.
4. `Commit` the operation.

```
delphix network route create *> commit
delphix network route> list
DESTINATION       GATEWAY        OUTINTERFACE   PROTOCOL
default           172.16.0.1     ens192         DHCP
10.1.2.0/24       -              ens224         KERNEL
172.16.0.0/24     -              ens192         KERNEL
192.168.11.0/24   10.1.2.1       ens224         STATIC
```

13.2.4.2.5 CLI cookbook: changing the default group name

This topic describes how to change the name of the default group <New Group> on the Delphix Engine as a simple example of CLI interactions. You must have delphix_admin credentials to perform this procedure.

13.2.4.2.5.1 Procedure

1. Switch to the group context and list groups on the system.

```
delphix> group
delphix group> list
NAME          DESCRIPTION
<New Group>   -
```

2. Select the default group and show current properties.

```
delphix group> select ""
delphix group ""> get
    name: <New Group>
    type: Group
    description: (unset)
    reference: GROUP-1
```

3. Run the `update` command and set the name.

```
delphix group ""> update
delphix group "" update *> set name=default
delphix group "" update *> get
    name: default (*)
    type: Group
    description: (unset)
    reference: GROUP-1
```

4. `Commit` the operation.

```
delphix group "" update *> commit
delphix group "default">
```

13.2.4.2.6 CLI cookbook: how to change a Delphix user password

> The default domain user created on Delphix Engines is now admin instead of delphix_admin. When engines created before 5.3.1 are upgraded to 5.3.1 or later they will retain their old username 'delphix_admin'. To avoid complications Delphix recommends creating users with an admin role and then Disabling delphix_admin.

1. ssh into your engine with a user that has Admin privileges.

```
ssh admin@delphix
```

2. Go to Users and select the User you would like to change the password of.

```
delphix > user
delphix user > ls
delphix user > select example_user
delphix user "example_user" > ls
```

3. Select `updateCredential` to allow you to change the password and set a new password.

```
delphix user "example user" > updateCredential
delphix user "example_user" updateCredential *> set newCredential.password=<new
 password>
```

4. Commit the operation.

```
delphix user "example_user" updateCredential *> commit
```

Example:

```
ssh admin@delphixengine
delphixengine > user
delphixengine user > ls

Objects

NAME           EMAILADDRESS

sysadmin       -

admin  no@delphix.com

test_user      no@delphix.com


Operations

create

current

delphixengine user > select test_user
delphixengine user "test_user" > ls

Properties

    type: User

    name: test_user

    authenticationType: NATIVE

    credential:

        type: PasswordCredential
```

```
        password: ********

    emailAddress: no@delphix.com

    enabled: true

    firstName: (unset)

    homePhoneNumber: (unset)

    isDefault: true

    lastName: (unset)

    locale: en_US

    mobilePhoneNumber: (unset)

    passwordUpdateRequested: false

    principal: test_user

    publicKey: (unset)

    reference: USER-2

    sessionTimeout: 30min

    userType: DOMAIN

    workPhoneNumber: (unset)



Operations

delete

update

disable

enable

updateCredential

delphixengine user "test_user" > updateCredential
delphixengine user "test_user" updateCredential *> set newCredential.password=<n
ew password>
delphixengine user "test_user" update *> commit
```

13.2.4.2.7 CLI cookbook: retrieve capacity information

This topic describes how to gather capacity information from your Delphix Engine. This information includes:

- dSource Space Breakdown
- Virtual Object Space Breakdown
- Total Space

13.2.4.2.7.1 Procedure

1. Switch to the capacity system context.

```
delphix> capacity system
```

2. List the properties of this content.

```
delphix capacity system> ls
Properties
type: CurrentSystemCapacityData
source:
type: CapacityBreakdown
activeSpace: 940582400B
actualSpace: 1075381760B
descendantSpace: 134583808B
logSpace: 145920B
manualSpace: 0B
policySpace: 0B
syncSpace: 134583808B
timeflowUnvirtualizedSpace: 7725215744B
unvirtualizedSpace: 2624235520B
timestamp: 2015-12-11T11:49:18.998Z
totalSpace: 25568477184B
virtual:
type: CapacityBreakdown
activeSpace: 176684032B
actualSpace: 313768448B
descendantSpace: 0B
logSpace: 47820288B
manualSpace: 85958144B
policySpace: 0B
syncSpace: 85958144B
timeflowUnvirtualizedSpace: 5475587584B
unvirtualizedSpace: 2667149312B
```

For more information about capacity management in Delphix, visit An Overview of Capacity and Performance Information (see page 748)

13.2.4.2.8 CLI cookbook: view storage test results

1. Log into the CLI as the sysadmin.

```
ssh sysadmin@yourengine
```

2. Navigate to storage test.

```
delphix > storage
delphix storage > test
```

3. List your tests and select the one that you would like to view and get the results.

```
delphix storage test > ls
delphix storage test > select STORAGE_TEST-X
delphix storage test STORAGE_TEST-X > result
delphix storage test STORAGE_TEST-X result *> commit
```

13.2.4.2.9 CLI cookbook: how to change IP address of Delphix Engine

This topic describes how to change an IP address on the Delphix Engine.

13.2.4.2.9.1 Procedure

1. Stop all running VDBs by clicking the **Stop** button on the VDB card.
2. Disable all dSources.
3. Log into the Delphix Engine as **sysadmin** user and switch to the network interface context. Then use the list command to view the available network interfaces, and select the public interface to be changed.

```
delphix> network
delphix network> setup
delphix network interface> list
NAME
vmxnet3s0
delphix network interface> select vmxnet3s0
delphix network interface 'vmxnet3s0'> get
    type: NetworkInterface
    name: vmxnet3s0
    addresses:
        0:
```

```
            type: InterfaceAddress
            address: 10.1.2.3/24
            addressType: STATIC
            enableSSH: true
            state: OK
    dataNode: DATA_NODE-34
    device: vmxnet3s0
    macAddress: 0:c:29:32:96:a3
    mtu: 1500
    mtuRange: 60-9000
    reference: NETWORK_INTERFACE-vmxnet3s0-DATA_NODE-34
    state: OK
```

4. Run the update command and update the address to the new IP address.

```
delphix network interface 'vmxnet3s0'> update
delphix network interface 'vmxnet3s0' update *> edit addresses.0
delphix network interface 'vmxnet3s0' update addresses.0 *> get
Properties
    type: InterfaceAddress
    address: 172.16.151.154/24
    addressType: STATIC
    enableSSH: true

delphix network interface 'vmxnet3s0' update addresses.0 *> set address=10.1.2.
4/24
delphix network interface 'vmxnet3s0' update addresses.0 *> get
    type: InterfaceAddress (*)
    address: 10.1.2.4/24 (*)
    addressType: STATIC (*)
    enableSSH: true (*)
```

5. Commit the operation.

```
delphix network interface 'vmxnet3s0' update addresses.0 *> commit
delphix network interface 'vmxnet3s0'> get
    type: NetworkInterface
    name: vmxnet3s0
    addresses:
        0:
            type: InterfaceAddress
            address: 10.1.2.4/24
            addressType: STATIC
            enableSSH: true
            state: OK
    dataNode: DATA_NODE-34
    device: vmxnet3s0
    macAddress: 0:c:29:32:96:a3
    mtu: 1500
```

```
    mtuRange: 60-9000
    reference: NETWORK_INTERFACE-vmxnet3s0-DATA_NODE-34
    state: OK
```

6. Re-enable the VDBs and dSources running from the engine.

13.2.4.2.10 CLI cookbook: about alert notifications

The Delphix Engine can send out email notifications when alerts happen. Alert profiles control this functionality.

An alert profile is composed of two things:

- **Filter specification:** A filter, or combination of filters, that specifies which alerts are of interest.
- **Alert action:** This specifies the email addresses to which the Delphix Engine will send an email when an alert matches the filter specification.

By default, the Delphix Engine has a single alert profile configured with the following parameters:

- Filter Specification: Match any alert with a severity level of **CRITICAL** or **WARNING**.
- Alert Actions: Send an email to the address defined for user **admin**.

> The default domain user created on Delphix Engines is now admin instead of delphix_admin. When engines created before 5.3.1 are upgraded to 5.3.1 or later they will retain their old username 'delphix_admin'. To avoid complications Delphix recommends creating users with an admin role and then Disabling delphix_admin.

Using the CLI, it is possible to:

- Modify the system default alert profile
- Create additional profiles in addition to the default one
- Set multiple actions for a single profile, such as "email dephix_admin" *and* "email user1@mycompany.com"

13.2.4.2.10.1 Simple filters

- Filtered by Owner of alerts target – for example, objects owned by user 1

13.2.4.2.10.2 **Complex filters**

Complex filters combine/modify other sub-filters:

- “And” filter – Used when all conditions defined must be met for the filter to notify the user with an email
- “Or” filter – Used when either one or the other of the conditions defined in the filters must be met for the filter to notify the user with an email

- “Not” filter – Used to exclude items

13.2.4.2.10.3 Limitations

- This is a CLI feature.
- Alert Profiles do not override permission settings. If you do not have Read permission on an object then your alert profile will never get triggered for that object's alerts, regardless of your filter settings.

The following CLI examples will run through how to create these three filters. Each example provides three different methods of setting up a profile. These include the following:

- A simple profile
- A profile with two filters
- A complicated profile

For more information, see CLI Cookbook: creating alert profiles (see page 2571)

13.2.4.2.10.4 A simple profile

A simple profile approach matches the Delphix out-of-the-box default alert profiles. To create a simple alert profile using the CLI as seen in the figure below, go into the alert profile section of the command-line interface (CLI) and create a new profile. Line four prompts the engine to send an email when the filters are triggered. The following three command lines refer to the filter specifications. Follow two severity levels: warning and critical. This will trigger an email alert when any warning or critical events occur.

```
twalsh-trunk.dcenter> cd alert
twalsh-trunk.dcenter alert> cd profile
twalsh-trunk.dcenter alert profile> create
twalsh-trunk.dcenter alert profile create *> set actions.0.type=AlertActionEmailUser
twalsh-trunk.dcenter alert profile create *> set filterSpec.type=SeverityFilter
twalsh-trunk.dcenter alert profile create *> set filterSpec.severityLevels.0=CRITICAL
twalsh-trunk.dcenter alert profile create *> set filterSpec.severityLevels.1=WARNING
twalsh-trunk.dcenter alert profile create *> commit
```

13.2.4.2.10.5 A compound alert profile

Creating a compound alert profile in the CLI will combine two filters together. This profile triggers an email about any alert on objects owned by the delphix_admin plus any other alert that is critical. The compound alert profile creates two filters. The first one will be the target owner filter, which in this case is **admin**. The second filter is the severity filter, allowing users to match anything that is critical. Combine these two filters using the OR logic so that if any of the sub-filters match, the whole filter matches. An example of this can be seen in the figure below.

Alert profile using OR logic

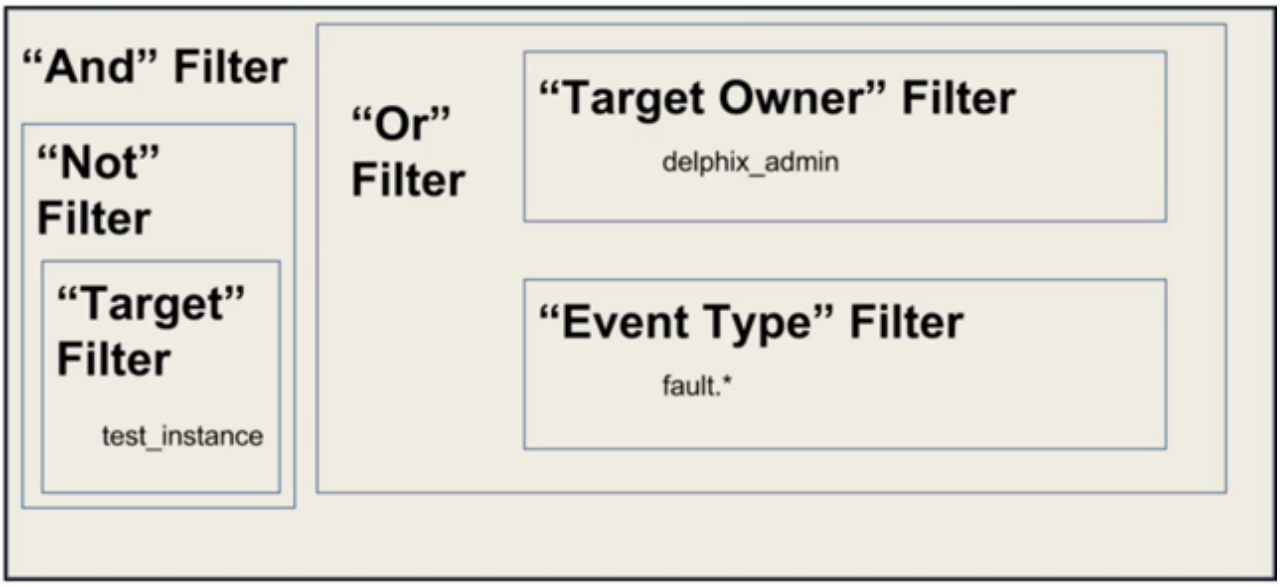
While working in the CLI, the first four lines describe a simple alert profile. The distinction between simple and compound alert profiles is that in a compound profile, the top-level filter uses an OR filter with sub-filters for target owner and severity level, as seen in line five of the figure below.

```
twalsh-trunk.dcenter> cd alert
twalsh-trunk.dcenter alert> cd profile
twalsh-trunk.dcenter alert profile> create
twalsh-trunk.dcenter alert profile create *> set actions.0.type=AlertActionEmailUser
twalsh-trunk.dcenter alert profile create *> set filterSpec.type=OrFilter
twalsh-trunk.dcenter alert profile create *> set filterSpec.subFilters.0.type=TargetO
wnerFilter
twalsh-trunk.dcenter alert profile create *> set filterSpec.subFilters.0.owners.0=del
phix_admin
twalsh-trunk.dcenter alert profile create *> set filterSpec.subFilters.1.type=Severit
yFilter
twalsh-trunk.dcenter alert profile create *> set filterSpec.subFilters.1.severityLeve
ls.0=CRITICAL
twalsh-trunk.dcenter alert profile create *> commit
```

13.2.4.2.10.6 Complex alert profile

A complex alert profile uses the profile filter created in the compound alert profile and modifies it. For the example shown in the figure below, you have a VDB named test_instance that you do not need any emails about. The following commands will create an effective filter.

1. Create an OR filter with two sub filters: target owner and event type.
2. Create a NOT filter which will exclude the VDB (test_instance) from which you do not want to receive notifications.
3. Use the AND logic to combine all these filters together, as seen below.



Complex alert profile

Below is an example of the command lines used to set up this complex profile.

```
twalsh-trunk.dcenter> cd alert
twalsh-trunk.dcenter alert> cd profile
twalsh-trunk.dcenter alert profile> create
twalsh-trunk.dcenter alert profile create *> set actions.0.type=AlertActionEmailUser
twalsh-trunk.dcenter alert profile create *> set filterSpec.type=AndFilter
twalsh-trunk.dcenter alert profile create *> set filterSpec.subFilters.0.type=NotFilter
twalsh-trunk.dcenter alert profile create *> edit filterSpec.subFilters.0.subFilter
twalsh-trunk.dcenter alert profile create filterSpec.subFilters.0.subFilter *> set type=TargetFilter
twalsh-trunk.dcenter alert profile create filterSpec.subFilters.0.subFilter *> set targets.0=test_instance
twalsh-trunk.dcenter alert profile create filterSpec.subFilters.0.subFilter *> back
twalsh-trunk.dcenter alert profile create *> set filterSpec.subFilters.1.type=OrFilter
twalsh-trunk.dcenter alert profile create *> set filterSpec.subFilters.1.subFilters.0.type=TargetOwnerFilter
twalsh-trunk.dcenter alert profile create *> set filterSpec.subFilters.1.subFilters.0.owners.0=delphix_admin
twalsh-trunk.dcenter alert profile create *> set filterSpec.subFilters.1.subFilters.1.type=SeverityFilter
twalsh-trunk.dcenter alert profile create *> set filterSpec.subFilters.1.subFilters.1.severityLevels=CRITICAL
twalsh-trunk.dcenter alert profile create *> commit
```

Complex alert profile CLI

13.2.4.2.10.7 CLI cookbook: creating alert profiles

This article describes how to create alert profiles.

Delphix generates alerts for different events. Users may want to be notified of events based on certain criteria such as the type of event or severity. An alert profile allows a user or group of users to be notified of the desired event.

The default domain user created on Delphix Engines is now admin instead of delphix_admin. When engines created before 5.3.1 are upgraded to 5.3.1 or later they will retain their old username 'delphix_admin'. To avoid complications Delphix recommends creating users with an admin role and then Disabling delphix_admin.

Procedure

1. ssh into your engine using your delphix_admin username and password.

```
ssh admin@yourdelphixengine
```

2. Go into your alerts and list the alerts you already have.

```
delphix > alert
delphix alert > ls
```

3. Create your profile.

```
delphix alert > profile
delphix alert profile > create
delphix alert profile create * > ls
```

4. Set Actions, filterSpec, and User.

1. **Warning:**
   Valid Values for Parameters

   actions.0.type:
   a. AlertActionEmailList: This type of alert is used to create an alert for any number of users. When this type is selected, an email address may be defined in each element of the "actions.0.addresses" array as illustrated above.
   b. AlertActionEmailUser: This type of alert is created for the email address of the user currently logged into the command-line interface. The "actions.0.addresses" array is not available for this type.

5. filterSpec.type:
   a. AndFilter
   b. EventFilter

c. NotFilter
d. OrFilter
e. SeverityFilter
f. TargetFilter
g. TargetOwnerFilter

```
delphix alert profile create *> set actions.0.type=<AlertActionEmailList or
AlertActionEmailUser>
delphix alert profile create *> set actions.0.addresses.0=<email address to send to>
delphix alert profile create *> set actions.0.addresses.1=<additional email address>
delphix alert profile create *> set actions.0.addresses.2=<additional email address>
delphix alert profile create *> ls

delphix alert profile create *> set filterSpec.type=SeverityFilter
delphix alert profile create *> set filterSpec.severityLevels=<AUDIT|WARNING|
CRITICAL|INFORMATIONAL>
```

5. Commit your changes.

```
delphix alert profile create *> commit
```

Example

```
ssh admin@yourengine
delphix > alert
delphix alert> ls
Objects
REFERENCE  TIMESTAMP                 TARGETNAME
EVENTTITLE
ALERT-102  2015-01-14T21:00:04.380Z  ASE/pubs2
Job complete
ALERT-101  2015-01-14T20:55:57.880Z  ASE/pubs2VDB
Job complete
ALERT-100  2015-01-14T19:35:32.958Z  ASE/pubs2VDB
Job complete
ALERT-99   2015-01-14T19:35:32.850Z  ASE/pubs2VDB
Job complete
ALERT-98   2015-01-14T19:34:58.744Z  ASE/pubs2
Error during job execution
ALERT-97   2015-01-14T18:12:01.928Z  ASE/pubs2
Job complete
ALERT-96   2015-01-14T18:03:10.664Z  ASE/pubs2
Job complete
ALERT-95   2015-01-14T17:16:07.464Z  ASE/pubs2
Job complete
ALERT-94   2015-01-14T17:15:55.298Z  ASE/market
Job complete
```

```
ALERT-93   2015-01-14T17:15:45.995Z  ASE/pubs2VDB
Job complete
ALERT-92   2015-01-14T16:39:33.133Z  nstacksolase2.acme.com-2015-01-14T16:39:13.821Z
Job complete
ALERT-91   2015-01-14T16:38:33.719Z  nstacksolase2.acme.com
Job complete
ALERT-90   2015-01-14T15:47:35.005Z  market
Validated sync failed for dSource
ALERT-89   2015-01-14T15:45:40.895Z  pubs2
Validated sync failed for dSource
ALERT-88   2015-01-14T15:02:14.874Z  ASE/market
Job complete
ALERT-87   2015-01-14T11:33:28.766Z  ASE/pubs2VDB
Job complete
ALERT-86   2015-01-13T23:11:46.838Z  ASE/market
Job complete
ALERT-85   2015-01-13T11:30:01.154Z  ASE/pubs2VDB
Job complete
ALERT-84   2015-01-13T11:07:04.385Z  pubs2
Backup detection failed
ALERT-83   2015-01-12T22:35:18.774Z  pubs2
Backup detection failed
ALERT-82   2015-01-12T11:30:00.063Z  ASE/pubs2VDB
Unable to connect to remote database during virtual database policy enforcement
ALERT-81   2015-01-12T11:30:00.054Z  ASE/pubs2
Unable to connect to remote database during dSource policy enforcement
ALERT-80   2015-01-12T08:38:26.983Z  pubs2
Backup detection failed
ALERT-79   2015-01-12T06:04:34.666Z  pubs2
Validated sync failed for dSource
ALERT-78   2015-01-11T11:30:03.393Z  ASE/pubs2VDB
Job complete
Children
profile
delphix alert> select ALERT-98
delphix alert "ALERT-98"> ls
Properties
    type: Alert
    event: alert.jobs.failed.object
    eventAction: Create the database on the target host.
    eventDescription: DB_EXPORT job for "ASE/pubs2" failed due to an error during
execution: Could not find database "pubs2VDB" on target instance "SRC_157_4K",
environment "ASE".
    eventSeverity: CRITICAL
    eventTitle: Error during job execution
    reference: ALERT-98
    target: ASE/pubs2
    targetName: ASE/pubs2
    targetObjectType: ASEDBContainer
    timestamp: 2015-01-14T19:34:58.744Z
delphix alert> profile
delphix alert profile> select ALERT_PROFILE-1
delphix alert profile "ALERT_PROFILE-1"> ls
```

```
Properties
    type: AlertProfile
    actions:
        0:
            type: AlertActionEmailList
            addresses: sys_admin@acme.com
            format: HTML
    filterSpec:
        type: SeverityFilter
        severityLevels: CRITICAL,WARNING
    reference: ALERT_PROFILE-1
    user: admin
Operations
delete
update
delphix alert profile> create
delphix alert profile create *> set actions.0.type=AlertActionEmailList
delphix alert profile create *> set actions.0.addresses.0=johndoe@acme.com
delphix alert profile create *> set actions.0.addresses.1=samsmith@acme.com
delphix alert profile create *> set filterSpec.type=SeverityFilter
delphix alert profile create *> set filterSpec.severityLevels=INFORMATIONAL
```

*The last piece of the alert profile that needs to be configured is the "targetFilter". This is an array so you can define multiple targets. In the following example, there is a dSource named "pubs2" the user wants to define an alert on. If they try to set the filter to just the name of the dSource itself ("pubs2"), it will warn them that this is ambiguous and gives a hint on how to fully qualify it:

```
delphix > alert profile create
delphix alert profile create *> ls
Properties
 type: AlertProfile
 actions:
 0:
 type: AlertActionEmailList (*)
 addresses: foo@bar.com (*)
 format: HTML (*)
 filterSpec: (unset)
delphix alert profile create *> edit actions
delphix alert profile create actions *> add
delphix alert profile create actions 0 *> set addresses=foo@bar.com
delphix alert profile create actions 0 *> back
delphix alert profile create actions *> back
delphix alert profile create *> set filterSpec.type=TargetFilter
delphix alert profile create *> set filterSpec.targets="REPLICATION_SPEC_EXECUTE"
delphix alert profile create *> commit
```

Use the tab button freely to autocomplete and also see available options, for instance, while changing the severityLevels property, you can use the tab key like so:

```
DELPHIX-4221.dcenter alert profile 'ALERT_PROFILE-1' update *> set
filterSpec.severityLevels= <I HIT TAB HERE TO SEE OPTIONS BELOW>
```

```
AUDIT          CRITICAL       INFORMATIONAL  WARNING
```

Please note that the above tab autocomplete feature will only work if the filterSpec.type is already set to SeverityFilter.

Note on names used in the example

SRC_157_4K: Repository (entity containing the database instances)

ASE: Group name

pubs2: Name of an individual database instance

The user sets the target filter to be equal to "pubs2/pubs2" and "ASE/pubs2" because if you review the "TARGETNAME" column from step 1 above, you will see alerts generated for both of these targets.

13.2.4.2.11 CLI cookbook: obtaining CPU performance information using CLI

There are times when it may be desirable to obtain analytics information about the Delphix Engine CPU, which is formatted differently than what is available from the Delphix User Interface (UI).

13.2.4.2.11.1 Troubleshooting

You can obtain running information by looking at the **GUI > Resources > Performance Analytics**. You will see the graph for CPU if the CPU is selected.

When logged into the Delphix Engine command-line interface (CLI) as delphix_admin, you can see the default analytics gathered:

```
dlpx5120.dcenter> analytics
dlpx5120.dcenter analytics> list
NAME STATISTICTYPE STATE COLLECTIONINTERVAL COLLECTIONAXES
default.cpu CPU_UTIL RUNNING 1sec idle,kernel,user
default.disk DISK_OPS RUNNING 1sec op,avgLatency,latency,count,throughput
default.iscsi iSCSI_OPS RUNNING 1sec op,latency,count,throughput
default.network NETWORK_INTERFACE_UTIL RUNNING 1sec
outBytes,networkInterface,inPackets,inBytes,outPackets
default.nfs NFS_OPS RUNNING 1sec op,latency,count,throughput
default.tcp TCP_STATS RUNNING 1sec
congestionWindowSize,localPort,remotePort,receiveWindowSize,inUnorderedBytes,s
endWindowSize,retransmittedBytes,outBytes,localAddress,roundTripTime,inBytes,u
nacknowledgedBytes,remoteAddress
```

13.2.4.2.11.2 Resolution

You can also obtain some information from the CLI, using the Performance Analytics Tool API Reference (see page 876). (see page 2576)

In order to view this information, you must log in with a user that had Delphix Admin privileges.

Option 1: Running an Existing Analytic

1.Login as admin and run `analytics.`

```
dlpx5120.dcenter> analytics
dlpx5120.dcenter analytics> list
NAME STATISTICTYPE STATE COLLECTIONINTERVAL COLLECTIONAXES
default.cpu CPU_UTIL RUNNING 1sec idle,kernel,user
default.disk DISK_OPS RUNNING 1sec op,avgLatency,latency,count,throughput
default.iscsi iSCSI_OPS RUNNING 1sec op,latency,count,throughput
default.network NETWORK_INTERFACE_UTIL RUNNING 1sec
outBytes,networkInterface,inPackets,inBytes,outPackets
default.nfs NFS_OPS RUNNING 1sec op,latency,count,throughput
default.tcp TCP_STATS RUNNING 1sec
congestionWindowSize,localPort,remotePort,receiveWindowSize,inUnorderedBytes,s
endWindowSize,retransmittedBytes,outBytes,localAddress,roundTripTime,inBytes,u
nacknowledgedBytes,remoteAddress

Children
statistic

Operations
create
```

2. To see the available properties for CPU analytics, `select` and list the `default.cpu.`

```
dlpx5120.dcenter analytics> select default.cpu
dlpx5120.dcenter analytics 'default.cpu'> ls
Properties
 type: StatisticSlice
 name: default.cpu
 axisConstraints: (empty)
 collectionAxes: idle,kernel,user
 collectionInterval: 1sec
 dataNode: DATA_NODE-1
 reference: ANALYTICS_STATISTIC_SLICE-1
 state: RUNNING
 statisticType: CPU_UTIL

Operations
delete
getData
pause
rememberRange
resume
stopRememberingRange
```

3. Specify `setopt trace=true` in order to see the `CpuUtilDatapointStream` datapoints.

```
dlpx5120.dcenter analytics 'default.cpu'> setopt trace=true

=== GET /resources/json/delphix/analytics/ANALYTICS_STATISTIC_SLICE-1 ===
=== RESPONSE ===
{
 "type": "OKResult",
 "status": "OK",
 "result": {
 "type": "StatisticSlice",
 "reference": "ANALYTICS_STATISTIC_SLICE-1",
 "namespace": null,
 "name": "default.cpu",
 "statisticType": "CPU_UTIL",
 "collectionInterval": 1,
 "state": "RUNNING",
 "collectionAxes": [
 "idle",
 "kernel",
 "user"
 ],
 "axisConstraints": [],
 "dataNode": "DATA_NODE-1"
 },
 "job": null,
 "action": null
}
=== END ===
```

4. Enter `getData` and `commit` to obtain the data gathered.

```
dlpx5120.dcenter analytics 'default.cpu'> getData
=== GET /resources/json/delphix/analytics/ANALYTICS_STATISTIC_SLICE-1 ===
=== RESPONSE ===
{
 "type": "OKResult",
 "status": "OK",
```

```
 "result": {
 "type": "StatisticSlice",
 "reference": "ANALYTICS_STATISTIC_SLICE-1",
 "namespace": null,
 "name": "default.cpu",
 "statisticType": "CPU_UTIL",
 "collectionInterval": 1,
 "state": "RUNNING",
 "collectionAxes": [
 "idle",
 "kernel",
 "user"
 ],
 "axisConstraints": [],
 "dataNode": "DATA_NODE-1"
 },
 "job": null,
 "action": null
}
=== END ===

dlpx5120.dcenter analytics 'default.cpu' getData *> commit
=== GET /resources/json/delphix/analytics/ANALYTICS_STATISTIC_SLICE-1/getData
===
=== RESPONSE ===
{
 "type": "OKResult",
 "status": "OK",
 "result": {
 "type": "DatapointSet",
 "resolution": 1,
 "datapointStreams": [
 {
 "type": "CpuUtilDatapointStream",
 "datapoints": [
 {
 "type": "CpuUtilDatapoint",
```

```
"timestamp": "2016-12-06T13:53:30.000Z",
"idle": 1946,
"kernel": 33,
"user": 19,
"dtrace": null
},
{
"type": "CpuUtilDatapoint",
"timestamp": "2016-12-06T13:53:31.000Z",
"idle": 1966,
"kernel": 17,
"user": 14,
"dtrace": null
},
{
"type": "CpuUtilDatapoint",
"timestamp": "2016-12-06T13:53:32.000Z",
"idle": 1968,
"kernel": 17,
"user": 13,
"dtrace": null
},
{
"type": "CpuUtilDatapoint",
"timestamp": "2016-12-06T13:53:33.000Z",
"idle": 1963,
"kernel": 19,
"user": 17,
"dtrace": null
},
{
"type": "CpuUtilDatapoint",
"timestamp": "2016-12-06T13:53:34.000Z",
"idle": 1968,
"kernel": 16,
"user": 15,
"dtrace": null
```

```
 },
...
...
 {
 "type": "CpuUtilDatapoint",
 "timestamp": "2016-12-06T19:54:16.000Z",
 "idle": 1922,
 "kernel": 36,
 "user": 41,
 "dtrace": null
 },
 {
 "type": "CpuUtilDatapoint",
 "timestamp": "2016-12-06T19:54:17.000Z",
 "idle": 1953,
 "kernel": 26,
 "user": 20,
 "dtrace": null
 }
 ],
 "cpu": null
 }
 ],
 "overflow": false
 },
 "job": null,
 "action": null
}
=== END ===
 type: DatapointSet
 datapointStreams:
 0:
 type: CpuUtilDatapointStream
 datapoints: [ ... ]
 overflow: false
 resolution: 1sec
```

Option 2: Creating a New Analytic

1. Log in as admin and then run `analytics`
2. Run `create` and `list` the properties.

```
dlpx5120.dcenter analytics> create
dlpx5120.dcenter analytics create *> ls
Properties
 type: StatisticSlice
 name: (required)
 axisConstraints: (unset)
 collectionAxes: (required)
 collectionInterval: (unset)
 dataNode: (unset)
 statisticType: (required)
```

3. Set the required Properties `name`, `collectionAxes` and statisticType (if you are unsure you can run help for options) for=""> for options)>

```
set name=test.cpu
set collectionAxes=kernel
set statisticType=CPU_UTIL

dlpx5120.dcenter analytics create *> ls
Properties
 type: StatisticSlice
 name: test.cpu
 axisConstraints: (unset)
 collectionAxes: kernel
 collectionInterval: (unset)
 dataNode: (unset)
 statisticType: CPU_UTIL
```

4. `Commit` to being data collection.

```
dlpx5120.dcenter analytics create *> commit


=== POST /resources/json/delphix/analytics ===
{
 "type": "StatisticSlice",
 "name": "test.cpu",
 "collectionAxes": [
 "kernel"
 ],
 "statisticType": "CPU_UTIL"
}
=== RESPONSE ===
{
 "type": "OKResult",
 "status": "OK",
 "result": "ANALYTICS_STATISTIC_SLICE-7",
 "job": null,
 "action": "ACTION-656"
}
=== END ===
 `ANALYTICS_STATISTIC_SLICE-7
```

5. Use `ls` to see the new `test.cpu` analytic

```
dlpx5120.dcenter analytics> ls

NAME STATISTICTYPE STATE COLLECTIONINTERVAL COLLECTIONAXES
default.cpu CPU_UTIL RUNNING 1sec idle,kernel,user
test.cpu CPU_UTIL RUNNING 1sec kernel
default.disk DISK_OPS RUNNING 1sec op,avgLatency,latency,count,throughput
default.iscsi iSCSI_OPS RUNNING 1sec op,latency,count,throughput
default.network NETWORK_INTERFACE_UTIL RUNNING 1sec
outBytes,networkInterface,inPackets,inBytes,outPackets
default.nfs NFS_OPS RUNNING 1sec op,latency,count,throughput
default.tcp TCP_STATS RUNNING 1sec
congestionWindowSize,localPort,remotePort,receiveWindowSize,inUnorderedBytes,s
endWindowSize,retransmittedBytes,outBytes,localAddress,roundTripTime,inBytes,u
nacknowledgedBytes,remoteAddress

Children
statistic

Operations
create
```

6. Select `test.cpu` and use `getData` and `commit` to see the data gathered.

```
dlpx5120.dcenter analytics> select test.cpu
dlpx5120.dcenter analytics  'test.cpu'> ls
dlpx5120.dcenter analytics  'test.cpu'> getData
dlpx5120.dcenter analytics  'test.cpu' getData *> commit
```

7. Use `delete` to stop the collection and delete `test.cpu`

```
dlpx5120.dcenter analytics 'test.cpu'> delete
dlpx5120.dcenter analytics 'test.cpu' delete *> commit
=== POST /resources/json/delphix/analytics/ANALYTICS_STATISTIC_SLICE-7/
delete ===
{}
=== RESPONSE ===
{
 "type": "OKResult",
 "status": "OK",
 "result": "",
 "job": null,
 "action": "ACTION-657"
}
=== END ===

dlpx5120.dcenter analytics> ls

NAME STATISTICTYPE STATE COLLECTIONINTERVAL COLLECTIONAXES
default.cpu CPU_UTIL RUNNING 1sec idle,kernel,user
default.disk DISK_OPS RUNNING 1sec op,avgLatency,latency,count,throughput
default.iscsi iSCSI_OPS RUNNING 1sec op,latency,count,throughput
default.network NETWORK_INTERFACE_UTIL RUNNING 1sec
outBytes,networkInterface,inPackets,inBytes,outPackets
default.nfs NFS_OPS RUNNING 1sec op,latency,count,throughput
default.tcp TCP_STATS RUNNING 1sec
congestionWindowSize,localPort,remotePort,receiveWindowSize,inUnorderedBytes,s
endWindowSize,retransmittedBytes,outBytes,localAddress,roundTripTime,inBytes,u
nacknowledgedBytes,remoteAddress

Children
statistic

Operations
create
dlpx5120.dcenter analytics>
```

13.2.4.2.12 CLI cookbook: rebooting the Delphix engine via CLI

Occasionally, the management stack will hang on the GUI, or routine maintenance will require a reboot of the Delphix Engine. You can do this safely through either the Delphix Setup or CLI. Before performing a reboot, it is important that all your VDBs are shut down and dSources disabled in order to maintain data integrity.

13.2.4.2.12.1 Prerequisites

- Shut down all VDBS
- Shut down dSources

13.2.4.2.12.2 Procedure

Complete the following steps to reboot the Delphix Engine via CLI:

1. Login to the CLI using the sysadmin **username** and **password**.

```
ssh sysadmin@yourdelphixengine
```

2. Go to **system > reboot**.

```
delphix > system
delphix system > reboot
```

3. Commit the action.

```
delphix system reboot *> commit
```

13.2.4.2.13 CLI cookbook: disabling user-click analytics

The Delphix User-click Analytics feature is a lightweight method to capture how users interact with Delphix product user interfaces. The goal of capturing this data is to get a better understanding of product usage, engagement, and user behavior, and to use this data to improve Delphix products and customer experience. This feature is enabled by default for customers deploying on or upgrading to this version. User-click Analytics may also be disabled via the UI.

> This procedure will disable user-click analytics on both the Delphix Engine and Delphix Self-Service.

13.2.4.2.13.1 Procedure to disable user-click analytics

1. ssh into your engine with a user that has Admin privileges.

```
ssh sysadmin@delphix
```

2. Go to Services and select the userInterface.

```
delphix > cd service
delphix service > cd userInterface
```

3. Update the analyticsEnabled option to false.

```
delphix service userInterface > ls
Properties
 Type userInterfaceConfig
 analyticsenabled: true
Operations
delphix service userInterface > update
delphix service userInterface update *> set analyticsEnabled=false
delphix service userInterface update *> commit
```

13.2.4.2.14 CLI cookbook: changing the API version

This topic describes how to change the API version.

13.2.4.2.14.1 Procedure

1. Log in to the Delphix Engine as the admin user.

```
delphix> su - admin@DOMAIN
```

2. Run the version command.

```
delphix> version
1.10.3 # This lists the current API verison.
delphix> version <version you want to set to>
delphix> version
1.8.2 # Displays the new version that you had set above.
```

13.2.4.3 CLI cookbook: hosts and environments

This section contains the following topics:

- CLI cookbook: adding a UNIX host (see page 2590)
- CLI cookbook: adding a SQL Server source environment (see page 2592)
- CLI cookbook: setting multiple addresses for a target host (see page 2592)
- CLI cookbook: how to change environment user (see page 2593)
- CLI cookbook: how to create or edit a privilege elevation profile and profile scripts (see page 2594)
- CLI cookbooks: enabling and configuring environment permissions (see page 2597)

13.2.4.3.1 CLI cookbook: adding a UNIX host

This topic describes the process of adding a UNIX host using the 3.0 command-line interface.

Within Delphix, there are both hosts and host environments. A host represents a remote system, but may or may not be a source or target for linking or provisioning. For example, in an Oracle RAC cluster, the cluster environment represents the location of the Oracle installation(s), and while there are hosts within that cluster they are not individually manageable as environments.

13.2.4.3.1.1 Procedure

1. Create a new environment and set the parameter type to be a UNIX host.

```
delphix> environment create
delphix environment create *> set type=HostEnvironmentCreateParameters
delphix environment create *> set hostEnvironment.type=UnixHostEnvironment
delphix environment create *> set hostParameters.type=UnixHostCreateParameters
delphix environment create *> set
primaryUser.credential.type=PasswordCredential
delphix environment create *> get
    type: HostEnvironmentCreateParameters
    hostEnvironment:
        type: UnixHostEnvironment
        name: (unset)
        aseHostEnvironmentParameters: (unset)
        description: (unset)
        logCollectionEnabled: false
    hostParameters:
        type: UnixHostCreateParameters
        host:
            type: UnixHost
            address: (required)
            dspKeystoreAlias: (unset)
            dspKeystorePassword: (unset)
            dspKeystorePath: (unset)
            dspTruststorePassword: (unset)
```

```
            dspTruststorePath: (unset)
            javaHome: (unset)
            nfsAddressList: (unset)
            oracleJdbcKeystorePassword: (unset)
            privilegeElevationProfile: (unset)
            sshPort: 22
            sshVerificationStrategy: (unset)
            toolkitPath: (required)
    logCollectionEnabled: false
    primaryUser:
        type: EnvironmentUser
        name: (unset)
        credential:
            type: PasswordCredential
            password: (required)
        environment: (unset)
        groupId: (unset)
        userId: (unset)
```

2. Set the host address. The name of the environment is derived from the address used, though you can provide a more descriptive name if desired. The address can be DNS names, IP addresses, or a comma-separated list of the above.

```
delphix environment create *> set hostParameters.host.address=192.168.1.2
```

3. Set the toolkit path. This is where Delphix will store temporary binaries used while the host is configured as part of Delphix.

```
delphix environment create *> set hostParameters.host.toolkitPath=/work
```

4. Set the username and password to use when connecting over SSH. This user must have the privileges described in the Delphix Administration Guide. To configure an SSH user, change the credential type to `SystemKeyCredential`.

```
delphix environment create *> set primaryUser.name=oracle
delphix environment create *> set primaryUser.credential.password
Enter primaryUser.credential.password: *******
```

5. Commit the result. The environment discovery process will execute as an asynchronous job. The default behavior is to wait for the result, so progress will be updated until the discovery process is complete or fails.

```
delphix environment create *> commit
    UNIX_HOST_ENVIRONMENT-4
    Dispatched job JOB-39
```

```
    ENVIRONMENT_CREATE_AND_DISCOVER job started for "192.168.1.2".
    ENVIRONMENT_CREATE_AND_DISCOVER job for "192.168.1.2" completed
successfully.
delphix>
```

13.2.4.3.2 CLI cookbook: adding a SQL Server source environment

This topic describes how to add a SQL Server source environment using the command line interface.

Since SQL Server source environments do not have the Delphix Connector running on them, you must use a target environment as a proxy when adding source environments. Delphix uses the connector running on the proxy environment to run commands against the source environment. See Overview of Requirements for SQL Server Environments (see page 2067) for more information.

13.2.4.3.2.1 Procedure

Enter these commands through the command-line interface:

```
/environment; create;set type=HostEnvironmentCreateParameters; set
hostEnvironment.type=WindowsHostEnvironment;set hostEnvironment.name=<Source
environment name>;set hostEnvironment.proxy=<target host name>; set
hostParameters.type=WindowsHostCreateParameters;set
hostParameters.host.type=WindowsHost;set hostParameters.host.address="<Source host IP
address or hostname>"; set primaryUser.name="<domain\username>";set
primaryUser.credential.type=PasswordCredential;set
primaryUser.credential.password=<password>; commit;
```

Example

The CLI commands for adding source host `"mssql_source_1"` using target host `“mssql_target_1”` a s proxy and environment user `“ad\delphix_user”` would be:

```
/environment; create;set type=HostEnvironmentCreateParameters; set
hostEnvironment.type=WindowsHostEnvironment;set hostEnvironment.name="mssql_source_1";
set hostEnvironment.proxy="mssql_target_1"; set
hostParameters.type=WindowsHostCreateParameters;set
hostParameters.host.type=WindowsHost;set hostParameters.host.address="mssql_source_1";
 set primaryUser.name="ad\delphix_user";set
primaryUser.credential.type=PasswordCredential;set
primaryUser.credential.password="i_am_the_password"; commit;
```

13.2.4.3.3 CLI cookbook: setting multiple addresses for a target host

This topic is an example of using arrays to configure a target host to support multiple IP addresses. The `nfsAddressList` property is an array of strings.

13.2.4.3.3.1 Procedure

1. Select the host to update

```
delphix> hostdelphix host> select exampledelphix host "example"> update
```

2. Set the address:

```
Delphix host '192.168.121.141' update *> set nfsAddressList="192.168.1.23,192.1
68.2.44"
```

3. Get the current addresses, both as a string and as an array object.

```
delphix host "example" update *> get nfsAddressList    192.168.1.23,192.168.2.4
4 (*)delphix host "example" update *> get nfsAddressList[0]    192.168.1.23
 (*)delphix host "example" update *> edit nfsAddressListdelphix host "example"
 update addresses *> get    0:        192.168.1.23 (*)    1:        192.168.2.4
4 (*)
```

4. Commit the result:

```
delphix host "example" update addresses *> commitdelphix host "example">
```

13.2.4.3.4 CLI cookbook: how to change environment user

1. ssh into your engine using Admin privileges

```
ssh admin@delphix
```

2. Go to Environment and find the Environment you would like to update

```
delphix > environmentdelphix environment > lsdelphix environment > select
test_env
```

3. Select Environment updating and Update

```
delphix environment "test_env" > updatedelphix environment "test_env" update *>
ls
```

4. Set `primaryUser` to new user you would like to use for the Environment (NOTE: new user should be added as an environment user before making as a primaryUser)

```
delphix environment "test_env update" *> set primaryUser=<new user>
```

5. Commit the operation.

```
delphix environment "test_env" update *> commit
```

Example:

```
ssh admin@delphixdelphix > environmentdelphix environment > ls ObjectsNAME
DESCRIPTIONDemo  ChildrenoracleuserOperationscreate delphix environment > select
Demodelphix environment "Demo" > updatedelphix environment "Demo" update *> ls
Properties    type: UnixHostEnvironment    name: Demo    description:
primaryUser: delphix delphix environment "Demo" update *> set primaryUser=<new
 user>delphix environment "Demo" update *> commit
```

13.2.4.3.5 CLI cookbook: how to create or edit a privilege elevation profile and profile scripts

13.2.4.3.5.1 Background:

If you are running a version prior to 6.0.0 please contact your Professional Services representative.

13.2.4.3.5.2 Procedure for creating an elevation profile:

1. Log into the CLI using delphix_admin or a user with Admin privileges and got to 'Host'.

```
ssh admin@yourengine
yourengine > host
```

2. Select `privilegeElevation` then `profile.`

```
yourengine host > privilegeElevation
yourengine host privilegeElevation > profile
```

3. Set the name of the profile and the version of the profile.

```
yourengine host privilegeElevation profile *> set name=<profilename>
yourengine host privilegeElevation profile *> set version=<profileversion>
```

4. Commit the profile to save it.

```
yourengine host privilegeElevation profile *> commit
```

13.2.4.3.5.3 Procedure for creating a profileScript

Please note that you will need to create the script yourself or with the help of the Professional Services team.

1. Log into the CLI using delphix_admin or a user with Admin privileges and got to 'Host'.

```
ssh delphix_admin@yourengine
yourengine > host
```

2. Select `privilegeElevation` then `profileScript.`

```
yourengine host > privilegeElevation
yourengine host privilegeElevation > profileScript
```

3. Create your script by setting, name, contents, and profile (this can be your previously created profile or the default sudo).

```
yourengine host privilegeElevation profileScript > create
yourengine host privilegeElevation profileScript *> set name=<scriptname>
yourengine host privilegeElevation profileScript *> set contents=<your script>
yourengine host privilegeElevation profileScript *> set profile=<yourprofile>
```

4. Commit to save.

```
yourengine host privilegeElevation profileScript *> commit
```

13.2.4.3.5.4 Privilege elevation profiles and Delphix replication

Because all Delphix Engines have a default Profile called sudo, which would normally exist on both source and target Delphix Engines, replication collisions which would normally prevent a successful failover are automatically resolved.

- Only Profiles which are actually assigned to a host are replicated. All currently unassigned profiles are ignored.
- Profile name collisions are resolved by the display names of duplicate Profiles being prefixed with a unique object identifier. This is described in more detail in the next section.
- A default Profile will not retain default status after replication failover. The Profile assigned as default on the replication target Delphix Engine will remain the default. Therefore, if the source Delphix

Engine has a non-standard default Profile, it will need to be manually set as the new default on the replication target Delphix Engine after failover.

Caveats

By design, the Delphix Engine allows the creation of Profiles with duplicate names. This is not a bug. It exists for several reasons:

- This allows replication failover to complete without duplicate Profile names triggering a collision.
- Makes versioning possible so that a profile with the same name can have multiple versions as iterations are made (some of which may not be production-ready).

However, this behavior has the consequence of changing the display name of Profiles. Once a duplicate name exists, a unique object identifier is prefixed to the name. Any references to such a profile (such as assigning to a host) must use the long format with the unique identifier. For example, standard Delphix Engine has the following Profile:

```
Delphix5031HWv8> host privilegeElevation profile ls
Objects
NAME      ISDEFAULT
sudo      true
Operations
create
```

If a new profile that is also called "sudo" is created, the display names automatically change as follows:

```
Objects
NAME                                        ISDEFAULT
`HOST_PRIVILEGE_ELEVATION_PROFILE-1/sudo    true
`HOST_PRIVILEGE_ELEVATION_PROFILE-5/sudo    false

Operations
create
```

Known issues

- It is not possible to delete a Profile. Attempting to do so results in an API error. However, Profiles can be renamed to something meaningful like "unused_1".
- Profiles created that contain single quotation marks can no longer be selected. They become orphaned Profiles.
- When pasting in script contents, the cursor does not correctly move to the end of the last line being pasted. Unless the cursor is moved to the end of the line before pressing ENTER, the script will not be complete.

Neither of the above issues has any operational impact.

13.2.4.3.6 CLI cookbooks: enabling and configuring environment permissions

This topic describes how to enable the environment permissions feature to restrict what users can do with environments.

By default, all engine users can list all environments and hosts and see their details. Moreover, all users are able to link dSources from and provision VDBs to any environment without requiring any permissions on environments, as long as they have appropriate permissions on the target group where the dsource or VDB will be located (see page 674).

13.2.4.3.6.1 Enabling environment and permissions

To restrict non-administrator users from seeing, linking from, and provisioning to any environment, Engine Administrators can enable environment authorizations.

```
delphix> authorization configurationdelphix authorization configuration >
lsProperties    type: AuthorizationConfig    environmentAndHostAuth: false
 Operationsupdatedelphix authorization configuration> updatedelphix authorization
configuration update *> set environmentAndHostAuth=truedelphix authorization
configuration update *> commit
```

Similarly, to go back to the default state in which all users have permission to perform those operations, the Engine Administrator must set the `environmentAndHostAuth` property back to `false`.

13.2.4.3.6.2 Granting and revoking permissions on environments and hosts

When environment permissions are enabled, only Engine Administrators can list environments and hosts, see their details, or link dSources from or provision VDBs to environments.

To authorize any other user to perform such an operation on an environment or host, a Engine Administrator must create an appropriate authorization.

```
delphix> authorization createdelphix authorization create *> set user=someuserdelphix
authorization create *> set role=PROVISIONERdelphix authorization create *> set
target=SourceEnvironment:/somehost.example.com
```

To revoke an authorization, a Engine Administrator must delete the corresponding authorization object.

```
delphix> authorizationdelphix> lsREFERENCE        USER      ROLE   TARGET
AUTHORIZATION-1  sysadmin  OWNER  sysadminAUTHORIZATION-2  admin     OWNER
adminAUTHORIZATION-3  admin     OWNER  domain0AUTHORIZATION-4  someuser  Data
SourceEnvironment:/somehost.example.com delphix authorization> select
`AUTHORIZATION-4delphix authorization '(USER-2, ROLE-2, UNIX_HOST_ENVIRONMENT-1)'>
deletedelphix authorization '(USER-2, ROLE-2, UNIX_HOST_ENVIRONMENT-1)' delete *>
commit
```

Permissions on Environments and Hosts

Role	Environment privileges	Host privileges
Owner	• Can provision VDBs from the environment • Can link dSources from the environment • Can access the same information as a Reader	• Can access the same information as a Reader
Provisioner	• Can access statistics on the dSource, VDB, or snapshot such as usage, history, and space consumption • Can provision VDBs from owned dSources and VDBs	• Can access the same information as a Reader
Data Operator	• Can access statistics on the dSource, VDB, or snapshot such as usage, history, and space consumption • Can refresh or rollback VDBs • Can snapshot dSources and VDBs	• Can access the same information as a Reader
Reader	• Can see the configuration of the environment	• Can see the configuration of the host
Self-Service Only	• Can access the same information as a Reader	• Can see the configuration of the host

13.2.4.4 CLI cookbook: source databases and dSources

These topics describe command-line interface procedures for working with dSources.

This section covers the following topics:

- CLI cookbook: linking an Oracle staging push database (see page 2599)
- CLI cookbook: detaching and attaching a SQL server dSource (see page 2602)
- CLI cookbook: disabling LogSync for a dSource (see page 2604)
- CLI cookbook: enabling Oracle validated sync (see page 2604)
- CLI cookbook: linking a SQL server database loading from a specific full backup of the source database (see page 2606)
- CLI cookbook: linking a SQL server database loading from the last full backup of the source database (see page 2607)
- CLI cookbook: linking to a single instance Oracle database (see page 2607)
- CLI cookbook: listing data source sizes (see page 2610)
- CLI cookbook: detaching and attaching an Oracle dSource (see page 2611)
- CLI cookbook: how to change database user password (see page 2612)

- CLI cookbook: changing an SAP ASE dSource's staging database (see page 2613)
- CLI cookbook: detaching and attaching a SAP ASE dSource (see page 2615)
- CLI cookbook: linking an SAP ASE database loading from the last full backup of the source database (see page 2616)

13.2.4.4.1 CLI cookbook: linking an Oracle staging push database

This topic describes how to link an Oracle Staging Push database using the Delphix Engine command-line interface.

13.2.4.4.1.1 Prerequisites

You will need the following information:

- The name of the dSource you want to create.
- The group in which you want to create the dSource.
- The Database Name of the Source Database whose data will be populated on this staging database.
- The Repository and the Mount Base.

13.2.4.4.1.2 Linking a CDB

Procedure

- Execute the *database link* command

```
delphix> database link
delphix database link>
```

- The default *linkData.type* in the *LinkParameters* is set to *ASELinkData*, but you can confirm that by getting the input type. Set *linkData.type* to *OracleLinkFromStaging* for Oracle Staging Push database.

```
delphix database link *> get linkData.type
    ASELinkData
delphix database link *> set linkData.type=OracleLinkFromStaging
delphix database link *> get linkData.type
    OracleLinkFromStaging (*)
delphix database link *> ls
Properties
    type: LinkParameters
    name: (required)
    description: (unset)
    group: (required)
    linkData:
        type: OracleLinkFromStaging (*)
        allowAutoStagingRestartOnHostReboot: (required)
        containerType: (required)
```

```
        customEnvVars: (unset)
        databaseName: (required)
        externalFilePath: (unset)
        operations: (unset)
        sourcingPolicy: (unset)
        stagingSourceParameters: (required)
        syncParameters:
            type: OracleStagingPushSyncParameters
        syncStrategy: (required)
        uniqueName: (required)

Operations
defaults
```

- Set the name for the CDB dSource and the group in which you want to create it.

```
delphix database link *> set name=exampleCDB1
delphix database link *> set group="<Group Name>"
```

- Set the container type to *ROOT_CDB* for the container database.

```
delphix database link *> set linkData.containerType=ROOT_CDB
```

- Set the *databaseName*, *uniqueName*, and other *linkData* properties. The database name here should be the same as the database name of the Source Database whose data will be populated on this staging database.

```
delphix database link *> set linkData.databaseName="<database_name>"
delphix database link *> set linkData.uniqueName=CDB_1_UNQ_NAME
delphix database link *> set linkData.allowAutoStagingRestartOnHostReboot=true
delphix database link *> set linkData.syncStrategy.type=OracleStagingPushSyncStrategy
```

- Set staging source properties - *mountBase, instanceName,* and *repository* among others.

```
delphix database link *> edit linkData.stagingSourceParameters
delphix database link linkData.stagingSourceParameters *> ls
Properties
    type: OracleStagingSourceParameters (*)
    configParams: (unset)
    configTemplate: (unset)
    environmentUser: (unset)
    instanceName: (required)
    mountBase: (required)
    physicalStandby: (unset)
    repository: (required)
 delphix database link linkData.stagingSourceParameters *> set mountBase=/mnt/staging
```

```
delphix database link linkData.stagingSourceParameters *> set
instanceName=CDB_1_INSTANCE_NAME
```

If you are unsure of the available repositories, you can list available repositories:

```
elphix database link linkData.stagingSourceParameters *> /repository list
NAME                                          VERSION     ENVIRONMENT
'/u01/app/oracle/product/18.0.0.0/dbhome_1'      18.0.0.0.0  env1
'/u01/app/oracle/product/19.8.0.0/dbhome_1'      19.8.0.0.0  env2
delphix database link linkData.stagingSourceParameters *> set repository='/u01/app/
oracle/product/19.8.0.0/dbhome_1'
```

- Commit the result.

```
delphix database link *> commit
    `ORACLE_DB_CONTAINER-45
    Dispatched job JOB-129
    DB_LINK job started for "<Group Name>/exampleCDB1".
    Creating new TimeFlow for dSource "exampleCDB1".
    Generating required scripts needed for the staging database.
    Mounting datasets on the staging host "<staging host>".
    Starting staging database "<database_name>" for dSource "exampleCDB1" in NOMOUNT
mode.
    DB_LINK job for "<Group Name>/exampleCDB1" completed successfully.
delphix>
```

13.2.4.4.1.3 Linking a PDB

Procedure

- Execute the *database link* command.

```
delphix> database link
delphix database link>
```

- Set *linkData.type* to *OraclePDBLinkFromStaging* for Oracle Staging Push pluggable database.

```
delphix database link *> set linkData.type=OraclePDBLinkFromStaging
```

- Set the name for the PDB dSource and the group in which you want to create it.

```
delphix database link *> set name=examplePDB1
delphix database link *> set group="<Group Name>"
```

- Set *linkData.cdbConfig* to the config of the linked staging push CDB dSource, where this PDB should be plugged into. For Oracle databases, configs are identified by the database's unique name. You can list available source configurations:

```
delphix database link *> /sourceconfig list
NAME                 REPOSITORY            LINKINGENABLED
CDB_1_UNQ_NAME    '/opt/ora/dexample1'  true
CDB_2_UNQ_NAME    '/opt/ora/dexample2'  true
delphix database link *> set linkData.cdbConfig=CDB_1_UNQ_NAME
```

- Set *databaseName, syncStrategy.type* and other *linkData* properties. The database name here should be the same as the database name of the source PDB whose data will be populated on this staging PDB.

```
delphix database link *> set linkData.databaseName=CDOMLOSR381EPDB1
delphix database link *> set linkData.syncStrategy.type=OracleStagingPushSyncStrategy
delphix database link *> set linkData.allowAutoStagingRestartOnHostReboot=true
```

- Commit the result.

```
delphix database link *> commit
examplePDB1
    Dispatched job JOB-136
    DB_LINK job started for "<Group Name>/examplePDB1".
    Creating new TimeFlow for dSource "examplePDB1".
    Generating required scripts needed for the staging database.
    Mounting datasets on the staging host "<staging_host>".
    DB_LINK job for "<Group Name>/examplePDB1" completed successfully.
```

13.2.4.4.2 CLI cookbook: detaching and attaching a SQL server dSource

This topic describes how to detach and attach a SQL server dSource. When attaching a SQL Server dSource to a new data source, the new data source must be in the same database satisfying the following constraints:

- pptRepository needs to be set to the name of the SQL instance on the staging server. The unlink operation removes the database from the SQL instance on the staging server and unmounts the iscsi luns, reattaching the dSource via the CLI will remount the iscsi luns and puts the database back.

13.2.4.4.2.1 Prerequisites

A dSource can only be attached to a new data source once it has been unlinked (see page 1040).

13.2.4.4.2.2 Procedure

1. Select dSource.

```
delphix> database "dexample"
```

2. Run the detachSourcecommand, specifying the current active source. This step can be skipped if the dSource has already been detached. through the GUI.

```
delphix database "dexample"> detachSourcedelphix database "dexample"
 detachSource *> set source=dexampledelphix database "dexample" detachSource *>
commit
```

3. Run the `attachSource` command.

```
delphix database "dexample"> attachSource
```

4. Set the following for SQL Server: You can also type help pptRepository to see what is wanted You can also set pptRepository=<then press tab> to list all values.

```
delphix database "dexample" attachSource *> edit attachDatadelphix database
"dexample" attachSource attachData *> set type=MSSqlAttachDatadelphix database
"dexample" attachSource attachData *> set config=SQLSERVER/dexampledelphix
database "dexample" attachSource attachData *> set sharedBackupLocations="<\
\SERVER1\Backups>"delphix database "dexample" attachSource attachData *> set
pptRepository=SQL2008R2delphix database "dexample" attachSource attachData *>
set mssqlUser.type=MSSqlDomainUserdelphix database "dexample" attachSource
attachData *> set mssqlUser.user=ad\dbuserdelphix database "dexample"
 attachSource attachData *> set mssqlUser.password.password=dbuserpwddelphix
database "dexample" attachSource attachData *> edit ingestionStrategydelphix
database "dexample" attachSource attachData ingestionStrategy *>set
type=ExternalBackupIngestionStrategydelphix database "dexample" attachSource
attachData ingestionStrategy *>set validatedSyncMode=TRANSACTION_LOGdelphix
database "dexample" attachSource attachData ingestionStrategy *> backdelphix
database "dexample" attachSource attachData *> edit operations.preSyncdelphix
database "dexample" attachSource attachData  operations.preSync *> backdelphix
database "dexample" attachSource attachData*> edit operations.postSyncdelphix
database "dexample" attachSource attachData operations.postSync *> back
```

5. Commit the operation.

```
delphix database "dexample" attachSource *> commit
```

13.2.4.4.3 CLI cookbook: disabling LogSync for a dSource

This topic provides a simple example of how the nested state is represented and manipulated. The LogSync state is maintained in the sourcingPolicy property of dSources, itself an object with several different fields.

13.2.4.4.3.1 Procedure

1. Select the dSource to be changed and run the `update` command.

```
delphix> database "example"
delphix "example"> update
```

2. Get the current property using dot-delimited notation.

```
delphix "example" update *> get sourcingPolicy.logsyncEnabled
    true
```

3. The property could also be set using dot-delimited notation, but for illustrative purposes, we can also use the `edit` command and set it directly.

```
delphix "example" update *> edit sourcingPolicy
delphix "example" update sourcingPolicy *> set logsyncEnabled=false
```

4. Commit the state, either from within the editing context or after running `back` to return to the parent context.

```
delphix "example" update sourcingPolicy *> commit
delphix "example">
```

13.2.4.4.4 CLI cookbook: enabling Oracle validated sync

13.2.4.4.4.1 Prerequisite - designating a staging host

In order to validate an Oracle dSource snapshot for syncing, the Delphix Engine requires a host with an Oracle installation that is compatible with the dSource. This machine is known as the **staging** host. You must explicitly designate which machines you want the Delphix Engine to use as staging hosts. All machines that have been marked as staging sites are added to a pool. During validated sync, the Delphix Engine will select a

compatible host from the pool, export the requisite archived redo logs and datafiles, and execute Oracle media recovery on the host. Follow these steps to designate a staging host.

1. Select the repository you want to designate as staging.

```
delphix>/repository/select '/u01/app/ora10205/product/10.2.0/db_1'
```

2. Execute the `update` command.

```
delphix repository "'/u01/app/ora10205/product/10.2.0/db_1'">update
```

3. Set staging to `true`.

```
delphix repository "'/u01/app/ora10205/product/10.2.0/db_1'" update *>set
staging=true
```

4. `Commit` the operation to designate the repository as staging.

```
delphix repository "'/u01/app/ora10205/product/10.2.0/db_1'" update *> commit
```

To configure validated sync for multiple dSources with different Oracle versions, you must designate a compatible staging source for each. If multiple compatible staging sites exist, the Delphix Engine will select one at random.

The validated sync process will consume some resources on the staging host when snapshots are taken. Designating a performance-critical host as a staging host is not recommended.

13.2.4.4.4.2 Procedure - Enabling validated sync

1. Select the dSource for which you want to enable validated sync.

```
delphix>/database/select redsox1
```

2. Execute the `update` command.

```
delphix database "redsox1">update
```

3. Set `preProvisioningEnabled` to `true`.

```
delphix database "redsox1" update *>set preProvisioningEnabled=true
```

4. `Commit` the operation to enable validated sync.

```
delphix database "redsox1" update *>commit
```

13.2.4.4.5 CLI cookbook: linking a SQL Server database loading from a specific full backup of the source database

This topic describes how to use the command-line interface to link a SQL Server database by loading from a specific full backup of the source database as indicated by the backup UUID.

13.2.4.4.5.1 Prerequisites

- You can get the backup UUID for the all backup files of a chosen database using the following query on the source database under the column ***backup_set_uuid***

```
Use masterselect backupset.database_name,      backupset.type,
backupset.backup_set_id,      backupset.backup_set_uuid,
backupset.family_guid,      backupset.position,      backupset.first_lsn,
backupset.last_lsn,      backupset.database_backup_lsn,      backupset.name,
backupset.has_bulk_logged_data,      backupset.is_damaged,
backupset.begins_log_chain,      backupset.is_copy_only,
backupset.backup_finish_date,      backupset.database_version,
backupset.database_guid,mediafamily.logical_device_name,mediafamily.physical_de
vice_namefrom msdb.dbo.backupmediafamily mediafamily join msdb.dbo.backupset
backupseton mediafamily.media_set_id = backupset.media_set_id where
backupset.database_name = N'<Database Name>'order by
backupset.backup_finish_date desc
```

13.2.4.4.5.2 Procedure

Enter these commands through the Delphix Engine command-line interface:

```
/database; link;set type=LinkParameters;set name=<dSource name>;set group=<group
name>;set linkData.type=MSSqlLinkData;set
linkData.syncParameters.type=MSSqlExistingSpecificBackupSyncParameters;set
linkData.config=<source database>;set linkData.sharedBackupLocations="<source
database backup locations>";set linkData.pptRepository=<SQL instance on the staging
server>;set linkData.sourcingPolicy.type=SourcingPolicy;set
linkData.mssqlUser.type=MSSqlDomainUserset linkData.mssqlUser.user=ad\dbuserset
linkData.mssqlUser.password.password=dbuserpwdset
linkData.syncParameters.backupUUID=<backup UUID>;set
```

```
linkData.ingestionStrategy.type=<ingestion strategy type>;set
linkData.ingestionStrategy.validatedSyncMode=<validated sync mode type>; commit;
```

13.2.4.4.6 CLI cookbook: linking a SQL Server database loading from the last full backup of the source database

This topic describes how to use the command-line interface to link a SQL Server database by loading from the last full backup of the source database.

13.2.4.4.6.1 Procedure

Enter the following commands in the Delphix Engine command-line interface:

```
/database; link;set type=LinkParameters;set name=<dSource name>;set group=<group
name>; set linkData.type=MSSqlLinkData;set
linkData.syncParameters.type=MSSqlExistingMostRecentBackupSyncParametersset
linkData.config=<source database>;set linkData.sharedBackupLocations="<source
database backup locations>";set linkData.pptRepository=<SQL instance on the staging
server>;set linkData.sourcingPolicy.type=SourcingPolicy;set
linkData.mssqlUser.type=MSSqlDomainUserset linkData.mssqlUser.user=ad\dbuserset
linkData.mssqlUser.password.password=dbuserpwdset
linkData.ingestionStrategy.type=<ingestion strategy type>; commit;
```

13.2.4.4.7 CLI cookbook: linking to a single instance Oracle database

This topic describes how to link to a single instance Oracle database using the Delphix Engine command-line interface.

13.2.4.4.7.1 Prerequisites

You will need the following information:

- The name of the dSource you want to create.
- The group in which you want to create the dSource.
- The host environment user with sufficient privileges as described in the Delphix User Guide.
- The database unique name of the Oracle database you want to link to.

13.2.4.4.7.2 Procedure

1. Execute the `database link` command.

   ```
   delphix> database link
   delphix database link *>
   ```

2. The default `linkData.type` in the `LinkParameters` is set to `ASELinkData`, but you can confirm that by getting the input type. Set `linkData.type` to `OracleLinkFromExternal` for Oracle database.

```
delphix database link *> get linkData.type
    ASELinkData
delphix database link *> set linkData.type=OracleLinkFromExternal
delphix database link *> get linkData.type
    OracleLinkFromExternal (*)
delphix database link *> ls
Properties
    type: LinkParameters
    name: (required)
    description: (unset)
    group: (required)
    linkData:
        type: OracleLinkFromExternal (*)
        diagnoseNoLoggingFaults: (unset)
        environmentUser: (required)
        externalFilePath: (unset)
        linkNow: (unset)
        nonSysCredentials: (unset)
        nonSysUser: (unset)
        operations: (unset)
        oracleFallbackCredentials: (unset)
        oracleFallbackUser: (unset)
        preProvisioningEnabled: (unset)
        sourcingPolicy: (unset)
        syncParameters:
            type: OracleSyncFromExternalParameters
            doNotResume: (unset)
            doubleSync: (unset)
            filesForFullBackup: (unset)
            forceFullBackup: (unset)
            skipSpaceCheck: (unset)
        syncStrategy: (required)

Operations
defaults
```

3. Set the name for the dSource and the group in which you want to create it.

```
delphix database link *> set name=example1
delphix database link *> set group="<Group Name>"
```

4. Adjust any other properties you may want, such as description, and whether to link now. The full set of options is described in the API documentation for the `OracleLinkFromExternal` type. If you

set the `linkNow` property, then this operation will wait for the sync to complete, otherwise, you can perform the initial link by running the sync command at a later point

```
delphix database link *> set linkData.linkNow=true
```

5. Set the privileged environment user.

This user must be from the same environment as the associated source config set in step 4. You can list the set of available users through the `environment user list` command.

```
delphix database link *> /environment/user list
NAME
oracle
delphix database link *> set linkData.environmentUser=oracle
```

6. Set the `linkData.syncStrategy.type`

```
delphix database link *> set linkData.syncStrategy.type=OracleSourceBasedSyncStrategy
```

7. Set the source configuration.
   For Oracle databases, the source configurations are identified by the database unique name. If you are unsure of the set of available databases, you can list available source configurations.

```
delphix database link *> /sourceconfig list
NAME      REPOSITORY            LINKINGENABLED
example1  '/opt/ora/dexample1'  true
example2  '/opt/ora/dexample1'  true
delphix database link *> set linkData.syncStrategy.config=example1
```

8. Adjust any other properties of `linkData.syncStrategy` you may want, such as RMAN tunables.
9. Check that all the settings you require are in place using the "ls" command.

```
delphix database link *> ls
Properties
    type: LinkParameters
    name: example1 (*)
    description: (unset)
    group: <Group Name> (*)
    linkData:
        type: OracleLinkFromExternal (*)
        diagnoseNoLoggingFaults: (unset)
        environmentUser: oracle (*)
```

```
        externalFilePath: (unset)
        linkNow: true (*)
        nonSysCredentials: (unset)
        nonSysUser: (unset)
        operations: (unset)
        oracleFallbackCredentials: (unset)
        oracleFallbackUser: (unset)
        preProvisioningEnabled: (unset)
        sourcingPolicy: (unset)
        syncParameters:
            type: OracleSyncFromExternalParameters
            doNotResume: (unset)
            doubleSync: (unset)
            filesForFullBackup: (unset)
            forceFullBackup: (unset)
            skipSpaceCheck: (unset)
        syncStrategy:
            type: OracleSourceBasedSyncStrategy (*)
            backupLevelEnabled: (unset)
            bandwidthLimit: 0 (*)
            checkLogical: false (*)
            compressedLinkingEnabled: true (*)
            config: example1 (*)
            encryptedLinkingEnabled: false (*)
            filesPerSet: 5 (*)
            numberOfConnections: 1 (*)
            rmanChannels: 2 (*)

Operations
defaults
```

10. Commit the result.

```
delphix database link *> commit
    `ORACLE_DB_CONTAINER-29
    Dispatched job JOB-263
    DB_LINK job started for "<Group Name>/example1".
    Obtaining information from source database "<Group Name>/example1".
    Creating new TimeFlow for dSource "<Group Name>/example1".
    The dSource "example1" was successfully linked from source database "<Group
Name>/example1".
    DB_LINK job for "<Group Name>/example1" completed successfully.
```

13.2.4.4.8 CLI cookbook: listing data source sizes

This topic describes a basic use of the CLI `list` command.

1. Switch to the source view and view the default list.

```
delphix> sourcedelphix source> listNAME      CONTAINER  VIRTUAL  CONFIG example
example    false    examplevexample  vexample   true     vexample
```

2. List sources with their database size.

```
delphix> sourcedelphix source> select exampledelphix source 'example'> get
runtime.databaseSize    2.80GB
```

13.2.4.4.9 CLI cookbook: detaching and attaching an Oracle dSource

This topic describes how to attach a dSource to a different data source.

13.2.4.4.9.1 Prerequisites

Before detaching an Oracle dSource, you must capture the following RMAN configuration:

- Level or SCN based backups
- Data load channel settings: number of channels and files per channel

The above RMAN configuration will be removed once the dSource is unlinked.

A dSource can only be attached to a new data source once it has been unlinked.

When attaching an Oracle dSource to a new data source, the new data source must be the same logical database satisfying the following constraints:

- Same dbid
- Same dbname
- Same creation time
- Same resetlogs SCN
- Same resetlogs time
- Same redo stream, where a log must exist with
    - Same sequence
    - Same thread
    - Same end SCN

For Oracle dSources, this procedure can be used to initially link from a standby server that is faster or less disruptive, unlink the dSource, and then attach it to the production server for subsequent incremental SnapSync operations. When you perform the attach operation, you will need the source config name of an unlinked database.

13.2.4.4.9.2 Procedure

1. Select **dSource**.

```
delphix> database "dexample"
```

2. Run the `detachSource` **command**, specifying the currently active source. This step can be skipped if the dSource has already been detached through the GUI.

```
delphix database "dexample"> detachSourcedelphix database "dexample"
 detachSource *> set source=name-of-old-src-DB-serverdelphix database
"dexample" detachSource *> commit
```

3. Run the `attachSource` **command**.

```
delphix database "dexample"> attachSource
```

4. Set the config to point to an unlinked source. The following is an example to attach to an Oracle data source: (Hint: use <TAB> to complete variable names and known values)

```
delphix database "dexample" attachSource *> set
attachData.type=OracleAttachDatadelphix database "dexample" attachSource *> set
attachData.config=name-of-dSource-as-shown-in-environmentdelphix database
"dexample" attachSource *> set attachData.environmentUser=myuserdelphix
database "dexample" attachSource *> set
attachData.oracleFallbackUser=orauserdelphix database "dexample" attachSource
*> edit attachData.oracleFallbackCredentialsdelphix database "dexample"
 attachSource *> set set type=PasswordCredentialdelphix database "dexample"
 attachSource *> set password=orauserpwd
```

The attachData.config listing can be observed in the administration GUI, under Manage (pull-down at the top) --> Environments --> choose your dSource --> select Databases (upper right).
The dSources shown on the new source DB server may acquire odd names eg. "P02:UNKNOWN:vwxyz". These are a side-effect of having multiple instances of the same container name (possible in DR); Delphix needs to disambiguate. If these do show, they indicate containers in the new environment. Successfully attaching to these and deleting the old environment will re-establish the original names.

5.
```
delphix database "dexample" attachSource *> commit
```

13.2.4.4.10 CLI cookbook: how to change database user password

1. ssh into your engine using Admin privileges.

```
ssh admin@delphixengine
```

2. Go to sourceconfig and find the Database that you need to update the password on.

```
delphix > sourceconfigdelphix sourceconfig > lsdelphix sourceconfig > select
<yourdatabase>
```

3. Update the password.

```
delphix sourceconfig "yourdatabase" > updatedelphix sourceconfig "yourdatabase"
 update *> lsdelphix sourceconfig "yourdatabase" update *> set
credentials.password=<new password>
```

4. Commit the change.

```
delphix sourceconfig "database" update *> commit
```

13.2.4.4.10.1 Example:

```
ssh admin@exampledelphix > sourceconfigdelphix sourceconfig > lsObjects NAME
REPOSITORY                  LINKINGENABLEDmeta1  '/u01/oracle/10.2.0.4/ee1'  true
 Operationscreate delphix sourceconfig > select meta1delphix sourceconfig "meta1" >
ls  Properties    type: OracleSIConfig    name: meta1    credentials:        type:
PasswordCredential        password: ********    databaseName: meta1    discovered:
true    environmentUser: delphix    instance:        type: OracleInstance
instanceName: meta1        instanceNumber: 1    linkingEnabled: true
nonSysCredentials: (unset)    nonSysUser: (unset)    reference:
ORACLE_SINGLE_CONFIG-1    repository: '/u01/oracle/10.2.0.4/ee1'    services:
0:            type: OracleService            discovered: true
jdbcConnectionString: jdbc:oracle:thin:@172.16.100.69:1525:meta1        1:
type: OracleService            discovered: true            jdbcConnectionString:
jdbc:oracle:thin:@172.16.100.69:1521:meta1    uniqueName: meta1    user: delphix
Operationsdeleteupdatevalidate Credentials delphix sourceconfig "meta1" >
updatedelphix sourceconfig "meta1" update *> set credentials.password=<new
 password>delphix sourceconfig "meta1" update *> commit
```

13.2.4.4.11 CLI cookbook: changing an SAP ASE dSource's staging database

13.2.4.4.11.1 Prerequisites

In order to change an SAP ASE dSource's staging database, you need to know the name of the staging database and you need to disable the dSource.

13.2.4.4.11.2 Procedure

1. To find the name of the staging database, hover the mouse over the "Staging Database" name on the dSource card in the GUI or issue the following command via the CLI (replacing "pubs2" with the name of the dSource and "delphix.acme.com" with the hostname of your Delphix Engine):

```
$ echo "/source; select pubs2; ls" | ssh admin@delphix.acme.com | grep
stagingSourcePassword:     stagingSource: dxpb91a7LJlwj933xvlPvd_pubs2
```

2. Disable the dSource by either toggling the "Enable/Disable" toggle in the Delphix Management application or by using the CLI:

```
$ ssh admin@delphix.acme.comdelphix> sourcedelphix source> select pubs2delphix
source 'pubs2'> disabledelphix source 'pubs2' disable *> set
type=SourceDisableParameters delphix source 'pubs2' disable *> commit
Dispatched job JOB-365    SOURCE_DISABLE job started for "pubs2".
SOURCE_DISABLE job for "pubs2" completed successfully.
```

3. Change the repository of the staging database to reside in the desired SAP ASE instance:

```
$ ssh admin@delphix.acme.comdelphix> sourceconfigdelphix sourceconfig> select
dxpb91a7LJlwj933xvlPvd_pubs2delphix sourceconfig 'dxpb91a7LJlwj933xvlPvd_pubs2'>
 updatedelphix sourceconfig 'dxpb91a7LJlwj933xvlPvd_pubs2' update *>
lsProperties    type: ASESIConfig    credentials: (unset)    databaseName:
dxpb91a7LJlwj933xvlPvd_pubs2    environmentUser: nstacksolasetest/sybase
linkingEnabled: false    repository: nstacksolasetest/SRC_157_4K    user:
(unset)delphix sourceconfig 'dxpb91a7LJlwj933xvlPvd_pubs2' update *> set
environmentUser=nstacksolasestg/sybasedelphix sourceconfig
'dxpb91a7LJlwj933xvlPvd_pubs2' update *> set repository=nstacksolaseprod/
SRC_157_4K                              NSTACK_16K
RH68_ASE157_S1SQL2008R2                                                  SQL2012
nstacksolasestg                                            nstacksolasetest/
SRC_157_4Kdelphix sourceconfig 'dxpb91a7LJlwj933xvlPvd_pubs2' update *> set
repository=nstacksolasestgdelphix sourceconfig 'dxpb91a7LJlwj933xvlPvd_pubs2'
 update *> commit
```

Hint: Rapidly hit the "tab" key twice after typing "set repository=" to make the CLI present a list of available instances.

4. Enable the dSource:

```
delphix sourceconfig 'dxpb91a7LJlwj933xvlPvd_pubs2'> /sourcedelphix source>
select pubs2delphix source 'pubs2'> enabledelphix source 'pubs2' enable *> set
type=SourceEnableParameters delphix source 'pubs2' enable *> commit
```

```
Dispatched job JOB-367    SOURCE_ENABLE job started for "pubs2".
SOURCE_ENABLE job for "pubs2" completed successfully.
```

13.2.4.4.12 CLI cookbook: detaching and attaching a SAP ASE dSource

This CLI cookbook recipe describes how to Detach and Attach an SAP ASE dSource using the CLI.

13.2.4.4.12.1 Prerequisites

A dSource can only be attached to a new data source once it has been unlinked.
When attaching an SAP ASE dSource to a new data source, the new data source must be the same logical database satisfying the following constraints:

- Same dbid
- Same dbname
- Same creation time

You must also make sure that you follow the normal prerequisites for an SAP ASE data source found in SAP ASE Support and Requirements (see page 1877).

13.2.4.4.12.2 Procedure

13.2.4.4.12.3 Detach a dSource

1. Login to the CLI as admin or a user with Admin privileges.
2. Select dSource.

   ```
   delphix> database "dexample"
   ```

3. Run the `detachSourcecommand`, specifying the currently active source. Note: This step can be skipped if the dSource has already been detached through the GUI.

   ```
   delphix database "dexample"> detachSourcedelphix database "dexample"
    detachSource *> set source=dexampledelphix database "dexample" detachSource *>
   commit
   ```

13.2.4.4.12.4 Attach a dSource

1. Login to the CLI as admin or a user with Admin privileges.
2. Run the `attachSource` command.

```
delphix database "dexample"> attachSourcedelphix database "dexample"
 attachSource *> set attachData.config=dexampledelphix database "dexample"
 attachSource *> set attachData.dbCredentials.password=sybasedelphix database
"dexample" attachSource *> set attachData.dbUser=sadelphix database "dexample"
 attachSource *> set attachData.loadBackupPath=/tmp/backupsdelphix database
"dexample" attachSource *> set attachData.sourceHostUser="source_host_environme
nt/sybase"delphix database "dexample" attachSource *> set
attachData.stagingHostUser="staging_host_environment/sybase"delphix database
"dexample" attachSource *> set attachData.stagingRepository="staging_ASE_server
name_example"delphix database "dexample" attachSource *> lsProperties
type: ASEAttachSourceParameters      attachData:          type: ASEAttachData
config: dexample (*)          dbCredentials:              type:
PasswordCredential              password: ******** (*)         dbUser: sa (*)
dumpCredentials: (unset)         externalFilePath: (unset)
loadBackupPath: /tmp/backups (*)          loadLocation: (unset)
mountBase: (unset)          operations: (unset)          sourceHostUser:
source_ASE_servername_example/sybase (*)          stagingHostUser:
staging_ASE_servername_example/sybase (*)          stagingPostScript: (unset)
stagingPreScript: (unset)          stagingRepository:
staging_ASE_servername_example (*)             validatedSyncMode: ENABLED
delphix database "dexample" attachSource *> commit      ASE_DB_CONTAINER-3
Dispatched job JOB-25      DB_ATTACH_SOURCE job started for "Untitled/dexample".
      DB_ATTACH_SOURCE job for "Untitled/dexample" completed successfully.
```

This command is only necessary if you are using a Remote Backup Server configuration from staging to the source, instead of an NFS mounted shared directory for backups and transaction log dumps:

```
delphix database "dexample" attachSource *> set
attachData.loadLocation.backupServerName=source_backupserver_name_example
```

13.2.4.4.13 CLI cookbook: linking an SAP ASE database loading from the last full backup of the source database

This topic describes how to use the command-line interface to link an SAP ASE database by loading from the most recent full backup of the source database.

13.2.4.4.13.1 Procedure

Enter the following commands in the Delphix Engine command-line interface:

```
ssh delphix_admin@delphixPassword: delphix> /database linkdelphix database link *>
set linkData.type=ASELinkDatadelphix database link *> set name=db2delphix database
link *> set group="ASE dSource"delphix database link *> set
linkData.config=db2delphix database link *> set linkData.dbUser=sadelphix database
link *> set linkData.dbCredentials.password=sybasedelphix database link *> set
linkData.loadBackupPath=/mnt/dumpdelphix database link *> set linkData.mountBase=/
mnt/provision/vdb_or_staging_database_namedelphix database link *> set
linkData.sourceHostUser=nstackrh69/sybasedelphix database link *> set
linkData.stagingHostUser=nealorarh75/sybasedelphix database link *> set
linkData.stagingRepository=ASE157SP138delphix database link *> set
linkData.sourcingPolicy.logsyncEnabled=truedelphix database link *> set
linkData.sourcingPolicy.type=SourcingPolicydelphix database link *> unset
linkData.syncParametersdelphix database link *> edit linkData.syncParametersdelphix
database link linkData.syncParameters *> backdelphix database link *> lsProperties
type: LinkParameters    name: db2 (*)    description: (unset)    group: ASE dSource
(*)    linkData:        type: ASELinkData        config: db2 (*)
dbCredentials:            type: PasswordCredential            password: ******** (*)
dbUser: sa (*)        dumpCredentials: (unset)        dumpHistoryFileEnabled: false
externalFilePath: (unset)        loadBackupPath: /mnt/dump (*)        loadLocation:
(unset)        mountBase: /mnt/provision/vdb_or_staging_database_name (*)
operations: (unset)        sourceHostUser: nstackrh69/sybase (*)
sourcingPolicy:            type: SourcingPolicy (*)            logsyncEnabled: true
 (*)        stagingHostUser: nealorarh75/sybase (*)        stagingOperations: (unset)
stagingPostScript: (unset)        stagingPreScript: (unset)        stagingRepository:
ASE157SP138 (*)        syncParameters:            type: ASELatestBackupSyncParameters
(*)        validatedSyncMode: ENABLEDOperationsdefaultsdelphix database link *>
commit    `ASE_DB_CONTAINER-209    Dispatched job JOB-3704    DB_LINK job started for
 "ASE dSource/db2".    DB_LINK job for "ASE dSource/db2" completed successfully.
```

mountBase option

The **mountBase** parameter is an option that was added in Delphix 5.2 and higher. By default, Delphix mounts the staging database under the Delphix toolkit directory. If a directory is specified for this parameter, the staging database's NFS devices will be mounted under this directory rather than under the toolkit directory. This can be especially helpful when the dSource is linked with LogSync enabled. SAP ASE has a limit of 127 characters for the fully qualified path to the transaction logs specified in the "LOAD TRANSACTION" statement. When LogSync is enabled, Delphix keeps a copy of the transaction logs on the engine under the dSources "archive" folder. The default naming convention for the folders under the toolkit can easily cause the 127 character limit to be exceeded so it is highly recommended to use this parameter when enabling LogSync. The mountBase is limited to 87 characters (the device names Delphix generates are 32 characters and the subdirectory containing the archive files is about 8 characters long). This leaves approximately 40 characters for the name of the transaction logs themselves. The mountBase parameter must be unique for each VDB or staging database. The

path can reside under a common parent directory for example, /mnt/provision but you must specify a unique child directory under the parent for each VDB or staging database using this optional parameter.xs

13.2.4.5 CLI cookbook: VDBs

This section covers the following topics:

- CLI cookbook: attaching or detaching a PDB (see page 2619)
- CLI cookbook: attaching, detaching, or linking a CDB (see page 2620)
- CLI cookbook: changing SGA parameter (see page 2623)
- CLI cookbook: changing the SID of Oracle RAC VDBs (see page 2623)
- CLI cookbook: toggle new DBID generation upon refresh options for Oracle VDBs (see page 2625)
- CLI cookbook: creating a policy (see page 2626)
- CLI cookbook: creating a VDB config template (see page 2627)
- CLI cookbook: determining the snapshot used to provision a VDB (see page 2629)
- CLI cookbook: how to refresh a VDB from a specific snapshot (see page 2630)
- CLI cookbook: Oracle VDB migration (see page 2631)
- CLI cookbook: provisioning a SAP ASE VDB (see page 2633)
- CLI cookbook: provisioning a single instance non-multitenant Oracle VDB (see page 2634)
- CLI cookbook: provisioning a SQL server VDB (see page 2637)
- CLI cookbook: provisioning a VDB from a timeflow bookmark (see page 2639)
- CLI cookbook: provisioning a virtual PDB to a new virtual CDB (see page 2642)
- CLI cookbook: provisioning a virtual PDB in a target CDB (see page 2648)
- CLI cookbook: Provisioning a TDE-enabled virtual PDB to a new virtual CDB (see page 2651)
- CLI cookbook: Provisioning a TDE-enabled vPDB in a target CDB (see page 2653)
- CLI cookbook: refresh a VDB from a specific timepoint or latest (see page 2654)
- CLI cookbook: repairing a timeflow (see page 2656)
- CLI cookbook: rolling back a VDB (see page 2658)
- CLI cookbook: rolling forward a VDB (see page 2660)
- CLI cookbook: taking a snapshot (see page 2662)
- CLI cookbook: V2P (Virtual to Physical) of a single instance non-multitenant Oracle database (see page 2663)
- CLI cookbook: V2P (Virtual to Physical) of a single instance Oracle database with datafiles on separate file systems (see page 2664)
- CLI cookbook: export a non-multitenant virtual Oracle database to ASM (see page 2666)
- CLI cookbook: export a multitenant virtual pluggable Oracle database to ASM (see page 2668)
- CLI cookbook: export a snapshot or a timeflow point of a non-multitenant Oracle database to ASM (see page 2670)
- CLI cookbook: export a snapshot or a timeflow point of a multitenant pluggable Oracle database to ASM (see page 2675)
- CLI cookbook: V2P virtual to physical on SQL server (see page 2678)

- CLI cookbook: VDB status (see page 2680)
- CLI cookbook: provisioning a virtual PDB from a non-multitenant source database (see page 2680)
- CLI cookbook: migrating a virtual PDB and a virtual CDB (see page 2687)
- CLI cookbook: migrating an Oracle RAC virtual PDB and a virtual CDB (see page 2689)
- CLI cookbook: locating and updating the value of tdeKeyIdentifier (see page 2692)
- CLI cookbook: Force refresh/rewind a virtual PDB (see page 2699)
- CLI Cookbook: Starting or stopping cluster instances of an Oracle RAC virtual database (see page 2700)

13.2.4.5.1 CLI cookbook: attaching or detaching a PDB

This topic describes how to attach or detach a PDB using the command-line interface.

13.2.4.5.1.1 PDB CLI attach

1. Ensure the new source environment has been fully discovered by the Delphix Engine, including CDB discovery.
2. Login to the engine via SSH with a Delphix admin account.
3. Select the PDB dSource database to be attached.

```
delphix> database
delphix database> select "R268PDB1"
delphix database 'R268PDB1'>
```

4. Attach the PDB dSource to the correct source PDB in the new environment using the `attachSource` command; you will need to specify the environment user and login credentials for the PDB dSource.

```
delphix database 'R268PDB1'> attachSource
delphix database 'R268PDB1' attachSource *> set
attachData.type=OraclePDBAttachData
delphix database 'R268PDB1' attachSource *> set attachData.config=R268PDB1
delphix database 'R268PDB1' attachSource *> set
attachData.environmentUser=ora12201
delphix database 'R268PDB1' attachSource *> commit
    R268PDB1
    Dispatched job JOB-46
    DB_ATTACH_SOURCE job started for "Untitled/R268PDB1".
    Starting validation of attach parameters for database "Untitled/R268PDB1".
    Validation finished for database "Untitled/R268PDB1"
    Obtaining information from source database "Untitled/R268PDB1".
    The dSource "R268PDB1" was successfully linked from source database
"Untitled/R268PDB1".
    DB_ATTACH_SOURCE job for "Untitled/R268PDB1" completed successfully.
```

13.2.4.5.1.2 PDB CLI detach

1. Login to the engine via SSH with a Delphix admin account.
2. Select the PDB dSource database to be detached.

```
delphix> database
delphix database> select "R268PDB1"
delphix database 'R268PDB1'>
```

3. Detach the PDB dSource using the `detachSource` command:

```
delphix database 'R268PDB1'> detachSource
delphix database 'R268PDB1' detachSource *> set source=R268PDB1
delphix database 'R268PDB1' detachSource *> commit
    Dispatched job JOB-45
    DB_DETACH_SOURCE job started for "Untitled/R268PDB1".
    DB_DETACH_SOURCE job for "Untitled/R268PDB1" completed successfully.
```

13.2.4.5.2 CLI cookbook: attaching, detaching, or linking a CDB

This topic describes how to attach, detach, or link a CDB using the command-line interface.

A CDB is not automatically detached when you detach the last PDB of the CDB. The CDB still remains in an inactive state which in turn prevents you from removing the environment. Therefore, you must detach the CDB in order to remove the environment.

> You can perform detach or attach operations only via Command-Line Interface (CLI).

13.2.4.5.2.1 CDB CLI detach

Perform the following steps to detach a CDB.

> You can detach a CDB only when there are no PDBs linked and the CDB does not have any vPDBs.

1. Login to the engine via SSH with a Delphix admin account.

```
$ ssh admin@YOUR_ENGINE
```

2. Select the CDB dSource database to be detached.

```
delphix> cd database
delphix database> select CDOMLOSR421F
delphix database 'CDOMLOSR421F'>
```

3. Detach the CDB dSource using the detachSource command.

```
delphix database 'CDOMLOSR421F'> detachSource
delphix database 'CDOMLOSR421F' detachSource *> set source=CDOMLOSR421F
delphix database 'CDOMLOSR421F' detachSource *> commit
    Dispatched job JOB-45
    DB_DETACH_SOURCE job started for "Untitled/CDOMLOSR421F".
    DB_DETACH_SOURCE job for "Untitled/CDOMLOSR421F" completed successfully.
```

You must attach the CDB back before performing the following operations: linking a new PDB, attaching a PDB, provisioning a vPDB, or attaching a converted PDB. Failure to do this will either result in an error due to duplicated object or will create a second CDB that will duplicate the space used for the CDB.

13.2.4.5.2.2 CDB CLI link

You can not perform an attach operation for PDBs that have already been detached from the primary CDB and are required to be attached to a standby CDB if the standby CDB is not already linked or attached. You must link the CDB manually before performing the PDB attach operation.

Perform the following steps to link a CDB.

1. Login to the engine via SSH with a Delphix admin account.

```
$ ssh admin@YOUR_ENGINE
```

2. Select the CDB dSource database to be linked.

```
delphix> cd database
delphix database> link
delphix database link *>
```

3. Link the CDB dSource using the link command.

```
delphix database link *> set name=mycdb
delphix database link *> set group=Untitled
delphix database link *> edit linkData
```

```
delphix database link linkData *> set type=OracleLinkFromExternal
delphix database link linkData *> set syncStrategy.config=CDOMSHSR6706
delphix database link linkData *> set environmentUser=oracle
delphix database link linkData *> set linkNow=true
delphix database link linkData *> commit
    `ORACLE_DB_CONTAINER-23
    Dispatched job JOB-171
    DB_LINK job started for "Untitled/mycdb".
    Obtaining information from source database "Untitled/mycdb".
    Creating new TimeFlow for dSource "Untitled/mycdb".
    Request to SnapSync container database "Untitled/mycdb" after attaching it
will be ignored.
    Action: Direct SnapSync of container database is not allowed. Container
database SnapSync will be taken when SnapSync of corresponding
                pluggable database is taken.
    The dSource "mycdb" was successfully linked from source database "Untitled/
mycdb".
    DB_LINK job for "Untitled/mycdb" completed successfully.
```

13.2.4.5.2.3 CDB CLI Attach

Perform the following steps to attach a CDB.

1. Ensure the new source environment has been fully discovered by the Delphix Engine.
2. Login to the engine via SSH with a Delphix admin account.

```
$ ssh admin@YOUR_ENGINE
```

3. Select the CDB dSource database to be attached.

```
delphix> cd database
delphix database> select CDOMLOSR421F
delphix database 'CDOMLOSR421F'>
```

4. Attach the CDB dSource to the correct source CDB in the new environment using the attachSource command.

> When attaching a CDB, the `linkNow=true` command is ignored. Snapshots for CDBs cannot be taken directly. CDB snapshots will be taken when a PDB requires a snapshot of the CDB.

```
delphix database 'CDOMLOSR421F'> attachSource
delphix database 'CDOMLOSR421F' attachSource *> edit attachData
delphix database 'CDOMLOSR421F' attachSource attachData *> set type=OracleAttachData
```

```
delphix database 'CDOMLOSR421F' attachSource attachData *> set config=CDOMLOSR421F
delphix database 'CDOMLOSR421F' attachSource attachData *> set environmentUser=oracle
delphix database 'CDOMLOSR421F' attachSource *> commit
    CDOMLOSR421F
    Dispatched job JOB-46
    DB_ATTACH_SOURCE job started for "Untitled/CDOMLOSR421F".
        Starting validation of attach parameters for database "CDOMLOSR421F".
    Obtaining information from source database "Untitled/CDOMLOSR421F".
    The dSource "CDOMLOSR421F" was successfully linked from source database
"Untitled/CDOMLOSR421F".
```

13.2.4.5.3 CLI cookbook: changing SGA parameter

Below outlines the procedure to change SGA parameter setting on a provisioned VDB.

13.2.4.5.3.1 Procedure

1. Log into the Delphix Management application as `admin` or a user with Admin privileges.
2. Go to `source` and then `select` the name of the VDB that you would like to change the parameters of.
3. You are then going to `update` and edit the `configParams.`
4. Next, you are going to set `sga_target=` correct value.
5. `Commit` the operation so that it saves.

Example

```
ssh admin@enginedelphix > source
delphix source > select "vdb_example"
delphix source "vdb_example" > update
delphix source "vdb_example" *> edit configParams
delphix source "vdb_example" *> set sga_target=new value
delphix source "vdb_example" *> commit
```

> Modifying configuration parameters for a vPDB is not supported, so this workflow will not succeed for a vPDB.

13.2.4.5.4 CLI cookbook: changing the SID of Oracle RAC VDBs

eThis topic describes how to change the SID of instances in an Oracle RAC VDB.

This example demonstrates how to switch the instance name and number between two different hosts, from

```
SQL> select * FROM V$ACTIVE_INSTANCES;
INST_NUMBER INST_NAME
----------- ------------------------------------------------------------
      1 cnrac3:VchiBEB1
      2 cnrac4:VchiBEB2
```

to

```
SQL> select * FROM V$ACTIVE_INSTANCES;
INST_NUMBER INST_NAME
----------- ------------------------------------------------------------
      1 cnrac4:VchiBEB1
      2 cnrac3:VchiBEB2
```

13.2.4.5.4.1 Procedure

1. Stop the VDB through the GUI and login to the Delphix CLI.
2. Select the sourceconfig of the RAC VDB whose instances you would like to rename.

   ```
   kfc-manual.dcenter> sourceconfig
   kfc-manual.dcenter sourceconfig> select Vchicago_BEB
   ```

3. Use the update command to change the properties of the sourceconfig.

   ```
   kfc-manual.dcenter sourceconfig "Vchicago_BEB"> update
   ```

4. Type 'ls' to view the complete list of properties associated with the VDB's sourceconfig. For configurations with larger numbers of RAC instances, the listing may not show the individual instances but will instead display [...]. In order to see the instance configuration, type 'edit instances'.

   ```
   kfc-manual.dcenter sourceconfig "Vchicago_BEB" update *> ls
   Properties
       type: OracleRACConfig
       credentials:
           type: PasswordCredential
           password: ********
       environmentUser: ora1024
       instances:
           0:
               type: OracleRACInstance
               instanceName: VchiBEB1
               instanceNumber: 1
               node: cnrac4
           1:
   ```

```
            type: OracleRACInstance
            instanceName: VchiBEB2
            instanceNumber: 2
            node: cnrac3
    linkingEnabled: true
    nonSysCredentials: (unset)
    nonSysUser: (unset)
    repository: '/u01/app/ora1024/product/10.2.0/db_1'
    services: [ ... ]
    user: delphix
```

5. Use the Set command to change the values for instanceName and instanceNumber for each instance.

```
kfc-manual.dcenter sourceconfig "Vchicago_BEB" update *> set instances.0.instan
ceName=VchiBEB2
kfc-manual.dcenter sourceconfig "Vchicago_BEB" update *> set instances.0.instan
ceNumber=2
kfc-manual.dcenter sourceconfig "Vchicago_BEB" update *> set instances.1.instan
ceName=VchiBEB1
kfc-manual.dcenter sourceconfig "Vchicago_BEB" update *> set instances.1.instan
ceNumber=1
```

6. Finally, commit the changes.

```
kfc-manual.dcenter sourceconfig "Vchicago_BEB" update *> commit;
```

7. Restart the VDB through the GUI for the changes to take effect on the VDB.

13.2.4.5.5 CLI cookbook: toggle new DBID generation upon refresh options for Oracle VDBs

This topic describes how to toggle **Generate new DBID upon refresh** option for Oracle VDBs.

13.2.4.5.5.1 Procedure

1. Login to the Delphix CLI.
2. Go to source.

```
delphix> source
```

3. Select the VDB that you need to update

```
delphix> source select 'VDBOMSRE71EE4_QGE'
```

4. Toggle new DBID value. The new DBID value can be true or false. If the `newDBID` parameter is set to false then the new DBID generation will stop.

```
delphix source VDBOMSRE71EE4_QGE> update
delphix source VDBOMSRE71EE4_QGE> set newDBID=<new value>
```

5. Finally, commit the changes to save the changes.

```
delphix source VDBOMSRE71EE4_QGE> commit
```

6. Refresh the VDB through CLI to generate a new DBID if the `newDBID` parameter was set to true. If the `newDBID` parameter was set to false, the Delphix engine will use the same DBID as the parent.

```
delphix> /database
delphix database>  select 'VDBO_QGE'
delphix database 'VDBO_QGE'> refresh
delphix database 'VDBO_QGE' refresh> commit
```

13.2.4.5.6 CLI cookbook: creating a policy

This will outline how to create a policy in the CLI, please note that you can also do this in the GUI.

13.2.4.5.6.1 Procedure

1. ssh into your Delphix Engine using admin credentials.

```
ssh admin@delphixengine
delphix > ls
```

2. Go to `policies` and `createAndApply` (please note that you cannot just create a policy, you must `createAndApply`, in the GUI you have the option to just create) and set your policy parameter.

```
delphix > policy
delphix policy > createAndApply
delphix policy createAndApply *> set policy.type=< choose from QuotaPolicy,
RefreshPolicy, RetentionPolicy, SnapshotPolicy or SyncPolicy)
delphix policy createAndApply *> set policy.name=< name your policy>
delphix policy createAndApply *> set policy.customized=true
delphix policy createAndApply *> set policy.
```

```
delphix policy createAndApply *> set policy.provisionSource=(LATEST_SNAPSHOT or
LATEST_TIME_FLOW_LOG)
```

Info:
If doing a `RefreshPolicy, SyncPolicy` or `SnapshotPolicy` you are also going to need to add the following:

1.
```
delphix policy createAndApply *> edit policy.scheduleList
delphix policy createAndApply policy.scheduleList * > add
delphix policy createAndApply policy.scheduleList * > set cronString=
delphix policy createAndApply policy.scheduleList * > set cutoffTime=
delphix policy createAndApply policy.scheduleList * > back
```

3. Set your target parameters which are going to be a container, group etc.

```
delphix policy createAndApply *> set target=
```

4. Verify and `commit.`

```
delphix policy createAndApply *> ls
delphix policy createAndApply *> commit
```

13.2.4.5.7 CLI cookbook: creating a VDB config template

This topic will address how to create a VDB Config Template in the CLI; this functionality is also available in the GUI.

13.2.4.5.7.1 Procedure

1. ssh into your Delphix Engine using admin credentials.

```
ssh admin@<yourdelphixengine>
```

2. Go to `database` and then `template` and then `create.`

```
delphix > database template
delphix database template > create
```

```
delphix database template create *> set name=<set the name of database
template>
delphix database template create *> set parameters.<set parameters you want>
delphix database template create *> set sourceType=<set the source>
```

3. Verify information and commit.

```
delphix database template create *> ls
delphix database template create *> commit
```

Example for Oracle:

```
ssh admin@testenginetestengine > database templatetestengine
database template > createtestengine
database template create *> set name=oracletestengine
database template create > set parameters.PGA_TARGET=100Mtestengine
database template create > set parameters.MEMORY_TARGET=100Mtestengine
database template create > set sourceType=OracleVirtualSourcetestengine
database template create > ls
  Properties
  type: DatabaseTemplate
  name: oracle ()
  description: (unset)
  parameters:
    MEMORY_TARGET: 100M ()
    PGA_TARGET: 100M ()
  sourceType: OracleVirtualSource ()testengine
database template create *> commit
```

Example for SQL Server:

```
ssh admin@testengine testengine > database template
 testengine database template > create
testengine database template create *> set name=mssqlTemplate
testengine database template create *> set parameters.READ_COMMITTED_SNAPSHOT=ON
testengine database template create *> set sourceType=MSSqlVirtualSource
testengine database template create *> ls
Properties  type: DatabaseTemplate  name: mssqlTemplate (*)     description: (unset)

    parameters:
         READ_COMMITTED_SNAPSHOT: ON (*)
    sourceType: MSSqlVirtualSource (*)
testengine database template create *> commit
```

13.2.4.5.8 CLI cookbook: determining the snapshot used to provision a VDB

13.2.4.5.8.1 Procedure:

The parent snapshot can be determined using the CLI as follows:

1. Log into the server as a Engine Administrator:

```
ssh admin@<server_ip>
```

2. Select the VDB.

```
delphix> database
delphix database> ls
Objects
NAME          PARENTCONTAINER DESCRIPTION
dSource1      -
dSource2      -
VDB1          dSource1        -
VDB2          dSource2        -
VDB3          dSource1        -
delphix database> select VDB1
```

3. List the VDB parameters, and make a note of the currentTimeflow value.

```
delphix database "VDB1"> ls
Properties
    type: OracleDatabaseContainer
    name: VDB1
    currentTimeflow: VDB1/default
    description: (unset)
    diagnoseNoLoggingFaults: true
    endianness: BIG_ENDIAN
    group: <New Group>
    masked: false
    os: HP-UX
    parentContainer: dSource1
    performanceMode: false
    processor: ia64
    reference: ORACLE_DB_CONTAINER-10
    runtime:
        type: OracleDBContainerRuntime
        logSyncActive: true
    sourcingPolicy:
        type: OracleSourcingPolicy
        encryptedLinkingEnabled: false
```

```
        logsyncEnabled: true
        logsyncInterval: 300
        logsyncMode: ARCHIVE_ONLY_MODE
version:
```

4. Select the Timeflow listed for the VDB.

```
delphix database "VDB1"> /timeflow
delphix timeflow> select VDB1/default
List the timeflow parameters.  The Snapshot used to provision the VDB is listed
as parentSnapshot

delphix timeflow "VDB1/default"> ls
Properties
```

5. List the Timeflow parameters. The Snapshot used to provision the VDB is listed as parentSnapshot.

```
delphix timeflow "VDB1/default"> ls
Properties
type: OracleTimeflow
    name: VDB1/default
    container: VDB1
    parentPoint:
        type: OracleTimeflowPoint
        location: 141285148
        timeflow: dSource1/default
    parentSnapshot: @2013-02-14T15:07:28.491Z
    reference: ORACLE_TIMEFLOW-92572
```

13.2.4.5.9 CLI cookbook: how to refresh a VDB from a specific snapshot

These steps will allow you to refresh a VDB from any specific snapshot, not just the most recent one.

1. Identify the VDB and snapshot that you want to use.

```
ssh admin@<yourengine>
delphix > database ls
delphix database > /snapshot
delphix snapshot > list database=<SOURCEOFSNAPSHOT>
```

2. Go to the `database` and `refresh.`

```
delphix > /database
delphix database > refresh
```

3. Now set what type of refresh you are going to do.

```
delphix database 'VDB' refresh *> set
timeflowPointParameters.type=TimeflowPointSnapshot
```

4. Set the snapshot

```
delphix database 'VDB' refresh *> set timeflowPointParameters.snapshot=@XXXX-
XX-XXTXX:XX:XX.XXXZ
```

5. Commit the action.

```
delphix database 'VDB' refresh *> commit
```

> You can use tab to complete most actions in the CLI in addition to listing the possibilities that are available when setting parameters.

13.2.4.5.10 CLI cookbook: Oracle VDB migration

This topic describes moving a VDB from one environment or installation to another.

VDBs can be moved (or migrated) between hosts by changing the source repository associated with the VDB source config.

13.2.4.5.10.1 Restrictions

The following restrictions apply when migrating a VDB between repositories:

- When migrating a RAC VDB, the host of each `OracleRACInstance` must be updated as well.
- The mount point of the VDB source cannot be changed.
- The `database_unique_name` and `db_name` cannot be changed.
- The new environment and repository must be a compatible target environment.

13.2.4.5.10.2 Procedure

1. Select the source associated with the VDB. By default, sources are named the same as the VDB.

```
delphix> source "vexample"
```

2. Disable the source by running the `disable` command and committing the operation.

```
delphix source "vexample"> disable
delphix source "vexample" disable *> commit
    Dispatched job JOB-171
    SOURCE_DISABLE job started for "vexample".
    Starting disable of virtual database.
    Unexporting storage.
    Virtual database disable successful.
    SOURCE_DISABLE job for "vexample" completed successfully.
delphix source "vexample">
```

3. Select the source config associated with the source. By default this is also the same name as the VDB.

   Update the repository and repository user associated with the source config.

```
delphix source "vexample"> get config
    vexample
delphix source "vexample"> /sourceconfig "vexample"
delphix sourceconfig "vexample">
```

4. Update the repository and repository user associated with the source config.

Warning: You must use the Environment name of the Environment because the repository requires the environment name and Oracle home location. Enable the source.

```
delphix sourceconfig "vexample"> update
delphix sourceconfig "vexample" update *> set repository=192.168.100.247/'/opt/
oracle/product/10.2.0.4/db_1'
delphix sourceconfig "vexample" update *> set environmentUser=192.168.100.247/ora1024
delphix sourceconfig "vexample" update *> commit
delphix sourceconfig "vexample">
```

1. Enable the source.

```
delphix sourceconfig "vexample"> /source "vexample" enable
delphix source "vexample" enable *> commit
    Dispatched job JOB-18
    SOURCE_ENABLE job started for "vexample".
    Enabling dataset "vexample".
    Exporting storage containers from the Delphix Engine.
    Mounting datasets.
    Mounting filesystems for the virtual database instance "1".
    Starting virtual database.
    Starting instance 1 on virtual database "vexample".
    Virtual database "vexample" was successfully started.
```

```
    Dataset "vexample" enabled.
    SOURCE_ENABLE job for "vexample" completed successfully.
delphix sorceconfig "vexample">
```

13.2.4.5.11 CLI cookbook: provisioning a SAP ASE VDB

This topic describes how to provision an SAP ASE VDB using the command line interface.

13.2.4.5.11.1 Prerequisites

You will need the following information:

- The name of the VDB you want to create
- The group in which to create the VDB
- The SAP ASE database name for the VDB
- The source dSource or VDB from which you wish to provision
- The semanticLocation, LSN, or timestamp of the point you want to provision from (if not using the most recent). You can run these commands to get the list of snapshots or timeflow ranges:

  ```
  snapshot list database=dexample
  snapshot list timeflow=dexample
  snapshot list fromDate="2020-03-01T00:00:00.000Z" toDate="2020-03-04T11:31:27.8
  83Z"
  ```
- The target host on which you want to create the VDB. You can list the hosts with the /host list command.
- The source repository (SAP ASE instance on the target host) in which to create the VDB. These can be listed with the /repository list command.

13.2.4.5.11.2 Procedure

1. Execute the database provision command.

   ```
   delphix> database provision
   ```

2. Set the type for the new VDB

   ```
   delphix database provision *> set type=ASEProvisionParameters
   ```

3. Use defaults to fill in most of the information and then customize any additional information that you do not want defaulted, for what information has been filled in after defaults you can do an ls for all fields:

```
delphix database provision *> defaults
delphix database provision *> ls
Properties:
    type: TimeflowPointSemantic
    container: (unset)
    location: LATEST_POINT
delphix database provision *> set container=<dexample>
delphix database provision *> commit
```

4. Set the name and group for the new VDB.

```
delphix database provision *> set container.name=<vexample>
delphix database provision *> set container.group="<New Group>"
```

5. Set the name of the new VDB.

```
delphix database provision *> set sourceConfig.databaseName=<vexample>
```

6. Set the target Dataset Home.

```
delphix database provision *> set sourceConfig.repository=<Dataset Home>
delphix database provision *> set sourceConfig.environmentUser=<Host
environment name/sybase>
```

7. Set the source container from which to provision.

```
delphix database provision *> set timeflowPointParameters.container=<dexample>
```

8. Set the desired value for truncateLogOnCheckpoint

```
delphix database provision *> set truncateLogOnCheckpoint=false
```

9. Commit the configuration and start the DB_PROVISION job

```
delphix database provision *> commit
```

13.2.4.5.12 CLI cookbook: provisioning a single instance non-multitenant Oracle VDB

This topic describes how to provision a single instance non-multitenant Oracle VDB using the Delphix Engine command-line interface.

13.2.4.5.12.1 Prerequisites

You will need the following information:

- The name of the VDB you want to create
- The group in which to create the VDB
- The Oracle database name
- The Oracle database unique name
- The Oracle database instance number
- The Oracle database instance name
- The source dSource or VDB from which you wish to provision. This will be referenced as the "container" in the "defaults" command below.
- The semanticLocation, SCN, or timestamp of the point you want to provision from. You can run these commands to get the list of snapshots or Timeflow ranges:

  ```
  snapshot list database=dexample
  timeflow "dexample" timeflowRanges; commit
  ```

- The base mount point on the target server where VDB data should be mounted
- The source repository (oracle install) in which to create the VDB. These can be listed with the `/repository list` command.
- If you are using a VDB template, the name of the template to use.

13.2.4.5.12.2 Procedure

1. Execute the `database provision` command.

   ```
   delphix> database provision
   ```

2. Execute the `defaults` command. Once you commit this command, it will return a partially constructed provision parameters object.

   ```
   delphix database provision> defaults
   ```

3. Set the Timeflow point source Timeflow and location.

   ```
   delphix database provision defaults *> set type=TimeflowPointSemantic
   delphix database provision defaults *> set container=dexample
   delphix database provision defaults *> set location=LATEST_SNAPSHOT
   ```

4. Commit the operation to populate the defaults, as provided by the browser interface. At this point, the operation can be committed, though you will likely need to change the defaults to match the information.

```
delphix database provision defaults *> commit
```

5. Set the name and group for the new VDB

```
delphix database provision *> set container.name=vexample
delphix database provision *> set container.group=""
```

6. Set the base mount point.

```
delphix database provision *> set source.mountBase=/mnt
```

7. Set the source config type to be a single instance non-multitenant Oracle, and set the database name and database unique name. When provisioning from a RAC or single instance non-multitenant oracle source, the default type will match that of the repository selected by the defaults operation.

```
delphix database provision *> set sourceConfig.type=OracleSIConfig
delphix database provision *> set sourceConfig.databaseName=vexample
delphix database provision *> set sourceConfig.uniqueName=vexample123
```

8. Set the instance name and number.

```
delphix database provision *> edit sourceConfig.instance
delphix database provision sourceConfig.instance *> set instanceNumber=1
delphix database provision sourceConfig.instance *> set instanceName=vexample
delphix database provision sourceConfig.instance *> back
```

9. Set the target repository.

```
delphix database provision *> set sourceConfig.repository='/env/opt/oracle'
```

10. Configure the Oracle database parameters. If you are using manually specified parameters, you can set the contents of `source.configParams`. If you want to use a template, you can set `source.configTemplate`.
11. (Optional) Configure customer environment variables.
Setting Environment Pair Values

```
delphix database provision *> edit customEnvVars
```

```
delphix database provision source.customEnvVars *> add
delphix database provision source.customEnvVars 0 *> set
type=OracleCustomEnvVarSIPair
delphix database provision source.customEnvVars 0 *> set varName=MYVAR1
delphix database provision source.customEnvVars 0 *> set varValue=Value1
delphix database provision source.customEnvVars 0 *> back
delphix database provision source.customEnvVars 1 *> set
type=OracleCustomEnvVarSIPair
delphix database provision source.customEnvVars 1 *> set varName=MYVAR2
delphix database provision source.customEnvVars 1 *> set varValue=Value2
delphix database provision source.customEnvVars 1 *> back
```

Note:

1.

OracleCustomEnvVarSIFile file should be in the following format.

- ```
  export VARNAME1=VARVALUE1export VARNAME2=VARVALUE2
  ```

2. When provisioning to a RAC target, use the type as **OracleCustomEnvVarRACPair or OracleCustomEnvVarRACFile.**

- ```
  delphix database provision source.customEnvVars 0 *> set
  type=OracleCustomEnvVarRACPairdelphix database provision
  source.customEnvVars 0 *> set clusterNode=targetnode1delphix database
  provision source.customEnvVars 0 *> set varName=MYVAR1delphix database
  provision source.customEnvVars 0 *> set varValue=Value1delphix database
  provision source.customEnvVars 0 *> back
  ```

12. Commit the result.

```
delphix database provision *> commit
```

13.2.4.5.13 CLI cookbook: provisioning a SQL server VDB

This topic describes how to provision a SQL Server VDB using the command line interface.

13.2.4.5.13.1 Prerequisites

You will need the following information:

- The name of the VDB you want to create

- The group in which to create the VDB
- The SQL Server database name for the VDB
- The source dSource or VDB from which you wish to provision
- The semanticLocation, LSN, or timestamp of the point you want to provision from. You can run these commands to get the list of snapshots or timeflow ranges:

```
snapshot list database=dexample
snapshot list timeflow=dexample
snapshot list fromDate="2020-03-01T00:00:00.000Z" toDate="2020-03-04T11:31:27.8
83Z"
```

- The target host on which you want to create the VDB. You can list the hosts with the `/host list` command.
- The source repository (SQL Server instance on the target host) in which to create the VDB. These can be listed with the `/repository list` command.

13.2.4.5.13.2 Procedure

1. Execute the `database provision` command.

```
delphix> database provision
```

2. Execute the `defaults` command.

```
delphix database provision> defaults
```

3. Set the timeflow point source timeflow and location.

```
delphix database provision defaults *> set type=TimeflowPointSemantic
delphix database provision defaults *> set container=dexample
delphix database provision defaults *> set location=LATEST_SNAPSHOT
```

4. Commit the operation to populate the defaults, as provided by the browser interface. At this point, the operation can be committed, though you will likely need to change the defaults to match the information.

```
delphix database provision defaults *> commit
```

5. Set the name and group for the new VDB.

```
delphix database provision *> set container.name=vexample
```

```
delphix database provision *> set container.group=""
```

6. Set the database name for the VDB on the target SQL Server instance.

```
delphix database provision *> set sourceConfig.databaseName=vexample
```

7. Set the target repository

```
delphix database provision *> set sourceConfig.repository=targetEnv/
SQLServer2008
```

8. Commit the result.

```
delphix database provision *> commit
```

13.2.4.5.14 CLI cookbook: provisioning a VDB from a timeflow bookmark

This topic describes how to create a Timeflow bookmark and use it to provision a single instance Oracle VDB using the Delphix Engine command-line interface.

You can create Timeflow bookmarks to give a semantically meaningful name to a TimeFlow point (scn, location or timestamp within a Timeflow). You can then use the bookmarks you created to execute the following database operations:

- Provision
- Refresh
- Export
- Test file mappings
- VDB Rewind

13.2.4.5.14.1 Prerequisites

You will need the following information:

- The name of the Timeflow bookmark you want to create
- The name of the VDB you want to create
- The group in which to create the VDB
- The Oracle database name
- The Oracle database unique name
- The Oracle database instance number
- The Oracle database instance name
- The source dSource or VDB from which you wish to provision
- The SCN, or timestamp of the point you want to provision from. You can run these commands to get the list of snapshots or Timeflow ranges:

```
snapshot list database=dexample
timeflow "dexample" timeflowRanges; commit
```

- The base mountpoint on the target server where VDB data should be mounted
- The source repository (oracle install) in which to create the VDB. These can be listed with the `/repository list` command.

13.2.4.5.14.2 Creating the timeflow bookmark

1. Execute the `timeflow bookmark create` command.

```
delphix> timeflow bookmark create
```

2. Set the timeflow point to be Oracle timeflow point.

```
delphix timeflow bookmark create *> set timeflowPoint.type=OracleTimeflowPoint
```

3. Set the timeflow point timeflow and location

```
delphix timeflow bookmark create *> set timeflowPoint.timeflow=dexample/default
delphix timeflow bookmark create *> set timeflowPoint.location=1945519455791
```

4. Set the name of the timeflow bookmark

```
delphix timeflow bookmark create *> set name=myTimeFlowBookmark
```

5. Commit the result

```
delphix timeflow bookmark create *> commit
    TIMEFLOW_BOOKMARK-1
```

6. Display the list of timeflow bookmarks

```
delphix> timeflow bookmark ls
Objects
NAME                TAG    TIMEFLOW
myTimeFlowBookmark  -      dexample/default
Operations
create
```

13.2.4.5.14.3 Provisioning from a TimeFlow bookmark

1. Execute the `database provision` command.

```
delphix> database provision
```

2. Set `defaults` and provide container (VDB or dSource) that you will be provisioning from

```
delphix database provision > defaults
delphix database provision defaults > set container=<VDB or dSource>
delphix database provision defaults > commit
```

3. Set the timeflowPointParameters type to be TimeflowBookmark.

```
delphix database provision *> set
timeflowPointParameters.type=TimeflowPointBookmark
```

4. Set the timeflow bookmark.

```
database provision *> set timeflowPointParameters.bookmark=myTimeFlowBookmark
```

5. Set the name and group for the new VDB.

```
delphix database provision *> set container.name=vexample
delphix database provision *> set container.group="Untitled"
```

6. Set the base mountpoint

```
delphix database provision *> set source.mountBase=/mnt
```

7. Set the source config type to be single instance Oracle, and set the database name and database unique name.

```
delphix database provision *> set sourceConfig.type=OracleSIConfig
delphix database provision *> set sourceConfig.databaseName=vexample
delphix database provision *> set sourceConfig.uniqueName=vexample123
```

8. Set the instance name and number.

```
delphix database provision *> edit sourceConfig.instance
```

```
delphix database provision sourceConfig.instance *> set instanceNumber=1
delphix database provision sourceConfig.instance *> set instanceName=vexample
delphix database provision sourceConfig.instance *> back
```

9. Set the target repository.

```
delphix database provision *> set sourceConfig.repository=env/'/opt/oracle'
```

10. Commit the result.

```
delphix database provision *> commit
```

13.2.4.5.15 CLI cookbook: provisioning a virtual PDB to a new virtual CDB

This topic describes how to provision a virtual pluggable database (vPDB) to a virtual container database (vCDB) using the command-line interface.

> This process applies to Oracle container databases (CDBs) and the pluggable databases (PDBs) found within them that have been linked to Delphix as dSources. As container databases can only be found in Oracle 12.1.0.1 releases and above of Oracle, this topic is relevant to only these releases.

13.2.4.5.15.1 Prerequisites

The provisioning of virtual PDBs in Delphix is dependent on the following entities being in place prior to attempting the creation of this type of Delphix dataset.

- A source container database must be linked to the engine.
- The source container database must have pluggable databases within it linked as dSources to Delphix.
- A destination host with an Oracle home of the same release as the source database must be available and already discovered as an Environment in Delphix.

In the example CLI provision detailed below the following databases are in place:

- The source container database is called "CDOMLOSR1TB"
- The pluggable database found in "CDOMLOSR1TB" that has been linked as a dSource is called "CDOMLOSR1TBPDB1"
- The destination virtual container database will be called "cdbvirt"
- Within the Delphix Engine CLI the name of the repository (target Oracle home) on the target system is '/u01/app/oracle/product/12.2.0.1/dbhome_1'
- The destination virtual pluggable database will be called "PDBS3" and run from virtual container database "cdbvirt"

- Delphix creates a temporary container database on the target host which will be used to establish the virtualized copy of the dSource PDB "pdborcl". This temporary CDB will be created and running during the provisioning process, it will be destroyed at the end of the provision of the virtual PDB.

13.2.4.5.15.2 Procedure

1. Log into the Delphix command-line interface using the admin user or a user with admin privileges.

```
$ ssh admin@YOUR_ENGINE
```

2. Move to the database provisioning command line object.

```
delphix> database provision
```

3. Give the dataset a name.

```
delphix database provision *> set container.name=PDBS3
```

4. Place the new dataset in a Group that appears in the Delphix GUI, in this case, the Targets group.

```
delphix database provision *> set container.group=Targets
```

5. Set the type of provision to perform, for Oracle virtual database (VDB) - VDB/vPDBs, the type will be OracleVirtualPdbSource.

```
delphix database provision *> set source.type=OracleVirtualPdbSource
```

6. Set the destination mount point which Delphix NFS mounts are to be linked to under the virtual PDB. This folder must exist at a file system level on the target host. Do not use single quotes around the mount path.

```
delphix database provision *> set source.mountBase="/mnt/provision"
```

7. Name the vPDB. This is what it will appear as in the destination container database.

```
delphix database provision *> set sourceConfig.databaseName=PDBS3
```

8. Supply the dSource PDBs details. In this example, the provision will use the latest point in time available to the dSource PDB as the point in time from which to provision the vPDB. Setting a different timeflowPointParameters.type would allow you to use points in time other than the latest

snapshot or latest point in time if this is what you desire. Using other types is not covered in this example

```
delphix database provision *> set
timeflowPointParameters.container=CDOMLOSR1TBPDB1
```

9. Give the virtual CDB a name:

```
delphix database provision *> set virtualCdb.container.name=cdbvirt
```

10. Place the virtual CDB in a Group that appears in the Delphix GUI, in this case, the target group.

```
delphix database provision *> set virtualCdb.container.group=Targets
```

11. If automatically restarting the vPDB and vCDB is not required after a reboot of the target host, set this to option to false.

```
delphix database provision *> set virtualCdb.source.
allowAutoVDBRestartOnHostReboot=false
```

12. Set the destination mount point which Delphix NFS mounts are to be linked to under the virtual CDB. This folder must exist at a file system level on the target host. Do not use single quotes around the mount path.

```
delphix database provision *> set virtualCdb.source.mountBase="/mnt/provision"
```

13. Set the vCDB configuration type to one of OracleRACConfig or OracleSIConfig. RAC configurations are not covered in this article:

```
delphix database provision *> set virtualCdb.sourceConfig.type=OracleSIConfig
```

14. Set the database name for the virtual CDB:

```
delphix database provision *> set virtualCdb.sourceConfig.databaseName=cdbvirt
```

15. Set the target system environment user that while be used to run the virtual CDB:

```
delphix database provision *> set
virtualCdb.sourceConfig.environmentUser=oracle
```

16. Set the instance name and number for the virtual CDB

```
delphix database provision *> set
virtualCdb.sourceConfig.instance.instanceName=cdbvirt
delphix database provision *> set
virtualCdb.sourceConfig.instance.instanceNumber=1
```

17. Set the repository (Oracle home) on the target system that will be used to run the virtual CDB. In this example the Oracle home is */u01/app/oracle/product/12.2.0.1/dbhome_1*:

```
delphix database provision *> set virtualCdb.sourceConfig.repository='/u01/app/
oracle/product/12.2.0.1/dbhome_1'
```

18. Set the database unique name for the virtual CDB:

```
delphix database provision *> set virtualCdb.sourceConfig.uniqueName=cdbvirt
```

19. Set the following two parameters to true or false, depending on whether the vCDB and vPDB should be restarted automatically following a target host reboot:

```
delphix database provision *> set source.allowAutoVDBRestartOnHostReboot=true
delphix database provision *> set virtualCdb.source.
allowAutoVDBRestartOnHostReboot=true
```

20. Check that all the settings you require are in place using the "ls" command

```
delphix database provision *> ls
Properties
    type: OracleMultitenantProvisionParameters
    container:
        type: OracleDatabaseContainer
        name: PDBS3 (*)
        description: (unset)
        diagnoseNoLoggingFaults: true
        group: Untitled (*)
        performanceMode: DISABLED
        preProvisioningEnabled: false
        sourcingPolicy: (unset)
    credential: (unset)
    masked: (unset)
    maskingJob: (unset)
    source:
        type: OracleVirtualPdbSource
        name: (unset)
        allowAutoVDBRestartOnHostReboot: true (*)
        config: (unset)
        customEnvVars: (unset)
```

```
        fileMappingRules: (unset)
        logCollectionEnabled: false
        mountBase: /mnt/provision (*)
        operations: (unset)
    sourceConfig:
        type: OraclePDBConfig
        cdbConfig: (unset)
        databaseName: cdbvirt (*)
        environmentUser: (unset)
        linkingEnabled: true
        nonSysCredentials: (unset)
        nonSysUser: (unset)
        repository: (unset)
        services: (unset)
    timeflowPointParameters:
        type: TimeflowPointSemantic
        container: CDOMLOSR1TBPDB1 (*)
        location: LATEST_POINT
    username: (unset)
    virtualCdb:
        type: OracleVirtualCdbProvisionParameters (*)
        container:
            type: OracleDatabaseContainer (*)
            name: cdbvirt (*)
            description: (unset)
            diagnoseNoLoggingFaults: true (*)
            group: Untitled (*)
            performanceMode: DISABLED (*)
            preProvisioningEnabled: false (*)
            sourcingPolicy: (unset)
        source:
            type: OracleVirtualCdbSource (*)
            name: (unset)
            allowAutoVDBRestartOnHostReboot: true (*)
            config: (unset)
            configParams: (unset)
            configTemplate: vcdb (*)
            logCollectionEnabled: false (*)
            mountBase: /mnt/provision (*)
        sourceConfig:
            type: OracleSIConfig (*)
            databaseName: cdbvirt (*)
            environmentUser: (unset)
            instance:
                type: OracleInstance (*)
                instanceName: cdbvirt (*)
                instanceNumber: 1 (*)
            linkingEnabled: true (*)
            nonSysCredentials: (unset)
            nonSysUser: (unset)
            repository: '/u01/app/oracle/product/12.2.0.1/dbhome_1' (*)
            services: (unset)
```

```
            tdeKeystorePassword: (unset)
            uniqueName: cdbvirt (*)

Operations
defaults
```

21. Initiate the provision by committing the operation in the CLI.

```
delphix database provision *> commit
    PDBS3
    Dispatched job JOB-24
    DB_PROVISION job started for "Targets/PDBS3".
    Starting provision of the virtual database "PDBS3".
    Preparing multitenant container database "cdbvirt".
    Creating new TimeFlow.
    Generating recovery scripts.
    Exporting storage.
    Mounting filesystems to recover pluggable database on instance "1".
    Mounting read-only archive log filesystem for the virtual database instance
"1".
    Backing up Oracle spfile.
    Mounting virtual database instance.
    Disabling flashback on Oracle database.
    Renaming Oracle datafiles.
    Mounting virtual database instance.
    Recovering Oracle pluggable database.
    Creating control file.
    Creating Oracle online logs.
    Processing startup init file.
    Configuring initialization and server parameter files.
    Mounting virtual database instance.
    Performing Oracle resetlogs.
    Creating tempfiles.
    Renaming readonly datafiles.
    Finalizing Oracle pluggable database.
    Registering listeners.
    Unmounting read-only archive log filesystem for the virtual database
instance "1".
    Unmounting filesystems after recovering pluggable database on instance "1".
    Plugging in Oracle pluggable database.
    Opening Oracle pluggable database.
    Setting OMF destination for Oracle pluggable database.
    Creating tempfiles.
    Online readonly tablespaces.
    Enabling Oracle instances.
    Checking Oracle pluggable database plugin violations.
    DB_PROVISION job for "Targets/PDBS3" completed successfully.
```

13.2.4.5.16 CLI cookbook: provisioning a virtual PDB in a target CDB

This topic describes how to provision a virtual pluggable database (vPDB) using the command-line interface.

> This process applies to Oracle container databases (CDBs) and the pluggable databases (PDBs) found within them that have been linked to Delphix as dSources. As container databases can only be found in Oracle **12.1.0.1** releases and above of Oracle this topic is relevant to only these releases.

13.2.4.5.16.1 Prerequisites

Provisioning of virtual PDB's in Delphix is dependent on the following entities being in place prior to attempting the creation of this type of Delphix dataset.

- A source container database must be linked to the engine.
- The source container database must have pluggable databases within it linked as dSources to Delphix.
- A destination container database of the same Oracle release as the source database must exist in the target host. If this container database is a physical database, then it must be linked to Delphix. If it’s a virtual container database, then it must be of Oracle version 12.1.0.2 or later.

In the example CLI provision detailed below the following databases are in place:

- The source container database is called "cdb12"
- The destination container database is called "cdbstage"
- The pluggable database found in "cdb12" that has been linked as a dSource is called "pdborcl"
- The destination virtual pluggable database will be called "pdbs2" and run from container database "cdbstage"

Delphix creates a temporary container database on the target host which will be used to establish the virtualized copy of the dSource PDB "pdborcl". This temporary CDB will be created and running during the provisioning process, it will be destroyed at the end of the provision of the virtual PDB.

13.2.4.5.16.2 Procedure

1. Log into the Delphix command-line interface using the admin user or a user with admin privileges.

   ```
   $ ssh admin@YOUR_ENGINE
   ```

2. Set the type of provision to perform, for Oracle virtual database (VDB) - VDB/vPDBs, the type will be OracleVirtualSource.

```
set source.type=OracleVirtualPdbSource
```

3. Set the destination target environment/host through setting the sourceConfig environment user to perform the provision. A destination container database must already be running on this target host.

```
delphix database provision *> set sourceConfig.environmentUser=OEL6SIN1/delphix
```

4. Set the destination mount point which Delphix NFS mounts are to be linked to under the virtual PDB. This folder must exist at a file system level on the target host. Do not use single quotes around the mount path.

```
delphix database provision *> set source.mountBase="/mnt/provision"
```

5. Set the login details for the provision and Delphix OS user who is to perform the provision.

```
delphix database provision *> set username=delphix
delphix database provision *> set credential.type=PasswordCredential
delphix database provision *> set credential.password=delphix
```

6. Give the dataset a name.

```
delphix database provision *> set container.name=PDBS2
```

7. Place the new dataset in a Group that appears in the Delphix GUI, in this case, the target group.

```
delphix database provision *> set container.group=Targets
```

8. If automatically restarting the VDB is not required after a reboot of the VDB target host, set this to option to false. False is possibly a better option given the container database would need to be running prior to any attempt to pull up a vPDB.

```
delphix database provision *> set source.allowAutoVDBRestartOnHostReboot=false
```

9. Supply the destination container database name. This will be where the vPDB will ultimately be placed and run from on the target host.

```
delphix database provision *> set sourceConfig.cdbConfig=cdbstage
```

10. Name the vPDB. This is what it will appear as in the destination container database.

```
delphix database provision *> set sourceConfig.databaseName=pdbs2
```

11. Supply the dSource PDBs details. In this example, the provision will use the latest point in time available to the dSource PDB as the point in time from which to provision the vPDB. Setting a different timeflowPointParameters.type would allow you to use points in time other than the latest snapshot or latest point in time if this is what you desire. Using other types is not covered in this example.

```
delphix database provision *> set timeflowPointParameters.container=PDBORCL
```

12. Check that all the settings you require are in place using the "ls" command.

```
delphix database provision *> ls
Properties
  type: OracleProvisionParameters
  container:
    type: OracleDatabaseContainer
    name: PDBS2 (*)
    description: (unset)
    diagnoseNoLoggingFaults: true
    group: Targets (*)
    performanceMode: DISABLED
    preProvisioningEnabled: false
    sourcingPolicy: (unset)
  credential:
    type: PasswordCredential (*)
    password: ******** (*)
  maskingJob: (unset)
  newDBID: true (*)
  openResetlogs: true
  physicalStandby: false
  source:
    type: OracleVirtualSource (*)
    name: (unset)
    allowAutoVDBRestartOnHostReboot: false (*)
    archivelogMode: true
    config: (unset)
    configParams: (unset)
    configTemplate: (unset)
    customEnvVars: (unset)
    fileMappingRules: (unset)
    manualProvisioning: false
    mountBase: /mnt/provision (*)
    nodeListenerList: (unset)
    operations: (unset)
    redoLogGroups: 3
    redoLogSizeInMB: 0
  sourceConfig:
```

```
    type: OraclePDBConfig
    cdbConfig: cdbstage (*)
    databaseName: pdbs2 (*)
    environmentUser: (unset)
    linkingEnabled: true
    repository: (unset)
    services: (unset)
  timeflowPointParameters:
    type: TimeflowPointSemantic
    container: PDBORCL (*)
    location: LATEST_POINT
  username: delphix (*)
  Operationsdefaults
```

13. Initiate the provision by committing the operation in the CLI.

```
delphix database provision *> commit
  PDBS2
  Dispatched job JOB-333
  DB_PROVISION job started for "Targets/PDBS2".
  Starting provision of the virtual database "pdbs2".
  Preparing multitenant container database "cdbstage".
  Creating new TimeFlow.  Generating recovery scripts.
  Exporting storage.
  Mounting filesystems to recover pluggable database on instance "1".
  Mounting read-only archive log filesystem for the virtual database instance
"1".
  Recovering Oracle pluggable database.
  Mounting filesystems for the virtual database instance "1".
  Unmounting filesystems after recovering pluggable database on instance "1".
  Cleaning up objects created by pluggable database provisioning.
  Opening Oracle pluggable database.
  DB_PROVISION job for "Targets/PDBS2" completed successfully.
```

13.2.4.5.17 CLI cookbook: Provisioning a TDE-enabled virtual PDB to a new virtual CDB

This topic describes how to provision a TDE-enabled virtual pluggable database (vPDB) to a virtual container database (vCDB) using the command-line interface.

> This process applies to Oracle version 12.2.0.1 or later versions.

13.2.4.5.17.1 Prerequisites

The prerequisites are the same as described in CLI Cookbook: Provisioning a Virtual PDB to a new virtual CDB (see page 2642), additionally the following are the extra prerequisites:

- TDE must be configured for the source container database before it's linked to the engine.
- The source PDB must have TDE configured before it's linked as dSources to Delphix.
- The keystore file of the source container database must be accessible from the target host. If the target database is running in a RAC environment, the keystore file of the source container database must be accessible from all target nodes.
- If the target database is running in a RAC environment, **TDE Keystores Root** must be set for each node.

In the example CLI provision detailed below, assuming:

- The source container database **TDE Keystore Password** is `mySrcCdbTdePwd`.
- The source container database keystore file can be accessed from the target host(s) with path `/u01/app/oracle/keystores/cdb12/wallet`.
- The new vCDB's keystore file will be created under the folder `/u01/app/oracle/keystores/cdbvirt/wallet`.
- The new vCDB's **TDE Keystore Password** is `myVcdbTdePwd`.
- The vPDB's **TDE Secret for Exported Keys** is `myVpdbTdeSecret`.

For more information about TDE parameters, please refer to Provisioning a TDE-enabled vPDB (see page 1549).

13.2.4.5.17.2 Procedure

After following all steps in the **Procedure** section of CLI Cookbook: Provisioning a Virtual PDB to a new virtual CDB (see page 2642) to set provision parameters, set TDE-related parameters as follows before commit:

1. Set `parentTdeKeystorePath`, which is the path used to access the source CDB's TDE keystore file from target host(s).

```
delphix database provision *> set source.parentTdeKeystorePath=/u01/app/oracle/
keystores/cdb12/wallet
```

2. Set `parentTdeKeystorePassword`, which is the password of the source CDB's TDE keystore.

```
delphix database provision *> set
source.parentTdeKeystorePassword=mySrcCdbTdePwd
```

3. Supply `tdeExportedKeyFileSecret`, which is the password used for exporting the vPDB's keys to keyfile.

```
delphix database provision *> set
source.tdeExportedKeyFileSecret=myVpdbTdeSecret
```

4. Set `targetVcdbTdeKeystorePath`, which is the folder where the new vCDB's TDE keystore file will be created.

```
delphix database provision *> set source.targetVcdbTdeKeystorePath=/u01/app/
oracle/keystores/cdbvirt/wallet
```

5. Set `targetVcdbTdeKeystorePassword`, which is the password of the new vCDB's TDE keystore.

```
delphix database provision *> set
source.targetVcdbTdeKeystorePassword=myVcdbTdePwd
```

After all parameters are set, initiate the provision by committing the operation in the CLI:

```
delphix database provision *> commit
```

13.2.4.5.18 CLI cookbook: Provisioning a TDE-enabled vPDB in a target CDB

This topic describes how to provision a TDE-enabled virtual pluggable database (vPDB) in a target CDB (a linked CDB or existing vCDB) using the command-line interface.

> This process applies to Oracle version 12.2.0.1 or later versions.

13.2.4.5.18.1 Prerequisites

The prerequisites are the same as described in CLI Cookbook: Provisioning a Virtual PDB in a Target CDB (see page 2648), plus the following extra prerequisites:

- TDE must be configured for the source container database before it's linked to the engine.
- The source PDB must have TDE configured before it's linked as dSources to Delphix.
- The keystore file of the source container database must be accessible from the target host. If the target database is running in a RAC environment, the keystore file of the source container database must be accessible from all target nodes.
- **TDE Keystore Password** must be set for the target CDB.
- If the target database is running in a RAC environment, **TDE Keystores Root** must be set for each node.

In the example CLI provision detailed below, assuming:

- The source container database **TDE Keystore Password** is `mySrcCdbTdePwd`.
- The source container database keystore file can be accessed from the target host(s) with path `/u01/app/oracle/keystores/cdb12/wallet`.

- The vPDB's **TDE Secret for Exported Keys** is `myVpdbTdeSecret`.

For more information about TDE parameters, please refer to Provisioning a TDE-enabled vPDB (see page 1549).

13.2.4.5.18.2 Procedure

After following all steps in the **Procedure** section of CLI Cookbook: Provisioning a Virtual PDB in a Target CDB (see page 2648) to set provision parameters, set TDE-related parameters as follows before the commit:

1. Set `parentTdeKeystorePath`, which is the path used to access the source CDB's TDE keystore file from the target host(s).

   ```
   delphix database provision *> set source.parentTdeKeystorePath=/u01/app/oracle/
   keystores/cdb12/wallet
   ```

2. Set `parentTdeKeystorePassword`, which is the password of the source CDB's TDE keystore.

   ```
   delphix database provision *> set
   source.parentTdeKeystorePassword=mySrcCdbTdePwd
   ```

3. Supply `tdeExportedKeyFileSecret`, which is the password used for exporting the vPDB's keys to the keyfile.

   ```
   delphix database provision *> set
   source.tdeExportedKeyFileSecret=myVpdbTdeSecret
   ```

After all the parameters are set, initiate the provision by committing the operation in the CLI:

```
delphix database provision *> commit
```

13.2.4.5.19 CLI cookbook: refresh a VDB from a specific timepoint or latest

This topic describes the steps to Refresh a VDB from a specific Timepoint or from Latest.

You can refresh from any point on Timeflow using SCN, location, or timestamp.

13.2.4.5.19.1 Prerequisites

You will need the following information:

- The VDB name
- The TimeFlow location, SCN, or timestamp of the point you want to provision from.

The default domain user created on Delphix Engines is now **admin** instead of delphix_admin. When engines created before 5.3.1 are upgraded to 5.3.1 or later they will retain their old username 'delphix_admin'. To avoid complications Delphix recommends creating users with an admin role and then Disabling delphix_admin.

Log in to CLI:

```
$ ssh admin@<delphixengine>
```

Determine the timeflow

Run:

```
> timeflow "<dSource>" timeflowRanges
> commit
> cd
```

Perform the refresh from specific timepoint

```
> database
 > select <VDB name>
  > refresh
  > set timeflowPointParameters.type= (one of TimeflowPointBookmark,
TimeflowPointBookmarkTag, TimeflowPointLocation, TimeflowPointSemantic,
TimeflowPointTimestamp as appropriate)
  > set timeflowPointParameters.location= (the location, timestamp, or bookmark you
wish to refresh to)
  > set timeflowPointParameters.timeflow= (the timeflow associated with location)>
commit
```

Perform the refresh from latest

```
> database
> select <yourdatabase>
> refresh
> set timeflowPointParameters.container= (Parent of VDB)
> commit
```

13.2.4.5.20 CLI cookbook: repairing a timeflow

13.2.4.5.20.1 Prerequisites

- Know the dSource and Group you need to repair from
- Make sure that your retention policy is set correctly so that the ingested logs are within the retention

13.2.4.5.20.2 Procedure

1. Log into the Delphix Engine as an Admin user. Go to Timeflow and then list. Find the name of the Timeflow that needs to be repaired.

```
ssh admin@<yourengine>
delphix > timeflow
delphix timeflow > ls
```

2. List the missing logs for the Timeflow. The maximum number of logs reported is controlled by the value of the pageSize argument; if there are a very large number of missing logs, you may need to increase this value. Note the start and end scn of the missing log.

```
delphix timeflow > cd oracle/log
delphix timeflow oracle log> list timeflow=example missing=true pageSize=1000
```

> **Note:** If the output from the above command does not list any logs as missing, but there are one or more sequences reported as missing by the **Delphix Admin** browser-based app, please contact Delphix Technical Support for assistance.

3. Stage the missing logs.
   a. Verify that there is sufficient free space.
   b. Copy or restore the missing archive logs into an alternative directory on a server the Delphix Engine can access via the network. All files and subdirectories in the configured directory are searched recursively, so staging the archive logs to a location with no other files or folders will ensure the process completes in a timely manner.
   c. Verify that the user being specified in the next step has permission to read these archive log files in the directory. The user credentials are optional if the host and user credentials have already been added to the Delphix Engine.

```
delphix timeflow oracle log  > fetch
delphix timeflow oracle log fetch *> set type=TimeflowLogFetchParameters
delphix timeflow oracle log fetch *> set timeflow=example
delphix timeflow oracle log fetch *> set directory=[directory where you
restored the log file]
delphix timeflow oracle log fetch *> set endLocation=[end SCN of the
sequence]
delphix timeflow oracle log fetch *> set startLocation=[start SCN of the
sequence]
delphix timeflow oracle log fetch *> set host=[hostname or IP of the
host you restored the file to]
delphix timeflow oracle log fetch *> set username=[a user that can read
the file]
delphix timeflow oracle log fetch *> edit credentials
delphix timeflow oracle log fetch *> set type=PasswordCredential
delphix timeflow oracle log fetch *> set password=[password for this
 user]
```

4. Commit the changes.

```
delphix timeflow oracle log  > fetch
delphix timeflow oracle log fetch *> set type=TimeflowLogFetchParameters
delphix timeflow oracle log fetch *> set timeflow=example
delphix timeflow oracle log fetch *> set directory=[directory where you
restored the log file]
delphix timeflow oracle log fetch *> set endLocation=[end SCN of the sequence]
delphix timeflow oracle log fetch *> set startLocation=[start SCN of the
sequence]
delphix timeflow oracle log fetch *> set host=[hostname or IP of the host you
restored the file to]
delphix timeflow oracle log fetch *> set username=[a user that can read the
file]
delphix timeflow oracle log fetch *> edit credentials
delphix timeflow oracle log fetch *> set type=PasswordCredential
delphix timeflow oracle log fetch *> set password=[password for this user]
```

Only do ONE repair job at a time.

It is possible for there to be more than one Timeflow visible for a given container/source. If that is the case, you can verify the current TimeFlow being used with:

```
delphix > database
```

```
delphix database > select 'example'
delphix database "example"> ls
```

Look for the `currentTimeflow` value.

13.2.4.5.21 CLI cookbook: rolling back a VDB

The following sections provide examples of how to rollback or rewind your VDB.

- Rolling back or rewinding to a snapshot from a VDB (see page 2658)
- Rolling back or rewinding to a snapshot using timeflow (see page 2658)
- Rolling back or rewinding to a timeflow bookmark (see page 2659)

13.2.4.5.21.1 Rolling back or rewinding to a snapshot from a VDB

1. Log into the Delphix Engine.

```
ssh admin@delphix_engine
```

2. List Timeflows for the database that you want to rollback to.

```
de > ls
de > timeflow
de timeflow > ls
```

3. Switch to the VDB you want to rollback.

```
de timeflow > cd /database
de database > ls
de database > select "vdb_example"
```

4. Rollback VDB using the VDB rollback function (note this can also be done in the GUI).

```
de database 'vdb_example' > rollback
de database 'vdb_example' rollback *> set timeflowPointParameters.container=
de database 'vdb_example' rollback *> set timeflowPointParameters.location=
de database 'vdb_example' rollback *> commit
```

13.2.4.5.21.2 Rolling back or rewinding to a snapshot using timeflow

1. Log into the Delphix Engine.

```
ssh admin@delphix_engine
```

2. List Timeflows for the database that you want to rollback to.

```
de > ls
de > timeflow
de timeflow > ls
```

3. Switch to the VDB you want to rollback.

```
de timeflow > cd /database
de database > ls
de database > select "vdb_example"
```

4. Use the switchTimeflow operation.

```
de database 'vdb_example' > switchTimeflow
de database 'vdb_example' switchTimeflow *> set timeflow=<different timeflow>
de database 'vdb_example' switchTimeflow *> commit
```

13.2.4.5.21.3 Rolling back or rewinding to a timeflow bookmark

Requirements: Know the Timeflow bookmark that you want to use.

1. Log into the Delphix Engine.

```
ssh admin@<yourengine>
```

2. Retrieve database and Timeflow information that you would like to rewind/rollback to.

```
delphix > ls
delphix database > select "dexample"
delphix database "dexample" > get currentTimeflow
```

3. Rollback/Rewind VDB.

```
delphix database "dexample" > rollback
delphix database "dexample" rollback *> ls
delphix database "dexample" rollback *> set
timeflowPointParameters.type=TimeflowPointBookmark
delphix database "dexample" rollback *> set timeflowPointParameters.bookmark="b
ookmark example"
```

```
delphix database "dexample" rollback *> commit
```

13.2.4.5.22 CLI cookbook: rolling forward a VDB

This topic describes how to roll forward a virtual database after it has been rewound, as described in Provisioning and Managing Virtual Databases[763].

Once a VDB has rewound to a specific TimeFlow point, the snapshots of its previous states are still available in Delphix Engine storage and are accessed via the command-line interface to restore those previous states. This is referred to as "rolling forward" a VDB.

13.2.4.5.22.1 Procedure

1. Use the `ls` command to find the VDB you want to roll forward. In this example, the TimeFlows and their associated containers are listed. The VDB called `PVDB` will be the one to roll forward.

```
delphix timeflow> ls
ObjectsNAME                                CONTAINER  PARENTPOINT.
TIMEFLOW                PARENTPOINT.LOCATION  PARENTPOINT.TIMESTAMP
hrprod/default                      hrprod     -
-                    -
erpprod/default
                     erpprod    -                                    -
                    -
'DB_PROVISION@2013-11-25T17:37:06'  PVDB       erpprod/default
657925                -'DB_ROLLBACK@2013-11-25T18:24:16'   PVDB
'DB_PROVISION@2013-11-25T17:37:06'  678552
```

2. Use the `Select` command to select the database.

```
delphix database> select PVDB
```

3. Use the `rollback` command to roll forward the VDB.

```
delphix database "PVDB"> rollback
```

4. Use the `ls` command to display options for selecting TimeFlow parameters.

```
delphix database "PVDB" rollback *> ls
Properties
    type: OracleRollbackParameters
```

763 https://delphixdocs.atlassian.net/wiki/spaces/DCT/pages/1114600

```
    credential: (unset)
    timeflowPointParameters:
        type: TimeflowPointSemantic
        container: (required)
        location: LATEST_POINT
    username: (unset)
```

5. Because this VDB was rolled back, two TimeFlows now exist for it. To rollback the VDB and roll it forward, select the original TimeFlow, because the original snapshots are associated with that TimeFlow.

```
delphix database "PVDB" rollback *> set
timeflowPointParameters.type=TimeflowPointLocation
delphix database "PVDB" rollback *> set timeflowPointParameters.timeflow='DB_PR
OVISION@2013-11-25T17:37:06'
```

6. Use the `ls` command to view the parameter options for the TimeFlow you selected.

```
delphix database "PVDB" rollback *> ls
Properties
    type: OracleRollbackParameters
    credential: (unset)
    timeflowPointParameters:
        type: TimeflowPointLocation (*)
        location: LATEST_POINT
        timeflow: 'DB_PROVISION@2013-11-25T17:37:06' (*)
    username: (unset)
```

7. Set the TimeFlow location to rollback the VDB to a particular Oracle SCN.

```
delphix database "PVDB" rollback *> set timeflowPointParameters.location=678994
```

8. Use the `ls` command to review all the options you selected before executing the commit.

```
delphix database "PVDB" rollback *> ls
Properties
    type: OracleRollbackParameters
    credential: (unset)
    timeflowPointParameters:
        type: TimeflowPointLocation (*)
        location: 678994 (*)
        timeflow: 'DB_PROVISION@2013-11-25T17:37:06' (*)
    username: (unset)
```

9. `Commit` the changes.

```
delphix database "PVDB" rollback *> commit
    Dispatched job JOB-369
    DB_ROLLBACK job started for "ERP/PVDB".
    Starting provision of the virtual database "PVDB".
    Creating new TimeFlow.
    Generating recovery scripts.
    Exporting storage.
    Validating user environment settings on target host.
    Mounting filesystems for the virtual database instance "1".
    Mounting read-only archive log filesystem for the virtual database instance
"1".
    Running user-specified pre-provisioning script.
    Recovering Oracle database.
    Opening the virtual database "PVDB".
    Opening Oracle database.
    Oracle recovery was successful.
    Unmounting read-only archive log filesystem for the virtual database
instance "1".
    Running user-specified post-provisioning script.
    The virtual database "PVDB" was successfully provisioned.
    Starting snapshot of virtual database.
    Processing database files of virtual database.
    Creating snapshot of virtual database.
    Finalizing snapshot of virtual database.
    Virtual database "PVDB" snapshot successful.
    DB_ROLLBACK job for "ERP/PVDB" completed successfully.
```

13.2.4.5.23 CLI Cookbook: taking a snapshot

This article is to document how to take a Snapshot outside of the normal snapshot policy time using the CLI, you can also do this in the GUI using the camera icon. A Snapshot of a VDB is similar to bookmarking a point in time in the life of the VDB.

13.2.4.5.23.1 Procedure:

1. ssh into the Delphix Engine using delphix_admin credentials.
2. Go into databases and select the VDB or dSource you would like to take a Snapshot of .

```
ssh admin@dengine
delphix > database
delphix database > select vdb_test
```

3. You are now going to sync and commit the operation .

```
delphix database "vdb_test" > sync
```

```
delphix database "vdb_test" sync *> commit
```

4. You can verify the snapshot by going to snapshots and listing them

```
delphix database "vdb_test" > /snapshot
delphix snapshot > ls
```

13.2.4.5.24 CLI cookbook: V2P (Virtual to Physical) of a single instance non-multitenant Oracle database

This topic describes how to provision a physical single instance non-multitenant Oracle database using the Delphix Engine command-line interface.

13.2.4.5.24.1 Prerequisites

You will need the following information:

- The instance name, instance number, and unique name of the Oracle database you wish to create
- The source dSource or VDB from which you wish to provision.
- The semanticLocation, SCN, or timestamp of the point you want to provision from. You can run these commands to get the list of snapshots or Timeflow ranges:

```
snapshot list database=dexample
timeflow "dexample" timeflowRanges; commit
```

- The layout of the filesystems on the target server where data should be exported.
- The source repository (oracle install) in which to create the VDB. These can be listed with the `/repository list` command.

13.2.4.5.24.2 Procedure

1. Execute the `database export` command.

```
delphix> database export
```

2. Set the Timeflow point type, source container, and location.

```
delphix database export *> set
timeflowPointParameters.type=TimeflowPointSemantic
```

```
delphix database export *> set timeflowPointParameters.container=dexample
delphix database export *> set timeflowPointParameters.location=LATEST_SNAPSHOT
```

3. Edit the sourceConfig configuration, specifying the parameters for the database created via V2P.

```
delphix database export *> edit sourceConfig
delphix database export sourceConfig *> set type=OracleSIConfig
delphix database export sourceConfig *> edit instance
delphix database export sourceConfig instance *> ls
delphix database export sourceConfig instance *> set instanceName=v2p_db
delphix database export sourceConfig instance *> set instanceNumber=1
delphix database export sourceConfig instance *> back
delphix database export sourceConfig *> set repository=tserver/'/u01/app/
ora11204/product/11.2.0/dbhome_1'
 delphix database export sourceConfig *> set uniqueName=v2p_db
delphix database export sourceConfig *> set databaseName=v2p_db
delphix database export sourceConfig *> back
```

4. Set the destination locations.

```
delphix database export *> set filesystemLayout.targetDirectory=//u01/app/
oracle/oradata/v2p_db
delphix database export *> set filesystemLayout.archiveDirectory=archive
delphix database export *> set filesystemLayout.dataDirectory=datafiles
delphix database export *> set filesystemLayout.externalDirectory=external
delphix database export *> set filesystemLayout.scriptDirectory=script
delphix database export *> set filesystemLayout.tempDirectory=temp
```

5. Commit the configuration to execute the job.

```
delphix database export *> commit
```

13.2.4.5.25 CLI cookbook: V2P (Virtual to Physical) of a single instance Oracle database with datafiles on separate file systems

This topic describes how to provision a physical single-instance Oracle database, with datafiles on separate file systems, using the Delphix Engine command-line interface.

By default, all of the customizable directories - `data`, `archive`, `temp`, `external`, and `script` will be placed under the `target` directory, with additional config files placed in the `target` directory itself. This provides no mechanism to place datafiles on separate file systems (i.e. `/data1` and `/data2`, which have the root directory as a common point). In order to allow this, the `useAbsolutePathForDataFiles` parameter must be specified.

13.2.4.5.25.1 Prerequisites

You will need the following information:

- The instance name, instance number, and unique name of the Oracle database to create.
- The source dSource or VDB from which to provision.
- The semanticLocation, SCN, or timestamp of the point of which to provision from. Run these commands to get the list of snapshots or Timeflow ranges:

```
snapshot list database=dexample
timeflow "dexample" timeflowRanges; commit
```

- The layout of the filesystems on the target server where data should be exported.
- The source repository (oracle install) in which to create the VDB. These can be listed with the `/repository list` command.

13.2.4.5.25.2 Procedure

1. Execute the `database export` command.

```
delphix> database export
```

2. Set the Timeflow point type, source container, and location.

```
delphix database export *> set
timeflowPointParameters.type=TimeflowPointSemanticdelphix database export *>
set timeflowPointParameters.container=dexample
delphix database export *> set timeflowPointParameters.location=LATEST_SNAPSHOT
```

3. Edit the sourceConfig configuration, specifying the parameters for the database created via V2P.

```
delphix database export *> edit sourceConfig
delphix database export sourceConfig *> set type=OracleSIConfig
delphix database export sourceConfig *> edit instance
delphix database export sourceConfig instance *> ls
delphix database export sourceConfig instance *> set instanceName=v2p_db
delphix database export sourceConfig instance *> set instanceNumber=1
delphix database export sourceConfig instance *> back
delphix database export sourceConfig *> set repository=tserver/'/u01/app/
ora11204/product/11.2.0/dbhome_1'
delphix database export sourceConfig *> set uniqueName=v2p_db
delphix database export sourceConfig *> set databaseName=v2p_db
delphix database export sourceConfig *> back
```

4. Set the destination locations, including `setAbsolutePathForDataFiles` to be true.

```
delphix database export *> set filesystemLayout.targetDirectory=/v2p_db
delphix database export *> set filesystemLayout.archiveDirectory=archive
delphix database export *> set filesystemLayout.dataDirectory=datafiles
delphix database export *> set filesystemLayout.externalDirectory=external
delphix database export *> set filesystemLayout.scriptDirectory=script
delphix database export *> set filesystemLayout.tempDirectory=tempdelphix
database export *> set filesystemLayout.useAbsolutePathForDataFiles=true
```

5. Specify the file mapping rules for each datafile and the location to which it should be exported. Each entry should be separated by a newline, and the entire mapping within double quotes.

```
set fileMappingRules="/u01/app/oracle/oradata/datafile1.dbf:/data1/
datafile1.dbf
/u01/app/oracle/oradata/datafile2.dbf:/data1/datafile2.dbf
/u01/app/oracle/oradata/datafile3.dbf:/data2/datafile3.dbf
/u01/app/oracle/oradata/datafile4.dbf:/data2/datafile4.dbf"
```

6. Commit the configuration to execute the job.

```
delphix database export *> commit
```

13.2.4.5.26 CLI cookbook: export a non-multitenant virtual Oracle database to ASM

This topic describes how to perform an export or an in-place conversion of a non-multitenant virtual database to a physical database using the Delphix Engine command-line interface. This procedure will work irrespective of the VDB being a single instance configuration or RAC.

13.2.4.5.26.1 Prerequisites

You must have the following configuration before you start the export:

- Target ASM data diskgroup, or the diskgroup that will contain all the database files
- The VDB that needs to be exported to ASM.
- Optionally, the target ASM disk group for redo log files, database unique name for the resulting physical database and the number of RMAN channels.

> ⚠ Ensure that a database instance with the same unique name as the VDB is not running on the target host before starting the below procedure, otherwise the export operation will fail.

13.2.4.5.26.2 Procedure

1. Execute the `database export` command.

```
delphix> database export
```

2. Set the database export parameters type, transfer strategy type, storage strategy type, default target data diskgroup, and virtual source.

```
delphix database export *> set type=OracleDBExportParameters
delphix database export *> set storageStrategy.type=OracleExportASMStorageStrat
egy
delphix database export *> set
 storageStrategy.asmLayout.defaultDataDiskgroup=+DATA
delphix database export *> set transferStrategy.type=OracleExportDBInPlaceTrans
ferStrategy
delphix database export *> set transferStrategy.virtualSource=v2p_db
```

> **Info:** The above values are just examples. Replace the default data diskgroup and virtual source name to match your needs.

3. Optionally set the following parameters:
   a. Target ASM disk group for redo log files.
   b. Database unique name for the resulting physical database.
      i. If no database unique name is specified, the default is the unique name of the VDB.
      ii. If a database unique name is specified, as noted in step 2 of this procedure, you need to ensure that a database instance with the same database unique name as the VDB is not running on the target host.
   c. Number of RMAN channels. Default value for the RMAN channels is 8.
   d. RMAN file section size. Default value is 0 (i.e. RMAN file section size is not set).

```
delphix database export *> set
storageStrategy.asmLayout.redoDiskgroup=+REDO
delphix database export *> set transferStrategy.dbUniqueName=v2asm_db
delphix database export *> set transferStrategy.rmanChannels=10
delphix database export *> set
transferStrategy.rmanFileSectionSizeInGb=64
```

> **Info:** The above values are just examples. Replace the redo diskgroup, database unique name, number of RMAN channels and RMAN file section size to match your needs. For more details on RMAN channels and RMAN file section size, refer to the section titled "*Performance tuning considerations for Oracle databases with bigfile tablespaces*" on the Export an Oracle VDB or a vPDB to a Physical ASM or Exadata Database (see page 1628) page.

4. Commit the configuration to execute the job.

```
delphix database export *> commit
```

The export operation takes a snapshot of the VDB before proceeding with the export. After a successful export operation, the original VDB will be left in a DISABLED state. If the VDB needs to be re-enabled, the VDB needs to be migrated to a different host. Ensure there is no instance running with the same name as the VDB on the target host. Subsequently, you must rewind the VDB to the latest snapshot, after which the VDB will be back up and running. Since the export operation takes a snapshot before the actual export, the rewind can be performed to the point just before the export started.

Refer to the Export an Oracle VDB or a vPDB to a Physical ASM or Exadata Database (see page 1628) page for additional considerations after a successful export.

13.2.4.5.27 CLI cookbook: export a multitenant virtual pluggable Oracle database to ASM

This topic describes how to perform an export or an in-place conversion of a multitenant virtual pluggable database (vPDB) to a multitenant physical Oracle pluggable database (PDB) using the Delphix Engine Command-line Interface (CLI). This procedure will work irrespective of the vPDB being a single instance configuration or RAC.

13.2.4.5.27.1 Prerequisites

You must have the following configuration before you start the export:

- Target ASM data diskgroup, or the diskgroup that will contain all the database files
- The vPDB that needs to be exported to ASM.
- Optionally, the pluggable database name for the resulting physical pluggable database and the number of RMAN channels.

> If the vPDB is being renamed during the export, confirm there is no other PDB with the same name in the target CDB before starting the below procedure. If a PDB with the same name exists in the target CDB, the export operation will fail before starting the actual movement of datafiles.

13.2.4.5.27.2 Procedure

1. Execute the `database export` command.

```
delphix> database export
```

2. Set the database export parameters type, transfer strategy type, storage strategy type, default target data diskgroup, and virtual source.

```
delphix database export *> set type=OraclePDBExportParameters
delphix database export *> set
storageStrategy.type=OracleExportASMStorageStrategy
delphix database export *> set
storageStrategy.asmLayout.defaultDataDiskgroup=+DATA
delphix database export *> set
transferStrategy.type=OracleExportPDBInPlaceTransferStrategy
delphix database export *> set transferStrategy.virtualSource=v2p_pdb
```

> **Info:** The above values are just examples. Replace the default data diskgroup and virtual source name to match your needs.

3. Optionally set the following parameters:
   a. pluggable database name for the resulting physical pluggable database.
      i. If a pluggable database name is specified, the target CDB should not have a PDB with the same name, otherwise the export will fail.
      ii. If no pluggable database name is specified, the default is the current name of the pluggable database.
   b. Number of RMAN channels. Default value for the RMAN channels is 8.
   c. RMAN file section size. Default value is 0 (i.e. RMAN file section size is not set).

```
delphix database export *> set transferStrategy.pdbName=v2asm_pdb
delphix database export *> set transferStrategy.rmanChannels=10
delphix database export *> set
transferStrategy.rmanFileSectionSizeInGb=64
```

> The above values are just examples. Please note that redoDiskgroup parameter is not required for PDB export. Replace the PDB name, number of RMAN channels and RMAN file section size to match your needs. For more details on RMAN channels and RMAN file section size, refer to the

section titled "*Performance tuning considerations for Oracle databases with bigfile tablespaces*" on the Export an Oracle VDB or a vPDB to a Physical ASM or Exadata Database (see page 1628) page.

4. Commit the configuration to execute the job.

```
delphix database export *> commit
```

The export operation takes a snapshot of the vPDB before proceeding with the export. After a successful export operation, the original vPDB will be left in a DISABLED state. If the vPDB needs to be re-enabled, the vPDB needs to be migrated to a different host and/or CDB. Ensure there is no PDB with the same name as the vPDB on the target CDB. Subsequently, you must rewind the vPDB to the latest snapshot, after which the vPDB will be back up and running. Since the export operation takes a snapshot before the actual export, the rewind can be performed to the point just before export started.

Refer to the Export an Oracle VDB or a vPDB to a Physical ASM or Exadata Database (see page 1628) page for additional considerations after a successful export.

13.2.4.5.28 CLI cookbook: export a snapshot or a timeflow point of a non-multitenant Oracle database to ASM

This topic describes how to perform an export of a snapshot or a timeflow point belonging to a non-multitenant Oracle database to a physical database stored on an Oracle Automatic Storage Management (ASM) diskgroup using the Delphix Engine command-line interface. The export procedure provisions a temporary VDB from the snapshot or timeflow point as specified in the CLI parameters and then performs an in-place conversion of the temporary VDB to a physical database in an ASM diskgroup. This temporary VDB is destroyed at the end of the export procedure.

13.2.4.5.28.1 Prerequisites

You must specify the following parameters for the export:

- Target ASM data diskgroup, or the diskgroup that will contain all the database files.
- The base mount point on the target server where temporary VDB data should be mounted.
- Database name for the physical non-multitenant database.
- Unique name for the physical non-multitenant database.
- The target repository for the physical non-multitenant database. These can be listed with the `/repository list` command.
- The instance name for the physical non-multitenant database.
- The instance number for the physical non-multitenant database.
- The snapshot or Timeflow point of an Oracle non-multitenant dSource or a VDB that needs to be exported to ASM. This will be referenced as the "container" in the Timeflow point parameters in step 6 below. You can run these commands to get the list of snapshots or Timeflow ranges:

```
snapshot list database=dexample
timeflow "dexample" timeflowRanges; commit
```

- Optionally, the target ASM disk group for redo log files, number of RMAN channels and the RMAN file section size.

In the example CLI export detailed below, the following configuration parameters are being passed:

- The snapshot that is being exported to ASM belongs to a non-multitenant database "dexample".
- The database name and instance name for the physical non-multitenant database are both "asmdb".
- The database unique name for the physical non-multitenant database is "asmdb_uniq".
- The instance number for the physical non-multitenant database, which is 1 in this example, being a single-instance database.
- The target ASM data diskgroup is "+DATA".
- The base mount point on the target server is "/mnt/provision".
- The target repository is "ora1914-asm-tgt/'/u01/app/oracle/product/19.14.0.0/dbhome_1'"

13.2.4.5.28.2 Procedure

1. Execute the `database export` command.

```
delphix> database export
```

2. Set the database export parameters type, storage strategy type, ASM layout type and the target ASM data diskgroup.

```
delphix database export *> set type=OracleDBExportParameters
delphix database export *> set
storageStrategy.type=OracleExportASMStorageStrategy
delphix database export *> set storageStrategy.asmLayout.type=OracleASMLayout
delphix database export *> set
storageStrategy.asmLayout.defaultDataDiskgroup=+DATA
```

3. Set the transfer strategy type and base mount point.

```
delphix database export *> set
transferStrategy.type=OracleExportDBTimeflowPointTransferStrategy
delphix database export *> set transferStrategy.mountBase=/mnt/provision
```

4. Set the source config type to be a single instance non-multitenant Oracle, and set the database name, database unique name and target repository.

```
delphix database export *> edit transferStrategy.sourceConfig
```

```
delphix database export transferStrategy.sourceConfig *> set
type=OracleSIConfig
delphix database export transferStrategy.sourceConfig *> set databaseName=asmdb
delphix database export transferStrategy.sourceConfig *> set
uniqueName=asmdb_uniq
delphix database export transferStrategy.sourceConfig *> set
repository=ora1914-asm-tgt/'/u01/app/oracle/product/19.14.0.0/dbhome_1'
```

5. Set the instance name and number. In case of a single-instance database, the instance number will be 1. The instance name that is specified will be the SID of the resulting physical database upon successful completion of the export procedure.

```
delphix database export transferStrategy.sourceConfig *> edit instance
delphix database export transferStrategy.sourceConfig instance *> set
instanceName=asmdb
delphix database export transferStrategy.sourceConfig instance *> set
instanceNumber=1
delphix database export transferStrategy.sourceConfig instance *> back
delphix database export transferStrategy.sourceConfig *> back
```

6. Set the Timeflow point parameters. In the below example, the latest timeflow point for the non-multitenant database is being specified for export.

```
delphix database export *> edit transferStrategy.timeflowPointParameters
delphix database export transferStrategy.timeflowPointParameters *> set type=Ti
meflowPointSemantic
delphix database export transferStrategy.timeflowPointParameters *> set
 container=dexample
delphix database export transferStrategy.timeflowPointParameters *> set
 location=LATEST_POINT
delphix database export transferStrategy.timeflowPointParameters *> back
```

Info: The parameter values in steps 2 to 6 above are just examples. Replace the appropriate parameter values to match your needs.

7. Optionally set the following parameters:
   a. Target ASM diskgroup for redo log files.
   b. Number of RMAN channels. Default value for the RMAN channels is 8.
   c. RMAN file section size. Default value is 0 (i.e. RMAN file section size is not set).

```
delphix database export *> set
storageStrategy.asmLayout.redoDiskgroup=+REDO
delphix database export *> set transferStrategy.rmanChannels=10
```

```
delphix database export *> set
transferStrategy.rmanFileSectionSizeInGb=64
```

Info: The above values are just examples. Replace the redo diskgroup, number of RMAN channels and RMAN file section size to match your needs. For more details on RMAN channels and RMAN file section size, refer to the section titled "*Performance tuning considerations for Oracle databases with bigfile tablespaces*" in the Export an Oracle VDB or a vPDB to a Physical ASM or Exadata Database (see page 1628) article.

8. Check that all the settings you require are in place using the `ls` command.

```
delphix database export *> ls
Properties
    type: OracleDBExportParameters (*)
    storageStrategy:
        type: OracleExportASMStorageStrategy (*)
        asmLayout:
            type: OracleASMLayout (*)
            defaultDataDiskgroup: +DATA (*)
            redoDiskgroup: (unset)
    transferStrategy:
        type: OracleExportDBTimeflowPointTransferStrategy (*)
        configParams: (unset)
        mountBase: /mnt/provision (*)
        rmanChannels: 8 (*)
        rmanFileSectionSizeInGb: 0 (*)
        sourceConfig:
            type: OracleSIConfig (*)
            databaseName: asmdb (*)
            environmentUser: (unset)
            instance:
                type: OracleInstance (*)
                instanceName: asmdb (*)
                instanceNumber: 1 (*)
            linkingEnabled: true (*)
            nonSysCredentials: (unset)
            nonSysUser: (unset)
            repository: ora1914-asm-tgt/'/u01/app/oracle/product/19.14.0.0/dbhome_1'
 (*)
            services: (unset)
            tdeKeystorePassword: (unset)
            uniqueName: asmdb_uniq (*)
        timeflowPointParameters:
            type: TimeflowPointSemantic (*)
            container: dexample (*)
            location: LATEST_POINT (*)
delphix database export *>
```

8. Initiate the export by committing the operation in the CLI.

```
delphix database export *> commit
    Dispatched job JOB-65
    DB_EXPORT job started for "dexample".
    Provisioning temporary virtual database "asmdb_uniq" from the chosen snapshot or
point-in-time for exporting.
    Provisioning of temporary virtual database "asmdb_uniq" completed successfully.
    Starting export of database "asmdb" at "Untitled/dexample time: Wed Sep 20
14:00:13 PDT 2023" to location "+DATA" on target environment "ora1914-asm-tgt".
    Generating V2P scripts.
    Setting up environment map.
    Setting up database for V2P.
```

```
    Performing backup as copy for database V2P.
    Performing Switch to copy for database V2P.
    Finalizing database V2P.
    Disabling virtual database "asmdb_uniq".
    Unexporting storage containers.
    Virtual database "asmdb_uniq" disabled.
    The export job for virtual source 'ORACLE_VIRTUAL_SOURCE-9' with name
'asmdb_uniq' got COMPLETED. Started cleanup for the same. Monitor job '<JOB-69>' and
take suggested action if the cleanup fails.
    DB_EXPORT job for "dexample" completed successfully.
```

13.2.4.5.29 CLI cookbook: export a snapshot or a timeflow point of a multitenant pluggable Oracle database to ASM

This topic describes how to perform an export of a snapshot or a timeflow point belonging to a multitenant pluggable Oracle database to a physical pluggable database stored on an Oracle Automatic Storage Management (ASM) diskgroup using the Delphix Engine command-line interface. The export procedure provisions a temporary vPDB from the snapshot or timeflow point as specified in the CLI parameters and then performs an in-place conversion of the temporary vPDB to a physical database on an ASM diskgroup. This temporary vPDB is destroyed at the end of the export procedure.

13.2.4.5.29.1 Prerequisites

You must have the following configuration before you start the export:

- Target ASM data diskgroup, or the diskgroup on the target server that will contain all the database files.
- The base mount point on the target server where the temporary vPDB data should be mounted.
- Database name for the physical pluggable database.
- A linked container database on the target system. This will be where the physical pluggable database will be plugged into, on the target system.
- The snapshot or Timeflow point of an Oracle multitenant dSource or a PDB that needs to be exported to ASM. This will be referenced as the "container" in the Timeflow point parameters in step 5 below. You can run these commands to get the list of snapshots or Timeflow ranges:

  ```
  snapshot list database=dexample
  timeflow "dexample" timeflowRanges; commit
  ```

- Optionally, the target ASM disk group for redo log files, number of RMAN channels and the RMAN file section size.

In the example CLI export detailed below, the following configuration parameters are being passed:

- The snapshot that is being exported to ASM belongs to PDB "CDOMLOTGAS2FPDB1“.
- The database name for the physical pluggable database is "asmpdb".
- The container database where the physical pluggable database will be exported into is "CDOMLOTGAS2F”.
- The target ASM data diskgroup is "+DATA".

- The base mount point on the target server is "/mnt/provision".

13.2.4.5.29.2 Procedure

1. Execute the `database export` command.

```
delphix> database export
```

2. Set the database export parameters type, storage strategy type, ASM layout type and the target ASM data diskgroup.

```
delphix database export *> set type=OraclePDBExportParameters
delphix database export *> set
storageStrategy.type=OracleExportASMStorageStrategy
delphix database export *> set storageStrategy.asmLayout.type=OracleASMLayout
delphix database export *> set
storageStrategy.asmLayout.defaultDataDiskgroup=+DATA
```

3. Set the transfer strategy type and base mount point.

```
delphix database export *> set
transferStrategy.type=OracleExportPDBTimeflowPointTransferStrategy
delphix database export *> set transferStrategy.mountBase=/mnt/provision
```

4. Set the source config type as PDB config, and set the database name, database unique name and the linked container database where the resulting physical database will be exported into.

```
delphix database export *> edit transferStrategy.sourceConfig
delphix database export transferStrategy.sourceConfig *> set
type=OraclePDBConfig
delphix database export transferStrategy.sourceConfig *> set
databaseName=asmpdb
delphix database export transferStrategy.sourceConfig *> set
cdbConfig=CDOMLOTGAS2F
delphix database export transferStrategy.sourceConfig *> back
```

5. Set the Timeflow point parameters. In the below example, the latest timeflow point for the pluggable database is being specified for export.

```
delphix database export *> edit transferStrategy.timeflowPointParameters
delphix database export transferStrategy.timeflowPointParameters *> set
type=TimeflowPointSemantic
delphix database export transferStrategy.timeflowPointParameters *> set
container=CDOMLOTGAS2FPDB1
```

```
delphix database export transferStrategy.timeflowPointParameters *> set
location=LATEST_POINT
delphix database export transferStrategy.timeflowPointParameters *> back
```

Info: The parameter values in steps 2 to 5 above are just examples. Replace the appropriate parameter values to match your needs.

6. Optionally set the following parameters:
   a. Target ASM disk group for redo log files.
   b. Number of RMAN channels. Default value for the RMAN channels is 8.
   c. RMAN file section size. Default value is 0 (i.e. RMAN file section size is not set).

```
delphix database export *> set
storageStrategy.asmLayout.redoDiskgroup=+REDO
delphix database export *> set transferStrategy.rmanChannels=10
delphix database export *> set
transferStrategy.rmanFileSectionSizeInGb=64
```

Info: The above values are just examples. Replace the redo diskgroup, number of RMAN channels and RMAN file section size to match your needs. For more details on RMAN channels and RMAN file section size, refer to the section titled "*Performance tuning considerations for Oracle databases with bigfile tablespaces*" on the Export an Oracle VDB or a vPDB to a Physical ASM or Exadata Database (see page 1628) page.

7. Check that all the settings you require are in place using the `ls` command.

```
delphix database export *> ls
Properties
    type: OraclePDBExportParameters (*)
    storageStrategy:
        type: OracleExportASMStorageStrategy (*)
        asmLayout:
            type: OracleASMLayout (*)
            defaultDataDiskgroup: +DATA (*)
            redoDiskgroup: (unset)
    transferStrategy:
        type: OracleExportPDBTimeflowPointTransferStrategy (*)
        configParams: (unset)
        mountBase: /mnt/provision (*)
        parentTdeKeystorePassword: (unset)
        parentTdeKeystorePath: (unset)
```

```
        rmanChannels: 8 (*)
        rmanFileSectionSizeInGb: 0 (*)
        sourceConfig:
            type: OraclePDBConfig (*)
            cdbConfig: CDOMLOTGAS2F (*)
            databaseName: asmpdb (*)
            environmentUser: (unset)
            linkingEnabled: true (*)
            nonSysCredentials: (unset)
            nonSysUser: (unset)
            repository: (unset)
            services: (unset)
        tdeExportedKeyFileSecret: (unset)
        tdeKeyIdentifier: (unset)
        timeflowPointParameters:
            type: TimeflowPointSemantic (*)
            container: CDOMLOTGAS2FPDB1 (*)
            location: LATEST_POINT (*)
delphix database export *>
```

8. Initiate the export by committing the operation in the CLI.

```
delphix database export *> commit
    Dispatched job JOB-72
    DB_EXPORT job started for "CDOMLOTGAS2FPDB1".
    Provisioning temporary virtual database "asmpdb" from the chosen snapshot or
point-in-time for exporting.
    Provisioning of temporary virtual database "asmpdb" completed successfully.
    Starting export of database "asmpdb" at "Untitled/CDOMLOTGAS2FPDB1 time: Wed Sep
20 14:00:40 PDT 2023" to location "+DATA" on target environment "ora1914-asm-tgt".
    Generating V2P scripts.
    Setting up environment map.
    Setting up database for V2P.
    Performing backup as copy for database V2P.
    Performing Switch to copy for database V2P.
    Finalizing database V2P.
    Disabling virtual database "asmpdb".
    Unexporting storage containers.
    Virtual database "asmpdb" disabled.
    The export job for virtual source 'ORACLE_VIRTUAL_PDB_SOURCE-3' with name
'asmpdb' got COMPLETED. Started cleanup for the same. Monitor job '<JOB-82>' and take
suggested action if the cleanup fails.
    DB_EXPORT job for "CDOMLOTGAS2FPDB1" completed successfully.
```

13.2.4.5.30 CLI Cookbook: V2P virtual to physical on SQL server

This topic describes how to export a physical single instance SQL Server database from a VDB using the Delphix Engine command-line interface.

13.2.4.5.30.1 Prerequisites

You will need the following information:

- The VDB name which you wish to export.
- The database name of the SQL Server database you wish to create
- The layout of the filesystems on the target server where data should be exported.
- The target repository in which to create the physical database. Repositories can be listed with the `/repository list` command.

13.2.4.5.30.2 Procedure

1. Execute the command `database export`.

```
delphix> database export
```

2. Specify the set type=MSSqlExportParameters

```
delphix database export *> set type=MSSqlExportParameters
```

3. Set the VDB you wish to export

```
delphix database export *> edit timeflowPointParameters
delphix database export timeflowPointParameters *> set container=vdb
delphix database export timeflowPointParameters *> back
```

4. Edit the sourceConfig configuration, specifying the parameters for the database created via V2P.

```
delphix database export *> edit sourceConfig
delphix database export sourceConfig *> set type=MSSqlSIConfig
delphix database export sourceConfig *> set databaseName=v2p_db
delphix database export sourceConfig *> set repository=win2012/SQL2016
delphix database export sourceConfig *> back
```

5. Set the destination locations.

```
delphix database export *> edit filesystemLayout
delphix database export filesystemLayout *> set targetDirectory=C:\temp
delphix database export filesystemLayout *> set archiveDirectory=archive
delphix database export filesystemLayout *> set dataDirectory=datafiles
delphix database export filesystemLayout *> set externalDirectory=external
delphix database export filesystemLayout *> set scriptDirectory=script
```

```
delphix database export filesystemLayout *> set tempDirectory=temp
```

6. Commit the configuration to execute the job.

```
delphix database export *> commit
```

13.2.4.5.31 CLI Cookbook: VDB status

It is possible to get a virtual database (VDB) status from the CLI.

1. Log into the CLI as any user that has privileges on the VDB.

```
ssh admin@yourengine
```

2. From source go to the VDB you want to get a status on.

```
delphix > sourcedelphix source > lsdelphix source > select <yourvdb>
```

3. Run the get runtime command to see all information or just get runtime.status for if the VDB is running.

```
delphix source 'vdb' > get runtime
```

or

```
delphix source 'vdb' > get runtime.status
```

4. If you would like to see more than one VDBs status you can also do the following:

```
delphix > sourcedelphix source > list display=name,runtime.status
```

13.2.4.5.32 CLI cookbook: provisioning a virtual PDB from a non-multitenant source database

Delphix supports provisioning a vPDB from a non-multitenant source database. This feature is only available through the API or command-line interface.

This topic describes how to provision a virtual pluggable database (vPDB) from a non-multitenant source database using the command-line interface.

This feature has the following restrictions:

Transparent Data Encryption (TDE) is not supported.
The provision point must correspond to a snapshot. Provisioning from a point in time between snapshots is not supported.
The target CDB must be a physical CDB. Virtual CDB targets are not supported.

13.2.4.5.32.1 Prerequisites

Provisioning a vPDB from a non-multitenant source has the following environment requirements:

- Source host with a non-multitenant Oracle 11g or newer source database.
- VDB target host for provisioning a virtual non-multitenant VDB from the source database.
- CDB target host with a running Oracle target version CDB. The target CDB will be automatically linked if it is not already linked.

The target CDB can be a newer Oracle version than the source database (for example, the source is 12.2 and target is 19c). When an upgrade is also required, there are two options for upgrading the database:

- Upgrade Option 1: After provisioning the VDB. This option requires the ability to upgrade to the Oracle target version on the VDB target host.
- Upgrade Option 2: After plugging into the Linked CDB target database.

There are three scripts used during this procedure:

1. Pre-snapshot Hook on VDB: This hook will open the database in "read only" mode and issue a call to the `dbms_pdb.describe` procedure to generate an XML file describing the VDB.
2. **Post-snapshot Hook on VDB**: This hook will return the VDB to "read write" mode.
3. **Non-CDB to PDB Script**: This script will run as a hook present on the Linked CDB target host. This should call into the Oracle script `$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql` to convert the VDB into a PDB. This script should also upgrade the vPDB if doing Upgrade Option 2.

13.2.4.5.32.2 Workflow

1. Link the non-multitenant Oracle 11g or newer source database as a dSource within Delphix.
2. Provision a non-multitenant Oracle VDB from the dSource onto the VDB target host. This will be referred to as the **Golden VDB**.
3. (If using Upgrade Option 1) Upgrade the **Golden VDB** to the Oracle target version: manually upgrade the database and point it to the new Oracle home. This step is only necessary if the source and target Oracle versions are not the same and the data files will not be upgraded when they are converted below.
4. Create a Pre-snapshot hook[764] on the **Golden VDB** to open the database in read only mode and issue the `dbms_pdb.describe` procedure call to create an XML file called `delphix_plugin.xml.` T he XML file will be used to plug the **Golden VDB** data files into the target CDB. The **Golden VDB** must

764 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/19378387

be open read only during the subsequent snapshot so that the VDB data files do not require recovery when plugging them into the target CDB.

5. (Optional) Create a Post-snapshot hook[765] on the **Golden VDB** to remove the database from read only mode. The **Golden VDB** can also remain read only.
6. Take a snapshot of the **Golden VDB**.
7. Create a PDB conversion script[766] named `dx-post-plug-hook.sh` in the root of the Delphix toolkit directory of the Linked CDB target host. The name of the vPDB being provisioned/converted will be supplied by Delphix as the first parameter to the script when it invokes the script. The VDB data files will already be plugged into the Linked CDB target at the time the script is invoked. The script should do the following:
   a. (If using Upgrade Option 2) Upgrade[767] the vPDB data files prior to the conversion.
   b. Call into `$ORACLE_HOME/rdbms/admin/noncdb_to_pdb.sql` and perform any customizations for the multitenant conversion.
8. Select a **snapshot** (point-in-time not supported) on the **Golden VDB** that has the `delphix_plugin.xml` file and provision a virtual PDB to the Linked CDB target. Virtual CDB targets are not supported.Note: This step can be executed via the **API / CLI only**, and will not be allowed via the Delphix UI.

13.2.4.5.32.3 CLI procedure to provision a vPDB from a VDB

1. Log into the Delphix command-line interface using the admin user or a user with admin privileges.

```
$ ssh admin@YOUR_ENGINE
```

2. Move to the database provisioning command line object.

```
delphix> database provision
```

3. Set the parameter type to `OracleMultitenantProvisionParameters`.

```
set type=OracleMultitenantProvisionParameters
```

4. Set the login details for the provision and Delphix OS user who is to perform the provision.

```
delphix database provision *> set username=delphix
delphix database provision *> set credential.type=PasswordCredential
```

765 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/19378387
766 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/19376259
767 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/19443813

```
delphix database provision *> set credential.password=delphix
```

5. Give the dataset a name.

```
delphix database provision *> set container.name=vpdb
```

6. Place the new dataset in a Group that appears in the Delphix GUI, in this case, the Targets group.

```
delphix database provision *> set container.group=Targets
```

7. Set the destination mount point which Delphix NFS mounts are to be linked to under the virtual PDB. This folder must exist at a file system level on the Linked CDB target host. Do not use single quotes around the mount path.

```
delphix database provision *> set source.mountBase="/mnt/provision"
```

8. If automatically restarting the vPDB is not required after a reboot of the Linked CDB target host, set this to option to false. False is possibly a better option given the container database would need to be running prior to any attempt to pull up a vPDB.

```
delphix database provision *> set source.allowAutoVDBRestartOnHostReboot=false
```

9. Supply the destination container database name. The container database should already be discovered. This will be where the vPDB will ultimately be placed.

```
delphix database provision *> set sourceConfig.cdbConfig=CDBSTAGE
```

10. Name the vPDB. This is what it will appear as in the destination container database.

```
delphix database provision *> set sourceConfig.databaseName=vpdb
```

11. Supply the source **Golden VDB** details. In this example, the provision will use the latest snapshot available from the **Golden VDB** as the point in time from which to provision the vPDB. A specific snapshot can also be picked, but an arbitrary point in time is not supported.

```
delphix database provision *> set
timeflowPointParameters.type=TimeflowPointSemantic
delphix database provision *> set timeflowPointParameters.container=gold_vdb
delphix database provision *> set
timeflowPointParameters.location=LATEST_SNAPSHOT
```

12. Check that all the settings you require are in place using the "ls" command.

```
delphix database provision *> ls
Properties
  type: OracleMultitenantProvisionParameters
  container:
    type: OracleDatabaseContainer
    name: vpdb (*)
    description: (unset)
    diagnoseNoLoggingFaults: true
    group: Targets (*)
    performanceMode: DISABLED
    preProvisioningEnabled: false
    sourcingPolicy: (unset)
  credential:
    type: Password Credential (*)
    password: ******** (*)
  masked: (unset)
  maskingJob: (unset)
  source:
    type: OracleVirtualPdbSource (*)
    name: (unset)
    allowAutoVDBRestartOnHostReboot: false (*)
    config: (unset)
    customEnvVars: (unset)
    fileMappingRules: (unset)
    LogCollectionEnabled: false
    mountBase: /mnt/provision (*)
    operations: (unset)
    parentTdeKeystorePassword: (unset)
    parentTdeKeystorePath: (unset)
    tdeExportedKeyFileSecret: (unset)
  sourceConfig:
    type: OraclePDBConfig
    cdbConfig: CDBSTAGE (*)
    databaseName: vpdb (*)
    environmentUser: (unset)
    linkingEnabled: true
    nonSysCredentials: (unset)
    nonSysUser: (unset)
    repository: (unset)
    services: (unset)
  timeflowPointParameters:
    type: TimeflowPointSemantic
    container: gold_vdb (*)
    location: LATEST_SNAPSHOT (*)
  username: delphix (*)
  VirtualCdb: (unset)  Operationsdefaults
```

13. Initiate the provision by committing the operation in the CLI.

```
delphix database provision *> commit
```

```
  vpdb
  Dispatched job JOB-333
  DB_PROVISION job started for "Targets/vpdb".
  Starting provision of virtual PDB database "vpdb" converted from a single
tenant database.
  Preparing multitenant container database "CDBSTAGE".
  Creating new TimeFlow.
  Generating recovery scripts.
  Exporting storage.
  Preparing XML manifest file prior to plugin.
  Plugging in Oracle pluggable database.
  Running user-defined post plug hook.
  Opening Oracle pluggable database.
  Setting OMF destination for Oracle pluggable database.
  Creating PDB tempfiles.
  Checking Oracle pluggable database plugin violations.
  DB_PROVISION job for "Targets/vpdb" completed successfully.
```

To refresh the data in the vPDB from production, first, refresh the Golden VDB from the dSource, then refresh the vPDB from the new snapshot in the Golden VDB.

There are some workflow customizations required for RAC databases:

1. The PDB conversion script must be in the root of the Delphix toolkit directory for all the target CDB RAC instances.
2. The **Golden VDB** Pre-Snapshot hook, as provided below, will not work in a clustered (RAC) environment with more than one active instance because it only shuts down the local instance. `dbms_pdb.describe` will not execute while an instance is open read-write. The workarounds are:
    a. Provision the **Golden VDB** as single-instance, either by provisioning to a non-RAC target or by provisioning to a RAC target with only one active instance. The sample hook will work in this case.
    b. Write a customized pre-snapshot hook that shuts down all instances, restarts only one instance in read-only mode, and runs `dbms_pdb.describe`.
    c. Manually perform the actions of the hook: shutdown the **Golden VDB**, restart one of the instances in read-only mode and then run `dbms_pdb.describe`.

13.2.4.5.32.4 Sample scripts

Golden VDB pre-snapshot hook

Restarts the source VDB in read only mode and runs `dbms_pdb.describe` to generate an XML file describing the VDB. The XML file will be used to plug the VDB into the Linked CDB target. The target for the XML file must be `$DELPHIX_MOUNT_PATH/$DELPHIX_DATABASE_UNIQUE_NAME/datafile/delphix_plugin.xml.`

```
#!/bin/sh

sqlplus "/ AS SYSDBA" <<-EOF
  whenever sqlerror exit 2;
  spool $DELPHIX_MOUNT_PATH/$DELPHIX_DATABASE_UNIQUE_NAME/datafile/presnapshot.
out replace
  shutdown immediate
  startup mount
  alter database open read only;
  exec dbms_pdb.describe(pdb_descr_file=>'$DELPHIX_MOUNT_PATH/
$DELPHIX_DATABASE_UNIQUE_NAME/datafile/delphix_plugin.xml');
  exit;
EOF
```

Golden VDB post-snapshot hook

This is only necessary if the VDB should not be left in read-only mode after the snapshot.

```
#!/bin/sh

sqlplus "/ AS SYSDBA" <<-EOF
  whenever sqlerror exit 2;
  spool $DELPHIX_MOUNT_PATH/$DELPHIX_DATABASE_UNIQUE_NAME/datafile/postsnapshot.out
replace
  shutdown immediate
  startup
  exit;
EOF
```

PDB conversion script

The script should be named `dx-post-plug-hook.sh` and reside in the root of the Delphix toolkit directory of the Linked CDB target host. Delphix will supply the name of the PDB being provisioned/converted as the first parameter.

The VDB datafiles will have already been plugged into the target CDB at the time the script is invoked and the virtual PDB will be in the mounted (not open) state. The PDB conversion script should return with the virtual PDB in either the mounted or open (not restricted) state. Delphix does not enforce a time-out for the script.

```
#!/bin/sh
DELPHIX_PDB_NAME=$1SCRIPT_DIR="$( cd "$( dirname "$0" )" && pwd )"
CONVERT_LOGFILE=$SCRIPT_DIR/$DELPHIX_PDB_NAME-pdbconvert.log

sqlplus "/ AS SYSDBA" <<-EOF
  whenever sqlerror exit 2;
  spool $CONVERT_LOGFILE replace
  alter session set container=$DELPHIX_PDB_NAME;
  @?/rdbms/admin/noncdb_to_pdb.sql
  exit;
EOF
```

PDB upgrade script

The following script will upgrade the vPDB. Use a wrapper that runs both this script and the prior conversion script (or combine the two in a single script) if doing both an upgrade and a conversion.

```
#!/bin/sh

DELPHIX_PDB_NAME=$1
SCRIPT_DIR="$( cd "$( dirname "$0" )" && pwd )"
UPGRADE_LOGFILE=$SCRIPT_DIR/$DELPHIX_PDB_NAME-dx-post-plug-upgrade.log
UPGRADE_LOGDIR=$SCRIPT_DIR/$DELPHIX_PDB_NAME-upgrade

mkdir $UPGRADE_LOGDIR
cd $ORACLE_HOME/rdbms/admin
switches="-c '$DELPHIX_PDB_NAME' -l $UPGRADE_LOGDIR"
$ORACLE_HOME/perl/bin/perl catctl.pl $switches catupgrd.sql &>> $UPGRADE_LOGFILE
```

13.2.4.5.33 CLI cookbook: migrating a virtual PDB in a virtual CDB

Delphix supports migrating from a vPDB in a virtual CDB, to another linked or virtual CDB.

This topic describes how to migrate a virtual pluggable database (vPDB) in a virtual CDB to another virtual CDB using the command-line interface.

13.2.4.5.33.1 Pre-requisites

You should already set up and have Delphix discover a container database in the same environment as the vPDB currently is or from an environment to which the vPDB will be migrated to.

13.2.4.5.33.2 Procedure

1. Log into the Delphix command-line interface using the admin user or a user with admin privileges.

```
ssh admin@YOUR_ENGINE
```

2. Disable the vPDB.

```
delphix> /sourced
elphix source> select myPDB
delphix source 'myPDB'> disable
delphix source 'myPDB' disable *> commit
    Dispatched job JOB-35
    SOURCE_DISABLE job started for "myPDB".
    Disabling virtual database "myPDB".
    Unexporting storage containers.    Virtual database "myPDB" disabled.
    SOURCE_DISABLE job for "myPDB" completed successfully.
delphix source 'myPDB'>
```

3. Change the environment user and repository (ORACLE_HOME) for the vPDB to the appropriate user from the new host.

```
delphix> /sourceconfig
delphix sourceconfig> select myPDB
delphix sourceconfig 'myPDB'> update
delphix sourceconfig 'myPDB' update *> set repository='/u01/app/oracle/product/
19.0.0.0/dbhome_1'
delphix sourceconfig 'myPDB' update *> set environmentUser='/oracle'
delphix sourceconfig 'myPDB' update *> commit
    Dispatched job JOB-38
    SOURCE_CONFIG_UPDATE job started for "myPDB".
    SOURCE_CONFIG_UPDATE job for "myPDB" completed successfully.
```

4. Change the environment user and repository (ORACLE_HOME) for the vCDB to the appropriate user from the new host. This must match the settings in step 3.

```
delphix sourceconfig 'myPDB'> /sourceconfig/
delphix sourceconfig> select myCDB
delphix sourceconfig 'myCDB'> update
delphix sourceconfig 'myCDB' update *> set repository='/u01/app/oracle/product/
19.0.0.0/dbhome_1'
delphix sourceconfig 'myCDB' update *> set environmentUser='/oracle'
delphix sourceconfig 'myCDB' update *> commit
    Dispatched job JOB-39
    SOURCE_CONFIG_UPDATE job started for "myCDB".
```

```
    SOURCE_CONFIG_UPDATE job for "myCDB" completed successfully.
```

5. Enable the vPDB.

```
delphix sourceconfig 'myCDB'> /source
delphix source> select myPDB
delphix source 'myPDB'> enable
delphix source 'myPDB' enable *> commit
    Dispatched job JOB-40
    SOURCE_ENABLE job started for "myPDB".
    Enabling dataset "myPDB".
    Exporting storage containers from the Delphix Engine.
    Mounting datasets.
    Mounting filesystems for the virtual database instance "1".
    Starting virtual database.
    Starting instance 1 on virtual database "myCDB".
    Virtual database "myCDB" was successfully started.
    Starting virtual database.
    Starting instance 1 on virtual database "myPDB".
    Plugging in Oracle pluggable database.
    Opening Oracle pluggable database.
    Setting OMF destination for Oracle pluggable database.
    Creating PDB tempfiles.
    Checking Oracle pluggable database plugin violations.
    Virtual database "myPDB" was successfully started.
    Dataset "myPDB" enabled.
    SOURCE_ENABLE job for "myPDB" completed successfully.
```

13.2.4.5.34 CLI cookbook: Migrating an Oracle RAC virtual PDB and a virtual CDB

Delphix supports migrating from a RAC vPDB in a virtual CDB, to another RAC linked CDB or virtual CDB. This feature is only available through the command-line interface.

> Attempts to perform this migration via GUI will fail with the following error:
> **Error**
> The operation could not be completed because the virtual database “vPDB1" is disabled.
> **Error Code**
> exception.db.genericvdb.disabled

This topic describes how to migrate a RAC virtual pluggable database (vPDB) in a virtual CDB to another RAC virtual CDB using the command-line interface.

13.2.4.5.34.1 Pre-requisites

You should already be set up and have Delphix discover a container database in the same environment as the vPDB currently is or from an environment to which the vPDB will be migrated to.

13.2.4.5.34.2 Procedure

1. Log into the Delphix command-line interface using the admin user or a user with admin privileges.

```
ssh admin@YOUR_ENGINE
```

2. Disable the vPDB.

```
delphix> /source
delphix source> select racPDB
delphix source 'racPDB'> disable
delphix source 'racPDB' disable *> commit
    Dispatched job JOB-74
    SOURCE_DISABLE job started for "racPDB".
    Disabling virtual database "racPDB".
    Unexporting storage containers.
    Virtual database "racPDB" disabled.
    SOURCE_DISABLE job for "racPDB" completed successfully.
```

3. Change the environment user and repository (ORACLE_HOME) for the vPDB to the appropriate user from the new host.

```
delphix source 'racPDB'> /sourceconfig/
delphix sourceconfig> select racPDB
delphix sourceconfig 'racPDB'> update
delphix sourceconfig 'racPDB' update *> set repository='/u01/app/oracle/
product/12.1.0.2/dbhome_1'
delphix sourceconfig 'racPDB' update *> set environmentUser=' /oracle'
delphix sourceconfig 'racPDB' update *> commit
    Dispatched job JOB-76
    SOURCE_CONFIG_UPDATE job started for "racPDB".
    SOURCE_CONFIG_UPDATE job for "racPDB" completed successfully.
```

4. Change the environment user and repository (ORACLE_HOME) for the vCDB to the appropriate user from the new host. This must match the settings in step 3. The instance configuration must also be configured for the new hosts.

```
delphix sourceconfig 'racPDB'> /sourceconfig/
delphix sourceconfig> select racCDB
```

```
delphix sourceconfig 'racCDB'> update
delphix sourceconfig 'racCDB' update *> set repository='/u01/app/oracle/
product/12.1.0.2/dbhome_1'
delphix sourceconfig 'racCDB' update *> set environmentUser=' /oracle'
delphix sourceconfig 'racCDB' update *> edit instances
delphix sourceconfig 'racCDB' update instances *> ls
Properties
    0:
        type: OracleRACInstance
        instanceName: racCDB1
        instanceNumber: 1
        node: mwtestx1    1:
        type: OracleRACInstance
        instanceName: racCDB2
        instanceNumber: 2
        node: mwtestx2Use the "add" command to add an element to this array.
delphix sourceconfig 'racCDB' update instances *> set 0.node=mwtestz1
delphix sourceconfig 'racCDB' update instances *> set 1.node=mwtestz2
delphix sourceconfig 'racCDB' update instances *> commit
    Dispatched job JOB-77
    SOURCE_CONFIG_UPDATE job started for "racCDB".
    SOURCE_CONFIG_UPDATE job for "racCDB" completed successfully.
```

5. Enable the vPDB.

```
delphix sourceconfig 'racCDB'> /sourced
elphix source> select racPDB
delphix source 'racPDB'> enable
delphix source 'racPDB' enable *> commit
    Dispatched job JOB-78
    SOURCE_ENABLE job started for "racPDB".
    Enabling dataset "racPDB".
    Exporting storage containers from the Delphix Engine.
    Mounting datasets.
    Mounting filesystems for the virtual database instance "2".
    Starting virtual database.
    Starting instance 2 on virtual database "racCDB".
    Virtual database "racCDB" was successfully started.
    Starting virtual database.
    Starting instance 2 on virtual database "racPDB".
    Plugging in Oracle pluggable database.
    Opening Oracle pluggable database.
    Setting OMF destination for Oracle pluggable database.
    Creating PDB tempfiles.
    Checking Oracle pluggable database plugin violations.
    Virtual database "racPDB" was successfully started.
    Mounting datasets.
    Mounting filesystems for the virtual database instance "1".
    Starting virtual database.
    Starting instance 1 on virtual database "racCDB".
```

```
    Virtual database "racCDB" was successfully started.
    Starting virtual database.
    Starting instance 1 on virtual database "racPDB".
    Opening Oracle database.
    Checking Oracle pluggable database plugin violations.
    Virtual database "racPDB" was successfully started.
    Dataset "racPDB" enabled.
    SOURCE_ENABLE job for "racPDB" completed successfully.
```

13.2.4.5.35 CLI cookbook: Locating and updating the value of tdeKeyIdentifier

This topic describes how to manage the `tdeKeyIdentifier` field that is associated with the vPDB or the vCDB source object using the command-line interface.

> This process is currently supported only via the CLI.

The following example lists the procedure for a vPDB source object. For a vCDB source object, similar steps need to be followed.

13.2.4.5.35.1 Procedure

1. Log into the Delphix command-line interface using the admin user or a user with admin privileges.

```
$ ssh admin@YOUR_ENGINE
```

2. Move to the database.

```
delphix> source
delphix source> "VCDO_1JL"
```

3. View all the settings using the "ls" command.

```
delphix source "VCDO_1JL" *> ls
Properties
    type: OracleVirtualPdbSource
    name: VCDO_1JL
    allowAutoVDBRestartOnHostReboot: false
    allowRefreshRewindPostV2P: false
    archivelogMode: true
    config: VCDO_1JL
    configParams:
        _bct_public_dba_buffer_size: 1826784
```

```
        _cdb_disable_pdb_limit: TRUE
        audit_file_dest: '/u01/app/oracle/admin/CDOMLOSR197/adump'
        audit_trail: 'DB'
        compatible: '19.0.0'
        core_dump_dest: '/u01/app/oracle/diag/rdbms/cdomlosr197/CDOMLOSR197/
cdump'
        diagnostic_dest: '/u01/app/oracle'
        dispatchers: '(PROTOCOL=TCP) (SERVICE=CDOMLOSRCA1DXDB)'
        enable_pluggable_database: TRUE
        log_archive_format: '%t_%s_%r.dbf'
        max_pdbs: 4098
        memory_max_target: 1342177280
        memory_target: 1342177280
        nls_language: 'AMERICAN'
        nls_territory: 'AMERICA'
        open_cursors: 300
        processes: 300
        remote_login_passwordfile: 'EXCLUSIVE'
    configTemplate: (unset)
    container: VCDO_1JL
    customEnvVars: (empty)
    linked: false
    logCollectionEnabled: false
    mountBase: /mnt/provision
    newDBID: false
    nodeListeners: (empty)
    operations:
        type: VirtualSourceOperations
        configureClone: (empty)
        postRefresh: (empty)
        postRollback: (empty)
        postSnapshot: (empty)
        postStart: (empty)
        postStop: (empty)
        preRefresh: (empty)
        preRollback: (empty)
        preSnapshot: (empty)
        preStart: (empty)
        preStop: (empty)
    parentTdeKeystorePassword: ********
    parentTdeKeystorePath: /u01/app/oracle/keystores/CDOMLOSR197/wallet
    redoLogGroups: 3
    redoLogSizeInMB: 50
    reference: ORACLE_VIRTUAL_PDB_SOURCE-2
    runtime:
        type: OraclePDBSourceRuntime
        accessible: true
        accessibleTimestamp: 2021-10-06T22:02:15.718Z
        activeInstances:
            0:
                type: OracleActiveInstance
                hostName: ip-10-110-234-67.delphix.com
```

```
                instanceName: CDOMLOSR197
                instanceNumber: 1
        databaseMode: READ_WRITE
        databaseRole: PRIMARY
        databaseSize: 913.4MB
        databaseStats: [ ... ]
        enabled: ENABLED
        lastNonLoggedLocation: 0
        status: RUNNING
    runtimeMountInformation:
        type: UnixRuntimeMountInformation
        name: (unset)
        nfsVersion: 4
        nfsVersionReason: DEFAULT
    staging: false
    status: DEFAULT
    tdeExportedKeyFileSecret: ********
    tdeKeyIdentifier: AbSP7gninU+Gv1YQ/iEcJAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    tdeUUID: a3f26971-1df6-4c81-994f-4b2c582ded87
    virtual: true
Operations
update
enable
disable
start
stop
upgrade
```

4. Note that `tdeKeyIdentifier` is one of the last fields listed above. If we query the vPDB via sqlplus on the target host, we can see the matching `key_id`.
   Note that any key generated by Delphix will include a tag with the format `dlpx_key_<tdeUUID>`.

```
SQL> alter session set container=VCDO_1JL;
Session altered.
SQL> select key_id, tag from v$encryption_keys;
KEY_ID
--------------------------------------------------------------------------------
TAG
--------------------------------------------------------------------------------
-
AbSP7gninU+Gv1YQ/iEcJAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
dlpx_key_a3f26971-1df6-4c81-994f-4b2c582ded87
```

5. To generate a new unique encryption key, unset the value of `tdeKeyIdentifier` before a refresh or rewind operation.

```
delphix source 'VCDO_1JL'> update
delphix source 'VCDO_1JL' update *> unset tdeKeyIdentifier
```

```
delphix source 'VCDO_1JL' update *> ls
Properties
    type: OracleVirtualPdbSource
    name: VCDO_1JL
    allowAutoVDBRestartOnHostReboot: false
    allowRefreshRewindPostV2P: false
    customEnvVars: (empty)
    logCollectionEnabled: false
    newDBID: false
    operations:
        type: VirtualSourceOperations
        configureClone: (empty)
        postRefresh: (empty)
        postRollback: (empty)
        postSnapshot: (empty)
        postStart: (empty)
        postStop: (empty)
        preRefresh: (empty)
        preRollback: (empty)
        preSnapshot: (empty)
        preStart: (empty)
        preStop: (empty)
    parentTdeKeystorePassword: ********
    parentTdeKeystorePath: /u01/app/oracle/keystores/CDOMLOSR197/wallet
    tdeKeyIdentifier: (unset) (*)
delphix source 'VCDO_1JL' update *> commit
    Dispatched job JOB-18
    SOURCE_UPDATE job started for "VCDO_1JL".
    SOURCE_UPDATE job for "VCDO_1JL" completed successfully.
```

6. After the refresh or rewind, the new key identifier is now associated with vPDB that can be used for all future Delphix operations. View all the settings using the "ls" command.

```
delphix source 'VCDO_1JL'> ls
Properties
    type: OracleVirtualPdbSource
    name: VCDO_1JL
    allowAutoVDBRestartOnHostReboot: false
    allowRefreshRewindPostV2P: false
    archivelogMode: true
    config: VCDO_1JL
    configParams:
        _bct_public_dba_buffer_size: 1826784
        _cdb_disable_pdb_limit: TRUE
        audit_file_dest: '/u01/app/oracle/admin/CDOMLOSR197/adump'
        audit_trail: 'DB'
        compatible: '19.0.0'
        core_dump_dest: '/u01/app/oracle/diag/rdbms/cdomlosr197/CDOMLOSR197/
cdump'
        diagnostic_dest: '/u01/app/oracle'
```

```
        dispatchers: '(PROTOCOL=TCP) (SERVICE=CDOMLOSRCA1DXDB)'
        enable_pluggable_database: TRUE
        log_archive_format: '%t_%s_%r.dbf'
        max_pdbs: 4098
        memory_max_target: 1342177280
        memory_target: 1342177280
        nls_language: 'AMERICAN'
        nls_territory: 'AMERICA'
        open_cursors: 300
        processes: 300
        remote_login_passwordfile: 'EXCLUSIVE'
    configTemplate: (unset)
    container: VCDO_1JL
    customEnvVars: (empty)
    linked: false
    logCollectionEnabled: false
    mountBase: /mnt/provision
    newDBID: false
    nodeListeners: (empty)
    operations:
       type: VirtualSourceOperations
        configureClone: (empty)
        postRefresh: (empty)
        postRollback: (empty)
        postSnapshot: (empty)
        postStart: (empty)
        postStop: (empty)
        preRefresh: (empty)
        preRollback: (empty)
        preSnapshot: (empty)
        preStart: (empty)
        preStop: (empty)
    parentTdeKeystorePassword: ********
    parentTdeKeystorePath: /u01/app/oracle/keystores/CDOMLOSR197/wallet
    redoLogGroups: 3
    redoLogSizeInMB: 50
    reference: ORACLE_VIRTUAL_PDB_SOURCE-2
    runtime:
        type: OraclePDBSourceRuntime
        accessible: true
        accessibleTimestamp: 2021-10-06T22:17:15.907Z
        activeInstances:
            0:
                type: OracleActiveInstance
                hostName: ip-10-110-234-67.delphix.com
                instanceName: CDOMLOSR197
                instanceNumber: 1
        databaseMode: READ_WRITE
        databaseRole: PRIMARY
        databaseSize: 913.4MB
        databaseStats: [ ... ]
        enabled: ENABLED
```

```
        lastNonLoggedLocation: 0
        status: RUNNING
    runtimeMountInformation:
        type: UnixRuntimeMountInformation
        name: (unset)
        nfsVersion: 4
        nfsVersionReason: DEFAULT
    staging: false
    status: DEFAULT
    tdeExportedKeyFileSecret: ********
    tdeKeyIdentifier: AVEhXrBvmU+Cv+lK6ghT6oMAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    tdeUUID: a3f26971-1df6-4c81-994f-4b2c582ded87
    virtual: true
```

7. To specify a user-defined encryption key to be used for future Delphix operations, set `tdeKeyIdentifier` to the value of a valid `key_id` in the CDB's keystore. This user-defined encryption key must be activated before updating it from Delphix CLI, otherwise subsequent Delphix operations may fail. Note that if an invalid `key_id` is provided, refresh or rewind will fail and it will be necessary to unset or update the `tdeKeyIdentifier` parameter with a valid `key_id`. Note that this `key_id` will not have a corresponding dlpx tag unless it is a key previously generated by Delphix.

```
delphix source 'VCDO_1JL'> update
delphix source 'VCDO_1JL' update *> set tdeKeyIdentifier="AbSP7gninU+Gv1YQ/
iEcJAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
delphix source 'VCDO_1JL' update *> ls
Properties
    type: OracleVirtualPdbSource
    name: VCDO_1JL
    allowAutoVDBRestartOnHostReboot: false
    allowRefreshRewindPostV2P: false
    customEnvVars: (empty)
    logCollectionEnabled: false
    newDBID: false
    operations:
        type: VirtualSourceOperations
        configureClone: (empty)
        postRefresh: (empty)
        postRollback: (empty)
        postSnapshot: (empty)
        postStart: (empty)
        postStop: (empty)
        preRefresh: (empty)
        preRollback: (empty)
        preSnapshot: (empty)
        preStart: (empty)
        preStop: (empty)
    parentTdeKeystorePassword: ********
```

```
    parentTdeKeystorePath: /u01/app/oracle/keystores/CDOMLOSR197/wallet
    tdeKeyIdentifier: AbSP7gninU+Gv1YQ/iEcJAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
delphix source 'VCDO_1JL' update *> commit
    Dispatched job JOB-22
    SOURCE_UPDATE job started for "VCDO_1JL".
    SOURCE_UPDATE job for "VCDO_1JL" completed successfully.
```

8. After a refresh or rewind, this key identifier will be associated with the vPDB and will be used for all future Delphix operations. View all the settings using the "ls" command.

```
delphix source 'VCDO_1JL'> ls
Properties
    type: OracleVirtualPdbSource
    name: VCDO_1JL
    allowAutoVDBRestartOnHostReboot: false
    allowRefreshRewindPostV2P: false
    archivelogMode: true
    config: VCDO_1JL
    configParams:
        _bct_public_dba_buffer_size: 1826784
        _cdb_disable_pdb_limit: TRUE
        audit_file_dest: '/u01/app/oracle/admin/CDOMLOSR197/adump'
        audit_trail: 'DB'
        compatible: '19.0.0'
        core_dump_dest: '/u01/app/oracle/diag/rdbms/cdomlosr197/CDOMLOSR197/
cdump'
        diagnostic_dest: '/u01/app/oracle'
        dispatchers: '(PROTOCOL=TCP) (SERVICE=CDOMLOSRCA1DXDB)'
        enable_pluggable_database: TRUE
        log_archive_format: '%t_%s_%r.dbf'
        max_pdbs: 4098
        memory_max_target: 1342177280
        memory_target: 1342177280
        nls_language: 'AMERICAN'
        nls_territory: 'AMERICA'
        open_cursors: 300
        processes: 300
        remote_login_passwordfile: 'EXCLUSIVE'
    configTemplate: (unset)
    container: VCDO_1JL
    customEnvVars: (empty)
    linked: false
    logCollectionEnabled: false
    mountBase: /mnt/provision
    newDBID: false
    nodeListeners: (empty)
    operations:
        type: VirtualSourceOperations
        configureClone: (empty)
        postRefresh: (empty)
```

```
        postRollback: (empty)
        postSnapshot: (empty)
        postStart: (empty)
        postStop: (empty)
        preRefresh: (empty)
        preRollback: (empty)
        preSnapshot: (empty)
        preStart: (empty)
        preStop: (empty)
    parentTdeKeystorePassword: ********
    parentTdeKeystorePath: /u01/app/oracle/keystores/CDOMLOSR197/wallet
    redoLogGroups: 3
    redoLogSizeInMB: 50
    reference: ORACLE_VIRTUAL_PDB_SOURCE-2
    runtime:
        type: OraclePDBSourceRuntime
        accessible: true
        accessibleTimestamp: 2021-10-06T22:17:15.907Z
        activeInstances:
            0:
                type: OracleActiveInstance
                hostName: ip-10-110-234-67.delphix.com
                instanceName: CDOMLOSR197
                instanceNumber: 1
        databaseMode: READ_WRITE
        databaseRole: PRIMARY
        databaseSize: 913.4MB
        databaseStats: [ ... ]
        enabled: ENABLED
        lastNonLoggedLocation: 0
        status: RUNNING
    runtimeMountInformation:
        type: UnixRuntimeMountInformation
        name: (unset)
        nfsVersion: 4
        nfsVersionReason: DEFAULT
    staging: false
    status: DEFAULT
    tdeExportedKeyFileSecret: ********
    tdeKeyIdentifier: AbSP7gninU+Gv1YQ/iEcJAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    tdeUUID: a3f26971-1df6-4c81-994f-4b2c582ded87
    virtual: true
```

13.2.4.5.36 CLI cookbook: Force refresh/rewind a virtual PDB

This topic describes how to force refresh/rewind a virtual pluggable database. To know more about the `force` option and when to use it refer to Refreshing or rewinding a broken/unusable virtual PDB (see page 1643).

13.2.4.5.36.1 Procedure

Refresh

1. Follow the steps in CLI Cookbook: Refresh a VDB from a specific timepoint or latest (see page 2654) to select a Specific or Latest Timepoint for refreshing the virtual pluggable database but do not commit.
2. Set the `force` parameter

```
delphix database '<virtual_pdb_name>' refresh *> set force=true
```

3. After all the parameters are set (including `timeflowPointParameters`), initiate the refresh by committing the operation in the CLI:

```
delphix database '<virtual_pdb_name>' refresh *> commit
```

Rewind (Rollback)

1. Follow the steps in Rolling back or rewinding to a snapshot from a VDB (see page 2658) to initiate rewind and set the `timeflowPointParameters` but do not commit.
2. Set the `force` parameter:

```
delphix database '<virtual_pdb_name>' rollback *> set force=true
```

3. After all the parameters are set (including `timeflowPointParameters`), initiate the rewind (rollback) by committing the operation in the CLI:

```
delphix database '<virtual_pdb_name>' rollback *> commit
```

13.2.4.5.37 CLI Cookbook: Starting or stopping cluster instances of an Oracle RAC virtual database

This topic describes the steps to start or stop cluster instances of an Oracle RAC virtual database.

13.2.4.5.37.1 Starting cluster instances of an Oracle RAC virtual database

Procedure

1. Log into the Delphix command-line interface using the admin user or a user with admin privileges.

```
$ ssh admin@YOUR_ENGINE
```

2. Select the source associated with the VDB. By default, sources are named the same as the VDB.

```
delphix> source "vdb"
```

3. Start a cluster instance by specifying desired instance number to the start command.

```
delphix> start; set instances.0=<instance number>; commit
```

Additional instance numbers can be specified using as 'instance.1', 'instances.2' and so on.

13.2.4.5.37.2 Stopping cluster instances of an Oracle RAC virtual database

Procedure

1. Log into the Delphix command-line interface using the admin user or a user with admin privileges.

```
$ ssh admin@YOUR_ENGINE
```

2. Select the source associated with the VDB. By default, sources are named the same as the VDB.

```
delphix> source "vdb"
```

3. Stop a cluster instance by specifying desired instance number to the stop command.

```
delphix> stop; set instances.0=<instance number>; commit
```

Additional instance numbers can be specified using as 'instance.1', 'instances.2' and so on.

13.2.4.6 CLI cookbook: replication

These topics describe how to use the command-line interface for replication tasks.

This section covers the following topics:

- CLI cookbook: adding a replication spec (see page 2702)
- CLI cookbook: deleting a replication spec (see page 2704)
- CLI cookbook: failing over a namespace (see page 2704)
- CLI cookbook: mapping replication Specs to Objects (see page 2705)

- CLI cookbook: triggering immediate execution of a replication spec (see page 2706)

13.2.4.6.1 CLI cookbook: adding a replication spec

This topic describes how to use the command-line interface to add a replication specification to the Delphix Engine.

Unlike the GUI, the CLI supports the ability to manage multiple replication specifications within a single system. This allows updates to be sent to multiple systems from a single point.

13.2.4.6.1.1 Prerequisites

You should review the topic Replication Overview (see page 2368) to understand which objects are copied as part of a backup or restore operation, as well as the dependencies between objects.

13.2.4.6.1.2 Procedure

1. Switch to the replication spec context.

```
delphix> cd replication/spec
delphix replication spec> ls
Operations
create
```

2. Create a new replication spec.

```
delphix replication spec> create
delphix replication spec create *> ls
Properties
    type: ReplicationSpec
    name: (unset)
    bandwidthLimit: (unset)
    enabled: (unset)
    encrypted: (unset)
    objects: (required)
    schedule: (unset)
    targetCredential:
        type: PasswordCredential
        password: (required)
    targetHost: (required)
    targetPrincipal: (required)
```

3. Specify the target hostname, user, and credentials.

```
delphix replication spec create *> set targetHost=exampleHost.mycompany.com
delphix replication spec create *> set targetPrincipal=delphix_admin
```

```
delphix replication spec create *> set targetCredential.password=password
```

Info:
SDD Secure replication
To create a Selective Data Distribution (SDD) type spec set the objectSpecification.type

1.
```
delphix replication spec create *> set
objectSpecification.type=ReplicationSecureList
```

This parameter defaults to ReplicationList which implies a regular replication spec.

Target Principal
The target principal must be a Delphix user on the target host who has domain privileges.

4. Specify the set of objects to replicate.
   a. To replicate all dSources and VDBs on the system, specify `` `DOMAIN `` as the list of objects.

   ```
   delphix replication spec create *> set
   objectSpecification.objects=`DOMAIN
   ```

   b. To replicate a subset of Groups, VDBs, dSources, and Timeflows specify their names as a comma-separated list.

   ```
   delphix replication spec create *> set
   objectSpecification.objects=Group1/vdb1,Group2/vdb2
   ```

1. **Name Completion**
   The CLI will provide possible completions for all objects in the system, but only groups, dSources and VDBs can be specified. Attempts to replicate other types of objects will generate an error when the operation is committed.
2. **SSD Secure replication**
   If objectSpecification.type=ReplicationSecureList was selected, then objectSpecification.containers needs to be used instead of objectSpecification.objects. The containers have to be Masked VDBs

5. Commit the operation.

```
delphix replication spec create *> commit
    `REPLICATION_SPEC-1
```

13.2.4.6.2 CLI cookbook: deleting a replication spec

This topic describes how to use the command-line interface to delete a replication spec.

13.2.4.6.2.1 Procedure

1. Switch to the replication spec context and list the specs on the system.

```
delphix> cd replication/spec
delphix replication spec> ls
Objects
REFERENCE           TARGETHOST
REPLICATION_SPEC-1  exampleHost.mycompany.com

Operations
create
```

2. Select the replication spec to remove.

```
delphix replication spec> select REPLICATION_SPEC-1
delphix replication spec "exampleHost.mycompany.com">
```

3. Remove the spec.

```
delphix replication spec "exampleHost.mycompany.com"> delete
delphix replication spec "exampleHost.mycompany.com" delete *> commit
```

13.2.4.6.3 CLI cookbook: failing over a namespace

This topic describes how to use the command line interface to fail over a namespace.

13.2.4.6.3.1 Procedure

1. Switch to the namespace context and list the namespaces on the system.

```
delphix> cd namespace
```

```
delphix namespace> ls
Objects
NAME
[172.16.203.93]

Operations
lookup
```

2. Select the namespace to failover.

```
delphix namespace> select [172.16.203.93]
delphix namespace "[172.16.203.93]">
```

3. Failover the namespace.

```
delphix namespace "[172.16.203.93]"> failover
delphix namespace "[172.16.203.93]" failover *> commit
```

Failover

Failover will sever the replication connection and make objects in the namespace part of the live system. This will prevent the target from receiving subsequent incremental updates.

13.2.4.6.4 CLI cookbook: mapping replication specs to objects

After creating replication specs often you will want to see what is mapping to which target, this document will show you how to find that information in the CLI. It will also show you how to navigate the replication directory in the CLI.

1. ssh into the CLI as a user with delphix_admin privileges.

```
ssh admin@yourengine
```

2. Next go to the replication directory in the CLI and list all of the specs.

```
delphix > replication
delphix replication > spec
delphix replication spec > ls
Objects
REFERENCE           TARGETHOST
REPLICATION_SPEC-1  test1
Operations
```

```
create
```

3. Then select the replication spec to find where is maps to and what objects are selected with it.

```
delphix replication spec> select REPLICATION_SPEC-1
delphix replication spec 'test'> ls
Properties
    type: ReplicationSpec
    name: test
    bandwidthLimit: 0
    description: (unset)
    enabled: false
    encrypted: false
    numberOfConnections: 1
    objectSpecification:
        type: ReplicationList
        name: (unset)
        objects: Untitled/dbdhcp3
    reference: REPLICATION_SPEC-1
    runtime:
        type: ReplicationSpecRuntime
    schedule: (unset)
    tag: 0ddae174-9486-4363-9704-bfc3398e547e
    targetCredential:
        type: PasswordCredential
        password: ********
    targetHost: test1
    targetPort: 8415
    targetPrincipal: delphix
    useSystemSocksSetting: false
Operations
delete
update
execute
```

13.2.4.6.5 CLI cookbook: triggering immediate execution of a replication spec

This topic describes how to use the command-line interface to trigger an immediate execution of a replication spec in the Delphix Engine.

13.2.4.6.5.1 Procedure

1. Switch to the replication spec context and list the specs on the system.

```
delphix> cd replication/spec
delphix replication spec> ls
Objects
```

```
REFERENCE           TARGETHOST
REPLICATION_SPEC-1  exampleHost.mycompany.com

Operations
create
```

2. Select the replication spec to execute.

```
delphix replication spec> select REPLICATION_SPEC-1
delphix replication spec "exampleHost.mycompany.com">
```

3. Execute the spec.

```
delphix replication spec "exampleHost.mycompany.com"> execute
delphix replication spec "exampleHost.mycompany.com" execute *> commit
    Dispatched job JOB-7
    REPLICATION_SEND job started.
    Connecting to target "exampleHost.mycompany.com".
    Preparing replication update.
    Starting incremental replication update.
    Sending metadata.
    Sending data for "Untitled".
    Sending data for "Untitled/redsox1".
    Transfer completed in 0:00:01, sent 1.39MB (1.39MB/s).
    Committing serialization state.
    REPLICATION_SEND job completed successfully.
```

13.2.4.7 CLI cookbook: Delphix self-service actions

These entries will help with some of the Delphix Self-Service Actions that can be performed in the CLI and therefore with the Delphix APIs:

- CLI cookbook: how to create a Delphix self-service bookmark (see page 2708)
- CLI cookbook: how to create a Delphix self-service branch (see page 2712)
- CLI cookbook: how to create a Delphix self-service container (see page 2716)
- CLI cookbook: how to create a Delphix self-service database template (see page 2721)
- CLI cookbook: how to create a Delphix self-service user (see page 2724)
- CLI cookbook: how to delete a Delphix self-service bookmark (see page 2728)
- CLI cookbook: how to delete a Delphix self-service container (see page 2729)
- CLI cookbook: how to delete a Delphix self-service template (see page 2729)
- CLI cookbook: how to refresh a Delphix self-service container (see page 2730)
- CLI cookbook: how to share a Delphix self-service bookmark (see page 2734)
- CLI cookbook: how to update a Delphix self-service bookmark (see page 2737)

13.2.4.7.1 CLI cookbook: how to create a Delphix self-service bookmark

13.2.4.7.1.1 Prerequisites:

- Have Delphix Self-Service user privileges.
- Know the branch you would like to create the bookmark on.
- Know the container or template that the branch belongs to.
- (Optional) Know when you would like the bookmark to expire.

Delphix Self-Service administrators can use this CLI cookbook recipe to create a bookmark on Delphix Self-Service using the Delphix Engine CLI.

> The following script is for educational and demonstration purposes only and is not supported by Delphix.

This script can be downloaded by selecting createBookmark.sh[768].

Creating a bookmark in Delphix Self-Service

```
#!/bin/bash
# A sample script for calls to the CLI. This one creates a Jet Stream bookmark.
#
# VERY IMPORTANT: In order for this to work, you need to go through the steps here:
# https://docs.delphix.com/display/DOCS43/CLI+Cookbook%3A+Configuring+Key-
Based+SSH+Authentication+for+Automation
# After this you will not need to use a username and password to log into the delphix
engine. If you do not
# setup the SSH authentication you will have to manually enter the password.
#


##example##
#./createBookmark.sh -e "2016-07-27T23:38:56.453Z" -t "2016-07-27T01:45:56.453Z" -T
[tag1,tag2,tag3,tag4,tag5] bkmrk3 JS_BRANCH-41


##### Constants

# Describes a Delphix software revision.
VERSION="1.11.10"


##### Default Values. These can be overwriten with optional arguments.
engine="ars-6010.dlpxdc.co"
```

768 https://docs.delphix.com/docs/files/191896831/191896832/1/1629788977903/createBookmark.sh

```
username="admin"



##### Functions


# Help Menu
function usage {
    echo "Usage: createBookmark.sh [[-h] | options...] <bookmark name> <branch name
format JS_BRANCH-n>"
    echo "Create a Jet Stream Bookmark on the given branch."
    echo ""
    echo "Positional arguments"
    echo "  <bookmark name> "
    echo "  <branch name>"
    echo ""
    echo "Optional Arguments:"
    echo "  -h                Show this message and exit"
    echo "  -d                Delphix engine IP address or host name, otherwise
revert to default"
    echo "  -u                Server user. Password needs to manually provide at run
time, otherwise revert to default"
    echo "  -D                Description of this bookmark. Type: string"
    echo "  -s                Pass [-s true]  if need to make bookmark in shared
mode"
    echo "  -e                The time at which the bookmark should be expired"
    echo "  -t                The time at which the bookmark should be created. If no
time is included, the bookmark will be created at the latest point in time. Type:
date, must be in ISO 8601 extended format [yyyy]-[MM]-[dd]T[HH]:[mm]:[ss].[SSS]Z"
    echo "  -T                A set of user-defined labels for this bookmark. No
spaces allowed. Array of Type: string. In format, [tag1,tag2,..] "
}

# Create Our Session, including establishing the API version.
function create_session
{
    echo "creating session..."
    SSH_CMD="ssh ${username}@${engine}"
    ${SSH_CMD}  "version $VERSION"
    check_result
}


# Check the result of the curl. If there are problems, inform the user then exit.
function check_result
{
    exitStatus=$?
    if [ $exitStatus -ne 0 ]
    then
        echo "command failed with exit status $exitStatus"
        echo $result
        exit 1
```

```
    fi
}


function get_branch
{
    echo "retrieveing branch $branchRef to find Source Data Layout..."
    result=$(${SSH_CMD}  "selfservice branch; ls; select ${branchRef}; ls")

    # Get everything in the result that comes after dataLayout.
    temp=${result#*dataLayout: }

    # Get rid of everything after
    dataLayout=${temp%% *}

}



function create_bookmark
{
    get_branch

    echo "creating bookmark..."

    # If there is not creation time, we need to use JSTimelinePointLatestTimeInput.
    if [ -z $creationTime ]
    then
        pointParams="set timelinePointParameters.sourceDataLayout=$dataLayout;"
        pointParams="$pointParams set
timelinePointParameters.type=JSTimelinePointLatestTimeInput;"
    else
        pointParams="set timelinePointParameters.sourceDataLayout=$dataLayout;"
        pointParams="$pointParams set
timelinePointParameters.type=JSTimelinePointTimeInput;"
        pointParams="$pointParams set
timelinePointParameters.time=\"$creationTime\";"
        pointParams="$pointParams set timelinePointParameters.branch=\"$branchRef\";"

    fi

    if [[ -n $expirationTime ]]
    then
        pointParams="$pointParams set bookmark.expiration=\"$expirationTime\";"
    fi

    # These are the required parameters.
    paramString="
        \"bookmark\": {
            \"branch\": \"${branchRef}\",
            \"name\": \"${bookmarkName}\","
    paramString="selfservice bookmark; create; set bookmark.name=$bookmarkName; set
bookmark.branch=$branchRef;"
```

```
    paramString="$paramString $pointParams"

    # Fill in optional parameters if there are any.
    if [[ -n $description ]]
    then
        paramString="$paramString set bookmark.description=\"$description\";"
    fi

    if [[ -n $shared ]]
    then
        paramString="$paramString set bookmark.shared=$shared;"
    fi

    if [[ -n $tags ]]
    then
        # Add quotes back to the passed in tags so they are processed correctly.
        tags=${tags//[/[\"}
        tags=${tags//,/\",\"}
        tags=${tags//]/\"]}

        paramString="$paramString set bookmark.tags=$tags;"
    fi
    paramString="$paramString commit;"
    result=$(${SSH_CMD} $paramString)
    check_result
    echo "Verifying job status..."
    # Get everything in the result that comes after job.
        temp=${result#*job}
    # Get rid of everything after
    resultArray=($temp)
    jobRef=($resultArray)
        jobString="job;select $jobRef;ls"
        result=$(${SSH_CMD} $jobString)
    check_result
    # Get everything in the result that comes after job.
    temp=${result#*jobState:}
    # Get rid of everything after
    resultArray=($temp)
    jobState=($resultArray)

    if [ $jobState = "COMPLETED" ]
    then
        echo "successfully created bookmark $bookmarkName"
    else
        echo "unable to create bookmark"
        echo result
    fi

}


##### Main
```

```
while getopts "u:d:D:s:e:t:T:h" flag; do
    case "$flag" in
            u )             username=${OPTARG%:*}
                        ;;
        d )             engine=$OPTARG
                        ;;
        D )             description=$OPTARG
                        ;;
        s )             shared=true
                        ;;
        t )             creationTime=$OPTARG
                        ;;
        e )             expirationTime=$OPTARG
                        ;;
        T )             tags=$OPTARG
                        ;;
        h )             usage
                        exit
                        ;;
        * )             usage
                        exit 1
    esac
done


# Shift the parameters so we only have the positional arguments left
shift $((OPTIND-1))

# Check that there are 2 positional arguments
if [ $# != 2 ]
then
    usage
    exit 1
fi

# Get the two positional arguments
bookmarkName=$1
shift
branchRef=$1
create_session
create_bookmark
```

13.2.4.7.2 CLI cookbook: how to create a Delphix self-service branch

Delphix Self-Service administrators can use this CLI cookbook recipe to create a branch on Delphix Self-Service using the Delphix Engine CLI.

> The following script is for educational and demonstration purposes only and is not supported by Delphix.

This script can be downloaded by selecting **createBranch.sh**[769].

Creating a branch in Delphix Self-Service

```
#!/bin/bash

# A sample script for calls to the CLI. This one creates a Jet Stream branch.
#
# VERY IMPORTANT: In order for this to work, you need to go through the steps here:
# https://docs.delphix.com/display/DOCS43/CLI+Cookbook%3A+Configuring+Key-
Based+SSH+Authentication+for+Automation
# After this you will not need to use a username and password to log into the delphix
engine. If you do not
# setup the SSH authentication you will have to manually enter the password.
#
# Note that the CLI only allows branches to be created from below
# 1) From latest point in time
# 2) From specific bookmark
# 3) From specific point in time



##examples##
# Create branch from latest point in time
#./createBranch.sh NewBranchName JS_DATA_CONTAINER-20
# Create branch from specific bookmark
#./createBranch.sh -b ExistingBookmarkName NewBranchName JS_DATA_CONTAINER
# Create branch from specific point in time
#./createBranch.sh -t "2016-07-27T01:45:56.453Z" -B JS_BRANCH-2 NewBranchName
JS_DATA_CONTAINER-20


##### Constants

# Describes a Delphix software revision.
VERSION="1.11.10"


##### Default Values. These can be overwriten with optional arguments.
engine="ars3-6010.dlpxdc.co"
username="admin"



##### Functions
```

769 https://docs.delphix.com/docs/files/191896834/191896835/1/1629714698220/createBranch.sh

```
# Help Menu
function usage {
    echo "Usage: createBranch.sh [[-h] | options...] <name> <container>"
    echo "Create a Jet Stream Bookmark on the given branch."
    echo ""
    echo "Positional arguments"
    echo "  <NewBranchName>"
    echo "  <container> format JS_DATA_CONTAINER-<n>"
    echo ""
    echo "Optional Arguments:"
    echo "  -h                Show this message and exit"
    echo "  -d                Delphix engine IP address or host name, otherwise
revert to default"
    echo "  -u                Server user. Password needs to manually provide at run
time, otherwise revert to default"
    echo "  -b                Bookmark name from which need to create branch. If no
bookmark is included, the branch will be created at the latest point in time. Type:
string"
    echo "  -t                The time at which the branch should be created. If no
time is included, the branch will be created at the latest point in time. Type: date,
must be in ISO 8601 extended format [yyyy]-[MM]-[dd]T[HH]:[mm]:[ss].[SSS]Z"
}

# Create Our Session, including establishing the API version.
function create_session
{
    echo "creating session..."
    SSH_CMD="ssh ${username}@${engine}"
    ${SSH_CMD}  "version $VERSION"
    check_result
}


# Check the result of the curl. If there are problems, inform the user then exit.
function check_result
{
    exitStatus=$?
    if [ $exitStatus -ne 0 ]
    then
        echo "command failed with exit status $exitStatus"
        echo $result
        exit 1
    fi
}



function create_branch
{
```

```
    # If there is not timeInput and no bookmark name, we need to use
JSTimelinePointLatestTimeInput.
    if [ -z $inputTime ] && [ -z $bookmark ]
    then
        #code to use JSTimelinePointLatestTimeInput
        pointParams="edit timelinePointParameters; set
type=JSTimelinePointLatestTimeInput;"
        pointParams="$pointParams set sourceDataLayout=$container; commit;"
    # If there is a timeInput and no bookmark name, we need to use Input Time.
    elif [ -n $inputTime ] && [ -z $bookmark ]
     then
        #code to use JSTimelinePointTimeInput
        pointParams="edit timelinePointParameters; set
type=JSTimelinePointTimeInput;"
        pointParams="$pointParams set time=$inputTime; set branch=$branch; commit;"
    # If there is a bookmark name and no time input, we need to use bookmark
    elif [ -z $inputTime ] && [ -n $bookmark ]
    then
        #code to use JSTimelinePointBookmarkInput
        pointParams="set timelinePointParameters.bookmark=$bookmark;"
        pointParams="$pointParams commit;"
    fi

    # These are the required parameters.
    paramString="selfservice branch create;set name=$branchName; set
dataContainer=$container;"

    #Add additional optional parameter
    paramString="$paramString $pointParams"
    #echo $paramString
    echo "Creating Branch..."
    result=$(${SSH_CMD}  $paramString)
    check_result

    echo "Verifying job status..."
    # Get everything in the result that comes after job.
        temp=${result#*job}
    # Get rid of everything after
    resultArray=($temp)
    jobRef=($resultArray)
        jobString="job;select $jobRef;ls"
        result=$(${SSH_CMD} $jobString)
    check_result
    # Get everything in the result that comes after job.
    temp=${result#*jobState:}
    # Get rid of everything after
    resultArray=($temp)
    jobState=($resultArray)
        if [ $jobState = "COMPLETED" ]
    then
        echo "Successfully created branch $branchName"
    else
        echo "Unable to create branch"
```

```
        echo $result
    fi

}


##### Main

while getopts "u:d:b:t:B:h" flag; do
    case "$flag" in
        u )             username=${OPTARG%:*}
                        ;;
        d )             engine=$OPTARG
                        ;;
        b )             bookmark=$OPTARG
                        ;;
        B )             branch=$OPTARG
                        ;;
        t )             inputTime=$OPTARG
                        ;;
        h )             usage
                        exit
                        ;;
        * )             usage
                        exit 1
    esac
done


# Shift the parameters so we only have the positional arguments left
shift $((OPTIND-1))

# Check that there are 2 positional arguments
if [ $# != 2 ]
then
    usage
    exit 1
fi

# Get the two positional arguments
branchName=$1
shift
container=$1

create_session
create_branch
```

13.2.4.7.3 CLI cookbook: how to create a Delphix self-service container

Delphix Self-Service administrators can use this CLI cookbook recipe to create a container on Delphix Self-Service using the Delphix Engine CLI.

> The following script is for educational and demonstration purposes only and is not supported by Delphix.

This script can be downloaded by selecting createContainer.sh.

Creating a container in Delphix self-service container

```
#!/bin/bash

# A sample script for calls to the CLI. This one creates a Jet Stream container.
#
# VERY IMPORTANT: In order for this to work, you need to go through the steps here:
# https://docs.delphix.com/display/DOCS43/CLI+Cookbook%3A+Configuring+Key-
Based+SSH+Authentication+for+Automation
# After this you will not need to use a username and password to log into the delphix
engine. If you do not
# setup the SSH authentication you will have to manually enter the password.
#


##### Constants

# Describes a Delphix software revision.
VERSION="1.11.10"


##### Default Values. These can be overwriten with optional arguments.
engine="name-of-engine.dlpxdc.co"
username="admin"



##examples##
# Create container from latest point in time
#./createContainer.sh -n "testsource" testcont ORACLE_DB_CONTAINER-269
 JS_DATA_TEMPLATE-13
# Create container from specific bookmark
#./createContainer.sh -n "testsource" -b JS_BOOKMARK-77 testcont ORACLE_DB_CONTAINER-2
69 JS_DATA_TEMPLATE-13
# Create container from specific point in time
#./createContainer.sh -n "testsource" -t "2016-08-08T10:00:00.000Z" -B JS_BRANCH-50
 testcont ORACLE_DB_CONTAINER-269 JS_DATA_TEMPLATE-13

#NOTE: this script will add one container and assign one owner for the container.


##### Functions

# Help Menu
```

```
function usage {
    echo "Usage: createContainer.sh [[-h] | options...] <containername> <vdb>
<template>"
    echo "Create a Jet Stream Container."
    echo ""
    echo "Positional arguments"
    echo "  <name>"
    echo "  <container> format JS_DATA_CONTAINER-<n>"
    echo ""
    echo "Optional Arguments:"
    echo "  -h                Show this message and exit"
    echo "  -d                Delphix engine IP address or host name, otherwise
revert to default"
    echo "  -u                Server user. Password needs to manually provide at run
time, otherwise revert to default"
        echo "  -n                SourceName need to display for container.
(Mandatory)"
    echo "  -b                Bookmark name from which need to create container. If
no bookmark is included, the branch will be created at the latest point in time.
Type: string. Format JS_BOOKMARK-<n> (Optional)"
    echo "  -B                Branch reference from which we need to pick up time
from where the container should be created. Type: string. Format JS_BRANCH-<n>"
    echo "  -t                The time at which the branch should be created. This
must be accompanied with branch name from which need to pick up time. Type: date,
must be in ISO 8601 extended format [yyyy]-[MM]-[dd]T[HH]:[mm]:[ss].[SSS]Z"
    echo "  -N                Optional container notes, if need to add any. Type:
String"
    echo "  -o                Optional owner, to whom we need to assign this
container. Type: String. Format USER-<n>"
}

# Create Our Session, including establishing the API version.
function create_session
{
    echo "creating session..."
    SSH_CMD="ssh ${username}@${engine}"
    ${SSH_CMD}  "version $VERSION"
    check_result
}

# Check the result of the curl. If there are problems, inform the user then exit.
function check_result
{
    exitStatus=$?
    if [ $exitStatus -ne 0 ]
    then
        echo "command failed with exit status $exitStatus"
        echo $result
        exit 1
    fi
}
```

```
function create_container
{
    # If there is not timeInput and no bookmark name, we need to use
JSTimelinePointLatestTimeInput.
    if [[ -z $inputTime  &&  -z $bookmark ]]
    then
        pointParams="set
timelinePointParameters.type=JSTimelinePointLatestTimeInput;"
        pointParams="$pointParams set
timelinePointParameters.sourceDataLayout=$template;"

   # If there is a timeInput, no bookmark name and a branch name, we need to use
Input Time.
    elif [[ -n $inputTime  && -z $bookmark && -n $branch ]]
     then
        pointParams="set timelinePointParameters.type=JSTimelinePointTimeInput;"
        pointParams="$pointParams set timelinePointParameters.time=\"${inputTime}\";"
        pointParams="$pointParams set timelinePointParameters.branch=$branch;"
# If there is a bookmark name and no time input, we need to use bookmark
    elif [[ -z $inputTime  &&  -n $bookmark ]]
     then
        pointParams="set timelinePointParameters.type=JSTimelinePointBookmarkInput;"
        pointParams="$pointParams set timelinePointParameters.bookmark=\"${bookmark}
\";"
    else
        usage
        exit 1
    fi

    # These are the required parameters.

    paramString="selfservice container create;set name=\"${containerName}\";"
    paramString="$paramString set template=\"${template}\";"

    if [[ -n $containerNotes ]]
    then
       paramString="$paramString set notes=\"${containerNotes}\";"
    fi

    if [[ -n $owners ]]
    then
      paramString="$paramString set owner=\"${owners}\";"
    fi
    paramString="$paramString $pointParams;"
    paramString="$paramString edit dataSources; add; set container=$VDB;"
    paramString="$paramString edit source; set type=JSDataSource; set priority=1;set
name=\"${sourceName}\";"

    if [[ -n $sourcedesc ]]
    then
       paramString="$paramString set description=\"${sourcedesc}\";"
    fi
```

```
    paramString="$paramString commit;"
    #echo $paramString
    echo "Creating Container..."
    result=$(${SSH_CMD}  $paramString)
    check_result

    echo "Verifying job status..."
    # Get everything in the result that comes after job.
        temp=${result#*job}
    # Get rid of everything after
    resultArray=($temp)
    jobRef=($resultArray)
        jobString="job;select $jobRef;ls"
        result=$(${SSH_CMD} $jobString)
    check_result
    # Get everything in the result that comes after job.
    temp=${result#*jobState:}
    # Get rid of everything after
    resultArray=($temp)
    jobState=($resultArray)
        if [ $jobState = "COMPLETED" ]
    then
        echo "Successfully created Container $containerName"
    else
        echo "Unable to create Container"
        echo $result
    fi
}

##### Main

while getopts "u:d:b:t:B:D:n:N:o:h" flag; do
    case "$flag" in
        u )             username=${OPTARG%:*}
                        ;;
        d )             engine=$OPTARG
                        ;;
        b )             bookmark=$OPTARG
                        ;;
        t )             inputTime=$OPTARG
                        ;;
        D )             sourcedesc=$OPTARG
                        ;;
        n )             sourceName=$OPTARG
                        ;;
    B )             branch=$OPTARG
            ;;
        N )             containerNotes=$OPTARG
                        ;;
        o )             owners=$OPTARG
                        ;;
        h )             usage
```

```
                        exit
                        ;;
        * )             usage
                        exit 1
    esac
done


# Shift the parameters so we only have the positional arguments left
shift $((OPTIND-1))

# Check that there are 3 positional arguments
if [ $# != 3 ]
then
    usage
    exit 1
fi

# Get the three positional arguments
containerName=$1
shift
VDB=$1
shift
template=$1

create_session
create_container
```

13.2.4.7.4 CLI cookbook: how to create a Delphix self-service database template

Delphix Self-Service administrators can use this CLI cookbook recipe to create a database template on Delphix Self-Service using the Delphix Engine CLI.

> The following script is for educational and demonstration purposes only and is not supported by Delphix.

This script can be downloaded by selecting createDBTemplate.sh[770].

Creating a database template in Delphix Self-Service

```
#!/bin/bash

# A sample script for calls to the CLI. This one creates a Jet Stream Template.
#
# VERY IMPORTANT: In order for this to work, you need to go through the steps here:
```

770 https://docs.delphix.com/docs/files/191896840/191896841/1/1629283143725/createDBTemplate.sh

```
# https://docs.delphix.com/display/DOCS43/CLI+Cookbook%3A+Configuring+Key-
Based+SSH+Authentication+for+Automation
# After this you will not need to use a username and password to log into the delphix
engine. If you do not
# setup the SSH authentication you will have to manually enter the password.
#
#

##### Constants

# Describes a Delphix software revision.
VERSION="1.11.10"


##### Default Values. These can be overwriten with optional arguments.
engine="ars-6010.dlpxdc.co"
username="admin"


##examples##
# Create template with mandatory params
# Create template with adding optional params, Notes and Description
#./createDBTemplate.sh -n <sourceName> -N "<templateNotes>" -D "<AnyDescription>"
 <templateName> <containerName>

## NOTE: This script is to add one source per template and it will not add any
properties for template, container or source.


##### Functions

# Help Menu
function usage {
    echo "Usage: createDBTemplate.sh [[-h] | options...] <template_name>
<source_container>"
    echo "Create a Jet Stream Dat Template."
    echo ""
    echo "Positional arguments"
    echo "  <template_name>"
    echo "  <source_container>"
    echo ""
    echo "Optional Arguments:"
    echo "  -h                Show this message and exit"
    echo "  -d                Delphix engine IP address or host name, otherwise
revert to default"
    echo "  -u        Server user. Password needs to manually provide at run time,
otherwise revert to default"
    echo "  -n                source name to display on JS template screen"
    echo "  -N                template notes, if any. Type: String"
    echo "  -D                source description, if any. Type: String"
}

# Create Our Session, including establishing the API version.
```

```
function create_session
{
    echo "creating session..."
    SSH_CMD="ssh ${username}@${engine}"
    ${SSH_CMD}  "version $VERSION"
    check_result
}

# Check the result of the curl. If there are problems, inform the user then exit.
function check_result
{
    exitStatus=$?
    if [ $exitStatus -ne 0 ]
    then
        echo "command failed with exit status $exitStatus"
        echo $result
        exit 1
    fi
}



function create_template
{
    paramString="selfservice template create;set name=$templateName;"
    if [[ -n $templatenotes ]]
        then
           paramString="$paramString set notes=\"$templatenotes\";"
    fi
    paramString="$paramString edit dataSources;add;"
    paramString="$paramString set container=$sourceContainer;set
source.name=$sourcename;set source.priority=1;"
    if [[ -n $sourcedesc ]]
    then
       paramString="$paramString set source.description=\"$sourcedesc\";"
    fi
    paramString="$paramString commit;"
    #echo $paramString
    echo "Creating Data Template..."
    result=$(${SSH_CMD}  $paramString)
    check_result
    echo "New JetStream template $templateName successfully created"
}


##### Main

while getopts "u:d:n:N:D:h" flag; do
    case "$flag" in
        u )             username=${OPTARG%:*}
                        ;;
        d )             engine=$OPTARG
                        ;;
```

```
        n )             sourcename=$OPTARG
                        ;;
        N )             templatenotes=$OPTARG
                        ;;
        D )             sourcedesc=$OPTARG
                        ;;
        h )             usage
                        exit
                        ;;
        * )             usage
                        exit 1

    esac
done

# Shift the parameters so we only have the positional arguments left
shift $((OPTIND-1))

# Check that there are 2 positional arguments
if [ $# != 2 ]
then
    usage
    exit 1
fi

# Get the two positional arguments
templateName=$1
shift
sourceContainer=$1


create_session
create_template
```

13.2.4.7.5 CLI cookbook: how to create a Delphix self-service user

Delphix Self-Service administrators can use this CLI cookbook recipe to create a user on Delphix Self-Service using the Delphix Engine CLI.

> The following script is for educational and demonstration purposes only and is not supported by Delphix.

This script can be downloaded by selecting createJSUser.sh[771].

Creating a Delphix self-service user

771 https://docs.delphix.com/docs/files/191896843/191896844/1/1652831344899/createJSUser+%285%29.sh

```
#!/bin/bash
# A sample script for calls to the CLI. This one creates a Self-Service user.
#
# VERY IMPORTANT: In order for this to work, you need to go through the steps here:
# https://docs.delphix.com/display/DOCS533/CLI+Cookbook%3A+Configuring+Key-
Based+SSH+Authentication+for+Automation
# After this you will not need to use a username and password to log into the delphix
engine. If you do not
# setup the SSH authentication you will have to manually enter the password.
#
# Note that the CLI only allows branches to be created from existing bookmarks.
##### Constants
# Describes a Delphix software revision.
VERSION="1.6.2"

##### Default Values. These can be overwriten with optional arguments.
engine="10.0.1.10"
username="dev"

##examples##
# Create user with NATIVE authentication
#./createJSUser.sh -P <password> NATIVE <username>
# Create user with LDAP authentication
#./createJSUser.sh -r <principal> <LDAP username>

##### Functions
# Help Menu
function usage {
echo "Usage: createJSUser.sh [[-h] | options...] <auth> <newjsuser>"
echo "Create a Self-Service Only user."
echo ""
echo "Positional arguments"
echo " <auth type NATIVE/LDAP>"
echo " <newjsuser>"
echo ""
echo "Optional Arguments:"
echo " -h Show this message and exit"
echo " -d Delphix engine IP address or host name, otherwise revert to default"
echo " -u Server user. Password needs to manually provide at run time, otherwise
revert to default"
echo " -P password for NATIVE authentication, MUST incase auth=NATIVE"
echo " -f firstName of user"
echo " -l lastName of user"
echo " -e emailAddress of user"
echo " -o homePhoneNumber of user"
echo " -m mobilePhoneNumber of user"
echo " -w workPhoneNumber of user"
echo " -r principal for LDAP authentication, MUST incase of auth=LDAP"
}

# Create Our Session, including establishing the API version.
function create_session
```

```
{
echo "creating session..."
SSH_CMD="ssh ${username}@${engine}"
${SSH_CMD} "version $VERSION"
check_result
}
# Check the result of the curl. If there are problems, inform the user then exit.
function check_result
{
exitStatus=$?
if [ $exitStatus -ne 0 ]
then
echo "command failed with exit status $exitStatus"
echo $result
exit 1
fi
}

function create_user
{
# Check on authorization type
paramString="user create;"
if [[ $authtype = "NATIVE" && -n $userpwd ]]
then
pointParams="set authenticationType=$authtype;"
pointParams="$pointParams set credential.type=PasswordCredential; set
credential.password=$userpwd;"
elif [[ $authtype = "LDAP" && -n $principal ]]
then
pointParams="set authenticationType=$authtype; set principal=$principal;"
fi
# These are the required parameters.
paramString="$paramString set type=User; set name=$newjsuser;"

# Fill in optional parameters if there are any.
if [[ -n $firstname ]]
then
paramString="$paramString set firstName=\"$firstname\";"
fi

if [[ -n $lastname ]]
then
paramString="$paramString set lastName=\"$lastname\";"
fi

if [[ -n $emailaddress ]]
then
paramString="$paramString set emailAddress=\"$emailaddress\";"
fi

if [[ -n $homephone ]]
then
paramString="$paramString set homePhoneNumber=\"$homephone\";"
```

```
fi

if [[ -n $mobilephone ]]
then
paramString="$paramString set mobilePhoneNumber=\"$mobilephone\";"
fi

if [[ -n $workphone ]]
then
paramString="$paramString set workPhoneNumber=\"$workphone\";"
fi
paramString="$paramString ${pointParams} commit;"
#echo $paramString
echo "Creating user..."
result=$(${SSH_CMD} $paramString)
check_result
#echo $result
echo "New user $newjsuser successfully created"

##### ROLE-3 is Self-Service Role
paramString="authorization create;"
paramString="$paramString set type=Authorization; set role=ROLE-3; set
target=$newjsuser; set user=$newjsuser;commit;"
#echo $paramString
result=$(${SSH_CMD} $paramString)
check_result
echo "Assigned Self-Service Role to user $newjsuser"
}

##### Main
##### Main
while getopts "u:d:P:r:f:l:e:o:m:w:h" flag; do
case "$flag" in
u ) username=${OPTARG%:*}
;;
d ) engine=$OPTARG
;;
P ) userpwd=$OPTARG
;;
r ) principal=$OPTARG
;;
f ) firstname=$OPTARG
;;
l ) lastname=$OPTARG
;;
e ) emailaddress=$OPTARG
;;
o ) homephone=$OPTARG
;;
m ) mobilephone=$OPTARG
;;
w ) workphone=$OPTARG
;;
```

```
h ) usage
exit
;;
* ) usage
exit 1

esac
done

# Shift the parameters so we only have the positional arguments left
shift $((OPTIND-1))
# Check that there are 2 positional arguments
if [ $# != 2 ]
then
echo "usage1"
usage
exit 1
fi
# Get the two positional arguments
authtype=$1
shift
newjsuser=$1
create_session
create_user
```

13.2.4.7.6 CLI cookbook: how to delete a Delphix self-service bookmark

13.2.4.7.6.1 Prerequisites:

- Have Delphix Self-Service user privileges
- Know the bookmark you would like to delete

13.2.4.7.6.2 Procedure:

1. Log into Delphix Engine as a Delphix Self-Serviceuser or admin.

   ```
   ssh jsUser@<yourengine>
   ```

2. Navigate to the Delphix Self-Servicebookmarks, and choose the one you would like to delete.

   ```
   jsUser > selfservice
   bookmark
   jsUser selfservice bookmark > ls
   jsUser selfservice bookmark > select <jsBookmark>
   ```

3. Delete the bookmark.

```
jsUser selfservice bookmark <jsBookmark> *> delete
jsUser selfservice bookmark <jsBookmark> delete *> commit
```

13.2.4.7.7 CLI cookbook: how to delete a Delphix self-service container

13.2.4.7.7.1 Prerequisites:

- Know the container you want to delete
- Have delphix_admin privileges

13.2.4.7.7.2 Procedure:

1. Log into Delphix Engine as admin

```
ssh delphix@<yourengine>
```

2. Navigate to the Delphix Self-Servicecontainer that you want to delete

```
delphix > selfservice container
delphix selfservice container > ls
delphix selfservice container > select CONTAINER_X
delphix selfservice container 'CONTAINER_X' >
```

3. Delete container, **note you need to set if you want to delete the VDBs in the container (false will preserve the VDBs and true the VDBs will be deleted along with the container)**

```
delphix selfservice container 'CONTAINER_X' > delete
delphix selfservice container 'CONTAINER_X'delete *> set deleteDataSources=<true/false>
delphix selfservice container 'CONTAINER_X'delete *> commit
```

13.2.4.7.8 CLI cookbook: how to delete a Delphix self-service template

13.2.4.7.8.1 Prerequisites:

- Template has no dependant containers
- Know the name of the template you are going to delete
- Have delphix_admin privileges (note JetStream Only users and Delphix GUI Owners cannot delete templates)

13.2.4.7.8.2 Procedure:

1. Log into your Delphix Engine using delphix_admin (or admin privileged account)

```
ssh delphix_admin@<yourengine>
```

2. Find the template you want to delete

```
delphix > selfservice
delphix selfservice template > ls
delphix selfservice template > select 'TEMPLATE_X'
delphix selfservice template 'TEMPLATE_X'>
```

3. Delete template and commit

```
delphix selfservice template 'TEMPLATE_X'> delete
delphix selfservice template 'TEMPLATE_X' delete *> commit
```

13.2.4.7.9 CLI cookbook: how to refresh a Delphix self-service container

> The following script is for educational and demonstration purposes only and is not supported by Delphix.

This script can be downloaded by selecting **refreshContainer.sh**[772].

Refreshing a container in Delphix self-service

```
#!/bin/bash

# A sample script for calls to the CLI. This one refresh Jet Stream container.
#
# VERY IMPORTANT: In order for this to work, you need to go through the steps here:
# https://docs.delphix.com/display/DOCS43/CLI+Cookbook%3A+Configuring+Key-
Based+SSH+Authentication+for+Automation
# After this you will not need to use a username and password to log into the delphix
engine. If you do not
# setup the SSH authentication you will have to manually enter the password.
#
```

772 https://docs.delphix.com/docs/files/191896848/191896849/1/1625398855538/refreshContainer+%286.0.4.1%29.sh

```
##### Constants

# Describes a Delphix software revision.
VERSION="1.11.4"


##### Default Values. These can be overwriten with optional arguments.
engine="de-6041-doc-938-vn.dlpxdc.co"
username="admin"


##examples##
# Refresh container from latest point in time of Template
#./refreshContainer.sh -T JS_DATA_TEMPLATE-13 JS_DATA_CONTAINER-20
# Refresh container from specific bookmark
#./refreshContainer.sh -b JS_BOOKMARK-76 JS_DATA_CONTAINER-20
# Refresh container from specific point in time of branch
#./refreshContainer.sh -t "2016-08-08T10:00:00.000Z" -B JS_BRANCH-4
 JS_DATA_CONTAINER-20


##### Functions

# Help Menu
function usage {
echo "Usage: refreshContainer.sh [[-h] | options...] <containername>"
echo "Create a Jet Stream Bookmark on the given branch."
echo ""
echo " You need to specify either -T, -B, -b or -t to refresh container. With -t
option you also need to specify -B also"
echo "Positional arguments"
echo " <containerName>"
echo ""
echo "Optional Arguments:"
echo " -h Show this message and exit"
echo " -d Delphix engine IP address or host name, otherwise revert to default"
echo " -u Server user. Password needs to manually provide at run time, otherwise
revert to default"
echo " -T template reference from which need to refresh from latest point in time"
echo " -B Branch reference from which we need to pick up time from where the
container should be refreshed. Type: string. Format JS_BRANCH-<n> (Optional)"
echo " -b Bookmark name from which need to refresh container. If no bookmark is
included, the branch will be created at the latest point in time. Type: string.
Format JS_BOOKMARK-<n> (Optional)"
echo " -t The time from where the container should be refreshed. This must be
accompanied with branch reference from which need to pick up time. Type: date, must
be in ISO 8601 extended format [yyyy]-[MM]-[dd]T[HH]:[mm]:[ss].[SSS]Z"
}

# Create Our Session, including establishing the API version.
function create_session
{
```

```
echo "Creating session ..."
SSH_CMD="ssh ${username}@${engine}"
${SSH_CMD} "version $VERSION"
check_result
}

# Check the result of the curl. If there are problems, inform the user then exit.
function check_result
{
exitStatus=$?
if [ $exitStatus -ne 0 ]
then
echo "command failed with exit status $exitStatus"
echo $result
exit 1
fi
}


function restore_container
{

paramString="selfservice/container/select ${containerRef}; restore;"

# If there is no time input and no bookmark name, we need to use
JSTimelinePointLatestTimeInput from template.
if [[ -n $template && -z $inputTime && -z $bookmark ]]
then
pointParams="edit timelinePointParameters; set type=JSTimelinePointLatestTimeInput;
set sourceDataLayout=\"${template}\";"
# If there is a timeInput and no bookmark name, we need to use Input Time.
elif [[ -n $inputTime && -n $branch && -z $bookmark && -z $template ]]
then
pointParams="edit timelinePointParameters; set type=JSTimelinePointTimeInput; set
time=\"${inputTime}\"; set branch=\"${branch}\";"
# If there is a bookmark name and no time input, we need to use bookmark
elif [[ -n $bookmark && -z $template && -z $inputTime ]]
then
pointParams="edit timelinePointParameters; set type=JSTimelinePointBookmarkInput; set
bookmark=\"${bookmark}\";"
else
usage
exit 1
fi

paramString="$paramString $pointParams commit;"
echo "Executing the command on CLI ..."
echo $paramString
result=$(${SSH_CMD} $paramString)
check_result

echo "Verifying job status ..."
# Get everything in the result that comes after job.
```

```
temp=${result#*job}
# Get rid of everything after
resultArray=($temp)
jobRef=($resultArray)
jobString="job;select $jobRef;ls"
result=$(${SSH_CMD} $jobString)
check_result
# Get everything in the result that comes after job.
temp=${result#*jobState:}
# Get rid of everything after
resultArray=($temp)
jobState=($resultArray)

if [ $jobState = "COMPLETED" ]
then
echo "Successfully refreshed container- [$containerRef]."
else
echo "Unable to refresh container- [$containerRef]."
echo result
fi

}


##### Main

while getopts "u:d:T:b:t:B:h" flag; do
case "$flag" in
u ) username=${OPTARG%:*}
;;
d ) engine=$OPTARG
;;
T ) template=$OPTARG
;;
b ) bookmark=$OPTARG
;;
t ) inputTime=$OPTARG
;;
B ) branch=$OPTARG
;;
h ) usage
exit
;;
* ) usage
exit 1
esac

done


# Shift the parameters so we only have the positional arguments left
shift $((OPTIND-1))
```

```
# Check that there are 1 positional arguments
if [ $# != 1 ]
then
usage
exit 1
fi

# Get the one positional arguments
containerRef=$1

create_session
restore_container
```

13.2.4.7.10 CLI cookbook: how to share a Delphix self-service bookmark

13.2.4.7.10.1 Prerequisites:

- Have Delphix Self-Service user privileges.
- Know the bookmark you would like to share.

Delphix Self-Service administrators can use this CLI cookbook recipe to share a bookmark on Delphix Self-Service using the Delphix Engine CLI.

> The following script is for educational and demonstration purposes only and is not supported by Delphix.

This script can be downloaded by selecting **shareBookmark.sh**[773].

Sharing a bookmark in Delphix self-service

```
#!/bin/bash

# A sample script for calls to the CLI. This one shares Bookmark across containers in
same template.
#
# VERY IMPORTANT: In order for this to work, you need to go through the steps here:
# https://docs.delphix.com/display/DOCS43/CLI+Cookbook%3A+Configuring+Key-
Based+SSH+Authentication+for+Automation
# After this you will not need to use a username and password to log into the delphix
engine. If you do not
# setup the SSH authentication you will have to manually enter the password.
#
```

773 https://docs.delphix.com/docs/files/191896852/191896854/1/1629791457243/shareBookmark.sh

```
##examples##
# Share Bookmark
#./shareBookmark.sh -a share JS_BOOKMARK-75
# Unshare Bookmark
#./shareBookmark.sh -a unshare JS_BOOKMARK-75



##### Constants

# Describes a Delphix software revision.
VERSION="1.11.10"


##### Default Values. These can be overwriten with optional arguments.
engine="10.110.213.109"
username="admin"



##### Functions


# Help Menu
function usage {
    echo "Usage: shareBookmark.sh [[-h] | options...] -a share/unshare
<bookmarkName>"
    echo "Share/Unshare JetStream bookmark"
    echo ""
    echo "Positional arguments"
    echo "bookmarkName. Format: JS_BOOKMARK-<n>"
    echo ""
    echo "Optional Arguments:"
    echo "  -h                Show this message and exit"
    echo "  -d                Delphix engine IP address or host name, otherwise
revert to default"
    echo "  -u                Server user. Password needs to manually provide at run
time, otherwise revert to default"
    echo "  -a                action to perform on bookmark. Type:String.
Values:share/unshare"
}

# Create Our Session, including establishing the API version.
function create_session
{
    echo "creating session..."
    SSH_CMD="ssh ${username}@${engine}"
    ${SSH_CMD}  "version $VERSION"
    check_result
}
```

```
# Check the result of the curl. If there are problems, inform the user then exit.
function check_result
{
    exitStatus=$?
    if [ $exitStatus -ne 0 ]
    then
        echo "command failed with exit status $exitStatus"
        echo $result
        exit 1
    fi
}


function bookmark_action
{
    # Change share mode of bookmark
    echo "Changing share mode.."
    if [[ $action = "share" ]]
    then
        paramString="selfservice bookmark select $bookmarkName share; commit;"
    elif [[ $action = "unshare" ]]
    then
        paramString="selfservice bookmark select $bookmarkName unshare; commit;"
    else
        usage
        exit 1
    fi
    result=$(${SSH_CMD}  $paramString)
    check_result
    if [[ $action = "share" ]]
    then
        echo "Bookmark ${bookmarkName} is now in shared mode"
    elif [[ $action = "unshare" ]]
    then
        echo "Bookmark ${bookmarkName} is now in not-share mode"
    fi
}


##### Main

while getopts "u:d:a:h" flag; do
    case "$flag" in
        u )             username=${OPTARG%:*}
                        ;;
        d )             engine=$OPTARG
                        ;;
        a )             action=$OPTARG
                        ;;
        h )             usage
                        exit
                        ;;
        * )             usage
```

```
                        exit 1
    esac

done


# Shift the parameters so we only have the positional arguments left
shift $((OPTIND-1))

# Check that there is 1 positional arguments
if [ $# != 1 ]
then
    usage
    exit 1
fi

# Get the one positional arguments
bookmarkName=$1

create_session
bookmark_action
```

13.2.4.7.11 CLI cookbook: how to update a Delphix self-service bookmark

13.2.4.7.11.1 Prerequisites:

- Have Delphix Self-Service user privileges
- Know the bookmark you would like to update

13.2.4.7.11.2 Procedure:

1. Log into Delphix Engine as a Delphix Self-Serviceuser or admin.

```
ssh jsUser@<yourengine>
```

2. Navigate to the Delphix Self-Servicebookmarks, and choose the one you would like to update.

```
jsUser > selfservice bookmark
jsUser selfservice bookmark > ls
jsUser selfservice bookmark > select <jsBookmark>
```

3. Update the bookmark.

```
jsUser selfservice bookmark <jsBookmark> *> update
```

```
jsUser selfservice bookmark <jsBookmark> update *> set tags="tag text"
jsUser selfservice bookmark <jsBookmark> update *> commit
```

13.2.4.8 CLI cookbook: hooks and hook templates

These topics describe the command-line interface procedures for working with hooks and hook templates.

This section covers the following topics:

- CLI cookbook: changing the PowerShell version associated with a hook template (see page 2738)

13.2.4.8.1 CLI cookbook: changing the PowerShell version associated with a hook template

This article is to document how to change the PowerShell version associated with a hook template using the CLI.

> This operation is not allowed through GUI.

13.2.4.8.1.1 Procedure:

1. Login into Delphix Engine CLI using "admin" credentials.

```
ssh admin@<delphix_engine>
```

2. Go to **source** and then to **operationTemplate**.

```
delphix> source
delphix source> operationTemplate
```

3. Execute the **ls**command to see the existing hooks template and select one of the existing hook templates which you want to update.

```
delphix source operationTemplate> ls
Objects
NAME                DESCRIPTION                  LASTUPDATED
OPERATION.TYPE
Test_Template       Test_Template                2020-06-12T04:15:01.759Z
RunDefaultPowerShellOnSourceOperation

delphix source operationTemplate> select Test_Template
```

```
delphix source operationTemplate 'Test_Template'> ls
Properties
    type: OperationTemplate
    name: Test_Template
    description: Test_Template
    lastUpdated: 2020-06-12T04:15:01.759Z
    operation:
        type: RunDefaultPowerShellOnSourceOperation
        name: Test_Template
        command: echo_command
    reference: OPERATION_TEMPLATE-1
```

4. Run the **update** command and change the hook template type from **RunDefaultPowerShellOnSourceOperation** to **RunPowerShellOnSourceOperation**.

> Users can also change the template type from RunPowerShellOnSourceOperation to RunDefaultPowerShellOnSourceOperation.

```
delphix source operationTemplate 'Test_Template'> update
delphix source operationTemplate 'Test_Template' update *> set
operation.type=RunPowerShellOnSourceOperation
delphix source operationTemplate 'Test_Template' update *> ls
Properties
    type: OperationTemplate
    name: Test_Template
    description: Test_Template
    operation:
        type: RunPowerShellOnSourceOperation (*)
        name: Test_Template
        command: echo_command
```

5. Run the **commit** command to perform the update on the hook template. Execute the **ls** command again to see the change in the hook template type.

```
delphix source operationTemplate 'Test_Template' update *> commit

delphix source operationTemplate 'Test_Template'> ls
Properties
    type: OperationTemplate
    name: Test_Template
    description: Test_Template
    lastUpdated: 2020-06-12T04:30:44.203Z
    operation:
        type: RunPowerShellOnSourceOperation
        name: Test_Template
        command: echo_command
```

```
reference: OPERATION_TEMPLATE-1
```

1. Template type can only be changed from "RunDefaultPowerShellOnSourceOperation" to "RunPowerShellOnSourceOperation" or vice versa. You **can not** change the type to other available types. If you try to do that, you will get this error: "Error: The type of this operation (RUN_POWERSHELL_ON_SOURCE_OPERATION) cannot be modified after creation.".
2. Template type "RunPowerShellOnSourceOperation" denotes "PowerShell version 2" and "RunDefaultPowerShellOnSourceOperation" denotes "Default PowerShell version". Here, default version is the version of PowerShell installed on the target host.

13.2.4.9 CLI cookbook: network performance

These topics describe command-line interface procedures using the iperf-based Network Performance Tool:

- CLI cookbook: Delphix session protocol test from primary engine to replication engine (see page 2740)
- CLI cookbook: running the network test via CLI - latency (see page 2742)
- CLI cookbook: running the network test via CLI - throughput (see page 2744)

13.2.4.9.1 CLI cookbook: Delphix session protocol test from primary engine to replication engine

13.2.4.9.1.1 Prerequisites

The network performance tool measures network performance between a Delphix Engine and an environment host. You must have added an environment in order to use this tool.

This transmission control protocol (TCP) throughput test uses TCP port 50001 by default. The port can also be configured on a per-test-run basis. For the duration of a given throughput test, a server on the receiver will be listening on this port. For a transmit test, the receiver is the remote host; for a receive test, the receiver is the Delphix Engine

13.2.4.9.1.2 Procedure

Delphix uses the Delphix Session Protocol (DSP) protocol to communicate between primary and replication engines.

1. Login to Delphix Engine using a Delphix administrator account such as **delphix_admin.** As soon as you login, you will get prompt with engine name.

```
-bash-4.3$ ssh delphix_admin@delphix
Password:
```

```
delphix>
```

2. Create a DSP test.

```
delphix> network test dsp create
```

3. Set the destinationType to DELPHIX_ENGINE to do a DSP test between two Delphix Engines. The default is REMOTE_HOST which executes a DSP test between a source or target host and the Delphix Engine.

```
delphix network test dsp create *> set destinationType=DELPHIX_ENGINE
```

4. Use **get** to see other optional arguments. Modify the test parameters as needed and commit to start the test.

```
delphix network test dsp create *> get
   type: NetworkDSPTestParameters
   blockSize: 64KB
   compression: false
   destinationType: REMOTE_HOST
   direction: TRANSMIT
   duration: 30
   encryption: false
   numConnections: 0
   queueDepth: 32
   receiveSocketBuffer: 256KB
   remoteDelphixEngineInfo: (unset)
   remoteHost: (unset)
   sendSocketBuffer: 256KB
delphix network test dsp create *> set remoteDelphixEngineInfo.address=delphix2
delphix network test dsp create *> set
remoteDelphixEngineInfo.principal=delphix_admin
delphix network test dsp create *> edit remoteDelphixEngineInfo
delphix network test dsp create remoteDelphixEngineInfo *> get
   type: RemoteDelphixEngineInfo (*)
   address: delphix2 (*)
   credential: (required)
   principal: delphix_admin (*)
delphix network test dsp create remoteDelphixEngineInfo *> set
credential.password=delphix
delphix network test dsp create remoteDelphixEngineInfo *> commit
   `NETWORK_DSP_TEST-2
   Dispatched job JOB-8
   NETWORK_DSP_TEST_EXECUTE job started.
   Measuring throughput with variable number of connections: 716 Mbps measured
for 1 connections.
```

```
   Measuring throughput with variable number of connections: 711 Mbps measured
for 2 connections.
   Measuring throughput with variable number of connections: 611 Mbps measured
for 4 connections.
   Measuring throughput with variable number of connections: 646 Mbps measured
for 6 connections.
   Measuring throughput with variable number of connections: 567 Mbps measured
for 8 connections.
   Measuring average throughput for 30 seconds with 1 connections.
   Measured throughput of 408 Mbps.
   NETWORK_DSP_TEST_EXECUTE job completed successfully.
```

5. Retrieve the test results.

```
delphix> network test dsp list
NAME                              PARAMETERS.DIRECTION  STATE      THROUGHPUT
delphix2-2018-01-23T21:25:31.172Z  TRANSMIT              COMPLETED  880.5Mbps
delphix2-2018-02-02T17:54:58.322Z  TRANSMIT              COMPLETED  408.5Mbps
```

13.2.4.9.2 CLI cookbook: running the network test via CLI - latency

13.2.4.9.2.1 Prerequisites

The network performance tool measures network performance between a Delphix Engine and an environment host. You must have added an environment in order to use this tool.

This transmission control protocol (TCP) throughput test uses TCP port 50001 by default. The port can also be configured on a per-test-run basis. For the duration of a given throughput test, a server on the receiver will be listening on this port. For a transmit test, the receiver is the remote host; for a receive test, the receiver is the Delphix Engine.

13.2.4.9.2.2 Procedure

The network latency test measures network round-trip latency by transmitting ICMP echo requests (like the ping utility) and measuring the time to receive replies from the remote host. To execute a test:

1. Login as a domain user to the Delphix Engine CLI using ssh.
2. Create a network latency test.

```
delphix> network test latency
delphix network test latency> create
delphix network test latency create *>
```

3. You must set **remoteHost** to the name of an environment host already configured in the Delphix Engine. (You should press tab after the "=" (equal sign) to auto-populate and confirm registered

destinations). Use 'get' to see other optional arguments. Modify the test parameters as needed and **commit** to start the test.

```
delphix network test latency create *> set remoteHost=oracletarget
delphix network test latency create *> get
    type: NetworkLatencyTestExecuteParameters
    remoteHost: oracletarget
    requestCount: 20
    requestSize: 8B
delphix network test latency create *> commit
    Dispatched job JOB-20
    NETWORK_LATENCY_TEST_EXECUTE job started for
"oracletarget-2014-06-20T18:57:28.659Z".
    Executing network latency test.
    NETWORK_LATENCY_TEST_EXECUTE job for
"oracletarget-2014-06-20T18:57:28.659Z" completed successfully.
```

4. The job will be submitted and visible in the Delphix Management application.
5. Retrieve the test results. All times are in microseconds.

```
delphix network test latency> list
NAME                                    AVERAGE
oraclesource-2014-06-20T18:57:28.659Z  872
delphix network test latency> select oraclesource-2014-06-20T18:57:28.659Z
delphix network test latency "oraclesource-2014-06-20T18:57:28.659Z"> get
    type: NetworkLatencyTest
    name: oraclesource-2014-06-20T18:57:28.659Z
    average: 872
    endTime: 2014-06-20T18:57:48.558Z
    loss: 0
    maximum: 2755
    minimum: 294
    reference: NETWORK_LATENCY_TEST-2
    remoteAddress: 172.16.203.184
    remoteHost: oraclesource
    requestCount: 20
    requestSize: 8B
    startTime: 2014-06-20T18:57:28.659Z
    stddev: 527
```

13.2.4.9.3 CLI cookbook: running the network test via CLI - throughput

13.2.4.9.3.1 Prerequisites

The network performance tool measures network performance between a Delphix Engine and an environment host. You must have added an environment in order to use this tool.

This transmission control protocol (TCP) throughput test uses TCP port 50001 by default. The port can also be configured on a per-test-run basis. For the duration of a given throughput test, a server on the receiver will be listening on this port. For a transmit test, the receiver is the remote host; for a receive test, the receiver is the Delphix Engine.

13.2.4.9.3.2 Procedure

The network throughput test measures sustained throughput using a synthetic workload to or from a remote host. To execute a test:

1. Login as a domain user to the **Delphix Engine CLI** using ssh.
2. Create a network throughput test.

```
delphix> network test throughput
delphix network test throughput> create
delphix network test throughput create *>
```

3. You must set **remoteHost** to the name of an environment host already configured in the Delphix Engine. Use 'get' to see other optional arguments. Modify the test parameters as needed and **commit** to start the test.

```
delphix network test throughput create *> set remoteHost=oraclesource
delphix network test throughput create *> ls
Properties
    type: NetworkThroughputTestParameters
    blockSize: 128KB
    direction: TRANSMIT
    duration: 30
    numConnections: 0
    port: 50001
    receiveSocketBuffer: 4MB
    remoteHost: oraclesource
    sendSocketBuffer: 4MB
delphix network test throughput create *> commit
    Dispatched job JOB-21
    NETWORK_THROUGHPUT_TEST_EXECUTE job started for
"oraclesource-2014-06-20T19:30:12.566Z".
    Executing network throughput transmit test.
    Measuring throughput with variable number of connections: 1.
```

```
    Measuring throughput with variable number of connections: 2.
    Measuring throughput with variable number of connections: 4.
    Measuring throughput with variable number of connections: 6.
    Measuring throughput with variable number of connections: 8.
    Measuring maximum sustained throughput for 30 seconds with 8 connections.
    NETWORK_THROUGHPUT_TEST_EXECUTE job for
"oraclesource-2014-06-20T19:30:12.566Z" completed successfully.
```

4. The job will be submitted and visible in the Delphix Management application.

```
delphix network test throughput> list
NAME                                    DIRECTION    STATE     THROUGHPUT
oraclesource-2014-06-20T19:30:12.566Z  TRANSMIT     COMPLETED 695.6Mbps
delphix network test throughput> select oraclesource-2014-06-20T19:30:12.566Z
delphix network test throughput "oraclesource-2014-06-20T19:30:12.566Z"> get
    type: NetworkThroughputTest
    name: oraclesource-2014-06-20T19:30:12.566Z
    endTime: 2014-06-20T19:31:15.041Z
    numConnections: 8
    parameters:
        type: NetworkThroughputTestParameters
        blockSize: 128KB
        direction: TRANSMIT
        duration: 30
        numConnections: 0
        port: 50001
        receiveSocketBuffer: 4MB
        remoteHost: oraclesource
        sendSocketBuffer: 4MB
    reference: NETWORK_THROUGHPUT_TEST-2
    remoteAddress: 172.16.203.184
    startTime: 2014-06-20T19:30:12.566Z
    state: COMPLETED
    throughput: 695.6Mbps
```

13.2.4.10 Kerberos CLIs

The following topics provide information related to the Kerberos CLIs.

- CLI cookbook: admin application configuration (see page 2745)
- CLI cookbook system app configuration (see page 2750)

13.2.4.10.1 CLI cookbook: admin application configuration

The following are done after logging into the Delphix admin app as `admin@<delphix engine hostname>.`

13.2.4.10.1.1 Add Kerberos environment

```
jkb-5160.dcenter> cd /environment/
jkb-5160.dcenter environment> ls
Children
oracle
user
windows

Operations
create
jkb-5160.dcenter environment> create
jkb-5160.dcenter environment create *> ls
Properties
   type: HostEnvironmentCreateParameters
   hostEnvironment:
       type: UnixHostEnvironment
       name: (unset)
       aseHostEnvironmentParameters: (unset)
       description: (unset)
   hostParameters:
       type: UnixHostCreateParameters
       host:
           type: UnixHost
           address: (required)
           privilegeElevationProfile: (unset)
           sshPort: 22
           toolkitPath: (required)
           truststorePassword: (unset)
   primaryUser:
       type: EnvironmentUser
       name: (unset)
       credential:
           type: PasswordCredential
           password: (required)
       environment: (unset)
       groupId: (unset)
       userId: (unset)
jkb-5160.dcenter environment create *> set hostParameters.host.address=ln-rh64-
tgt.dc2.delphix.com
jkb-5160.dcenter environment create *> set hostParameters.host.toolkitPath=/tmp
jkb-5160.dcenter environment create *> set
primaryUser.credential.type=KerberosCredential
jkb-5160.dcenter environment create *> commit
   `UNIX_HOST_ENVIRONMENT-2
   Dispatched job JOB-4
   ENVIRONMENT_CREATE_AND_DISCOVER job started for "ln-rh64-tgt.dc2.delphix.com".
   ENVIRONMENT_CREATE_AND_DISCOVER job for "ln-rh64-tgt.dc2.delphix.com" completed
successfully.
```

```
jkb-5160.dcenter environment>
jkb-5160-2.dcenter environment> ls
Objects
NAME                          DESCRIPTION
ln-rh64-tgt.dc2.delphix.com   -

Children
oracle
user
windows

Operations
Create
jkb-5160-2.dcenter environment> select ln-rh64-tgt.dc2.delphix.com
jkb-5160-2.dcenter environment 'ln-rh64-tgt.dc2.delphix.com'> ls
Properties
   type: UnixHostEnvironment
   name: ln-rh64-tgt.dc2.delphix.com
   aseHostEnvironmentParameters: (unset)
   description: (unset)
   enabled: true
   host: ln-rh64-tgt.dc2.delphix.com
   primaryUser: sybase
   reference: UNIX_HOST_ENVIRONMENT-1

Operations
delete
update
disable
enable
refresh
jkb-5160-2.dcenter environment 'ln-rh64-tgt.dc2.delphix.com'>
```

13.2.4.10.1.2 Add Kerberos ASE instance

```
jkb-5160-2.dcenter environment> cd /repository/
jkb-5160-2.dcenter repository> ls
Children
template

Operations
create
compatibleRepositories
jkb-5160-2.dcenter repository> create
jkb-5160-2.dcenter repository create *> ls
Properties
   type: ASEInstance
   credentials: (unset)
   dbUser: (unset)
```

```
   environment: (required)
   installationPath: (required)
   instanceName: (required)
   instanceOwner: (required)
   ports: (required)
   servicePrincipalName: (unset)
   version: (unset)
jkb-5160-2.dcenter repository create *> set credentials.type=KerberosCredential
jkb-5160-2.dcenter repository create *> set environment=ln-rh64-tgt.dc2.delphix.com
jkb-5160-2.dcenter repository create *> set installationPath=/opt/sybase/15-0
jkb-5160-2.dcenter repository create *> set ports=5100
jkb-5160-2.dcenter repository create *> set instanceName=ASE1570_S1
jkb-5160-2.dcenter repository create *> set instanceOwner=sybase
jkb-5160-2.dcenter repository create *> set servicePrincipalName=ASE1570_S1
jkb-5160-2.dcenter repository create *> commit
   `ASE_INSTANCE-1
jkb-5160-2.dcenter repository> ls
Objects
NAME        VERSION ENVIRONMENT
ASE1570_S1  15.7 SP138 ln-rh64-tgt.dc2.delphix.com

Children
template

Operations
create
compatibleRepositories
jkb-5160-2.dcenter repository> select ASE1570_S1
jkb-5160-2.dcenter repository 'ASE1570_S1'> ls
Properties
   type: ASEInstance
   name: ASE1570_S1
   credentials:
       type: KerberosCredential
   dbUser: sybase
   discovered: false
   environment: ln-rh64-tgt.dc2.delphix.com
   installationPath: /opt/sybase/15-0
   instanceName: ASE1570_S1
   instanceOwner: sybase
   instanceOwnerGid: 500
   instanceOwnerUid: 500
   internalVersion: 15.7 SP138
   linkingEnabled: true
   pageSize: 4096
   ports: 5100
   provisioningEnabled: true
   reference: ASE_INSTANCE-1
   servicePrincipalName: ASE1570_S1
   staging: false
   version: 15.7 SP138
```

```
Operations
delete
update
jkb-5160-2.dcenter repository 'ASE1570_S1'>
```

13.2.4.10.1.3 Link a dSource

```
jkb-5160-2.dcenter> cd /database
jkb-5160-2.dcenter database> ls
Children
template

Operations
createEmpty
createRestorationDataset
export
fileMapping
link
oracleSupportedCharacterSets
provision
validateXpp
xpp
jkb-5160-2.dcenter database> link
jkb-5160-2.dcenter database link *> ls
Properties
   type: LinkParameters
   name: (required)
   description: (unset)
   group: (required)
   linkData:
       type: ASELinkData
       config: (required)
       dbCredentials:
           type: PasswordCredential
           password: (required)
       dbUser: (unset)
       dumpCredentials: (unset)
       externalFilePath: (unset)
       loadBackupPath: (required)
       loadLocation: (unset)
       operations: (unset)
       sourceHostUser: (required)
       sourcingPolicy: (unset)
       stagingHostUser: (required)
       stagingPostScript: (unset)
       stagingPreScript: (unset)
       stagingRepository: (required)
       syncParameters:
           type: ASELatestBackupSyncParameters
```

```
        validatedSyncMode: ENABLED

Operations
defaults
jkb-5160-2.dcenter database link *> set name=test1
jkb-5160-2.dcenter database link *> set group=Untitled
jkb-5160-2.dcenter database link *> set linkData.config=test1
jkb-5160-2.dcenter database link *> set
linkData.dbCredentials.type=KerberosCredential
jkb-5160-2.dcenter database link *> set linkData.loadBackupPath=/home/sybase/db
jkb-5160-2.dcenter database link *> set linkData.stagingRepository=ASE1570_S1
jkb-5160-2.dcenter database link *> set linkData.sourceHostUser=sybase
jkb-5160-2.dcenter database link *> set linkData.stagingHostUser=sybase
jkb-5160-2.dcenter database link *> set
linkData.syncParameters.type=ASENewBackupSyncParameters
jkb-5160-3.dcenter database link linkData syncParameters *> commit
   `ASE_DB_CONTAINER-1
   Dispatched job JOB-4
   DB_LINK job started for "Untitled/test1".
   DB_LINK job for "Untitled/test1" completed successfully.
jkb-5160-2.dcenter database>
jkb-5160-2.dcenter database> ls
Objects
NAME   PROVISIONCONTAINER  DESCRIPTION
test1  -             -

Children
template

Operations
createEmpty
createRestorationDataset
export
fileMapping
link
oracleSupportedCharacterSets
provision
validateXpp
xpp
jkb-5160-2.dcenter database>
```

13.2.4.10.2 CLI cookbook system app configuration

The following are done after logging into the Delphix System configuration application as `sysadmin@<delphix_engine_hostname;.`

13.2.4.10.2.1 Define the system Kerberos configuration

```
jkb-5160.dcenter> cd /service/kerberos/
jkb-5160.dcenter service kerberos> ls
Properties
   type: KerberosConfig
   enabled: false
   kdcs:
       0:
           type: KerberosKDC
           hostname:
           port: 88
   keytab: (unset)
   principal:
   realm:

Operations
update
reset
jkb-5160.dcenter service kerberos> update
jkb-5160.dcenter service kerberos update *> set realm=DELPHIX.COM
jkb-5160.dcenter service kerberos update *> set principal=sybase
jkb-5160.dcenter service kerberos update *> set
keytab=BQIAAABEAAEAC0RFTFBISVguQ09NAAZzeWJhc2UAAAABWNVqQgMAEgAgjF/
zgNuw27Uy9vEgvPvpeevAaAw6c5HaVAWVSLZnmngAAAA0AAEAC0RFTFBISVguQ09NAAZzeWJhc2UAAAABWNVq
QgMAEQAQTsNZOXmHhaeuXPeyQRXelAAAADwAAQALREVMUEhJWC5DT00ABnN5YmFzZQAAAAFY1WpCAwAQABjHS
VHfC+p8yIYfgFTIv1T7m99n2Spn7wIAAAA0AAEAC0RFTFBISVguQ09NAAZzeWJhc2UAAAABWNVqQgMAFwAQhJ
PmRokSfz310fe3eVUD3wAAACwAAQALREVMUEhJWC5DT00ABnN5YmFzZQAAAAFY1WpCAwAIAAhiFUmzFjttmwA
AACwAAQALREVMUEhJWC5DT00ABnN5YmFzZQAAAAFY1WpCAwADAAj4rgff+Au/ng==
jkb-5160.dcenter service kerberos update *> edit kdcs
jkb-5160.dcenter service kerberos update kdcs *> edit 0
jkb-5160.dcenter service kerberos update kdcs 0 *> set hostname=kerberos-01.delphix.c
om
jkb-5160.dcenter service kerberos update kdcs 0 *> back
jkb-5160.dcenter service kerberos update kdcs *> back
jkb-5160.dcenter service kerberos update *> ls
Properties
   type: KerberosConfig
   kdcs:
       0:
           type: KerberosKDC (*)
           hostname: kerberos-01.delphix.com (*)
           port: 88 (*)
   keytab: ******** (*)
   principal: sybase (*)
   realm: DELPHIX.COM (*)
jkb-5160.dcenter service kerberos update *> commit
jkb-5160.dcenter service kerberos> ls
Properties
```

```
    type: KerberosConfig
    enabled: true
    kdcs:
        0:
            type: KerberosKDC
            hostname: kerberos-01.delphix.com
            port: 88
    keytab: (unset)
    principal: sybase
    realm: DELPHIX.COM

Operations
update
reset
jkb-5160.dcenter service kerberos>
```

13.2.4.10.2.2 Clear Kerberos configuration

```
jkb-5160.dcenter service kerberos> reset
jkb-5160.dcenter service kerberos reset *> commit
jkb-5160.dcenter service kerberos> ls
Properties
    type: KerberosConfig
    enabled: false
    kdcs:
        0:
            type: KerberosKDC
            hostname:
            port: 88
    keytab: (unset)
    principal:
    realm:

Operations
update
reset
jkb-5160.dcenter service kerberos>
```

13.3 Web services API guide

These topics describe interfacing with the public web service APIs, building automation facilities and integrating with third-party orchestration tools.

This section covers the following topics:

- API version information (see page 2753)
- Web service object model (see page 2757)
- Web service protocol (see page 2759)

- CLI to web services translation (see page 2761)
- GUI API mapping (see page 2764)
- CLI to Python transition (see page 2769)
- Python Cookbook: adding a UNIX host (see page 2772)
- So you want to work with Delphix APIs? (see page 2774)
- API Cookbook: common tasks, workflows, and examples (see page 2831)
- Kerberos APIs

13.3.1 API version information

This topic describes API version information for each release of the Delphix Engine, including schema changes and links to the relevant version of the schema.

Delphix engine version	API version	Link to schema within the appliance	
5.0.0.x - 5.0.3.x	1.7.0	http://<engine-address>/api/json/versions/1.7.0/delphix.json	
5.0.4.x	1.7.1	http://<engine-address>/api/json/versions/1.7.1/delphix.json	
5.1.0.x - 5.1.2.x	1.8.0	http://<engine-address>/api/json/versions/1.8.0/delphix.json	
5.1.3.x - 5.1.5.x	1.8.1	http://<engine-address>/api/json/versions/1.8.1/delphix.json	
5.1.6.0	1.8.2	http://<engine-address>/api/json/versions/1.8.2/delphix.json	
5.2.0.0 - 5.2.2.1	1.9.0	http://<engine-address>/api/json/versions/1.9.0/delphix.json	
5.2.3.0 - 5.2.3.1	1.9.1	http://<engine-address>/api/json/versions/1.9.1/delphix.json	
5.2.4.0	1.9.2	http://<engine-address>/api/json/versions/1.9.2/delphix.json	
5.2.5.0 - 5.2.6.2	1.9.3	http://<engine-address>/api/json/versions/1.9.3/delphix.json	

Delphix engine version	API version	Link to schema within the appliance	
5.3.0.0 - 5.3.0.3	1.10.0	http://<engine-address>/api/json/versions/1.10.0/delphix.json	
5.3.1.0	1.10.1	http://<engine-address>/api/json/versions/1.10.1/delphix.json	
5.3.2.0	1.10.2	http://<engine-address>/api/json/versions/1.10.2/delphix.json	
5.3.3.0 - 5.3.3.1	1.10.3	http://<engine-address>/api/json/versions/1.10.3/delphix.json	
5.3.4.0	1.10.4	http://<engine-address>/api/json/versions/1.10.4/delphix.json	
5.3.5.0	1.10.5	http://<engine-address>/api/json/versions/1.10.5/delphix.json	
5.3.6.0-5.3.9.0	1.10.6	http://<engine-address>/api/json/versions/1.10.6/delphix.json	
6.0.0.0	1.11.0	http://<engine-address>/api/json/versions/1.11.0/delphix.json	
6.0.1.0	1.11.1	http://<engine-address>/api/json/versions/1.11.1/delphix.json	
6.0.2.0	1.11.2	http://<engine-address>/api/json/versions/1.11.2/delphix.json	
6.0.3.0	1.11.3	http://<engine-address>/api/json/versions/1.11.3/delphix.json	
6.0.4.0	1.11.4	http://<engine-address>/api/json/versions/1.11.4/delphix.json	
6.0.5.0	1.11.5	http://<engine-address>/api/json/versions/1.11.5/delphix.json	

Delphix engine version	API version	Link to schema within the appliance	
6.0.6.0	1.11.6	http://<engine-address>/api/json/versions/1.11.6/delphix.json	
6.0.7.0	1.11.7	http://<engine-address>/api/json/versions/1.11.7/delphix.json	
6.0.8.0	1.11.8	http://<engine-address>/api/json/versions/1.11.8/delphix.json	
6.0.9.0	1.11.9	http://<engine-address>/api/json/versions/1.11.9/delphix.json	
6.0.10.0	1.11.10	http://<engine-address>/api/json/versions/1.11.10/delphix.json	
6.0.11.0	1.11.11	http://<engine-address>/api/json/versions/1.11.11/delphix.json	
6.0.12.0	1.11.12	http://<engine-address>/api/json/versions/1.11.12/delphix.json	
6.0.13.0	1.11.13	http://<engine-address>/api/json/versions/1.11.13/delphix.json	
6.0.14.0	1.11.14	http://<engine-address>/api/json/versions/1.11.14/delphix.json	
6.0.15.0	1.11.15	http://<engine-address>/api/json/versions/1.11.15/delphix.json	
6.0.16.0	1.11.16	http://<engine-address>/api/json/versions/1.11.16/delphix.json	
6.0.17.0	1.11.17	http://<engine-address>/api/json/versions/1.11.17/delphix.json	
7.0.0.0	1.11.18	http://<engine-address>/api/json/versions/1.11.18/delphix.json	

Delphix engine version	API version	Link to schema within the appliance	
8.0.0.0	1.11.19	http://<engine-address>/api/json/versions/1.11.19/delphix.json	
9.0.0.0	1.11.20	http://<engine-address>/api/json/versions/1.11.20/delphix.json	
10.0.0.0	1.11.21	http://<engine-address>/api/json/versions/1.11.21/delphix.json	
11.0.0.0	1.11.22	http://<engine-address>/api/json/versions/1.11.22/delphix.json	
12.0.0.0	1.11.23	http://<engine-address>/api/json/versions/1.11.23/delphix.json	
13.0.0.0	1.11.24	http://<engine-address>/api/json/versions/1.11.24/delphix.json	
14.0.0.0	1.11.25	http://<engine-address>/api/json/versions/1.11.25/delphix.json	
15.0.0.0	1.11.26	http://<engine-address>/api/json/versions/1.11.26/delphix.json	
16.0.0.0	1.11.27	http://<engine-address>/api/json/versions/1.11.27/delphix.json	
17.0.0.0	1.11.28	http://<engine-address>/api/json/versions/1.11.28/delphix.json	
18.0.0.0	1.11.29	http://<engine-address>/api/json/versions/1.11.29/delphix.json	
19.0.0.0	1.11.30	http://<engine-address>/api/json/versions/1.11.30/delphix.json	
20.0.0.0	1.11.31	http://<engine-address>/api/json/versions/1.11.31/delphix.json	

Delphix engine version	API version	Link to schema within the appliance	
21.0.0.0	1.11.32	http://<engine-address>/api/json/versions/1.11.32/delphix.json	
22.0.0.0	1.11.33	http://<engine-address>/api/json/versions/1.11.33/delphix.json	

For more details see API Changes (see page 367)

13.3.2 Web service object model

This topic describes the Delphix object model as exported over the web services.

13.3.2.1 Object types

All objects in the Delphix API are "typed objects." All such objects have a `type` field that indicates the type of the object and its associated semantics. This allows for object inheritance and polymorphism without requiring separate APIs for each type and allows generic client-specific semantic encoding and decoding without having to be aware of the context. This means that even APIs that operate only a specific type (such as the `Group` API) still require a type field to be specified as part of the input, and will continue to report the type of objects when listing or retrieving objects. This allows the APIs to evolve in a backward-compatible fashion through the introduction of new types.

Certain "root" object types (groups, containers, sources, etc) have an associated API. This API is rooted at a particular point under `/resources/json/delphix`, but all APIs follow a standard format beneath this namespace. The CLI namespace is a direct reflection of this URL, and the API reference has an index both by object type as well as by object (CLI) path. These APIs may operate on different extended types (such as `OracleSIConfig` and `OracleRACConfig`), but the base operations remain the same regardless of input type.

13.3.2.2 Object references

Some objects returned by the APIs are pure typed objects, in that they don't represent the persistent state on the Delphix Engine but are rather calculated and returned upon request. Most of the objects in the system, however, are "persistent objects." Persistent objects (of type `PersistentObject`) have a stable reference that uniquely identifies the object on the system. This reference is separate from its name so that objects can be renamed without affecting the programmatic API. More information about object names and how they can be represented generically can be found in the CLI documentation (see page 2521). Object references are opaque tokens; while they can be stored and reused for later use, and interpretation of their contents is unstable and may break in a future release.

The Delphix object hierarchy is stitched together by these object references. When fetching an object that refers to another object, the member will be returned as a reference, rather than being inserted directly within

the parent object. This allows consumers to control when and how links are resolved and make it clear when an object may change independently from its parent. The per-object APIs outlined below all operate on object references.

Note that some Delphix objects are singleton[774] objects, and there is only one such object on the system. These objects do not have references because the API URL uniquely identifies the object on the system.

13.3.2.3 API operations

All APIs optionally support the following operations:

- `CREATE` - Create a new instance of the given object type. This is used for most objects, but more complicated objects, such as dSources and VDBs, must be created through a dedicated `link` or `provision` operation. The input to this operation is typically the object itself, though some objects may have specialized parameters used during the creation of objects. An example of this is `HostEnvironmentCreateParameters`.
- `UPDATE` - Update properties of the given object, specified as an object reference in the URL.
- `DELETE` - Delete a particular object, specified as an object reference in the URL. These operations are typically done as HTTP `DELETE` operations, but it is also possible to do a `POST` operation to the `/delete` API to do the same thing. The `POST` form allows for delete-specific parameters, such as `OracleDeleteParameters`.
- `GET` - Get the properties for a particular object, specified as an object reference in the URL.
- `LIST` - List all objects of the given type within the system. These APIs typically take optional query parameters that allow the set of results to be constrained, filtered, paginated, or sorted.

In addition, the following non-CRUD operations may be supported:

- **Root Operation** - A `POST or GET` operation to the root of an API namespace, not associated with a particular object. This can be used for singleton objects, such as `NDMPConfig`, operations that create objects, such as `link`, and operations that operate on multiple objects at once.
- **Per-object Operation** - A `POST` operation to a particular object reference. These operations are typically read-write but are not required to be so. These would include read-only operations that cannot be modeled as CRUD operations or require complex input use per-object operations.

13.3.2.4 Database object mModel

In order to support a wide variety of databases and database configurations, the database object model is more complex than it may initially appear after having used the Delphix Management application. For example, there is no such thing as a "dSource" or "VDB" object, only data "containers" with attached "sources". More information about how Database objects are modeled within Delphix can be found in the CLI documentation (see page 2521)

774 http://en.wikipedia.org/wiki/Singleton_pattern

13.3.2.5 Asynchronous execution

All APIs are designed to be transactionally safe and "quick." However, there are operations that may take a long period of time or may need to reach out to external hosts or databases such that they cannot be done safely within the context of a single API call. Such operations will dispatch a `Job` to handle asynchronous execution of the operation. **Any API** can potentially spawn a job, and which APIs spawn jobs and which do not may differ between object types or releases. If you are developing a full-featured automation system, it is recommended that you build a generic infrastructure to handle job monitoring, rather than encoding the behavior of particular APIs that may change over time.

Every operation, except for `LIST` and `GET`, which are guaranteed to be read-only, can potentially spawn a job. This is represented by the `job` field of the `APIResult` object. If this field is `null`, then the action can be completed within the bounds of the API call. Otherwise, a reference to a dispatched job is returned.

Jobs can spawn other jobs for especially complex operations, such as provisioning to an Oracle cluster environment. The job returned in the API invocation is the root job, and overall success or failure of the operation is determined by the state of this job. Some operations may succeed even if a child job fails. An example would be provisioning to an Oracle cluster where one node failed, but others were successful.

Progress can be monitored by examining the `JobEvent` objects in the `Job` object returned through the job API.

13.3.3 Web service protocol

This topic describes an overview of the web service API and available facilities.

13.3.3.1 Introduction

The Delphix Engine provides a set of public stable web service APIs (application programming interfaces). The web services form the basis upon with the GUI and CLI are built, and are designed to be **public** and **stable**. This guide covers the basic operation of the protocol, concepts, and considerations when building layered infrastructure. It is not a reference for all available APIs. For more information on available APIs, go to the '/api' URL of a Delphix appliance, which will provide a complete reference of all available APIs for the version of Delphix running on that system.

```
http://<server>/api
```

The CLI is a thin veneer over the web services. If you are new to the web services, it is recommended you first test out operations with the CLI, and use the `setopt trace=true` option to dump the raw data being sent and received to see the API in action.

13.3.3.2 Protocol operation

The Delphix web services are a RESTful[775] API with loose CRUD[776] semantics using JSON[777] encoding. The following HTTP methods are supported:

- `GET` - Retrieve data from the server where complex input is not needed. All `GET` requests are guaranteed to be read-only, but not all read-only requests are required to use `GET`. Simple input (strings,number, boolean values) can be passed as query parameters.
- `POST` - Issue a read/write operation, or make a read-only call that requires complex input. The optional body of the call is expressed as JSON.
- `DELETE` - Delete an object on the system. For languages that don't provide a native wrapper for `DELETE`, or for delete operations with optional input, all delete operations can also be invoked as `POST` to the same URL with `/delete` appended to it.

Regardless of the operation, the result is returned as a JSON encoded result, the contents of which are covered below. For example, the following invocation of curl[778] demonstrates establishing a new Session (pretty-printing the result):

```
$ curl -s -X POST -k --data @- http://delphix-server/resources/json/delphix/session \
-c ~/cookies.txt -H "Content-Type: application/json" <<EOF{   "type": "APISession",
"version": {       "type": "APIVersion",       "major": 1,       "minor": 4,
"micro": 3   }}EOF{   "status":"OK",   "result": {       "type":"APISession",
"version": {           "type": "APIVersion",           "major": 1,           "minor":
4,           "micro": 3       },       "locale": "en_US",       "client": null   },
"job": null}EOF
```

NOTE: It is generally recommended to set the API session version to the highest level supported (see page 2753) by your Delphix Engine.

13.3.3.3 Session establishment

Login involves establishing a session and then authenticating to the Delphix Engine. Only authenticated users can access web APIs. Each user must establish a session prior to making any other API calls. This is done by sending a `Session` object to the URL `/resources/json/delphix/session`. This session object will specify the `APIVersion` to use for communication between the client and server. If the server doesn't support the version requested due to an incompatible change reflected in the API major version number, an error will be returned.

The resulting session object encodes the native server version, which can be different than the version requested by the client. If the server is running a more recent but compatible version, any translation of input and output to the native version is handled automatically. More information on versioning can be found in the documentation for the `APIVersion` object within the API reference on a Delphix system. If the client

775 http://en.wikipedia.org/wiki/Representational_state_transfer
776 http://en.wikipedia.org/wiki/Create,_read,_update_and_delete
777 http://www.json.org/
778 http://curl.haxx.se/

supports multiple versions, the appropriate type can be negotiated by trying to establish a session with each major version supported and then inspecting the version returned.

The session will also return an identifier through browser cookies that can be reused in subsequent calls to use the same session credentials and state without having to re-authenticate. The format of this cookie is private to the server and may change at any point. Sessions do not persist across a server restart or reboot. The mechanism by which this cookie is preserved and sent with subsequent requests is client-specific. The following demonstrates invoking the session login API call using curl[779] and storing the cookies in the file `~/cookies.txt` for later use:

```
$ curl -s -X POST -k --data @- http://delphix-server/resources/json/delphix/session \
-c ~/cookies.txt -H "Content-Type: application/json" <<EOF{   "type": "APISession",
"version": {       "type": "APIVersion",       "major": 1,       "minor": 4,
"micro": 3   }}EOF{   "status":"OK",   "result": {       "type":"APISession",
"version": {           "type": "APIVersion",           "major": 1,           "minor":
4,           "micro": 3       },       "locale": "en_US",       "client": null   },
"job": null}EOF
```

13.3.3.4 Authentication

Once the session has been established, the next step is to authenticate to the server by executing the `LoginRequest` API. Unauthenticated sessions are prohibited from making any API calls other than this login request. The username can be either a system user or domain user, and the backend will authenticate using the appropriate method. This example illustrates logging in via curl using cookies created when the session was established:

```
$ curl -s -X POST -k --data @- http://delphix-server/resources/json/delphix/login \-b
cookies.txt -c cookies2.txt -H "Content-Type: application/json" <<EOF{"type":
"LoginRequest","username": "delphix_user","password": "delphix_pass","target":
"DOMAIN"}EOF
```

The new cookie (cookie2.txt) will need to be used in subsequent API requests. The login API currently only supports authentication by a password. There is no way to authenticate using any shared key or alternate authentication strategy.

13.3.4 CLI to web services translation

This topic describes using the CLI to understand public web service APIs.

The command-line interface (see page 2521) is a direct translation of the web services API to an interactive environment. This allows you to use the CLI to explore functionality with tab completion, integrated help, stronger type checking, and an indication of expected types and required fields. When trying to determine how to invoke an operation through the web services or interpret the results, it is recommended that you first learn how to do the same through the CLI, and then use the provided tools to translate that into web services call.

779 http://curl.haxx.se/

13.3.4.1 CLI translation to HTTP

The CLI namespace is identical to the web service URLs for each base object and operation type. The root of all web services is `/resources/json/delphix`. Any additional CLI context is appended to this URL, joined by slashes. For example:

```
delphix> database provision
```

Is equivalent to:

```
POST /resources/json/delphix/database/provision
```

All operations in the CLI (those that require an explicit `commit` command) are modeled as `POST` HTTP calls. This is an example of a "root operation", as they are invoked not on any particular object, but across the system as a whole. Operations that are invoked on a particular object use a URL specific to that object:

```
delphix> database "dexample" refresh
```

Is equivalent to:

```
POST /resources/json/delphix/database/ORACLE_DB_CONTAINER-3/refresh
```

While the CLI uses names to refer to objects, at the API level we use references. Persistent objects (those with a permanent unique identity) have a `reference` field that is used in all cases to refer to the object. This allows references to remain constant even if objects are renamed.

For the standard CRUD (create, read, update, delete) operations, the mapping is slightly different:

CLI operation	HTTP API
`database list`	`GET /resources/json/delphix/database`
`database create`	`POST /resources/json/delphix/database`
`database "dexample" get`	`GET /resources/json/delphix/database/<>`
`database "dexample" update`	`POST /resources/json/delphix/database/<>`

CLI operation	HTTP API
`database "dexample" delete`	`DELETE /resources/json/delphix/database/<>` `POST /resources/json/delphix/database/>>`

The `DELETE` operation has an optional `POST` form that can take complex input for clients that don't support sending a payload as part of a `DELETE` operation.

13.3.4.2 Tracing HTTP calls

The CLI also provides facilities to see the raw HTTP calls being made as part of any operation. To start with, viewing data in JSON format (`setopt format=json`) will provide an example of what the raw output looks like from the server. In its raw form, the CLI does not make any attempt to interpret the results, so references are displayed as references (and not names), and sizes are displayed as their raw numeric value.

This is helpful for scripting, but the CLI also has a mode to display the requests being sent to the server, the responses received, and the URLs used. To enable this mode, run `setopt trace=true` . Once you have determined how to do something through the CLI, you can use this mode as the basis for building direct integration with the raw HTTP APIs.

```
delphix group> setopt trace=true
delphix group> create
delphix group create *> set name=example
delphix group create *> set description="this is an example"
delphix group create *> commit
=== POST /resources/json/delphix/group ===
{
    "type": "Group",
    "name": "example",
    "description": "this is an example"
}
=== RESPONSE ===
{
    "status": "OK",
    "result": "GROUP-3",
    "action": "ACTION-37",
    "job": null
}
=== END ===
    GROUP-3
delphix group> "example"
delphix group "example"> delete
=== GET /resources/json/delphix/group/GROUP-3 ===
=== RESPONSE ===
{
    "status": "OK",
    "result": {
```

```
        "type": "Group",
        "reference": "GROUP-3",
        "namespace": null,
        "name": "example",
        "description": "this is an example"
    },
    "action": null,
    "job": null
}
=== END ===
delphix group "example" delete *> commit
=== POST /resources/json/delphix/group/GROUP-3/delete ===
{}
=== RESPONSE ===
{
    "status": "OK",
    "result": "",
    "action": "ACTION-38",
    "job": null
}
=== END ===
delphix group>
```

When using trace mode within the context of a specific object, we refresh the contents of the object before executing each command. This results in the `GET` request before the `delete` command in the above example.

13.3.5 GUI API mapping

This topic describes how to map from GUI operations to the corresponding CLI operation.

It is not always straightforward to convert from the visual layout of the GUI to the corresponding CLI operations. This topic outlines some common operations and indicates how they are represented in the CLI. web services, and the API documentation.

13.3.5.1 dSource operations

GUI	CLI	API topic	Input object	Web services
Link	`database link`	`Container`	`LinkParameters`	`POST /resources/json/delphix/database/link`

GUI	CLI	API topic	Input object	Web services
Show configuration	`database "name" get` `source "name" get`	`Container` `Source`	`-`	`GET /resources/json/delphix/database/{ref}` `GET /resources/json/delphix/source/{ref}`
Sync	`database "name" sync`	`Container`	`SyncParameters`	`POST /resources/json/delphix/database/{ref}/sync`
Update	`database "name" update`	`Container`	`Container`	`POST /resources/json/delphix/database/{ref}`
Delete	`database "name" delete`	`Container`	`DeleteParameters`	`POST /resources/json/delphix/database/{ref}/delete`
Detach	`database "name" detachSource`	`Container`	`DetachSourceParameters`	`POST /resources/json/delphix/database/{ref}/detachSource`
Attach	`database "name" attachSource`	`Container`	`AttachSourceParameters`	`POST /resources/json/delphix/database/{ref}/attachSource`
Disable	`source "name" disable`	`Source`	`SourceDisableParameters`	`POST /resources/json/delphix/source/{ref}/disable`
Enable	`source "name" enable`	`Source`	`SourceEnableParameters`	`POST /resources/json/delphix/source/{ref}/enable`

13.3.5.2 VDB operations

GUI	CLI	API topic	Input object	Web services
Provision	`database provision`	`Container`	`ProvisionParameters`	`POST /resources/json/delphix/database/provision`
V2P	`database export`	`Container`	`ExportParameters`	`POST /resources/json/delphix/database/export`
Refresh	`database "name" refresh`	`Container`	`RefreshParameters`	`POST /resources/json/delphix/database/{ref}/refresh`
Snapshot	`database "name" sync`	`Container`	`SyncParameters`	`POST /resources/json/delphix/database/{ref}/sync`
Update	`database "name" update`	`Container`	`Container`	`POST /resources/json/delphix/database/{ref}`
Delete	`database "name" delete`	`Container`	`DeleteParameters`	`POST /resources/json/delphix/database/{ref}/delete`
Start	`source "name" start`	`Source`	`StartParameters`	`POST /resources/json/delphix/source/{ref}/start`
Stop	`source "name" stop`	`Source`	`StopParameters`	`POST /resources/json/delphix/source/{ref}/stop`

GUI	CLI	API topic	Input object	Web services
Enable	`source "name" enable`	`Source`	`SourceEnableParameters`	`POST /resources/json/delphix/source/{ref}/enable`
Disable	`source "name" disable`	`Source`	`SourceDisableParameters`	`POST /resources/json/delphix/source/{ref}/disable`

13.3.5.3 Environment operations

GUI	CLI	API topic	Input object	Web services
Add environment	`environment create`	`SourceEnvironment`	`SourceEnvironmentCreateParameters`	`POST /resources/json/delphix/environment`
Update	`environment "name" update`	`SourceEnvironment`	`Environment`	`POST /resources/json/delphix/environment/{ref}`
Delete	`environment "name" delete`	`SourceEnvironment`	`-`	`DELETE /resources/json/delphix/environment/{ref}`
Refresh	`environment "name" refresh`	`SourceEnvironment`	`-`	`POST /resources/json/delphix/environment/{ref}/refresh`
Enable	`environment "name" enable`	`SourceEnvironment`	`-`	`POST /resources/json/delphix/environment/{ref}/enable`

GUI	CLI	API topic	Input object	Web services
Disable	`environment "name" disable`	`SourceEnvironment`	`-`	`POST /resources/json/delphix/environment/{ref}/disable`
Add manual repository	`repository create`	`SourceRepository`	`SourceRepository`	`POST /resources/json/delphix/repository`
Update repository	`repository "name" update`	`SourceRepository`	`SourceRepository`	`POST /resources/json/delphix/repository/{ref}`
Remove manual repository	`repository "name" delete`	`SourceRepository`	`-`	`DELETE /resources/json/delphix/repository/{ref}`
Show host details	`host "name" get`	`Host`	`-`	`GET /resources/json/delphix/host/{ref}`
Add cluster node	`environment oracle clusternode create`	`OracleClusterNode`	`OracleClusterNode`	`POST /resources/json/delphix/environment/oracle/clusternode`
Update cluster node	`environment oracle clusternode "name" update`	`OracleClusterNode`	`OracleClusterNode`	`POST /resources/json/delphix/environment/oracle/clusternode/{ref}`
Remove cluster node	`environment oracle clusternode "name" delete`	`OracleClusterNode`	`-`	`DELETE /resources/json/delphix/environment/oracle/clusternode/{ref}`

GUI	CLI	API topic	Input object	Web services
Add listener	`environment oracle listener create`	`OracleListener`	`OracleListener`	`POST /resources/json/delphix/environment/oracle/listener`
Update listener	`environment oracle listener "name" update`	`OracleListener`	`OracleListener`	`POST /resources/json/delphix/environment/oracle/listener/{ref}`
Remove listener	`environment oracle listener "name" delete`	`OracleListener`	`-`	`DELETE /resources/json/delphix/environment/oracle/listener/{ref}`

13.3.6 CLI to Python transition

This topic describes using the CLI to understand the Python APIs.

The command-line interface (see page 2521) is a direct translation of the web services API to an interactive environment. This allows you to use the CLI to explore functionality with tab completion, integrated help, stronger type checking, and indication of expected types and required fields. When trying to determine how to invoke an operation through the Python API or interpret the results, it is recommended that you first learn how to do the same through the CLI, and then use the provided tools to translate that into Python calls.

13.3.6.1 Installation

The Delphix Python API is available through PyPI and you may install it with pip.

```
pip install delphixpy
```

Minimum python version
Requiring Version 2.7 and above

13.3.6.2 Connecting to the Delphix engine

In the Delphix Python API, all operations take an engine object which represents your connection to a Delphix Engine. Here is how you connect to the Delphix Engine using the Python API and acquire the engine object.

```
from delphixpy.delphix_engine import DelphixEngineengine = DelphixEngine("delphix-address", "delphix-user", "delphix-password", "DOMAIN")  # Instead of DOMAIN, use SYSTEM if you are using the sysadmin user.
```

13.3.6.3 CLI translation to Python

For backward compatibility purposes, delphixpy provides the ability to write integrations against a specific API version. The latest version is always in the root of the package. Writing against the latest version requires you to update your integrations if the API changes in future versions of the API.

Specific API versions can be used by importing the corresponding sub-package. The sub-packages are named after the API versions in the format `v<major>_<minor>_<micro>`. For example, if you want to look into API 1.5.0, you should be using modules from `delphixpy.v1_5_0`. Modules from different sub-package versions cannot interact with each other so be careful if you wish to mix API versions in the same code base.

All CLI namespaces have a corresponding Python package in which operations can be accessed. The main Python package is called `web`. All value objects which can be manipulated or read through the CLI can be found in `web.vo`.

```
delphix> database provision
```

Is equivalent to:

```
from delphixpy.web import databasedatabase.provision(engine, provision_parameters)
```

The provision_parameters object in this example is an instance of ProvisionParameters which can be found in delphixpy.web.vo. The properties of the object map to the parameters you would need to specify before doing a commit in the CLI provision context.

This is an example of an "operation", as they are invoked on an object. Operations that are invoked on a particular object take a reference to that object.

```
delphix> database "dexample" refresh
```

Is equivalent to (connection code omitted):

```
from delphixpy.web import databasedatabase.refresh(engine, "ORACLE_DB_CONTAINER-3",
RefreshParameters)
```

While the CLI uses names to refer to objects, the Python API, just like the web services, use references (ORACLE_DB_CONTAINER-3). Persistent objects (those with a permanent unique identity) have a `reference` field that is used in all cases to refer to the object. This allows references to remain constant even if objects are renamed.

For the standard CRUD (create, read, update, delete) operations, the mapping is slightly different:

CLI operation	Python API
`group list`	`group.get_all(engine)`
`group create`	`group.create(engine, group=`
`group "dexample" get`	`group.get(engine,`
`group "dexample" update`	`group.update(engine, , group=)`
`group "dexample" delete`	group.delete(engine,

13.3.6.4 Example: creating a group

This is how you can create a group as a fully working example.

```
from delphixpy.web import groupfrom delphixpy.web.vo import Groupfrom
delphixpy.delphix_engine import DelphixEngineengine = DelphixEngine("delphix-address",
 "delphix-user", "delphix-password", "DOMAIN")my_group = Group()my_group.name = "My
Group"my_group.description = "This is my new group!"group.create(engine, my_group)
```

13.3.6.5 Asynchronous mode

The Python API runs by default in synchronous mode. If you would wish to perform operations asynchronously there is a context manager that allows you to do that. If you need to track job progress in asynchronous mode, you can get the reference of the last job started from engine.last_job. When exiting the async context manager, it will wait for all jobs started within the context to finish. You can also clear the job from the context so that you do not wait for its completion or status when exiting the context manager. If a job fails, exceptions.JobError will be thrown.

Here is how you would perform a sync operation on all databases asynchronously.

```
from delphixpy.delphix_engine import DelphixEnginefrom delphixpy import
 job_contextfrom delphixpy.web import databaseengine = DelphixEngine("delphix-
address", "delphix-user", "delphix-password", "DOMAIN")all_databases =
database.get_all(engine)with job_context.asyncly(engine):    for db in all_databases:
database.sync(engine, db.reference)
```

13.3.7 Python cookbook: adding a UNIX host

This topic describes the process of adding a UNIX host using the delphixpy Python API.

Within Delphix, there are both hosts and host environments. A host represents a remote system, but may or may not be a source or target for linking or provisioning. For example, in an Oracle RAC cluster, the cluster environment represents the location of the Oracle installation(s), and while there are hosts within that cluster they are not individually manageable as environments.

13.3.7.1 Procedure

1. Create new environment creation parameters and initialize the structure for a UNIX host.

   ActionScript

   ```
   from delphixpy.web.vo import HostEnvironmentCreateParameters,
   UnixHostEnvironment, UnixHostCreateParameters, UnixHost, EnvironmentUser,
   PasswordCredentialhost_environment_create_parameters_vo =
   HostEnvironmentCreateParameters()host_environment_create_parameters_vo.host_env
   ironment =
   UnixHostEnvironment()host_environment_create_parameters_vo.host_parameters =
   UnixHostCreateParameters()host_environment_create_parameters_vo.host_parameters
   .host = UnixHost()host_environment_create_parameters_vo.primary_user =
   EnvironmentUser()host_environment_create_parameters_vo.primary_user.credential
   = PasswordCredential()
   ```

2. Set the host address and port. The name of the environment is derived from the address used, though you can provide a more descriptive name if desired. The address can be a DNS names, IP addresses, or a comma separated list of the above.

   ActionScript

   ```
   host_environment_create_parameters_vo.host_parameters.host.addresses =
   ["192.168.1.2"]host_environment_create_parameters_vo.host_parameters.host.port
   = 22
   ```

3. Set the toolkit path. This is where Delphix will store temporary binaries used while the host is configured as part of Delphix.

   ActionScript

```
host_environment_create_parameters_vo.host_parameters.host.toolkit_path = "/
var/delphix"
```

4. Set the username and password to use when connecting over SSH. This user must have the privileges described in the Delphix Administration Guide. To configure a SSH user, change the credential object to `SystemKeyCredential`.

   ActionScript

```
host_environment_create_parameters_vo.primary_user.name = "oracle"host_environm
ent_create_parameters_vo.primary_user.credential.password = "my secret
password"
```

5. Commit the result. A reference to your new environment will be returned from the create call. The environment discovery process will execute as an asynchronous job. The default behavior is to wait for the result, so progress will be updated until the discovery process is complete or fails.

   ActionScript

```
from delphixpy.delphix_server import DelphixServerfrom delphixpy.web import
 environmentserver = DelphixServer("delphix-address", "delphix-user", "delphix-
password", "DOMAIN")new_environment_reference = environment.create(server,
host_environment_create_parameters_vo)
```

6. Full example

   ActionScript

```
from delphixpy.delphix_server import DelphixServerfrom delphixpy.web import
 environmentfrom delphixpy.web.vo import HostEnvironmentCreateParameters,
UnixHostEnvironment, UnixHostCreateParameters, UnixHost, EnvironmentUser,
PasswordCredentialhost_environment_create_parameters_vo =
HostEnvironmentCreateParameters()host_environment_create_parameters_vo.host_env
ironment =
UnixHostEnvironment()host_environment_create_parameters_vo.host_parameters =
UnixHostCreateParameters()host_environment_create_parameters_vo.host_parameters
.host = UnixHost()host_environment_create_parameters_vo.primary_user =
EnvironmentUser()host_environment_create_parameters_vo.primary_user.credential
=
PasswordCredential()host_environment_create_parameters_vo.host_parameters.host.
addresses = ["192.168.1.2"]host_environment_create_parameters_vo.host_parameter
s.host.port =
22host_environment_create_parameters_vo.host_parameters.host.toolkit_path = "/
var/delphix"host_environment_create_parameters_vo.primary_user.name = "oracle"h
ost_environment_create_parameters_vo.primary_user.credential.password = "my
secret password"server = DelphixServer("delphix-address", "delphix-user",
```

```
"delphix-password", "DOMAIN")new_environment_reference =
environment.create(server, host_environment_create_parameters_vo)
```

13.3.8 Working with Delphix APIs

Use the pages in this section to learn more about how you can leverage Delphix APIs.

- Background information (see page 2774)
- Delphix API reference URLs (see page 2774)
- API prerequisite knowledge (see page 2777)
- Delphix RESTFul APIs command line basics (see page 2784)
- API shell scripts programming language examples (see page 2789)
- JSON parsing (see page 2793)
- API use case commands and scripts (see page 2803)
- API programming language examples (see page 2823)
- API timeflows (see page 2827)

13.3.8.1 Background information

This document assumes that you have some basic Delphix product experience and entry-level programming knowledge. The first two sections of this document, Delphix API Reference URLs (see page 2774) and API Prerequisite Knowledge (see page 2777), are focused on providing the required information and reference material/URLs.

This document is for informational and demonstration purposes only. The examples are for demonstration purposes only and must be used at your own risk. As always, test and verify on development systems prior to migrating code to production environments.

13.3.8.1.1 What is RESTFul programming?

http://stackoverflow.com/questions/671118/what-exactly-is-restful-programming

> A great way to learn how to generate the Delphix RESTFul API calls and the required JSON content is to use the Delphix CLI (Command Line Interface) and turn on the trace option.
> `Delphix5110HWv8>` ***setopt trace=true***
> All subsequent CLI commands will display the GET or POST API URL with the respective input or output JSON data string. This guide will walk you through an example later.

13.3.8.2 Delphix API reference URLs

There are a number of sources available to provide details, examples, and techniques for working with Delphix APIs. This section contains a small list of URLs that are worth reviewing/reading as required.

13.3.8.2.1 CLI (command line interface)

The Delphix Engine provides a native command-line interface (CLI) accessible over SSH. This CLI provides an interactive layer on top of the public web service APIs, and is intended for users that wish to automate interactions with the Delphix Engine, or simply prefer a text-based interface. All of the functionality available in the CLI is also available through the public stable web service APIs should more full-featured automation be required. For more information on automation using CLI commands see Command Line Interface Guide (see page 2521)

13.3.8.2.2 RESTFul APIs

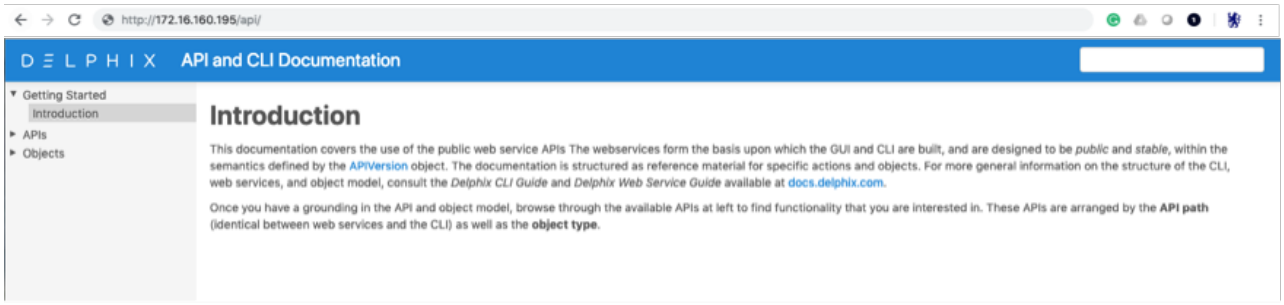
> **Must be logged in**
> Users must be successfully logged in before /API pages can be accessed.

For more information on automation using the web service APIs, see the Web Services API Guide (see page 2752).

API Documentation is also included within the Delphix Engine using the following formula:

http://<delphix_engine>/api/

For Example http://172.16.160.195/api/



For a complete list of Delphix APIs - JSON schema format, use the following URL:

http://<delphix_engine>/api/json/delphix.json

172.16.160.195/api/json/delphix.json


```
{
    "/delphix-about.json": {
        "cliVisibility": [
            "DOMAIN",
            "SYSTEM"
        ],
        "description": "Retrieve static system-wide properties.",
        "extends": {
            "$ref": "/delphix-typed-object.json"
        },
        "name": "PublicSystemInfo",
        "properties": {
            "apiVersion": {
                "$ref": "/delphix-version.json",
                "description": "Maximum supported API version of the current system software.",
                "type": "object"
            },
            "banner": {
                "description": "Security banner to display prior to login.",
                "type": "string"
            },
            "buildTimestamp": {
                "description": "Time at which the current system software was built.",
                "format": "date",
                "type": "string"
            },
            "buildTitle": {
                "description": "Description of the current system software.",
                "type": "string"
            },
            "buildVersion": {
                "$ref": "/delphix-version-info.json",
                "description": "Delphix version of the current system software.",
                "type": "object"
            },
            "configured": {
                "description": "Indicates whether the server has gone through initial setup or not.",
                "type": "boolean"
            },
            "currentLocale": {
                "description": "The current system locale.",
                "format": "locale",
                "type": "string"
            },
            "enabledFeatures": {
                "description": "The list of enabled features on this Delphix Engine.",
                "items": {
                    "type": "string"
                },
                "type": "array"
```


So, looking at the first JSON key/name:

```
"/delphix-about.json": {    "cliVisibility": [        "DOMAIN",         "SYSTEM"    ]
,     "description": "Retrieve static system-wide properties.",     . . .
```

And after logging into the Delphix Engine, translating this into the URL API for *about*:

http://172.16.160.195/resources/json/delphix/about will respond with the returned JSON data string.


```
{"type":"OKResult","status":"OK","result":{"type":"PublicSystemInfo","productType":"s
tandard","productName":"Delphix Engine","buildTitle":"Delphix Engine 5.1.2.0","buildT
imestamp":"2016-09-02T22:28:43.000Z","buildVersion":{"type":"VersionInfo","major":5,"
minor":1,"micro":2,"patch":0},"configured":true,"enabledFeatures":["XPP","JETSTREAM"]
```

```
,"apiVersion":{"type":"APIVersion","major":1,"minor":8,"micro":0},"banner":null,"loca
les":["en-US"],"currentLocale":"en-US"},"job":null,"action":null}
```

For now, just remember that the Delphix Engine contains the API Documentation and Delphix JSON schema.

13.3.8.2.3 Masking APIs

Please refer to the Masking documentation at https://maskingdocs.delphix.com[780] for information on the Masking APIs.

13.3.8.2.4 Cookbook examples

Delphix documentation includes a number of cookbook examples that will not be duplicated in this section but may be referenced.

API Cookbook: Common Tasks, Workflows, and Examples (see page 2831)

There are also working examples provided within this document and are available for download.

13.3.8.3 API prerequisite knowledge

13.3.8.3.1 JSON

JSON (JavaScript Object Notation) is a minimal, readable format for structuring data. It is a simple format for transmitting data between applications, as an alternative to XML. The Delphix API uses JSON data structure in the format of strings to send and receive data from the API calls, as you will see later in the examples. First, let's look at the JSON fundamentals.

13.3.8.3.1.1 Keys and values

The two primary parts that makeup JSON are keys and values. Together they make key/value pairs, also called name/value pairs.

- **Key** – Always a string enclosed in quotation marks.
- **Value** – Can be a string, number, boolean expression, array, or object.
- **Key/Value Pair** – Follows a specific syntax, with the key followed by a colon followed by the value. Key/value pairs are comma separated.

Let's take a JSON sample string and identify each part of the code.

```
{    "foo" : "bar",    "rows" : 100}
```

The curly brackets start and end the string. The key is "foo" and the value is "bar". A colon (:) is the delimiter between them. A comma (,) is the delimiter for multiple key/value pairs. The second pair is "rows" and the value is a number of 100.

780 https://maskingdocs.delphix.com/

13.3.8.3.1.2 Types of values

Number	An integer or a decimal number
Boolean	True or false
String	Plain text alphanumeric readable characters
Null	Empty
Array	An associative array of values
Object	An associative array of key/value pairs

13.3.8.3.1.3 Numbers, booleans, and strings.

It is very important to understand the APIs JSON object definitions. Quoted values are treated as strings!

"x" : "1" is treated as a string, while

"x" : 1 is treated as a number

"y" : "true" is treated as a string, while

"y" : true is treated as a boolean true (false)

13.3.8.3.2 Null values

```
{    "z" :    , "b" : "World"}
```

Nulls are empty values, but sometimes programmers code "" as a null value.

```
{    "z" : ""   , "b" : "World"   }
```

So always verify how the null values are defined and handled by the application.

13.3.8.3.3 Arrays

An array is indicated with the square brackets: [value1, value2, etc.]. In this example, we have added a categories key with an array of values.

```
..."foo" : {  "bar" : "Hello",  "category" : [ "greetings", "morals" ]}...
```

13.3.8.3.4 Objects

An object is indicated by curly brackets: {"key", "value"}. Everything inside of the curly brackets is part of the object. We already learned that a value could be an object. Therefore, "foo" and the corresponding object are a key/value pair.

```
..."foo" : { "bar" : "Hello" }...
```

The key/value pair "bar" : "Hello" is nested inside the key/value pair "foo" : { ... }. That is an example of a hierarchy (or nested data) within JSON data.

Arrays and Objects can be nested or contained within the same level.

13.3.8.3.4.1 Summary

JSON arrays are `[ , , ]`

JSON nested objects are `, , "x":{ "a":"1", "b":"2" }, ,`

JSON data can be passed within the HTTP URL (file or argument), the header, or other handlers.

From within Shell Scripts or Programming Languages, JSON data is typically processed through a "JSON parser." This topic is covered later.

13.3.8.3.5 Delphix CLI

13.3.8.3.5.1 Connecting to the Delphix engine CLI

Reference: Connecting to the CLI (see page 2522)

There are two user roles accessible, the **sysadmin** and the **delphix_admin**.

From a shell environment, you can connect using the ssh command. The IP Address (or Hostname) represents the Delphix Engine (case sensitive):

`ssh sysadmin@127.16.160.195`

`ssh delphix_admin@127.16.160.195`

From a putty session, open an ssh connection to the Delphix Engine IP Address or Hostname (case sensitive):

`open 127.16.160.195`

`Login User: sysadmin@SYSTEM`

`#...` or `...`

`Login User: delphix_admin@DOMAIN`

After entering the correct password for the respective user, the menus for that user's role will be different. For example, the **sysadmin@SYSTEM** user has engine storage, whereas the **delphix_admin@DOMAIN** user has database provisioning.

You can use the CLI for scripting and configure the connection for ssh passwordless connections.

CLI Cookbook: Configuring Key-Based SSH Authentication for Automation (see page 2547)

13.3.8.3.5.2 How to use the CLI to learn the APIs

As stated earlier, a great way to learn how to generate the Delphix RESTFul API calls and the required JSON content is to use the Delphix CLI (Command Line Interface) and turn on the `CLI> setopt trace=true` option.

Below is an example of how to get the JSON required parameters for a database refresh per the type of refresh performed.

> Other types or options may require other JSON parameters, so after changing any parameter, we recommend performing an "ls" command to see if there are any new parameters and/or required values.

```
The refresh database example below shows how to use the CLI to identify reference
objects for other CLI commands and the respective RESTFul API structure when
the  setopt trace=true  option is set.

$ ssh delphix_admin@172.16.160.195 Password: Delphix5030HWv8> ls
Childrenaboutaction...connectivitydatabaseenvironment... toolkituser
OperationsversionDelphix5030HWv8> database Delphix5030HWv8 database> ls
ObjectsNAME            PROVISIONCONTAINER    DESCRIPTIONDPXDEV01        -
Vdelphix_demo    delphix_demo        -delphix_demo    - Scripts         -
V_2C1            Scripts                -Vvfiles            -                    -
Childrentemplate
OperationscreateEmptycreateRestorationDatasetexportfileMappinglinkoracleSupportedChar
acterSetsprovisionvalidateXppxpp
```

First, we need to identify the target Delphix virtualized database object to refresh ...

> ⚠ Each Delphix object has a reference that is typically used for parameter values.

```
Delphix5030HWv8 database> select Vdelphix_demo Delphix5030HWv8 database
'Vdelphix_demo'> ls Properties    type: MSSqlDatabaseContainer    name:
Vdelphix_demo    creationTime: 2016-06-16T14:30:03.033Z    currentTimeflow:
'DB_PROVISION@2016-06-16T10:30:08'    delphixManaged: true    description:
```

```
(unset)    group: Windows    masked: false    os: Windows    performanceMode:
DISABLED    processor: x86    provisionContainer: delphix_demo    reference:
MSSQL_DB_CONTAINER-39     restoration: false    runtime:        type:
MSSqlDBContainerRuntime        logSyncActive: false    sourcingPolicy:        type:
SourcingPolicy        loadFromBackup: false        logsyncEnabled: false    transform
ation: false Operationsdelete...purgeLogsrefreshremoveLiveSource...Delphix5030HWv8
database 'Vdelphix_demo'> refresh Delphix5030HWv8 database 'Vdelphix_demo' refresh *>
ls Properties    type: RefreshParameters    timeflowPointParameters:        type:
TimeflowPointSemantic        container: (required)        location:
LATEST_POINTDelphix5030HWv8 database 'Vdelphix_demo' refresh *> set
timeflowPointParameters.container=delphix_demo Delphix5030HWv8 database
'Vdelphix_demo' refresh *> ls Properties    type:
RefreshParameters    timeflowPointParameters:        type:
TimeflowPointSemantic        container: delphix_demo (*)         location:
LATEST_POINTDelphix5030HWv8 database 'Vdelphix_demo' refresh > *commit     Dispatched
job JOB-100    DB_REFRESH job started for "Windows/Vdelphix_demo".    Validating that
this dataset is managed by Delphix.    Stopping virtual database.    Unmounting
datasets.    Unexporting storage containers.    Metadata for dSource "Vdelphix_demo"
 successfully deleted.    Starting provisioning of virtual database "Vdelphix_demo".
   Creating new TimeFlow.    Generating recovery scripts.    Mounting
datasets.    Mounting read-only source logs dataset.    Running user-specified pre-
provisioning script.    Recovering virtual database.    The virtual database recovery
was successful.    Unmounting read-only source logs dataset.    Running user-
specified post-provisioning script.    The virtual database "Vdelphix_demo" was
successfully provisioned.    DB_REFRESH job for "Windows/Vdelphix_demo" completed
successfully.
```

Refresh again but this time turn on the `setopt trace=true` option.

```
Delphix5030HWv8 database 'Vdelphix_demo'> refreshDelphix5030HWv8 database
'Vdelphix_demo' refresh *> lsProperties    type:
RefreshParameters    timeflowPointParameters:        type:
TimeflowPointSemantic        container: (required)        location:
LATEST_POINTDelphix5030HWv8 database 'Vdelphix_demo' refresh *> set
timeflowPointParameters.container=delphix_demoDelphix5030HWv8 database
'Vdelphix_demo' refresh *> lsProperties    type:
RefreshParameters    timeflowPointParameters:        type:
TimeflowPointSemantic        container: delphix_demo        location:
LATEST_POINTDelphix5030HWv8 database 'Vdelphix_demo' refresh *> setopt
trace=trueDelphix5030HWv8 database 'Vdelphix_demo' refresh *> commit=== POST /
resources/json/delphix/database/MSSQL_DB_CONTAINER-39/refresh ===={    "type":
"RefreshParameters",    "timeflowPointParameters": {        "type":
"TimeflowPointSemantic",        "container": "MSSQL_DB_CONTAINER-38"    }}...
```

> ⚠ The "container" value in the JSON output above is different from the target VDB reference because we are refreshing from the source database container! In this example, the **set timeflowPointParameters.container=delphix_demo** is represented in JSON output as **"container": "MSSQL_DB_CONTAINER-38"**

Using the CLI, you can identify the RESTFul API POST and GET commands along with the JSON input data requirements.

```
=== POST /resources/json/delphix/database/MSSQL_DB_CONTAINER-39/refresh ==={     "typ
e": "RefreshParameters",     "timeflowPointParameters": {         "type":
"TimeflowPointSemantic",         "container": "MSSQL_DB_CONTAINER-38"     } }
```

So framing the RESTFul URL for a virtual database refresh, the URL will look like

http://<delphix_engine>/resources/json/delphix/database/ MSSQL_DB_CONTAINER-39 /refresh

where the **MSSQL_DB_CONTAINER-39** represents the target virtualized database to refresh. We need to POST the JSON data to the URL for processing.

```
{    "type": "RefreshParameters",     "timeflowPointParameters": {          "type":
"TimeflowPointSemantic",         "container": "MSSQL_DB_CONTAINER-38"     }}
```

The **"timeflowPointParameters"** key has 6 **"type": "..."** options, each of which has its own set of parameters. The type **"TimeflowPointSemantic"** uses the default LATEST_POINT within the source container, so for simplicity, we will use this type. For more information on timeflowPointParameters 6 types, see the Advanced Section.

If this is a little confusing at this point, do not worry, that's typical. Complete examples will be shown later. The important items to remember are:

- Delphix often uses object reference names within the JSON data.
- Using the `setopt trace=true` the option provides the construct for the RESTFul API URLs and the JSON data for POST / GET operations.

13.3.8.3.6 HTTP

We use the HTTP protocol every day for web browsing and commercial business. From finding a new restaurant to buying a 1986 Ford Thunderbird Turbo Coupe!

Most people see the HTTP within the URL Address field within the Web Browser window – for example, http://www.google.com[781]

But behind the scenes, HTTP is performing a wide range of functionality. For RESTFul APIs, they use HTTP's GET and POST form functionality to process data. In Delphix's case, the data is also represented as JSON structures.

781 http://www.google.com/

HTTP GET operation is used to return data only, while HTTP POST operation is used to provide data input in the form of a structured JSON data string or file.

13.3.8.3.7 cURL

13.3.8.3.7.1 What is cURL?

The cURL[782] client command is based on a library supporting a number of web protocols, including HTTP. The "curl" command can be called from the command line, while the cURL library is commonly integrated with your favorite programming languages, such as Java, JSP, Python, Perl, PHP, .NET, and PowerShell.

Due to its widespread adoption, we will use cURL for making the Delphix RESTFul API calls within this document. Some operating systems or languages support their own HTTP commands / related libraries, and you can use these instead of cURL. One alternative is the "wget" command described later.

13.3.8.3.8 Is cURL installed?

```
Operating System Prompt>    curl --versioncurl 7.19.7 (x86_64-redhat-linux-gnu)
libcurl/7.19.7 NSS/3.19.1 Basic ECC zlib/1.2.3 libidn/1.18 libssh2/1.4.2Protocols:
tftp ftp telnet dict ldap ldaps http file https ftps scp sftp Features: GSS-Negotiate
IDN IPv6 Largefile NTLM SSL libz
```

Get the HTTP output from google.com[783]

```
Operating System Prompt> curl www.google.com
```

13.3.8.3.9 Wget

An alternative to cURL is Wget, which is typically a native command on all Linux environments. See the Appendix for a complete comparison between Wget and cURL.

13.3.8.3.10 dxtoolkit2

Delphix has developed a very robust toolkit, dxtoolkit2, which utilizes the Delphix RESTFul APIs. This toolkit is cross-platform. Its commands are built with the Perl programming language.

We recommend that you review the dxtoolkit2 documentation; you may find a utility that already performs your desired function. For example, the utility ***dx_get_analytics*** is absolutely great for dumping analytic data from the Delphix Engine into a .csv (comma-separated value) format, which you can then easily integrate into your enterprise monitoring tools. See the sample "Analytics" use case.

Contact Delphix personnel for the latest download.

782 https://en.wikipedia.org/wiki/CURL
783 http://google.com/

13.3.8.4 Delphix RESTFul APIs command line basics

13.3.8.4.1 Authentication

RESTFul APIs require authentication. Just plugging the URL into a web browser or running an operating system cURL command will return an authentication/login required error message.

Command line:

curl http://172.16.160.195/resources/json/delphix/environment

Response:

```
<!DOCTYPE html><html><head><title>Apache Tomcat/8.0.29 - Error report</title><style
type="text/css">H1 {font-family:Tahoma,Arial,sans-serif;color:white;background-
color:#525D76;font-size:22px;} H2 {font-family:Tahoma,Arial,sans-
serif;color:white;background-color:#525D76;font-size:16px;} H3 {font-
family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:14px;}
BODY {font-family:Tahoma,Arial,sans-serif;color:black;background-color:white;} B
{font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;} P {font-
family:Tahoma,Arial,sans-serif;background:white;color:black;font-size:12px;}A
{color : black;}A.name {color : black;}.line {height: 1px; background-color: #525D76;
border: none;}</style> </head><body><h1>HTTP Status 403 - Use /resources/json/
delphix/login to log in first</h1><div class="line"></div><p><b>type</b> Status
report</p><p><b>message</b> <u>Use /resources/json/delphix/login to log in first</
u></p><p><b>description</b> <u>Access to the specified resource has been forbidden.</
u></p><hr class="line"><h3>Apache Tomcat/8.0.29</h3></body></html>
```

The authentication process requires you to establish a session first.

The example within this section illustrates the session/login and subsequent API calls with cURL using cookies created when the session was established.

13.3.8.4.1.1 Session

API Version Information (see page 2753)

When programming for compatibility, the API version number is very important. Please be aware of the differences between versions for enterprise applications. For example, if you specify a 1.11.6 version, ONLY

the available calls and functionality for that version will be used, and it will only be operational on Delphix Engine versions that support that version.

> The session uses the –c for the cookie creation, the login uses the -b for the exiting cookie created by the session, and -c to create a new cookie, the other commands use the –b for using the existing cookie created by the login.

From the Unix/Linux command line

```
$ curl -s -X POST -k --data @- http://delphix-server/resources/json/delphix/session \
-c ~/cookies.txt -H "Content-Type: application/json" <<EOF{   "type": "APISession",
"version": {       "type": "APIVersion",       "major": 1,       "minor": 11,
"micro": 6   }}EOF
```

Returned to the command line are the results (added linefeeds for readability)

```
{   "status":"OK",   "result": {       "type":"APISession",       "version": {
"type": "APIVersion",           "major": 1,           "minor": 11,           "micro":
6       },       "locale": "en_US",       "client": null   },   "job": null}
```

13.3.8.4.1.2 Login

Once you have established the session, the next step is to authenticate to the server by executing the Login Request API. Unauthenticated sessions are prohibited from making any API calls other than this login request. The username can be either a system user or domain user; the backend will authenticate using the appropriate method.

> The session uses the –c for the cookie creation, the login uses the -b for the exiting cookie created by the session, and -c to create a new cookie, the other commands use the –b for using the existing cookie created by the login.

From the Unix/Linux command line

```
$ curl -s -X POST -k --data @- http://delphix-server/resources/json/delphix/login \
-b ~/cookies.txt -c ~/cookies.txt -H "Content-Type: application/json"
<<EOF{   "type": "LoginRequest",   "username": "delphix_username",   "password":
"delphix_password"}EOF
```

Returned to the command line are the results (added linefeeds for readability)

```
{"status":"OK","result":"USER-2","job":null,"action":null}
```

13.3.8.4.2 Sample Delphix API call

With a successful authentication (session, login, and saved cookie), calls to the Delphix Engine can now be made to perform the desired functionality.

> The session uses the –c for the cookie creation, the login uses the -b for the exiting cookie created by the session, and -c to create a new cookie, the other commands use the –b for using the existing cookie created by the login.

For starters, let's create a session, login, and get the existing environments defined within the Delphix Engine.

```
curl -s -X POST -k --data @- http://172.16.160.195/resources/json/delphix/session \-c
~/cookies.txt -H "Content-Type: application/json" <<EOF{    "type":
"APISession",    "version": {        "type": "APIVersion",        "major":
1,        "minor": 7,        "micro": 0    }}EOF
```

```
curl -s -X POST -k --data @- http://172.16.160.195/resources/json/delphix/login
\    -b ~/cookies.txt -c ~/cookies.txt -H "Content-Type: application/json"
<<EOF{    "type": "LoginRequest",    "username": "admin",    "password": "delphix"}
EOF
```

```
curl -X GET -k http://172.16.160.195/resources/json/delphix/environment \    -b ~/
cookies.txt -H "Content-Type: application/json"
```

Returned to the command line are the results (added linefeeds for readability)

```
{"type":"ListResult","status":"OK","result":[ {"type":"WindowsHostEnvironment",
"reference":"WINDOWS_HOST_ENVIRONMENT1",  "namespace":null,  "name":"Window Target",
"description":"",  "primaryUser":"HOST_USER-1",  "enabled":false,  "host":"WINDOWS_HO
ST1",  "proxy":null }, {  "type":"UnixHostEnvironment",  "reference":"UNIX_HOST_ENVIR
ONMENT-3",  "namespace":null,  "name":"Oracle Target",  "description":"",
"primaryUser":"HOST_USER-3",  "enabled":true,  "host":"UNIX_HOST-3","aseHostEnvironme
ntParameters":null }],"job":null,"action":null,"total":2,"overflow":false}
```

13.3.8.4.3 Windows PowerShell authentication example

See the PowerShell section below if cURL is not yet available on your operating system.

These commands work on Windows Command Prompt with the respective JSON files: session.json and login.json
Filename: session.json

```
{    "type": "APISession",    "version": {        "type": "APIVersion",        "major
": 1,        "minor": 7,        "micro": 0    }}
```

Filename: login.json

```
{"type": "LoginRequest","username": "admin","password": "delphix"}
```

(In Powershell you use curl.exe or modify the default alias)...

```
curl.exe --insecure -c cookies.txt -sX POST -H "Content-Type: application/json" -d
"@session.json" http://172.16.160.195/resources/json/delphix/sessioncurl.exe --
insecure -b cookies.txt -c cookies.txt -sX POST -H "Content-Type: application/json"
-d "@login.json" http://172.16.160.195/resources/json/delphix/logincurl.exe --
insecure -b cookies.txt -sX GET -H "Content-Type: application/json" -k http://
172.16.160.195/resources/json/delphix/system
```

Putting the above commands within a Powershell script:

Filename: auth1.ps1

```
<#Filename: auth.ps1Description: Delphix Powershell Sample Authentication
Script ...Date: 2016-08-02Author: Bitt...#>
```

Variables ...

```
$nl = [Environment]::NewLine$BaseURL = "http://172.16.160.195/resources/json/delphix"$
cookie = "cookies.txt"
```

Session JSON Data ...

```
write-output "${nl}Creating session.json file ..."$json = @"{    "type": "APISession"
,    "version": {        "type": "APIVersion",        "major": 1,        "minor":
11,        "micro": 6    }}"@
```

Output File using UTF8 encoding ...

```
write-output $json | Out-File "session.json" -encoding utf8
```

Delphix cURL Session API ...

```
write-output "${nl}Calling Session API ...${nl}"$results = (curl.exe --insecure -c
"$cookie" -sX POST -H "Content-Type: application/json" -d "@session.json" -k
$BaseURL/session)write-output "Session API Results: ${results}"
```

Login JSON Data ...

```
write-output "${nl}Creating login.json file ..."$user = "admin"$pass = "delphix"$json
= @"{    "type": "LoginRequest",    "username": "${user}",    "password": "${pass}"}"@
```

Output File using UTF8 encoding ...

```
write-output $json | Out-File "login.json" -encoding utf8
```

Delphix cURL Login API ...

```
write-output "${nl}Calling Login API ...${nl}"$results = (curl.exe --insecure -b
"$cookie" -c "$cookie" -sX POST -H "Content-Type: application/json" -d "@login.json"
 -k $BaseURL/login)write-output "Login API Results: ${results}"
```

Delphix cURL system API ...

```
write-output "${nl}Calling System API ...${nl}"$results = (curl.exe --insecure -b
"$cookie" -sX GET -H "Content-Type: application/json" -k $BaseURL/system)write-output
"System API Results: ${results}"
```

The end is near ...

```
echo "${nl}Done ...${nl}"exit;
```

Sample Powershell Script Output:

```
PS> . .\auth.ps1Creating session.json file ...Calling Session API ...Session API
Results:{"type":"OKResult","status":"OK","result":{"type":"APISession","version":{"ty
pe":"APIVersion","major":1,"minor":7,"micro":0},"locale":null,"client":null},"job":nu
ll,"action":null}Creating login.json file ...Calling Login API ...Login API Results:
{"type":"OKResult","status":"OK","result":"USER2","job":null,"action":null}Calling
System API ... System API Results: {"type":"OKResult","status":"OK","result":{"type":"
SystemInfo","productType":"standard","productName":"Delphix Engine","buildTitle":"Del
phix Engine 5.1.1.0","buildTimestamp":"20160721T07:23:41.000Z","buildVersion":{"type":"
VersionInfo","major":5,"minor":1,"micro":1,"patch":0},"configured":true,"enabedFeatur
es":["XPP","MSSQLHOOKS"],"apiVersion":{"type":"APIVersion","major":1,"minor":8,"micro
":0},"banner":null,"locals":["enUS"],"currentLocale":"enUS","hostname":"Delphix5110HW
v8","sshPublicKey":"ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDOsrp7Aj6hFQh9yBq7273B+qtPKmCu1B18nPvr08yjt/
IZeM4qKk7caxExQS9rpfU8AWoT7e8ESV7NkBmUzOHrHnLsuJtxPqeYoqeMubVxYjJuxlH368sZuYsnB04KM0m
```

```
i39e15lxVGvxQk9tyMpl7gs7cXRz1k6puncyiczU/
axGq7ALHU2uyQoVmlPasuHJbq23d21VAYLuscbtgpZLAFlR8eQH5Xqaa0RT+aQJ6B1ihZ7S0ZN914M2gZHHNY
cSGDWZHwUnBGttnxx1ofRcyN4/qwT5iHq5kjApjSaNgSAU0ExqDHiqgTq0wttf5nltCqGMTFR7XY38HiNq+
+atDroot@Delphix5110HWv8\n","memorySize":8.58107904E9,"platform":"VMware with BIOS
date 05/20/2014","uuid":"564d7e1df4cb-f91098fd348d74817683","processors":[{"type":"CP
UInfo","speed":2.5E9,"cores":1}],"storageUsed":2.158171648E9,"storageTotal":2.0673724
416E10,"installationTime":"2016-07-27T13:28:46.000Z"},"job":null,"action":null}
Done ... PS>
```

13.3.8.5 API shell scripts programming language examples

13.3.8.5.1 Why use shell scripts?

Shell scripts are great tools for rapid development and validation for simple (smaller) requirements. Most development is done iteratively, and shell scripts provide immediate feedback on logic and code.

Company-supported programming languages are the preferred tools for enterprise applications.

Most likely, your company employs more Java and/or PHP programmers than Linux Shell script programmers.

The "use cases" will be programmed using either Unix/Linux Shell and/or Windows PowerShell scripts. You can easily port the logic from these scripts into your favorite programming language. Basic examples of connection with API will be provided for a number of major programming languages in a later section of this document.

13.3.8.5.2 Linux/Unix/(and Mac too) shell scripts

There are numerous shell environments, including sh, bash, csh, ksh, and tsh. Identify the current shell environment using any of the commands below:

```
ps -p $$
ps -p $$ -ocomm=
echo $0
ps -ef | grep $$ | grep -v grep
ps -ef | egrep "^\s*\d+\s+$$\s+"
```

The examples provided have all been run from the bash shell environment and may or may not run the same as the other shells. We recommend that you start a bash shell by typing **bash** at the operating system prompt, like this:

```
Operating_System_Prompt> bash
```

The scripts included within this document have all been run on Linux and Mac environments within a bash shell and are NOT certified by any means. As always, test and verify in development for your environment.

For non-Linux platforms and non-bash shell environments, please re-validate for your configuration and search the web for any alternative methods/tools/utilities that may perform the same actions.

13.3.8.5.3 Windows PowerShell

Powershell Open Source is now available for Linux and Mac OS. for more information refer to Microsoft Open Source[784]

13.3.8.5.3.1 Requirements

Windows has a number of versions of Powershell. The minimum version for Delphix is 2.0 for SQL Server 2008 environments. There are numerous enhancements and features with subsequent Powershell versions. Additionally, you must be aware of the architecture of 32bit or 64bit Powershell versions you are running from within.

```
PS> $PSVersionTable.PSVersion

Major Minor Build Revision
----- ----- ----- --------
2 0 -1 -1
```

13.3.8.5.3.2 32bit or 64 bit

If executing Powershell scripts from within Delphix Pre/Post Scripts commands or Delphix hooks, the default Powershell used is 32 bit, whereas the typical default Windows Powershell is 64 bit. However, Powershell allows you to execute 64 bit Powershell command from within the 32 bit environment. Shown below is a simple alias, ps64, to execute 64bit Powershell scripts.

```
PS> set-alias
ps64 "$env:windir\sysnative\WindowsPowerShell\v1.0\powershell.exe"
```

Sample call to execute 64bit Powershell script

```
PS> ps64 [path\to\any_64bit_powershell_script].ps1
```

784 https://techcrunch.com/2016/08/18/microsoft-open-sources-powershell-brings-it-to-linux-and-os-x/

Courtesy of this article: http://www.gregorystrike.com/2011/01/27/how-to-tell-if-powershell-is-32-bit-or-64-bit/

```
PS> if ($env:Processor_Architecture -eq "x86") { write "running on 32bit" }
else {write "running on 64bit"}
running on 32bit
```

. . . or . . .

```
PS> if ([System.IntPtr]::Size -eq 4) { "32-bit" } else { "64-bit" }
32-bit
```

It is worth noting that the locations of the 32-bit and 64-bit versions of Powershell are somewhat misleading. The 32-bit PowerShell is found at C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe

and the 64-bit PowerShell is at C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

13.3.8.5.4 Execution of scripts security disabled

It is possible to disable Powershell environments on the system. If they are disabled, you will see the following error for any Powershell script that you try to execute.

```
PS> . .
[any_powershell_script].ps1
File [any_powershell_script].ps1 cannot be loaded because the execution of
scripts is disabled on this system. Please see "get-help about signing" for
more details.
At line:1 char:2
+ . <<<< .\ [any_powershell_script].ps1
 + CategoryInfo : NotSpecified: ( [], PSSecurityException
 + FullyQualifiedErrorId : RuntimeException
```

To enable Powershell scripts to be executed, set the execution policy to Yes.

```
PS> set-executionpolicy remotesigned
Execution Policy Change
The execution policy helps protect you from scripts that you do not trust.
Changing the execution policy might expose you to the security risks
described in the about_Execution_Policies help topic. Do you want to change
the execution policy?
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): Y
PS>
```

Now your Powershell scripts will be executed.

13.3.8.5.5 curl.exe

Not all Windows platforms have the cURL executable installed, the easiest way to install and use cURL on Windows is as follows:

1. Download the package from https://curl.se/windows/ and unzip.
2. In the "bin" folder find "curl.exe" and move it to "C:\Windows\System32".

Then you will be able to use the curl command from the Windows Command Prompt or PowerShell console.

```
PS> curl.exe --version
curl 7.49.1 (x86_64-w64-mingw32) libcurl/7.49.1 OpenSSL/1.0.2h zlib/1.2.8
libidn/1.32 libssh2/1.7.0 nghttp2/1.12.0 librtmp/2.3
Protocols: dict file ftp ftps gopher http https imap imaps ldap ldaps pop3
pop3s rtmp rtsp scp sftp smtp smtps telnet tftp
Features: IDN IPv6 Largefile SSPI Kerberos SPNEGO NTLM SSL libz TLS-SRP HTTP2
Metalink
```

Invoking the curl or curl.exe from Powershell command line.

```
PS> Get-Command curl

CommandType Name ModuleName
----------- ---- ----------
Alias curl -> Invoke-WebRequest


PS> Get-Command curl.exe


CommandType Name ModuleName
----------- ---- ----------
Application curl.exe
```

If the alias curl name is to the Invoke-WebRequest, you will need to use the curl.exe command explicitly or remove the alias.

```
PS> Remove-item alias:curl
```

Verify that curl and/or curl.exe work from the respective Powershell environment:

```
PS> curl.exe --version
curl 7.49.1 (x86_64-w64-mingw32) ...

PS> curl --version
curl 7.49.1 (x86_64-w64-mingw32) ...
```

13.3.8.6 JSON parsing

13.3.8.6.1 Unix/Linux/Mac shell

Unix/Linux tools come natively with a host of shell utilities that one can use for parsing out the desired name/value pairs. Tools include sed, awk, cut, tr, and grep, to name a few. System administrators use these utilities frequently and may be able to assist with the methods for parsing JSON strings. For more information please refer to Parsing JSON with UNIX tools[785] and Extract a JSON value from a BASH script[786]

785 https://stackoverflow.com/questions/1955505/parsing-json-with-unix-tools
786 https://gist.github.com/cjus/1047794

13.3.8.6.1.1 Basic awk and sed parsing

```
json='{"type":"OKResult","status":"OK","result":
{"type":"Job","reference":"JOB-53","namespace":null,"name":null,"actionType":"DB_SYNC
","target":"ORACLE_DB_CONTAINER-9","targetObjectType":"OracleDatabaseContainer","jobS
tate":"RUNNING","startTime":"2016-08-12T19:58:59.811Z","updateTime":"2016-08-12T19:58
:59.828Z","suspendable":true,"cancelable":true,"queued":false,"user":"USER-2","emailA
ddresses":null,"title":"Run SnapSync for database
\"VDPXDEV1\".","percentComplete":0.0,"targetName":"Oracle_Source/VDPXDEV1","events":
[{"type":"JobEvent","timestamp":"2016-08-12T19:58:59.840Z","state":null,"percentCompl
ete":0.0,"messageCode":"event.job.started","messageDetails":"DB_SYNC job started for
\"Oracle_Source/
VDPXDEV1\".","messageAction":null,"messageCommandOutput":null,"diagnoses":
[],"eventType":"INFO"}],"parentActionState":"WAITING","parentAction":"ACTION-238"},"j
ob":null,"action":null}' echo $json | sed -e 's/[{}]/''/g' | awk -v RS=',' -F:
'{print $1 $2}'"type""OKResult""status""OK""result""type""reference""JOB-53""namespac
e"null"name"null"actionType""DB_SYNC""target""ORACLE_DB_CONTAINER-9""targetObjectType
""OracleDatabaseContainer""jobState""RUNNING""startTime""2016-08-12T19"updateTime""20
16-08-12T19"suspendable"true"cancelable"true"queued"false"user""USER-2""emailAddresse
s"null"title""Run SnapSync for database \"VDPXDEV1\"""percentComplete"0.0"targetName
""Oracle_Source/VDPXDEV1""events"["type""timestamp""2016-08-12T19"state"null"percentC
omplete"0.0"messageCode""event.job.started""messageDetails""DB_SYNC job started for \"
Oracle_Source/VDPXDEV1\""messageAction"null"messageCommandOutput"null"diagnoses"[]"
eventType""INFO"]"parentActionState""WAITING""parentAction""ACTION-238""job"null"acti
on"null
```

Find jobState. Print the second argument, and remove the double-quotes.

```
echo $json | sed -e 's/[{}]/''/g' | sed s/\"//g | awk -v RS=',' -F: '$1=="jobState"{p
rint $2}' RUNNING
```

The first sed removed the brackets and braces. The second sed removes the double-quotes. The awk command parses the line by comma delimiters and then parses each line by the semi-colon delimiter and if the first variable $1 is equal to the **jobState** value then print the second $2 variable.

If the results contain an array of values, then you need to loop through each set and parse out the desired value. For example,

```
json='{"type":"ListResult","status":"OK","result":
[{"type":"WindowsHostEnvironment","reference":"WINDOWS_HOST_ENVIRONMENT-1","namespace
":null,"name":"Window
Target","description":"","primaryUser":"HOST_USER-1","enabled":false,"host":"WINDOWS_
HOST-1","proxy":null},
{"type":"UnixHostEnvironment","reference":"UNIX_HOST_ENVIRONMENT-3","namespace":null,
"name":"Oracle
Target","description":"","primaryUser":"HOST_USER-3","enabled":true,"host":"UNIX_HOST
```

```
-3","aseHostEnvironmentParameters":null}],"job":null,"action":null,"total":2,"overflo
w":false}'
```

Parse out array object into separate lines

```
SOURCE_ENV="Oracle Target"lines=`echo ${json} | cut -d "[" -f2 | cut -d "]" -f1 | awk
-v RS='},{}' -F: '{print $0}' `while read -r linedo    #echo "Processing $line"    #e
cho $line | sed -e 's/[{}]/''/g' | sed s/\"//g | awk -v RS=',' -F: '$1=="name"{print
$2}'    TMPNAME=`echo $line | sed -e 's/[{}]/''/g' | sed s/\"//g | awk -v RS=',' -F:
'$1=="name"{print $2}' `    #echo "Name: |${TMPNAME}| |${SOURCE_ENV}|"    if [[ "$
{TMPNAME}" == "${SOURCE_ENV}" ]]    then        echo $line | sed -e 's/[{}]/''/g' |
sed s/\"//g | awk -v RS=',' -F: '$1=="primaryUser"{print $2}'        PRI_USER=`echo
$line | sed -e 's/[{}]/''/g' | sed s/\"//g | awk -v RS=',' -F: '$1=="primaryUser"{pri
nt $2}' `        break    fidone <<< "$(echo -e "$lines")" echo "primaryUser
reference: ${PRI_USER}"
```

Output:

```
primaryUser reference: HOST_USER-3
```

The above methods will be used within the sample scripts since they use the native Linux tools. They typically do not require you to load extra packages or libraries onto the system.

There are a number of open-source utilities designed to simplify the parsing of JSON, such as *jsawk* and *jq*.

jsawk

Linux:

- jswak[787]

Mac:

- jswak on MAC[788]

jq

Reference- https://stedolan.github.io/jq/

64-bit system:

- wget https://github.com/stedolan/jq/releases/download/jq-1.6/jq-linux64
- chmod +x ./jq
- sudo cp jq /usr/local/bin

Older versions:

- Reference- https://stedolan.github.io/jq/download/

Another method is to use an existing programming language typically available with your native operating systems, such as Perl or Python.

787 https://github.com/micha/jsawk
788 http://macappstore.org/jsawk/

```
$ which perl/usr/bin/perl$ which python/usr/bin/python
```

Example: Use Python to pretty format the JSON data string.

Pretty JSON using Python ...

```
json='{"type":"OKResult","status":"OK","result":
{"type":"SystemInfo","productType":"standard","productName":"Delphix
Engine","buildTitle":"Delphix Engine
5.1.1.0","buildTimestamp":"20160721T07:23:41.000Z","buildVersion":
{"type":"VersionInfo","major":5,"minor":1,"micro":1,"patch":0},"configured":true,"ena
bedFeatures":["XPP","MSSQLHOOKS"],"apiVersion":
{"type":"APIVersion","major":1,"minor":8,"micro":0},"banner":null,"locals":
["enUS"],"currentLocale":"enUS","hostname":"Delphix5110HWv8","sshPublicKey":"ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDOsrp7Aj6hFQh9yBq7273B+qtPKmCu1B18nPvr08yjt/
IZeM4qKk7caxExQS9rpfU8AWoT7e8ESV7NkBmUzOHrHnLsuJtxPqeYoqeMubVxYjJuxlH368sZuYsnB04KM0m
i39e15lxVGvxQk9tyMpl7gs7cXRz1k6puncyiczU/
axGq7ALHU2uyQoVmlPasuHJbq23d21VAYLuscbtgpZLAFlR8eQH5Xqaa0RT+aQJ6B1ihZ7S0ZN914M2gZHHNY
cSGDWZHwUnBGttnxx1ofRcyN4/qwT5iHq5kjApjSaNgSAU0ExqDHiqgTq0wttf5nltCqGMTFR7XY38HiNq+
+atDroot@Delphix5110HWv8\n","memorySize":8.58107904E9,"platform":"VMware with BIOS
date 05/20/2014","uuid":"564d7e1df4cb-f91098fd348d74817683","processors":
[{"type":"CPUInfo","speed":2.5E9,"cores":1}],"storageUsed":2.158171648E9,"storageTota
l":2.0673724416E10,"installationTime":"2016-07-27T13:28:46.000Z"},"job":null,"action"
:null}'
```

Pipe the JSON data to Python programming language to pretty up the format the output for the $json string/data.

```
$ echo $json | python -mjson.tool{    "action": null,     "job": null,     "result":
{        "apiVersion": {            "major": 1,             "micro": 0,             "
minor": 8,             "type": "APIVersion"        },         "banner": null,
        "buildTimestamp": "20160721T07:23:41.000Z",         "buildTitle": "Delphix
Engine 5.1.1.0",         "buildVersion": {            "major": 5,             "micro":
 1,             "minor": 1,             "patch": 0,             "type": "VersionInfo"
       },         "configured": true,         "currentLocale": "enUS",         "enabe
dFeatures": [            "XPP",             "MSSQLHOOKS"    ],     "hostname":
"Delphix5110HWv8",     "installationTime": "2016-07-27T13:28:46.000Z",     "locals":
[        "enUS"    ],     "memorySize": 8581079040.0,     "platform": "VMware with
BIOS date 05/20/2014",     "processors": [        {            "cores": 1,
            "speed": 2500000000.0,             "type": "CPUInfo"        }    ],     "
productName": "Delphix Engine",     "productType": "standard",     "sshPublicKey":
"ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDOsrp7Aj6hFQh9yBq7273B+qtPKmCu1B18nPvr08yjt/
IZeM4qKk7caxExQS9rpfU8AWoT7e8ESV7NkBmUzOHrHnLsuJtxPqeYoqeMubVxYjJuxlH368sZuYsnB04KM0m
i39e15lxVGvxQk9tyMpl7gs7cXRz1k6puncyiczU/
axGq7ALHU2uyQoVmlPasuHJbq23d21VAYLuscbtgpZLAFlR8eQH5Xqaa0RT+aQJ6B1ihZ7S0ZN914M2gZHHNY
cSGDWZHwUnBGttnxx1ofRcyN4/qwT5iHq5kjApjSaNgSAU0ExqDHiqgTq0wttf5nltCqGMTFR7XY38HiNq+
+atDroot@Delphix5110HWv8\n",    "storageTotal": 20673724416.0,     "storageUsed":
2158171648.0,     "type": "SystemInfo",     "uuid": "564d7e1df4cb-
f91098fd348d74817683"    },     "status": "OK",     "type": "OKResult"}
```

13.3.8.6.2 jq parser

The "jq" command line parser is available on Unix, Linux, Mac, and Windows platforms. Typically, for Windows, the built-in ConvertFrom/To-Json object parser will be used. "jq" is being included in most native Linux distributions and is easy to install on the Mac OS.

References:

Parsing JSON with jq[789]

Installation FAQ[790]

Mac Installation[791]

Example:

```
json='{"type":"ListResult","status":"OK","result":
[{"type":"OracleLinkedSource","reference":"ORACLE_LINKED_SOURCE-52","namespace":null,"name":"DPXDEV0
1","description":null,"virtual":false,"restoration":false,"staging":false,"container":"ORACLE_DB_CON
TAINER-120","config":"ORACLE_SINGLE_CONFIG-40","status":"DEFAULT","runtime":
{"type":"OracleSourceRuntime","status":"RUNNING","accessible":true,"databaseSize":2.409529344E9,"not
AccessibleReason":null,"databaseMode":"READ_WRITE","lastNonLoggedLocation":"0","activeInstances":
[{"type":"OracleActiveInstance","instanceNumber":1,"instanceName":"DPXDEV01","hostName":"linuxtarget
.delphix.local"}],"databaseStats":null,"bctEnabled":true,"racEnabled":null,"dnfsEnabled":false,"arch
ivelogEnabled":null},"backupLevelEnabled":false,"rmanChannels":2,"filesPerSet":5,"checkLogical":fals
e,"externalFilePath":null,"encryptedLinkingEnabled":false,"compressedLinkingEnabled":true,"bandwidth
Limit":0,"numberOfConnections":1,"enabled":true,"preScript":"","postScript":"","role":"PRIMARY"},
{"type":"OracleVirtualSource","reference":"ORACLE_VIRTUAL_SOURCE-25","namespace":null,"name":"VBITT"
,"description":null,"virtual":true,"restoration":false,"staging":false,"container":"ORACLE_DB_CONTAI
NER-121","config":"ORACLE_SINGLE_CONFIG-47","status":"DEFAULT","runtime":
{"type":"OracleSourceRuntime","status":"RUNNING","accessible":true,"databaseSize":2.410053632E9,"not
AccessibleReason":null,"databaseMode":"READ_WRITE","lastNonLoggedLocation":"0","activeInstances":
[{"type":"OracleActiveInstance","instanceNumber":1,"instanceName":"VBITT","hostName":"linuxtarget.de
lphix.local"}],"databaseStats":[{"type":"OracleDatabaseStatsSection","sectionName":"Open
Transactions","columnHeaders":["Transaction Count"],"rowValues":
[{"type":"OracleDatabaseStatistic","statisticValues":["0"]}]},
{"type":"OracleDatabaseStatsSection","sectionName":"Session Statistics","columnHeaders":["Current
Session","Total Session","High Watermark"],"rowValues":
[{"type":"OracleDatabaseStatistic","statisticValues":["2","46","5"]}]},
{"type":"OracleDatabaseStatsSection","sectionName":"Top Wait Events","columnHeaders":["Event","Wait
Count","Total Wait Time (s)"],"rowValues":[{"type":"OracleDatabaseStatistic","statisticValues":
["Disk file operations I/O","13","13"]},{"type":"OracleDatabaseStatistic","statisticValues":["log
file sequential read","11","12"]},{"type":"OracleDatabaseStatistic","statisticValues":["control file
parallel write","8","8"]},{"type":"OracleDatabaseStatistic","statisticValues":["control file
sequential read","6","3"]},{"type":"OracleDatabaseStatistic","statisticValues":["ARCH wait for
process start 3","2","2"]},{"type":"OracleDatabaseStatistic","statisticValues":["db file sequential
read","9","1"]},{"type":"OracleDatabaseStatistic","statisticValues":["rdbms ipc reply","1","1"]},
{"type":"OracleDatabaseStatistic","statisticValues":["JS coord start wait","1","1"]},
{"type":"OracleDatabaseStatistic","statisticValues":["os thread startup","2","0"]},
{"type":"OracleDatabaseStatistic","statisticValues":["Parameter File I/O","1","0"]}]},
{"type":"OracleDatabaseStatsSection","sectionName":"Top SQL by CPU","columnHeaders":["Percentage of
Load","SQL Statement"],"rowValues":
[]}],"bctEnabled":false,"racEnabled":null,"dnfsEnabled":false,"archivelogEnabled":null},"operations"
:{"type":"VirtualSourceOperations","configureClone":[],"preRefresh":[],"postRefresh":
[]},"mountBase":"/mnt/provision","fileMappingRules":null,"manualProvisioning":null,"configParams":
{"memory_target":"1191182336","processes":"150","log_archive_dest_1":"location=/mnt/provision/VBITT/
archive/
MANDATORY","_omf":"ENABLED","filesystemio_options":"setall","compatible":"11.2.0.4.0","audit_trail":
"NONE","remote_login_passwordfile":"EXCLUSIVE","open_cursors":"300","audit_sys_operations":"FALSE"},
```

789 http://www.compciv.org/recipes/cli/jq-for-parsing-json/
790 https://github.com/stedolan/jq/wiki/FAQ#installation
791 http://macappstore.org/jq/

```
"configTemplate":null,"nodeListenerList":
[],"enabled":true,"role":"PRIMARY"}],"job":null,"action":null,"total":2,"overflow":false}'
```

We have a very big JSON string above. Let's perform some basic jq parsing.

1. Pipe JSON string into jq command line parser.

```
ActionScript

echo $json | jq '.'
```

2. The output is a pretty human-readable JSON formatted string.
3. Get the first-level status value (. . . ,"status":"OK", . . .)

```
ActionScript

echo $json | jq '.status'"OK"
```

4. Get raw values (not quoted).

```
ActionScript

echo $json | jq --raw-output '.status'OK
```

5. Get a number of rows returned for the type equal to "ListResult" API returned request.

```
ActionScript

echo $json | jq --raw-output '.total'2
```

6. Get the first result set.

```
ActionScript

echo $json | jq '.result[0] '{    "type": "OracleLinkedSource",    "reference":
"ORACLE_LINKED_SOURCE-52",    "namespace": null,    "name": "DPXDEV01",    "des
cription": null,    "virtual": false,    "restoration": false,    "staging":
false,    "container": "ORACLE_DB_CONTAINER-120",    "config":
"ORACLE_SINGLE_CONFIG-40",    "status": "DEFAULT",    "runtime": {        "type
": "OracleSourceRuntime",        "status": "RUNNING",        "accessible": true,
        "databaseSize": 2409529344,        "notAccessibleReason": null,        "
databaseMode": "READ_WRITE",        "lastNonLoggedLocation": "0",        "activ
eInstances": [            {                "type": "OracleActiveInstance",
           "instanceNumber": 1,                "instanceName": "DPXDEV01",
           "hostName": "linuxtarget.delphix.local"            }        ],
  "databaseStats": null,        "bctEnabled": true,        "racEnabled": null,
       "dnfsEnabled": false,        "archivelogEnabled": null    },    "backupL
evelEnabled": false,    "rmanChannels": 2,    "filesPerSet": 5,    "checkLogica
```

```
l": false,    "externalFilePath": null,    "encryptedLinkingEnabled": false,
 "compressedLinkingEnabled": true,    "bandwidthLimit": 0,    "numberOfConnecti
ons": 1,    "enabled": true,    "preScript": "",    "postScript": "",    "role":
 "PRIMARY"}
```

7. Get the first result set name value.

```
ActionScript

echo $json | jq --raw-output '.result[0].name'DPXDEV01
```

8. Get first result set reference value.

```
ActionScript

echo $json | jq --raw-output '.result[0].reference'
```

9. Get first result set name=value pairs.

```
ActionScript

echo $json | jq '.result[0]' | jq -r "to_entries|map(\"(.key)=(.value|tostring)
\")|.[]" | grep container container=ORACLE_DB_CONTAINER-120
```

10. Get ALL result sets name values.

```
ActionScript

echo $json | jq '.result[].name'"DPXDEV01""VBITT"
```

11. Get ALL result sets "reference" and "container" values.

```
ActionScript

echo $json | jq '.result[].reference,.result[].container'"ORACLE_LINKED_SOURCE-
52""ORACLE_VIRTUAL_SOURCE-25""ORACLE_DB_CONTAINER-120""ORACLE_DB_CONTAINER-121"
```

12. Now, let's scan ALL result sets for a conditional match and return a related value.

```
echo $json | jq --raw-output '.result[] | select(.name=="VBITT") | .container'
 ORACLE_DB_CONTAINER-121echo $json | jq --raw-output '.result[] |
select(.name=="VBITT") | .reference' ORACLE_VIRTUAL_SOURCE-25echo $json | jq --raw-
output '.result[] | select(.name=="VBITT") | .container, .reference'ORACLE_DB_CONTAIN
ER-121ORACLE_VIRTUAL_SOURCE-25
```

This is the typical usage for Delphix, where the human-readable name is provided and we need to look up the object reference, container, status, etc. for the respective name. Some object references are based on expressions such as "and" or "or" conditions.

```
echo $json | jq --raw-output '.result[] |
select(.environment=="UNIX_HOST_ENVIRONMENT-9" and .name=="/u02/ora/app/product/
11.2.0/dbhome_1" ) | .reference '
```

In this case, the jq select command has an "and" condition in order to correctly identify the target result object index. This is important for getting the correct and single return value for `| .reference`, since there might be more than one instance within the environment.

For a working example of using the jq JSON parser, see the VDB Init using jq command-line JSON Parser use case, **Filename: vdb_init.sh**. A version of all the Unix/Linux/Mac shell scripts exists within the code provided. It contains the ***_jq.sh** within the filename.

13.3.8.6.3 PowerShell

Starting with Powershell 3.0, there are ConvertFrom-Json and ConvertTo-Json modules/commands to parse the JSON string data to/from objects. If you are stuck with Powershell 2.x., the next section provides similar functions as a method of working with JSON strings.

> These 2.x functions are not 100% the same as the Powershell 3.0 ConvertFrom-Json/ConvertTo-Json modules.

13.3.8.6.3.1 PowerShell 2 example

Filename: parse_2.0.ps1

For Powershell 2.0, there are no JSON-provided functions or commands, so the following will serialize the JSON data to a serialized array.

```
function ConvertTo-Json20([object] $item){    add-type -assembly
system.web.extensions    $ps_js=new-object
system.web.script.serialization.javascriptSerializer    return
 $ps_js.Serialize($item)}function ConvertFrom-Json20([object] $item){     add-type
-assembly system.web.extensions    $ps_js=new-object
system.web.script.serialization.javascriptSerializer    # The comma operator is the
array construction operator in PowerShell    return ,$ps_js.DeserializeObject($item)}
```

Use the JSON from the system API Call.

```
$json='{"type":"OKResult","status":"OK","result":
{"type":"SystemInfo","productType":"standard","productName":"Delphix
Engine","buildTitle":"Delphix Engine
```

```
5.1.1.0","buildTimestamp":"20160721T07:23:41.000Z","buildVersion":
{"type":"VersionInfo","major":5,"minor":1,"micro":1,"patch":0},"configured":true,"ena
bedFeatures":["XPP","MSSQLHOOKS"],"apiVersion":
{"type":"APIVersion","major":1,"minor":8,"micro":0},"banner":null,"locals":
["enUS"],"currentLocale":"enUS","hostname":"Delphix5110HWv8","sshPublicKey":"ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDOsrp7Aj6hFQh9yBq7273B+qtPKmCu1B18nPvr08yjt/
IZeM4qKk7caxExQS9rpfU8AWoT7e8ESV7NkBmUzOHrHnLsuJtxPqeYoqeMubVxYjJuxlH368sZuYsnB04KM0m
i39e15lxVGvxQk9tyMpl7gs7cXRz1k6puncyiczU/
axGq7ALHU2uyQoVmlPasuHJbq23d21VAYLuscbtgpZLAFlR8eQH5Xqaa0RT+aQJ6B1ihZ7S0ZN914M2gZHHNY
cSGDWZHwUnBGttnxx1ofRcyN4/qwT5iHq5kjApjSaNgSAU0ExqDHiqgTq0wttf5nltCqGMTFR7XY38HiNq+
+atDroot@Delphix5110HWv8\n","memorySize":8.58107904E9,"platform":"VMware with BIOS
date 05/20/2014","uuid":"564d7e1df4cb-f91098fd348d74817683","processors":
[{"type":"CPUInfo","speed":2.5E9,"cores":1}],"storageUsed":2.158171648E9,"storageTota
l":2.0673724416E10,"installationTime":"2016-07-27T13:28:46.000Z"},"job":null,"action"
:null}'
```

Convert the JSON string.

> The job and action are null values.

```
PS> $o = ConvertFrom-Json20 $jsonPS> $o Key                Value---
                -----
type            OKResultstatus            OKresult            {[type, SystemInfo],
[productType, standard], [productNa...jobaction
```

Extract the result JSON string array.

```
PS> $a = $o.resultPS> $a Key                    Value---                    -----
type                SystemInfoproductType            standardproductName            D
elphix EnginebuildTitle            Delphix Engine 5.1.1.0buildTimestamp        201607
21T07:23:41.000ZbuildVersion        {[type, VersionInfo], [major, 5], [minor, 1],
[micro, 1]...configured            TrueenabedFeatures        {XPP, MSSQLHOOKS}
apiVersion            {[type, APIVersion], [major, 1], [minor, 8], [micro, 0]}
bannerlocals                {enUS}
currentLocale        enUShostname            Delphix5110HWv8sshPublicKey        ssh-
rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDOsrp7Aj6hFQh9yBq7...memorySize            8581079040pl
atform            VMware with BIOS date 05/20/2014uuid                564d7e1df4cb-
f91098fd348d74817683processors            {System.Collections.Generic.Dictionary`2[Sy
stem.String,S...storageUsed            2158171648storageTotal        20673724416insta
llationTime    2016-07-27T13:28:46.000Z
```

Same output as above.

```
PS> foreach ($element in $a) {$element}PS> $a.typeSystemInfoPS> $a.buildTitleDelphix
Engine 5.1.1.0PS> $a.hostnameDelphix5110HWv8
```

Extract the result.buildVersion object.

```
PS> $a1 = $o.result.buildVersionPS> $a1Key                    Value---
                     -----
type                 VersionInfomajor                5minor                1micro
           1patch                0PS> $a1.major 5
```

Extract the result.processors array collection.

```
PS> $b = $o.result.processors PS> $b Key                       Value---
                        -----
    type                     CPUInfospeed                    2500000000cores
          1PS> $a -is [Array] FalsePS> $a -is [Object] TruePS> $b -is [Array] True
```

Convert Array Collection to Object.

```
PS> $b1 = $b | Select-Object PS> $b1 Key                    Value---
                     ------
type                 CPUInfospeed                2500000000cores                1PS>
  $b1.type CPUInfoPS> $b1.speed2500000000
```

13.3.8.6.4 PowerShell 3 or greater example

Starting with Powershell 3.0, there is are ConvertFrom-Json and ConvertTo-Json commands to parse the JSON data to/from objects.

Reference:

- ConvertFrom-Json[792]
- ConvertTo-Json[793]

`$o = $json | ConvertFrom-Json`

There are a number of tutorials and functional examples on the web. Below is an excerpt from the Powershell introduction video for Linux[794] / Mac Open Source announcement.

Powershell JSON ConvertTo-Json and Python Example 15:55 through 21:16

The concept is straightforward:

- The `ConvertFrom-Json` JSON string is converted into a Powershell object that you can reference directly.
- The `ConvertTo-Json` takes the JSON object and converts it to a string.

792 https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/convertfrom-json?view=powershell-7.1

793 https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/convertto-json?view=powershell-7.1

794 https://www.youtube.com/watch?v=2WZwv7TxqZ0&feature=youtu.be

13.3.8.6.5 JSON parsing from within programming languages

Most programming languages provide their own libraries, functions, and methods for parsing JSON data strings into objects/hashes/arrays/xml that the native programming language can easily process.

13.3.8.7 API use case commands and scripts

13.3.8.7.1 Sample script parsers

The Delphix Use Cases scripts provided use the native operating system "curl" command and then a JSON parser, depending on the engine on which you are runnning the scripts.

For Unix/Linux/Mac, the scripts provided use both native shell commands and/or the jq parser program commands. Subroutines have been provided for both methods:

```
Native Shell commands:        parseJSON_subroutines.sh
jq Parser commands:            jqJSON_subroutines.sh
```

For Windows, the scripts are for Powershell 2.0 and utilitze the custom `ConvertFrom-Json20` and `ConvertTo-Json20` functions provided. As noted, with Powershell 3.0, there are `ConvertFrom-Json` and `ConvertTo-Json` command-lets provided by Powershell.

13.3.8.7.2 Using the jq parser

These are some of the jq commands used within the scripts. The first is a shell script subroutine which is used for finding and returning a single item value.

Filename: jqJSON_subroutines.sh

Subroutines

This code requires the jq Linux/Mac JSON parser program

```
jqParse() {
    STR=$1                    # json string
    FND=$2                    # name to find
    RESULTS=""                # returned name value
    RESULTS=`echo $STR | jq --raw-output '.'"$FND"'''`
    #echo "Results: ${RESULTS}"
    if [ "${FND}" == "status" ] && [ "${RESULTS}" != "OK" ]
    then
        echo "Error: Invalid Satus, please check code ... ${STR}"
        exit 1;
    elif [ "${RESULTS}" == "" ]
    then
        echo "Error: No Results ${FND}, please check code ... ${STR}"
        exit 1;
```

```
    fi
    echo "${RESULTS}"
}
```

The subroutine is called from the shell script to return values based on the key (or name) value provided.

Usage – After every curl command, check that the returned status value is "OK".

```
RESULTS=$( jqParse "${STATUS}" "status" )
```

Call the jqParse subroutine, where

- ${STATUS} is the returned JSON string from the curl command, and
- the value we want returned is where the name/key is equal to "status"

Usage – Get a single value within the returned nested result object:

```
JOBSTATE=$( jqParse "${JOB_STATUS}" "result.jobState" )
```

Call the jqParse subroutine, where

- ${JOB_STATUS} is the returned JSON string from the cURL command, and
- the value we want returned is where the name/key is equal to "jobState" with the nested ".result" object.

Usage – Find name/value result object and return another value within the select result object:

Use jq to parse out container reference for name of $SOURCE_SID ...

```
CONTAINER_REFERENCE=`echo ${STATUS} | jq --raw-output '.result[] | select(.name=="'"$
{SOURCE_SID}"'") | .reference '`
```

where

- ${STATUS} is the returned JSON string from the cURL command, and
- the value we want returned is based on the selected nested result object where the .result[].name is equal to "${SOURCE_SID}" and return the .reference value for the selected result object.

13.3.8.7.3 Delphix engine use cases

13.3.8.7.4 Delphix user session timeout

Some activities can take longer than the default 30 minute session timeout value. Therefore, the following script allows you to change the timeout value using the RESTFul API. As always, you can change it easily through the CLI.

> This code is the first example showing how object references are used for input (either JSON or URL) into API calls. The name will be the DE_USER variable value delphix_admin. The object reference that the code identifies is USER-2, which in this case is passed into the API URL to update the user parameters passed via the JSON string.

Filename: user_timeout.sh

Edit the file to update the parameters as required for your environment.

```
#####################
#    DELPHIX CORP    #
#####################
#Parameter Initialization
DMIP=172.16.160.195
#DMPORT=8282
DMUSER=delphix_admin
DMPASS=delphix
COOKIE="~/cookies.txt"
COOKIE=`eval echo $COOKIE`
CONTENT_TYPE="Content-Type: application/json"
BaseURL="http://${DMIP}/resources/json/delphix"
#
Required for user timeout ...
#
DE_USER="delphix_admin" # Delphix Engine User
DE_TIMEOUT=120 # Timeout integer in minutes
##########################################
#    NO CHANGES REQUIRED BELOW THIS POINT    #
##########################################
```

Sample Output

```
$ ./user_timeout.sh # or ./user_timeout_jq.sh
Authenticating on http://172.16.160.195/resources/json/delphix
Session and Login Successful ...
user reference: USER-2
Update delphix_admin session timeout value to 120 minutes ...
Returned JSON: {"type":"OKResult","status":"OK","result":"","job":null,"action":"ACTI
ON-423"}
Results: OK
Done ...
$
```

13.3.8.7.5 VDB Init (start | stop | enable | disable | status | delete)

This script is used to start, stop, enable, disable, and delete a Delphix platform source object. Typically, this is done on a virtual databases (VDBs), but you can use it for dSources as well.

> ⚠ The `vdb_init.sh` and `vdb_operations.sh` require the "jq" command line json parser.

Filename: vdb_init.sh

The `vdb_init.sh` supports the start, stop, enable, disable, status, and delete command line options.

```
$ ./vdb_init.sh something VBITT
. . .
Unknown option (start | stop | enable | disable | status | delete): something
Exiting ...
$ ./vdb_init.sh status VBITT
database container reference: ORACLE_DB_CONTAINER-121
source reference: ORACLE_VIRTUAL_SOURCE-25
Runtime Status: "INACTIVE"
Enabled: true
Done ...
```

```
$ ./vdb_init.sh start VBITT
database container reference: ORACLE_DB_CONTAINER-121
source reference: ORACLE_VIRTUAL_SOURCE-25
Job: JOB-894
Current status as of Wed Sep 7 16:04:04 EDT 2016 : RUNNING 0% Completed
Current status as of Wed Sep 7 16:04:14 EDT 2016 : RUNNING 25% Completed
Current status as of Wed Sep 7 16:04:24 EDT 2016 : RUNNING 45% Completed
Job: JOB-894 COMPLETED 100% Completed ...
Done ...
```

```
$ ./vdb_init.sh delete VBITT
database container reference: ORACLE_DB_CONTAINER-123
source reference: ORACLE_VIRTUAL_SOURCE-27
vendor source: OracleVirtualSource
delete parameters type: OracleDeleteParameters
Job: JOB-927
Current status as of Sat Sep 10 12:55:32 EDT 2016 : RUNNING 0% Completed
Current status as of Sat Sep 10 12:55:32 EDT 2016 : RUNNING 0% Completed
Job: JOB-927 COMPLETED 100% Completed ...
Done ...
```

13.3.8.7.6 VDB operations (sync, refresh, rollback)

This script is used to perform a sync (snapshot), refresh, or rollback (reset) on the Delphix Engine source object. All these work on a virtual databases (VDBs), but only a sync operation can be used on dSources.

> ⚠ The `vdb_init.sh` and `vdb_operations.sh` require the "jq" command line json parser.

Filename: vdb_operations.sh

```
$ ./vdb_operations.sh sync VBITT
Session and Login Successful ...
database container reference: ORACLE_DB_CONTAINER-131
provision source container: ORACLE_DB_CONTAINER-129
json> {
    "type": "OracleSyncParameters"
}
Job: JOB-998
Current status as of Wed Sep 14 17:05:00 EDT 2016 : RUNNING 0% Completed
```

```
Current status as of Wed Sep 14 17:05:24 EDT 2016 : RUNNING 97% Completed
Job: JOB-998 COMPLETED 100% Completed ...
Done ...
```

Rollback rewinds the virtual database back to the last point in time within the source TimeFlow.

```
$ ./vdb_operations.sh rollback VBITT
Session and Login Successful ...
database container reference: ORACLE_DB_CONTAINER-131
provision source container: ORACLE_DB_CONTAINER-129
json> {
    "type": "OracleRollbackParameters",
    "timeflowPointParameters": {
        "type": "TimeflowPointSemantic",
        "container": "ORACLE_DB_CONTAINER-131"
    }
}
Job: JOB-1000
Current status as of Wed Sep 14 17:06:33 EDT 2016 : RUNNING 0% Completed
Current status as of Wed Sep 14 17:06:43 EDT 2016 : RUNNING 0% Completed
Current status as of Wed Sep 14 17:06:53 EDT 2016 : RUNNING 34% Completed
Current status as of Wed Sep 14 17:07:03 EDT 2016 : RUNNING 34% Completed
Current status as of Wed Sep 14 17:07:13 EDT 2016 : RUNNING 58% Completed
Current status as of Wed Sep 14 17:07:23 EDT 2016 : RUNNING 67% Completed
Current status as of Wed Sep 14 17:07:33 EDT 2016 : RUNNING 70% Completed
```

```
Current status as of Wed Sep 14 17:07:43 EDT 2016 : RUNNING 70% Completed
Current status as of Wed Sep 14 17:07:53 EDT 2016 : RUNNING 71% Completed
Current status as of Wed Sep 14 17:08:03 EDT 2016 : RUNNING 72% Completed
Current status as of Wed Sep 14 17:08:13 EDT 2016 : RUNNING 73% Completed
Current status as of Wed Sep 14 17:08:23 EDT 2016 : RUNNING 94% Completed
Current status as of Wed Sep 14 17:08:33 EDT 2016 : RUNNING 96% Completed
Current status as of Wed Sep 14 17:08:43 EDT 2016 : RUNNING 96% Completed
Current status as of Wed Sep 14 17:08:53 EDT 2016 : RUNNING 96% Completed
Job: JOB-1000 COMPLETED 100% Completed ...
Done ...
```

Refresh recreates the virtual database to the last point in time in the respective parent, provision source TimeFlow.

```
$ ./vdb_operations.sh refresh VBITT
Session and Login Successful ...
database container reference: ORACLE_DB_CONTAINER-131
provision source container: ORACLE_DB_CONTAINER-129
json> {
    "type": "OracleRefreshParameters",
    "timeflowPointParameters": {
        "type": "TimeflowPointSemantic",
        "container": "ORACLE_DB_CONTAINER-129"
    }
}
Job: JOB-1005
Current status as of Wed Sep 14 17:10:11 EDT 2016 : RUNNING 0% Completed
Current status as of Wed Sep 14 17:10:21 EDT 2016 : RUNNING 0% Completed
Current status as of Wed Sep 14 17:10:31 EDT 2016 : RUNNING 34% Completed
Current status as of Wed Sep 14 17:10:41 EDT 2016 : RUNNING 35% Completed
Current status as of Wed Sep 14 17:10:51 EDT 2016 : RUNNING 70% Completed
Current status as of Wed Sep 14 17:11:01 EDT 2016 : RUNNING 70% Completed
Current status as of Wed Sep 14 17:11:20 EDT 2016 : RUNNING 72% Completed
Current status as of Wed Sep 14 17:11:31 EDT 2016 : RUNNING 73% Completed
Current status as of Wed Sep 14 17:11:41 EDT 2016 : RUNNING 73% Completed
Job: JOB-1005 COMPLETED 100% Completed ...
Done ...
```

13.3.8.7.7 Delphix self-service use cases for APIs

13.3.8.7.7.1 Create Delphix self-service template

> ⚠ Jet Stream is now known as Delphix Self-Service.

Filename: jetstream_template.sh or *jetstream_template_jq.sh*

Edit the file to update the parameters as required for your environment.

Required for Delphix Self-Service Template ...

```
TPL_NAME="jstpl"            # JetStream Template Name
DATASOURCE_NAME="jsds"        # JetStream Data Source Name
DATASOURCE_VDB="VBITT"        # JetStream Data Source VDB or dSource
```

Sample Output

```
$ ./jetstream_template.sh# or ./Jetstream_template_jq.sh
Authenticating on http://172.16.160.195/resources/json/delphix
Session and Login Successful ...
Getting Database Container Reference Value ...
container reference: ORACLE_DB_CONTAINER-45
Create JetStream Template jstpl with Data Source DB VBITT ...
Database: {"type":"OKResult","status":"OK","result":"JS_DATA_TEMPLATE-3","job":null,"
action":"ACTION-547"}


Done ... (no job required for this action)
```

13.3.8.7.7.2 Create Delphix self-service data container

Filename: jetstream_container.sh# or *jetstream_container_jq.sh*

Edit the file to update the parameters as required for your environment.

Required for Delphix Self-Service Container ...

```
TPL_NAME="jstpl"        # JetStream Template Name
DS_NAME="jsds"            # JetStream Data Source Name


DC_NAME="jsdc"            # JetStream Data Container Name
DC_VDB="VBITT2"            # JetStream Data Container VDB
```

Sample Output

```
$ ./jetstream_container.sh# or ./jetstream_container_jq.sh
Authenticating on http://172.16.160.195/resources/json/delphix
Session and Login Successful ...
Getting Database Container Reference Value ...
container reference: ORACLE_DB_CONTAINER-46
JetStream Data Template: JS_DATA_TEMPLATE-4
JetStream sourceDataLayout: JS_DATA_TEMPLATE-4
Create JetStream Container jsdc with Data Source DB VBITT2 ...
```

```
JetStream Data Container Creation Results: {"type":"OKResult","status":"OK","result":"
JS_DATA_CONTAINER-4","job":"JOB-240","action":"ACTION-569"}
Job: JOB-240
Current status as of Wed Aug 17 04:11:54 EDT 2016 : RUNNING 0.0% Completed
Current status as of Wed Aug 17 04:11:54 EDT 2016 : RUNNING 0.0% Completed
Current status as of Wed Aug 17 04:12:04 EDT 2016 : RUNNING 5.0% Completed
Current status as of Wed Aug 17 04:12:14 EDT 2016 : RUNNING 5.0% Completed
Current status as of Wed Aug 17 04:12:24 EDT 2016 : RUNNING 30.0% Completed
Current status as of Wed Aug 17 04:12:34 EDT 2016 : RUNNING 31.0% Completed
Current status as of Wed Aug 17 04:12:44 EDT 2016 : RUNNING 53.0% Completed
Current status as of Wed Aug 17 04:12:54 EDT 2016 : RUNNING 57.0% Completed
Current status as of Wed Aug 17 04:13:04 EDT 2016 : RUNNING 57.0% Completed
Current status as of Wed Aug 17 04:13:14 EDT 2016 : RUNNING 57.0% Completed
Current status as of Wed Aug 17 04:13:24 EDT 2016 : RUNNING 59.0% Completed
Current status as of Wed Aug 17 04:13:34 EDT 2016 : RUNNING 60.0% Completed
Current status as of Wed Aug 17 04:13:44 EDT 2016 : RUNNING 77.0% Completed
Current status as of Wed Aug 17 04:13:54 EDT 2016 : RUNNING 77.0% Completed
Current status as of Wed Aug 17 04:14:04 EDT 2016 : RUNNING 77.0% Completed
Current status as of Wed Aug 17 04:14:14 EDT 2016 : RUNNING 77.0% Completed
Job: JOB-240 COMPLETED 100.0% Completed ...


Done ...
```

13.3.8.7.7.3 Create Delphix self-service bookmark

Filename: jetstream_api_examples.txt (part 1)

Create Bookmark ...

Change parameters as required and desired.

```
curl -X POST -k --data @-
http://172.16.160.177/resources/json/delphix/jetstream/bookmark \
    -b cookies.txt -H "Content-Type: application/json" <<EOF
{
    "type": "JSBookmarkCreateParameters",
    "bookmark": {
        "type": "JSBookmark",
        "name": "aalen",
        "branch": "JS_BRANCH-5",
        "shared": false,
        "tags": [
            "A",
            "B",
            "C"
        ]
    },
    "timelinePointParameters": {
        "type": "JSTimelinePointLatestTimeInput",
        "sourceDataLayout": "JS_DATA_CONTAINER-2"
    }
```

```
}
EOF
{"type":"OKResult","status":"OK","result":"JS_BOOKMARK-5","job":"JOB-512","action":"A
CTION-921"}
```

> The timelinePointParameters type "JSTimelinePointLatestTimeInput" is the last point / latest time in the branch!

Filename: jetstream_bookmark.sh or *jetstream_bookmark_jq.sh*

Edit the file to update the parameters as required for your environment.

```
DT=`date '+%Y%m%d%H%M%S'`
```

Required for Delphix Self-Service Bookmark ...

```
JS_BRANCH="default"            # JetStream Branch
BM_NAME="aalen_${DT}"        # JetStream Bookmark Name appended timestamp
SHARED="false"                # Share Bookmark true/false
TAGS='"API","Created"'        # Tags Array Values
```

Sample Output

```
$ ./jetstream_bookmark.sh # or ./jetstream_bookmark_jq.sh
Authenticating on http://172.16.160.195/resources/json/delphix
Session and Login Successful ...
Getting Jetstream Branch Reference Value ...
branch reference: JS_BRANCH-7
dataLayout container reference: JS_DATA_CONTAINER-4
JetStream Bookmark Creation Results: {"type":"OKResult","status":"OK","result":"JS_BO
OKMARK-4","job":"JOB-251","action":"ACTION-591"}
Job: JOB-251
Current status as of Wed Aug 17 04:59:53 EDT 2016 : COMPLETED 100.0% Completed
Job: JOB-251 COMPLETED 100.0% Completed ...


Done ...
```

13.3.8.7.7.4 Delphix self-service refresh

Filename: jetstream_api_examples.txt (part 2)

Use CLI command to get Delphix Self-Service Container Reference

```
/jetstream/container/list

            ...
            "reference": " JS_DATA_CONTAINER-4 ",
            "namespace": null,
            "name": " jsdc ",
            ...
```

Refresh Container Information ...

```
=== POST /resources/json/delphix/jetstream/container/ JS_DATA_CONTAINER-4 /refresh


curl -X POST -k --data @http://172.16.160.177/resources/json/delphix/jetstream/
container/ JS_DATA_CONTAINER-4 /refresh \ -b cookies.txt -H "Content-Type:
application/json" <<EOF
{}
EOF


=== RESPONSE ===
{
    "type": "OKResult",
    "status": "OK",
    "result": "",
    "job": "JOB-514",
    "action": "ACTION-924"
}
=== END ===
```

Filename: jetstream_refresh.sh or *jetstream_refresh_jq.sh*

Edit the file to update the parameters as required for your environment.

Required for Delphix Self-Service Refresh ...

```
CONTAINER_NAME="jsdc"        # Jetstream Container Name
```

Sample Output

```
$ ./jetstream_refresh.sh # or ./jetstream_refresh_jq.sh
Authenticating on http://172.16.160.195/resources/json/delphix
Session and Login Successful ...
Getting Jetstream Container Reference Value ...
container reference: JS_DATA_CONTAINER-4
abitterman-mbpro:JetStream abitterman$ vi jetstream_refresh.sh
abitterman-mbpro:JetStream abitterman$ ./jetstream_refresh.sh
Authenticating on http://172.16.160.195/resources/json/delphix
Session and Login Successful ...
Getting Jetstream Container Reference Value ...
```

```
container reference: JS_DATA_CONTAINER-4
JetStream Refresh API Results: {"type":"OKResult","status":"OK","result":"","job":"JO
B-257","action":"ACTION-602"}
Job: JOB-257
Current status as of Wed Aug 17 05:13:15 EDT 2016 : RUNNING 2.0% Completed
Current status as of Wed Aug 17 05:13:15 EDT 2016 : RUNNING 2.0% Completed
Current status as of Wed Aug 17 05:13:25 EDT 2016 : RUNNING 5.0% Completed
Current status as of Wed Aug 17 05:13:35 EDT 2016 : RUNNING 5.0% Completed
Current status as of Wed Aug 17 05:13:45 EDT 2016 : RUNNING 30.0% Completed
Current status as of Wed Aug 17 05:13:55 EDT 2016 : RUNNING 42.0% Completed
Current status as of Wed Aug 17 05:14:05 EDT 2016 : RUNNING 55.0% Completed
Current status as of Wed Aug 17 05:14:15 EDT 2016 : RUNNING 58.0% Completed
Current status as of Wed Aug 17 05:14:25 EDT 2016 : RUNNING 58.0% Completed
Current status as of Wed Aug 17 05:14:35 EDT 2016 : RUNNING 58.0% Completed
Current status as of Wed Aug 17 05:14:45 EDT 2016 : RUNNING 60.0% Completed
Current status as of Wed Aug 17 05:14:55 EDT 2016 : RUNNING 62.0% Completed
Current status as of Wed Aug 17 05:15:05 EDT 2016 : RUNNING 77.0% Completed
Current status as of Wed Aug 17 05:15:15 EDT 2016 : RUNNING 77.0% Completed
Job: JOB-257 COMPLETED 100.0% Completed ...


Done ...
```

13.3.8.7.8 Masking use cases

13.3.8.7.8.1 Masking API client

The Continuous Compliance Engine now features an interactive API client that can generate commands specific to your masking engine. With those commands, you can:

- make changes to your engine
- copy and paste the commands to write code that can automate your masking activities

> The API client will make real changes to your virtual machine. Any operations you run using the API Client will persist on the machine!

To access the Masking API client, use the following URL: http://myMaskingEngine.com/masking/api-client/, replacing "myMaskingEngine.com" with the hostname or IP address of your virtual machine.

For detailed examples of using API calls to automate masking, see the Masking APIs[795]

795 https://maskingdocs.delphix.com/Delphix_Masking_APIs/Masking_Client/Masking_API_Client/

13.3.8.7.8.2 Masking in Parallel

Continuous Compliance supports launching masking jobs in parallel. When jobs have no dependencies, you can initiate parallel masking API jobs (with wrapper code as required) to allow the jobs to be run as a pre and/or post hook.

13.3.8.7.9 Oracle use cases for APIs

13.3.8.7.9.1 Oracle link + snapshot (sync)

The following script ingests links an environment database dSource (Oracle SID / Instance) and then takes a snapshot. See parameters for required values that you must provide.

This script demonstrates how to use name values inputs and get the respective Delphix object and/or object reference for use in the json input in downstream API calls.

Filename: link_oracle.sh # or link_oracle_jq.sh

Edit the file to update the parameters as required for your environment.

```
##########################################################Parameter
InitializationDMIP=172.16.160.195#DMPORT=8282DMUSER=delphix_adminDMPASS=delphix . . .
Required for Database Link and Sync ...SOURCE_SID="DPXDEV01"            # Source
Environment Database SIDSOURCE_NAME="DPXDEV01"            # Delphix dSource
NameSOURCE_ENV="Oracle Target"        # Source Environment NameSOURCE_GRP="Oracle_Sou
rce"        # Delphix Group NameDB_USER="delphixdb"                # Source Database
SID user accountDB_PASS="delphixdb"                # Source Database SID user
password ##############################################    NO CHANGES REQUIRED BELOW
THIS POINT    ##############################################$ ./
link_oracle.sh        # or    ./link_oracle_jq.shAuthenticating on http://
172.16.160.195/resources/json/delphixSession and Login Successful ...group reference:
GROUP-35sourceconfig reference: ORACLE_SINGLE_CONFIG-1primaryUser reference:
HOST_USER-3Linking Source Database ...Job: JOB-92Job: JOB-92 100.0%
Completed ...Container: ORACLE_DB_CONTAINER-19Running SnapSync ...Job: JOB-93Current
status as of Mon Aug 15 13:07:53 EDT 2016 : RUNNING : 0.0% CompletedCurrent status as
of Mon Aug 15 13:08:03 EDT 2016 : RUNNING : 15.0% CompletedCurrent status as of Mon
Aug 15 13:08:13 EDT 2016 : RUNNING : 35.0% CompletedCurrent status as of Mon Aug 15
13:08:24 EDT 2016 : RUNNING : 59.0% Completed
```

```
Current status as of Mon Aug 15 13:08:34 EDT 2016 : RUNNING : 66.0% CompletedCurrent
status as of Mon Aug 15 13:08:44 EDT 2016 : RUNNING : 74.0% CompletedJob: JOB-93
100.0% Completed ... Done ... $
```

13.3.8.7.9.2 Oracle provision

Filename: provision_oracle.txt

Shown below is how to use the CLI to provision an Oracle 11g database that is already ingested into the Delphix Engine.

The key is to get the object reference names first. For example, to get the source database container name:

```
ssh delphix_admin[delphix_engine_ip_address_or_hostname]> database> ls...> select
"[database_name]"> lsDelphix5002HWv7 database> select 'DPXDEV01'Delphix5002HWv7
database 'DPXDEV01'> lsProperties    type: OracleDatabaseContainer    name:
DPXDEV01    . . .        reference: ORACLE_DB_CONTAINER-18     . . .
```

Minimum parameters required to provision:

```
Delphix5002HWv7 database provision > *commit=== POST /resources/json/delphix/
database/provision ===={    "type": "OracleProvisionParameters",    "container":
{        "type": "OracleDatabaseContainer",        "name": "VBITT" , # Delphix Object
Name, Typically matches VDB name         "group": "GROUP-36" # group ls select
"[group_name]" ls     },    "source": {        "type": "OracleVirtualSource",
        "mountBase": "/mnt/provision" # Delphix Filesystem Mount path     },    "sour
ceConfig": {        "type": "OracleSIConfig",         "repository":
"ORACLE_INSTALL-3" , # repository, select "[repository_name]"         "databaseName":
"VBITT" , # New VDB Name         "uniqueName": "VBITT",         "instance":
{            "type": "OracleInstance",            "instanceName": "VBITT",
 "instanceNumber": 1        }    },    "timeflowPointParameters": {        "type":
"TimeflowPointSemantic",        "container": "ORACLE_DB_CONTAINER-18" # select
"[datbase_name]" ls     }}=== RESPONSE ===
```

Sample CLI session

```
Delphix5002HWv7 database> setopt trace=false Delphix5002HWv7 database> provision
Delphix5002HWv7 database provision > *ls Properties    type:
OracleProvisionParameters    container:        type:
OracleDatabaseContainer        name: (required)        description:
(unset)        diagnoseNoLoggingFaults: true        group:
(required)        performanceMode: DISABLED        preProvisioningEnabled: false
   sourcingPolicy: (unset)    credential: (unset)    maskingJob: (unset)    newDBID:
false    openResetlogs: true    physicalStandby: false    source:        type:
OracleLiveSource        name: (unset)        archivelogMode: true        config:
(unset)        configParams: (unset)        configTemplate:
(unset)        customEnvVars: (unset)        dataAgeWarningThreshold:
900sec        fileMappingRules: (unset)        manualProvisioning: false        mount
Base: (required)        nodeListenerList: (unset)        operations:
(unset)        redoLogGroups: 3        redoLogSizeInMB: 0    sourceConfig:        typ
e: OraclePDBConfig        cdbConfig: (required)        databaseName:
(required)        environmentUser: (unset)        linkingEnabled: true        reposit
ory: (unset)        services: (unset)    timeflowPointParameters:        type:
TimeflowPointSemantic        container: (required)        location:
LATEST_POINT    username: (unset) OperationsdefaultsDelphix5002HWv7 database
provision > *edit container Delphix5002HWv7 database provision container> *ls
```

```
Properties    type: OracleDatabaseContainer    name: (required)    description:
(unset)    diagnoseNoLoggingFaults: true    group: (required)    performanceMode:
DISABLED    preProvisioningEnabled: false    sourcingPolicy: (unset)Delphix5002HWv7
database provision container> *set name=VBITT Delphix5002HWv7 database provision
container> *set group=GROUP-36 Delphix5002HWv7 database provision container> *back
Delphix5002HWv7 database provision > *edit source Delphix5002HWv7 database provision
source > *ls Properties    type: OracleLiveSource    name: (unset)    archivelogMode:
true    config: (unset)    configParams: (unset)    configTemplate:
(unset)    customEnvVars: (unset)    dataAgeWarningThreshold:
900sec    fileMappingRules: (unset)    manualProvisioning: false    mountBase:
(required)    nodeListenerList: (unset)    operations: (unset)    redoLogGroups: 3
 redoLogSizeInMB: 0Delphix5002HWv7 database provision source > *set
type=OracleVirtualSource Delphix5002HWv7 database provision source > *set mountBase=/
mnt/provision Delphix5002HWv7 database provision source > *back Delphix5002HWv7
database provision > *edit sourceConfig Delphix5002HWv7 database provision
sourceConfig > *ls Properties    type: OraclePDBConfig    cdbConfig:
(required)    databaseName: (required)    environmentUser: (unset)    linkingEnabled:
true    repository: (unset)    services: (unset)Delphix5002HWv7 database provision
sourceConfig > *set type=OracleSIConfig Delphix5002HWv7 database provision
sourceConfig > *ls Properties    type: OracleSIConfig      databaseName:
(required)    environmentUser: (unset)    instance: (required)    linkingEnabled:
true    nonSysCredentials: (unset)    nonSysUser: (unset)    repository:
(required)    services: (unset)    uniqueName: (required)Delphix5002HWv7 database
provision sourceConfig > *set databaseName=VBITT Delphix5002HWv7 database provision
sourceConfig > *set repository=ORACLE_INSTALL-3 Delphix5002HWv7 database provision
sourceConfig > *set uniqueName=VBITT Delphix5002HWv7 database provision sourceConfig
> *set instance.instanceName=VBITT Delphix5002HWv7 database provision sourceConfig >
*set instance.instanceNumber=1 Delphix5002HWv7 database provision sourceConfig > *ls
Properties    type: OracleSIConfig      databaseName: VBITT      environmentUser:
(unset)    instance:        type: OracleInstance          instanceName: VBITT
        instanceNumber: 1      linkingEnabled: true    nonSysCredentials:
(unset)    nonSysUser: (unset)    repository: '/u02/ora/app/product/11.2.0/dbhome_1'
    services: (unset)    uniqueName: VBITT  Delphix5002HWv7 database provision
sourceConfig > *back Delphix5002HWv7 database provision > *edit
timeflowPointParameters Delphix5002HWv7 database provision timeflowPointParameters>
*ls Properties    type: TimeflowPointSemantic    container: (required)    location:
LATEST_POINTDelphix5002HWv7 database provision timeflowPointParameters> *set
container=ORACLE_DB_CONTAINER-18 Delphix5002HWv7 database provision
timeflowPointParameters> *back Delphix5002HWv7 database provision > *commit
    VBITT    Dispatched job JOB-348    DB_PROVISION job started for "Oracle Target
Virtual Databases/VBITT".    Starting provision of the virtual database "VBITT".    C
reating new TimeFlow.    Generating recovery scripts.    Exporting
storage.    Mounting filesystems for the virtual database instance "1".    Mounting
read-only archive log filesystem for the virtual database instance "1".    Recovering
Oracle database.    \|/-    Opening the virtual database "VBITT".    Opening Oracle
database.    Oracle recovery was successful.    Unmounting read-only archive log
filesystem for the virtual database instance "1".    The virtual database "VBITT" was
successfully provisioned.    DB_PROVISION job for "Oracle Target Virtual Databases/
VBITT" completed successfully.Delphix5002HWv7 database>
```

With the **setopt trace=true** option set, you can convert the JSON output from the above CLI provision command to the RESTful API cURL commands. If VBITT exists, be sure to delete it first.

Request:

```
curl X POST -k --data @http://172.16.160.177/resources/json/delphix/database/
provision \    -b cookies.txt -H "Content-Type: application/json" <<EOF{    "type":
"OracleProvisionParameters",    "container": {        "type":
"OracleDatabaseContainer",        "name": "VBITT",         "group": "GROUP-36"
    },    "source": {        "type": "OracleVirtualSource",         "mountBase": "/
mnt/provision"    },    "sourceConfig": {        "type": "OracleSIConfig",
        "repository": "ORACLE_INSTALL-3",         "databaseName": "VBITT",
        "uniqueName": "VBITT",         "instance": {            "type":
"OracleInstance",            "instanceName": "VBITT",            "instanceNumber":
1        }    },    "timeflowPointParameters": {        "type":
"TimeflowPointSemantic",        "container": "ORACLE_DB_CONTAINER-18"     }}EOF
```

Response:

```
{"type":"OKResult","status":"OK","result":"ORACLE_DB_CONTAINER-22","job":"JOB-353","a
ction":"ACTION-649"}
```

Put all the commands above within a shell script to automate the complete process of provisioning an Oracle 11.2.0.4 database.

Notice that the script below looks up 4 object references for use within the JSON input into the API.

Filename: provision_oracle.sh# or provision_oracle_jq.sh

Edit the file to update the parameters as required for your environment.

```
#######################    DELPHIX CORP    ########################Parameter
Initialization DMIP=172.16.160.195DMUSER=delphix_adminDMPASS=delphixCOOKIE="~/
cookies.txt"COOKIE=`eval echo $COOKIE`CONTENT_TYPE="Content-Type: application/json"DE
LAYTIMESEC=10BaseURL="http://${DMIP}/resources/json/delphix"#Required for Database
Link and Sync ...#VDB_NAME="VBITT"                # Delphix VDB NameMOUNT_BASE="/mnt/
provision"        # Delphix Engine Mount PathSOURCE_GRP="Oracle_Target"        #
Delphix Engine Group NameTARGET_ENV="Oracle Target"        # Target Environment used
to get repository reference valueSOURCE_SID="DPXDEV01"            # dSource name used
to get db container reference value
##############################################    NO CHANGES REQUIRED BELOW THIS
POINT    ##############################################
```

Sample Output

```
$ ./provision_oracle.sh# or ./provision_oracle_jq.shAuthenticating on http://
172.16.160.195/resources/json/delphixSession and Login Successful ...group reference:
GROUP-36 container reference: ORACLE_DB_CONTAINER-36 env reference:
UNIX_HOST_ENVIRONMENT-3 repository reference: ORACLE_INSTALL-1 Provisioning VDB from
```

```
Source Database ...Job: JOB-155Current status as of Mon Aug 15 23:40:51 EDT 2016 :
RUNNING 0.0% CompletedCurrent status as of Mon Aug 15 23:40:51 EDT 2016 : RUNNING
0.0% CompletedCurrent status as of Mon Aug 15 23:41:01 EDT 2016 : RUNNING 9.0%
CompletedCurrent status as of Mon Aug 15 23:41:11 EDT 2016 : RUNNING 45.0%
CompletedCurrent status as of Mon Aug 15 23:41:21 EDT 2016 : RUNNING 45.0%
CompletedCurrent status as of Mon Aug 15 23:41:31 EDT 2016 : RUNNING 46.0%
CompletedCurrent status as of Mon Aug 15 23:41:41 EDT 2016 : RUNNING 48.0%
CompletedCurrent status as of Mon Aug 15 23:41:51 EDT 2016 : RUNNING 60.0%
CompletedJob: JOB-155 COMPLETED 100.0% Completed ... Done ... $
```

Filename: provision_oracle_child.sh# or provision_oracle_child_jq.sh

```
$ ./provision_oracle_child.sh# or ./provision_oracle_child_jq.shAuthenticating on
http://172.16.160.195/resources/json/delphixSession and Login Successful ...group
reference: GROUP-36 container reference: ORACLE_DB_CONTAINER-118 env reference:
UNIX_HOST_ENVIRONMENT-9 repository reference: ORACLE_INSTALL-6 Provisioning VDB from
Source Database ...Job: JOB-857Current status as of Mon Sep 5 22:48:28 EDT 2016 :
RUNNING 0.0% CompletedCurrent status as of Mon Sep 5 22:48:28 EDT 2016 : RUNNING 0.0%
CompletedCurrent status as of Mon Sep 5 22:48:38 EDT 2016 : RUNNING 9.0%
CompletedCurrent status as of Mon Sep 5 22:48:48 EDT 2016 : RUNNING 27.0%
CompletedCurrent status as of Mon Sep 5 22:48:58 EDT 2016 : RUNNING 42.0%
CompletedCurrent status as of Mon Sep 5 22:49:08 EDT 2016 : RUNNING 45.0%
CompletedCurrent status as of Mon Sep 5 22:49:28 EDT 2016 : RUNNING 46.0%
CompletedCurrent status as of Mon Sep 5 22:49:38 EDT 2016 : RUNNING 48.0%
CompletedCurrent status as of Mon Sep 5 22:49:48 EDT 2016 : RUNNING 51.0%
CompletedCurrent status as of Mon Sep 5 22:50:08 EDT 2016 : RUNNING 71.0%
CompletedJob: JOB-857 COMPLETED 100.0% Completed ... Done ...
```

13.3.8.7.10 SQL server API use cases

13.3.8.7.10.1 SQL server link/ingest environment dSource

For the **Window Target** environment, the dSource **delphixdb** in **MSSQLSERVER** instance will be linked/ingested into the Delphix Engine. It will appear in the **Windows_Source** group below.

Filename: link_sqlserver.ps1

```
PS> . .\link_sqlserver.ps1Authenticating on http://172.16.160.195/resources/json/
delphixLogin Successful ...group reference: GROUP-34 sourceconfig reference:
MSSQL_SINGLE_CONFIG-26 env reference: WINDOWS_ HOST_ENVIRONMENT-7 repository
reference: MSSQL_INSTANCE-4 database link API Results:
{"type":"OKResult","status":"OK","result":"MSSQL_DB_CONTAINER-114","job":"JOB-819","a
ction":"ACTION-1659"}DB Container: MSSQL_DB_CONTAINER-114
```

```
Job # JOB-819*****    waiting for status    *****Current status as of 09/05/2016 11:4
1:13 : COMPLETED : 100.0% CompletedJob COMPLETED Succesfully.
```

```
JOB JOB-820waiting for status *****Current status as of 09/05/2016 11:41:23 :
RUNNING : 5.0% CompletedCurrent status as of 09/05/2016 11:41:44 : RUNNING : 9.0%
Completed
```

```
Current status as of 09/05/2016 11:41:54 : RUNNING : 56.0% CompletedJob COMPLETED
Succesfully. Done ...
```

Successful dSource linked/ingested into the Delphix Engine.

13.3.8.7.10.2 SQL server provision

The example below is done from the command line once you know the parameters and reference object names.

Filename: windows_sqlserver_provision.txt

Create these 3 JSON text files:

```
session.json{    "type": "APISession",    "version": {        "type": "APIVersion",
      "major": 1,        "minor": 5,        "micro": 3    }} login.json{    "type":
"LoginRequest",    "username": "delphix_admin",    "password": "delphix"}
provision.json{    "type": "MSSqlProvisionParameters",    "container": {        "type
": "MSSqlDatabaseContainer",        "name": "Vbitt00",        "group": "GROUP-36",
     "sourcingPolicy": {            "type": "SourcingPolicy",            "loadFromBac
kup": false,            "logsyncEnabled": false        },        "validatedSyncMode":
"TRANSACTION_LOG"    },    "source": {        "type": "MSSqlVirtualSource",        "o
perations": {            "type": "VirtualSourceOperations",            "configureClon
e": [],            "postRefresh": [],            "postRollback": [],            "post
Snapshot": [],            "preRefresh": [],            "preSnapshot":
[]        }    },    "sourceConfig": {        "type": "MSSqlSIConfig",        "linkin
gEnabled": false,        "repository": "MSSQL_INSTANCE-1",        "databaseName":
"Vbitt00",        "recoveryModel": "SIMPLE",        "instance": {            "type":
"MSSqlInstanceConfig",            "host": "WINDOWS_HOST-1"        }    },    "timeflo
wPointParameters": {        "type": "TimeflowPointSemantic",        "container":
"MSSQL_DB_CONTAINER-23",        "location": "LATEST_SNAPSHOT"    }}
```

This works on Windows Powershell Command Prompt

> ⚠ Use curl, curl.exe or modify the default alias.

```
curl --insecure -c cookies.txt -i -X POST -H "Content-Type: application/json" -d
"@session.json" http://172.16.160.153/resources/json/delphix/sessioncurl --insecure
-b cookies.txt -i -X POST -H "Content-Type: application/json" -d "@login.json"
http://172.16.160.153/resources/json/delphix/logincurl --insecure -b cookies.txt -i
```

```
-X POST -H "Content-Type: application/json" -d "@provision.json" http://
172.16.160.153/resources/json/delphix/database/provision
```

Plug in the returned JOB #

```
curl --insecure -b cookies.txt -i -X GET -H "Content-Type: application/json" -k http:/
/172.16.160.153/resources/json/delphix/notification?channel=JOB-428
```

Get Example

```
curl --insecure -b cookies.txt -i -X GET -H "Content-Type: application/json" -k http:/
/172.16.160.153/resources/json/delphix/system
```

Complete example.

Provision the newly created **delphixdb** dSource in the **Windows_Source** group to a virtual database VBITT in the **Windows_Target** group.

Filename: provision_sqlserver.ps1

Variables ...

```
$nl = [Environment]::NewLine$BaseURL = " http://172.16.160.195/resources/json/delphix
"$cookie = "cookies.txt"$delphix_user = "delphix_admin"$delphix_pass = "delphix". . .
```

Required for Provisioning Virtual Database ...

```
$SOURCE_SID="delphixdb"            # dSource name used to get db container reference
value $VDB_NAME="VBITT"                # Delphix VDB Name$TARGET_GRP="Windows_Target"
   # Delphix Engine Group Name$TARGET_ENV="Window Target"        # Target Environment
used to get repository reference value $TARGET_REP="MSSQLSERVER"    # Target
Environment Repository / Instance name
##############################################    NO CHANGES REQUIRED BELOW THIS
POINT    ##############################################
```

Sample Run Output

```
PS> . .\provision_sqlserver.ps1Authenticating on http://172.16.160.195/resources/
json/delphixLogin Successful ...group reference:  GROUP-37 container reference:
MSSQL_DB_CONTAINER-114 env reference:  WINDOWS_HOST_ENVIRONMENT-7 repository
reference:  MSSQL_INSTANCE-4 database provision API Results:
{"type":"OKResult","status":"OK","result":"MSSQL_DB_CONTAINER-115","job":"JOB-822","a
ction":"ACTION-1664"}DB Container:  MSSQL_DB_CONTAINER-115 Job # JOB-822 jobState
RUNNINGpercentComplete 0.0***** waiting for status *****Current status as of
09/05/2016 11:43:51 : RUNNING : 0.0% CompletedCurrent status as of 09/05/2016
11:44:01 : RUNNING : 3.0% CompletedCurrent status as of 09/05/2016 11:44:12 :
RUNNING : 11.0% CompletedCurrent status as of 09/05/2016 11:44:22 : RUNNING : 18.0%
CompletedCurrent status as of 09/05/2016 11:44:32 : RUNNING : 18.0% Completed
```

```
Current status as of 09/05/2016 11:44:52 : RUNNING : 75.0% CompletedJob COMPLETED
Succesfully. Done ...
```

13.3.8.7.10.3 SQL server refresh

The following are curl commands that can be issued from the Powershell command line. For inclusion within a Powershell script, see the masking example, *masking.ps1*.

Filename: windows_sqlserver_refresh.txt

MS SQL Server Refresh Example ...

Session ...

```
curl --insecure -c cookies.txt -i -X POST -H "Content-Type: application/json" -d
"@session.json" http://172.16.160.179/resources/json/delphix/session
```

Filename: session.json

```
{    "type": "APISession",    "version": {        "type": "APIVersion",        "major
": 1,        "minor": 5,        "micro": 3    }} PS> *curl --insecure -c cookies.txt
-i -X POST -H "Content-Type: application/json" -d "@session.json" http://
172.16.160.179/resources/json/delphix/session*HTTP/1.1 200 OKServer: Apache-Coyote/
1.1Set-Cookie: JSESSIONID=8DE0362F5BBD73E6BFA9E13FF111E78C; Path=/resources/;
HttpOnlyContent-Type: application/jsonContent-Length: 179Date: Thu, 16 Jun 2016
07:24:34 GMT{"type":"OKResult","status":"OK","result":{"type":"APISession","version":
{"type":"APIVersion","major":1,"minor":5,"micro":3},"locale":null,"client":null},"job
":null,"action":null} PS>

Login ...

curl --insecure -b cookies.txt -i -X POST -H "Content-Type: application/json" -d
"@login.json" http://172.16.160.179/resources/json/delphix/login

Filename:login.json

{    "type": "LoginRequest",    "username": "delphix_admin",    "password": "delphix"}
  PS> *curl --insecure -b cookies.txt -i -X POST -H "Content-Type: application/json"
 -d "@login.json" http://172.16.160.179/resources/json/delphix/login*HTTP/1.1 200
OKServer: Apache-Coyote/1.1Content-Type: application/jsonContent-Length: 76Date: Thu,
16 Jun 2016 07:25:39 GMT
{"type":"OKResult","status":"OK","result":"USER-2","job":null,"action":null}PS C:
\Users\Administrator>
```

List Databases ...

```
curl --insecure -b cookies.txt -i -X GET -H "Content-Type: application/json" -k http:/
/172.16.160.179/resources/json/delphix/databasePS> *curl --insecure -b cookies.txt -i
-X GET -H "Content-Type: application/json" -k http://172.16.160.179/resources/json/
```

```
delphix/database*HTTP/1.1 200 OKServer: Apache-Coyote/1.1Content-Type: application/
jsonContent-Length: 4062Date: Thu, 16 Jun 2016 07:27:14 GMT
{"type":"ListResult","status":"OK","result":[......
{"type":"MSSqlDatabaseContainer","reference":"MSSQL_DB_CONTAINER-37","namespace":null
,"name":"Vdelphix_demo","group":"GROUP-35","provisionContainer":"MSSQL_DB_CONTAINER-3
6","creationTime":"2016-06-16T07:09:06.222Z","currentTimeflow":"MSSQL_TIMEFLOW-38","p
reviousTimeflow":"MSSQL_TIMEFLOW-37","description":null,"runtime":...
{"type":"So{"type":"MSSqlDatabaseContainer","reference":"MSSQL_DB_CONTAINER-36","name
space":null,"name":"delphix_demo","group":"GROUP-35","provisionContainer":null,"creat
ionTime":"2016-06-16T07:07:49.939Z","currentTimeflow":"MSSQL_TIMEFLOW-36","previousTi
meflow":null,"description":"","runtime":......}],"job":null,"action":null,"total":6,"
overflow":false} PS>
```

Need Reference Object from Database Information ...

For Parent Source Database delphix_demo, reference object is MSSQL_DB_CONTAINER-36

For Virtual Database Vdelphix_demo, reference object is MSSQL_DB_CONTAINER-37

[Optional: Get Database Info ...]

```
curl --insecure -b cookies.txt -i -X GET -H "Content-Type: application/json" -k http:/
/172.16.160.179/resources/json/delphix/database/MSSQL_DB_CONTAINER-37 PS> *curl --
insecure -b cookies.txt -i -X GET -H "Content-Type: application/json" -k http://
172.16.160.179/resources/json/delphix/database/MSSQL_DB_CONTAINER-37*HTTP/1.1 200
OKServer: Apache-Coyote/1.1Content-Type: application/jsonContent-Length: 696Date:
Thu, 16 Jun 2016 07:35:42 GMT {"type":"OKResult","status":"OK","result":
{"type":"MSSqlDatabaseContainer","reference":"MSSQL_DB_CONTAINER-37","namespace":null
,"name":"Vdelphix_demo","group":"GROUP-35","provisionContainer":"MSSQL_DB_CONTAINER-3
6","creationTime":"2016-06-16T07:09:06.222Z","currentTimeflow":"MSSQL_TIMEFLOW-38","p
reviousTimeflow":"MSSQL_TIMEFLOW-37","description":null,"runtime":
{"type":"MSSqlDBContainerRuntime","logSyncActive":false,"preProvisioningStatus":null,
"lastRestoredBackupSetUUID":null},"os":"Windows","processor":"x86","sourcingPolicy":
{"type":"SourcingPolicy","logsyncEnabled":false,"loadFromBackup":false},"performanceM
ode":"DISABLED","delphixManaged":true,"masked":false},"job":null,"action":null} PS>
```

Refresh Vdelphix_demo using parent delphix_demo (MSSQL_DB_CONTAINER-36) with the latest timecard ...

```
curl --insecure -b cookies.txt -i -X POST -H "Content-Type: application/json" -d
"@refresh.json" http://172.16.160.179/resources/json/delphix/database/
MSSQL_DB_CONTAINER-37/refresh=== POST /resources/json/delphix/database/
MSSQL_DB_CONTAINER-37/refresh ===refresh.json{    "type":
"RefreshParameters",    "timeflowPointParameters": {        "type":
"TimeflowPointSemantic",        "container": "MSSQL_DB_CONTAINER-36"    }} PS> *curl
--insecure -b cookies.txt -i -X POST -H "Content-Type: application/json" -d
"@refresh.json" http://172.16.160.179/resources/json/delphix/database/
MSSQL_DB_CONTAINER-37/refresh*HTTP/1.1 200 OKServer: Apache-Coyote/1.1Content-Type:
application/jsonContent-Length: 82Date: Thu, 16 Jun 2016 07:40:44 GMT
{"type":"OKResult","status":"OK","result":"","job":"JOB-60","action":"ACTION-167"}
PS>
```

[Observer Delphix GUI Action]

Done with SQL Server VDB Refresh ...

13.3.8.8 API programming language examples

The following programming language examples are just to show the bare minimum authentication and a sample functional API call. There are numerous modules, libraries, methods, functions, and code examples to further demonstrate how the languages work with the Delphix APIs and JSON data strings/objects.

You can execute PHP, Perl, and Python languages from the command line and/or from within a Web Server such as Apache or IIS. The following examples are formatted for command line / terminal output.

13.3.8.8.1 PHP

PHP provides cURL and JSON modules.

```
$ php -i | grep -iE "cURL|json"
curl
cURL support => enabled
cURL Information => 7.43.0
json
json support => enabled
json version => 1.2.1
```

Filename: delphix_curl.php

Sample Output:

```
$ php -f delphix_curl.php
Session json> {"type":"APISession","version":{"type":"APIVersion","major":1,"minor":7,"
micro":0}}
Session Results> {"type":"OKResult","status":"OK","result":{"type":"APISession","vers
ion":{"type":"APIVersion","major":1,"minor":7,"micro":0},"locale":null,"client":null}
,"job":null,"action":null}
Login json> {"type":"LoginRequest","username":"delphix_admin","password":"delphix"}
Login Results> {"type":"OKResult","status":"OK","result":"USER-2","job":null,"action":n
ull}
Calling About API ...
About Results> {"type":"OKResult","status":"OK","result":{"type":"PublicSystemInfo","
productType":"standard","productName":"Delphix Engine","buildTitle":"Delphix Engine
5.1.1.0","buildTimestamp":"2016-07-21T07:23:41.000Z","buildVersion":{"type":"VersionI
nfo","major":5,"minor":1,"micro":1,"patch":0},"configured":true,"enabledFeatures":
["XPP","MSSQLHOOKS"],"apiVersion":{"type":"APIVersion","major":1,"minor":8,"micro":0}
,"banner":null,"locales":["en-US"],"currentLocale":"en-US"},"job":null,"action":null}
Converting json string to a PHP Array
stdClass Object
(
    [type] => OKResult
    [status] => OK
    [result] => stdClass Object
```

```
        (
            [type] => PublicSystemInfo
            [productType] => standard
            [productName] => Delphix Engine
            [buildTitle] => Delphix Engine 5.1.1.0
            [buildTimestamp] => 2016-07-21T07:23:41.000Z
            [buildVersion] => stdClass Object
                (
                    [type] => VersionInfo
                    [major] => 5
                    [minor] => 1
                    [micro] => 1
                    [patch] => 0
                )
            [configured] => 1
            [enabledFeatures] => Array
                (
                    [0] => XPP
                    [1] => MSSQLHOOKS
                )
             [apiVersion] => stdClass Object
                (
                    [type] => APIVersion
                    [major] => 1
                    [minor] => 8
                    [micro] => 0
                )
             [banner] =>
             [locales] => Array
                (
                    [0] => en-US
                )
             [currentLocale] => en-US
        )
    [job] =>
    [action] =>
)
```

13.3.8.8.2 Perl

Perl provides a couple of methods for working with cURL: operating system calls, WWW::Curl (libcurl) module, or LWP::Curl module. The sample below simply logs into the Delphix Engine and lists the current Delphix Environments.

Filename: perl_curl.pl

Sample Output:

```
$ perl perl_curl.pl
Testing cURL on Perl ...
```

```
Session Results: {"type":"OKResult","status":"OK","result":{"type":"APISession","vers
ion":{"type":"APIVersion","major":1,"minor":7,"micro":0},"locale":null,"client":null}
,"job":null,"action":null}
Login Results: {"type":"OKResult","status":"OK","result":"USER-2","job":null,"action":n
ull}

Enviornment Results: {"type":"ListResult","status":"OK","result":[{"type":"WindowsHos
tEnvironment","reference":"WINDOWS_HOST_ENVIRONMENT-7","namespace":null,"name":"Windo
w Target","description":null,"primaryUser":"HOST_USER-7","enabled":false,"host":"WIND
OWS_HOST-6","proxy":null},{"type":"UnixHostEnvironment","reference":"UNIX_HOST_ENVIRO
NMENT-9","namespace":null,"name":"Oracle Target","description":"","primaryUser":"HOST
_USER-9","enabled":true,"host":"UNIX_HOST-8","aseHostEnvironmentParameters":null}],"j
ob":null,"action":null,"total":2,"overflow":false}

Done
```

13.3.8.8.3 Python

Delphix has an extensive resource library for using Python with the Delphix Engine.

CLI to Python transition (see page 2769)

Delphix python module

13.3.8.8.4 Blogs

https://github.com/CloudSurgeon/delphixpy-examples

13.3.8.8.5 Related Videos

- https://vimeo.com/164779308
- https://vimeo.com/170187276
- https://vimeo.com/170896907

Simple Python program to authenticate and get the "about" API results. This script requires the "request" and "json" modules.

http://stackoverflow.com/questions/17309288/importerror-no-module-named-requests

Filename: auth.py

Sample Output

```
$ python auth.py
Authenticating URL http://172.16.160.195/resources/json/delphix ...
{"type":"OKResult","status":"OK","result":{"type":"APISession","version":{"type":"API
Version","major":1,"minor":7,"micro":0},"locale":null,"client":null},"job":null,"acti
on":null}
Login ...
{"type":"OKResult","status":"OK","result":"USER-2","job":null,"action":null}
```

```
About ...
{"type":"OKResult","status":"OK","result":{"type":"PublicSystemInfo","productType":"s
tandard","productName":"Delphix Engine","buildTitle":"Delphix Engine 5.1.1.0","buildT
imestamp":"2016-07-21T07:23:41.000Z","buildVersion":{"type":"VersionInfo","major":5,"
minor":1,"micro":1,"patch":0},"configured":true,"enabledFeatures":["XPP","MSSQLHOOKS"]
,"apiVersion":{"type":"APIVersion","major":1,"minor":8,"micro":0},"banner":null,"loca
les":["en-US"],"currentLocale":"en-US"},"job":null,"action":null}
JSON Parsing Examples ...
OK
Delphix Engine 5.1.1.0
1
```

13.3.8.8.6 JSP (Java server pages)

Java Server Pages are typically used for the web formatting and output, but you can also use JSP for application logic processing and native Java code integration, although this is scorned by the purest and most logical thinking programmers.

Filename: delphix_http.jsp

Sample Output:

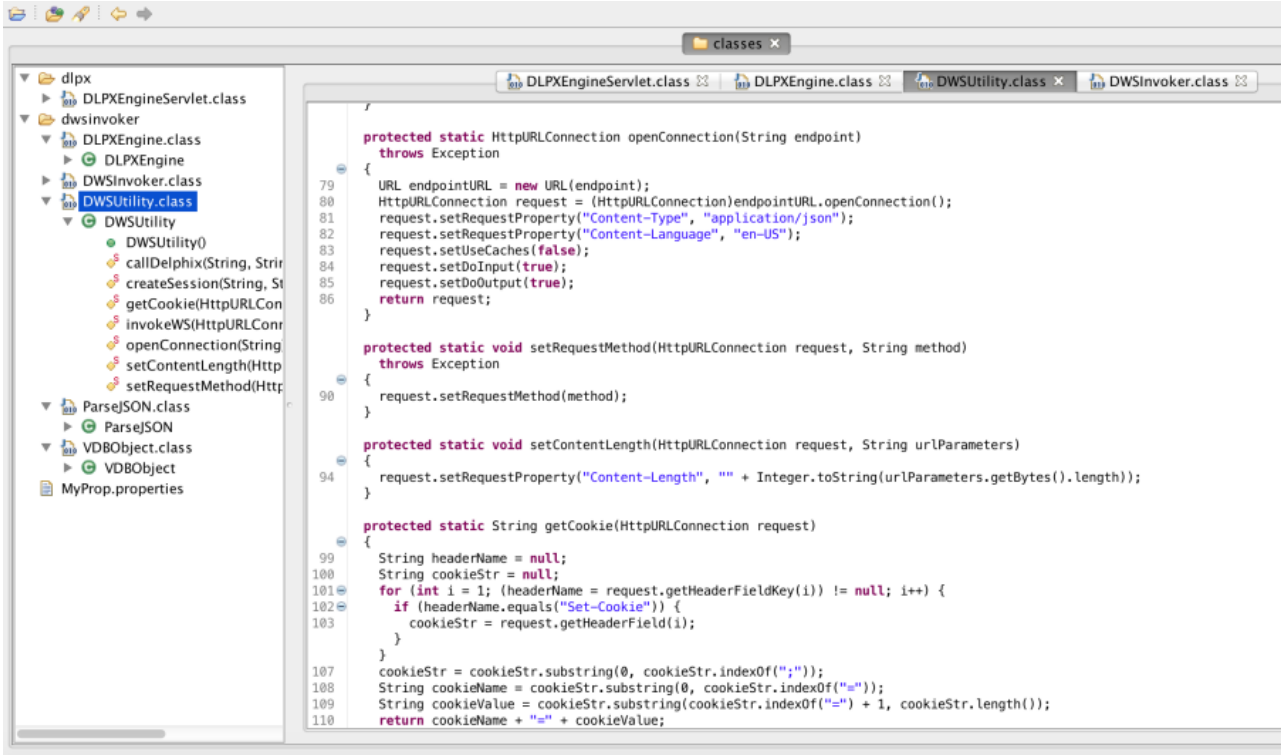
Browser URL: http://localhost:8080/delphix_http.jsp

localhost:8080/delphix_http.jsp


```
Trying ...
session> {"type":"OKResult","status":"OK","result":{"type":"APISession","version":{"type":"APIVersion","major":1,"minor":7,"micro":0},"locale":null,"client":null},"job":null,"action":null}
cookie> JSESSIONID=D16845959B873C64F7460E99A67BFFEB
login> {"type":"OKResult","status":"OK","result":"USER-2","job":null,"action":null}
system results> {"type":"OKResult",
"status":"OK",
"result":{"type":"SystemInfo",
"productType":"standard",
"productName":"Delphix Engine",
"buildTitle":"Delphix Engine 5.1.1.0",
"buildTimestamp":"2016-07-21T07:23:41.000Z",
"buildVersion":{"type":"VersionInfo",
"major":5,
"minor":1,
"micro":1,
"patch":0},
"configured":true,
"enabledFeatures":["XPP",
"MSSQLHOOKS"],
"apiVersion":{"type":"APIVersion",
"major":1,
"minor":8,
"micro":0},
"banner":null,
"locales":["en-US"],
"currentLocale":"en-US",
"hostname":"Delphix5110HWv8",
"sshPublicKey":"ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDOsrp7Aj6hFQh9yBq7273B+qtPKmCu1B18nPvr08yjt/IZeM4qKk7caxExQS9rpfU8AWoT7e8ESV7NkBmUzOHrHnLsuJtxPqeYoqeMubVxYjJ
root@Delphix5110HWv8\n",
"memorySize":8.58107904E9,
"platform":"VMware with BIOS date 05/20/2014",
"uuid":"564d7e1d-f4cb-f910-98fd-348d74817683",
"processors":[{"type":"CPUInfo",
"speed":2.5E9,
"cores":1}],
"storageUsed":3.121203712E9,
"storageTotal":2.0673724416E10,
"installationTime":"2016-07-27T13:28:46.000Z"},
"job":null,
"action":null}
```

13.3.8.8.7 Java

Java methods and classes allow coding logic to be effectively re-used, extended and modularized for flexible applications. Using the Java code embedded within the JSP file, code is logically placed into respective classes and methods.



13.3.8.9 API timeflows

From earlier, the RESTFul URL for a virtual database refresh will look like:

http://<delphix_engine>/resources/json/delphix/database/MSSQL_DB_CONTAINER-39/refresh

where the **MSSQL_DB_CONTAINER-39** represents the target virtualized database to refresh and we need to POST the JSON data to the URL for processing.

```
{    "type": "RefreshParameters",    "timeflowPointParameters": {        "type":
"TimeflowPointSemantic",        "container": "MSSQL_DB_CONTAINER-38"    }}
```

13.3.8.9.1 Timeflow parameters

The **"timeflowPointParameters"** key has 6 **"type": "..."** options which each have their own set of parameters. The type **"TimeflowPointSemantic"** uses the default LATEST_POINT within the source container. Now, for more on timeflowPointParameters.

http://<delphix_engine>/api/#TimeflowPointParameters

13.3.8.9.1.1 TimeflowPointParameters

Parameters indicating a TimeFlow point to use as input to database operations.

TypedObject
TimeflowPointParameters

Direct Known Subclasses:

TimeflowPointTimestamp, TimeflowPointSnapshot, TimeflowPointSemantic, TimeflowPointLocation, TimeflowPointBookmark, TimeflowPointBookmarkTag

13.3.8.9.1.2 TimeflowPointTimestamp

timeflow	Reference to TimeFlow containing this point. Type: Reference to Timeflow Constraints: Required: true
timestamp	The logical time corresponding to the TimeFlow location. Type: date Constraints: Required: true

13.3.8.9.1.3 TimeflowPointSnapshot

snapshot	Reference to the snapshot. Type: Reference to TimeflowSnapshot

13.3.8.9.1.4 TimeflowPointSemantic

Semantic reference to a Timeflow point.

> The reference is relative to a container and not a TimeFlow. If the container contains multiple TimeFlows, the Delphix Engine will evaluate the semantic reference with regards to all TimeFlows in that container.

container	Reference to the container. Type: Reference to Container Constraints: Required: true

location	A semantic description of a TimeFlow location. Type: string Constraints: Default: LATEST_POINT Acceptable values: LATEST_POINT, LATEST_SNAPSHOT Create: optional Update: optional

13.3.8.9.1.5 TimeflowPointLocation

TimeFlow point based on a database-specific identifier (SCN, LSN, etc).

location	The TimeFlow location. Type: string Constraints: Required: true
timeflow	Reference to TimeFlow containing this location. Type: Reference to Timeflow Constraints: Required: true

13.3.8.9.1.6 TimeflowPointBookmark

bookmark	Reference to the bookmark. Type: Reference to TimeflowBookmark Constraints: Required: true

13.3.8.9.1.7 TimeflowPointBookmarkTag

container	Reference to the container. Type: Reference to Container Constraints: Required: true
tag	The name of the tag. Type: string Constraints: Required: true

13.3.8.9.2 Timeflow API objects: timeflow, snapshot, timeflowRanges

The sample code provides information on the Timeflow, timeflowRanges and snapshot objects for the respective VDB.

Filename: flows.sh

```
$ ./flows.sh VBITT10Session API Login API Login {"type":"OKResult","status":"OK","res
ult":"USER-2","job":null,"action":null}Source: VBITT10container reference:
ORACLE_DB_CONTAINER-75 Timeflows API timeflow names:DB_PROVISION@2016-09-27T13:15:18D
B_ROLLBACK@2016-09-28T00:33:54DB_ROLLBACK@2016-09-28T00:38:33DB_ROLLBACK@2016-09-28T0
0:41:44
```

Select Timeflow Name (copy-n-paste from the above list):

```
DB_ROLLBACK@2016-09-28T00:33:54timeflow reference: ORACLE_TIMEFLOW-97
```

TimeflowRanges for this Timeflow ...


```
{    "type": "ListResult",    "status": "OK",    "result": [        {            "typ
e": "TimeflowRange",            "startPoint": {                "type":
"OracleTimeflowPoint",                "location": "5475918",                "timestam
p": "2016-09-28T04:35:39.000Z",                "timeflow": "ORACLE_TIMEFLOW-97"
      },            "endPoint": {                "type": "OracleTimeflowPoint",
          "location": "5476181",                "timestamp":
"2016-09-28T04:37:53.000Z",                "timeflow": "ORACLE_TIMEFLOW-97"
  },            "provisionable": true        }    ],    "job":null,    "action":null,
    "total": 1,    "overflow": false}
```


Snapshot per Timeflow ...

snapshots:
@2016-09-28T04:35:39.826Z
@2016-09-28T04:37:38.537Z
@2016-09-28T04:37:51.273Z

Select Snapshot Name (copy-n-paste from the above list):


```
@2016-09-28T04:37:38.537Zsnapshot reference: ORACLE_SNAPSHOT-152{    "type":
"OracleSnapshot",    "reference": "ORACLE_SNAPSHOT-152",    "namespace":null,    "nam
e": "@2016-09-28T04:37:38.537Z",    "consistency": "CRASH_CONSISTENT",    "missingNon
LoggedData": false,    "container": "ORACLE_DB_CONTAINER-75",    "creationTime":
"2016-09-28T04:37:38.537Z",    "firstChangePoint": {        "type":
"OracleTimeflowPoint",        "location": "5476152",        "timestamp":null,        "
timeflow": "ORACLE_TIMEFLOW-97"    },    "latestChangePoint": {        "type":
"OracleTimeflowPoint",        "location": "5476157",        "timestamp":
"2016-09-28T04:37:38.000Z",        "timeflow": "ORACLE_TIMEFLOW-97"    },    "retenti
on": 0,    "timeflow": "ORACLE_TIMEFLOW-97",    "timezone": "US/Eastern,EDT-0400",
 "version": "11.2.0.4.0",    "runtime": {        "type": "OracleSnapshotRuntime",
```

```
    "provisionable": true,        "missingLogs":null    },    "temporary": false,    "
fromPhysicalStandbyVdb": false,    "fractionTimeflows":null,    "redoLogSizeInBytes":
52428800}Done
```

The following scripts demonstrate REFRESH and RESET/ROLLBACK for the Timeflow types of timestamp, snapshot and scn/lsn.

Filename: vdb_refresh_timestamp.sh
Filename: vdb_refresh_snapshot.sh
Filename: vdb_refresh_scn.sh

Filename: vdb_rollback_timestamp.sh
Filename: vdb_rollback_snapshot.sh
Filename: vdb_rollback_scn.sh

Usage: ./vdb_[*all_the_above_scripts*].sh [source_vdb_name]

Sample Usage: Copy-and-Paste the timeflow name to the "Select timeflow Name:" prompt. Enter any timestamp between the startPoint and endPoint values using the format:
[yyyy] - [MM] - [dd]T[HH] : [mm] : [ss] . [SSS]Z

```
$ ./vdb_rollback_timestamp.sh VBITTSession and Login Successful ...database container
reference: ORACLE_DB_CONTAINER-73 Timeflows API Timeflow Names:DB_REFRESH@2016-09-28T
00:13:13DB_PROVISION@2016-09-26T07:00:08DB_ROLLBACK@2016-09-26T07:45:30DB_REFRESH@201
6-09-26T12:20:57DB_ROLLBACK@2016-09-27T12:29:47
```

Select Timeflow Name (copy-n-paste from above list):

```
DB_ROLLBACK@2016-09-26T07:45:30Timeflow Reference: ORACLE_TIMEFLOW-89TimeflowRanges
for this timeflow ... {    "type": "ListResult",    "status": "OK",    "result":
[        {            "type": "TimeflowRange",            "startPoint":
{                "type": "OracleTimeflowPoint",                "location": "5474012",
                "timestamp": "2016-09-26T15:30:35.000Z",                "timeflow":
"ORACLE_TIMEFLOW-89"            },            "endPoint": {                "type":
"OracleTimeflowPoint",                "location": "5482482",                "timestam
p": "2016-09-26T21:15:17.000Z",                "timeflow": "ORACLE_TIMEFLOW-89"
      },            "provisionable": true        }    ],    "job": null,    "action":
null,    "total": 1,    "overflow": false}Timestamp Format "[yyyy][MM][dd]T[HH]:[mm]:
[ss].[SSS]Z"Enter Timestamp between Start and End Point values (exclude quotes): 2016-0
9-26T15:45:00.000Zjson> {    "type": "OracleRollbackParameters",    "timeflowPointPar
ameters": {        "type": "TimeflowPointTimestamp",        "timeflow":
"ORACLE_TIMEFLOW-89",        "timestamp": "2016-09-26T15:45:00.000Z"    },    "userna
me": ""}Job: JOB-515Current status as of Wed Sep 28 00:49:39 EDT 2016 : RUNNING 0%
Completed. . .Current status as of Wed Sep 28 00:51:10 EDT 2016 : RUNNING 73%
CompletedCurrent status as of Wed Sep 28 00:51:50 EDT 2016 : RUNNING 96%
CompletedJob: JOB-515 COMPLETED 100% Completed ...Done ...
```

13.3.8.10 API Cookbook: common tasks, workflows, and examples

These topics describe approaches to common tasks and workflows using the Delphix Engine API.

This section covers the following topics:

- API Cookbook: authentication (see page 2832)
- API Cookbook: host environment details (see page 2834)
- API Cookbook: list alerts and list jobs (see page 2834)
- API Cookbook: list dSources and VDBs (see page 2835)
- API Cookbook: list snapshots (see page 2836)
- API Cookbook: example provision Of an Oracle VDB (see page 2837)
- API Cookbook: refresh VDB (see page 2838)
- API Cookbook: rewind a VDB (see page 2838)
- API Cookbook: stop/start a VDB (see page 2840)
- API Cookbook: creating a database template in Delphix self-service (see page 2842)
- API Cookbook: creating a container in Delphix self-service (see page 2846)
- API Cookbook: refreshing a container in Delphix self-service (see page 2852)
- API Cookbook: creating a user in Delphix self-service (see page 2858)
- API Cookbook: creating a branch in Delphix self-service (see page 2831)
- API Cookbook: create a bookmark in Delphix self-service (see page 2864)
- API Cookbook: delete a bookmark in Delphix self-service (see page 2870)
- API Cookbook: get a bookmark in Delphix self-service (see page 2871)
- API Cookbook: share a bookmark in Delphix self-service (see page 2872)
- API Cookbook: update a bookmark in Delphix self-service (see page 2876)
- API Cookbook: delete Delphix self-service container (see page 2877)
- API Cookbook: delete Delphix self-service template (see page 2879)
- API Cookbook: uploadUpgrade (see page 2880)

13.3.8.10.1 API cookbook: authentication

This API cookbook recipe describes how to create an authenticated session for using the Delphix Server web services.

Before you can use any Delphix Web Service APIs, you need to create a session, and then authenticate the session by providing valid Delphix account credentials.

To learn more about the latest API version for each release of the Delphix Engine, including schema changes and links to the relevant version of the schema, refer to the API version information[796] page.

> **Note**: Please replace the version number in the code snippet below with the latest API version before running this API call.

Create Delphix API Session

```
$ curl -s -X POST -k --data @- http://delphix-server/resources/json/delphix/session \
   -c ~/cookies.txt -H "Content-Type: application/json" <<EOF
```

796 https://cd.delphix.com/docs/latest/api-version-information

```
{
   "type": "APISession",
   "version": {
       "type": "APIVersion",
       "major": 1,
       "minor": 4,
       "micro": 3
   }
}
EOF

Response
{
   "status":"OK",
   "result": {
       "type":"APISession",
       "version": {
           "type": "APIVersion",
           "major": 1,
           "minor": 4,
           "micro": 3
       },
       "locale": "en_US",
       "client": null
   },
   "job": null
}
EOF
```

Once the session has been established, the next step is to authenticate to the server by executing the `LoginRequest` API. Unauthenticated sessions are prohibited from making any API calls other than this login request. The username can be either a system user or domain user, and the backend will authenticate using the appropriate method. This example illustrates logging in via curl using cookies created when the session was established:

```
$ curl -s -X POST -k --data @- http://delphix-server/resources/json/delphix/login \
-b cookies.txt -c cookies2.txt -H "Content-Type: application/json" <<EOF
{
"type": "LoginRequest",
"username": "delphix_user",
"password": "delphix_pass",
"target": "DOMAIN"
}
EOF
```

The new cookie (cookie2.txt) will need to be used in subsequent API requests. The login API currently only supports authentication by a password. There is no way to authenticate using any shared key or alternate authentication strategy.

> It is generally recommended to set the API session version to the highest level supported (see page 2832) by your Delphix Engine.

13.3.8.10.2 API cookbook: host environment details

This API cookbook recipe describes how to obtain host environment details using the Delphix Engine API.

To obtain details about target host environments, list available `Environment` objects on the system. These environments can represent either a single host, or an Oracle cluster.

List Environment

```
curl -X GET -k http://delphix-server/resources/json/delphix/environment \
    -b ~/cookies.txt -H "Content-Type: application/json"
```

For single-host environments, the reference can be used to get information about the associated host. It is also possible to get information about all hosts (regardless of whether they are in a single-host environment or cluster) by omitting the `environment` query parameter.

List UNIX Environment

```
curl -X GET -k http://services.cloud.skytap.com:23173/resources/json/delphix/host?
environment=UNIX_HOST_ENVIRONMENT-1 \
    -b ~/cookies.txt -H "Content-Type: application/json"
```

For more information about the content of host objects, see the `/api/#Host` reference on your local Delphix Engine. Depending on the type of the host, additional information may be available through the following types:

- `UnixHost`
- `WindowsHost`

13.3.8.10.3 API cookbook: list alerts and list jobs

This API cookbook recipe describes how to obtain lists of jobs and alerts using the Delphix Engine API.

The `List Alerts` and `List Jobs` API calls can both accept the `toDate` and `fromDate` query parameters to limit rows returned. These parameters require the date to be expressed in ISO 8601[797] format.

List Alerts

797 http://en.wikipedia.org/wiki/ISO_8601

```
$ curl -X GET -k http:/delphix-server/resources/json/delphix/alert \
    -b ~/cookies.txt -H "Content-Type: application/json"
```

For more information about the structure of an alert object, see the "/api/#Alert" link on your local Delphix Engine.

List Jobs (using fromDate)

```
$ curl -X GET -k http://delphix-server/resources/json/delphix/job?
addEvents=true&fromDate=2012-11-08T00:00:00.0000Z \
    -b ~/cookies.txt -H "Content-Type: application/json"
```

For more information about the structure of a job object, see the "/api/#Job" link on your local Delphix Engine.

13.3.8.10.4 API cookbook: list dSources and VDBs

This API cookbook recipe describes how to obtain a list of dSources and VDBs using the Delphix Engine API.

To obtain a list of dSources and VDBs, list available `Container` (also known as `database`) objects on the system:

List Databases

```
$ curl -X GET -k http://delphix-server/resources/json/delphix/database  \
    -b ~/cookies.txt -H "Content-Type: application/json"
```

For more information on the structure of a database object, see the `/api/#Container` reference on your local Delphix Engine. The following sub-types are available depending on the type of database:

- `OracleDatabaseContainer`
- `MSSqlDatabaseContainer`

Each database has zero or one source associated with it. This source could be a linked source, indicating that the database is a dSource, or it could be a virtual source, indicating that it is a VDB. If there are no sources, it is a detached dSource. The `parentContainer` property indicates the reference to the parent container, also indicating that the database is a VDB. To get runtime information about the source associated with the dSource or VDB, use the `Source` API with a `database` parameter set to the reference of the database in question.

List Sources

```
$ curl -X GET -k http://delphix-server/resources/json/delphix/source?
database=DB_CONTAINER-13  \
    -b ~/cookies.txt -H "Content-Type: application/json"
```

If the `virtual` flag is true, the source is a VDB, otherwise it is a dSource. For more information about the contents of a source object, see the `/api/#Source` reference on your local Delphix Engine. The following sub-types are available depending on the type of source:

- `OracleSource`
  - `OracleLinkedSource`
  - `OracleVirtualSource`
- `MSSqlSource`
  - `MSSqlLinkedSource`
  - `MSSqlVirtualSource`

13.3.8.10.5 API cookbook: list snapshots

This API cookbook recipe describes how to obtain a list of available snapshots for a VDB or dSource.

Snapshots represent points in time where a `sync` operation has occurred on either a dSource or VDB. Provisioning from snapshots is much faster than provisioning between snapshots, as the latter requires replaying LogSync records to arrive at the requested point. Given a reference to a database, the `snapshot` API can be used to retrieve the set of snapshots within the database. See the topic API Cookbook: List dSources and VDBs (see page 2835) for information on how to obtain the database reference.

List Snapshots

```
curl -X GET -k http://services.cloud.skytap.com:23173/resources/json/delphix/
snapshot?database=ORACLE_DB_CONTAINER-15  \
    -b ~/cookies.txt -H "Content-Type: application/json"
```

For more information about the structure of a snapshot object, see the `/api/#TimeflowSnapshot` reference on your local Delphix Engine. Snapshots, while representing the point where provisioning will be most efficient, are not the only provisionable points within a database. To get a list of all provisioning points, use the `timeflowRange` API. This API is based on a Timeflow, which is the representation of one timeline within a database. Currently, all databases have a single Timeflow, though this may change in the future. To query for the ranges for a particular database, you will need to use the `currentTimeflow` member of the target database.

List Timeflow Ranges

```
curl -X POST -k --data @- http://services.cloud.skytap.com:23173/resources/json/
delphix/timeflow/ORACLE_TIMEFLOW-11/timeflowRanges  \
    -b ~/cookies.txt -H "Content-Type: application/json" <<EOF
{
    "type": "TimeflowRangeParameters"
}
EOF
```

13.3.8.10.6 API cookbook: example provision of an Oracle VDB

This API cookbook recipe demonstrates how to provision an Oracle VDB using the Delphix Engine API.

In order to provision an Oracle VDB using the API, you need to provide a set of parameters of type **OracleProvisionParameters** (having already authenticated as per API Cookbook: Authentication (see page 2832)).

There are a number of parameters you will need to know:

- **Group reference** - See the list operation in "/api#group" on your Delphix Engine
- **VDB name** - The name you want the new VDB to be called
- **Mount path** - Where to mount datasets on the target host.
- **DB/unique names** - The Oracle DB and unique names, often the same as the VDB name
- **Instance name/number** - The Oracle instance name and number to use (dictated by your environment, but often VDB name and 1)
- **Repository reference** - See the list operation on "/api#repository" on your Delphix Engine
- **TimeFlow point** - See API Cookbook: List Snapshots (see page 2836) for more information on finding a TimeFlow point, as well as the reference at "/api#TimeflowPoint Parameters

You will need to use the structure of the OracleProvisionParameters object to fill it out, see "/api/#OracleProvisionParameters" for details on which fields are mandatory/optional.

Here is a minimal example using curl to communicate with the API, provisioning a VDB called "EGVDB" (authentication omitted)

```
curl -X POST -k --data @- http://delphix1.company.com/resources/json/delphix/
database/provision \
    -b cookies.txt -H "Content-Type: application/json" <<EOF
{
    "container": {
        "group": "GROUP-2",
        "name": "EGVDB",
        "type": "OracleDatabaseContainer"
    },
    "source": {
        "type": "OracleVirtualSource",
        "mountBase": "/mnt/provision",
        "allowAutoVDBRestartOnHostReboot": true
    },
    "sourceConfig": {
        "type": "OracleSIConfig",
        "databaseName": "EGVDB",
        "uniqueName": "EGVDB",
        "repository": "ORACLE_INSTALL-3",
        "instance": {
            "type": "OracleInstance",
            "instanceName": "EGVDB",
            "instanceNumber": 1
        }
    },
    "timeflowPointParameters": {
```

```
        "type": "TimeflowPointLocation",
        "timeflow": "ORACLE_TIMEFLOW-123",
        "location": "3043123"
    },
    "type": "OracleProvisionParameters"
}
EOF
```

13.3.8.10.7 API cookbook: refresh VDB

This API cookbook recipe describes how to refresh a VDB using the Delphix Engine API.

To refresh a VDB you need a reference to the `Database` object, the location of the point to which you wish to refresh and the reference container associated with the object. See the topic API Cookbook: List (see page 2835)dSources and VDBs (see page 2835) for information on how to obtain the database reference and current Timeflow. The Timeflow point can be specified either by timestamp, by location (SCN), semantic location or Timeflow bookmark. The TimeflowPointSemantic type allows you to specify a semantically meaningful Timeflow location (i.e. the latest snapshot or the latest Timeflow point). The TimeflowPointBookmark type allows you to reference a previously created Timeflow bookmark. See API Cookbook: List Snapshots (see page 2836) topic for information on how to determine provisionable points in the parent database.

Refresh VDB

```
curl -v -X POST -k --data @- http://delphix-server/resources/json/delphix/database/
ORACLE_DB_CONTAINER-13/refresh  \
    -b ~/cookies.txt -H "Content-Type: application/json" <<EOF
{
       "type": "OracleRefreshParameters",
       "timeflowPointParameters": {
               "type": "TimeflowPointSemantic",
               "container": "ORACLE_DB_CONTAINER-1",
               "timeflow": "ORACLE_TIMEFLOW-13",
               "location": "LATEST_SNAPSHOT"
       }
}
EOF
```

For more information about the content of refresh parameters, see the `/api/#RefreshParameters` reference on your local Delphix Engine. Depending on the type of the database, the following parameter types are available:

- `OracleRefreshParameters`
- `MSSqlRefreshParameters`

13.3.8.10.8 API cookbook: rewind a VDB

This API cookbook recipe describes how to rewind a VDB using the Delphix Engine API.

To rewind a VDB, you need a reference to the `Database` object. See the topic, API Cookbook: List dSources and VDBs (see page 2835), for information on how to obtain the database reference. The following sample script includes a working example for creating a session, authenticating to the Delphix Engine, and rewinding the VDB. Please update the script variables to match your environment before using it.

```
#!/bin/bash
#
# sample script to start or stop a VDB.
#
# set this to the FQDN or IP address of the Delphix Engine
DE="192.168.2.131"
# set this to the Delphix admin user name
DELPHIX_ADMIN="delphix_admin"
# set this to the password for the Delphix admin user
DELPHIX_PASS="delphix"
# set this to the object reference for the VDB
VDB="ORACLE_DB_CONTAINER-57"
#
# create our session
$ curl -s -X POST -k --data @- http://delphix-server/resources/json/delphix/session \

   -c ~/cookies.txt -H "Content-Type: application/json" <<EOF
{
   "type": "APISession",
   "version": {
       "type": "APIVersion",
       "major": 1,
       "minor": 4,
       "micro": 3
   }
}
EOF
{
   "status":"OK",
   "result": {
       "type":"APISession",
       "version": {
           "type": "APIVersion",
           "major": 1,
           "minor": 4,
           "micro": 3
       },
       "locale": "en_US",
       "client": null
   },
   "job": null
}
EOF
echo
#
# authenticate to the DE
```

```
$ curl -s -X POST -k --data @- http://delphix-server/resources/json/delphix/login \
   -b ~/cookies.txt -c ~/cookies.txt -H "Content-Type: application/json" <<EOF
{
   "type": "LoginRequest",
   "username": "delphix_username",
   "password": "delphix_password"
}

EOF
echo
#
# rewind VDB
curl -s -X POST -k --data @- http://${DE}/resources/json/delphix/database/${VDB}/
rollback \
    -b ~/cookies.txt -H "Content-Type: application/json" <<EOF
{
    "type": "OracleRollbackParameters",
    "timeflowPointParameters": {
    "type" : "TimeflowPointSnapshot",
    "snapshot" : "ORACLE_SNAPSHOT-172"
    }
}
EOF
echo
```

> While rewinding a VDB, you can use different parameter types. In the above example, "timeflowPointParameters" type is used as "TimeflowPointSnapshot" and an appropriate snapshot name is provided. Instead of "TimeflowPointSnapshot", you can also choose from "TimeflowPointLocation" or "TimeflowPointTimestamp" or "TimeflowPointBookmark" etc. and pass the relevant parameters.

You can list your Snapshots by following the instructions on API Cookbook: List Snapshots (see page 2836)

13.3.8.10.9 API cookbook: stop/start a VDB

This API cookbook recipe describes how to stop and start a VDB using the Delphix Engine API.

To stop or start a VDB, you need a reference to the `Database` object. See the topic, API Cookbook: List dSources and VDBs (see page 2835), for information on how to obtain the database reference. The following script example includes working examples for creating a session, authenticating to the Delphix Engine, and stopping or starting a VDB. Please update the script variables to match your environment before using it. This script requires a single argument which is 'start' or 'stop'.

```
#!/bin/bash
#
# sample script to start or stop a VDB.
#
```

```
# set this to the FQDN or IP address of the Delphix Engine
DE="192.168.2.131"
# set this to the Delphix admin user name
DELPHIX_ADMIN="delphix_admin"
# set this to the password for the Delphix admin user
DELPHIX_PASS="delphix"
# set this to the object reference for the VDB
VDB="ORACLE_VIRTUAL_SOURCE-5"
#
# create our session
curl -s -X POST -k --data @- http://${DE}/resources/json/delphix/session \
    -c ~/cookies.txt -H "Content-Type: application/json" <<EOF
{
    "type": "APISession",
    "version": {
        "type": "APIVersion",
        "major": 1,
        "minor": 11,
        "micro": 8
    }
}
EOF
echo
#
# authenticate to the DE
curl -s -X POST -k --data @- http://${DE}/resources/json/delphix/login \
    -c ~/cookies.txt -b ~/cookies.txt -H "Content-Type: application/json" <<EOF
{
    "type": "LoginRequest",
    "username": "${DELPHIX_ADMIN}",
    "password": "${DELPHIX_PASS}"
}
EOF
echo
#
# start or stop the vdb based on the argument passed to the script
case $1 in
start)
  curl -s -X POST -k http://${DE}/resources/json/delphix/source/${VDB}/start \
    -c ~/cookies.txt -b ~/cookies.txt -H "Content-Type: application/json"
;;
stop)
  curl -s -X POST -k http://${DE}/resources/json/delphix/source/${VDB}/stop \
    -c ~/cookies.txt -b ~/cookies.txt -H "Content-Type: application/json"
;;
*)
  echo "Unknown option: $1"
;;
esac
echo
```

13.3.8.10.10 API cookbook: creating a database template in Delphix self-service

Jet Stream is now known as Delphix Self-Service.

Delphix Self-Service administrators can use this API cookbook recipe to create a database template on Delphix Self-Service using the Delphix Engine API.

The following script is for educational and demonstration purposes only and is not supported by Delphix.

The following script can be downloaded by selecting **createDBTemplate.sh**[798]

Create Self-Service Database Template

```
#!/bin/bash

# A sample script for calls to the API. This one creates a Jet Stream Template.

##### Constants

# Describes a Delphix software revision.
# Please change version are per your Delphix Engine CLI, if different
VERSION="1.8.0"

##### Default Values. These can be overwriten with optional arguments.
engine="172.16.151.154"
username="delphix_admin"
password="landshark"

##examples##
# Create template with mandatory params
#./createBranch.sh -d 172.16.151.154 -u delphix_admin:landshark -n <sourceName>
<templateName> <containerName>
#Ex ./createBranch.sh -d 172.16.151.154 -u delphix_admin:landshark -n oraclesrc
template1 ORACLE_DB_CONTAINER-191
# Create template with adding optional params, Notes and Description
#./createDBTemplate.sh -n <sourceName> -N "<templateNotes>" -D "<AnyDescription>"
 <templateName> <containerName>

## NOTE: This script is to add one source per template and it will not add any
properties for template, container or source.
```

798 https://docs.delphix.com/docs/files/191896903/191896904/1/1652831467365/createDBTemplate+%284%29.sh

```
##### Functions

# Help Menu
function usage {
    echo "Usage: createDBTemplate.sh [[-h] | options...] <template_name>
<source_container>"
    echo "Create a Jet Stream Dat Template."
    echo ""
    echo "Positional arguments"
    echo "  <template_name>"
    echo "  <source_container>"
    echo ""
    echo "Optional Arguments:"
    echo "  -h                Show this message and exit"
    echo "  -d                Delphix engine IP address or host name, otherwise
revert to default"
    echo "  -u USER:PASSWORD  Server user and password, otherwise revert to default"
    echo "  -n                source name to display on JS template screen"
    echo "  -N                template notes, if any. Type: String"
    echo "  -D                source description, if any. Type: String"
}

# Create Our Session, including establishing the API version.
function create_session
{
    # Pulling the version into parts. The {} are necessary for string manipulation.
    # Strip out longest match following "."  This leaves only the major version.
    major=${VERSION%%.*}
    # Strip out the shortest match preceding "." This leaves minor.micro.
    minorMicro=${VERSION#*.}
    # Strip out the shortest match followint "." This leaves the minor version.
    minor=${minorMicro%.*}
    # Strip out the longest match preceding "." This leaves the micro version.
    micro=${VERSION##*.}

    # Quick note about the <<-. If the redirection operator << is followed by a -
(dash), all leading TAB from the document data will be
    # ignored. This is useful to have optical nice code also when using here-
documents. Otherwise you must have the EOF be on a line by itself,
    # no parens, no tabs or anything.

    echo "creating session..."
    result=$(curl -s -S -X POST -k --data @- http://${engine}/resources/json/delphix/
session \
        -c ~/cookies.txt -H "Content-Type: application/json" <<-EOF
    {
        "type": "APISession",
        "version": {
            "type": "APIVersion",
            "major": $major,
            "minor": $minor,
            "micro": $micro
        }
```

```
    }
    EOF)

    check_result
}

# Authenticate the DE for the provided user.
function authenticate_de
{
    echo "authenticating delphix engine..."
    echo ${engine}
    echo ${username}
    echo ${password}
    result=$(curl -s -S -X POST -k --data @- http://${engine}/resources/json/delphix/
login \
        -b ~/cookies.txt -H "Content-Type: application/json" <<-EOF
    {
        "type": "LoginRequest",
        "username": "${username}",
        "password": "${password}"
    }
    EOF)

    check_result
}

function create_template
{

   paramString="\"type\": \"JSDataTemplateCreateParameters\",\"name\":
\"$templateName\","

   if [[ -n $templatenotes ]]
    then
       paramString="$paramString \"notes\": \"$templatenotes\","
   fi

    paramString="$paramString \"dataSources\": [{\"type\":
\"JSDataSourceCreateParameters\",
                              \"container\": \"$sourceContainer\",
                              \"source\": {\"type\": \"JSDataSource\",
                              \"priority\": 1,
                              \"name\": \"$sourcename\""

   if [[ -n $sourcedesc ]]
    then
       paramString="$paramString ,\"description\": \"$sourcedesc\"}}]"
    else
       paramString="$paramString }}]"
   fi

    result=$(curl -s -X POST -k --data @- http://${engine}/resources/json/delphix/
jetstream/template \
```

```
        -b ~/cookies.txt -H "Content-Type: application/json" <<-EOF
    {
        $paramString
    }
    EOF)

    check_result

    echo "New JetStream template $templateName successfully created"

}

# Check the result of the curl. If there are problems, inform the user then exit.
function check_result
{
    exitStatus=$?
    if [ $exitStatus -ne 0 ]
    then
        echo "command failed with exit status $exitStatus"
        exit 1
    elif [[ $result != *"OKResult"* ]]
    then
        echo ""
        echo $result
        exit 1
    fi
}

##### Main

while getopts "u:d:n:N:D:h" flag; do
    case "$flag" in
        u )             username=${OPTARG%:*}
                        password=${OPTARG##*:}
                        ;;
        d )             engine=$OPTARG
                        ;;
        n )             sourcename=$OPTARG
                        ;;
        N )             templatenotes=$OPTARG
                        ;;
        D )             sourcedesc=$OPTARG
                        ;;
        h )             usage
                        exit
                        ;;
        * )             usage
                        exit 1

    esac

done
```

```
# Shift the parameters so we only have the positional arguments left
shift $((OPTIND-1))

# Check that there are 2 positional arguments
if [ $# != 2 ]
then
  echo "usage1"
    usage
    exit 1
fi

# Get the two positional arguments
templateName=$1
shift
sourceContainer=$1

create_session
authenticate_de
create_template
```

13.3.8.10.11 API cookbook: creating a container in Delphix self-service

Delphix Self-Service administrators can use this API cookbook recipe to create a container on Delphix Self-Service (Jet Stream) using the Delphix Engine API.

> The following script is for educational and demonstration purposes only and is not supported by Delphix.

This script can be downloaded by selecting

[799]

[799] https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149457600?preview=%2F149457600%2F149457613%2FcreateContainer.sh

Create Self-Service Container

```
#!/bin/bash

# A sample script for calls to the API. This one creates a Jet Stream container.

##### Constants

# Describes a Delphix software revision.
# Please change version are per your Delphix Engine CLI, if different
VERSION="1.11.10"


##### Default Values. These can be overwriten with optional arguments.
engine="10.110.248.170"
username="admin"
password="delphix"

##examples##
# Create container from latest point in time
#./createContainer.sh -n "testsource" testcont ORACLE_DB_CONTAINER-269
 JS_DATA_TEMPLATE-13
# Create container from specific bookmark
#./createContainer.sh -n "testsource" -b JS_BOOKMARK-77 testcont ORACLE_DB_CONTAINER-2
69 JS_DATA_TEMPLATE-13
# Create container from specific point in time
#./createContainer.sh -n "testsource" -t "2016-08-08T10:00:00.000Z" -B JS_BRANCH-50
 testcont ORACLE_DB_CONTAINER-269 JS_DATA_TEMPLATE-13

#NOTE: this script will add one container and assign one owner for the container.

##### Functions

# Help Menu
function usage {
    echo "Usage: createContainer.sh [[-h] | options...] <containername> <vdb>
<template>"
    echo "Create a Jet Stream Bookmark on the given branch."
    echo ""
    echo "Positional arguments"
    echo "  <name>"
    echo "  <container> format JS_DATA_CONTAINER-<n>"
    echo ""
    echo "Optional Arguments:"
    echo "  -h                Show this message and exit"
    echo "  -d                Delphix engine IP address or host name, otherwise
revert to default"
    echo "  -u USER:PASSWORD  Server user and password, otherwise revert to default"
    echo "  -n                SourceName need to display for container.(Mandatory)"
    echo "  -b                Bookmark name from which need to create container. If
no bookmark is included, the branch will be created at the latest point in time.
Type: string. Format JS_BOOKMARK-<n> (Optional)"
```

```
    echo "  -t                The time at which the branch should be created. This
must be accompanied with branch name from which need to pick up time. Type: date,
must be in ISO 8601 extended format [yyyy]-[MM]-[dd]T[HH]:[mm]:[ss].[SSS]Z"
    echo "  -B                Branch name from which need to create new container, at
specific time. Type: string. Format JS_BRANCH-<n> (Optional)"
    echo "  -N                Optional container notes, if need to add any. Type:
String"
    echo "  -o                Optional owner, to whom we need to assign this
container. Type: String. Format USER-<n>"
}

# Create Our Session, including establishing the API version.
function create_session
{
    # Pulling the version into parts. The {} are necessary for string manipulation.
    # Strip out longest match following "."  This leaves only the major version.
    major=${VERSION%%.*}
    # Strip out the shortest match preceding "." This leaves minor.micro.
    minorMicro=${VERSION#*.}
    # Strip out the shortest match followint "." This leaves the minor version.
    minor=${minorMicro%.*}
    # Strip out the longest match preceding "." This leaves the micro version.
    micro=${VERSION##*.}

    # Quick note about the <<-. If the redirection operator << is followed by a -
(dash), all leading TAB from the document data will be
    # ignored. This is useful to have optical nice code also when using here-
documents. Otherwise you must have the EOF be on a line by itself,
    # no parens, no tabs or anything.

    echo "creating session..."
    result=$(curl -s -S -X POST -k --data @- http://${engine}/resources/json/delphix/
session \
        -c ~/cookies.txt -H "Content-Type: application/json" <<-EOF
    {
        "type": "APISession",
        "version": {
            "type": "APIVersion",
            "major": $major,
            "minor": $minor,
            "micro": $micro
        }
    }
    EOF)

    check_result
}

# Authenticate the DE for the provided user.
function authenticate_de
{
    echo "authenticating delphix engine..."
```

```
    result=$(curl -s -S -X POST -k --data @- http://${engine}/resources/json/delphix/
login \
        -b ~/cookies.txt -c ~/cookies.txt -H "Content-Type: application/json" <<-EOF
    {
        "type": "LoginRequest",
        "username": "${username}",
        "password": "${password}"
    }
    EOF)

    check_result
}

function create_container
{

    # If there is not timeInput and no bookmark name, we need to use
JSTimelinePointLatestTimeInput.
    if [[ -z $inputTime  &&  -z $bookmark ]]
    then
        pointParams="\"timelinePointParameters\":{
            \"sourceDataLayout\": \"${template}\",
            \"type\":\"JSTimelinePointLatestTimeInput\"}"

   # If there is a timeInput and no bookmark name, we need to use Input Time.

    elif [[ -n $inputTime  && -n $branchRef && -z $bookmark ]]
     then
        pointParams="\"timelinePointParameters\":{
            \"time\":\"${inputTime}\",
            \"branch\":\"${branchRef}\",
            \"type\":\"JSTimelinePointTimeInput\"}"

   # If there is a bookmark name and no time input, we need to use bookmark

   elif [[ -z $inputTime  &&  -n $bookmark ]]
   then
        pointParams="\"timelinePointParameters\":{
        \"bookmark\":\"${bookmark}\",
        \"type\":\"JSTimelinePointBookmarkInput\"}"

    fi

    # These are the required parameters.

    paramString="\"type\": \"JSDataContainerCreateWithRefreshParameters\",
                 \"name\": \"${containerName}\",
                 \"template\": \"${template}\","


    paramString="$paramString \"dataSources\": [{\"type\":
\"JSDataSourceCreateParameters\",
                \"container\": \"${VDB}\",
```

```
                \"source\": {
                \"type\": \"JSDataSource\",
                \"priority\": 1,
                \"name\": \"${sourceName}\""

    if [[ -n $sourcedesc ]]
    then
       paramString="$paramString ,\"description\": \"${sourcedesc}\"}}],"
    else
       paramString="$paramString }}],"
   fi

    if [[ -n $containerNotes ]]
    then
       paramString="$paramString \"notes\": \"${containerNotes}\","
   fi

   if [[ -n $owners ]]
   then
      paramString="$paramString \"owners\": [\"${owners}\"],"
   fi

    paramString="$paramString ${pointParams}"


    result=$(curl -s -X POST -k --data @- http://${engine}/resources/json/delphix/
jetstream/container \
        -b ~/cookies.txt -H "Content-Type: application/json" <<-EOF
    {
        $paramString
    }
    EOF)


    check_result


    echo "confirming job completed successfully..."
    # Get everything in the result that comes after job.
    temp=${result#*\"job\":\"}
    # Get rid of everything after
    jobRef=${temp%%\"*}


    result=$(curl -s -X GET -k http://${engine}/resources/json/delphix/job/${jobRef}
\
    -b ~/cookies.txt -H "Content-Type: application/json")

    # Get everything in the result that comes after job.
    temp=${result#*\"jobState\":\"}
    # Get rid of everything after
    jobState=${temp%%\"*}
```

```
    check_result

    while [ $jobState = "RUNNING" ]
    do
        sleep 1
        result=$(curl -s -X GET -k http://${engine}/resources/json/delphix/job/$
{jobRef} \
        -b ~/cookies.txt -H "Content-Type: application/json")

        # Get everything in the result that comes after job.
        temp=${result#*\"jobState\":\"}
        # Get rid of everything after
        jobState=${temp%%\"*}

        check_result

    done

    if [ $jobState = "COMPLETED" ]
    then
        echo "successfully created container $containerName"
    else
        echo "unable to create container"
        echo result
    fi

}

# Check the result of the curl. If there are problems, inform the user then exit.
function check_result
{
    exitStatus=$?
    if [ $exitStatus -ne 0 ]
    then
        echo "command failed with exit status $exitStatus"
        exit 1
    elif [[ $result != *"OKResult"* ]]
    then
        echo ""
        echo $result
        exit 1
    fi
}

##### Main

while getopts "u:d:b:t:B:D:n:N:o:h" flag; do
    case "$flag" in
        u )             username=${OPTARG%:*}
                        password=${OPTARG##*:}
                        ;;
        d )             engine=$OPTARG
```

```
                        ;;
        b )             bookmark=$OPTARG
                        ;;
        t )             inputTime=$OPTARG
                        ;;
        B )             branchRef=$OPTARG
                        ;;
        D )             sourcedesc=$OPTARG
                        ;;
        n )             sourceName=$OPTARG
                        ;;
        N )             containerNotes=$OPTARG
                        ;;
        o )             owners=$OPTARG
                        ;;
        h )             usage
                        exit
                        ;;
        * )             usage
                        exit 1
    esac

done


# Shift the parameters so we only have the positional arguments left
shift $((OPTIND-1))

# Check that there are 3 positional arguments
if [ $# != 3 ]
then
    usage
    exit 1
fi

# Get the three positional arguments
containerName=$1
shift
VDB=$1
shift
template=$1

create_session
authenticate_de
create_container
```

13.3.8.10.12 API cookbook: refreshing a container in Delphix self-service

Delphix Self-Service administrators can use this API cookbook recipe to refresh a container in Delphix Self-Service (Jet Stream) using the Delphix Engine API.

> The following script is for educational and demonstration purposes only and is not supported by Delphix.

This script can be downloaded by selecting refreshContainer 2.sh[800].

Refreshing a Self-Service container

```
#!/bin/bash

# A sample script for calls to the API. This one refresh Jet Stream container.

##### Constants

# Describes a Delphix software revision.
# Please change version are per your Delphix Engine CLI, if different
VERSION="1.8.0"


##### Default Values. These can be overwriten with optional arguments.
engine="172.16.151.154"
username="delphix_admin"
password="landshark"

##examples##
# Refresh container from latest point in time of Template
#./refreshContainer.sh -T JS_DATA_TEMPLATE-13 JS_DATA_CONTAINER-20
# Refresh container from specific bookmark
#./refreshContainer.sh -b JS_BOOKMARK-76 JS_DATA_CONTAINER-20
# Refresh container from specific point in time of branch
#./refreshContainer.sh -t "2016-08-08T10:00:00.000Z" -B JS_BRANCH-50
 JS_DATA_CONTAINER-20
```

800 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149523770?preview=%2F149523770%2F149523781%2FrefreshContainer+2.sh

```
##### Functions

# Help Menu
function usage {
    echo "Usage: refreshContainer.sh [[-h] | options...] <containername> <template>"
    echo "Create a Jet Stream Bookmark on the given branch."
    echo ""
    echo "Positional arguments"
    echo "  <containerName>"
    echo "  <template>"
    echo ""
    echo "Optional Arguments:"
    echo "  -h                Show this message and exit"
    echo "  -d                Delphix engine IP address or host name, otherwise
revert to default"
    echo "  -u USER:PASSWORD  Server user and password, otherwise revert to default"
    echo "  -T                template reference from which need to refresh from
latest point in time"
    echo "  -b                Bookmark name from which need to refresh container. If
no bookmark is included, the branch will be created at the latest point in time.
Type: string. Format JS_BOOKMARK-<n> (Optional)"
    echo "  -t                The time from where the container should be refreshed.
This must be accompanied with branch name from which need to pick up time. Type:
date, must be in ISO 8601 extended format [yyyy]-[MM]-[dd]T[HH]:[mm]:[ss].[SSS]Z"
    echo "  -B                Branch name from which need to refresh container, at
specific time. Type: string. Format JS_BRANCH-<n> (Optional)"
}

# Create Our Session, including establishing the API version.
function create_session
{
    # Pulling the version into parts. The {} are necessary for string manipulation.
    # Strip out longest match following "."  This leaves only the major version.
    major=${VERSION%%.*}
    # Strip out the shortest match preceding "." This leaves minor.micro.
    minorMicro=${VERSION#*.}
    # Strip out the shortest match followint "." This leaves the minor version.
    minor=${minorMicro%.*}
    # Strip out the longest match preceding "." This leaves the micro version.
    micro=${VERSION##*.}

    # Quick note about the <<-. If the redirection operator << is followed by a -
(dash), all leading TAB from the document data will be
    # ignored. This is useful to have optical nice code also when using here-
documents. Otherwise you must have the EOF be on a line by itself,
    # no parens, no tabs or anything.

    echo "creating session..."
    result=$(curl -s -S -X POST -k --data @- http://${engine}/resources/json/delphix/
session \
        -c ~/cookies.txt -H "Content-Type: application/json" <<-EOF
    {
        "type": "APISession",
```

```
        "version": {
            "type": "APIVersion",
            "major": $major,
            "minor": $minor,
            "micro": $micro
        }
    }
    EOF)

    check_result
}

# Authenticate the DE for the provided user.
function authenticate_de
{
    echo "authenticating delphix engine..."
    result=$(curl -s -S -X POST -k --data @- http://${engine}/resources/json/delphix/
login \
        -b ~/cookies.txt -c ~/cookies.txt -H "Content-Type: application/json" <<-EOF
    {
        "type": "LoginRequest",
        "username": "${username}",
        "password": "${password}"
    }
    EOF)

    check_result
}

function restore_container
{

    # If there is not timeInput and no bookmark name, we need to use
JSTimelinePointLatestTimeInput from template.
    if [[ -n $template && -z $inputTime  &&  -z $bookmark ]]
    then
        pointParams="\"type\": \"JSTimelinePointLatestTimeInput\",
                     \"sourceDataLayout\": \"${template}\""

   # If there is a timeInput and no bookmark name, we need to use Input Time.

    elif [[ -n $inputTime  && -n $branchRef && -z $bookmark && -z $template ]]
     then
        pointParams="\"type\": \"JSTimelinePointTimeInput\",
                 \"branch\": \"${branchRef}\",
                 \"time\": \"${inputTime}\""

   # If there is a bookmark name and no time input, we need to use bookmark

   elif [[ -n $bookmark && -z $template && -z $inputTime ]]
   then
        pointParams="\"type\": \"JSTimelinePointBookmarkInput\",
                     \"bookmark\": \"${bookmark}\""
```

```
    fi

    echo "pointParams" $pointParams



    result=$(curl -s -X POST -k --data @- http://${engine}/resources/json/delphix/
jetstream/container/${containerRef}/restore \
        -b ~/cookies.txt -H "Content-Type: application/json" <<-EOF
    {
        $pointParams
    }
    EOF)


    check_result


    echo "confirming job completed successfully..."
    # Get everything in the result that comes after job.
    temp=${result#*\"job\":\"}
    # Get rid of everything after
    jobRef=${temp%%\"*}


    result=$(curl -s -X GET -k http://${engine}/resources/json/delphix/job/${jobRef}
\
    -b ~/cookies.txt -H "Content-Type: application/json")

    # Get everything in the result that comes after job.
    temp=${result#*\"jobState\":\"}
    # Get rid of everything after
    jobState=${temp%%\"*}


    check_result

    while [ $jobState = "RUNNING" ]
    do
        sleep 1
        result=$(curl -s -X GET -k http://${engine}/resources/json/delphix/job/$
{jobRef} \
        -b ~/cookies.txt -H "Content-Type: application/json")

        # Get everything in the result that comes after job.
        temp=${result#*\"jobState\":\"}
        # Get rid of everything after
        jobState=${temp%%\"*}

        check_result

    done
```

```
    if [ $jobState = "COMPLETED" ]
    then
        echo "successfully refresh container $containerName"
    else
        echo "unable to refresh container"
        echo result
    fi

}

# Check the result of the curl. If there are problems, inform the user then exit.
function check_result
{
    exitStatus=$?
    if [ $exitStatus -ne 0 ]
    then
        echo "command failed with exit status $exitStatus"
        exit 1
    elif [[ $result != *"OKResult"* ]]
    then
        echo ""
        echo $result
        exit 1
    fi
}

##### Main

while getopts "u:d:T:b:t:B:h" flag; do
    case "$flag" in
        u )             username=${OPTARG%:*}
                        password=${OPTARG##*:}
                        ;;
        d )             engine=$OPTARG
                        ;;
        T )             template=$OPTARG
                        ;;
        b )             bookmark=$OPTARG
                        ;;
        t )             inputTime=$OPTARG
                        ;;
        B )             branchRef=$OPTARG
                        ;;
        h )             usage
                        exit
                        ;;
        * )             usage
                        exit 1
    esac

done
```

```
# Shift the parameters so we only have the positional arguments left
shift $((OPTIND-1))

# Check that there are 1 positional arguments
if [ $# != 1 ]
then
    usage
    exit 1
fi

# Get the one positional arguments
containerRef=$1

create_session
authenticate_de
restore_container
```

13.3.8.10.13 API cookbook: creating a user in Delphix self-service

Delphix Self-Service administrators can use this API cookbook recipe to create a user on Delphix Self-Service (Jet Stream) using the Delphix Engine API.

> The following script is for educational and demonstration purposes only and is not supported by Delphix.

This script can be downloaded by selecting createJSUser 2.sh[801].

Creating a Self-Service User

```
#!/bin/bash
```

801 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149491242?preview=%2F149491242%2F149491268%2FcreateJSUser+2.sh

```
# A sample script for calls to the API. This one creates a Jet Stream user.

##### Constants

# Describes a Delphix software revision.
# Please change version are per your Delphix Engine CLI, if different
VERSION="1.11.9"


##### Default Values. These can be overwriten with optional arguments.
engine="10.43.90.86"
username="admin"
password="delphix"

##examples##
# Create user with NATIVE authentication
#./createJSUser.sh -P <password> NATIVE <username>
# Create user with LDAP authentication
#./createJSUser.sh -r <principal> <LDAP username>

##### Functions

# Help Menu
function usage {
echo "Usage: createJSUser.sh [[-h] | options...] <auth> <newjsuser>"
echo "Create a Jet Stream Only user."
echo ""
echo "Positional arguments"
echo " <auth>"
echo " <newjsuser>"
echo ""
echo "Optional Arguments:"
echo " -h Show this message and exit"
echo " -d Delphix engine IP address or host name, otherwise revert to default"
echo " -u USER:PASSWORD Server user and password, otherwise revert to default"
echo " -P password for NATIVE authentication"
echo " -f firstName of user"
echo " -l lastName of user"
echo " -e emailAddress of user"
echo " -o homePhoneNumber of user"
echo " -m mobilePhoneNumber of user"
echo " -w workPhoneNumber of user"
echo " -r principal for LDAP authentication"
}

# Create Our Session, including establishing the API version.
function create_session
{
# Pulling the version into parts. The {} are necessary for string manipulation.
# Strip out longest match following "." This leaves only the major version.
major=${VERSION%%.*}
# Strip out the shortest match preceding "." This leaves minor.micro.
```

```
minorMicro=${VERSION#*.}
# Strip out the shortest match followint "." This leaves the minor version.
minor=${minorMicro%.*}
# Strip out the longest match preceding "." This leaves the micro version.
micro=${VERSION##*.}

# Quick note about the <<-. If the redirection operator << is followed by a - (dash),
all leading TAB from the document data will be
# ignored. This is useful to have optical nice code also when using here-documents.
Otherwise you must have the EOF be on a line by itself,
# no parens, no tabs or anything.

echo "creating session..."
result=$(curl -s -S -X POST -k --data @- http://${engine}/resources/json/delphix/
session \
-c ~/cookies.txt -H "Content-Type: application/json" <<-EOF
{
"type": "APISession",
"version": {
"type": "APIVersion",
"major": $major,
"minor": $minor,
"micro": $micro
}
}
EOF)

check_result
}

# Authenticate the DE for the provided user.
function authenticate_de
{
echo "authenticating delphix engine..."
result=$(curl -s -S -X POST -k --data @- http://${engine}/resources/json/delphix/
login \
-b ~/cookies.txt -c ~/cookies.txt -H "Content-Type: application/json" <<-EOF
{
"type": "LoginRequest",
"username": "${username}",
"password": "${password}"
}
EOF)

check_result
}

function create_user
{
# Check on authorization type

if [[ $authtype = "NATIVE" && -n $userpwd ]]
then
```

```
pointParams="\"authenticationType\":\"$authtype\",
\"credential\":{
\"type\":\"PasswordCredential\",
\"password\":\"$userpwd\"}"

elif [[ $authtype = "LDAP" && -n $principal ]]
then
pointParams="\"authenticationType\":\"$authtype\",
\"principal\":\"$principal\""

fi

# These are the required parameters.
paramString="
\"type\": \"User\",
\"name\": \"${newjsuser}\","


# Fill in optional parameters if there are any.
if [[ -n $firstname ]]
then
paramString="$paramString \"firstName\": \"$firstname\","
fi

if [[ -n $lastname ]]
then
paramString="$paramString \"lastName\": \"$lastname\","
fi

if [[ -n $emailaddress ]]
then
paramString="$paramString \"emailAddress\": \"$emailaddress\","
fi

if [[ -n $homephone ]]
then
paramString="$paramString \"homePhoneNumber\": \"$homephone\","
fi

if [[ -n $mobilephone ]]
then
paramString="$paramString \"mobilePhoneNumber\": \"$mobilephone\","
fi

if [[ -n $workphone ]]
then
paramString="$paramString \"workPhoneNumber\": \"$workphone\","
fi


paramString="$paramString
${pointParams}"
```

```
result=$(curl -s -X POST -k --data @- http://${engine}/resources/json/delphix/user \
-b ~/cookies.txt -H "Content-Type: application/json" <<-EOF
{
$paramString
}
EOF)

check_result

# Extracting USER ID from result
temp=${result#*\"result\":\"}
userRef=${temp%%\"*}

echo "New user $newjsuser successfully created"

##### ROLE-3 is Jet Stream Role

result=$(curl -s -X POST -k --data @- http://${engine}/resources/json/delphix/
authorization \
-b ~/cookies.txt -H "Content-Type: application/json" <<-EOF
{
"type": "Authorization",
"role": "ROLE-3",
"target": "$userRef",
"user": "$userRef"
}
EOF)

check_result

echo "Assigned Jet Stream Role to user $newjsuser"

}

# Check the result of the curl. If there are problems, inform the user then exit.
function check_result
{
exitStatus=$?
if [ $exitStatus -ne 0 ]
then
echo "command failed with exit status $exitStatus"
exit 1
elif [[ $result != *"OKResult"* ]]
then
echo ""
echo $result
exit 1
fi
}

##### Main

while getopts "u:d:P:r:f:l:e:o:m:w:h" flag; do
```

```
case "$flag" in
u ) username=${OPTARG%:*}
password=${OPTARG##*:}
;;
d ) engine=$OPTARG
;;
P ) userpwd=$OPTARG
;;
r ) principal=$OPTARG
;;
f ) firstname=$OPTARG
;;
l ) lastname=$OPTARG
;;
e ) emailaddress=$OPTARG
;;
o ) homephone=$OPTARG
;;
m ) mobilephone=$OPTARG
;;
w ) workphone=$OPTARG
;;
h ) usage
exit
;;
* ) usage
exit 1

esac

done


# Shift the parameters so we only have the positional arguments left
shift $((OPTIND-1))

# Check that there are 2 positional arguments
if [ $# != 2 ]
then
echo "usage1"
usage
exit 1
fi

# Get the two positional arguments
authtype=$1
shift
newjsuser=$1

create_session
authenticate_de
create_user
```

13.3.8.10.14 API cookbook: create a bookmark in Delphix self-service

Delphix Self-Service administrators can use this API cookbook recipe to create a branch on Delphix Self-Service (Jet Stream) using the Delphix Engine API.

> The following script is for educational and demonstration purposes only and is not supported by Delphix.

This script can be downloaded by selecting **createBranch.sh.**

Create Self-Service Branch

```
#!/bin/bash

# A sample script for calls to the API. This one creates a Jet Stream bookmark.

##### Constants

# Describes a Delphix software revision.
# Please change version are per your Delphix Engine CLI, if different
VERSION="1.8.0"


##### Default Values. These can be overwriten with optional arguments.
engine="172.16.151.154"
username="delphix_admin"
password="landshark"

shared=false

##example##
#./createBookmark.sh -d 172.16.151.154 -u delphix_admin:landshark -p "bookmark test"
 -e "2016-07-27T23:38:56.453Z" -t "2016-07-27T01:45:56.453Z" -T
"[tag1,tag2,tag3,tag4,tag5]" bkmrk3 JS_BRANCH-41


##### Functions

# Help Menu
function usage {
    echo "Usage: createBookmark.sh [[-h] | options...] <name> <branch>"
    echo "Create a Jet Stream Bookmark on the given branch."
    echo ""
    echo "Positional arguments"
    echo "  <name>"
    echo "  <branch>"
    echo ""
    echo "Optional Arguments:"
```

```
    echo "  -h                Show this message and exit"
    echo "  -d                Delphix engine IP address or host name, otherwise
revert to default"
    echo "  -u USER:PASSWORD  Server user and password, otherwise revert to default"
    echo "  -D                Description of this bookmark. Type: string"
    echo "  -e                A policy will automatically delete this bookmark at
this time. If not present the bookmark will be kept until manually deleted. Type:
date, must be in ISO 8601 extended format [yyyy]-[MM]-[dd]T[HH]:[mm]:[ss].[SSS]Z"
    echo "  -s                Present if need to make bookmark in shared mode"
    echo "  -t                The time at which the bookmark should be created. If no
time is included, the bookmark will be created at the latest point in time. Type:
date, must be in ISO 8601 extended format [yyyy]-[MM]-[dd]T[HH]:[mm]:[ss].[SSS]Z"
    echo "  -T                A set of user-defined labels for this bookmark. No
spaces allowed. Array of Type: string. In format, [tag1,tag2,..] "
}

# Create Our Session, including establishing the API version.
function create_session
{
    # Pulling the version into parts. The {} are necessary for string manipulation.
    # Strip out longest match following "."  This leaves only the major version.
    major=${VERSION%%.*}
    # Strip out the shortest match preceding "." This leaves minor.micro.
    minorMicro=${VERSION#*.}
    # Strip out the shortest match followint "." This leaves the minor version.
    minor=${minorMicro%.*}
    # Strip out the longest match preceding "." This leaves the micro version.
    micro=${VERSION##*.}

    # Quick note about the <<-. If the redirection operator << is followed by a -
(dash), all leading TAB from the document data will be
    # ignored. This is useful to have optical nice code also when using here-
documents. Otherwise you must have the EOF be on a line by itself,
    # no parens, no tabs or anything.

    echo "creating session..."
    result=$(curl -s -S -X POST -k --data @- http://${engine}/resources/json/delphix/
session \
        -c ~/cookies.txt -H "Content-Type: application/json" <<-EOF
    {
        "type": "APISession",
        "version": {
            "type": "APIVersion",
            "major": $major,
            "minor": $minor,
            "micro": $micro
        }
    }
    EOF)

    check_result
}
```

```
# Authenticate the DE for the provided user.
function authenticate_de
{
    echo "authenticating delphix engine..."
    result=$(curl -s -S -X POST -k --data @- http://${engine}/resources/json/delphix/
login \
        -b ~/cookies.txt -c ~/cookies.txt -H "Content-Type: application/json" <<-EOF
    {
        "type": "LoginRequest",
        "username": "${username}",
        "password": "${password}"
    }
    EOF)

    check_result
}

# Get the branch info so the bookmark to fill in dataLayout
function get_branch
{
    echo "retrieveing branch $branchRef to find Source Data Layout..."
    result=$(curl -s -X GET -k http://${engine}/resources/json/delphix/jetstream/
branch/${branchRef} \
    -b ~/cookies.txt -H "Content-Type: application/json")

    check_result

    # Get everything in the result that comes after dataLayout.
    temp=${result#*\"dataLayout\":\"}
    # Get rid of everything after creat
    dataLayout=${temp%%\"*}

    echo "temp" $temp

    echo "dataLayout" $dataLayout
}

function create_bookmark
{
    get_branch

    # If there is not creation time, we need to use JSTimelinePointLatestTimeInput.
    if [ -z $creationTime ]
    then
        pointParams="\"timelinePointParameters\":{
            \"sourceDataLayout\": \"$dataLayout\",
            \"type\":\"JSTimelinePointLatestTimeInput\"}"

    else
        pointParams="\"timelinePointParameters\":{
            \"sourceDataLayout\": \"$dataLayout\",
            \"time\":\"$creationTime\",
            \"branch\":\"$branchRef\",
```

```
            \"type\":\"JSTimelinePointTimeInput\"}"
    fi

    # These are the required parameters.
    paramString="
        \"bookmark\": {
            \"branch\": \"${branchRef}\",
            \"name\": \"${bookmarkName}\","

    # Fill in optional parameters if there are any.
    if [[ -n $description ]]
    then
        paramString="$paramString \"description\": \"$description\","
    fi

    if [[ -n $expiration ]]
    then
        paramString="$paramString \"expiration\": \"$expiration\","
    fi

    if [[ -n $shared ]]
    then
        paramString="$paramString \"shared\": $shared,"
    fi

    if [[ -n $tags ]]
    then
        # Add quotes back to the passed in tags so they are processed correctly.
        tags=${tags//[/[\"}
        tags=${tags//,/\",\"}
        tags=${tags//]/\"]}

        paramString="$paramString \"tags\": $tags,"
    fi

    paramString="$paramString \"type\": \"JSBookmark\"
        },
        ${pointParams},
        \"type\": \"JSBookmarkCreateParameters\""

    result=$(curl -s -X POST -k --data @- http://${engine}/resources/json/delphix/
jetstream/bookmark \
        -b ~/cookies.txt -H "Content-Type: application/json" <<-EOF
    {
        $paramString
    }
    EOF)

    check_result

    echo "confirming job completed successfully..."
    # Get everything in the result that comes after job.
    temp=${result#*\"job\":\"}
```

```
    # Get rid of everything after
    jobRef=${temp%%\"*}


    result=$(curl -s -X GET -k http://${engine}/resources/json/delphix/job/${jobRef}
\
    -b ~/cookies.txt -H "Content-Type: application/json")

    # Get everything in the result that comes after job.
    temp=${result#*\"jobState\":\"}
    # Get rid of everything after
    jobState=${temp%%\"*}

    check_result

    while [ $jobState = "RUNNING" ]
    do
        sleep 1
        result=$(curl -s -X GET -k http://${engine}/resources/json/delphix/job/$
{jobRef} \
        -b ~/cookies.txt -H "Content-Type: application/json")

        # Get everything in the result that comes after job.
        temp=${result#*\"jobState\":\"}
        # Get rid of everything after
        jobState=${temp%%\"*}

        check_result

    done

    if [ $jobState = "COMPLETED" ]
    then
        echo "successfully created bookmark $bookmarkName"
    else
        echo "unable to create bookmark"
        echo result
    fi

}

# Check the result of the curl. If there are problems, inform the user then exit.
function check_result
{
    exitStatus=$?
    if [ $exitStatus -ne 0 ]
    then
        echo "command failed with exit status $exitStatus"
        exit 1
    elif [[ $result != *"OKResult"* ]]
    then
        echo ""
        echo $result
```

```
        exit 1
    fi
}

##### Main

while getopts "u:d:D:e:s:t:T:h" flag; do
    case "$flag" in
        u )             username=${OPTARG%:*}
                        password=${OPTARG##*:}
                        ;;
        d )             engine=$OPTARG
                        ;;
        D )             description=$OPTARG
                        ;;
        e )             expiration=$OPTARG
                        ;;
        s )             shared=true
                        ;;
        t )             creationTime=$OPTARG
                        ;;
        T )             tags=$OPTARG
                        ;;
        h )             usage
                        exit
                        ;;
        * )             usage
                        exit 1
    esac

echo "OPTARG" $OPTARG ####

done


# Shift the parameters so we only have the positional arguments left
shift $((OPTIND-1))

# Check that there are 2 positional arguments
if [ $# != 2 ]
then
    usage
    exit 1
fi

# Get the two positional arguments
bookmarkName=$1
shift
branchRef=$1

create_session
authenticate_de
create_bookmark
```

13.3.8.10.15 API cookbook: delete a bookmark in Delphix self-service

This API cookbook recipe describes how to delete a bookmark in Delphix Self-Service (Jet Stream).

> The following script is for educational and demonstration purposes only and is not supported by Delphix.

Deleting a bookmark in Self-Service

```
#!/bin/bash
#
# sample script to delete a bookmark on a Jet Stream container.
#
# Please set the following variables to suit your purposes.
# set this to the FQDN or IP address of the Delphix Engine
DE="ars-dlpx-6010-3.dlpxdc.co"
# set this to the Delphix admin user name
DELPHIX_ADMIN="admin"
# set this to the password for the Delphix admin user
DELPHIX_PASS="delphix"
# reference of bookmark you want to delete
BOOKMARK_REF="JS_BOOKMARK-2"
#
# create our session
curl -s -X POST -k --data @- http://${DE}/resources/json/delphix/session \
    -c ~/cookies.txt -H "Content-Type: application/json" <<EOF
{
    "type": "APISession",
    "version": {
        "type": "APIVersion",
        "major": 1,
        "minor": 6,
        "micro": 2
    }
}
EOF
echo
#
# authenticate to the DE
curl -s -X POST -k --data @- http://${DE}/resources/json/delphix/login \
    -b ~/cookies.txt -c ~/cookies.txt -H "Content-Type: application/json" <<EOF
{
    "type": "LoginRequest",
    "username": "${DELPHIX_ADMIN}",
    "password": "${DELPHIX_PASS}"
}
EOF
```

```
echo
#
# delete the bookmark
curl -s -X DELETE -k http://${DE}/resources/json/delphix/jetstream/bookmark/$
{BOOKMARK_REF} \
    -b ~/cookies.txt -H "Content-Type: application/json"
echo
```

13.3.8.10.16 API cookbook: get a bookmark in Delphix self-service

This API cookbook recipe describes how to get a bookmark (see page 2871) in Delphix Self-Service (Jet Stream).

> The following script is for educational and demonstration purposes only and is not supported by Delphix.

```
#!/bin/bash
#
# sample script to get a bookmark on a Jet Stream container.
#
# Please set the following variables to suit your purposes.
# set this to the FQDN or IP address of the Delphix Engine
DE="110.110.200.107"
# set this to the Delphix admin user name
DELPHIX_ADMIN="admin"
# set this to the password for the Delphix admin user
DELPHIX_PASS="delphix"
# reference of bookmark you want to get
BOOKMARK_REF="JS_BOOKMARK-2"
#
# create our session
curl -s -X POST -k --data @- http://${DE}/resources/json/delphix/session \
    -c ~/cookies.txt -H "Content-Type: application/json" <<EOF
{
    "type": "APISession",
    "version": {
        "type": "APIVersion",
        "major": 1,
        "minor": 6,
        "micro": 2
    }
}
EOF
echo
#
# authenticate to the DE
curl -s -X POST -k --data @- http://${DE}/resources/json/delphix/login \
    -b ~/cookies.txt -c ~/cookies.txt -H "Content-Type: application/json" <<EOF
```

```
{
    "type": "LoginRequest",
    "username": "${DELPHIX_ADMIN}",
    "password": "${DELPHIX_PASS}"
}
EOF
echo
#
# get the bookmark
curl -s -X GET -k http://${DE}/resources/json/delphix/jetstream/bookmark/$
{BOOKMARK_REF} \
    -b ~/cookies.txt -H "Content-Type: application/json"
echo
```

13.3.8.10.17 API cookbook: share a bookmark in Delphix self-service

Delphix Self-Service administrators can use this API cookbook recipe to share a bookmark in Delphix Self-Service (Jet Stream) using the Delphix Engine API.

> The following script is for educational and demonstration purposes only and is not supported by Delphix.

This script can be downloaded by selecting shareBookmark 2.sh [802]

```
#!/bin/bash

# A sample script for calls to the API. This one shares Bookmark across containers in
same template.
```

802 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149457558?preview=%2F149457558%2F149457569%2FshareBookmark+2.sh

```
##### Constants

# Describes a Delphix software revision.
# Please change version are per your Delphix Engine CLI, if different.
VERSION="1.11.10"


##### Default Values. These can be overwriten with optional arguments.
engine="ars-dlpx-6010-3.dlpxdc.co"
username="admin"
password="delphix"

##examples##
# Share Bookmark
#./shareBookmark.sh -a share JS_BOOKMARK-75
# Unshare Bookmark
#./shareBookmark.sh -a unshare JS_BOOKMARK-75

##### Functions

# Help Menu
function usage {
    echo "Usage: shareBookmark.sh [[-h] | options...] <bookmarkName>"
    echo "Share/Unshare JetStream bookmark"
    echo ""
    echo "Positional arguments"
    echo "bookmarkName. Format: JS_BOOKMARK-<n>"
    echo ""
    echo "Optional Arguments:"
    echo "  -h                Show this message and exit"
    echo "  -d                Delphix engine IP address or host name, otherwise
revert to default"
    echo "  -u USER:PASSWORD  Server user and password, otherwise revert to default"
    echo "  -a                action to perform on bookmark. Type:String.
Values:share/unshare"
}

# Create Our Session, including establishing the API version.
function create_session
{
    # Pulling the version into parts. The {} are necessary for string manipulation.
    # Strip out longest match following "."  This leaves only the major version.
    major=${VERSION%%.*}
    # Strip out the shortest match preceding "." This leaves minor.micro.
    minorMicro=${VERSION#*.}
    # Strip out the shortest match followint "." This leaves the minor version.
    minor=${minorMicro%.*}
    # Strip out the longest match preceding "." This leaves the micro version.
    micro=${VERSION##*.}

    # Quick note about the <<-. If the redirection operator << is followed by a -
(dash), all leading TAB from the document data will be
```

```
    # ignored. This is useful to have optical nice code also when using here-
documents. Otherwise you must have the EOF be on a line by itself,
    # no parens, no tabs or anything.

    echo "creating session..."
    result=$(curl -s -S -X POST -k --data @- http://${engine}/resources/json/delphix/
session \
        -c ~/cookies.txt -H "Content-Type: application/json" <<-EOF
    {
        "type": "APISession",
        "version": {
            "type": "APIVersion",
            "major": $major,
            "minor": $minor,
            "micro": $micro
        }
    }
    EOF)

    check_result
}

# Authenticate the DE for the provided user.
function authenticate_de
{
    echo "authenticating delphix engine..."
    result=$(curl -s -S -X POST -k --data @- http://${engine}/resources/json/delphix/
login \
        -b ~/cookies.txt -c ~/cookies.txt -H "Content-Type: application/json" <<-EOF
    {
        "type": "LoginRequest",
        "username": "${username}",
        "password": "${password}"
    }
    EOF)

    check_result
}

function bookmark_action
{
    # Change share mode of bookmark

    if [[ $action = "share" ]]
    then
        result=$(curl -s -X POST -k --data @- http://${engine}/resources/json/
delphix/jetstream/bookmark/${bookmarkName}/$action \
       -b ~/cookies.txt -H "Content-Type: application/json" <<-EOF
       {}
       EOF)

    check_result
```

```
    echo "Bookmark ${bookmarkName} is now in shared mode"

    elif [[ $action = "unshare" ]]
    then
        result=$(curl -s -X POST -k --data @- http://${engine}/resources/json/
delphix/jetstream/bookmark/${bookmarkName}/$action \
       -b ~/cookies.txt -H "Content-Type: application/json" <<-EOF
       {}
       EOF)

    check_result

    echo "Bookmark ${bookmarkName} is now in not-share mode"

    fi

}

# Check the result of the curl. If there are problems, inform the user then exit.
function check_result
{
    exitStatus=$?
    if [ $exitStatus -ne 0 ]
    then
        echo "command failed with exit status $exitStatus"
        exit 1
    elif [[ $result != *"OKResult"* ]]
    then
        echo ""
        echo $result
        exit 1
    fi
}

##### Main

while getopts "u:d:a:h" flag; do
    case "$flag" in
        u )             username=${OPTARG%:*}
                        password=${OPTARG##*:}
                        ;;
        d )             engine=$OPTARG
                        ;;
        a )             action=$OPTARG
                        ;;
        h )             usage
                        exit
                        ;;
        * )             usage
                        exit 1
    esac

done
```

```
# Shift the parameters so we only have the positional arguments left
shift $((OPTIND-1))

# Check that there is 1 positional arguments
if [ $# != 1 ]
then
    usage
    exit 1
fi

# Get the one positional arguments
bookmarkName=$1

create_session
authenticate_de
bookmark_action
```

13.3.8.10.18 API cookbook: update a bookmark in Delphix self-service

This API cookbook recipe describes how to update a Bookmark (see page 2876) in Delphix Self-Service (Jet Stream). Note that the following example includes updating the "tags" on a Delphix Self-Service bookmark.

> The following script is for educational and demonstration purposes only and is not supported by Delphix.

```
#!/bin/bash
#
# sample script to update a bookmark on a Jet Stream container.
#
# Please set the following variables to suit your purposes.
# set this to the FQDN or IP address of the Delphix Engine
DE="ars-dlpx-6010-3.dlpxdc.co"
# set this to the Delphix admin user name
DELPHIX_ADMIN="admin"
# set this to the password for the Delphix admin user
DELPHIX_PASS="delphix"
# reference of bookmark you want to update
BOOKMARK_REF="JS_BOOKMARK-2"
#
# create our session
curl -s -X POST -k --data @- http://${DE}/resources/json/delphix/session \
    -c ~/cookies.txt -H "Content-Type: application/json" <<EOF
{
    "type": "APISession",
    "version": {
        "type": "APIVersion",
```

```
        "major": 1,
        "minor": 6,
        "micro": 2
    }
}
EOF
echo
#
# authenticate to the DE
curl -s -X POST -k --data @- http://${DE}/resources/json/delphix/login \
    -b ~/cookies.txt -c ~/cookies.txt -H "Content-Type: application/json" <<EOF
{
    "type": "LoginRequest",
    "username": "${DELPHIX_ADMIN}",
    "password": "${DELPHIX_PASS}"
}
EOF
echo
#
# Update the bookmark. Note that only fields you want to change must be included in
the bookmark
# json.
curl -s -X POST -k --data @- http://${DE}/resources/json/delphix/jetstream/bookmark/$
{BOOKMARK_REF} \
    -b ~/cookies.txt -H "Content-Type: application/json" <<EOF
{
    "type": "JSBookmark",
    "tags": ["tagA", "tabB"]
}
EOF
echo
```

13.3.8.10.19 API cookbook: delete Delphix self-service container

This API cookbook recipe describes how to delete a container in Delphix Self-Service.

> The following script is for educational and demonstration purposes only and is not supported by Delphix.

This script can be downloaded by selecting deleteContainer.sh[803]

Delete Delphix Self-Service Container

```
#!/bin/bash
#
# sample script to delete a bookmark on a Jet Stream container.
#
# Please set the following variables to suit your purposes.
# set this to the FQDN or IP address of the Delphix Engine
DE="ars-6010.dlpxdc.co"
# set this to the Delphix admin user name
DELPHIX_ADMIN="admin"
# set this to the password for the Delphix admin user
DELPHIX_PASS="delphix"
# reference of container you want to delete
CONTAINER_REF="JS_DATA_CONTAINER-1"
#

# create our session
curl -s -X POST -k --data @- http://${DE}/resources/json/delphix/session \
    -c ~/cookies.txt -H "Content-Type: application/json" <<EOF
{
    "type": "APISession",
    "version": {
        "type": "APIVersion",
        "major": 1,
        "minor": 6,
        "micro": 2
    }
}
EOF
echo

#
```

803 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149491509?preview=%2F149491509%2F149491520%2FdeleteContainer.sh

```
# authenticate to the DE
curl -s -X POST -k --data @- http://${DE}/resources/json/delphix/login \
    -b ~/cookies.txt -c ~/cookies.txt -H "Content-Type: application/json" <<EOF
{
    "type": "LoginRequest",
    "username": "${DELPHIX_ADMIN}",
    "password": "${DELPHIX_PASS}"
}
EOF
echo

#
# delete the bookmark
curl -s -X DELETE -k http://${DE}/resources/json/delphix/jetstream/container/$
{CONTAINER_REF} \
    -b ~/cookies.txt -H "Content-Type: application/json"
echo
```

13.3.8.10.20 API cookbook: delete Delphix self-service template

This API cookbook recipe describes how to delete a template in Delphix Self-Service

> The following script is for educational and demonstration purposes only and is not supported by Delphix

This script can be downloaded by selecting deleteTemplate.sh[804]

Delete Delphix Self-Service Template

```
#!/bin/bash
```

804 https://delphixdocs.atlassian.net/wiki/spaces/CD/pages/149491564?preview=%2F149491564%2F149491575%2FdeleteTemplate.sh

```
#
# sample script to delete a bookmark on a Jet Stream container.
#
# Please set the following variables to suit your purposes.
# set this to the FQDN or IP address of the Delphix Engine
DE="ars-dlpx-6010.dlpxdc.co"
# set this to the Delphix admin user name
DELPHIX_ADMIN="admin"
# set this to the password for the Delphix admin user
DELPHIX_PASS="delphix"
# reference of template you want to delete
TEMPLATE_REF="JS_DATA_TEMPLATE-1"
#

# create our session
curl -s -X POST -k --data @- http://${DE}/resources/json/delphix/session \
    -c ~/cookies.txt -H "Content-Type: application/json" <<EOF
{
    "type": "APISession",
    "version": {
        "type": "APIVersion",
        "major": 1,
        "minor": 6,
        "micro": 2
    }
}
EOF
echo

#
# authenticate to the DE
curl -s -X POST -k --data @- http://${DE}/resources/json/delphix/login \
    -b ~/cookies.txt -c ~/cookies.txt -H "Content-Type: application/json" <<EOF
{
    "type": "LoginRequest",
    "username": "${DELPHIX_ADMIN}",
    "password": "${DELPHIX_PASS}"
}
EOF
echo

#
# delete the bookmark
curl -s -X DELETE -k http://${DE}/resources/json/delphix/jetstream/template/$
{TEMPLATE_REF} \
    -b ~/cookies.txt -H "Content-Type: application/json"
echo
```

13.3.8.10.21 API cookbook: uploadUpgrade

This API cookbook recipe describes how to use uploadUpgrade.

```
curl -s -X POST -k --data @- http://delphix.engine/resources/json/delphix/session -c
~/cookies.txt -H "Content-Type: application/json" <<EOF
{
"type": "APISession",
"version": {
"type": "APIVersion",
"major": 1,
"minor": 10,
"micro": 0
}
}
EOF

curl -s -X POST -k --data @- httpk --data @- http://delphix.engine/resources/json/
delphix/login -b ~/cookies.txt -c ~/cookies.txt -H "Content-Type: application/json"
<<EOF
{
"type": "LoginRequest",
"username": "sysadmin",
"password": "sysadmin"
}
EOF


curl -s -X POST -F file=@upgrade_image_path http://delphix.engine/resources/json/
system/uploadUpgrade -b ~/cookies.txt
```

13.3.8.10.22 Kerberos APIs

13.3.8.10.22.1 API cookbook: ASEDBConfig

This API cookbook recipe describes how to configure your SAP ASE database using the Delphix Engine API.

```
{
    "name":"ASEDBConfig",
    "description":"A SAP ASE Database Config.",
    "abstract": true,
    "extends": {
        "$ref": "/delphix-source-config.json"
    },
    "properties": {
        "databaseName": {
            "type": "string",
            "description": "The name of the database.",
            "create": "required",
            "update": "optional",
            "pattern": "^[a-zA-Z0-9_]+$",
            "maxLength": 30
```

```
        },
        "user": {
            "type": "string",
            "description": "The username of the database user.",
            "update": "optional",
            "maxLength": 256
        },
        "credentials": {
            "type": "object",
            "description": "The password of the database user.",
            "$ref": "/delphix-credential.json",
            "update": "optional"
        },
        "repository": {
            "type": "string",
            "description": "The object reference of the source repository.",
            "format": "objectReference",
            "referenceTo": "/delphix-ase-instance.json",
            "create": "required",
            "update": "optional"
        }
    }
}
```

13.3.8.10.22.2 API cookbook: ASEhostEnvironmentParameters

This API cookbook recipe describes how to configure your SAP ASE host environment parameters using the Delphix Engine API.

```
{
    "name": "ASEHostEnvironmentParameters",
    "description": "SAP ASE host environment parameters.",
    "extends": {
        "$ref": "/delphix-typed-object.json"
    },
    "properties": {
        "dbUser": {
            "type": "string",
            "description": "The username of the database user.",
            "create": "optional",
            "update": "optional",
            "maxLength": 256
        },
        "credentials": {
            "type": "object",
            "description": "The credentials of the database user.",
            "$ref": "/delphix-credential.json",
            "create": "required",
            "update": "optional",
            "properties": {
                "type": {
```

```
                    "type": "string",
                    "description": "Object type.",
                    "required": true,
                    "format": "type",
                    "default": "PasswordCredential"
                }
            }
        }
    }
}
```

13.3.8.10.22.3 API cookbook: SAP ASE instance

This API cookbook recipe describes how to configure your SAP ASE instance using the Delphix Engine API.

```
{
    "name": "ASEInstance",
    "description": "The SAP ASE source repository.",
    "extends": {
        "$ref": "/delphix-source-repository.json"
    },
    "properties": {
        "instanceName": {
            "type": "string",
            "description": "The name of the SAP ASE instance.",
            "create": "required"
        },
        "installationPath": {
            "type": "string",
            "description": "The SAP ASE instance home.",
            "create": "required",
            "update": "optional"
        },
        "ports": {
            "type": "array",
            "description": "The network ports for connecting to the SAP ASE
instance.",
            "items": {
                type: "integer"
            },
            "create": "required",
            "update": "optional"
        },
        "instanceOwner": {
            "type": "string",
            "description": "The username of the account the SAP ASE instance is
running as.",
            "create": "required",
            "update": "optional"
        },
        "instanceOwnerUid": {
```

```
            "type": "integer",
            "description": "The uid of the account the SAP ASE instance is running
as.",
            "create": "readonly",
            "update": "readonly"
        },
        "instanceOwnerGid": {
            "type": "integer",
            "description": "The gid of the account the SAP ASE instance is running
as.",
            "create": "readonly",
            "update": "readonly"
        },
        "pageSize": {
            "type": "integer",
            "description": "Database page size for the SAP ASE instance."
        },
        "servicePrincipalName": {
            "type": "string",
            "description": "The Kerberos SPN of the database.",
            "create": "optional",
            "update": "optional"
        },
        "dbUser": {
            "type": "string",
            "description": "The username of the database user.",
            "create": "optional",
            "update": "optional",
            "maxLength": 256
        },
        "isqlPath" : {
            "type" : "string",
            "description" : "The path to the isql binary to use for this SAP ASE
instance.",
            "create" : "optional",
            "update" : "optional"
        },
        "credentials": {
            "type": "object",
            "description": "The credentials of the database user.",
            "$ref": "/delphix-credential.json",
            "create": "optional",
            "update": "optional",
            "properties": {
                "type": {
                    "type": "string",
                    "description": "Object type.",
                    "required": true,
                    "format": "type",
                    "default": "PasswordCredential"
                }
            }
        },
```

```
        "discovered": {
            "type": "boolean",
            "description": "True if the SAP ASE instance was automatically
discovered."
        }
    }
}
```

13.3.8.10.22.4 API cookbook: ASELinkData

This API cookbook recipe describes how to configure your SAP ASE link data using the Delphix Engine API.

```
{
    "name": "ASELinkData",
    "description": "SAP ASE specific parameters for a link request.",
    "extends": {
        "$ref": "/delphix-link-data.json"
    },
    "properties": {
        "config": {
            "type": "string",
            "description": "Reference to the configuration for the source.",
            "format": "objectReference",
            "referenceTo": "/delphix-ase-db-config.json",
            "required": true
        },
        "externalFilePath": {
            "type": "string",
            "description": "External file path.",
            "maxLength": 1024,
            "create": "optional",
        },
        "operations": {
            "description": "User-specified operation hooks for this source.",
            "type": "object",
            "$ref": "/delphix-linked-source-operations.json",
            "create": "optional"
        },
        "mountBase" : {
            "type" : "string",
            "description" : "The base mount point to use for the NFS mounts.",
            "maxLength" : 87,
            "create" : "optional"
        },
        "loadBackupPath": {
            "type": "string",
            "description": "Source database backup location.",
            "maxLength": 1024,
            "required": true
        },
        "loadLocation": {
```

```
            "type": ["object", "null"],
            "description": "Backup location to use for loading backups from the
source.",
            "$ref": "/delphix-ase-backup-location.json",
            "create": "optional"
        },
        "dumpCredentials": {
            "type": ["object", "null"],
            "description": "The credential for the source DB user.",
            "$ref": "/delphix-password-credential.json",
            "create": "optional"
        },
        "sourceHostUser": {
            "type": "string",
            "description": "Information about the host OS user on the source to use
for linking.",
            "format": "objectReference",
            "referenceTo": "/delphix-source-environment-user.json",
            "required": true
        },
        "dbUser": {
            "type": "string",
            "description": "The user name for the source DB user.",
            "create": "optional"
        },
        "dbCredentials": {
            "type": "object",
            "description": "The credentials of the database user.",
            "$ref": "/delphix-credential.json",
            "required": true,
            "properties": {
                "type": {
                    "type": "string",
                    "description": "Object type.",
                    "required": true,
                    "format": "type",
                    "default": "PasswordCredential"
                }
            }
        },
        "stagingRepository": {
            "type": "string",
            "description": "The SAP ASE instance on the staging environment that we
want to use for validated sync.",
            "format": "objectReference",
            "referenceTo": "/delphix-ase-instance.json",
            "required": true
        },
       "stagingHostUser": {
            "type": "string",
            "description": "Information about the host OS user on the staging
environment to use for linking.",
            "format": "objectReference",
```

```
            "referenceTo": "/delphix-source-environment-user.json",
            "required": true
        },
        "stagingPreScript": {
            "type": "string",
            "description": "A user-provided shell script or executable to run prior
to restoring from a backup during validated sync.",
            "maxLength": 1024,
            "create": "optional"
        },
        "stagingPostScript": {
             "type": "string",
             "description": "A user-provided shell script or executable to run after
restoring from a backup during validated sync.",
             "maxLength": 1024,
             "create": "optional"
        },
        "syncParameters": {
            "type": "object",
            "description": "Sync parameters for the container.",
            "$ref": "/delphix-ase-sync-parameters.json",
            "required": true,
            "properties": {
                "type": {
                    "type": "string",
                    "description": "Object type.",
                    "required": true,
                    "format": "type",
                    "default": "ASELatestBackupSyncParameters"
                }
            }
        },
        "validatedSyncMode": {
            "type": "string",
            "description": "Specifies the validated sync mode to synchronize the
dSource with the source database.",
            "enum": ["ENABLED", "DISABLED"],
            "default": "ENABLED",
            "create": "optional"
        }
    }
}
```

13.3.8.10.22.5 API cookbook: EnvironmentUser

This API cookbook recipe describes how to configure your environment user using the Delphix Engine API.

```
{
    "root": "/resources/json/delphix/environment/user",
    "name": "EnvironmentUser",
    "description": "The representation of an environment user object.",
```

```
    "extends": {
        "$ref": "/delphix-user-object.json"
    },
    "nameParent": "environment",
    "properties": {
        "credential": {
            "type": "object",
            "$ref": "/delphix-credential.json",
            "description": "The credential for the environment user.",
            "create": "required",
            "update": "optional",
            "properties": {
                "type": {
                    "type": "string",
                    "description": "Object type.",
                    "required": true,
                    "format": "type",
                    "default": "PasswordCredential"
                }
            }
        },
        "environment": {
            "type": "string",
            "description": "A reference to the associated environment.",
            "format": "objectReference",
            "referenceTo": "/delphix-source-environment.json",
            "create": "optional"
        },
        "groupId": {
            "type": "integer",
            "description": "Group ID of the user.",
            "create": "optional",
            "update": "optional",
            "minimum": 0,
            "maximum": 4294967295
        },
        "userId": {
            "type": "integer",
            "description": "User ID of the user.",
            "create": "optional",
            "update": "optional",
            "minimum": 0,
            "maximum": 4294967295
        }
    },
    "create": {
        "description": "Create a new EnvironmentUser object.",
        "payload" : {
            "type": "object",
            "$ref": "/delphix-source-environment-user.json"
         },
         "return": {
             "type": "string",
```

```
                "format": "objectReference",
                "referenceTo": "/delphix-source-environment-user.json"
            }
        },
        "read": {
            "description": "Retrieve the specified EnvironmentUser object.",
            "return": {
                "type": "object",
                "$ref": "/delphix-source-environment-user.json"
            }
        },
        "update": {
            "description": "Update the specified EnvironmentUser object.",
            "payload": {
                "type": "object",
                "$ref": "/delphix-source-environment-user.json"
            }
        },
        "delete": {
            "payload": {
                "type": "object",
                "$ref": "/delphix-delete-parameters.json",
                "required": false
            },
            "description" : "Delete the specified EnvironmentUser object."
        },
        "list": {
            "description": "Returns the list of all environment users in the system.",
            "parameters": {
                "environment": {
                    "type": "string",
                    "description": "Limit results to users within the given environment.",
                    "format": "objectReference",
                    "referenceTo": "/delphix-source-environment.json",
                    "mapsTo": "environment"
                }
            }
            "return": {
                "type": "array",
                "items": {
                  "type": "object",
                  "$ref": "/delphix-source-environment-user.json"
                }
            }
        }
    }
```

13.3.8.10.22.6 API cookbook: KerberosConfig

This API cookbook recipe describes how to configure Kerberos using the Delphix Engine API.

```
{
   name: "KerberosConfig",
   description: "Kerberos Client Configuration.",
   root: "/resources/json/delphix/service/kerberos",
   singleton: true,
   cliVisibility: [ "DOMAIN", "SYSTEM" ],
   extends: {
       $ref: "/delphix-user-object.json"
   },
   properties: {
       realm: {
           description: "Kerberos Realm name.",
           type: "string",
           create: "required",
           update: "optional"
       },
       kdcs: {
           description: "One of more KDC servers.",
           type: "array",
           create: "required",
           update: "optional",
           minItems: 1,
           items: {
               type: "object",
               $ref: "/delphix-kerberos-kdc.json"
           }
```

```
        },
        keytab: {
            description: "Kerberos keytab file data in base64 encoding.",
            type: "string",
            format: "password",
            create: "required",
            update: "optional"
        },
        principal: {
            description: "Kerberos principal name.",
            type: "string",
            create: "required",
            update: "optional"
        },
        enabled: {
            description: "Indicates whether kerberos has been configured or not.",
            type: "boolean"
        }
    },
    read: {
        description: "Retrieve the specified KerberosConfig object.",

        return: {

            type: "object",

            $ref: "/delphix-kerberos-config.json"

        }
```

```
    },

    update: {
        description: "Update the specified KerberosConfig object.",


        payload: {


            type: "object",


            $ref: "/delphix-kerberos-config.json"


       }


    },

    rootOperations: {

        reset: {

            description: "Reset kerberos configuration and disable the feature.",

            payload: {}

        }

    }

}
```

13.3.8.10.22.7 API cookbook: KerberosCredential

This API cookbook recipe describes how to configure Kerberos credentials using the Delphix Engine API.

```
{

    "name": "KerberosCredential",

    "description": "Kerberos based security credential.",

    "extends": {

        "$ref": "/delphix-credential.json"

    },

    "properties": {
```

```
    }

}
```

13.3.8.10.22.8 API cookbook: KerberosKDC

This API cookbook recipe describes how to configure KerberosKDC using the Delphix Engine API.

```
{
    name: "KerberosKDC",
    description: "Kerberos Client Configuration.",
    extends: {
        $ref: "/delphix-typed-object.json"
    },
    properties: {
        hostname: {
            description: "KDC Server hostname.",
            type: "string",
            format: "host",
            create: "required",
            update: "optional"
        },
        port: {
            description: "KDC Server port number.",
            type: "integer",
            create: "required",
            update: "optional",
            minimum: 0,
            maximum: 65535,
            default: 88
        }
    }
}
```