# LifeKeeper for Linux LIVE

9.7.0 — 最終更新日時: 29 January 2024

SIOS TECHNOLOGY CORP.

Copyright © 2024 SIOS TECHNOLOGY CORP.

# 目次

**1. LifeKeeper for Linux** ..... **28**

**2. LifeKeeper for Linux** リリースノート ..... **29**

**3. LifeKeeper for Linux** スタートアップガイド ..... **52**

**4. LifeKeeper for Linux** インストレーションガイド ..... **53**

4.1. LifeKeeper ソフトウェアのパッケージ ..... 54
4.2. LifeKeeper 環境のプランニング ..... 56
4.2.1. サーバー構成のマッピング ..... 57
4.2.2. ストレージとアダプターの要件 ..... 59
4.2.3. ストレージとアダプターのオプション ..... 60
4.3. LifeKeeper 環境のセットアップ ..... 61
4.3.1. Linux OS および関連する通信パッケージのインストール ..... 62
4.3.2. Linux の依存関係 ..... 63
4.3.3. サーバと共有ストレージの接続 ..... 67
4.3.4. 共有ストレージの設定 ..... 68
4.3.5. ネットワーク設定の確認 ..... 69
4.3.6. 切り替え可能な IP アドレスの作成 ..... 71
4.3.7. データベースアプリケーションのインストールとセットアップ ..... 72
4.3.8. GUI ユーザーの設定 ..... 73
4.3.9. LifeKeeper ライセンスについて ..... 75
4.3.9.1. ライセンスの取得方法 ..... 76
4.3.9.2. ライセンスのインストール方法 ..... 77
4.3.9.3. インターネットホスト ID の取得 ..... 78
4.4. LifeKeeper ソフトウェアのインストール ..... 79
4.5. セットアップスクリプトの操作 ..... 82
4.6. LifeKeeper インストールの確認 ..... 89
4.7. LifeKeeper のアップデート ..... 90
4.8. LifeKeeper を使用したノードの OS / カーネルのアップデート (OS パッチ適用) ..... 91

**5. LifeKeeper for Linux** テクニカルドキュメンテーション ..... **94**

5.1. はじめに ..... 95
5.2. ドキュメンテーションとトレーニング ..... 96
5.3. lkbackup ..... 98
5.4. LifeKeeper ..... 100
5.4.1. はじめに ..... 101
5.4.1.1. 保護対象のリソース ..... 103
5.4.1.2. LifeKeeper Core ..... 104
5.4.1.3. 設定の概念 ..... 107
5.4.1.3.1. 共通のハードウェアコンポーネント ..... 108
5.4.1.3.2. システムのグループ化の配置 ..... 110
5.4.1.3.3. アクティブ - アクティブのグループ化 ..... 111
5.4.1.3.4. アクティブ - スタンバイのグループ化 ..... 112

5.4.1.3.5. Intelligent スイッチバックと Automatic スイッチバックの違い ..... 114
5.4.1.3.6. syslog によるログの記録 ..... 115
5.4.1.3.7. リソース階層 ..... 116
5.4.1.3.7.1. リソースタイプ ..... 117
5.4.1.3.7.2. リソースの状態 ..... 118
5.4.1.3.7.3. 階層の関係 ..... 119
5.4.1.3.7.4. イクイバレンシ情報 ..... 120
5.4.1.3.7.5. リソース階層の情報 ..... 121
5.4.1.3.7.6. リソース階層の例 ..... 122
5.4.1.3.7.7. ステータスの詳細表示 ..... 123
5.4.1.3.7.8. ステータスの簡略表示 ..... 129
5.4.1.4. 障害検出とリカバリのシナリオ ..... 131
5.4.1.4.1. IP ローカルリカバリー ..... 132
5.4.1.4.2. リソースエラーのリカバリーシナリオ ..... 133
5.4.1.4.3. サーバー障害のリカバリーシナリオ ..... 136
5.4.2. インストールと設定 ..... 138
5.4.2.1. LifeKeeper の設定手順 ..... 139
5.4.2.1.1. TTY 接続のセットアップ ..... 141
5.4.2.2. SNMP による LifeKeeper イベント転送 ..... 142
5.4.2.2.1. SNMP による LifeKeeper イベント転送の概要 ..... 143
5.4.2.2.2. LifeKeeper イベント転送の設定 ..... 147
5.4.2.2.3. SNMP のトラブルシューティング ..... 149
5.4.2.3. LifeKeeper イベントメール通知 ..... 150
5.4.2.3.1. LifeKeeper イベントメール通知の概要 ..... 151
5.4.2.3.2. LifeKeeper イベントメール通知の設定 ..... 153
5.4.2.3.3. メール通知のトラブルシューティング ..... 155
5.4.2.4. 任意の設定作業 ..... 156
5.4.2.4.1. [Confirm Failover] と [Block Resource Failover] の設定 ..... 157
5.4.2.4.2. サーバーのシャットダウンストラテジーの設定 ..... 165
5.4.2.4.3. LifeKeeper ハートビートの調整 ..... 166
5.4.2.4.4. LifeKeeper API でカスタム証明書を使用する ..... 169
5.4.2.5. Linux の設定 ..... 171
5.4.2.6. データレプリケーションの設定 ..... 177
5.4.2.7. ネットワーク設定 ..... 178
5.4.2.8. アプリケーションの設定 ..... 179
5.4.2.9. ストレージとアダプターの設定 ..... 180
5.4.2.10. LifeKeeper I/O フェンシングの概要 ..... 203
5.4.2.10.1. SCSI リザベーション ..... 204
5.4.2.10.2. リザベーションの無効化 ..... 206
5.4.2.10.2.1. I/O フェンシング表 ..... 209
5.4.2.10.3. Quorum/Witness ..... 211
5.4.2.10.3.1. majority モード ..... 216
5.4.2.10.3.2. tcp_remote モード ..... 221
5.4.2.10.3.3. storage モード ..... 222
5.4.2.10.3.4. AWS 環境の Quorum/Witness クラスターの推奨事項 ..... 230

5.4.2.10.4. STONITH ... 236
5.4.2.10.5. Watchdog ... 241
5.4.2.10.6. I/O フェンシング機能組み合わせ情報 ... 244
5.4.2.10.6.1. フェンシング機能利用可否リスト（物理構成） ... 246
5.4.2.10.6.2. フェンシング機能利用可否リスト（仮想構成 VMware編） ... 250
5.4.2.11. リソースポリシー管理 ... 255
5.4.2.12. 認証情報の設定 ... 260
5.4.2.13. Standby Node Health Check ... 262
5.4.2.13.1. Node監視 ... 264
5.4.2.13.2. OSU リソース監視 ... 265
5.4.3. LifeKeeper 管理 ... 267
5.4.3.1. エラーの検出および通知 ... 269
5.4.3.2. N-Way リカバリ ... 270
5.4.3.3. 管理作業 ... 271
5.4.3.3.1. サーバプロパティの編集 ... 272
5.4.3.3.2. コミュニケーションパスの作成 ... 273
5.4.3.3.3. コミュニケーションパスの削除 ... 275
5.4.3.3.4. サーバのプロパティ - フェイルオーバ ... 276
5.4.3.3.5. リソース階層の作成 ... 278
5.4.3.3.5.1. ファイルシステムリソース階層の作成 ... 280
5.4.3.3.5.2. Generic Application リソース階層の作成 ... 282
5.4.3.3.5.3. Raw デバイスリソース階層の作成 ... 284
5.4.3.3.5.4. Quick Service Protection (QSP) Recovery Kit ... 286
5.4.3.3.6. リソースのプロパティの編集 ... 292
5.4.3.3.7. リソースの優先順位の編集 ... 293
5.4.3.3.8. リソース階層の拡張 ... 295
5.4.3.3.8.1. ファイルシステムリソース階層の拡張 ... 297
5.4.3.3.8.2. Generic Application リソース階層の拡張 ... 298
5.4.3.3.8.3. Raw デバイスリソース階層の拡張 ... 299
5.4.3.3.9. 階層の拡張解除 ... 300
5.4.3.3.10. リソース依存関係の作成 ... 301
5.4.3.3.11. リソース依存関係の削除 ... 303
5.4.3.3.12. すべてのサーバからの階層の削除 ... 304
5.4.4. LifeKeeper User Guide ... 305
5.4.4.1. LifeKeeper for Linux の使用 ... 306
5.4.4.1.1. GUI ... 307
5.4.4.1.1.1. GUI の概要 – 全般 ... 308
5.4.4.1.1.1.1. LifeKeeper GUI ソフトウェアパッケージ ... 309
5.4.4.1.1.2. メニュー ... 310
5.4.4.1.1.2.1. リソースのコンテキストメニュー ... 311
5.4.4.1.1.2.2. サーバのコンテキストメニュー ... 313
5.4.4.1.1.2.3. [File] メニュー ... 314
5.4.4.1.1.2.4. [Edit] メニュー - [Resource] ... 315
5.4.4.1.1.2.5. [Edit] メニュー - [Server] ... 316
5.4.4.1.1.2.6. [View] メニュー ... 317

5.4.4.1.1.2.7. [Help] メニュー ..... 319
5.4.4.1.1.3. ツールバー ..... 320
5.4.4.1.1.3.1. GUI のツールバー ..... 321
5.4.4.1.1.3.2. リソースのコンテキストツールバー ..... 323
5.4.4.1.1.3.3. サーバのコンテキストツールバー ..... 324
5.4.4.1.1.4. GUI の実行の準備 ..... 325
5.4.4.1.1.4.1. LifeKeeper の GUI の設定 ..... 326
5.4.4.1.1.4.2. GUI サーバの開始 / 停止 ..... 329
5.4.4.1.1.4.3. Java のセキュリティポリシー ..... 331
5.4.4.1.1.4.4. LifeKeeper サーバでの GUI の実行 ..... 334
5.4.4.1.1.4.5. LifeKeeper GUI – 概要 ..... 335
5.4.4.1.2. ステータスの表 ..... 336
5.4.4.1.3. プロパティパネル ..... 337
5.4.4.1.4. 出力パネル ..... 338
5.4.4.1.5. メッセージバー ..... 339
5.4.4.1.6. GUI の終了 ..... 340
5.4.4.1.7. 共通の作業 ..... 341
5.4.4.1.7.1. LifeKeeper の起動 ..... 342
5.4.4.1.7.2. LifeKeeper の停止 ..... 344
5.4.4.1.7.3. LifeKeeper プロセスの表示 ..... 347
5.4.4.1.7.4. LifeKeeper GUI サーバプロセスの表示 ..... 348
5.4.4.1.7.5. LifeKeeper の制御プロセスの表示 ..... 349
5.4.4.1.7.6. サーバのクラスタへの接続 ..... 350
5.4.4.1.7.7. クラスタからの切断 ..... 351
5.4.4.1.7.8. 接続サーバの表示 ..... 352
5.4.4.1.7.9. サーバのステータスの表示 ..... 353
5.4.4.1.7.10. サーバのプロパティの表示 ..... 354
5.4.4.1.7.11. サーバのログファイルの表示 ..... 355
5.4.4.1.7.12. リソースのタグと ID の表示 ..... 356
5.4.4.1.7.13. リソースのステータスの表示 ..... 357
5.4.4.1.7.14. リソースのプロパティの表示 ..... 359
5.4.4.1.7.15. Resource Labels ..... 360
5.4.4.1.7.16. メッセージ履歴の表示 ..... 361
5.4.4.1.7.17. リソース階層ツリーの展開と折り畳み ..... 362
5.4.4.1.7.18. [Cluster Connect] ダイアログ ..... 364
5.4.4.1.7.19. [Cluster Disconnect] ダイアログ ..... 365
5.4.4.1.7.20. [Resource Properties] ダイアログ ..... 366
5.4.4.1.7.21. [Server Properties] ダイアログ ..... 368
5.4.4.1.8. オペレータの作業 ..... 372
5.4.4.1.8.1. リソースを In Service にする ..... 373
5.4.4.1.8.2. リソースを Out of Service にする ..... 374
5.4.4.1.9. 高度な作業 ..... 375
5.4.4.1.9.1. LCD ..... 376
5.4.4.1.9.1.1. LCDI のコマンド ..... 377
5.4.4.1.9.1.2. LCD の設定データ ..... 381

5.4.4.1.9.1.3. LCD のディレクトリ構造 ..... 382
5.4.4.1.9.1.4. LCD のリソースタイプ ..... 383
5.4.4.1.9.1.5. LifeKeeper のフラグ ..... 384
5.4.4.1.9.1.6. リソースのサブディレクトリ ..... 385
5.4.4.1.9.1.7. /opt/LifeKeeper の LCD のディレクトリ構造 ..... 387
5.4.4.1.9.2. LCM ..... 388
5.4.4.1.9.2.1. 通信ステータスの情報 ..... 389
5.4.4.1.9.2.2. LifeKeeper の警報とリカバリ ..... 390
5.4.4.1.9.3. LifeKeeper API for Monitoring ..... 392
5.4.4.1.10. メンテナンス作業 ..... 401
5.4.4.1.10.1. LifeKeeper の設定値の変更 ..... 402
5.4.4.1.10.2. ファイルシステムの健全性の監視 ..... 405
5.4.4.1.10.3. LifeKeeper が保護するシステムのメンテナンス ..... 408
5.4.4.1.10.4. リソース階層のメンテナンス ..... 410
5.4.4.1.10.5. フェイルオーバ後の復旧 ..... 411
5.4.4.1.10.6. LifeKeeper の削除 ..... 412
5.4.4.1.10.7. ファイアーウォールを使用した状態での LifeKeeper の実行 ..... 413
5.4.4.1.10.8. ファイアーウォール経由での LifeKeeper GUI の実行 ..... 415
5.4.4.1.10.9. リソース階層の転送 ..... 417
5.4.4.1.11. テクニカルノート ..... 418
5.4.4.2. クラスターの例 ..... 423
5.4.4.3. ダイアログ ..... 424
5.4.5. トラブルシューティング ..... 434
5.4.5.1. ソリューション ..... 436
5.4.5.1.1. ビデオソリューション ..... 445
5.4.5.2. LifeKeeper が開始するフェイルオーバーの一般的な原因 ..... 446
5.4.5.3. 既知の問題と制限 ..... 451
5.4.5.3.1. インストール - 既知の問題と制限 ..... 452
5.4.5.3.2. LifeKeeper Core – 既知の問題と制限 ..... 455
5.4.5.3.3. インターネット / IP ライセンス - 既知の問題と制限 ..... 463
5.4.5.3.4. GUI – 既知の問題と制限 ..... 464
5.4.5.3.5. データレプリケーション – 既知の問題と制限 ..... 467
5.4.5.3.6. IPv6 - 既知の問題と制限 ..... 472
5.4.5.3.7. Apache - 既知の問題と制限 ..... 474
5.4.5.3.8. Oracle Recovery Kit – 既知の問題と制限 ..... 475
5.4.5.3.9. MySQL - 既知の問題と制限 ..... 476
5.4.5.3.10. NAS Recovery Kit – 既知の問題と制限 ..... 477
5.4.5.3.11. NFS Server Recovery Kit – 既知の問題と制限 ..... 478
5.4.5.3.12. SAP Recovery Kit – 既知の問題と制限 ..... 480
5.4.5.3.13. LVM Recovery Kit – 既知の問題と制限 ..... 482
5.4.5.3.14. Multipath Recovery Kits (DMMP / HDLM / PPATH /NECSPS) - 既知の問題と制限 ..... 483
5.4.5.3.15. DMMP Recovery Kit – 既知の問題と制限 ..... 484
5.4.5.3.16. DB2 Recovery Kit – 既知の問題と制限 ..... 485
5.4.5.3.17. Sybase ASE Recovery Kit – 既知の問題と制限 ..... 486

5.4.5.3.18. WebSphere MQ Recovery Kit – 既知の問題と制限 ..... 489
5.4.5.3.19. SAP HANA – 既知の問題と制限 ..... 490
5.4.5.3.20. Recovery Kit for EC2 – 既知の問題と制限 ..... 492
5.4.5.3.21. Oracle Cloud Infrastructure (OCI) 上で LifeKeeper を使用する際の既知の問題/制限事項 ..... 493
5.4.5.4. コミュニケーションパスの稼働と停止 ..... 494
5.4.5.5. 不完全なリソースの作成 ..... 495
5.4.5.6. 不完全なリソースの優先順位の変更 ..... 496
5.4.5.7. 階層の設定中に共有ストレージが見つからない ..... 498
5.4.5.8. LifeKeeper サーバ障害からの復旧 ..... 500
5.4.5.9. 停止できないプロセスからの復旧 ..... 501
5.4.5.10. 手動リカバリ時のパニックからの復旧 ..... 502
5.4.5.11. Out-of-Service 階層の復旧 ..... 503
5.4.5.12. リソースタグ名の制限 ..... 504
5.4.5.13. シリアル (TTY) コンソールの警告 ..... 505
5.4.5.14. システムが init 状態 S に遷移しているという警告 ..... 506
5.4.5.15. 共有ストレージでスレッドがハングしているというメッセージ ..... 507
5.5. SIOS DataKeeper for Linux ..... 508
5.5.1. SIOS DataKeeper for Linux によるミラーリング ..... 509
5.5.2. SIOS DataKeeper の仕組み ..... 511
5.5.3. インストールと設定 ..... 522
5.5.3.1. ハードウェアとソフトウェアの要件 ..... 523
5.5.3.2. 全般的な設定 ..... 525
5.5.3.3. DataKeeper for Linux のネットワーク設定 ..... 526
5.5.3.4. データレプリケーションパスの変更 ..... 527
5.5.3.5. ネットワーク帯域幅の要件の特定 ..... 528
5.5.3.5.1. Linux システム（物理または仮想）での変化率の測定 ..... 529
5.5.3.6. WAN の設定 ..... 539
5.5.3.7. SIOS DataKeeper for Linux のリソースタイプ ..... 540
5.5.3.8. DataKeeper 構成での I/O フェンシング ..... 542
5.5.3.9. リソースの設定作業 ..... 543
5.5.3.9.1. DataKeeper リソース階層の作成 ..... 544
5.5.3.9.1.1. Replicate New File System ..... 547
5.5.3.9.1.2. Replicate Existing File System ..... 550
5.5.3.9.1.3. DataKeeper Resource ..... 552
5.5.3.9.2. リソース階層の拡張 ..... 555
5.5.3.9.3. リソース階層の拡張解除 ..... 559
5.5.3.9.4. リソース階層の削除 ..... 560
5.5.3.9.5. DataKeeper リソースを Out of Service にする ..... 561
5.5.3.9.6. DataKeeper リソースを In Service にする ..... 562
5.5.3.9.7. リソース階層のテスト ..... 563
5.5.3.10. DataKeeper イベントテーブル ..... 564
5.5.4. 管理 ..... 565
5.5.4.1. ミラーのステータスの表示 ..... 566
5.5.4.2. GUI からのミラーの管理 ..... 568

5.5.4.2.1. 一時停止と再開 ... 570
5.5.4.2.2. 圧縮レベルの設定 ... 571
5.5.4.3. コマンドラインからのミラー管理 ... 572
5.5.4.4. コマンドラインからのミラーステータスの監視 ... 576
5.5.4.5. サーバの障害 ... 578
5.5.4.6. 再同期 ... 579
5.5.4.7. 全同期の回避 ... 580
5.5.4.8. 再同期前のデータ検証 (Wait to Resync) ... 585
5.5.5. DataKeeper で LVM を使用する ... 592
5.5.6. Fusion-io を使用するクラスター化 ... 593
5.5.7. DataKeeper for Linux が保護している領域に対する、スナップショット機能の利用についての注意事項 ... 596
5.5.8. DataKeeper for Linuxトラブルシューティング ... 597
5.6. コマンドラインインターフェース ... 604
5.6.1. コマンド ... 606
5.6.1.1. lklogmsg ... 610
5.6.1.2. SYS – LifeKeeper クラスタ内のシステムに関する LifeKeeper のコマンド ... 612
5.6.1.3. NET – コマンドに関するコミュニケーションパス ... 614
5.6.1.4. FLAG – LifeKeeper の内部フラグに関するコマンド ... 616
5.6.1.5. TYP – リソース階層タイプに関する LifeKeeper のコマンド ... 617
5.6.1.6. APP – リソースアプリケーションに関する LifeKeeper のコマンド (関連タイプのグループ) ... 618
5.6.1.7. DEP – リソースアプリケーションを相互に関連付ける方法に関する LifeKeeper のコマンド ... 619
5.6.1.8. INS – 個々の LifeKeeper 階層インスタンスに関するコマンド ... 626
5.6.1.8.1. 階層の拡張解除 ... 628
5.6.1.9. マニュアルページへのアクセス ... 629
5.6.2. LKCLI（LifeKeeper コマンドラインインターフェース） ... 631
5.6.2.1. ARK 毎のサブコマンド ... 646
5.6.3. Ansible による LifeKeeper のセットアップ ... 669
5.6.4. LKCLI ガイド ... 678
5.6.4.1. コミュニケーションパスの作成･削除 ... 679
5.6.4.2. リソースの作成 ... 685
5.6.4.2.1. ファイルシステムリソースの作成 ... 686
5.6.4.2.2. IP リソースの作成 ... 689
5.6.4.2.3. PostgreSQL リソースの作成 ... 692
5.6.4.2.4. DataKeeper リソースの作成 ... 695
5.6.4.3. ステータスを確認するには？ ... 699
5.6.4.4. スイッチオーバーの動作確認 ... 700
5.6.4.5. メンテナンス ... 702
5.6.4.6. クラスターの設定を引き継いで複製する ... 707

**6. アプリケーションリカバリーキット ... 711**
6.1. Apache Recovery Kit 管理ガイド ... 713
6.1.1. LifeKeeper のドキュメンテーションとApache リファレンス ... 714
6.1.2. Apache Recovery Kit の要件 ... 715

6.1.3. LifeKeeper における Apache Web Server の設定 ..... 716
6.1.3.1. 設定の定義と例 ..... 717
6.1.3.1.1. アクティブ / スタンバイ設定とアクティブ / アクティブ設定 ..... 722
6.1.3.2. Apache Web Server の設定に関する考慮事項 ..... 723
6.1.4. Apache のLifeKeeper の設定作業 ..... 727
6.1.4.1. Apache Web Server リソース階層の作成 ..... 728
6.1.4.2. Apache Web Server リソース階層の拡張 ..... 730
6.1.4.3. Apache Web Server リソース階層の拡張解除 ..... 733
6.1.4.4. Apache Web Server リソース階層の削除 ..... 734
6.1.4.5. Apache Web Server リソース階層のテスト ..... 735
6.1.5. Apache Web Server トラブルシューティング ..... 736
6.1.5.1. Apache 階層作成時のエラー ..... 737
6.1.5.2. Apache 階層拡張時のエラー ..... 741
6.1.5.3. Apache 階層リソースの リストア、リムーブ、リカバリー時のメッセージとエラー 743
6.1.6. Apache Recovery Kit 処理概要 ..... 746
6.2. DB2 Recovery Kit 管理ガイド ..... 748
6.2.1. ドキュメンテーションとリファレンス ..... 749
6.2.2. ハードウェア要件とソフトウェア要件 ..... 750
6.2.3. DB2 Recovery Kit の概要 ..... 751
6.2.4. LifeKeeper for Linux DB2 Recovery Kit の設定 ..... 752
6.2.4.1. DB2 で RAW I/O を使用する場合 ..... 753
6.2.4.2. DB2 の使用 ..... 754
6.2.4.3. DB2 シングルパーティションの設定の検討事項 ..... 755
6.2.4.4. DB2 マルチ・パーティションの設定の検討事項 ..... 756
6.2.4.4.1. DB2 EEE またはマルチ・パーティション ESE と NFS に関する問題 ..... 757
6.2.4.4.2. 設定条件 ..... 759
6.2.4.5. すべての DB2 構成に共通する設定の検討事項 ..... 762
6.2.4.6. 構成例 ..... 763
6.2.5. LifeKeeper for Linux DB2 Recovery Kitの設定タスク ..... 768
6.2.5.1. DB2 リソース階層の作成 ..... 769
6.2.5.2. DB2 リソース階層の削除 ..... 772
6.2.5.3. DB2 リソース階層の拡張 ..... 774
6.2.5.4. DB2 階層の拡張解除 ..... 777
6.2.5.5. DB2 リソース階層のテスト ..... 778
6.2.6. DB2 トラブルシューティング ..... 779
6.2.7. 付録： DB2 で RAW I/O を使用する設定 ..... 781
6.3. Recovery Kit for EC2™ 管理ガイド ..... 784
6.3.1. Recovery Kit for EC2™ 運用の原則 ..... 785
6.3.2. Recovery Kit for EC2™ 処理概要 ..... 789
6.3.3. Recovery Kit for EC2™ の要件 ..... 791
6.3.4. Recovery Kit for EC2™ の設定 ..... 793
6.3.4.1. EC2™ イベントテーブル ..... 795
6.3.4.2. Recovery Kit for EC2™ のチューニング ..... 796
6.3.4.3. EC2™ リソース階層の作成 ..... 797
6.3.4.4. EC2™ リソース階層の削除 ..... 799

6.3.4.5. EC2™ リソース階層の拡張 ..... 800
6.3.4.6. EC2™ ローカルリカバリと設定に関する考慮事項 ..... 802
6.3.4.7. EC2™ リソース監視と設定に関する考慮事項 ..... 804
6.3.4.8. EC2™ リソース階層の拡張解除 ..... 805
6.3.4.9. EC2™ ユーザーシステムのセットアップ ..... 806
6.3.5. Recovery Kit for EC2™ トラブルシューティング ..... 808
6.4. LB Health Check Kit 管理ガイド ..... 809
6.4.1. 環境構成例 ..... 810
6.4.2. Generic ARK for Load Balancer probe reply から LB Health Check Kit へのアップグレード ..... 813
6.4.3. リソースの作成/拡張/変更 ..... 815
6.4.4. 処理概要 ..... 819
6.5. Logical Volume Manager Recovery Kit 管理ガイド ..... 821
6.5.1. ドキュメンテーションとリファレンス ..... 822
6.5.2. Logical Volume Manager Recovery Kit の要件 ..... 823
6.5.2.1. Logical Volume Manager ハードウェア要件とソフトウェア要件 ..... 824
6.5.3. 概要 ..... 825
6.5.3.1. Logical Volume Manager Recovery Kit 注意事項と制約事項 ..... 827
6.5.4. LifeKeeper Logical Volume Manager リソース階層の作成と管理 ..... 829
6.5.4.1. Logical Volume Manager リソース階層作成の手順 ..... 830
6.5.4.2. DataKeeper で LVM を使用する ..... 831
6.5.4.3. ボリュームグループの再設定 ..... 832
6.5.5. LVM トラブルシューティング ..... 836
6.5.6. LVM Recovery Kit 処理概要 ..... 838
6.6. IP Recovery Kit 管理ガイド ..... 841
6.6.1. IP Recovery Kit 運用の原則 ..... 842
6.6.2. IP Recovery Kit の要件 ..... 846
6.6.3. IP Recovery Kit の設定 ..... 847
6.6.3.1. IP インターフェースの選択 ..... 849
6.6.3.2. IP ユーザーシステムのセットアップ ..... 850
6.6.3.3. 一般的な IP プランニングの考慮事項 ..... 851
6.6.3.4. IP のリソース監視と設定に関する考慮事項 ..... 852
6.6.3.5. IP ローカルリカバリーと設定に関する考慮事項 ..... 853
6.6.3.6. 設定例 ..... 854
6.6.3.7. IP リソース階層の作成 ..... 861
6.6.3.8. リソース階層の削除 ..... 863
6.6.3.9. リソース階層の拡張 ..... 864
6.6.3.10. リソース階層の拡張解除 ..... 867
6.6.3.11. リソース階層のテスト ..... 868
6.6.3.12. IP 構成の確認および編集 ..... 869
6.6.3.13. IP Recovery Kit のチューニング ..... 881
6.6.4. IP Recovery Kit (IPv4) 処理概要 ..... 882
6.7. MySQL Recovery Kit 管理ガイド ..... 889
6.7.1. 要件 ..... 890
6.7.2. MySQL Recovery Kit の構成 ..... 891

6.7.2.1. MySQL の設定上の考慮事項 ... 892
6.7.2.2. MySQL クライアント設定の考慮事項 ... 895
6.7.2.3. MySQL 設定の要件 ... 896
6.7.2.4. MySQL の設定例 ... 897
6.7.2.5. アクティブ / スタンバイ設定 ... 898
6.7.2.6. アクティブ / アクティブ設定 ... 900
6.7.2.7. 複数データベースサーバ環境に関する考慮事項 ... 905
6.7.2.8. LifeKeeper で mysqld グループを使用する ... 906
6.7.2.9. Network Attached Storage の使用 ... 911
6.7.2.10. systemd 環境で使用する場合の考慮事項 ... 915
6.7.3. MySQL インストール ... 916
6.7.3.1. MySQL Recovery Kit の設定タスク ... 917
6.7.3.2. MySQLリソース階層の作成 ... 918
6.7.3.3. MySQLリソース階層の削除 ... 922
6.7.3.4. MySQLリソース階層の拡張 ... 924
6.7.3.5. MySQLリソース階層の拡張解除 ... 928
6.7.4. 管理 ... 930
6.7.4.1. GUI による手動スイッチオーバの実行 ... 931
6.7.5. MySQL トラブルシューティング ... 932
6.7.6. MySQL Recovery Kit 処理概要 ... 935
6.8. WebSphere MQ Recovery Kit 管理ガイド ... 937
6.8.1. 略語 ... 939
6.8.2. MQ Recovery Kit の要件 ... 940
6.8.2.1. MQ ハードウェア要件とソフトウェア要件 ... 941
6.8.2.2. LifeKeeper MQ クラスタのアップグレード ... 943
6.8.3. WebSphere MQ Recovery Kit の概要 ... 944
6.8.3.1. WebSphere MQ リソース階層 ... 945
6.8.3.2. MQ Recovery Kit の機能 ... 946
6.8.4. WebSphere MQ を設定する際の考慮事項 ... 947
6.8.4.1. MQ 設定の要件 ... 948
6.8.4.1.1. MQ でサポートされるファイルシステムレイアウト ... 951
6.8.4.1.1.1. 設定 1 – 共有ストレージ上の /var/mqm ... 952
6.8.4.1.1.2. 設定 2 – 直接マウント ... 953
6.8.4.1.1.3. 設定 3 – シンボリックリンク ... 954
6.8.4.1.2. LifeKeeper 使用時の WebSphere MQ の設定 ... 955
6.8.4.1.3. MQ リソース作成後の設定変更 ... 960
6.8.4.1.3.1. QMDIR と QMLOGDIR の再配置 ... 961
6.8.4.1.3.2. リスナーポートの変更 ... 962
6.8.4.1.3.3. キューマネージャーの IP の変更 ... 963
6.8.4.1.4. WebSphere MQ の設定例 ... 964
6.8.4.1.4.1. 共有ストレージ上の /var/mqm によるアクティブ / スタンバイ設定 ... 965
6.8.4.1.4.2. NAS ストレージにおけるアクティブ / スタンバイ設定 ... 966
6.8.4.1.4.3. 共有ストレージにおけるアクティブ / アクティブ設定 ... 968
6.8.4.1.4.4. NAS ストレージにおけるアクティブ / アクティブ設定 ... 970
6.8.5. MQ用 LifeKeeper の設定作業 ... 972

6.8.5.1. WebSphere MQ リソース階層の作成 ... 974
6.8.5.2. WebSphere MQ 階層の拡張 ... 976
6.8.5.3. WebSphere MQ 階層の拡張解除 ... 978
6.8.5.4. WebSphere MQ リソース階層の削除 ... 979
6.8.5.5. WebSphere MQ リソース階層のテスト ... 980
6.8.5.5.1. MQ クライアント接続のテスト ... 982
6.8.5.6. MQ リソースのプロパティの表示 ... 984
6.8.5.7. MQ リソースプロパティ設定の編集 ... 985
6.8.5.7.1. リスナー保護の有効化 / 無効化 ... 989
6.8.5.7.2. LifeKeeper テストキュー名の変更 ... 991
6.8.5.7.3. ログレベルの変更 ... 992
6.8.5.7.4. シャットダウンタイムアウト値の変更 ... 993
6.8.5.7.5. サーバ接続チャネルの変更 ... 995
6.8.5.7.6. コマンドサーバ保護設定の変更 ... 996
6.8.5.7.7. LifeKeeper WebSphere MQ Recovery Kit のデフォルト設定の変更 ... 998
6.8.6. WebSphere MQ のトラブルシューティング ... 1000
6.8.6.1. MQ エラーメッセージ ... 1001
6.8.7. 付録 A – mqs.ini 設定ファイルの例 ... 1009
6.8.8. 付録 B – qm.ini 設定ファイルの例 ... 1011
6.8.9. 付録 C – WebSphere MQ 設定シート ... 1012
6.9. NAS Recovery Kit 管理ガイド ... 1015
6.9.1. NAS ドキュメンテーションとリファレンス ... 1016
6.9.2. NAS Recovery Kit ハードウェア要件とソフトウェア要件 ... 1017
6.9.3. NAS Recovery Kit 概要 ... 1018
6.9.4. LifeKeeper for Linux NAS Recovery Kit の設定 ... 1020
6.9.4.1. NAS 構成に関する考慮事項 ... 1021
6.9.4.2. NAS の構成例 ... 1023
6.9.5. NAS の LifeKeeper の設定作業 ... 1025
6.9.5.1. NAS リソース階層の作成 ... 1026
6.9.5.2. NAS リソース階層の削除 ... 1029
6.9.5.3. NAS 階層の拡張 ... 1031
6.9.5.4. NAS 階層の拡張解除 ... 1034
6.9.5.5. NAS リソース階層のテスト ... 1035
6.9.6. NAS トラブルシューティング ... 1036
6.9.6.1. NAS エラーメッセージ ... 1037
6.9.6.2. LifeKeeper GUI 関連のエラー ... 1042
6.10. NFS Recovery Kit 管理ガイド ... 1043
6.10.1. NFS Server Recovery Kit 概要 ... 1044
6.10.2. NFS Server Recovery Kit 要件 ... 1045
6.10.3. NFS Server Recovery Kit 設定上の考慮事項 ... 1046
6.10.3.1. LifeKeeper における NFS サーバーの設定 ... 1047
6.10.3.2. NFS 固有の構成に関する考慮事項 ... 1051
6.10.3.3. NFS 構成例 ... 1052
6.10.3.3.1. アクティブ / スタンバイ設定 ... 1053
6.10.3.3.2. アクティブ / アクティブ設定 ... 1055

6.10.4. NFS 設定作業 ... 1057
6.10.4.1. NFS リソース階層の作成 ... 1058
6.10.4.2. NFS リソース階層の削除 ... 1061
6.10.4.3. NFS リソース階層の拡張 ... 1063
6.10.4.4. NFS リソース階層の拡張解除 ... 1067
6.10.4.5. NFS リソース階層のテスト ... 1069
6.10.5. NFS トラブルシューティング ... 1071
6.10.5.1. HA nfs-utils のインストールと設定 ... 1072
6.10.5.2. NFS リソース階層作成時のエラー ... 1073
6.10.5.3. NFS リソース階層拡張時のエラー ... 1077
6.10.5.4. NFS リソース階層の起動、停止、quickCheck、ローカルリカバリー時のメッセージとエラー ... 1078
6.10.5.5. NFS リソース階層削除のメッセージとエラー ... 1083
6.11. Recovery Kit for Oracle® Cloud Infrastructure 管理ガイド ... 1084
6.11.1. 運用の原則 ... 1085
6.11.2. リソースの監視とローカルリカバリー動作 ... 1086
6.11.3. 要件 ... 1088
6.11.4. Recovery Kit for Oracle Cloud Infrastructure を利用する際の注意事項 ... 1090
6.11.5. 設定 ... 1091
6.11.5.1. Recovery Kit for Oracle Cloud Infrastructure リソース階層の作成 ... 1092
6.11.5.2. Recovery Kit for Oracle Cloud Infrastructure リソース階層の削除 ... 1094
6.11.5.3. Recovery Kit for Oracle Cloud Infrastructure リソース階層の拡張 ... 1095
6.11.5.4. Recovery Kit for Oracle Cloud Infrastructure リソース階層の拡張解除 ... 1097
6.11.5.5. Recovery Kit for Oracle Cloud Infrastructure のチューニング ... 1098
6.11.6. トラブルシューティング ... 1099
6.11.6.1. 既知の問題と制限事項 ... 1100
6.11.6.2. エラーメッセージ ... 1102
6.12. Oracle® Recovery Kit 管理ガイド ... 1103
6.12.1. Oracle Recovery Kit のハードウェアとソフトウェアの要件 ... 1104
6.12.2. LifeKeeper での Oracle の設定 ... 1106
6.12.2.1. Oracle 特有の設定上の考慮事項 ... 1107
6.12.2.2. LifeKeeper 保護用の Oracle Net Listener の設定 ... 1112
6.12.2.3. LifeKeeper での透過的アプリケーションフェイルオーバの設定 ... 1116
6.12.2.4. Oracle Multitenant 構成による Pluggable Database の設定 ... 1118
6.12.2.5. Oracle の構成例 ... 1123
6.12.2.5.1. Oracle 設定の要件 ... 1124
6.12.2.5.2. アクティブ / スタンバイ設定 ... 1125
6.12.2.5.3. アクティブ / アクティブ設定 ... 1128
6.12.3. Oracle Recovery Kitの設定タスク ... 1133
6.12.3.1. リソース階層の作成 ... 1135
6.12.3.2. リソース階層の削除 ... 1137
6.12.3.3. リソース階層の拡張 ... 1139
6.12.3.4. リソース階層の拡張解除 ... 1143
6.12.3.5. Oracle 設定の参照 ... 1145
6.12.3.6. Oracle データベースアカウントに対するユーザ名 / パスワードの変更 ... 1146

6.12.3.7. リソース階層のテスト ..... 1148
6.12.3.8. DataKeeper を使用した Oracle ノード（SAP/Oracle）へのパッチ適用 ..... 1149
6.12.4. Oracle トラブルシューティング ..... 1150
6.12.4.1. Oracle 既知の問題と制限 ..... 1151
6.12.4.1.1. データベース作成時の問題 ..... 1156
6.12.4.1.2. データベース起動時の問題 ..... 1157
6.12.4.1.3. LifeKeeper の inqfail エラーログ ..... 1158
6.12.5. 付録 ..... 1159
6.12.5.1. Raw I/O を使用するための Oracle のセットアップ ..... 1160
6.12.5.1.1. 階層作成後のテーブルスペースの追加 ..... 1163
6.12.5.2. 複数のリソースに対する共有 Oracle Listener の作成 ..... 1164
6.12.5.2.1. Listener 保護レベルのアップデート ..... 1168
6.12.5.2.2. Listener リカバリレベルのアップデート ..... 1169
6.12.5.2.3. 保護された Listener のアップデート ..... 1170
6.12.5.3. Pluggable Database の移行 ..... 1171
6.13. PostgreSQL Recovery Kit 管理ガイド ..... 1174
6.13.1. PostgreSQL リソース階層 ..... 1175
6.13.2. PostgreSQLのハードウェアおよびソフトウェア要件 ..... 1176
6.13.3. PostgreSQL 設定上の考慮事項 ..... 1177
6.13.3.1. Postgre SQL の保護: ベストプラクティス ..... 1178
6.13.3.2. DataKeeperによるミラーファイルシステムの使用 ..... 1179
6.13.4. PostgreSQL インストール ..... 1180
6.13.4.1. PostgreSQL ソフトウェアのインストール ..... 1182
6.13.4.2. PostgreSQLデータベースの作成 ..... 1184
6.13.4.3. LifeKeeper ソフトウェアのインストール ..... 1186
6.13.4.4. PostgreSQL リソースのための LifeKeeper 設定項目 ..... 1187
6.13.4.5. PostgreSQL リソース階層の作成 ..... 1189
6.13.4.6. PostgreSQL リソース階層の削除 ..... 1191
6.13.4.7. PostgreSQL リソース階層の拡張 ..... 1192
6.13.4.8. PostgreSQL リソース階層の拡張解除 ..... 1194
6.13.4.9. PostgreSQL 設定の参照 ..... 1195
6.13.4.10. PostgreSQL Recovery Kit の旧バージョンからのアップグレード ..... 1196
6.13.5. 管理 ..... 1199
6.13.5.1. LifeKeeper GUI からの手動スイッチオーバの実行 ..... 1200
6.13.5.2. EnterpriseDB Postgres Plus Advanced Server の保護 ..... 1201
6.13.5.3. Symfoware Server/Enterprise Postgres の保護 ..... 1202
6.13.5.4. データベース管理者ユーザの更新 ..... 1203
6.13.6. PostgreSQL トラブルシューティング ..... 1204
6.13.6.1. PostgreSQL 全般項目 ..... 1205
6.13.6.2. PostgreSQL 設定項目 ..... 1207
6.14. Postfix Recovery Kit 管理ガイド ..... 1209
6.14.1. Postfix ハードウェア要件とソフトウェア要件 ..... 1211
6.14.1.1. Postfix Recovery Kit のインストール ..... 1212
6.14.2. LifeKeeper for Linux Postfix Recovery Kit の設定 ..... 1213
6.14.2.1. Postfix の保護対象 ..... 1214

6.14.2.2. Postfix の設定要件 ..... 1215
6.14.2.3. ポート定義とPostfix Recovery Kit ..... 1216
6.14.2.4. DNS、Postfix とLifeKeeper ..... 1217
6.14.2.5. 構成例 ..... 1218
6.14.3. Postfix の構成確認 ..... 1222
6.14.3.1. Postfix のLifeKeeper 設定作業 ..... 1224
6.14.3.1.1. Postfix リソース階層の作成 ..... 1225
6.14.3.1.2. Postfix リソース階層の拡張 ..... 1227
6.14.3.1.3. Postfix リソース階層の拡張解除 ..... 1229
6.14.3.1.4. Postfix リソース階層の削除 ..... 1230
6.14.3.1.5. スプール (メールボックス) の依存関係作成 ..... 1231
6.14.3.1.6. Postfix リソース階層のテスト ..... 1232
6.14.4. Postfix トラブルシューティング ..... 1234
6.14.4.1. Postfix リソース階層作成時のエラーメッセージ ..... 1235
6.14.4.2. Postfix リソース階層拡張時のエラーメッセージ ..... 1236
6.14.4.3. Postfix リソース In-Service / Out-of-Service / Health Monitoring エラーメッセージ ..... 1237
6.15. Recovery Kit for Route 53 管理ガイド ..... 1238
6.15.1. Recovery Kit for Route 53™ の要件 ..... 1239
6.15.2. 構成例 ..... 1240
6.15.3. Recovery Kit for Route 53™ の設定 ..... 1242
6.15.3.1. Route53™ リソース階層の作成 ..... 1244
6.15.3.2. Route53™ リソース階層の削除 ..... 1246
6.15.3.3. Route53™ リソース階層の拡張 ..... 1247
6.15.3.4. Route53™ リソース階層の拡張解除 ..... 1249
6.15.3.5. Recovery Kit for Route 53 のチューニング ..... 1250
6.15.3.6. Route53™ リソースの監視とリカバリ動作 ..... 1251
6.15.3.7. Route53™ ユーザーシステムのセットアップ ..... 1252
6.15.4. Recovery Kit for Route 53™ のトラブルシューティング ..... 1253
6.15.5. Recovery Kit for Route53™ 処理概要 ..... 1254
6.16. Samba Recovery Kit 管理ガイド ..... 1257
6.16.1. Samba Recovery Kit の要件 ..... 1259
6.16.2. Samba Recovery Kit のインストール ..... 1260
6.16.3. Samba Recovery Kit の概要 ..... 1261
6.16.4. LifeKeeper for Linux Samba Recovery Kit の設定 ..... 1262
6.16.4.1. Samba 設定ファイル ..... 1263
6.16.4.2. 設定ファイルの [global] セクション ..... 1264
6.16.4.3. 設定ファイルの [homes] セクション ..... 1266
6.16.4.4. 設定ファイルの [printers] セクション ..... 1267
6.16.4.5. 設定ファイルの共用定義セクション ..... 1268
6.16.4.6. Samba の複数インスタンスの実行 ..... 1269
6.16.4.7. 構成例 ..... 1271
6.16.5. Samba の設定手順 ..... 1276
6.16.6. Samba の LifeKeeper の設定作業 ..... 1278
6.16.6.1. Samba のリソース階層の作成 ..... 1279

6.16.6.2. Samba のリソース階層の拡張 ..... 1281
6.16.6.3. Samba のリソース階層の拡張解除 ..... 1283
6.16.6.4. Samba のリソース階層の削除 ..... 1284
6.16.6.5. Samba のリソース階層のテスト ..... 1285
6.16.7. Samba 階層の管理 ..... 1286
6.16.7.1. Samba 設定ファイルの修正 ..... 1287
6.16.7.2. smbpasswd ファイルの管理 ..... 1290
6.16.8. Samba トラブルシューティング ..... 1291
6.16.8.1. Samba のエラーメッセージ ..... 1292
6.16.8.2. リソース階層作成時のエラーメッセージ ..... 1293
6.16.8.3. リソース拡張時のエラーメッセージ ..... 1297
6.16.8.4. リソース復元時のエラーメッセージ ..... 1298
6.16.8.5. リソース削除時のエラーメッセージ ..... 1299
6.16.8.6. リソース監視時のエラーメッセージ ..... 1300
6.16.8.7. 設定ファイル同期ユーティリティのエラーメッセージ ..... 1301
6.17. SAP Recovery Kit 管理ガイド ..... 1302
6.17.1. SAP 略語と定義 ..... 1303
6.17.2. LifeKeeper – SAP アイコン ..... 1305
6.17.3. SAP Recovery Kit の概要 ..... 1306
6.17.4. LifeKeeper SAP ソリューションページ ..... 1309
6.17.5. SAP ハードウェア / ソフトウェア要件 ..... 1311
6.17.6. SAP 設定上の考慮事項 ..... 1313
6.17.6.1. ABAP+Java の設定 (ASCS および SCS) ..... 1315
6.17.6.2. ABAP SCS (ASCS) ..... 1318
6.17.6.3. Java のみの構成 (SCS) ..... 1320
6.17.6.4. SAP ディレクトリ構造 ..... 1322
6.17.6.5. SAP 仮想サーバ名 ..... 1329
6.17.6.6. SAP ヘルスモニタリング ..... 1330
6.17.6.7. SAP ライセンス ..... 1332
6.17.6.8. その他の注記 ..... 1333
6.17.7. SAP のインストール ..... 1334
6.17.7.1. SAP 構成を計画する ..... 1337
6.17.7.2. ファイルシステムとディレクトリ構造の作成 ..... 1339
6.17.7.3. コアサービスのインストール ..... 1341
6.17.7.4. データベースのインストール ..... 1342
6.17.7.5. プライマリアプリケーションサーバインスタンスのインストール ..... 1343
6.17.7.6. 追加のアプリケーションサーバインスタンスのインストール ..... 1344
6.17.7.7. バックアップサーバへのインストール ..... 1345
6.17.7.8. LifeKeeper のインストール ..... 1346
6.17.7.9. ASCS および ERS インスタンスプロファイル設定の変更 ..... 1348
6.17.7.10. SAP IP リソース ..... 1351
6.17.7.11. SAP リソース階層の作成 ..... 1352
6.17.7.12. SAP リソース階層の削除 ..... 1363
6.17.7.13. SAP Recovery Kit の一般的な作業 ..... 1364
6.17.7.14. コマンドラインからの SAP のセットアップ (LKCLI) ..... 1365

6.17.7.15. SAP SIOS HA Cluster Connector（SHCC）のアクティブ化 ..... 1368
6.17.7.16. SAP テストの準備 ..... 1370
6.17.7.17. SAP テストの実行 ..... 1371
6.17.8. SAP の管理 ..... 1373
6.17.8.1. NFS の考慮事項 ..... 1374
6.17.8.2. SAP クライアントの再接続 ..... 1375
6.17.8.3. SAP Recovery Kit のチューニング ..... 1376
6.17.8.4. SAP 保護レベルの更新 ..... 1378
6.17.8.5. SAP リカバリレベルの更新 ..... 1380
6.17.8.6. SAP Protection / Recovery Level レベルの推奨事項 ..... 1382
6.17.8.7. SAP プロパティの表示 ..... 1384
6.17.8.8. Oracle に関する特別な考慮事項 ..... 1385
6.17.8.9. SSHCC HA アクション ..... 1386
6.17.8.10. LifeKeeper の ERS リソースタイプ ..... 1387
6.17.8.11. ENSAv1 から ENSAv2 へのアップグレード ..... 1393
6.17.8.12. ERSv1 から ERSv2 へのアップグレード ..... 1395
6.17.9. SAP トラブルシューティング ..... 1399
6.17.9.1. ERSプロファイルでAutostartを無効にする ..... 1400
6.17.9.2. ASCS + ERS Restart_Program パラメーター ..... 1401
6.17.9.3. SAP 階層削除時のエラー ..... 1402
6.17.9.4. SAP 階層のリストアエラー ..... 1403
6.17.9.5. フェイルオーバ中または In Service の SAP エラーメッセージ ..... 1405
6.17.9.6. SAP インストールエラー ..... 1406
6.17.9.7. sapinit のトラブルシューティング ..... 1407
6.17.10. SAP メンテナンスモード ..... 1408
6.17.11. リソースポリシー管理 ..... 1412
6.17.12. LifeKeeper ログファイルに出現する「tset」エラー ..... 1417
6.18. SAP HANA Recovery Kit 管理ガイド ..... 1419
6.18.1. v9.7.0 未満からのアップグレード ..... 1421
6.18.2. SAP HANA サポート構成 ..... 1423
6.18.3. SAP HANA Recovery Kit ハードウェアおよびソフトウェアの要件 ..... 1425
6.18.4. SAP HANA Recovery Kit 概要 ..... 1427
6.18.4.1. SAP HANA GUI ステータス ..... 1429
6.18.4.2. SAP HANAリソース階層 ..... 1432
6.18.4.3. マルチテナントデータベースのサポート ..... 1433
6.18.5. LifeKeeper での SAP HANA の設定 ..... 1434
6.18.5.1. SAP HANA ソフトウェアのインストール ..... 1435
6.18.5.2. SAP HANA システムレプリケーションの構成 ..... 1436
6.18.5.3. SAP HANA インスタンスプロファイルの変更 ..... 1437
6.18.5.4. LifeKeeper ソフトウェアのインストール ..... 1438
6.18.6. SAP HANAリソースの設定タスク ..... 1439
6.18.6.1. SAP HANA リソース階層の作成 ..... 1440
6.18.6.2. SAP HANA リソース階層の拡張 ..... 1444
6.18.6.3. SAP HANAリソース階層の拡張解除 ..... 1449
6.18.6.4. SAP HANAリソース階層の削除 ..... 1451

6.18.6.5. SAP HANA リソース階層のテスト ..... 1453
6.18.7. SAP HANA リソース階層の管理 ..... 1466
6.18.7.1. レプリケーションモードおよびオペレーションモードの変更 ..... 1470
6.18.7.2. Takeover with Handshake ..... 1474
6.18.7.3. SAP HANA リソースのローカルリカバリーポリシーおよび TemporalRecoveryPolicy の設定 ..... 1479
6.18.7.4. マルチターゲットレプリケーション環境でのログ保持の管理 ..... 1484
6.18.8. SAP HANAトラブルシューティング ..... 1485
6.18.8.1. SAP HANA 階層の restore 失敗 ..... 1486
6.18.8.2. スプリットブレインシナリオの解決 ..... 1490
6.18.8.3. v9.7.0 未満のクラスターのローリングアップグレード中の障害 ..... 1495
6.19. SAP MaxDB Recovery Kit 管理ガイド ..... 1497
6.19.1. SAP MaxDB Recovery Kit の要件 ..... 1499
6.19.2. SAP MaxDB Recovery Kit の概要 ..... 1500
6.19.2.1. SAP MaxDB リソース階層 ..... 1501
6.19.3. SAP MaxDB の構成に関する考慮事項 ..... 1502
6.19.3.1. Raw I/O の使用 ..... 1503
6.19.3.2. DataKeeper でのミラーファイルシステムの使用 ..... 1504
6.19.3.3. 内部向けロードバランサーの使用 ..... 1505
6.19.3.4. アクティブ/スタンバイの場合の考慮事項 ..... 1506
6.19.3.4.1. アクティブ/スタンバイ設定の例 ..... 1508
6.19.3.5. アクティブ/アクティブの場合の考慮事項 ..... 1509
6.19.3.5.1. アクティブ/アクティブ設定の例 ..... 1510
6.19.4. LifeKeeper での SAP MaxDB の設定 ..... 1511
6.19.4.1. SAP MaxDB ソフトウェアのインストール ..... 1512
6.19.4.2. SAP MaxDB データベースの作成 ..... 1513
6.19.4.3. User_Key の作成 ..... 1514
6.19.4.4. LifeKeeper ソフトウェアのインストール ..... 1516
6.19.5. SAP MaxDB リソース設定作業 ..... 1517
6.19.5.1. SAP MaxDB リソース階層の作成 ..... 1518
6.19.5.2. SAP MaxDB リソース階層の拡張 ..... 1520
6.19.5.3. SAP MaxDB リソース階層の拡張解除 ..... 1522
6.19.5.4. SAP MaxDB リソース階層の削除 ..... 1523
6.19.5.5. リソース階層のテスト ..... 1524
6.19.6. LifeKeeper リソース階層の管理 ..... 1525
6.19.6.1. User_Key の変更 ..... 1526
6.19.6.2. OS User の変更 ..... 1527
6.19.6.3. パラメーターの更新 ..... 1528
6.19.7. SAP MaxDB トラブルシューティング ..... 1529
6.19.7.1. SAP MaxDB Recovery Kit エラーメッセージ ..... 1531
6.19.8. 付録：Raw I/O によるデバイス領域を作成する ..... 1534
6.19.8.1. Raw I/O セットアップ手順 ..... 1535
6.19.8.2. 階層作成後のデバイス領域の追加 ..... 1536
6.20. Sybase Recovery Kit 管理ガイド ..... 1537
6.20.1. Sybase ASE Recovery Kit の概要 ..... 1538

6.20.2. Sybase ASE Recovery Kit ハードウェア要件とソフトウェア要件 ..... 1540
6.20.3. Sybase ASE Recovery Kit 設定上の考慮事項 ..... 1542
6.20.3.1. raw I/O の使用 ..... 1543
6.20.3.2. DataKeeper でのレプリケーションファイルシステムの使用 ..... 1544
6.20.3.3. interfaces ファイルの考慮事項 ..... 1545
6.20.3.4. Sybase Software Asset Manager (SySAM) ..... 1546
6.20.3.5. アクティブ / スタンバイの考慮事項 ..... 1547
6.20.3.6. アクティブ / アクティブの考慮事項 ..... 1549
6.20.3.7. Sybase Monitor Server と Backup Server ..... 1551
6.20.3.8. Network Attached Storage の使用 ..... 1552
6.20.4. LifeKeeper を使用した Sybase ASE のインストールと設定 ..... 1555
6.20.4.1. Sybase ASE ソフトウェアのインストール ..... 1557
6.20.4.2. Sybase ASE サーバの作成 ..... 1558
6.20.4.3. LifeKeeper ソフトウェアのインストール ..... 1560
6.20.4.4. Sybase ASE リソース階層の作成 ..... 1561
6.20.4.5. Sybase ASE リソース階層の拡張 ..... 1563
6.20.4.6. Sybase ASE リソース階層の拡張解除 ..... 1565
6.20.4.7. Sybase ASE リソース階層の削除 ..... 1566
6.20.4.8. Sybase ASE のリソース階層のテスト ..... 1567
6.20.5. Sybase ASE Recovery Kit の管理 ..... 1568
6.20.5.1. Sybase Backup Server の保護の変更 ..... 1569
6.20.5.2. Sybase Monitor Server の保護の変更 ..... 1571
6.20.5.3. Sybase ASE Recovery Kit のパラメータの更新 ..... 1573
6.20.6. Sybase ASE Recovery Kit トラブルシューティング ..... 1574
6.20.7. Appendix – Sybase ASE で Raw I/O を使用してデバイス空間を作成する ..... 1578
6.20.7.1. raw I/O で Sybase ASE を使用するための要件 ..... 1579
6.20.7.2. 命名規則 ..... 1580
6.20.7.3. raw I/O の設定手順 ..... 1581
6.20.7.4. 階層作成後のデータベースデバイスの追加 ..... 1582
6.20.7.5. ASE と OCS のリンクの作成 ..... 1583
6.21. VMDK as Shared Storage Recovery Kit 管理ガイド ..... 1586
6.21.1. VMDK のドキュメンテーションとリファレンス ..... 1587
6.21.2. VMDK のハードウェア要件とソフトウェア要件 ..... 1588
6.21.3. VMDK Recovery Kit の概要 ..... 1589
6.21.4. LifeKeeper for Linux VMDK Recovery Kitの設定 ..... 1590
6.21.4.1. VMDK設定上の考慮事項 ..... 1591
6.21.4.2. VMDKの構成例 ..... 1592
6.21.5. LifeKeeper VMDK Recovery Kit の設定作業 ..... 1594
6.21.5.1. ESXiホストの登録 ..... 1595
6.21.5.2. 仮想マシンオプションの設定変更 ..... 1596
6.21.5.3. VMDK リソース階層の作成 ..... 1598
6.21.5.4. VMDK リソース階層の削除 ..... 1601
6.21.5.5. VMDK 階層の拡張 ..... 1603
6.21.5.6. VMDK 階層の拡張解除 ..... 1606
6.21.5.7. VMDK リソース階層のテスト ..... 1607

6.21.5.8. VMDK のメンテナンス ..... 1608
6.21.6. VMDK トラブルシューティング ..... 1609
6.21.6.1. VMDK エラーメッセージ ..... 1611

**7. パラメーター一覧 ..... 1617**
7.1. Recovery Kit for EC2パラメーター一覧 ..... 1624
7.2. IPパラメーター一覧 ..... 1627
7.3. LB Health Check パラメーター一覧 ..... 1630
7.4. MQパラメーター一覧 ..... 1631
7.5. NFSパラメーター一覧 ..... 1636
7.6. Recovery Kit for Oracle Cloud Infrastructure パラメーター一覧 ..... 1637
7.7. Oracleパラメーター一覧 ..... 1638
7.8. PostgreSQLパラメーター一覧 ..... 1639
7.9. Quorumパラメーター一覧 ..... 1641
7.10. Route53 パラメーター一覧 ..... 1645
7.11. SAPパラメーター一覧 ..... 1648
7.12. DataKeeperパラメーター一覧 ..... 1651
7.13. Standby Node Health Check パラメーター一覧 ..... 1658
7.14. SAP HANA パラメーターリスト ..... 1661
7.15. SAP MaxDB パラメーター一覧 ..... 1664

**8. エラーコード一覧 ..... 1666**
8.1. 総合メッセージカタログ ..... 1667
8.1.1. DataKeeper メッセージカタログ ..... 1969
8.1.2. DB2 Kit メッセージカタログ ..... 1993
8.1.3. DMMP Kit メッセージカタログ ..... 2007
8.1.4. EC2 Kit メッセージカタログ ..... 2022
8.1.5. ファイルシステムキットメッセージカタログ ..... 2034
8.1.6. Gen/App キットメッセージカタログ ..... 2056
8.1.7. IP Kit メッセージカタログ ..... 2068
8.1.8. LB Health Check メッセージカタログ ..... 2074
8.1.9. Recovery Kit for Oracle Cloud Infrastructure メッセージカタログ ..... 2077
8.1.10. Oracle キットメッセージカタログ ..... 2083
8.1.11. Oracle Listener キットメッセージカタログ ..... 2114
8.1.12. Oracle PDB キットメッセージカタログ ..... 2132
8.1.13. SCSI キットメッセージカタログ ..... 2137
8.1.14. Quick Service Protection Recovery Kit メッセージカタログ ..... 2139
8.1.15. Quorum メッセージカタログ ..... 2145
8.1.16. GUI メッセージカタログ ..... 2151
8.1.17. SAP メッセージカタログ ..... 2153
8.1.18. SAP HANA Recovery Kit メッセージカタログ ..... 2177

**9. LifeKeeper for Linuxサポートマトリックス ..... 2206**

**10. サポートストレージ ..... 2217**

**11. LifeKeeper for Linux 評価ガイド ..... 2239**

11.1. DataKeeper for Linux 評価ガイド ..... 2240
11.1.1. DataKeeper for Linux 関連用語 ..... 2241
11.1.2. 評価プロセス ..... 2244
11.1.3. DataKeeper for Linux インストールの準備 ..... 2245
11.1.4. DataKeeper for Linux ストレージの構成 ..... 2248
11.1.5. LifeKeeperのインストール ..... 2250
11.1.6. DataKeeper for Linuxクラスターの構成 ..... 2252
11.1.7. DataKeeper for Linux の環境テスト ..... 2266
11.2. クラウド環境向けLifeKeeper for Linux 評価ガイド ..... 2270
11.2.1. LifeKeeper for Linuxの評価を開始する前に ..... 2271
11.2.1.1. 高可用性、RTO（目標復旧時間）、RPO（目標復旧時点） ..... 2272
11.2.1.2. LifeKeeper for Linux – 統合コンポーネント ..... 2273
11.2.1.3. LifeKeeper for Linuxの利点 ..... 2274
11.2.1.4. クラウド環境に移行する際のワークロード（ノード）の分散方法 ..... 2275
11.2.1.5. パブリッククラウドプラットフォームとそのネットワーク構造の違い ..... 2277
11.2.1.6. クライアントがアクティブノードに接続する方法 ..... 2278
11.2.1.6.1. AWS Route Table シナリオ ..... 2280
11.2.1.6.2. AWS Elastic IPシナリオ ..... 2282
11.2.1.6.3. AWS Route53 シナリオ ..... 2283
11.2.1.6.4. Azure Internal Load Balancer シナリオ ..... 2284
11.2.1.6.5. Google Cloud Internal Load Balancer シナリオ ..... 2286
11.2.1.7. ノード間のデータレプリケーションが行われる方法 ..... 2287
11.2.1.8.「スプリットブレイン」とは何か、そしてそれを回避する方法 ..... 2288
11.2.2. 本書で使用するドキュメントのスタイル ..... 2290
11.2.3. ネットワークコンポーネントの構成とインスタンスの作成 ..... 2292
11.2.3.1. 本チュートリアルで使用するネットワーク構成 ..... 2293
11.2.3.2. 本チュートリアルで使用するコンピューティングリソース ..... 2295
11.2.3.3. AWSでゼロからインスタンスを作成する ..... 2298
11.2.3.3.1. AWSサービス間の切り替え ..... 2301
11.2.3.3.2. AWSリージョンの決定 ..... 2302
11.2.3.3.3. VPCの作成 ..... 2303
11.2.3.3.4. サブネットの作成 ..... 2306
11.2.3.3.5. インターネットゲートウェイの作成とVPCへの割り当て ..... 2311
11.2.3.3.6. ルートテーブルの作成 ..... 2313
11.2.3.3.7. セキュリティグループの作成 ..... 2317
11.2.3.3.8. 最初の EC2 インスタンスの作成 ..... 2322
11.2.3.3.9. 2番目と3番目のインスタンスの作成 ..... 2330
11.2.3.4. Azureでゼロからインスタンスを作成する ..... 2331
11.2.3.4.1. Azureサービスの切り替え ..... 2334
11.2.3.4.2. Azure リージョンの決定 ..... 2335
11.2.3.4.3. リソースグループの作成 ..... 2336
11.2.3.4.4. 仮想ネットワークの作成 ..... 2338
11.2.3.4.5. ネットワークセキュリティグループの作成 ..... 2341
11.2.3.4.6. 1つ目のAzure仮想マシンの作成 ..... 2347
11.2.3.4.7. 2番目と3番目の仮想マシンの作成 ..... 2357

11.2.3.5. Google Cloud でゼロからインスタンスを作成する ..... 2358
11.2.3.5.1. Google Cloudサービス間の切り替え ..... 2361
11.2.3.5.2. Google Cloud リージョンの決定 ..... 2362
11.2.3.5.3. プロジェクトの作成 ..... 2363
11.2.3.5.4. VPCネットワークの作成 ..... 2366
11.2.3.5.5. 新しいSSH認証鍵の作成 ..... 2368
11.2.3.5.6. 1つ目のGoogle Cloud VMの作成 ..... 2370
11.2.3.5.7. ファイアーウォールルールの設定 ..... 2379
11.2.3.5.8. 2番目と3番目の仮想マシンの作成 ..... 2384
11.2.4. LifeKeeper for Linuxを実行するためのLinuxノードの設定 ..... 2385
11.2.4.1. sshを使用してWindowsクライアントからLinuxノードに接続する ..... 2386
11.2.4.2. 各インスタンスにホスト名を設定する ..... 2390
11.2.4.3. SELinux を無効にする ..... 2392
11.2.4.4. ファイアーウォールを無効にする ..... 2393
11.2.4.5. rootユーザーのパスワードを設定する ..... 2396
11.2.4.6. x11をインストールする ..... 2397
11.2.5. LifeKeeper for Linux のインストール ..... 2399
11.2.5.1. Amazon Web Services ..... 2404
11.2.5.1.1. AWS CLIをインストールする ..... 2405
11.2.5.1.2. Recovery Kit for EC2を使用するためのアクセス権限を割り当てる ..... 2406
11.2.5.1.3. AWS – PINGブロードキャストを無効にする ..... 2407
11.2.5.1.4. AWS – 送信元/宛先チェックを無効にする ..... 2408
11.2.5.2. Microsoft Azure ..... 2409
11.2.5.2.1. Azure – PING ブロードキャストを無効にする ..... 2410
11.2.6. ログインと基本設定タスク ..... 2411
11.2.6.1. x11転送を使用したLinuxノードへの接続 ..... 2413
11.2.6.2. Microsoft Windowsにクライアントソフトウェアをインストールする ..... 2415
11.2.6.3. 1番目のノードに接続する (node-a) ..... 2419
11.2.6.4. 他のノードに接続する (node-b and node-c) ..... 2421
11.2.6.5. コミュニケーションパスを定義する ..... 2423
11.2.7. リソースの保護 ..... 2427
11.2.7.1. IP リソースの作成 ..... 2430
11.2.7.2. クラウド環境でのノードの切り替え ..... 2435
11.2.7.2.1. AWS EC2リソースの作成（ルートテーブルシナリオ） ..... 2436
11.2.7.2.2. AWS EC2 リソースの作成（Elastic IP シナリオ） ..... 2442
11.2.7.2.3. AWS Route 53リソースの作成 ..... 2446
11.2.7.2.4. Azure – 内部ロードバランサーの使用 ..... 2457
11.2.7.2.5. Google Cloud – 内部ロードバランサーの使用 ..... 2464
11.2.7.2.6. ロードバランサーのヘルスチェックに応答する ..... 2475
11.2.7.3. スタンバイノードに切り替えて、スイッチオーバーが機能していることを確認する ..... 2486
11.2.7.4. ファイルシステムのデータレプリケーションを作成する方法 ..... 2489
11.2.7.4.1. AWS でレプリケーション用にディスクを準備する方法 ..... 2496
11.2.7.4.2. Azure でレプリケーション用にディスクを準備する方法 ..... 2499
11.2.7.4.3. Google Cloudでレプリケーション用にディスクを準備する方法 ..... 2501

11.2.7.5. 他のリソース（データーベースまたはアプリケーション）を保護する方法 ..... 2503
11.2.7.5.1. Oracleリソースの保護（PDBを除く） ..... 2504
11.2.7.5.1.1. Oracleのインストール ..... 2509
11.2.7.5.1.2. Oracleデータベースの作成（PDBを除く） ..... 2510
11.2.7.5.1.3. Oracleインスタンスを停止する ..... 2512
11.2.7.5.1.4. /datakeeper/oradata/ORCLの名前を変更する ..... 2514
11.2.7.5.1.5. 両方のノードでOracleリスナーの設定ファイルを更新する ..... 2516
11.2.7.5.1.6. Node-aでデータベースとリスナーを起動する ..... 2518
11.2.7.5.1.7. Oracleリスナーリソースの設定 ..... 2519
11.2.7.5.1.8. Oracleリソースの設定 ..... 2526
11.2.7.5.1.9. Oracleリソースのスイッチオーバーをテストする ..... 2531
11.2.7.5.2. Quick Service Protectionを使用したMicrosoft SQL Serverの保護 ..... 2535
11.2.7.5.2.1. Microsoft SQL Server 2017のインストール ..... 2539
11.2.7.5.2.2. マスターデーターベースとログファイルを複製されたストレージに再配置する ..... 2540
11.2.7.5.2.3. /dataKeeper/MSSQLの下のフォルダーの名前を変更する ..... 2542
11.2.7.5.2.4. MSSQLリソースの設定 ..... 2544
11.2.7.5.2.5. 両方のノードでLocalRecoveryパラメーターをカスタマイズする ..... 2550
11.2.7.5.2.6. リソース間の依存関係を更新する ..... 2552
11.2.7.5.3. PostgreSQLリソースの保護 ..... 2555
11.2.7.5.3.1. LinuxノードにPostgreSQL 12をインストールする ..... 2559
11.2.7.5.3.2. NFSリソースの保護 ..... 2561
11.2.7.5.4. Protecting MySQL Resources ..... 2566
11.2.7.5.4.1. Prepare to Install ..... 2567
11.2.7.5.4.2. Configure Storage ..... 2571
11.2.7.5.4.3. Install, Configure and Start MySQL ..... 2574
11.2.7.5.4.4. Install LifeKeeper for Linux – MySQL Cluster ..... 2578
11.2.7.5.4.5. Configure the Cluster ..... 2581
11.2.7.5.5. SAPリソースの保護 ..... 2583
11.2.7.5.5.1. ASCSおよびERSの仮想IPを作成する ..... 2589
11.2.7.5.5.1.1. AWS – ASCSおよびERSの仮想IPを作成する ..... 2590
11.2.7.5.5.1.2. Azure – ASCSおよびERSの内部ロードバランサーを作成する ..... 2596
11.2.7.5.5.1.3. Google Cloud – ASCSおよびERSの内部ロードバランサーを作成する ..... 2604
11.2.7.5.5.2. SAP ファイルシステムを作成する ..... 2613
11.2.7.5.5.2.1. AWS/Azure – SAP共有ファイルシステムおよびレプリケーションファイルシステムの作成 ..... 2614
11.2.7.5.5.2.2. Google Cloud – SAP 共有ファイルシステムおよびレプリケーションファイルシステムの作成 ..... 2625
11.2.7.5.5.3. SAPインスタンスのインストール ..... 2637
11.2.7.5.5.4. LifeKeeper SAP リソースを作成する ..... 2648
11.2.7.5.5.4.1. AWS/Azure – LifeKeeper SAP リソースを作成する ..... 2649
11.2.7.5.5.4.2. Google Cloud – LifeKeeper SAP リソースを作成する ..... 2655
11.2.7.5.5.5. スイッチオーバーとフェイルオーバーのテスト ..... 2661
11.2.7.5.6. SAP HANA リソースの保護 ..... 2672

11.2.7.5.6.1. SAP HANA プライマリーデータベースの仮想 IP を作成する ..... 2676
11.2.7.5.6.1.1. AWS – SAP HANA 仮想 IP を作成する ..... 2677
11.2.7.5.6.1.2. Azure – SAP HANA プライマリーデータベースロードバランサーを作成する ..... 2680
11.2.7.5.6.1.3. Google Cloud – SAP HANA プライマリーデータベースロードバランサーを作成する ..... 2686
11.2.7.5.6.2. SAP HANA ファイルシステムのディスクをアタッチする ..... 2692
11.2.7.5.6.3. SAP HANA のインストールとシステムレプリケーションの設定 ..... 2694
11.2.7.5.6.4. LifeKeeper SAP HANA リソースを作成する ..... 2700
11.2.7.5.6.4.1. AWS/Azure – LifeKeeper SAP HANA リソースを作成する ..... 2701
11.2.7.5.6.4.2. Google Cloud – LifeKeeper SAP HANA リソースを作成する ..... 2703
11.2.7.5.6.5. スイッチオーバーとフェイルオーバーのテスト ..... 2706
11.2.7.6. 共通タスク ..... 2715
11.2.7.6.1. ノードでデータストレージが利用可能かどうかを確認する方法 ..... 2716
11.2.7.6.2. 他ノードへのデータストレージのスイッチオーバー ..... 2718

**12. Quick Start Guides ..... 2720**
12.1. AWS Direct Connect クイックスタートガイド ..... 2721
12.1.1. 利用のための要件 ..... 2722
12.1.2. AWS Direct Connect 構築手順 ..... 2724
12.1.2.1. AWS Direct Connect 準備 ..... 2725
12.1.2.2. Direct Connect リソースの作成 ..... 2726
12.1.2.3. ルートテーブルの設定 ..... 2727
12.1.3. 本構成における設定および運用上の留意点 ..... 2728
12.2. Microsoft Azure クイックスタートガイド ..... 2729
12.2.1. Microsoft Azure の概要 ..... 2730
12.2.2. Azure Shared Storageのセットアップ方法 ..... 2731
12.2.3. 構成情報 ..... 2742
12.2.4. Azure 上の LifeKeeper 特有の設定について ..... 2746
12.2.5. 仮想マシンの構築、OS の起動 ..... 2748
12.2.5.1. リソースグループの作成 ..... 2749
12.2.5.2. 仮想ネットワークの作成 ..... 2750
12.2.5.3. 仮想マシンの作成 ..... 2752
12.2.5.3.1. クラスターノード（稼働系）の作成 ..... 2753
12.2.5.3.2. クラスターノード（待機系）の作成 ..... 2765
12.2.5.3.3. クライアント兼 Witness サーバーの作成 ..... 2768
12.2.5.4. ロードバランサーの作成 ..... 2771
12.2.5.5. OSの設定 ..... 2777
12.2.6. LifeKeeper による HA クラスターの構築 ..... 2784
12.2.6.1. コミュニケーションパスの作成 ..... 2786
12.2.6.2. Quorum / Witness の設定 ..... 2787
12.2.6.3. LB Health Checkリソース階層の作成 ..... 2788
12.2.6.4. IP リソースの作成 ..... 2792
12.2.6.5. Data Replication リソース階層の作成 ..... 2795
12.2.6.6. Oracle リソース階層の作成 ..... 2800
12.2.6.6.1. Oracle DB のインストール ..... 2804

12.2.6.6.2. リスナー設定 ... 2806
12.2.6.6.3. DB の作成 ... 2809
12.2.6.6.4. Oracle の設定 ... 2811
12.2.6.6.5. Oracle database listener リソース階層の作成 ... 2813
12.2.6.6.6. Oracleリソース階層の作成 ... 2817
12.2.6.6.7. Oracle Pluggable Databaseリソース階層の作成 ... 2821
12.2.6.6.8. 依存関係の作成 ... 2825
12.2.6.7. 接続確認 ... 2827
12.2.7. Availability Zone (高可用性ゾーン） ... 2831
12.2.7.1. Azure の構成 ... 2833
12.2.7.2. クライアントリダイレクトの動作確認方法 ... 2836
12.2.8. 参考資料 ... 2837
12.3. LifeKeeper for Linux in the AWS Cloud (SAP) ... 2838
12.3.1. LifeKeeper 上で SAP を設定する際の追加手順 ... 2839
12.3.2. NFS なしの ASCS 構成 ... 2845
12.3.2.1. NFS を使用しない ASCS の一般的なセットアップ手順 ... 2846
12.3.2.2. SAP のインストール ... 2847
12.3.2.3. SAP リソース階層の作成 ... 2849
12.3.3. NFS を使用している ASCS + ERS 構成 ... 2854
12.3.3.1. 一般的なセットアップ手順 ... 2855
12.3.3.2. SAP のインストール（NFS使用） ... 2857
12.3.3.3. NFS のセットアップ ... 2859
12.3.3.4. NFS リソース階層の作成 ... 2865
12.3.3.5. SAP リソース階層の作成 ... 2868
12.3.3.6. ERS リソースの作成 ... 2873
12.3.3.7. ENSAv2/ERSv2使用時にASCS/ERSに回避動作を強制する ... 2875
12.3.4. スイッチオーバーおよびフェールオーバーのテスト ... 2884
12.3.4.1. その他のリソース ... 2886
12.4. AWS Transit Gatewayを使用したLifeKeeperクラスターとクライアント間接続クイックスタートガイド ... 2887
12.4.1. 利用のための要件 ... 2889
12.4.1.1. LifeKeeper ソフトウェアの要件 ... 2891
12.4.2. AWS VPC ピア接続構築手順 ... 2892
12.4.3. ルートテーブルの設定 ... 2893
12.4.4. 本構成における設定および運用上の留意点 ... 2894
12.4.4.1. LifeKeeper I/O フェンシングの利用を検討してください ... 2895
12.4.5. AWS Direct Connect 既知の問題とトラブルシューティング ... 2896
12.5. AWS VPC ピアリング接続を使用した複数 VPC クラスター構成クイックスタートガイド ... 2897
12.5.1. 利用のための必要要件 ... 2899
12.5.1.1. LifeKeeper ソフトウェアの要件 ... 2900
12.5.1.2. その他 ... 2901
12.5.2. 構築手順 ... 2902
12.5.3. 関連する LifeKeeper リソースについて ... 2904
12.5.4. 本構成における設定および運用上の留意点 ... 2905
12.5.4.1. Quorum/Witness もしくは STONITH の利用を検討してください ... 2906

12.6. PostgreSQL Cluster with Shared Storage (ISCSI) ..... 2907
12.6.1. Terms to Know – PostgreSQL ..... 2908
12.6.2. The Evaluation Process – PostgreSQL ..... 2911
12.6.3. Prepare to Install – PostgreSQL ..... 2912
12.6.4. Configure Storage – PostgreSQL ..... 2916
12.6.5. Install, Configure and Start PostgreSQL ..... 2918
12.6.6. Install LifeKeeper for Linux – PostgreSQL ..... 2920
12.6.7. Configure the Cluster – PostgreSQL ..... 2923
12.6.8. Test Your Environment – PostgreSQL ..... 2938
12.7. Apache/MySQL Cluster Using Both Shared and Replicated Storage ..... 2943
12.7.1. Terms to Know – Apache ..... 2944
12.7.2. The Evaluation Process – Apache ..... 2947
12.7.3. Prepare to Install – Apache ..... 2948
12.7.4. Configure Storage – Apache ..... 2952
12.7.5. Install and Configure Apache and PHP ..... 2956
12.7.6. Install, Configure, and Start MySQL – Apache ..... 2958
12.7.7. Install LifeKeeper for Linux – Apache ..... 2961
12.7.8. Configure the Cluster – Apache ..... 2964
12.7.9. Test Your Environment – Apache ..... 2987

**13. LifeKeeper Single Server Protection ..... 2998**
13.1. LifeKeeper Single Server Protection for Linux リリースノート ..... 2999
13.2. LifeKeeper Single Server Protection インストレーションガイド ..... 3012
13.2.1. はじめに ..... 3014
13.2.2. LifeKeeper Single Server Protection ソフトウェアのインストール ..... 3016
13.2.3. セットアップスクリプトの操作 ..... 3018
13.2.4. SSP のアップグレード ..... 3025
13.2.5. LifeKeeper Single Server Protection ライセンスについて ..... 3026
13.2.5.1. ライセンスの取得方法 ..... 3027
13.2.5.2. ライセンスのインストール方法 ..... 3028
13.2.6. リソースポリシー管理 ..... 3029
13.2.7. LifeKeeper Single Server Protection のインストールの検証 ..... 3034
13.3. LifeKeeper Single Server Protection テクニカルドキュメンテーション ..... 3035
13.3.1. ドキュメンテーションとトレーニング ..... 3037
13.3.2. VMware HA との連携 ..... 3039
13.3.3. LifeKeeper Single Server Protection の管理の概要 ..... 3040
13.3.3.1. VMware HA と LifeKeeper Single Server Protection の連携を有効にする ..... 3041
13.3.3.2. VMware HA を有効化した障害検出およびリカバリシナリオ ..... 3042
13.3.3.2.1. LifeKeeper Single Server Protection で保護するシステムのメンテナンス ..... 3043
13.3.3.3. LifeKeeper Single Server Protection ハートビートと VMware HA ..... 3044
13.3.3.4. Quick Service Protection (QSP) Recovery Kit ..... 3045
13.3.3.5. LifeKeeper API for Monitoring ..... 3046
13.3.3.6. Watchdog ..... 3047
13.3.3.7. LKCLI（LifeKeeper コマンドラインインターフェース） ..... 3050
13.3.4. トラブルシューティング ..... 3052
13.3.4.1. 既知の問題と回避策 ..... 3053

13.4. アプリケーションリカバリーキット ..... 3057

**14. プロダクトライフサイクル ..... 3058**

# 1. LifeKeeper for Linux

# 2. LifeKeeper for Linux リリースノート

## Version 9.7.0

リリース日： **March 28, 2023**

**重要!!**

**重要** ：SIOS は次の Recovery Kit のサポートを終了します。これらは LifeKeeper for Linux の次のメジャーリリースでサポートされなくなります。
(なお、新規受注は2022年12月31日をもって終了しています。)
- NEC iStorage StoragePathSavior
- Samba
- Postfix
- HULFT HUB Recovery Kit

**重要** ：業界のベストプラクティスに従い、v9.6.2 の新規環境、v9.7.0 以降のすべての環境において、SIOS では SAP システムのセットアップ時に、NFS サーバー/クライアントを SAP サービスと同じノードにインストールすること (「ループバック構成」とも呼ばれます) をサポートしません。 代わりに、次のような代替構成を使用してください。
- EFS in AWS
- Azure Premium File Shares with NFS
- Google NFS Share
- NFS as separate cluster

**この製品をインストールまたは使用する前に、このドキュメントをお読みください。このドキュメントには、インストール前、インストール中、およびインストール後に考慮する必要がある最新情報が含まれています。**

## はじめに

このリリースノート（以下本文書）は、 LifeKeeper for Linux 製品のインストール、設定、管理を行うユーザーを対象として記述されています。本文書には、製品のテクニカルドキュメンテーション には詳細に記述されていない重要な情報（製品の最終テスト時に明らかになったパッケージのバージョン、指示や手順に関する最終段階での変更点、トラブルシューティングセクションへのリンク、製品の制限、トラブル解決のヒントなど）が記載されています。 LifeKeeper for Linux をインストールして設定する前に、必ず本文書の内容を確認してください。

### クイックリンク

このリリースの新機能

バグの修正
Hotfixes と Add-on サポートパッケージ
廃止された機能
LifeKeeper コンポーネント
システム要件
オープンソースパッケージ
インストールと構成
アップグレード
ストレージとアダプタのオプション
テクニカルノート
既知の問題

# SIOS の製品説明

## LifeKeeper for Linux

LifeKeeper for Linux（以下LifeKeeper）は、Linux 上で動作する様々なリソースの高可用性を実現する、アクティブ･スタンバイ型の HAクラスターソフトウェアです。LifeKeeper によって保護されたリソースは、障害発生時やメンテナンス時、稼働するサーバーを切り替えてサービスを継続することが可能になります。稼働するサーバーの切り替えが行われた際も、リソースは切り替え前と同等の環境･同等のパフォーマンスでサービスを提供し続けることができます。これはアクティブ･スタンバイ型クラスターの特性です。クラスターを構成するサーバーのうち、主として起動するサーバーを稼働系、その他のサーバーを待機系と呼びます。LifeKeeperでは、待機系サーバーの OS を常時起動しておくことで、切り替えに要する時間を短縮し、結果としてシステムの高い可用性を実現することができます。

LifeKeeper が保護できるリソースには、次の種類があります。

- アプリケーション
- 共有ストレージデバイス（VMWareの仮想ハードディスクを含む）
- Network Attached Storage デバイス
- LVM ボリュームグループと論理ボリューム
- ファイルシステム (ext3、ext4、vxfs、xfs、nfs)

> 注意：btrfsは、現在 LifeKeeper for Linux でサポートされていません。 bind マウントもサポートされません。詳細は、トラブルシューティング > 既知の問題と制限 をご確認ください。

- 通信リソース（TCP/IP）
- データベースアプリケーション (Oracle 、 MySQL 、 DB2 、 SAP MaxDB 、 PostgreSQL 、 EnterpriseDB Postgres Plus Advanced Server/EDB Postgres Advanced Server 、 Sybase )
- Web サーバリソース
- Samba リソース
- DataKeeper for Linux
- SAP アプリケーション環境リソース
- WebSphere MQ リソース

- Postfix リソース

# DataKeeper for Linux

DataKeeper for Linux（以下DataKeeper）は、ボリュームベースの同期および非同期のデータレプリケーション機能を提供するソフトウェアです。LifeKeeper によって手順のリソース切り替え（スイッチオーバー）や障害を契機とした自動切り替え（フェイルオーバー）が行われた際、本製品のレプリケーション機能によって、サーバー間のデータを自動的に同一に保ちます。レプリケーションの管理および監視は、LifeKeeper Graphical User Interface（LifeKeeper GUI）より統合的に行えます。

# LifeKeeper for Linux バージョン 9 / サポート対象バージョンの新機能

古いバージョンのドキュメントについては、 https://docs.us.sios.com/alllinux.php を参照してください。

| 製品 | 機能 |
|---|---|
| バージョン **9.7.0** の新機能 | |
| LifeKeeper Core | Red Hat Enterprise Linux 9.1 をサポートしました（2023年6月認定） |
| | Red Hat Enterprise Linux 9.0 をサポートしました。<br>**注意** ：カーネルの問題により、シンプロビジョニングディスク上の DataKeeper 非同期モードは使用できません。 |
| | Red Hat Enterprise Linux 8.7 をサポートしました。(2023年6月認定)<br>**注意** ：DataKeeperを使用する場合、LifeKeeperインストール時に特定の手順に従う必要があります。詳細は こちら をご参照ください。 |
| | セットアップ実行時のファイアウォール検出を強化しました。 |
| | Rocky Linux 9.0 をサポートしました。<br>**注意** ：カーネルの問題により、シンプロビジョニングディスク上の DataKeeper 非同期モードは使用できません。<br>• Rocky Linux は GCP および OCI をサポートしていません。<br>• Rocky Linux v9.0 は、LifeKeeper for Linux v9.7.0 でのみサポートされています。<br>• SAP Recovery Kit および SAP HANA Recovery Kit はサポートされていません。 |
| | Rocky Linux 8.7 をサポートしました。(2023年6月認定)<br>**注意** ：DataKeeper を使用する場合、LifeKeeper インストール時に特定の手順に従う必要があります。詳細は こちら をご参照ください。<br>• SAP Recovery Kit と SAP HANA Recovery Kit はサポートされていません。 |

| | |
|---|---|
| | VMware vSphere 8.0 をサポートしました。(2023年6月認定) 注意: vSAN は VMware Cloud on AWS 上でのみサポートされます。 |
| | VMware Cloud on AWS（シングルAZ構成）上でのDataKeeperの利用をサポートしました。（2023年7月認定） |
| | VMware Cloud on AWS（シングルAZ構成）上でのReservationを使用しない共有ディスクの利用（別途 IOフェンシングの対応 が必須）をサポートしました。（2023年8月認定） |
| | Oracle Linux 9.1 をサポートしました。（2023年6月認定） |
| | Oracle Linux 9.0 をサポートしました。（2023年6月認定）<br>**注意** ：カーネルの問題により、シン プロビジョニングディスク上の DataKeeper 非同期モードは使用できません。 |
| | Oracle Linux 8.7 をサポートしました。（2023年6月認定）<br>**注意** ：DataKeeper を使用する場合、LifeKeeper インストール時に特定の手順に従う必要があります。詳細は こちら をご参照ください。 |
| | Miracle Linux 9.0 をサポートしました。（2023年6月認定）<br>**注意** ：カーネルの問題により、シンプロビジョニングディスク上の DataKeeper 非同期モードは使用できません。 |
| PostgreSQL Recovery Kit | PostgreSQL 15 をサポートしました。 |
| | PowerGres Plus V13をサポートしました。（2023年3月認定） |
| NFS | LifeKeeper は、サーバーがファイルシステムをエクスポートし、エクスポートされたファイルシステムを同じシステムにマウントする NFS ループバック構成をサポートしません。 NFS リソースのインストール / 更新または作成中に、サポートされていない構成が検出された場合、エラーメッセージがログに記録されます。 |
| LVM | RHEL9 で追加された LVM デバイスファイル機能をサポートしました。 |
| SAP HANA | SAP HANA Recovery Kit は、最大 3 つのセカンダリーレプリケーションサイトへのマルチターゲットシステムレプリケーションをサポートしました。 |
| バグの修正 | |
| **バージョン 9.6.2 の新機能** | |
| LifeKeeper Core | Rocky Linux 8.5 をサポートしました。(2023年4月に認定)<br>• Rocky Linux 8.5 は GCP および OCI をサポートしていません。<br>• SAP Recovery Kit と SAP HANA Recovery Kit はサポートしていません。 |
| | Rocky Linux 8.6 をサポートしました。<br>**注意:** カーネルの問題により、シンプロビジョニングディスク上の DataKeeper 非同期モードは使用できません。<br>• Rocky Linux 8.6 は GCP および OCI をサポートしていません。<br>• SAP Recovery Kit と SAP HANA Recovery Kit はサポートしていません。 |
| | SLES 15 SP4 をサポートしました。(2023 年 1 月に認定) LifeKeeper を SLES 15 SP 4 にインストールするには、 こちら の指示に従って、lk-sles15sp4-support を |

| | |
|---|---|
| | ダウンロードして ‘setup --addinfo lk-sles15sp4-support’ を実行する必要があります。<br>**注意:** Kernel 5.14.21-150400.24.21.2 以降が必要です。 |
| | 新しい steeleye-lk SCSI3 Recovery Kit を使用した Azure 共有ディスクのサポートが追加されました。SCSI3リカバリーキットは、Linux SD サブシステム（/dev/sd デバイスノード）を使用するシングルパスデバイスのための SCSI-3 持続的予約のサポートを追加します。このキットは、共有ストレージを持つLifeKeeper クラスタ内の Microsoft Azure 共有ディスクの信頼性の高いロックを可能にします。 |
| | Red Hat Enterprise Linux 8.6 をサポートしました。<br>**注意** : kernel の問題に起因して シンプロビジョニングディスクで DataKeeper の非同期モードは利用できません。 |
| | Oracle Linux 8.6 をサポートしました。<br>**注意** : kernel の問題に起因して シンプロビジョニングディスクで DataKeeper の非同期モードは利用できません。 |
| | Miracle Linux 8.4をサポートしました。 |
| | STANDBY NODE WRITE PROTECTION 機能を追加しました。<br>・RESERVATIONSを無効にした時に、不注意な意図しないアクセスによるデータ破損を防ぐための機能です。詳細は テクニカルドキュメンテーション を参照してください。 |
| LifeKeeper LB Health Check Kit | LifeKeeper LB（ロードバランサー）ヘルスチェックキットは、GenLB（ジェネリックロードバランサー）リカバリーキットに代わるものです。 GenLB リカバリーキットは、ジェネリックアプリケーション（gen/app）リソースで使用するための 9.6.1 用にダウンロード可能なアクションスクリプトのコレクションでした。<br>GenLB キットは、使いやすくするために完全な ARK に移行されました。<br>• GenLB (Generic Load Balancer) Recovery Kit は v9.6.2 以降の新しいデプロイメントではサポートされません。<br>• SIOS は、v9.6.2 にアップデートしても GenLB Kit を使用し続ける構成に対して、KBA (Knowledge Based Article) サポートのみを提供する予定です。<br>• 有効なサポートをお持ちのお客様には、Generic Load balancer RK を使用した v9.6.1 のサポートが提供されます。<br>• SIOSは、v9.6.2へのアップグレードと 新しい LifeKeeper LB Health Check Kit へのアップグレード を強く推奨します。 |
| バグ の修正 | |
| **バージョン 9.6.1 の新機能** | |
| LifeKeeper Core | Rocky Linux 8.4 をサポートしました。<br>・Rocky Linux は、クラウドには対応していません。(例: AWS/Azure/GCP/OCI)<br>・LifeKeeper Single Server Protection はサポートされていません。<br>・SAP Recovery Kit と SAP HANA Recovery Kit はサポートされていません。 |
| | Oracle Linux 8.5 をサポートしました。<br>注意: RHCKでDataKeeperを利用する場合には、LifeKeeperインストール時に特定 |

| | |
|---|---|
| | の手順に従う必要があります。詳細は こちら をご参照ください。 |
| | Red Hat Enterprise Linux 8.5 をサポートしました。 |
| | バグの修正 |
| SAP Recovery Kit | SAP Recovery Kit は、LifeKeeper 固有の critical_nfs_mounts_<tag> ファイルの使用をサポートするようになりました。指定されたサーバー上の指定されたSAPリソースのこのファイルに追加されたNFSマウントエントリー (例えば、/sapmnt/<SID> NFS共有) は、リソースが In Service になる前にマウントされます。 |
| | バグの修正 |
| Recovery Kit for Oracle Cloud Infrastructure | Oracle Cloud 環境でLifeKeeper IP Recovery Kit を用いた仮想 IP アドレスによる通信ができるようになりました。 |
| NAS Recovery Kit | UDPが無効になっているNFSを保護する際に、「NFS_RPC_PROTOCOL=tcp」を設定する必要がなくなりました。 |
| | バグの修正 |
| GUI、CLI、DataKeeper、Route53、Quorum/Witness、NAS、EC2 | バグの修正 |
| PostgreSQL Recovery Kit | PowerGres on Linux 13 をサポートしました。（2022年5月認定） |
| | FUJITSU Software Enterprise Postgres 14 をサポートしました。(2022年10月認定) |
| **バージョン 9.6.0 の新機能** | |
| LifeKeeper Core | Oracle Linux 8.4 をサポートしました。 |
| | SUSE Linux Enterprise Server 15 SP3 をサポートしました。<br>※サポートされるKernel Versionは5.3.18-59.5以降です。 |
| | Microsoft Azure 上で STONITH 機能を利用できるようになりました。 |
| | バグの修正 |
| Recovery Kit for EC2 ™; | EC2 ARK のルートテーブルシナリオで、送信元/送信先チェックを確認して、有効の場合は、無効にする機能が追加されました。新たな IAM 権限 (ec2:DescribeNetworkInterfaceAttribute, ec2:ModifyNetworkInterfaceAttribute) が必要になります。 |
| Oracle Recovery Kit | オンプレミス上で動作するOracle 21c(21.3)をサポートしました。 |
| PostgreSQL Recovery Kit | FUJITSU Software Enterprise Postgres 13 （Advanced、Standard、Community）をサポートしました。 |
| | PostgreSQL14 をサポートしました |
| | EDB Postgres Advanced Server 14.0をサポートしました (2022年1月認定) |

| | |
|---|---|
| SAP Recovery Kit | Red Hat Enterprise Linux 8.4 をサポートしました。 |
| | SUSE Linux Enterprise Server 15 SP3 をサポートしました。 |
| | SAP S/4HANA 2021 Platform（SAP kernel 7.85）をサポートしました。（2022年3月認定） |
| | バグの修正 |
| SAP HANA Recovery Kit | Red Hat Enterprise Linux 8.4 をサポートしました。 |
| | SUSE Linux Enterprise Server 15 SP3 をサポートしました。 |
| | SAP HANA2 SP6 をサポートしました。 (2022年2月認定) |
| | ローカルリカバリーの機能強化(デモ動画 (英語版) はこちら) |
| | バグの修正 |
| IP、VMDK、NFS | バグの修正 |
| **バージョン 9.5.2 の新機能** | |
| LifeKeeper Core | Red Hat Enterprise Linux 8.4 をサポートしました。 |
| | CentOS 8.3 をサポートしました。 |
| | Oracle Linux 8.3 をサポートしました。 |
| | Oracle Linux 7 UEK 6をサポートしました。<br>注意：カーネルバージョンが5.4.17-2102.202.5 以降のみサポートされます。 |
| | Oracle Linux 8 UEK 6をサポートしました。<br>注意：カーネルバージョンが5.4.17-2102.202.5 以降のみサポートされます。 |
| | 以下の ARK で LKCLI の利用をサポートしました。<br>• SAP RK (デモ動画 (英語版) はこちら)<br>• SAP MaxDB RK<br>• MQ RK (デモ動画 (英語版) はこちら)<br>• DMMP RK<br>• HDLM RK<br>• NEC LifeKeeper RK |
| | 以下のARKにおいてスタンバイノード監視機能が利用できるようになりました。<br>• Filesystem RK<br>• NAS RK |
| | lk group に所属したroot以外のユーザーからでも lkcliを実行できるようになりました。 |
| | lkcli stop でも lkstop 同様に「-i」オプションを使用できるようになりました。 |
| | バグの修正 |
| MySQL | MariaDB 10.5 をサポートしました。 |
| SAP HANA | SAP HANA support for RHEL 8.2 をサポートしました。 |

| | |
|---|---|
| | SAP HANA support for SLES 15.2 をサポートしました。 |
| | SAP HANA データベースの Takeover with Handshake機能をサポートしました。 |
| | /etc/default/LifeKeeperに調整可能なパラメーターを追加。 |
| | バグの修正 |
| MQ | WebSphere MQ 9.2 for RHEL 7.9 をサポートしました。 |
| | WebSphere MQ 9.2 for RHEL 8.3 をサポートしました。 |
| DataKeeper、Install、NFS、IP、Oracle、SAP、Quorum/Witness、Filesystem、Generic、lksupport、DMMP | バグの修正 |
| **バージョン 9.5.1 の新機能** | |
| LifeKeeper Core | Red Hat Enterprise Linux 7.9 をサポートしました。（2020年12月認定）<br>注意: DataKeeperを使用する場合、LifeKeeperインストール時に特定の手順に従う必要があります。詳細は こちら をご参照ください。 |
| | CentOS 7.9 をサポートしました。（2020年12月認定）<br>注意: DataKeeperを使用する場合、LifeKeeperインストール時に特定の手順に従う必要があります。詳細は こちら をご参照ください。 |
| | Oracle Linux 7.9 をサポートしました。（2020年12月認定）<br>注意: DataKeeperを使用する場合、LifeKeeperインストール時に特定の手順に従う必要があります。詳細は こちら をご参照ください。 |
| | Microsoft Azure の Load Balancer を利用した構成において、任意のポートでヘルスチェックプローブに応答する機能を追加。（2020年12月認定）<br>詳細は こちら をご参照ください。 |
| | Google Cloud ™; の Cloud Load Balancing を利用した構成において、任意のポートでヘルスチェックプローブに応答する機能を追加。（2020年12月認定）<br>詳細は こちら をご参照ください。 |
| | 以下の ARK で LKCLI の利用をサポートしました。<br>• DB2 RDBMS RK<br>• RAW RK<br>• Postfix RK<br>• VMDK as Shared Storage RK<br>• SAP ASE RK<br>• Samba RK<br>• HULFT RK / HULFT HUB RK<br>• Oracle RK の Oracle PDB 保護機能 |
| | Red Hat Enterprise Linux 8.3 をサポートしました。（2021年1月認定） |

| | |
|---|---|
| | Red Hat Enterprise Linux 8.2 をサポートしました。 |
| | CentOS 8.2 をサポート。 |
| | Oracle Linux 8.2（UEK6 は除く）をサポートしました。 |
| | SLES15 SP2 をサポートしました。 |
| | LifeKeeper 内部で使用している OpenSSL パッケージを 1.1.1g に更新。 |
| | LifeKeeper 内部で使用している cURL パッケージを 7.86.0 に更新。 |
| | lkstop コマンドに -i オプション を追加。このオプションは保護するリソースを停止せずに LifeKeeper のみを停止します。また、停止処理を続行するために「yes/no」の確認を行う必要があります。 |
| | バグの修正 |
| MQ | SIOS Protection for Linux は IBM MQ 9.2 の保護をサポートしました。(2021年1月) |
| QSP | Quick Service Protection(QSP) RK によるApache Tomcat の保護をサポート。 |
| install | LifeKeeper の 構成のみ行うセットアップのモード をサポート。その他、セットアップに関する改修。 |
| | バグの修正 |
| PostgreSQL | PostgreSQL 13 をサポートしました。（2020年12月認定） |
| | EDB Postgres Advanced Server 13.0 をサポートしました。（2020年12月認定） |
| | FUJITSU Software Enterprise Postgres 12 をサポートしました。（2020年12月認定） |
| SAP HANA | SAP HANA2 SP5 をサポートしました。（2020年12月認定） |
| | SAP HANA support for RHEL 8.1 をサポートしました。（2020年12月認定） |
| | SAP HANA support for SLES 12.5 をサポートしました。（2020年12月認定） |
| | バグの修正 |
| SAP | SAP S/4HANA 2020をサポートしました。（2020年12月認定） |
| | バグの修正 |
| SAP MaxDB、DB2、DataKeeper、Route 53、Filesystem、SAP、GenericApplication、IP | バグの修正 |

# バグの修正

下記は、本バージョンで修正されたバグおよび拡張機能の一覧です。

| バグ | 説明 |
|---|---|
| PL-7133 | LifeKeeper がリモートシステムで初期化中に、SAP HANA Recovery Kit によるリモート呼び出しが予期せず失敗する可能性がある問題を修正しました。 |
| PL-10994 | 診断エージェントインスタンスがインスタンスディレクトリにインストールされている場合、リソースの作成中に特定の SAP インスタンスが選択肢として表示されない問題を修正しました。 |
| PL-12327 | ハングした exportfs コマンドが原因で、NFS リソースをインサービスにするプロセスがハングする可能性がある問題を修正しました。 |
| PL-13792 | DataKeeper または HANA リソースを持つ階層は、階層内に「AUTOMATIC」に設定されたリソースがあっても、自動的にスイッチバックしようとしません。 |
| PL-14290 | lkGUIapp のEdit -> Server -> Properties で NullPointerException で落ちてしまう問題を修正しました。 |
| PL-14355 | 同梱する OpenSSL を 3.0.7 に更新しました。これにより TLS 1.1 以前のプロトコルは使用されなくなりました。 |
| PL-14825 | lvmvg の restore で active with error code “5” で fail してしまう現象を修正しました。 |
| PL-14872 | SAP HANA Recovery Kit では、一貫性のないデータまたは古いデータを持つシステムでリソースを稼働させることはできません。 |
| PL-15331 | lifekeeper start 時に nfs-server service も起動してしまう問題を修正しました。 |
| PL-15862 | LifeKeeper は、ライセンスキーの有効期限が切れたために起動が失敗した場合にメッセージをログに記録します。 |

# Hotfixes と Add-on サポートパッケージ

なし

# 廃止された機能

| 製品 | 機能 |
|---|---|
| バージョン **9.7.0**で廃止された機能 | |
| SAP システムのセットアップ時に、SAP サービスと同じノードに NFS サーバー/クライアントを使用した新規または既存の構成 (「ループバック構成」とも呼ばれます) のサポートを終了。 | |
| バージョン **9.6.2**で廃止された機能 | |
| | なし |

| | |
|---|---|
| **バージョン 9.6.1 で廃止された機能** | |
| | なし |
| **バージョン 9.6.0 で廃止された機能** | |
| | クライアント端末からWeb ブラウザを使用した LifeKeeper GUI へのアクセスのサポートを終了。 |
| **バージョン 9.5.2 で廃止された機能** | |
| Core | Red Hat Enterprise Linux 6 のサポートを終了。 |
| | CentOS 6 のサポートを終了。 |
| | Oracle Linux 6 のサポートを終了。 |
| Software RAID | LifeKeeper Software RAID (md) Recovery Kit のサポートを終了。 |
| Oracle | Oracle virtual machine (OVM) 3.4.6 のサポートを終了。 |
| PostgreSQL | PostgreSQL 9.5 のサポートを終了。 |
| | EnterpriseDB Postgres Plus Advanced Server 9.5 のサポートを終了。 |
| VMware | VMware vSphere 5.5 , 6.0 のサポートを終了。 |
| **バージョン 9.5.1 で廃止された機能** | |
| Core | Chefを用いた環境構築のサポートを終了。 |
| **バージョン 9.5.0 で廃止された機能** | |
| LifeKeeper Core | syslog-ng を用いた system ログ管理のサポートを終了。rsyslog を使用してください。 |
| | SUSE Linux Enterprise Server (SLES) 11.0 ～ SP4 のサポートを終了。 |
| | IAM ロール移行ツール /opt/LifeKeeper/lkadm/bin/aws_iam_migration のサポートを終了。IAM ロールは以前のバージョンで移行を済ませてから使用してください。 |
| DataKeeper | DataKeeper for Linux（DK リソース）を利用して、ディスク識別に DEVNAME を用いる環境のサポートを終了。GPT パーティション（GUID Partition Table）環境を使用してください。 |
| Oracle | Oracle Database Enterprise Edition 11g R2 のサポートを終了。 |
| | Oracle Database Standard Edition 11g R2 のサポートを終了。 |
| | Oracle Database Standard Edition One 11g R2 のサポートを終了。 |
| MySQL | MariaDB 5.5, 10.0 はサポートのサポートを終了。 |
| PostgreSQL | PostgreSQL 9.4 のサポートを終了。 |
| | EnterpriseDB Postgres Plus Advanced Server 9.4 のサポートを終了。 |
| **バージョン 9.4.1 で廃止された機能** | |
| | なし |

| バージョン 9.4で廃止された機能 | |
|---|---|
| DataKeeper | マルチサイトクラスター機能の廃止。 |

# LifeKeeper コンポーネント

## LifeKeeper Core

LifeKeeper for Linux は、64ビットシステム（AMD64およびEM64Tシステム）用にバンドルされ、64ビットシステム上でのみ動作します。

### LifeKeeper Core のバンドルには以下が含まれます。

- LifeKeeper
- DataKeeper
- Application Recovery Kits

LifeKeeper Core パッケージには、以下のインストール可能なパッケージが含まれています。

| パッケージ | パッケージ名 | 説明 |
|---|---|---|
| LifeKeeper Core | steeleye-lk-9.7.0-7520.x86_64.rpm | メモリー、CPU 、OS 、SCSI ディスクサブシステム 、ファイルシステムなど、のシステムの中核となるコンポーネントの障害回復機能を提供します。 |
| DataKeeper Core | steeleye-lkDR-9.7.0-7520.x86_64.rpm | DataKeeperパッケージは、データレプリケーション（インテントログを備えた同期または非同期ミラー）を提供します。 |
| LifeKeeper GUI | steeleye-lkGUI-9.7.0-7520.x86_64.rpm | LifeKeeper および Datakeeper の管理および健全性監視用のグラフィカルユーザーインターフェースです。 |
| LifeKeeper IP Recovery Kit | steeleye-lkIP-9.7.0-7520.noarch.rpm | IP アドレスの障害回復機能を提供します。 |
| LifeKeeper Raw I/O Recovery Kit | steeleye-lkRAW-9.7.0-7520.noarch.rpm | raw I/O を使用するアプリケーションを保護します。 |
| Quick Service Protection | steeleye-lkQSP-9.7.0-7520.noarch.rpm | 様々なサービスに対して、簡易的な障害回復機能を提供します。 |
| LifeKeeper Man Page | steeleye-lkMAN-9.7.0-7520.noarch.rpm | LifeKeeper 製品のリファレンスマニュアルのページが含まれます。 |

## LifeKeeper のオプションリカバリーソフトウェア

次のオプションのソフトウェアは、リストされているアプリケーションバージョンのリソース定義およびリカバリーソフトウェアを提供します。各リカバリーキットの要件は サポートマトリックス と リカバリーキット管理ガイド を参照してください。

| パッケージ | パッケージ名 | 説明 |
|---|---|---|
| LifeKeeper Apache Web Server Recovery Kit | steeleye-lkAPA-9.7.0-7520.noarch.rpm | Apache Web Server ソフトウェアの障害回復機能を提供します。 |
| LifeKeeper SAP Recovery Kit | steeleye-lkSAP-9.7.0-7520.noarch.rpm | SAP NetWeaver の障害回復機能を提供します。また、他のいくつかの LifeKeeper Recovery Kit と連携して包括的にフェイルオーバーを行います。 |
| LifeKeeper SAP HANA Recovery Kit | steeleye-lkSAPHANA-9.7.0-7520.noarch.rpm | SAP HANA データベースの障害回復機能を提供します。 |
| LifeKeeper SAP MaxDB Recovery Kit | steeleye-lkSAPDB-9.7.0-7520.noarch.rpm | SAP MaxDB データベースの障害回復機能を提供します。 |
| LifeKeeper DB2 Recovery Kit | steeleye-lkDB2-9.7.0-7520.noarch.rpm | DB2 データベースインスタンスの障害回復機能を提供します。 |
| LifeKeeper Oracle Recovery Kit | steeleye-lkORA-9.7.0-7520.noarch.rpm | Oracle ソフトウェアの障害回復機能を提供します。 |
| LifeKeeper MySQL Recovery Kit | steeleye-lkSQL-9.7.0-7520.noarch.rpm | MySQL の障害回復機能を提供します。 |
| LifeKeeper PostgreSQL Recovery Kit | steeleye-lkPGSQL-9.7.0-7520.noarch.rpm | PostgreSQL インスタンスの障害回復機能を提供します。 |
| LifeKeeper Sybase ASE Recovery Kit | steeleye-lkSYBASE-9.7.0-7520.noarch.rpm | Sybase ASE コンポーネントのAdaptive Server、Monitor Server、Backup Server の障害回復機能を提供します。 |
| LifeKeeper Postfix Recovery Kit | steeleye-lkPOSTFIX-9.7.0-7520.noarch.rpm | Postfixの障害回復機能を提供します。 |
| LifeKeeper Samba Recovery Kit | steeleye-lkSMB-9.7.0-7520.noarch.rpm | 異種ネットワーク内の Linux サーバー上の Samba によるファイルと印刷の共有に対して障害回復機能を提供します。 |
| LifeKeeper NFS Server Recovery | steeleye-lkNFS-9.7.0-7520.noarch.rpm | Network File System(NFS) ソフトウェアの障害回復機能を提供します。 |

| | | |
|---|---|---|
| Kit | | |
| LifeKeeper Network Attached Storage Recovery Kit | steeleye-lkNAS-9.7.0-7520.noarch.rpm | Network Attached Storage（NAS）によりエクスポートされた領域を使用するアプリケーションを保護します。 |
| LifeKeeper Logical Volume Manager (LVM) Recovery Kit | steeleye-lkLVM-9.7.0-7520.noarch.rpm | 他の LifeKeeper Recovery Kit に対して論理ボリュームのサポートを提供します。LifeKeeper で保護されたアプリケーションは、ストレージ管理の簡素化、要件変更に応じた動的なボリュームのサイズ変更など、Logical Volume Manager が提供するメリットを活用できます。 |
| LifeKeeper PowerPath Recovery Kit | steeleye-lkPPATH-9.7.0-7520.noarch.rpm | EMC PowerPath マルチパス I/O デバイスを使用するアプリケーションおよびファイルシステムを保護します。 |
| LifeKeeper Device Mapper Multipath (DMMP) Recovery Kit | steeleye-lkDMMP-9.7.0-7520.noarch.rpm | DMMP デバイスを使用するアプリケーションおよびファイルシステムを保護します。 |
| Hitachi Dynamic Link Manager Software (HDLM) Recovery Kit | steeleye-lkHDLM-9.7.0-7520.noarch.rpm | Hitachi Dynamic Link Manager Software デバイスを使用するアプリケーションおよびファイルシステムを保護します。 |
| LifeKeeper NEC iStorage StoragePathSavior (NECLifeKeeper) Recovery Kit | steeleye-lkSPS-9.7.0-7520.noarch.rpm | NEC iStorage StoragePathSavior v3.3 以降のマルチパス I/O デバイスを使用するアプリケーションおよびファイルシステムを保護します。 |
| SIOS DataKeeper | steeleye-lkDR-9.7.0-7520.x86_64.rpm | LifeKeeper 環境に統合データミラーリング機能を提供します。物理的な共有ストレージを持たない環境でも、 LifeKeeperリソースの運用が可能になります。 |
| LifeKeeper WebSphere MQ Recovery Kit | steeleye-lkMQS-9.7.0-7520.noarch.rpm | WebSphere MQ キューマネージャーおよびキューマネージャーストレージサイトの障害回復機能を提供します。 |
| Quorum/Witness Package | steeleye-lkQWK-9.7.0-7520.noarch.rpm | SCSI リザベーションによる排他制御を利用できない 共有ストレージ（DataKeeperやNASなど）をもつ環境で、スプリットブレイン状態の発生を抑制してより安全なクラスターを構築することができます。 |
| Quick Service Protection | steeleye-lkQSP-9.7.0-7520.noarch.rpm | LifeKeeper Quick Service Protectionは、OSのサービスを簡単に保護するための機能を提供します。 |

| | | |
|---|---|---|
| Recovery Kit for EC2 ™; | steeleye-lkECC-9.7.0-7520.noarch.rpm | AWS環境で、LifeKeeperでクラスター化されたサービスに対して、クライアントから切り替え前後で同一のIPアドレスを使ってアクセスを行うことができます。 |
| Recovery Kit for Route 53 ™ | steeleye-lkROUTE53-9.7.0-7520.noarch.rpm | AWS Route 53 上のDNSレコードを更新し、接続先の切り替えを行う機能を提供します。AWS環境で、クライアントがLifeKeeperでクラスター化されたサービスに対して、Direct Connect 接続のクライアントまたは異なるAmazon VPC上のクライアントからアクセスを行うことができます。 |
| VMDK as Shared Storage Recovery Kit | ssteeleye-lkVMDK-9.7.0-7520.noarch.rpm | 共有ディスクとして使用している VMware の仮想ハードディスクおよびそのファイルシステムをLifeKeeperリソースとして保護することが可能になります。 |
| Recovery Kit for Oracle Cloud Infrastructure | steeleye-lkOCIVIP-9.7.0-7520.noarch.rpm | Oracle Cloud 環境で、セカンダリー IP アドレスの割り当て状態を監視することができます。 |
| LifeKeeper SCSI3 Recovery Kit | steeleye-lkSCSI3-9.7.0-7520.noarch.rpm | LifeKeeper SCSI3 リカバリーキットは、SCSI-3 永続的な予約で標準の Linux SCSI デバイスインターフェイス（/dev/sdデバイスノード）を使用するアプリケーションとファイルシステムを保護します。 |
| LifeKeeper LB Health Check Kit | steeleye-lkLBHC-9.7.0-7520.noarch.rpm | LifeKeeper LB (ロード バランサー) ヘルスチェックキットは、ジェネリックロードバランサー (GenLB) リカバリーキットに代わるものです。GenLB リカバリーキットは、Generic アプリケーション (gen/app) リソースで使用するために 9.6.1 用にダウンロードできるアクションスクリプトのコレクションです。 GenLB キットは、使いやすいように完全な ARK に移行されました。 |

## システム要件

LifeKeeper for Linux は、 Linux 構成テーブル に含まれる最小要件を満たす、すべての Linux プラットフォームでサポートされています。 サポートされているオペレーティング システム、アプリケーション、および仮想化環境については、 LifeKeeper サポートマトリックス も参照してください。

注記: Linux サーバー上の LifeKeeper は、LifeKeeper for Windows と相互運用できません。

| 説明 | 要件 |
| --- | --- |
| Linux オペレーティングシステム | クラスターを構成する全てのサーバーは、同一のオペレーティングシステムを利用する必要があります。個々のオペレーティングシステム情報については、「Linux の設定」を参照してください。 |
| 仮想化環境 | 仮想マシン内で起動するゲスト OS は「Linux の設定」に記載のバージョンを利用する必要があります。<br>サポート対象の仮想化環境の詳細については、サポートマトリックス を参照ください。<br>LifeKeeper for Linux が利用可能なクラウド環境については こちら を参照してください。<br>以下の仮想化環境は LifeKeeperuite for Linux がサポートする仮想化環境の一例です。<br><br>• KVM<br>• Oracle VM Server for x86<br>• VMware vSphere v6.5、v6.7、v7.0、v8.0<br>• Amazon EC2<br>• Microsoft Azure<br>• Nutanix Acropolis Hypervisor<br>• Google Cloud<br>• Oracle Cloud Infrastructure (**注意 2**)<br><br>VMware vSphere v6.5 と v6.7 では vSAN の利用をサポートします。ただし RDM は、vSAN で非サポートとなるため、利用できません。<br>VMware vSphere v7.0 以降では vSAN の利用は VMware Cloud on AWS 上でのみサポートされます。<br><br>KVM および Oracle VM Server for x86 仮想化環境上の LifeKeeper for Linux では、ファイバーチャネルSANおよびSCSIディスクを使用するクラスター構成をサポートしません。<br><br>**注意 1：** AWSやAzure環境で動作するSLESにおいて、クラウドネットワークプラグインによって仮想 IP アドレスの動的な変更が行われることにより、LifeKeeper クラスターの動作に影響を与えることがあります。詳細や回避策は トラブルシューティング > 既知の問題と制限 >「LifeKeeper Core」をご確認ください。<br><br>**注意 2：** Oracle Cloud Infrastructure 上では LifeKeeper がサポートする OS と ARK について一部制限があります。詳しくは こちら をご確認ください。 |
| メモリー | システムの最小メモリー要件は、OS が必要とする最小要件を満たしてください。システムのメモリーは LifeKeeper が保護するシステム上で動作するアプリケーションに対してサイジングする必要があります。 |
| ディスク容量 | LifeKeeper for Linux の導入に必要なディスク容量は次のとおりです。<br>• /opt – 約 100MB (インストールするキットに依存します)<br>• / – 約 110MB |
| Java Runtime Environment | • OpenJDK 1.8, OpenJDK 1.10, OpenJDK 1.11, OpenJDK 1.12 |

## LifeKeeper のオプションリカバリーソフトウェアの要件

LifeKeeper が保護するアプリケーションに適用される追加の要件や制限については、アプリケーション設定 を参照してください。

| 製品 | 要件 |
|---|---|
| LifeKeeper Apache Web Server Recovery Kit | Apache Web Server v2.4 |
| SAP Recovery Kit | • SAP NetWeaver 7.0 、 Enhancement Package 1,2 および 3 を含む<br>• SAP NetWeaver 7.3 、 Enhancement Package 1 を含む<br>• SAP NetWeaver 7.4<br>• SAP NetWeaver 7.5<br>• SAP NetWeaver AS for ABAP 7.51 innovation package |
| LifeKeeper SAP MaxDB Recovery Kit | • SAP MaxDB v7.9<br>• LifeKeeper for Linux v6 以降 |
| LifeKeeper Postfix Recovery Kit | • Postfix ソフトウェアは、LifeKeeper for Linuxがサポートするディストリビューションに付属しているものを使用し、（クラスターを構成する）各サーバー上に構成します。Postfixのバージョンは各サーバーで同一である必要があります。<br>• LifeKeeper for Linux v6 以降 |
| LifeKeeper Oracle Recovery Kit | • Oracle Database Enterprise Edition v12c 、 v12c R2 、 v18c、 v19c (ASM 機能除く)、v21c(ASM 機能除く)<br>• Oracle Database Standard Edition 2 (SE2) v12c 、 v12c R2 、 v18c、 v19c (ASM 機能除く)、v21c (ASM 機能除く) |
| LifeKeeper DB2 Recovery Kit | • IBM Db2 Universal Database v10.5、v11.1<br>• IBM Db2 Enterprise Server Edition (ESE) v10.5 、 v11.1、v11.5<br>• IBM Db2 Workgroup Server Edition (WSE) v10.5 、 v11.1、v11.5<br>• IBM Db2 Express Edition v10.5 、 v11.1、v11.5<br><br>• LifeKeeper for Linux v6 以降<br>• LifeKeeper NFS Server Recovery Kit v5.1 以降 (マルチパーティションを持つ DB2 EEE 用および DB2 ESE 用のみ) |
| LifeKeeper MySQL Recovery Kit | • MySQL および MySQL Enterprise v5.7 、 v8.0<br>• MariaDB v10.3、v10.4、v10.5 |
| LifeKeeper PostgreSQL Recovery Kit | • PostgreSQL v9.6、v10、v11、v12、v13、v14、v15<br>• EnterpriseDB Postgres Plus Advanced Server/EDB Postgres Advanced Server v9.6、v10.0、v11.0、v12.0、v13.0、v14.0<br>• PowerGres Plus (Linux 版) V10、V13<br>• PowerGres on Linux V11、V13 |

| | |
|---|---|
| | • FUJITSU Software Symfoware Server (Openインタフェース)<br>◦ Enterprise Edition V12.2<br>◦ Standard Edition V12.2<br>• FUJITSU Software Symfoware Server (Postgres)<br>◦ Enterprise Edition V12.3, V12.4<br>◦ Standard Edition V12.3, V12.4<br>◦ Lite Edition V12.3<br>• FUJITSU Software Enterprise Postgres<br>◦ Advanced Enterprise Server Edition 14SP1<br>◦ Advanced Edition 10, 11, 12, 13, 14<br>◦ Standard Edition 9.6, 10, 11, 12, 13, 14, 14SP1<br>◦ Community Edition 10, 11, 12, 13, 14, 14SP1 |
| LifeKeeper Sybase ASE Recovery Kit | Sybase ASE 15.7、16.0 |
| LifeKeeper Samba Recovery Kit | サポート対象の Linux ディストリビューションに付属の標準 samba ファイルサービス |
| LifeKeeper NFS Server Recovery Kit | • Linux kernel version 2.6 以降<br>• SLES を利用する場合、NFS サーバーパッケージおよびNFSクライアントパッケージを追加でインストールする必要があります。<br><br>注意：<br>Red Hat Enterprise Linux 7以降、CentOS 7以降、Oracle Linux 7以降では NFSv2 はサポートされません。<br>Red Hat Enterprise Linux 8以降では NFS over UDP はサポートされません。<br>一部の環境では追加の設定を必要とする場合があります。詳細は「特定の設定上の考慮事項」を参照してください。 |
| LifeKeeper Network Attached Storage Recovery Kit | NFS バージョン v2、v3、v4 のいずれかでのマウントをサポートする NFS サーバーまたは NAS デバイス<br><br>注意：<br>Red Hat Enterprise Linux 7以降、CentOS 7以降、Oracle Linux 7以降では NFSv2 はサポートされません。<br>Red Hat Enterprise Linux 8以降では NFS over UDP はサポートされません。<br>一部の環境では追加の設定を必要とする場合があります。詳細は「特定の設定上の考慮事項」を参照してください。 |
| LifeKeeper Logical Volume Manager (LVM) Recovery Kit | Linux Logical Volume Manager (LVM) バージョン 1 または 2 のボリュームグループおよび論理ボリューム |
| EMC PowerPath | • PowerPath for Linux v5.3 またはそれ以降<br>• sg3_utils パッケージがインストールされている必要があります。 |

| | |
|---|---|
| Device Mapper Multipath (DMMP) | • オペレーティングシステム付属の device-mapper-multipath パッケージがインストールされている必要があります<br>• sg3_utils パッケージがインストールされている必要があります |
| Hitachi Dynamic Link Manager Software (HDLM) | • 日立 HDLM のマルチパス I/O 設定を参照してください。<br>• sg3_utils パッケージがインストールされている必要があります |
| NEC iStorage Storage Path Savior (NECLifeKeeper) | • iStorage StoragePathSavior for Linux v3.3 以降<br>• 対応するLinux カーネルおよびディストリビューションは StoragePathSavior for Linux のサポート情報を参照してください<br>• Red Hat および SLES では sg3_utils パッケージがインストールされている必要があります<br>• LifeKeeper for Linux v6 以降 |
| WebSphere MQ リソース | • IBM MQ v8.0, v9.0, v9.1, v9.2<br>• 既知の問題と制限 > インストール を参照してください。 |
| Quorum/Witness Package | quorum/witness モードのクラスターに参加するすべてのノード (witness 専用のノードを含む) には 、 Quorum/Witness Server Support Package for LifeKeeper をインストールする必要があります。 |
| LifeKeeper SCSI-3 Recovery Kit | sg3_utilsパッケージがインストールされている必要があります。 |

## オープンソースパッケージ

以下のオープンソースパッケージは、LifeKeeper インストールイメージに含まれています。

| 名前 | バージョン | ライセンスのタイプとバージョン |
|---|---|---|
| curl | 7.86.0 | curl |
| libxml2 | 2.7.8 | MIT |
| lighttpd | 1.4.68 | BSD-3-Clause |
| openssl | 3.0.7 | Apache-2.0 |
| openssl | 1.0.1e | OpenSSL |
| pcre | 4.5 | PCRE |
| pdksh | 5.2.14 | Public Domain and GPL |
| perl | 5.8.8 | GPL-1.0-or-later or Artistic-1.0-Perl |
| Perl Addon Packages | – | Various |
| powercli | 11.5.0 | various license |
| powershell | 6.2.3 | MIT |

| | | |
|---|---|---|
| readline | 4.3 | GPL-2.0-or-later |
| runit | 2.0.0 | BSD-3-Clause |
| util-linux | 2.31.1 | GPLv2 and GPLv2+ and LGPLv2+ and BSD with advertising and Public Domain |
| openjdk | 12.0.2 | GPL-2.0-or-later |
| kconfig-frontends | 4.11.0 | GPL-2.0-only |
| ping | – | BSD-4-Clause |
| zlib-static | 1.2.3 | Zlib |
| Perl Config::IniFiles (CPAN module) | 2.27 | GPL/Artistic (Same as Perl) |
| mdadm | 3.2.6 | GPL-2.0-or-later |
| balance | 3.54 | GPL-2.0-only |
| nbd-client | 1.0 | GPL |
| nbd-server | 1.3 | GPL |
| nbd (kernel module) | 2.6.32<br>3.10.0<br>4.18.0 | GPL-2.0-only |
| raid1 (kernel module) | 2.6.32<br>4.12.14<br>4.14.35<br>4.18.0<br>5.3.18 | GPL-2.0-only |

# インストールと構成

インストールおよび構成の詳しい情報については 、 LifeKeeper for Linux インストレーションガイド を参照してください。

# アップグレード

LifeKeeper for Linux v9.7.0 へは、v9.5.x および v9.6.x からアップグレードすることができます。これらより前のバージョンからアップグレードする場合、それまでのバージョンをアンインストールし、LifeKeeper for Linux を再インストールする必要があります。それまでのバージョンをアンインストールしないで v9.7.0 へアップグレードしたい場合には、一度 v9.5.x または v9.6.x にアップグレードした後、さらに v9.7.0 にアップグレードする方法をとることもできます。

## ストレージとアダプターのオプション

共有ストレージ構成において LifeKeeper for Linux がサポートするストレージモデルやアダプターについては、「ストレージとアダプターのオプション」を参照してください。また、ドライバーのバージョンやその他の設定要件の詳細については、「ストレージとアダプターの設定」を参照してください。

## テクニカルノート

LifeKeeper 環境に関連した構成および動作上の問題に関して、テクニカルノート の参照を強く推奨します。

# 既知の問題

LifeKeeper for Linux テクニカルドキュメンテーション の 既知の問題と制限 と DataKeeper トラブルシューティング の トラブルシューティング を参照してください。

商標について：
「Amazon Web Services」、「Powered by Amazon Web Services」のロゴ、「AWS」、「Amazon EC2」、「EC2」、「Amazon Elastic Compute Cloud」、「Amazon Route 53」、「Amazon Virtual Private Cloud」および「Amazon VPC」は、米国その他の国における Amazon.com, Inc. またはその関連会社の商標です。
「Google Cloud」、「BigQuery」および「Google Compute Engine」は Google LLC の商標です。
「Oracle」および「Java」は、オラクルおよびその関連会社の登録商標です。その他の社名、商品名等は各社の商標または登録商標である場合があります。
なお、本文および図表中では、「 ™ 」、「®」は一部を除き明記しておりません。

# 重要なお知らせ: SIOS DataKeeper for Linux とカーネルの問題

**重要なお知らせ:** 本事象に関するパッチは現在入手可能です。対象となるお客様は速やかにパッチを適用してください。

以下のカーネルおよびバージョンにおいて影響を受ける事象は、全ての開発、品質保証作業を終えたことをご案内いたします。

| OS | 影響を受けるカーネルの範囲 | 左記OSをサポートしている SIOS Protection Suite for Linux(LifeKeeper/DtaKeeper) のバージョン |
|---|---|---|
| RHEL/CentOS/ OEL 8.2 | 全て | v9.5.1 |
| RHEL 8.3 | 全て | v9.5.1 |
| SLES 12 SP4 | 4.12.14-95.51 以上 | v9.3.2 – v9.5.1 |
| SLES 12 SP5 | 4.12.14-122.20 ~ 4.12.14-122.74.0 | v9.5.0 – v9.5.1 |
| SLES 15 SP1 | 4.12.14-197.37.1 以上 | v9.4.1 – v9.5.1 |
| SLES 15 SP2 | 5.3.18-22.2 ~ 5.3.18-24.67.1 | v9.5.1 |
| OEL 7.x UEK 5 | 4.14.35-2025.400.8 ~ 4.14.35-2047.504.1 | v9.4.0 – v9.5.1 |

md/raid1カーネルモジュールで発見された問題を解決するパッチおよび適用手順については弊社 LifeKeeper / DataKeeper ユーザーポータルの下記 URL をご参照の上、ダウンロードしてください。

LifeKeeper + DataKeeper構成のシステムにおける データ破損問題の対策についてのご案内

パッチ適用後は、カーネルをパッチが適用されたraid1 kernel moduleを含まない別のバージョンにアップデートすることをお勧めしません。カーネルの自動アップデートを制限する方法については、ご使用のOSのマニュアルを参照してください。カーネルアップデートする際の注意点に関しては、以下のページも併せて参照ください。

カーネルアップデートする際の注意点

最終的な解決策については、引き続き情報を提供してまいります。

パッチ適用後 (セキュリティ問題やレギュレーションの理由で) カーネルのアップグレードが必要な場合は、パッチ申請時にダウンロード可能な readme ファイルに記載されている「パッチ適用後の計画されたカーネルのアップグレード」のセクションを参照してください。

ご不明な点がございましたら、SIOSテクニカルサポートまたはSIOS営業担当者までお問い合わせください。

# 3. LifeKeeper for Linux スタートアップガイド

LifeKeeper for Linux のスタートアップガイドにつきましては、LifeKeeper/DataKeeper User Portal （参照にはログインが必要です）をご参照ください。

# 4. LifeKeeper for Linux インストレーションガイド

「LifeKeeperインストレーションガイド」には、LifeKeeper 環境をプランニングおよびインストールする方法が記載されています。サーバ、ストレージデバイス、ネットワークコンポーネントをセットアップするために必要な手順の他、LifeKeeper のグラフィカルユーザインターフェース（GUI）の詳しい設定についても示されます。

このガイドの手順をすべて終えれば、LifeKeeper および DataKeeper リソースを設定できる状態になります。LifeKeeper を設定するために必要な情報は、LifeKeeper for Linux テクニカルドキュメンテーションで提供されています。

## システム要件

ハードウェアおよびソフトウェアの要件およびバージョンに関する総合的なリストについては、LifeKeeper for Linux リリースノート を参照してください。

また、LifeKeeper をインストールする前に、本書に記載されているプランニング作業およびハードウェア構成作業を完了していることを確認してください。

## テクニカルノート

本ソフトウェアに関する問題のトラブルシューティングや制限事項などの詳細については、LifeKeeper for Linux テクニカルドキュメンテーションのテクニカルノート および トラブルシューティング セクションを参照してください。

# 4.1. LifeKeeper ソフトウェアのパッケージ

LifeKeeper for Linux ソフトウェア (Optional LifeKeeper Recovery Kit を含む) は、単一のイメージファイル (sps.img) に格納されています。

## LifeKeeper for Linux のインストールイメージファイル

LifeKeeper for Linux のイメージファイル (sps.img) では、システムにLifeKeeperをインストールする際に必要なユーザー対話型システムセットアップタスクを実行するように設計された、一連のインストールスクリプトを提供しています（対話モード を参照してください）。 非ユーザー対話型インストールを実行することもできます。非ユーザー対話型インストールについての詳細は、非対話モード を参照してください。
LifeKeeperのインストールプロセスは、次の3つのステップに分けられます。

- 情報収集
- 選択（ユーザー対話型のみ）
- インストールと設定

このプロセスの最初のステップは情報収集で、使用中のLinuxディストリビューションなどのシステムに関する情報を収集して、システムがインストールに必要な要件を満たしていることを確認します。2番目のステップは選択で、メニューベースの選択プロセスを通じてユーザーと対話し、インストールするLifeKeeperパッケージとそのサポートに必要な構成を決定します。最後のステップは実際のインストールと設定で、LifeKeeper Coreパッケージクラスタとオプションのリカバリーソフトウェアのインストール、およびLifeKeeper用のシステムの設定を実施します。このステップでは、システムにまだインストールされていない、必要なOSサポートパッケージもインストールされます。

LifeKeeper for Linux のイメージファイルには、次のソフトウェアパッケージを含む Core パッケージクラスタが格納されています。

## LifeKeeper for Linux Core パッケージクラスタ

- LifeKeeper (**steeleye-lk**) LifeKeeper Core パッケージには、メモリ、CPU、OS、SCSI ディスクサブシステム、ファイルシステムなどの中核システムコンポーネント用のリカバリソフトウェアがあります。
- LifeKeeper GUI (**steeleye-lkGUI**) LifeKeeper GUI パッケージは、LifeKeeper の管理および監視用のグラフィカルユーザインターフェースです。
- DataKeeper (**steeleye-lkDR**)。DataKeeper パッケージは、インテントログ記録を使用するデータ複製 (同期ミラーまたは非同期ミラー) を実行します。
- IP Recovery Kit (**steeleye-lkIP**) LifeKeeper IP Recovery Kit は、IP アドレスの自動リカバリ用のスイッチオーバソフトウェアです。

- Raw I/O Recovery Kit (**steeleye-lkRAW**) LifeKeeper Raw I/O Recovery Kit は、Raw I/O を使用してカーネルのバッファリングを迂回するアプリケーションをサポートします。

- マニュアルページ (**steeleye-lkMAN**) LifeKeeper マニュアルページパッケージは、LifeKeeper 製品のリファレンスマニュアルです。

## オプションのリカバリソフトウェア

また、Recovery Kit は、LifeKeeper Core ソフトウェアとともにリリースされています。インストール中、提供されている最新のリカバリキットを選択可能な全リストが表示されます。リカバリキットの詳細については、LifeKeeper テクニカルドキュメンテーションのアプリケーション Recovery Kit セクションを参照してください。

# 4.2. LifeKeeper 環境のプランニング

次のトピックは、LifeKeeper for Linux のクラスタ環境の定義に役立ちます。

---

サーバ構成のマッピング

ストレージとアダプタの要件

ストレージとアダプタのオプション

# 4.2.1. サーバー構成のマッピング

以下のガイドラインを使用して、サーバー構成を文書化してください。

1. 構成に応じたサーバー名、プロセッサーの種類、メモリー、その他の I/O デバイスを決定します。バックアップサーバーはプライマリーサーバーに障害が発生した場合に、プライマリーサーバーと同等に処理を実行する能力を有しているかを確認する必要があります。

2. 通信接続要件を決定してください。

   **重要**:クラスター化された構成には可能性として2 種類の通信要件 (クラスター要件とユーザー要件) があります。

   - クラスター LifeKeeper クラスターでは、サーバー間に少なくとも 2 つのコミュニケーションパス (「ハートビート」とも呼ばれます) が必要になります。この冗長性により、通信障害が原因で発生する「スプリットブレイン」シナリオを回避することができます。独立した 2 つのサブネットを使用する 2 つの分離した LAN ベースの (TCP) コミュニケーションパスが推奨され、これらの 1 つ以上をプライベートネットワークとして構成する必要があります。TCP と TTY のの組み合わせもサポートされています。TTY コミュニケーションパスは、サーバーのシリアルポート間でRS-232ヌルモデム通信を使用します。

     コミュニケーションパスを 1 つしか使用しない場合、互いに通信する LifeKeeper クラスタ内のシステムの機能に支障をきたす可能性があります。単一のコミュニケーションパスを使用しているときに、そのコミュニケーションパスで障害が発生した場合、複数のシステム上で同時に LifeKeeper の階層が使用可能になることがあります。これは、偽のフェイルオーバまたは「スプリットブレイン」シナリオと呼ばれます。「スプリットブレイン」シナリオでは、各サーバが、アプリケーションを制御できると認識しているため、データにアクセスしようとしたり共有ストレージデバイスにデータを書き込もうとする場合があります。スプリットブレインシナリオを解決するために、LifeKeeper では、サーバの電源をオフにしたり、再起動したり、階層を使用できなくすることで、すべての共有データに対するデータの整合性を保証することができます。また、TCP コミュニケーションパス上のネットワークトラフィックが大きくなると、偽のフェイルオーバや LifeKeeper が適切に初期化できなくなるなど、予期せぬ動作が生じる可能性があります。

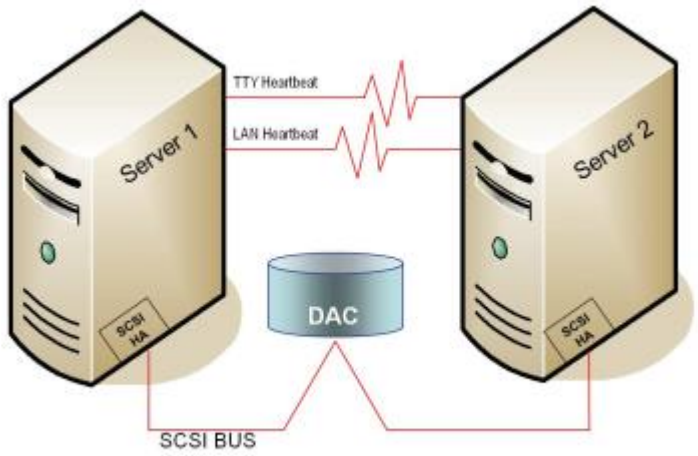
   - ユーザー ユーザートラフィックに対する代替の LAN 接続、つまり、クラスターハートビートに使用するものとは別の LAN 接続を用意することをお勧めします。ただし、(推奨通りに) 2 つの TCP コミュニケーションパスを構成した場合、これらのコミュニケーションパスのいずれかが、サーバーに出入りするその他のトラフィックとネットワークアドレスを共有することができます。
   - 注記: リソースが必要なときのみに in service になるようにするには、Quorum/Witness Server Support Package for LifeKeeper を使用することができます。

3. 共有リソースアクセス要件を確認して理解してください。データーレプリケーション（ミラーリング）環境でLifeKeeperを使用する場合は、 SIOS DataKeeper管理ガイド を参照してください。 ネットワーク接続ストレージ（NAS）環境で LifeKeeper を使用する場合は、NAS Recovery Kit 管理ガイド を参照してください。共有ストレージを使用するクラスタは、共有 SCSI バスまたはファイバチャネルループを利用できます。 LifeKeeper ではリソースが 1 つのサーバにロックされるため、ロックされたすべてのリソースへのアクセスが必要になるサーバーは常時 1 つだけであることを確認する必要があります。LifeKeeper デバイスのロックは、論理ユニット (LUN) レベルで行われま

す。アクティブ/アクティブ構成では、各階層が独自の一意の LUN にアクセスする必要があります。共通の LUN にアクセスするすべての階層は、同じサーバー上でアクティブ (稼働中) である必要があります。

4. 共有メモリー要件を決定してください。共有メモリーおよびセマフォパラメーターを設定する場合は、LifeKeeper だけでなくサードパーティ製アプリケーションの共有メモリー要件も考慮に入れてください。LifeKeeper の共有メモリー要件については、 テクニカルノート のチューニング を参照してください。

## LifeKeeper ペアに対する構成マップの例




この構成マップの例は、ディスクアレイサブシステムを共有する LifeKeeper サーバーのペアを図示しています。通常は、Server 1 がアプリケーションを実行し、Server 2 がバックアップサーバーまたはセカンダリサーバになります。このケースでは、同時に 1 つのサーバがディスクアレイのディスクストレージスペース全体を保有しているので、ディスクリソースの競合はありません。ディスクアレイコントローラは「DAC」、SCSI ホストアダプタ (パラレル SCSI、ファイバチャネルなど) は「SCSI HA」と表記されています。

サーバのペアが、最も単純な LifeKeeper 構成となります。3 つ以上のサーバーで構成されるクラスターを計画する場合、複数のサーバー間が適切に接続されるようにマッピングすることが非常に重要になります。たとえば、多方向フェイルオーバー構成では、物理的な接続が存在しない場合でも LifeKeeper 内のコミュニケーションパスを定義することが可能です。カスケーディングフェイルオーバー機能を実現するために、各サーバーがクラスター内の他のすべてのサーバーへの物理的な接続パスを持つ必要があります。

# 4.2.2. ストレージとアダプターの要件

以下のガイドラインを使用して、ストレージとホストアダプタの要件を決定してください。

**ストレージデバイス** - アプリケーションのデータストレージ要件に基づいて、構成に必要なデータストレージデバイスの種類と数を決定する必要があります。共有ファイルは、ディスクアレイサブシステム (RAID : Redundant Array of Inexpensive Disks) 上に置く必要があります。LifeKeeper は、構成に使用できるハードウェア RAID 周辺装置を多数サポートしています。サポートされている周辺装置のリストについては、ストレージとアダプタのオプションを参照してください。

ストレージデバイスの構成を計画する際には、以下の点を考慮してください。

- LifeKeeper では物理ディスクまたは論理ユニット (LUN) レベルでリソースを管理し、その構成内では同時に 1 つのサーバのみが各物理ディスクまたは LUN 上のリソースを利用できます。そのため、LifeKeeper の構成を始める前に、ディスク割り当ての計画を立てることをお勧めします。たとえば、アクティブ/アクティブ構成の各階層は、独自の一意の LUN にアクセスする必要があるので、2 ノードアクティブ/アクティブ構成の場合は最低 2 つの LUN が必要になります。

- 一部のモデル固有の問題およびハードウェア設定の詳細は、サポートストレージ一覧 で管理されています。

**アダプタ** - 構成の種類および周辺装置の数に基づいて、必要な SCSI またはファイバチャネルホストアダプタの種類と数を決定してください。 選択するアダプタは、ドライバが使用できるように、LifeKeeper だけでなく使用している Linux ディストリビューションでもサポートされていることが重要です。サポートされているホストアダプタのリストについては、サポートされているアダプタモデルを参照してください。 参照用に、構成マップにホストアダプタを追加する必要があります。

# 4.2.3. ストレージとアダプターのオプション

共有ストレージ構成で LifeKeeper が現在サポートしているディスクアレイのストレージモデルの一覧については、サポートストレージ一覧 を参照してください。ドライバのバージョンと、これらのアレイとアダプタに関するその他の設定要件については、ストレージとアダプタの設定 を参照してください。

IP フェイルオーバのみを使用する非共有ストレージを含む LifeKeeper 構成あるいは SIOS Data Replication または Network Attached Storage の使用時には、サポートされているディスクアレイおよびアダプタは必要ありません。

SIOS Technology Corp. では、ファイバチャネルのハブとスイッチを特に認定していません。これは、これらのデバイスに対する LifeKeeper 固有の既知の制限事項や要件がないためです。ストレージとアダプタの設定 の指定されたアレイで特に記載のない限り、LifeKeeper は、ディスクアレイベンダがサポートするハブとスイッチを推奨します。

# 4.3. LifeKeeper 環境のセットアップ

これまでのセクションでは、要件を決定し、LifeKeeper 設定をマッピングしました。これで、LifeKeeper 環境のコンポーネントをセットアップできるようになりました。

注記: 一部のセットアップ作業は異なる順序で実行することが可能ですが、このリストは推奨された順序で示されています。

---

Linux OS および関連する通信パッケージのインストール

Linuxの依存関係

サーバと共有ストレージの接続

共有ストレージの設定

ネットワーク設定の確認

切り替え可能な IP アドレスの作成

データベースアプリケーションのインストールとセットアップ

GUI ユーザーの設定

# 4.3.1. Linux OS および関連する通信パッケージのインストール

LifeKeeper for Linux をインストールする前に Linux オペレーティングシステムが正常にインストールされ動作可能であることを確認する必要があります。完全なインストールの詳細については、Linux のディストリビューションに付属の Linux インストール手順書を参照してください。

注記:

- 必須パッケージに必要となる場合がある依存関係については、Linux の依存関係 トピックを参照してください。
- 共有ストレージを接続して設定した後に Linux をインストールすることも可能ですが、 新しい周辺装置を導入する前に、Linux をインストールして実行する方が簡単な場合があります。
- LifeKeeper for Linux のインストールイメージファイルは、LifeKeeper をシステムにインストールするときに必要なユーザーインタラクティブなシステムセットアップ作業とインストール作業を実行するよう設計されたインストールスクリプト一式を提供します。

# 4.3.2. Linux の依存関係

LifeKeeper for Linux を正常にインストールするには、いくつかの前提条件パッケージをインストールする必要があります。スクリプトが失敗しないように、これらのパッケージはインストールセットアップスクリプトを実行する前にインストールしてください。

前提条件パッケージは以下の 3 つのグループに分類されます。

- 一般的なパッケージの依存関係
- オプションのリカバリーキットパッケージの依存関係

オペレーティングシステムのバージョン と、選択したオペレーティングシステムの種類 (最小、デフォルトなど) に基づいてインストールされるパッケージによっては、追加の従属パッケージが必要な場合があります。

依存関係は LifeKeeper がサポートする OS のバージョンが基本となります。詳細については LifeKeeper for Linux サポートマトリックス を参照してください。

オペレーティングシステムをアップグレードする前に、パッケージマネージャーツール (yum または zypper) を介して使用される **依存関係とパッケージのリストを文書化** し、LifeKeeper がインストールしたパッケージへの影響を記録して **依存関係に基づいて予想される削除を防止してください。**
**注意:** 以下にリストされている LifeKeeper 依存パッケージは完全ではない可能性があるため、OS のアップグレードを実行する前に、パッケージマネージャーツールのアクションを確認する必要があります。

**注意: yum** や **zypper** など、リポジトリーベースのパッケージマネージャーの使用を推奨します。これらのパッケージマネージャーは、定義済みのソフトウェアリポジトリーを検索して依存関係を自動的に解決するように設計されているので、これらの必須パッケージのインストールが容易になります。依存するパッケージのインストールを行いやすくするため、LifeKeeper for Linux のインストーラーは、 **yum** もしくは **zypper** を使用します。そのため、OS パッケージのリポジトリーをセットアップしておくことを強く推奨します。それにより LifeKeeper for Linux をインストールする前に下記に記載の OS に依存するパッケージをインストールする必要性がなくなります。

**rpm** のインストール例

```
rpm -ivh <package(s)>
```

**yum** のインストール例

```
yum install <package(s)>
```

**Zypper** のインストール例

```
zypper install <package(s)>
```

## yum/Zypper パッケージリスト

列挙されたディストリビューションごとに、以下のリストの rpm パッケージが対応するパッケージインストーラーによってインストールされます。これらは、LifeKeeper for Linux に必要なすべての依存関係を解決するパッケージの最小リストです。

**Red Hat Enterprise Linux、CentOS、Oracle Linux**

```
yum install libXtst libstdc++ bzip2-libs pam zlib patch util-linux
```

**SLES**

```
zypper install libstdc++ bzip2 pam pam-modules zlib util-linux
```

## 一般的なパッケージの依存関係

LifeKeeper for Linux を正常にインストールするには、以下のパッケージが必要です。インストールするパッケージのアーキテクチャーバージョンは、必ずオペレーティングシステムのアーキテクチャーと一致させてください (x86 または x86_64)。

**Red Hat Enterprise Linux、CentOS、Oracle Linux**

- bzip2
- iproute
- iputils
- patch (バージョン 2.5 以降)
- util-linux

**注記:** オペレーティングシステムのインストール時の選択によっては、これらのパッケージの一部またはすべてが既にインストールされている場合があります。

**SLES**

- bzip2
- iproute2
- iptables
- iputils
- insserv
- patch (バージョン 2.5 以降)
- libXtst6 (SLES15のみ)
- libXi6 (SLES15のみ)
- util-linux
- module-legacy (SLES 15 のみ)

**注記:** オペレーティングシステムのインストール時の選択によっては、これらのパッケージの一部またはす

べてが既にインストールされている場合があります。

## syslogデーモンとの依存関係

LifeKeeper のログは syslog デーモンを利用して記録されます。LifeKeeper では、rsyslog をサポートしています。LifeKeeper をインストールする前に、rsyslog デーモンをインストールして動作させておく必要があります。

**注記:** RHEL7 や SLES12 など systemd を採用しているディストリビューションでは、journald がログを一括して管理しています。LifeKeeper は syslog デーモンを利用してログ出力を行っていますので、syslog デーモンも動作させる必要があります。したがって、LifeKeeper を使用する場合は syslog デーモンを動作させるように設定してください。

**注記:** journald はデフォルトで /run/log/journal の tmpfs にログを記録します。したがって、OS シャットダウン時のシステムログなどが記録されません。journald のログが永続化する様に設定を変更してください。 journald のログを永続化するためには、/etc/systemd/journald.conf で「Storage=persistent」と設定するか、「Storage=auto」(デフォルト) の設定で /var/log/journal ディレクトリーを作成してください。設定変更後、systemd-journald.service を再起動してください。

## オプションのリカバリキットパッケージの依存関係

さらに、LifeKeeper for Linux のオプションの Application Recovery Kit (ARK) の一部はサポートパッケージをインストールする必要があります。

LifeKeeper for Linux NFS Application Recovery Kit を用いて NFS エクスポートを保護する場合は、以下の従属パッケージが必要です。

- nfs-utils (Red Hat Enterprise Linux、CentOS、Oracle Linux)
- nfs-client (SLES)
- nfs-kernel-server (SLES)

Device Mapper Multipath (DMMP)、Hitachi Dynamic Link Manager Software (HDLM)、Power Path、または NEC iStorage StoragePathSavior (NECSPS) を用いてマルチパスデバイスを保護する場合は、以下の従属パッケージが必要です。

- sg3_utils (すべてのマルチパスキット)
- sg3_utils-libs (すべてのマルチパスキット)
- HDLM (Hitachi Dynamic Link Manager Software Kit)
- EMCpower.LINUX (Power Path Kit)
- sps (NEC iStorage StoragePathSavior Kit **4.2.0** 以前)
- sps-utils および sps-driver (NEC iStorage StoragePathSavior Kit **4.2.1** 以降)

LifeKeeper for Linux Websphere MQ/MQSeries Application Recovery Kit を用いて Websphere MQSeries キューマネージャーを保護する場合は、以下の従属 Websphere MQ パッケージが必要です。

- MQSeriesServer
- MQSeriesSamples

- MQSeriesClient
- MQSeriesRuntime
- MQSeriesSDK

LifeKeeper for Linux Software RAID (md) Recovery Kit を用いて Software RAID デバイスを保護する場合は、以下の従属パッケージが必要です。

- mdadm

# 4.3.3. サーバと共有ストレージの接続

非共有ストレージ環境で LifeKeeper を使用することを計画している場合は、この情報をスキップできます。データレプリケーション (ミラーリング) 環境で LifeKeeper を使用している場合は、この文書の DataKeeper セクションを参照してください。ネットワーク接続ストレージ環境で LifeKeeper を使用している場合は、LifeKeeper Network Attached Storage Recovery Kit 管理ガイド を参照してください。

Linux がインストールされたら、ホストアダプタおよび共有周辺装置のアドレスを設定する必要があります。具体的な詳細については、アダプタおよびストレージデバイスに付属のドキュメンテーションを参照してください。

# 4.3.4. 共有ストレージの設定

LifeKeeper では、共有 SCSI (Small Computer System Interface) ホストアダプターおよび共有ディスクハードウェアの機能を使用して、障害が発生したサーバーから指定のバックアップサーバーにリソースを切り替えるように設定できます。ファイバーチャネルのストレージエリアネットワーク (SAN) も、障害の発生したサーバーから指定のバックアップサーバーにリソースを切り替えるのに使用できます。

LifeKeeper がフェイルオーバーによる保護を提供できるようにするディスクベースのアプリケーションリソース階層を作成する前に、次のタスクを実行してください。

1. ディスクおよび LUN のパーティションを分割してください。LifeKeeper の保護下にあるすべてのディスクのパーティションを分割する必要があるので、共有ディスクアレイは論理ユニット (LUN) に設定する必要があります。ディスクアレイ管理ソフトウェアを使用して、この設定を実行してください。詳細な手順については、ディスクアレイソフトウェアのマニュアルを参照してください。共有ストレージの場合は、ディスクをパーティション化することをお勧めします。

   **注意**: LifeKeeper では LUN レベルで **ディスク** をロックすることに注意してください。したがって、アクティブ/スタンバイ設定では 1 つの LUN が適切と考えられます。ただし、アクティブ/アクティブ設定を使用する場合は、少なくとも 2 つの別個の LUN を設定して、各階層が **独自の一意の** LUN にアクセスできるようにする必要があります。

2. 共有ディスクは **gdisk** などのユーティリティを使用してパーティション化することをお勧めします。

3. 両方のサーバーが共有ディスクを認識していることを確認します (たとえば、gdisk コマンドを使用)。 Linux が作成した LUN を認識しない場合、LifeKeeper も認識しません。

4. LifeKeeper 階層内のプライマリーサーバーとして使用する予定のシステムから、共有ディスク上にファイルシステムを作成します。 ファイルシステムの管理に関する詳細な手順については、Linux ドキュメントを参照してください。

# 4.3.5. ネットワーク設定の確認

LifeKeeper をインストールする前に、ネットワークが適切に設定され動作することを確認することが重要です。ネットワークの動作を確認するために、この時点で実行する必要のある作業を以下に示します。

1. もしサーバーのインストールでファイアーウォールが有効になっている場合、LifeKeeper のポートを開放するか、ファイアーウォールを無効にする必要があります。ファイアーウォールを使用した状態での LifeKeeper の実行 を参照してください。

2. 各サーバーから、ローカルサーバーを ping し、クラスター内の他のサーバーを ping してください。ping が失敗した場合には、必要なトラブルシューティングを行い、修正処置を実行してください。

3. サーバーに複数のネットワークアダプターがある場合は、アダプターが異なるサブネット上にあるように設定する必要があります。アダプターが同じサブネット上にある場合、TCP/IP は 2 つ目のアダプターを有効に利用できません。

4. *localhost* がクラスター内の各サーバーで解決可能であることを確認してください。DNS が実装されていない場合は、 */etc/hosts* ファイルを編集して、 *localhost* 名のエントリを追加してください。このエントリは、ローカルサーバーの IP アドレスをリストしたり、デフォルトのエントリ (127.0.0.1) をリストすることができます。 *localhost* が解決できない場合、LifeKeeper GUI が正常に機能しない可能性があります。

5. DNS が実装されている場合は、LifeKeeper クラスター内のサーバーが DNS を使用して解決できるように設定されていることを確認してください。

6. 各サーバーのホスト名が正しく、LifeKeeper をインストールした後に変更されないことを確認してください。後で LifeKeeper システムのホスト名を変更するように決定した場合は、 クラスター内のすべてのサーバーについて 以下の手順に従う必要があります。

   a. 次のコマンドを使用して、クラスター内のすべてのサーバー上の LifeKeeper を停止してください。

   ```
   /opt/LifeKeeper/bin/lkstop -f
   ```

   b. Linux **hostname** コマンドを使用して、サーバーのホスト名を変更してください。

   c. 続行する前に、新しいホスト名がクラスター内の各サーバーで解決可能であることを確認する必要があります (前の項目を参照)。

   d. クラスター内の各サーバーで次のコマンドを実行して、LifeKeeper のホスト名を更新してください。(詳細については、lk_chg_value(1M) を参照してください。)

   ```
   /opt/LifeKeeper/bin/lk_chg_value -o oldhostname -n newhostname
   ```

   e. 次のコマンドを使用して LifeKeeper を起動してください。

   ```
   /opt/LifeKeeper/bin/lkstart
   ```

LifeKeeper for Linux v7.x では、コミュニケーションパスおよび IP リソース用の VLAN インターフェースがサポートされています。VLAN インターフェースの種類は、以下に示すように選択することができます。

# VLAN インターフェースサポートマトリックス

- サポートなし \ x サポートあり

**LK Linux v7.1 以前のバージョン**

| VLAN_NAME_TYPE | コミュニケーションパス | IP リソース |
|---|---|---|
| DEV_PLUS_VID (eth0.0100) | - | x |
| DEV_PLUS_VID_NO_PAD (eth0.100) | - | x |
| VLAN_PLUS_VID (vlan0100) | x | x |
| VLAN_PLUS_VID_NO_PAD (vlan100) | x | x |

**LK Linux v7.2 以降のバージョン**

| VLAN_NAME_TYPE | コミュニケーションパス | IP リソース |
|---|---|---|
| DEV_PLUS_VID (eth0.0100) | x | x |
| DEV_PLUS_VID_NO_PAD (eth0.100) | x | x |
| VLAN_PLUS_VID (vlan0100) | x | x |
| VLAN_PLUS_VID_NO_PAD (vlan100) | x | x |
| 注： NIC名は何でも構いません（例：eth0、ens192…） | | |

# 4.3.6. 切り替え可能な IP アドレスの作成

切り替え可能な IP アドレスとは、クラスターノード間で切り替えることができる「仮想的な」IP アドレスです。各ノードのネットワークインターフェースカードに関連付けられた IP アドレスとは別個のものです。LifeKeeper の保護下にあるアプリケーションは、切り替え可能な IP アドレスに関連付けられています。これにより、アクティブノードで障害が発生した場合、その IP アドレスがスタンバイノードに「切り替わり」ます。

切り替え可能な IP アドレスに対してリソース階層を設定することを計画している場合は、クラスタの各ノードで以下の作業を実行する必要があります。

- コンピューター名が正しく、変更されないことを確認する。
- ping コマンドを使用して切り替え可能な IP アドレスが一意であることを確認する。
- */etc/hosts* ファイルを編集して、切り替え可能な IP アドレスごとにエントリを追加する。

詳細については、LifeKeeper for Linux IP Recovery Kit テクニカルドキュメンテーション を参照してください。

# 4.3.7. データベースアプリケーションのインストールとセットアップ

お使いの環境に Oracle、DB2、MySQL などの保護されたデータベースアプリケーションが含まれている場合は、データベースに付属のドキュメンテーションを使用してアプリケーションをインストールする必要があります。データベースが共有ファイルシステム上にあり、設定ファイルが共有ファイルシステム上にあることを確認してください。実行可能ファイルは、ローカルまたは共有のファイルシステム上に置くことができます。

LifeKeeper をインストールした *後に* アプリケーションをインストールすることも可能ですが、LifeKeeper の保護下に置く前に、適切に設定され稼動することを確認するためにアプリケーションをテストする必要があります。その他のインストールおよびセットアップの特定の考慮事項については、LifeKeeper データベースリカバリキットドキュメンテーション を参照してください。

# 4.3.8. GUI ユーザーの設定

## PAM による GUI の認証

LifeKeeper for Linux は、Linux Standard Base (LSB) で提供されている Pluggable Authentication Module (PAM) を利用するようになりました。LifeKeeper では、以前 /opt/LifeKeeper/website/passwd に置かれていた個別のパスワードファイルを現在は使用していません。それに代わり、ユーザーは、システムの PAM 設定に対して識別および認証されます。権限のレベルは、PAM が提供するグループメンバーシップから決定されます。

GUI にアクセスするには、ユーザーは LifeKeeper の次の 3 つのグループのいずれかのメンバーである必要があります: lkadmin、lkoper または lkguest。これらのグループのメンバーシップは、クラスター全体で使用されているユーザーアカウントデータベースのタイプに応じた適切な方法により、システム管理者が設定してください。

> ユーザーのパスワードを変更すると、クラスター内のすべてのノードで lkGUIserver の再起動が必要になる場合があります。

LifeKeeper のこれら 3 つのグループには、それぞれの権限セットが用意されています (権限テーブル を参照)。

1. クラスター全体にわたって ***Administrator (管理者)*** の権限を持つユーザー (`lkadmin`) は、GUI から可能な動作のすべてを実行できます。
2. 1 台のサーバー上で ***Operator (オペレータ)*** の権限を持つユーザー (`lkoper`) は、LifeKeeper の設定やステータスの情報を表示でき、そのサーバ上のリソースを in service や out of service にできます。
3. 1 台のサーバー上で ***Guest (ゲスト)*** の権限を持つユーザー (`lkguest`) は、そのサーバーの LifeKeeper の設定やステータスの情報を表示できます。

GUI パッケージのインストール時に、そのシステムの *root* ユーザーはシステムのローカルグループデータベースの `lkadmin` グループに自動的に追加されるので、 *root* は、そのサーバーから GUI アプリケーションまたは Web クライアント経由で LifeKeeper のすべての作業を実行できます。 *root* 以外のユーザーに LifeKeeper GUI クライアントの使用を許可するように計画している場合は、ユーザーを適切なグループに追加することにより LifeKeeper GUI のユーザーを設定する必要があります。

NIS、LDAP、AD などの非ローカルデータベースを使用するように PAM を設定している場合は、それらのデータベースでアカウントが正しく設定されていることをシステム管理者は確認する必要があります。上記のグループが存在していなければなりません。また、LifeKeeper GUI へのログインが許可されているユーザーがそれらのグループの 1 つに属するメンバーでなければなりません。これらのグループは、リモートデータベース内のみに作成する必要があり、ローカルの `/etc/group` ファイルからは削除する必要があります。

クラスター内のシステムで異なる「root」パスワードを使用している場合、LK GUI のログインは失敗します。クラスター内の各システムの root パスワードを同一にすれば、LK GUIに「root」でログインできます。

> **注記**：混乱を避け整合性を保つため、LDAP、NIS、AD などのより複雑な PAM 設定を利用する場合は、LifeKeeper のインストールまたはアップグレードを実行する前に、すべてのユーザーアカウントおよび LifeKeeper グループアカウントを準備しておいてください。

最良の方法は、常にクラスター全体を単位として許可を付与することです。サーバー単位で許可を付与することもできますが、ユーザーを混乱させてしまい、管理作業が実行できなくなります。

# 4.3.9. LifeKeeper ライセンスについて

LifeKeeper for Linux では、サーバーごとに一意のライセンスが必要です。

ライセンスは、ランタイムライセンスです。つまり、LifeKeeper のインストールはライセンスなしでも可能ですが、正常に製品を起動して実行するためには、起動前にライセンスをインストールする必要があります。

# 4.3.9.1. ライセンスの取得方法

日本の SIOS テクノロジーから購入されたお客様はユーザーポータルの「ライセンス取得方法解説ページ」を参照ください。

# 4.3.9.2. ライセンスのインストール方法

https://fast.wistia.net/embed/iframe/mtl4wubwfk

**参考動画 ライセンスのインストールについて（英語）**

クラスターの各ノードで以下の操作を実行します。

1. ライセンスをインストールします。

```
/opt/LifeKeeper/bin/lkkeyins <path to license file>
```

2. LifeKeeper を起動します。
   ライセンス情報を確認して、正しく登録されていることを確認します。
   正規ライセンスを適用している場合は、全てのライセンス情報が UNLICENSED、 または PERMANENT となります。

```
/opt/LifeKeeper/bin/typ_list -l
```

   a. UNLICENSED
      Core Application のライセンスが適用されている状態です。
   b. EVAL
      評価版ライセンスが適用されている状態です。
   c. PERMANENT
      オプション Recovery Kit のライセンスが適用されている状態です。
   d. NEEDED
      オプション Recovery Kit のライセンスが適用されていない状態です。

# 4.3.9.3. インターネットホスト ID の取得

マシンのインターネットホスト ID を取得するには、lmutil を使用します。インターネットホスト ID は、通常、システムのプライマリネットワークインターフェースのプライマリ IP アドレスです。インターネットホスト ID は、Ethernet (または MAC) ホスト ID の代替として使用することができ、VM クローンのために MAC アドレスが変更される可能性がある仮想環境において望ましいと考えられます。

1. 次のコマンドを実行します。

```
# /opt/LifeKeeper/bin/lmutil lmhostid -internet -n
```

2. プログラムから返される ID を記録してください。

例:

```
# /opt/LifeKeeper/bin/lmutil lmhostid -internet -n

“INTERNET=172.17.100.161”
```

注記: この情報は、SIOS Technology Corp. から取得したパーマネントライセンスキーに記載されている情報と一致する必要があります。

# 4.4. LifeKeeper ソフトウェアのインストール

本ページでは Linux OS について基本的な知識がある方にむけて LifeKeeper for Linux のインストールについて説明しています。 その他詳細については LifeKeeper for Linux テクニカルドキュメンテーション を参照してください。

## インストール前の要件

LifeKeeper for Linux をインストールする前に、下記の要件について確認してください。

- **LifeKeeper for Linux リリースノート** – LifeKeeper for Linux リリースノートにはサポート対象のプラットフォーム、オペレーティングシステム、アプリケーション、ストレージについて記載されています。最新のバグフィックスについても記載されています。
- **TCP/IP 接続と名前解決** – GUI の機能を使用するためにクラスターノードの双方で名前解決ができる必要があります。DNS サービスまたは */etc/hosts* を使用してください。 ローカルホストは 127.0.0.1 となります。
- ファイアーウォール – 以下のポートを使用します。
  - コミュニケーションパス (TCP): 7365/tcp
  - GUI サーバーとの通信: 81/tcp、82/tcp
  - GUI サーバーとクライアントとの RMI 通信: 1024/tcp 以降の全ポート
  - DataKeeper の同期 (DataKeeper 使用時): 10001+<mirror number>+<256 * i>

ファイアーウォールに関して

- GUI サーバー、クライアントの通信に使用するポートは LifeKeeper がインストールされるクラスターノードと、GUI クライアントを動作させる全てのシステムでポートが開放されている必要があります。
- DataKeeper で使用するポートは上記の計算式で割り出すことができます。i の値は 0から開始し、使用されていないポートが見つかるとそのポートを使用します。例えば mirror number が 0 の DataKeeper リソースが存在している環境で、10001 番ポートが別のアプリケーションによって使用されていた場合は、10257 番ポートを使用します。
- GUI サーバー、クライアント間の通信では Java の RMI にて 1024 番以降のポートをランダムに使用します。クラスターシステムにアクセス制御などを適用する場合は、これらのポートを考慮し、パケットフィルタリングを行う必要があります。セキュリティー対策の観点で本仕様が問題となる場合は、ssh の Xフォワーディングを使用して対応することもできます。設定方法等は テクニカルドキュメント をご参照ください。
- 使用しているポート番号に以下を追加します。 WebGUI サーバーのプロセスと lkpolicy コマンドによるポリシー設定で使用：778(SSL)/tcp

- ◦ **SELinux の設定確認** – SELinux のモードによっては、LifeKeeper for Linux はインストールできません。
  - enforcing モード – LifeKeeper for Linux をインストールできません。
  - permissive モード – LifeKeeper for Linux をインストールできます。(いくつかのテスト済み ARK 環境を除いて推奨できません。)
    - SAP 環境で必要にならない限り、SELinux を permissive モードにすることは推奨されていません。クラスターでの動作が必要となるアプリケーションが permissive モードをサポートしているか確認してください。SELinux の permissive モードは次の

ARK でテスト済みです。: SAP / SAP MaxDB / Sybase / Oracle / DB2 / NFS / DataKeeper / NAS / EC2 / IP / FileSystem / MQ。必要となるパッケージについては Linux の依存関係 を参照してください。

- disabled モード – LifeKeeper for Linux をインストールできます。
  - SELinux を無効にする方法は、OS 側のドキュメントを確認してください。
- ご利用のディストリビューションから提供される適切なパッケージをインストールしてください。
- sg3_utils パッケージは DMMP Recovery Kit、PowerPath Recovery Kit 等のマルチパス用のリカバリーキットを使用する環境に必要です。マルチパス用のリカバリーキット を使用しない環境には必要ありません。

- ◦ 既知の問題 を確認してください。– ご利用の環境に既知の問題がないか確認してください。

# LifeKeeper ソフトウェアのインストール

LifeKeeperソフトウェアを、LifeKeeper 設定内の各サーバーにインストールします。

LifeKeeper インストールセットアップスクリプトは依存するすべてのパッケージを確実にインストールするためにパッケージマネージャーツール（yum または zypper）を使用するため、LifeKeeper が依存するパッケージは自動的にインストールされます。

> **重要**：LifeKeeper を正しくインストールするには、yum または zypper の機能構成が必要です。非機能構成では、インストールが失敗する可能性があります（詳細については、インストールに関する既知の問題 を参照してください）。さらに、インストールがハングする可能性があるため、パッケージマネージャーレポジトリまたは rpm データベースはロックしないでください。依存パッケージをパッケージマネージャーから自動的にインストールできない場合は、事前にLinux の依存関係 を参照して必要なパッケージをインストールしてください。

LifeKeeper for Linux のイメージファイル (sps.img) では、システムに LifeKeeper をインストールする際に必要なユーザー対話型システムセットアップタスクを実行するように設計された、一連のインストールスクリプトを提供しています（対話モード を参照してください）。 非ユーザー対話型インストールを実行することもできます。非ユーザー対話型インストールについても同様です（非対話モード を参照してください）。

サーバーの Host ID または Entitlement ID を取得および表示するユーティリティを提供するライセンシングユーティリティパッケージもインストールされます。Host ID/Entitlement ID は、LifeKeeperを実行するための有効なライセンスを取得するために使用されます。ライセンスの取得とインストール方法については ライセンスの取得とインストール を参照してください。

> **注記:** これらのインストール手順は、読者がサーバーにインストールされた Linux オペレーティングシステムに精通していることを前提としています。

> **重要**：

• LifeKeeperの共有ストレージへのインストールはサポートされていません。各サーバーのローカルディスクに独自のコピーをインストールする必要があります。
• すべての LifeKeeperパッケージは、 /opt/LifeKeeper ディレクトリにインストールされます。

LifeKeeperは使用している Linux ディストリビューションに関わらず、コマンドラインでインストールされます。

- インストール作業は セットアップスクリプトの操作 を参照してください。
- アップグレード方法については、LifeKeeper のアップグレード を参照してください。

# 4.5. セットアップスクリプトの操作

LifeKeeper をインストールするために、セットアップスクリプトを使用して以下の作業を行います。

## セットアップスクリプトの機能

セットアップスクリプトは以下の2つの使用方法があります。

1. 対話型インストール
   メニュー画面からLifeKeeperの構成を設定して、インストールまで行います。
   このとき構成情報を保存すると、後述の非対話型インストールで使用できます。
2. 非対話型インストール
   保存した構成情報を用いて、LifeKeeperをインストールします。
   ユーザーへの問い合わせは発生しないので、構築ツール（Ansible）などから実行することも可能です。

## セットアップの手順

### 対話型インストール

1. root ユーザーでログイン後、次のコマンドを使用して、sps.img ファイルをマウントします。

   ```
   mount <PATH/IMAGE_NAME> <MOUNT_POINT> -t iso9660 -o loop
   ```

   ここで、PATH はイメージへのパスです
   IMAGE_NAME はイメージの名前です
   MOUNT_POINT はマウント位置へのパスです

2. sps.img がマウントされたディレクトリに移動して、次のコマンドを入力します。

   ```
   ./setup [-s <response_file>]
   ```

   -s オプションを指定すると、設定した構成情報をresponse_fileに保存できます。このファイルは非対話型インストールで使用します。

   以下は、 **mount** および **setup** の代替コマンドです。

   **Mount Image**: `mount -o loop <PATH/IMAGE_NAME> /mnt/`

   **Run Setup**: `/mnt/setup`

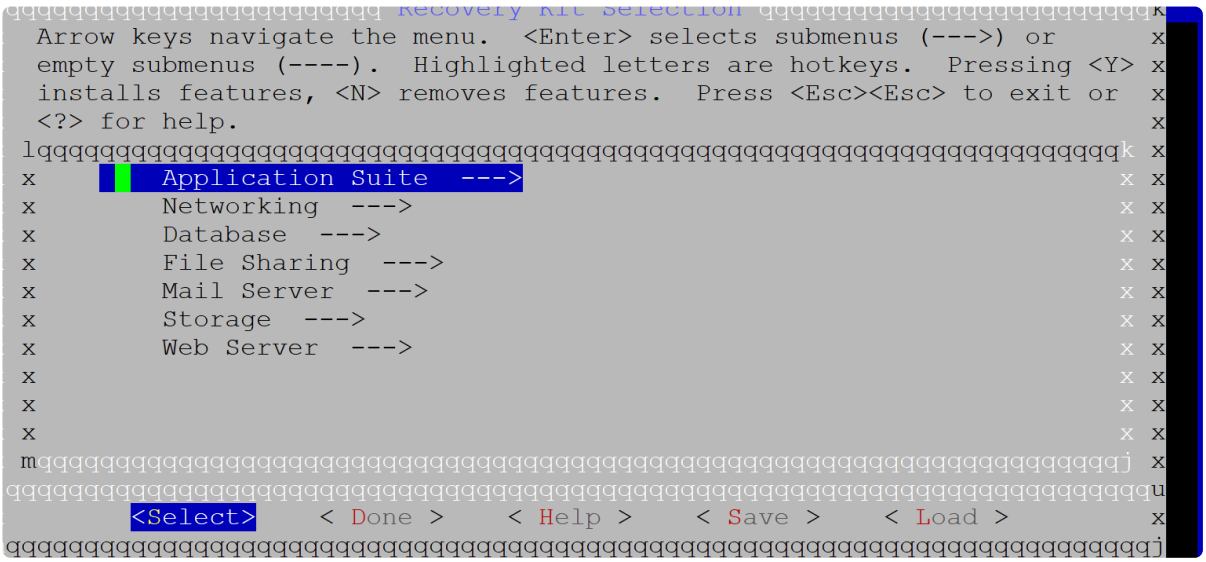
   ```
   [root@node2 ~]# mount -o loop /tmp/sps.img /mnt/
   mount: /mnt: WARNING: device write-protected, mounted read-only.
   [root@node2 ~]# /mnt/setup
   ```

3. スクリプトを実行すると、まず環境情報の収集を行います。

   インストールまたはアップグレードのシステム要件が満たされていない場合は、エラーメッセージが表示され、インストール/アップグレードはキャンセルされます。

   また、何らかの問題が発生した場合や設定の変更が必要な場合は、警告メッセージが表示され、インストールを続行するか中止するかをユーザーに決定するように求めます。

4. メインダイアログ画面からインストールする LifeKeeper の機能とアプリケーションリカバリキット (ARK) を選択します。




   詳細については ダイアログ画面の使い方 を参照してください。

5. インストールするSPSの機能やアプリケーションリカバリーキット（ARK）を選択し、 **<Done>** を選択すると、インストールが始まります。

   インストール完了後に何らかの通知が出力された場合は、その指示に従い必要な対応を行ってください。

## 非対話的インストール

1. root ユーザーでログイン後、次のコマンドを使用して、sps.img ファイルをマウントします。

   ```
   mount <PATH/IMAGE_NAME> <MOUNT_POINT> -t iso9660 -o loop
   ```

   ここで、PATH はイメージへのパスです
   IMAGE_NAME はイメージの名前です
   MOUNT_POINT はマウント位置へのパスです

2. LifeKeeperのインストールを行うシステムへ、構成情報ファイルをコピーした後、次のコマンドを実行します。

   ```
   ./setup -f <response_file> -q y
   ```

「-q y」オプションを指定すると、警告に対して了解したと渡します。

✻ ‘**sh setup**’ (bourne shell) は使用できません。代わりに **bash** を使用してください。

✻ **注記:** 非対話モードでのインストールで、-sオプションで保存した設定ファイルを使用する場合、そのファイルが使用されるシステムは、そのファイルが生成されたシステムと同じように構成されている必要があります。システムの違いが多すぎると、非対話モードでのインストールは失敗する可能性があります。
create_response_fileスクリプトで作成した設定ファイルはそのような制限がありません。

## 構成情報の作成

非対話型インストールで使用する構成情報ファイルは、`setup -s <response_file>`でセットアップ作業中に作成するか、create_response_fileスクリプトで事前に作成することが出来ます。

1. root ユーザーでログイン後、次のコマンドを使用して、sps.img ファイルをマウントします。

   ```
   mount <PATH/IMAGE_NAME> <MOUNT_POINT> -t iso9660 -o loop
   ```

   ここで、PATH はイメージへのパスです
   IMAGE_NAME はイメージの名前です
   MOUNT_POINT はマウント位置へのパスです

2. sps.img がマウントされたディレクトリに移動して、次のコマンドを入力します。

   ```
   ./create_response_file <response_file>
   ```

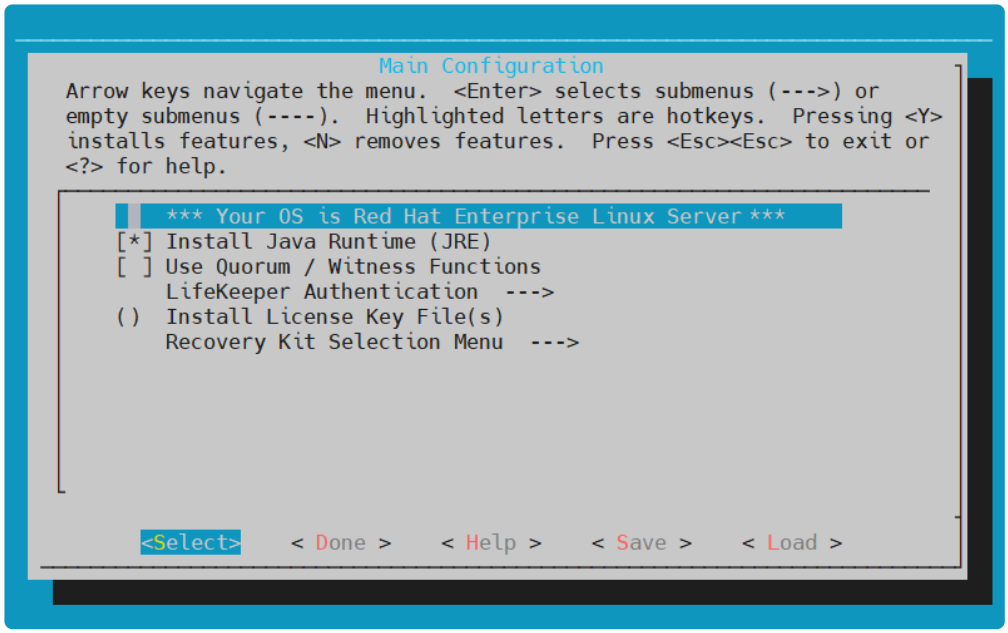
3. メニュー画面にてインストールするSPSの機能やアプリケーションリカバリーキット（ARK）を選択します。詳しくは「対話画面の操作」を参照してください。

4. インストールするSPSの機能やアプリケーションリカバリーキット（ARK）を選択し、 **<Done>** を選択すると、構成をresponse_fileへ保存してスクリプトが終了します。response_fileはインストール先のシステムへコピーして使用します。

   保存された構成情報を変更する場合は、再度手順2～4を実行してください。

## 対話画面の操作

以下のようなメニュー画面が表示されます。




以下のキーで項目選択を行います。

| | |
|---|---|
| ↑↓ | 選択項目の移動 |
| ← → | 最下行ボタンの移動 |
| ENTER | 選択したサブメニューを開く |
| Y / N / Enter | 選択した項目のオン、オフ、または反転 |

最下行ボタンは以下の動作を行います。

| | |
|---|---|
| Select | 選択項目の ON / OFF ・子画面へ移動 |
| Done | 現画面を終了して親画面に戻ります<br>Main Configuration 画面の場合は、メニュー画面を終了して、インストールの確認画面へ移行します |
| Help | 選択項目のヘルプ画面を表示します |
| Save | 設定した構成情報をファイルに保存します<br>保存した構成情報ファイルは非対話モードでのインストールに使用できます<br>(create_response_fileでは無効です) |
| Load | 保存済み構成情報ファイルの読み込み<br>(create_response_fileでは無効です) |

メニューボタン <Save> および <Load> をクリックすると、現在の設定を保存するとき、または保存した設定を読み込むときに使用する設定ファイル名を尋ねるダイアログ画面が表示されます。デフォルトのファイル名を変更したい場合は、[TAB] キーを使用してファイル名フィールドに移動し、新しいファイル名を入力してください。 **注記:** <Save> をクリックした場合、同じ名前でファイルを上書きする前に確認を求められます。

各項目では以下の機能を設定できます。

アップグレード中は、設定可能な項目のみが一覧表示されます。 ホットキー <Z> を使用すると、アップグレード時は変更されない項目が表示されます。

- **Install Java Runtime (JRE)**

  LifeKeeper GUI が使用する Java 実行環境をインストールします。

- **Restart NFS service**

  NFS ARK を使用する場合に選択可能です。NFS サービスの設定変更を即時反映させたい場合に選択します。

  注記： NFSサービスを自動的に再起動したくない場合は、NFS Recovery Kit を使用する前に設定変更を反映させるための再起動を行う必要があります。

- **Use Quorum / Witness functions**

  Quorum/Witness を使用します。機能の詳細はテクニカルドキュメンテーションのQuorum/Witness を参照してください。

- **LifeKeeper Authentication**

  SPS for Linux GUI へのログインを許可するユーザーとその権限を指定します。 複数のユーザー名を空白で区切って指定できます。 詳細は、 GUIユーザー設定 を参照してください。

- **Install License Key File(s)**

  LifeKeeper のライセンスファイルを指定します。ファイル名を入力できるダイアログに切り替わるので、ライセンスファイルのフルパス名を入力して下さい。スペース区切りで複数のライセンスファイルを指定することが出来ます。

  ライセンスの取得方法についてはライセンスの取得とインストール を参照してください。

- **Recovery Kit Selection Menu**

  インストールするアプリケーションリカバリーキットを選択します。 アプリケーションリカバリーキットは、機能に基づいていくつかのカテゴリーに分類されています。

  詳細については、アプリケーションリカバリーキットの分類 を参照してください。

- **LifeKeeper Startup After Install**

  インストールもしくはアップデート完了後に LifeKeeper を起動します。

注記: SPS-L データレプリケーションパッケージは、サポートされているOSディストリビューションによってはカーネルモジュールをインストールする場合があるため、カーネルのアップグレード時に SPS-L の再インストールが必要になることがあります。これは、

> カーネルバージョン 3.10.0-514 以降を実行している RedHat、CentOS、Oracle Enterprise Linux（UEK カーネル含む）、及び 4.12.14-95 以降の SUSE Linux Enterprise Server の OS ディストリビューションに適用されます。

## アプリケーションリカバリーキットの追加と削除

インストール完了後にアプリケーションリカバリーキットを追加するには、setup を実行し、[Recovery Kit] でリカバリーキットを選択し、次にアプリケーションリカバリーキットのカテゴリーを選択して、ご希望のキットを選択します。不要になったアプリケーションリカバリーキットの選択を解除すると、そのキットは削除されます。
ただし、使用中のリソースはキットを削除できないので、事前にリソースを削除してください。

## 修復インストール

SPS for Linux インストールを修復するには、「--force」オプションを使用してセットアップを実行します。 修復インストールでは、失われたファイルや破損したファイルを置き換えてインストールを更新します。

## setup スクリプトのオプション

setup スクリプトは以下のオプションを指定して実行できます。

- `-f <response_file>`

  非対話モードでインストールを行います。<response_file>には、インストール中に使用する設定情報が含まれています。

- `-s <response_file>`

  メニューで選択した内容を含む設定ファイルを保存します。このファイルを「-f」オプションと一緒に使用して、同じ LifeKeeper 構成を別のシステムにインストールすることができます。たとえば、次のコマンドを実行します。

  ```
  ./setup -s <response_file>
  ```

  必要なパッケージとオプションを選択し、セットアップを完了します。

  その後、以下のコマンドを実行し、最初にセットアップを実行したときに選択したのと同じオプションで LifeKeeper のサイレントインストールを（別のシステムで）実行します。

  ```
  ./setup -f <response_file> -q y
  ```

- `--force`

  SPS for Linux を強制的に再インストールします。

- `-q <y/n>`

  非対話モードでのインストール中に発生する可能性のある確認への回答を指定します。

## リカバリーキットの分類

| カテゴリー | 説明 |
|---|---|
| Application Suite | SAPやIBM MQ などアプリケーションを保護するリカバリーキットが所属します |
| Networking | EC2やRoute53など、クラウド内のネットワークサービスを保護するリカバリーキットが所属します |
| Database | Oracle、PostgreSQL、MaxDBなどのデータベースアプリケーションを保護するリカバリーキットが所属します |
| File Sharing | NFSやSambaなどのファイル共有サービスを保護するリカバリーキットが所属します |
| Mail Server | Postfixなどのメールサービスを保護するリカバリーキットが所属します |
| Storage | DataKeeper（レプリケーション）、Device Mapper（DM）マルチパス（DMMP）、Network Attached Storage（NAS）など（これらに限らない）のデータ保管方法を保護するリカバリーキットが所属します |
| Web Server | Apache などのウェブサービスを保護するリカバリーキットが所属します |

# 4.6. LifeKeeper インストールの確認

LifeKeeper パッケージが正しくインストールされていることを確認するには、コマンドラインで次のコマンドを入力してください。

```
rpm -V <package name>
```

> **注記:** パッケージが正しくインストールされている場合、このコマンドは何も出力しません。

コマンドラインからクエリを実行するには、次のコマンドを入力してください。

```
rpm -qi <package name>
```

> **注記:** このコマンドの予想される出力は、パッケージ情報です。

# 4.7. LifeKeeper のアップデート

✻ v9.5.0 より、SIOS は新しい SAP HANA Application Recovery Kit をリリースしました。SIOS は **2022年3月31日** まで、9.4.x リリースでの SAP HANA gen/app ベースの Recovery Kit のサポートを継続します。
• 既存の SAP HANA gen/app ベースの Recovery Kit は v9.5.0 では **サポートされていません** 。
• LifeKeeper for Linux v9.5 以降をご利用の場合は、新しい（ビルトインの）SAP HANA Application Recovery Kit を使用する必要があります。
• LifeKeeper for Linux v9.5.0 へのアップデートを希望するユーザーは、既存の SAP HANA gen/app ベースの Recovery Kit から新しい SAP HANA Recovery Kit に **移行する必要があります** 。

LifeKeeper for Linux は、既存の階層を保ちながら将来のリリースにアップデートすることができます。この情報をよく検討して、アプリケーションのダウンタイムを最小限に抑えるようにしてください。

✻ **注意**：最新バージョンの LifeKeeper にアップデートできるのは、2 世代前までのバージョンです。(2 世代間アップデートの例： **9.1.x -> 9.3.x**) 。それより前のバージョンからアップデートする場合、古いバージョンをアンインストールし、LifeKeeper を再インストールする必要があります。旧バージョンをアンインストールする代わりに、旧バージョンを 1 世代前もしくは 2 世代前のバージョンにアップデートした上で、最新バージョンにアップデートする方法もあります。

✻ **注意**：アップデート中に `lkbackup` を使用する場合の詳細については、lkbackupの既知の問題 を参照してください。

1. クラスターをアップデートする場合、アップデートするサーバーからすべてのアプリケーションを切り替えてください。この操作を実行するには手動で行うか、または LifeKeeper のシャットダウンストラテジーを「**Switchover**」に設定します。これにより、LifeKeeper が停止したり、サーバーがシャットダウンしたときにアプリケーションが切り替えられます。シャットダウンストラテジーの詳細については サーバーのシャットダウンストラテジーの設定 を参照してください.
2. セットアップスクリプトの操作 を参照して LifeKeeper をアップデートします
3. アップデートされたサーバーにすべてのアプリケーションを切り替えてください。
4. アップデートする LifeKeeper クラスタ内の各サーバーに対してこの手順を繰り返します。専用の Quorum/Witnessノード（リソースインスタンスを持たないノード）を含むクラスタでは、LifeKeeper をアップデートする前にアプリケーションを切り替える必要はありません。

! **注意**：同じクラスター内のすべてのシステム上に、同じバージョンおよびリリースの LifeKeeper をインストールする必要があります。通常バージョンまたはリリースの異なる LifeKeeper には互換性がありません。ローリングアップデート以外の状況で、異なるバージョンまたはリリースが存在し、クラスタ内の別のシステムで実行されている場合には、LifeKeeper を起動しないでください。

# 4.8. LifeKeeper を使用したノードの OS / カーネルのアップデート (OS パッチ適用)

> ! 弊社製品のバックアップに加え、仮想マシンにOSパッチを適用する前に、更新されたシステムを以前の日付/時間に復元する必要がある場合に備えて、 **システム全体** のスナップショット/バックアップを実行することを推奨します。 追加のスナップショット情報については、DataKeeper for Linux が保護している領域に対する、スナップショット機能の利用についての注意事項 を参照してください。

OS をアップデートする場合、現在インストールされている LifeKeeper のバージョンが アップデートする OS のバージョンをサポートしているか確認してください。 サポートされていない場合、新しいバージョンの OS をサポートする LifeKeeper のバージョンがリリースされていれば、LifeKeeper をアップデートする必要があります。 (LifeKeeper のアップデート を参照してください。) 新しい OS のバージョンをサポートする LifeKeeper がリリースされていない場合、OS をアップデートできない可能性があります。 サポート対象のオペレーティングシステム を参照してください。

OS をアップデートする前に、LifeKeeper の構成を lkbackup コマンド でバックアップすることをお勧めします。

> ✻ 注意: lkbackup を使用する際は、lkbackup の既知の問題 を参照してください。

1. クラスターをアップデートする場合、すべてのリソース階層とそれらが保護するアプリケーションは、アップデートするサーバーからクラスター内のスタンバイノードに切り替える必要があります。この作業は、手動で行うか、LifeKeeper Shutdown Strategy を「Switchover」に設定することで行うことができます。 Shutdown Strategy を「Switchover」に設定することで、LifeKeeper の停止時またはサーバのシャットダウン時に、リソース階層がスタンバイノードに切り替わります。

2. LifeKeeper を停止します。

3. OS / Kernel をアップデートします。 (**再起動** が必要かどうかを示すメッセージが表示されます)
   a. **注記**: アップデート後、再起動が **必要な場合** については、以下のシナリオを参照してください。

4. 新しいOS / Kernel に対応するために必要であれば、LifeKeeper をアップデートしてください。LifeKeeper をアップデートしない場合でも、LifeKeeperセットアップを再度実行し、新しいOS / カーネルに対応する設定を更新する必要があります。

5. LifeKeeper を起動します。

6. すべてのリソース階層をアップデートされたサーバーに切り替えます。

7. LifeKeeper クラスター内のすべてのノードに対してこれらの手順を実行します。専用の Quorum /Witness ノード (リソースインスタンスのないノード) を含むクラスターの場合、LifeKeeper を更新する前にアプリケーションを切り替える必要はありません。

8. アップデート中に LifeKeeper GUI が起動していた場合は、(`/opt/LifeKeeper/bin/lkGUIapp` ) LifeKeeper GUI を再起動します。

> **注意:** サポート対象と見なされるには、クラスター内のすべてのノードが同じバージョンの OS と同じバージョンの LifeKeeper を実行している必要があります。 アップデート作業中のみ、一時的にノードの OS と LifeKeeper のバージョンが異なる状態が許容されます。

インストール / アップデート後に再起動が必要なシナリオ

1. LifeKeeper セットアップスクリプトが必要なカーネルモジュールを再ロードできない場合は、システムを **再起動する必要があります。** この場合、セットアップスクリプトは、次のいずれかに似た警告メッセージを表示します。
   - “`Unable to reload modules after adding LifeKeeper for Linux specific configuration information in /etc/modprobe.d. Please reboot your system after setup completes to ensure these modules load correctly and allow LifeKeeper for Linux to function properly.`“
   - “`Updated some modules for DataKeeper. Reboot the system to use the new module.`“

2. セキュアブートを無効にした場合は、システムを **再起動する必要があります。** DataKeeper のインストール時にセキュアブートが有効になっている場合、セットアップスクリプトは次のような警告メッセージを表示します。
   - “`Secure Boot cannot be enabled in a DataKeeper environment. Please take one of the following actions: a) Disable Secure Boot [recommended] or b) Disable signature verification (mokutil –disable-validation)`“.

# カーネルのアップデート後の新しいマウントオプションまたは非推奨のマウントオプション

Linux カーネルをアップデートすると、既存のファイルシステムマウントオプションの一部が新しいカーネルで非推奨になったり、新しいカーネルが新しいデフォルトマウントオプションを既存のマウントに追加したりする可能性があります。例えば、 “nobarrier” マウントオプションは RedHat Enterprise Linux 8 で廃止され、一部のカーネルバージョンでは“logbufs=8” や “logbsize=32k”などの新しいデフォルトマウントオプションが追加されました。

LifeKeeper で保護されたファイルシステムリソースに、カーネルのアップデート後に廃止されるマウントオプションが含まれている場合、廃止されたオプションは、クラスター内のすべてのサーバーで LifeKeeper リソースのマウントオプションのリストから削除する必要があります。 詳細については、LifeKeeper ファイルシステムリソースのマウントオプションの変更 セクションを参照してください。

カーネルのアップデート後に、LifeKeeper で保護された既存のマウントポイントにカーネルによって新しいデフォルトのマウントオプションが追加された場合、クラスター内のすべてのサーバーで LifeKeeper リソースのマウントオプションのリストに新しいオプションを追加する必要があります。 詳細については、

LifeKeeper ファイルシステムリソースのマウントオプションの変更 セクションを参照してください。

# 5. LifeKeeper for Linux テクニカルドキュメンテーション

LifeKeeper for Linuxは、単一のエンタープライズクラスソリューションで、高可用性クラスタリングと革新的なデータ複製機能を統合します。

## LifeKeeper for Linuxで統合されたコンポーネント

**SIOS LifeKeeper** は、完全な耐障害性ソフトウェアソリューションを提供し、サーバーのファイルシステム、アプリケーション、およびプロセスの高可用性を実現します。LifeKeeperは、カスタマイズされたフォールトトレラントハードウェアを必要としません。LifeKeeperでは、ネットワーク内で2つ以上のシステムをグループ化するだけでサイト固有の設定データが作成され、自動的に障害を検出して回復します。

障害が発生した場合、LifeKeeperは保護されているリソースを障害が発生したサーバーから指定されたバックアップサーバに移行します。実際のスイッチオーバー中にユーザーの操作は一瞬中断されますが、LifeKeeperはオペレーターの介入なしにバックアップサーバーの操作を復元します。

**SIOS DataKeeper** は、LifeKeeper環境用の統合データミラーリング機能を提供します。この機能により、LifeKeeperリソースを共有ストレージを使用している環境および共有ストレージをしていない環境で運用できます。

---

ドキュメンテーションとトレーニング

# 5.1. はじめに

## LifeKeeper for Linux について

LifeKeeper for Linux は、高可用性のクラスタリングと革新的なデータ複製機能をエンタープライズクラスのソリューションに統合したものです。

### LifeKeeper for Linux の統合コンポーネント

**SIOS LifeKeeper** は、障害回復性の高いソフトウェアソリューションであり、お使いのサーバのファイルシステム、アプリケーション、およびプロセスの高い可用性を維持します。LifeKeeper には、カスタマイズした耐障害性のハードウェアは不要です。LifeKeeper を使用するには、ネットワーク内にある 2 台以上のシステムをグループ化するだけです。サイト固有の構成データが作成され、自動の障害検出とリカバリが実行されます。

障害が発生した場合、障害が発生したサーバから LifeKeeper が保護しているリソースを指定のバックアップサーバに移行します。実際のスイッチオーバ時に短時間の中断が発生します。ただし、オペレータの介入なしに LifeKeeper がバックアップサーバに動作をリストアします。

**SIOS DataKeeper** は、LifeKeeper 環境において統合データミラーリング機能を提供します。この機能により、LifeKeeper リソースが共有/非共有ストレージ環境で動作可能になります。

---

ドキュメンテーションとトレーニング

# 5.2. ドキュメンテーションとトレーニング

## ドキュメンテーション

LifeKeeper for Linux をインストール、設定、管理、およびトラブルシューティングするための方法を説明する完全なリファレンスです。次のセクションは、LifeKeeper for Linux のあらゆる側面を網羅します。

| セクション | 説明 |
|---|---|
| はじめに | LifeKeeper for Linux 製品の概要を説明します。ソフトウェアパッケージと設定コンセプトが含まれます。 |
| LifeKeeper for Linux インストールガイド | LifeKeeper 環境のプランニングと設定、LifeKeeper のインストールとライセンス、LifeKeeper のグラフィカルユーザインターフェース (GUI) の設定に役立つ情報を提供します。 |
| 設定 | クラスタ内の各サーバで LifeKeeper ソフトウェアを設定するための詳細情報と手順があります。 |
| 管理 | サーバのプロパティの編集やリソースの作成などのサーバレベルの作業、およびリソースの編集、拡張、削除などのリソースレベルの作業について説明します。 |
| User's Guide | LifeKeeper GUI で実行できる多数の作業を含めて、LifeKeeper の GUI に関する詳細情報があります。テクニカルノート セクション、および多数の高度なトピック もあります。 |
| DataKeeper | SIOS DataKeeper for Linux の計画とインストールの手順、および管理、設定、およびユーザの情報があります。 |
| トラブルシューティング | 既知の問題と制限について説明し、SIOS LifeKeeper for Linux のインストール、設定、および使用を行うときに発生する可能性がある問題に対する解決策を説明します。 |
| Recovery Kit | LifeKeeperで特定のアプリケーションを管理および制御するために必要なオプションの Recovery Kit のプランニングおよびインストール手順、管理、設定、およびユーザ情報が含まれます。 |
| エラーコードの検索 | LifeKeeper for Linux の使用中に表示される可能性のあるすべてのメッセージの一覧が含まれます。必要に応じて、エラーの原因およびエラー状態を解消するために必要な処置についても説明しています。この全一覧から、受信したエラーコードを検索できます。 |

## トレーニング

LifeKeeper トレーニングは、SIOS Technology Corp. または代理店から受講可能です。詳細については、営業担当者にお問い合わせください。

# テクニカルサポート

SIOS Technology Corp. と有効なサポート契約を結んだお客様は、SIOS Technology Corp. のセルフサービスサポートポータル にアクセスできます。

SIOS Technology Corp. のセルフサービスサポートポータル では、以下のことができます。

- 弊社の ソリューションナレッジベース から、問題の解決策と質問に対する回答を検索する。
- 次のメニューを選択して、年中無休の SIOS Technology Corp. のサポートチームにアクセスする。
  - **Log a Case** - 新しいインシデントを報告する。
  - **View Cases** - お客様の未解決と解決済みのインシデントをすべて表示する。
  - **Review Top Solutions** - 弊社のお客様が表示した、最も一般的な問題の解決策の情報を表示する。

セルフサービスポータルを設定してアカウントを有効にする方法については、SIOS Technology Corp. のサポート (support@us.sios.com) にお問い合わせください。

また、SIOS Technology Corp. のサポートには、以下の方法でも連絡できます。

1-877-457-5113 (通話料無料)

1-803-808-4270 (米国以外のお客様)

電子メール:support@us.sios.com

# 5.3. lkbackup

## lkbackup を使用したバックアップの作成

> ! LifeKeeper もしくは OS のアップグレードを実施する際は、全てのノードで LifeKeeper 階層のバックアップを作成してください。

以下のコマンドを実施してバックアップを作成してください。

```
/opt/LifeKeeper/bin/lkbackup -c
```

以下のファイルにバックアップが作成されます。

*/opt/LifeKeeper/config/archive.<date-time-stamp>.tar.gz*

### 自動 lkbackup

> ✻ 自動 **lkbackup** は 7.5 以上のバージョンで実行できます。バックアップファイルは以下の名前で保存されます。
> */opt/LifeKeeper/config/auto-backup.<x>.tgz* ( <x> は、シーケンス番号です。)

#### 自動 lkbackup の時間を変更するには

1. */etc/crontab* にアクセスします。

   エントリーは以下のようになっています。

   *0 3 * * * root /opt/LifeKeeper/bin/backupadm -c* (0 3 はバックアップが実施される時刻を表し、この例では毎日 3:00 にバックアップが実施されます。)

2. 時刻を変更します。

## バックアップからリストアを実行する

> ✻ バックアップからのリストアは現在ご利用中のバージョンに制限されます。バックアップはバージョンをまたいでリストアしないでください。

以下のコマンドを使用してリストアを実行します。

1. `/opt/LifeKeeper/bin/lkcli stop`
2. `/opt/LifeKeeper/bin/lkbackup -x -f <file-name>`

3. `/opt/LifeKeeper/bin/lkcli start`

クラスター内の全てのシステム上でバックアップからリストアを実行してください。

DataKeeper リソースで lkbackup を使用する場合は、 全同期 が必要です。

# 5.4. LifeKeeper

以下の LifeKeeper 製品のマニュアルは、サイオステクノロジーのウェブサイトでご覧いただけます。

- LifeKeeper for Linux リリースノート
- LifeKeeper for Linux テクニカルドキュメンテーション
- アプリケーションリカバリーキットのマニュアル

# 5.4.1. はじめに

SIOS LifeKeeper for Linux は、さまざまなストレージ構成をサポートし、最大 32 ノードの高可用性クラスタリングを提供します。共有ストレージ（ファイバチャネル SAN、iSCSI）、ネットワーク接続ストレージ（NAS）、ホストベースの複製、HP Continuous Access などのアレイベースの SAN 複製との統合、VMware 仮想ハードディスク（VMDK）などをサポートします。

---

保護対象のリソース

LifeKeeper Core

設定の概念

- 共通のハードウェアコンポーネント
- システムのグループ化の配置
- アクティブ - アクティブのグループ化
- アクティブ - スタンバイのグループ化
- Intelligent スイッチバックと Automatic スイッチバックの違い
- syslog によるログの記録
- リソース階層
  - リソースタイプ
  - リソースの状態
  - 階層の関係
  - イクイバレンシ情報
  - リソース階層の情報
  - リソース階層の例
  - ステータスの詳細表示
  - ステータスの簡略表示

障害検出とリカバリのシナリオ

- IP ローカルリカバリ

リソースのエラーリカバリのシナリオ

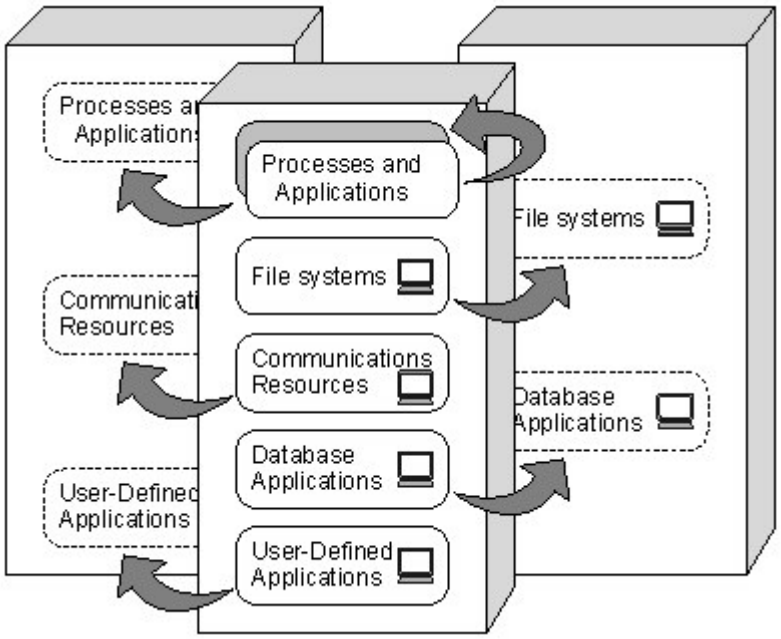
サーバの障害リカバリのシナリオ

# 5.4.1.1. 保護対象のリソース

LifeKeeper ファミリの製品には、多様なシステムリソースにフェイルオーバ保護を提供できるソフトウェアがあります。以下の図に、LifeKeeper の柔軟性、および自動リカバリを指定できるリソースタイプを示します。

- ファイルシステム 。LifeKeeper では、ext3、ext4、NFS、vxfs、xfs などのファイルシステムの指定とフェイルオーバができます。
- 通信リソース 。LifeKeeper には、TCP/IP のような通信リソースの通信 Recovery Kit が用意されています。
- インフラストラクチャリソース 。LifeKeeper には、NFS、Samba、LVM、WebSphere MQ、ソフトウェア RAID (md) など、Linux インフラストラクチャサービス用のオプションの Recovery Kit が用意されています。
- **Web** サーバリソース 。LifeKeeper には、Apache Web サーバリソース用のオプションの Recovery Kit が用意されています。
- データベースとその他のアプリケーション 。LifeKeeper には、Oracle、MySQL、PostgreSQL、Sybase、SAP DB/MaxDB および SAP HANA DB 1.0 / SAP HANA DB 2.0 などの主な RDBMS 製品、および SAP などのエンタープライズアプリケーション用のオプションの Recovery Kit が用意されています。
- クラウドリソース 。LifeKeeper には、EC2 EIP、Route53 などの通信リソース用のRecovery Kit が用意されています。

LifeKeeper は、多様なリソースタイプについて複数の回復方法 をサポートします。



# 5.4.1.2. LifeKeeper Core

LifeKeeper Core は、以下の 4 つの主要コンポーネントで構成されています。

- LifeKeeper Core ソフトウェア
- File System、Generic Application、Raw I/O、 IP、および Quick Service Protection (QSP) の Recovery Kit ソフトウェア
- LifeKeeper GUI ソフトウェア
- LifeKeeper のマニュアルページ

## LifeKeeper Core ソフトウェア

LifeKeeper Core ソフトウェアは、以下のコンポーネントで構成されます。

- LifeKeeper 構成データベース (LCD) - LCD は、LifeKeeper が保護するリソースの情報を保存します。リソースインスタンス、依存関係、イクイバレンシ情報、リカバリの方向、LifeKeeper の動作フラグに関する情報が含まれます。システムの起動後にデータが記憶されているように、データは共有メモリにキャッシュされ、ファイルに保存されます。
- LCD インターフェース (LCDI) - LCDI は、LCD に保存されているデータやデータの変更を要求するクエリを設定データベース (LCD) にクエリを送信します。また、リソースの状態や説明の情報を取得するために、Application Recovery Kit が LCDI を使用することもできます。
- LifeKeeper Communications Manager (LCM) - LCM は、クラスタ内にあるサーバのステータスの特定、および LifeKeeper のプロセス間通信 (ローカルとリモート) に使用されます。クラスタ内のあるサーバ上にあるすべてのコミュニケーションパスで LCM 通信がないことは、サーバに障害が発生したことを示します。
- LifeKeeper アラームインターフェース - LifeKeeper アラームインターフェースは、イベントを起動するためのインフラストラクチャです。LifeKeeper が保護するリソースに障害が検出された場合、アプリケーションデーモンにより sendevent プログラムが呼び出されます。sendevent プログラムが LCD と通信し、リカバリプロセスが使用可能かどうかを判断します。
- LifeKeeper のローカルリカバリ動作と制御のインターフェース （LRACI） - LRACI はリソースに適切なリカバリスクリプトを判断し、リソースに適切な restore / remove スクリプトを呼び出します。

## File System、Generic Application、IP、RAW I/O、および Quick Service Protection (QSP) の Recovery Kit ソフトウェア

LifeKeeper Core は、サーバ上の指定リソースを保護します。リソースを以下に示します。

- File Systems - LifeKeeper では、共有ストレージデバイス上にあるファイルシステムの指定とフェイルオーバができます。ファイルシステムは、共有 SCSI バス経由で 2 台のサーバからアクセス可能なディスク上に作成できます。LifeKeeper のファイルシステムリソースは、1 台目のサーバに作成されてから、2 台目のサーバに拡張されます。ファイルシステムの健全性監視 がディスクフルと不適切なマウント (またはアンマウント) のファイルシステム条件を検出します。検出した条件に従って、Recovery Kit が警告メッセージのログ記録、ローカルリカバリの試行、またはファイルシステムリソースのバックアップサーバへのフェイルオーバを実行できます。

  File System Recovery Kit に関連するヘルプトピックとして、ファイルシステムのリソース階層の作成 、拡張 、ファイルシステムの健全性の監視 などがあります。

- Generic Applications - Generic Application Recovery Kit は、リソースタイプに対して事前定義リカバリキットが指定されていない汎用アプリケーションやユーザ定義アプリケーションを保護できます。このキットを使用すると、特定アプリケーションについてカスタマイズした監視スクリプトやリカバリスクリプトを指定できます。

  Generic Application Recovery Kit に関連するヘルプトピックとして、汎用アプリケーションのリソース階層の作成 、拡張 などがあります。

- IP Addresses - IP Recovery Kit には、LifeKeeper 環境で、障害が発生したプライマリサーバから「切り替え可能な」IP アドレスをバックアップサーバにリカバリするメカニズムがあります。切り替え可能な IP アドレスとは、サーバ間で切り替えることができる仮想 IP アドレスであり、各サーバのネットワークインターフェースカードに関連付けられている IP アドレスとは別のものです。LifeKeeper で保護されているアプリケーションは切り替え可能な IP アドレスに関連付けられているので、プライマリサーバに障害が発生した場合、切り替え可能な IP アドレスはバックアップサーバに関連付けられます。LifeKeeper で保護されているリソースは、切り替え可能な IP アドレスです。

  特定の製品、構成、および管理に関する情報については、リカバリキットに含まれるIP Recovery Kit Technical Documentation を参照してください。

- RAW I/O - RAW I/O Recovery Kit は、カーネルのバッファリングを迂回するアプリケーションの RAW I/O デバイスをサポートします。RAW I/O Recovery Kit では、共有ストレージデバイスにボンディングされた RAW デバイスの指定とフェイルオーバができます。RAW デバイスは、リソースの作成前に、プライマリノードに設定する必要があります。RAW リソースを作成 した後、追加サーバに拡張 できます。

- Quick Service Protection (QSP) - QSP Recovery Kit は、OSのサービスを簡易的に保護する仕組みを提供します。OSのserviceコマンドで起動 (start)・停止 (stop) ができるサービスであれば、容易にリソースが作成できます。Generic Applicationを使っても同じことが実現できますが、QSPを使えばコードの開発が不要です。また、依存関係を作成することで、他のリソースで保護しているアプリケーションと連携してサービスを起動･停止させることができます。

  ただし、QSPのquickCheckでは簡易なチェックしか行っていません（serviceコマンドのstatusを利用しています）。実際にそのサービスが提供できているか、プロセスが動作しているかは保証されません。複雑な起動や停止処理、堅牢なチェックが必要な場合は、Generic Applicationの利用を検討してください。

QSPに関連するヘルプトピックスとして、QSPリソースの作成・拡張 があります。

## LifeKeeper GUI ソフトウェア

LifeKeeper GUI は、Java テクノロジを使用して開発されたクライアント/サーバアプリケーションであり、LifeKeeper およびその設定データ用のグラフィカルな管理インターフェースです。LifeKeeper GUI クライアントは、スタンドアロンの Java アプリケーション として実装されます。

## LifeKeeper のマニュアルページ

LifeKeeper 製品用の LifeKeeper Core のリファレンスマニュアルページです。

# 5.4.1.3. 設定の概念

LifeKeeper は、2 台以上のサーバを持つグループに対してユーザが定義したリソース階層に基づいて機能します。以下のトピックで、LifeKeeper のフェイルオーバ設定の概念を説明しています。

---

共通のハードウェアコンポーネント

システムのグループ化の配置

アクティブ - アクティブのグループ化

アクティブ - スタンバイのグループ化

Intelligent スイッチバックと Automatic スイッチバックの違い

syslog によるログの記録

リソース階層

# 5.4.1.3.1. 共通のハードウェアコンポーネント

LifeKeeper のすべての設定には、以下の共通コンポーネントが含まれます。

1. **サーバグループ** 。LifeKeeper が提供する障害回復機能は、2 台以上のサーバをクラスタにグループ化することを基礎にしています。サーバは、サポートする Linux のディストリビューションを実行するサポートするプラットフォームであれば、いずれでもかまいません。LifeKeeper には、複数の重なり合うグループにサーバを設定する柔軟性があります。ただし、リカバリ可能なリソースについての重要な要件は、リソースの役割と優先順位が定義されたサーバのグループをリンクすることです。リソースに対するサーバの優先順位は、現在実行中のサーバに障害が発生した場合に、どのサーバがそのリソースを復旧するかを決定するために使用されます。最高の優先順位を示す値は 1 です。特定のリソースについて、最高の優先順位の値 (通常は 1) を持つサーバが通常、そのリソースのプライマリサーバと呼ばれます。その他のサーバは、そのリソースのバックアップサーバとして定義されます。

2. **コミュニケーションパス** 。LifeKeeper のハートビートは、LifeKeeper クラスタ内にあるサーバ間の定期的なメッセージで、主要な障害検出機能です。クラスタ内のすべてのサーバには、単純な通信障害でシステムに障害が発生しないように、冗長なハートビートコミュニケーションパス (comm パス) が必要です。2 つの独立したサブネットを使用する LAN ベース (TCP) の個別な 2 つのコミュニケーションパスが推奨されます (少なくとも 1 つのコミュニケーションパスをプライベートネットワークとして設定してください)。ただし、TCP と TTY のコミュニケーションパスの組み合わせの使用もサポートしています。TCP コミュニケーションパスは、他のシステムの通信にも使用できます。

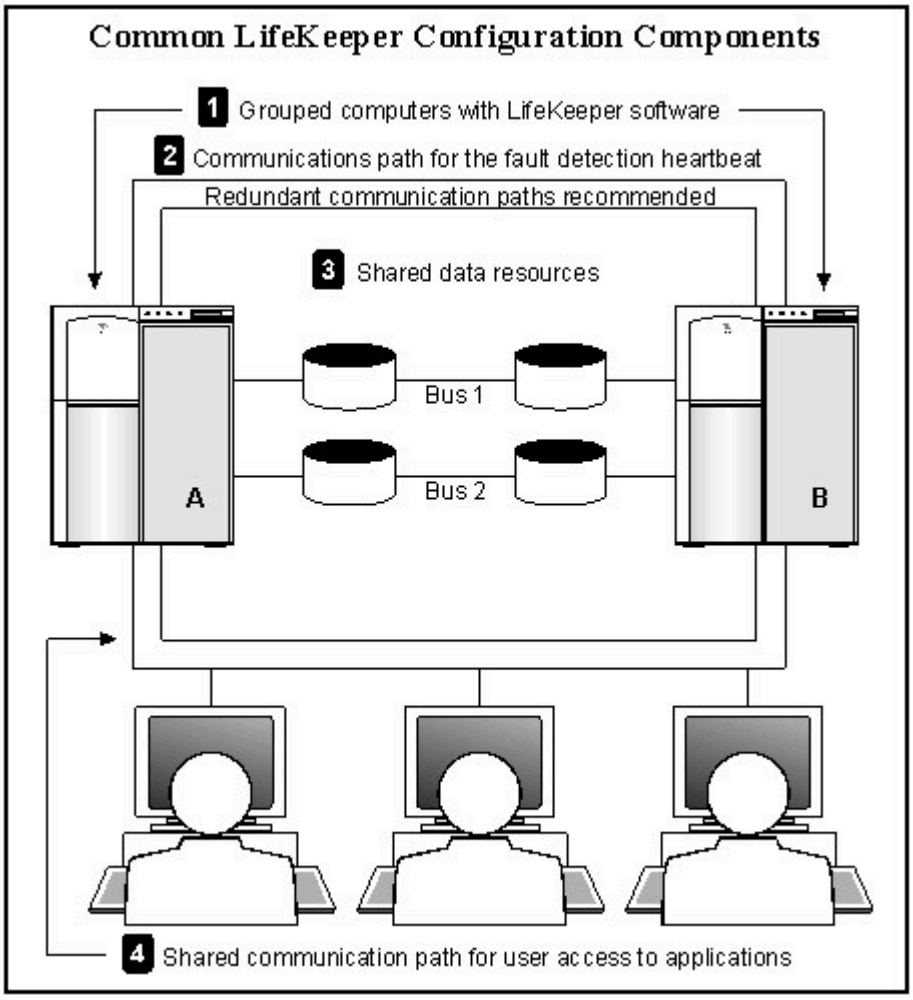
> **注意: クラウド環境** では一般にネットワークの内部構成が公開されていないので、物理的に別経路の LAN 回線を 2 つ用意することは困難です。しかしながら、基本的にクラウド側で物理的なネットワークの冗長化が行われている事が予想されるため、 **コミュニケーションパスが一つ** であっても動作上の信頼性を確保することができます。

**注意:** TTY コミュニケーションパスは、クラスタ内の他のサーバがアクティブかどうかを検出するためにのみ LifeKeeper で使用されます。LifeKeeper の GUI は、TCP/IP を使用して、保護するリソースに関するステータス情報を通信します。TCP コミュニケーションパスが 2 つ設定されている場合、LifeKeeper はパブリックネットワークのコミュニケーションパスをリソースステータスの通信に使用します。このため、LifeKeeper の GUI が使用しているネットワークがダウンすると、TTY (または他の TCP) コミュニケーションパスが動作可能な場合でも、GUI には他のサーバのステータスが UNKNOWN として表示されます。

3. **共有データリソース** 。共有ストレージの構成では、LifeKeeper クラスタ内のサーバは同一セットのディスクに対するアクセスを共有します。プライマリサーバに障害が発生した場合、LifeKeeper は障害が発生したサーバ上にあるディスクのロック解除、および次に使用可能なバックアップサーバのディスクのロックを自動管理します。

4. **共有通信** 。LifeKeeper は TCP/IP アドレスのような通信リソースの切り替えを自動管理できるので、アプリケーションが現在どのサーバでアクティブになっているかには無関係に、ユーザはアプリケーションに接続できます。

# すべての LifeKeeper 設定に共通するコンポーネント



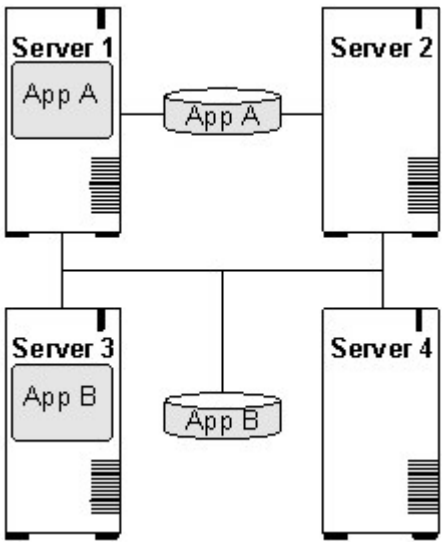
# 5.4.1.3.2. システムのグループ化の配置

リソース階層は、LifeKeeper サーバのクラスタに対して定義されます。ある階層について、各サーバに優先順位が割り当てられます。1 が最高の優先順位です。プライマリ、つまり優先順位が最高のサーバが、それらのリソースの通常動作に使用するコンピュータです。2 番目に高い優先順位を持つサーバがバックアップサーバであり、プライマリサーバに障害が発生した場合に、LifeKeeper がリソースを切り替える先のサーバです。

アクティブ / アクティブのグループ では、すべてのサーバがプロセスをアクティブに実行します。ただし、他のサーバのリソース階層ではバックアップサーバとしても機能します。アクティブ / スタンバイのグループ では、プライマリサーバは処理を実行し、バックアップサーバはプライマリサーバに障害が発生した場合に備えてスタンバイするように設定できます。スタンバイシステムは小型でパフォーマンスの低いシステムでもかまいませんが、プライマリサーバに障害が発生した場合にリソースの可用性を確保できるだけの処理能力が必要です。

共有リソースに対する物理的な接続とアクセスにより、グループ化のオプションが決まります。グループ化するサーバには、通信とハートビートパスがインストールされ、動作可能である必要があり、すべてのサーバが共有 SCSI またはファイバチャネルインターフェース経由で、ディスクリソースにアクセスできる必要があります。例えば、以下の図では、サーバ 1 のリソース *AppA* にはグループ化オプションが 1 つのみあります。この構成で *AppA* データベースへの共有アクセスを持つ他のサーバはサーバ 2 のみです。

ただし、サーバ 3 のリソース *AppB* は、その他 3 台のいずれを含むグループにも属するように設定できます。これは、この例の共有 SCSI バスが、構成内の 4 台すべてのサーバに *AppB* データベースへのアクセスを提供しているからです。



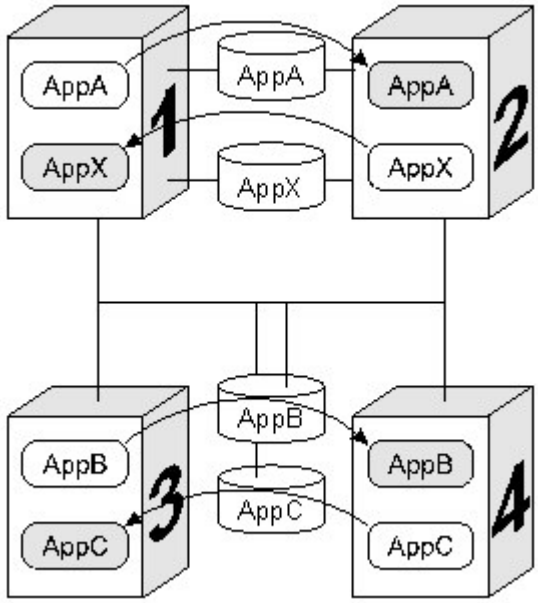
# 5.4.1.3.3. アクティブ - アクティブのグループ化

アクティブ / アクティブペアの設定では、すべてのサーバがプロセスをアクティブに実行します。また、他のサーバのリソース階層ではバックアップサーバとして機能します。

以下の設定例に、2 つのアクティブ / アクティブペアのサーバを示します。サーバ 1 は *AppA* を処理していますが、サーバ 2 で実行中の *AppX* のバックアップサーバとして機能します。この逆も当てはまります。サーバ 2 は *AppX* を処理していますが、サーバ 1 で実行中の *AppA* のバックアップサーバとして機能します。サーバ 3 とサーバ 4 の間には、同じタイプのアクティブ/アクティブの関係があります。

サーバ 1 とサーバ 2 の設定と、サーバ 3 とサーバ 4 の設定は似ていますが、大きな違いがあります。*AppA* と *AppX* のアプリケーションについて、サーバ 1 とサーバ 2 のみをグループ化できます。これらのサーバのみが、共有リソースにアクセスできます。

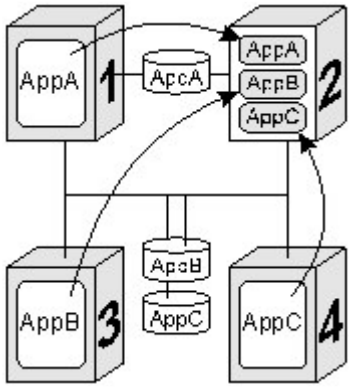
ただし、 *AppB* と *AppC* は、複数のグループ化オプションを持ちます。これは、4 台のサーバすべてが *AppB* と *AppC* の共有リソースにアクセスできるからです。 *AppB* と *AppC* は、第 3、第 4 のバックアップシステムとしてサーバ 1 や サーバ 2 にフェイルオーバするように設定することもできます。




**注記:** LifeKeeper はディスクレベルでロックを適用するので、 *AppB* と *AppC* のディスクリソースに接続する 4 つのシステムのうち、任意の時点でそれらにアクセスできるのは 1 つのみです。このため、サーバ 3 がアクティブに *AppB* を処理しているときには、サーバ 1、サーバ 2、および サーバ 4　は物理的に接続していても *AppB* のディスクリソースを使用できません。

# 5.4.1.3.4. アクティブ - スタンバイのグループ化

アクティブ/スタンバイのペア設定では、プライマリサーバは処理を実行し、バックアップサーバはプライマリサーバに障害が発生した場合に備えてスタンバイします。スタンバイシステムは小型でパフォーマンスの低いシステムでもかまいませんが、プライマリサーバに障害が発生した場合にリソースの可用性を確保できるだけの処理能力が必要です。




スタンバイサーバは、複数のアクティブサーバにバックアップを提供します。例えば、上の図では、3 つのアクティブ / スタンバイのリソースペアでサーバ 2 がスタンバイサーバです。LifeKeeper のリソース定義が、以下のアクティブ / スタンバイのペアの関係を指定します。

- サーバ *1* の *AppA* が サーバ *2* にフェイルオーバする。
- サーバ *3* の *AppB* が サーバ *2* にフェイルオーバする。
- サーバ *4* の *AppC* が サーバ *2* にフェイルオーバする。

複数のアクティブ / スタンバイグループを持つ設定を検討するときには、以下の 3 つの重要な設定概念を念頭に置いてください。

- **ディスクの所有権** 。複数の異なるアクティブなアプリケーションは、異なる複数のサーバから、同じ共有ディスクまた LUN にあるディスクパーティションを使用できません。LifeKeeper は、ディスクまたは LUN のレベルでロックを適用します。SCSI ロックが適用された場合、共有 SCSI バス上にあるシステム 1 台のみが、ディスクまたは LUN のパーティションにアクセスできます。このため、同一ディスク上の異なるパーティションにアクセスする複数のアプリケーションは、同一サーバ上でアクティブにする必要があります。この例では、サーバ 3 が *AppB* のディスクリソースを所有し、サーバ 4 が *AppC* のリソースを所有します。
- **処理能力** 。サーバ 1、サーバ 3、およびサーバ 4 に同時に障害が発生する可能性は非常に低いですが、複数のリソース関係をサポートするスタンバイサーバを指定するときには、複数の障害が発生した場合にスタンバイサーバが重要な処理のすべてを処理できるように注意する必要があります。
- **LifeKeeper の管理** 。この例では、サーバ 2 がその他 3 台のサーバをバックアップします。一般的に、LifeKeeper のデータベースを複数の論理グループで同時に管理することは望ましくありません。はじめに、予備システムと 1台のアクティブなシステムとの間でリソースを作成し、次に予備

システムと別のアクティブなシステムとの間、という手順を繰り返してリソースを作成する必要があります。

# 5.4.1.3.5. Intelligent スイッチバックと Automatic スイッチバックの違い

デフォルトでは、リソースのスイッチバック設定は *Intelligent* です。これは、そのリソースについて サーバ *A* から サーバ *B* にフェイルオーバが発生すると、別の障害が発生するか、管理者がリソースを別のサーバに *Intelligent* に 切り替えるまで、リソースは サーバ *B* に残ります。このため、 サーバ *A* が In Service に戻った後も、リソースは サーバ *B* で動作を続行します。この時点では、 サーバ *A* はリソースのバックアップとして機能します。

状況によっては、障害が発生したサーバが復旧したときに、リソースをそのサーバに自動でスイッチバックすることが望ましい場合があります。LifeKeeper には、前述したデフォルトの *Intelligent* スイッチバック 動作に代わる選択肢として、 *Automatic* スイッチバック オプションがあります。このオプションは、各サーバの個々のリソース階層に設定できます。特定のサーバ上にあるリソース階層に *Automatic* スイッチバック を選択し、そのサーバに障害が発生した場合、そのリソース階層はバックアップシステムにフェイルオーバします。障害が発生したサーバが復旧したときに、リソース階層は元のサーバに自動的にスイッチバックします。

**注記:**

- Automatic の場合、プライマリサーバがオンラインに戻り、LifeKeeper コミュニケーションパスを再確立した直後に自動的にスイッチバックが行われます。
- LifeKeeper は、優先順位が上位のサーバから下位のサーバへの *Automatic* スイッチバック を実行しません。
- 一部のリソース (DataKeeper や HANA など) は、Automatic スイッチバックをサポートしていません。 これらのリソースが階層内にある場合、階層はスイッチバックしません。 これらの階層は、レプリケーションが同期された後に手動で元に戻す必要があります。

# 5.4.1.3.6. syslog によるログの記録

LifeKeeper 8.0 から、標準の syslog 機能を使用してログの記録が行われます。LifeKeeper では、syslog の実装の1つである rsyslog をサポートしています。パッケージのインストール時には、すべての LifeKeeper ログメッセージに対して「local6」機能を使用するように syslog が設定されます（もし、local6 が既に使用されている場合は、他の local を使用します）。すべての LifeKeeper ログメッセージを */var/log/lifekeeper.log* に送信する LifeKeeper 固有のルーティングを含むように、 syslog 設定ファイル */etc/rsyslog.conf* が変更されます。(元の設定ファイルは、「~」で終わる同じ名前を使用してバックアップされます。)

> **重要:** LifeKeeper固有の設定を手動で編集しないでください。アップグレードやアンインストールが正しく機能しないことがあります。

この機能は、インストール後に /opt/LifeKeeper/bin にある lklogconfig ツールを使用して変更することができます。 例えば facility を local5 に変更する場合は、以下のコマンドを実行してください。

```
lkstop -f
lklogconfig --action=update --facility=local5
lkstart
```

このツールの詳細については、LifeKeeper がインストールされているシステム上の lklogconfig(8) マニュアルページを参照してください。

Generic Applicationのリソーススクリプトが /opt/LifeKeeper/out/log に直接メッセージを送ると、LifeKeeperはデフォルトで */var/log/lifekeeper.log* にERRORレベルのログメッセージを送ります。 */etc/default/LifeKeeper* に **LOGMGR_LOGLEVEL=LK_INFO** パラメータを追加すれば、ERRORレベルをINFOレベルに変更することができます。

> **注意:** LifeKeeper がサーバから削除されると、LifeKeeper 固有の syslog 設定が削除されます。

> **注記:** ログローテンションをコントロールする LifeKeeper 設定ファイルは /etc/logrotate.d にあり、lifekeeper.log ファイルについては /etc/logrotate.d/lifekeeper 、 lifekeeper.err ファイルについては /etc/logrotate.d/lifekeeper-err で確認することができます。デフォルトの設定ではサイズが 100MB を越えるとローテーションされます。

# 5.4.1.3.7. リソース階層

LifeKeeper の GUI を使用すると、あるサーバにリソース階層を作成し、次にその階層を 1 台以上のバックアップサーバに拡張できます。その後、LifeKeeper により、指定したすべてのサーバに指定階層が自動作成されます。LifeKeeper は、各サーバのデータベースで階層情報を管理します。コマンドラインインターフェースを使用する場合は、各サーバの階層を明示的に指定する必要があります。

リソース階層の作成後、LifeKeeper が階層内のリソースの停止と開始を管理します。以下の関連トピックで、階層の指定作業の基本情報を説明しています。

---

リソースタイプ

リソースの状態

階層の関係

イクイバレンシ情報

リソース階層の情報

リソース階層の例

ステータスの詳細表示

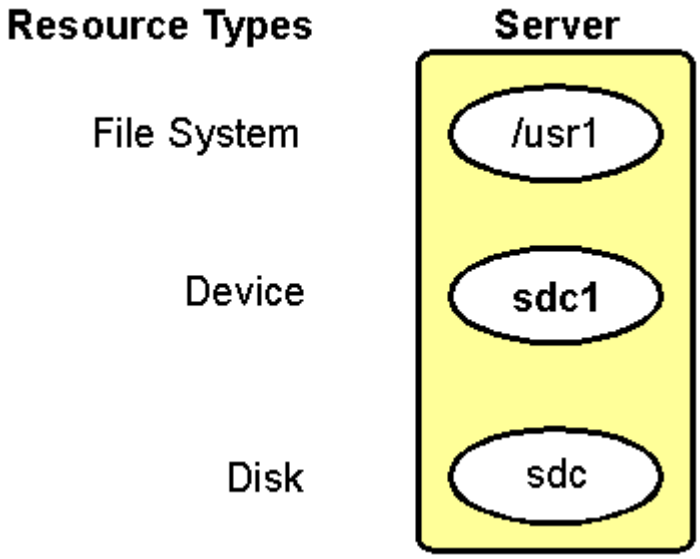
ステータスの簡略表示

# 5.4.1.3.7.1. リソースタイプ

リソースはハードウェアとソフトウェアのいずれかであり、リソースタイプ別に分類できます。LifeKeeper はファイルシステムと SCSI のリソースタイプに処理を提供し、リカバリキットは通信、RDBMS、その他のアプリケーションのリソースタイプに処理を提供します。

例えば、保護するファイルシステムの階層には、以下のタイプのリソースインスタンスが含まれます。

- **filesys** - Linux のファイルシステムリソースオブジェクトで、マウントポイントにより識別されます。
- **device** - SCSI ディスクパーティションと仮想ディスクで、デバイスファイル名で識別されます (例: *sdc1*)。
- **disk** - SCSI ディスクまたは RAID システム論理ユニットで、SCSI デバイス名で識別されます (例: *sd*)。

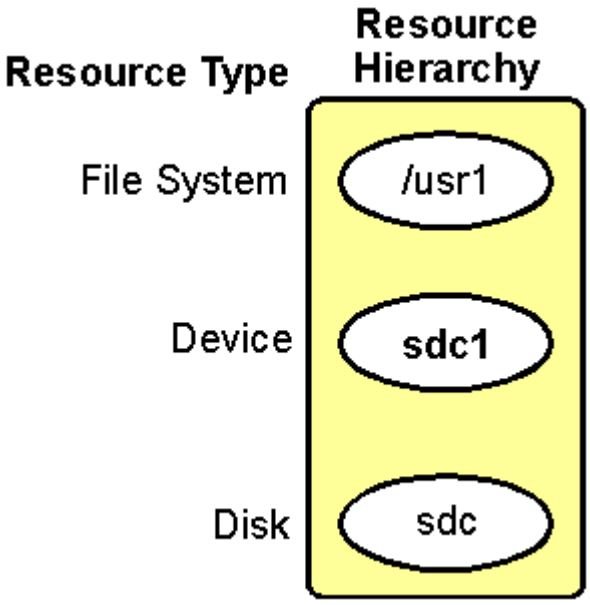


# 5.4.1.3.7.2. リソースの状態

| 状態 | 意味 |
|---|---|
| In Service、保護 (ISP) | リソースが動作可能です。LifeKeeper のローカルリカバリが正常に動作しています。LifeKeeper のサーバ間リカバリと障害検出が動作可能です。 |
| In Service、未保護 (ISU) | リソースは動作可能です。しかし、LifeKeeperのリソースヘルス監視は無効の状態であるため、ローカルリカバリーやフェイルオーバーは行われません。<br><br>**注記:** ファイルシステムリソース(filesys)で保護されたファイルシステムの使用容量が90%に（閾値のデフォルト）達した時、リソースステータスをISUにすることでそれを通知します。この場合、監視処理自体は継続されます。ファイルシステムリソースと使用容量の監視では、他のリソースタイプとは違って意味合いでISUステータスを使用しています。ファイルシステムの使用容量が一度閾値を下回れば、リソースステータスはISPに戻ります。 |
| Out of Service、障害 (OSF) | リソースが、障害により Out of Service になっています。リカバリは完了していないか、失敗しました。このリソースについて、LifeKeeper の警告機能は動作不能です。 |
| Out of Service、障害なし (OSU) | リソースは Out of Service ですが、別のサーバからリソースを引き継ぐことができます。 |
| 不正 (未定義) 状態 (ILLSTATE) | この状態は、リソースインスタンスについて状態が設定されていない場合に表示されます。通常の状況では、この不正状態が長く続くことはありません。ある状態から別の状態への移行が予測されます。LifeKeeper の情報テーブルがすべて更新される前 (LifeKeeper が初めて起動するときなど) にスイッチオーバが発生した場合に、この状態になります。 |

# 5.4.1.3.7.3. 階層の関係

LifeKeeper では、リソースインスタンス間の関係を作成できます。主な関係は依存関係で、例えばあるリソースインスタンスが動作するために、別のリソースインスタンスに依存します。リソースインスタンスと依存関係の組み合わせが、リソース階層です。



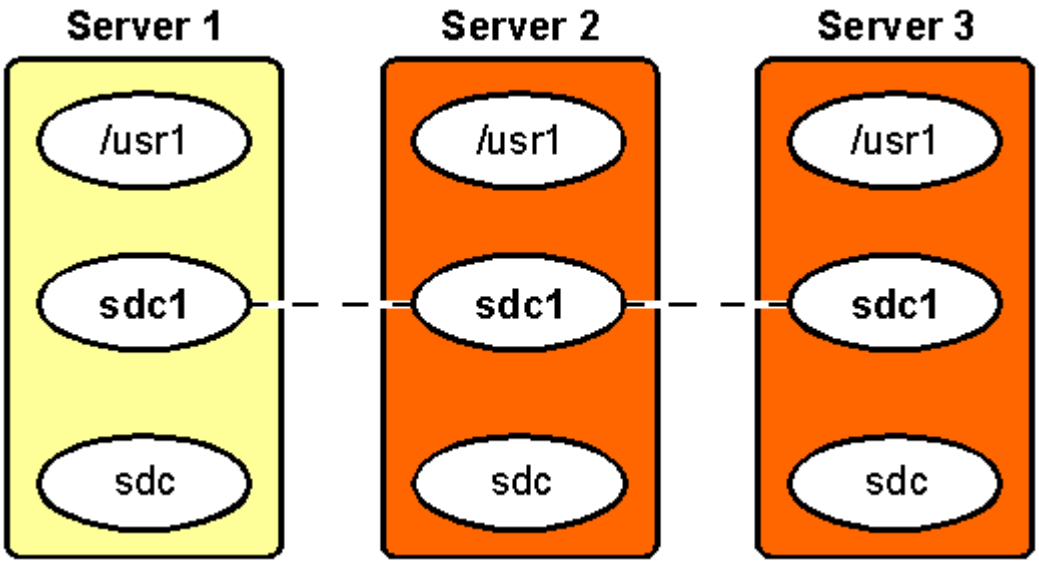

例えば、 */usr1* の動作はディスクサブシステムに依存するので、 */usr1* と、ディスクサブシステムを表すインスタンスとの間に順序付きの階層の関係を作成できます。

リソース階層により指定された依存関係は、リソースインスタンスを In Service と Out of Service にする適切な順序を LifeKeeper に示します。このリソース階層の例では、 *disk* と *device* のインスタンスを正常に In Service にするまで、LifeKeeper は */usr1* リソースを In Service にすることができません。

# 5.4.1.3.7.4. イクイバレンシ情報

LifeKeeper リソース階層を作成して拡張すると、そのリソース階層はプライマリサーバとセカンダリサーバの両方に存在します。ほとんどのリソースインスタンスは、1 台のサーバでのみ同時にアクティブにできます。このようなリソースについて、LifeKeeper は「イクイバレンシ情報」という第 2 の種類の関係を定義します。これにより、リソースがあるサーバで *In Service* になると、イクイバレンシ情報が定義されている他のサーバでは *Out of Service* になります。

以下の例に、各サーバのディスクパーティションのリソースインスタンス間のイクイバレンシ情報を示します。この例では、各リソースインスタンスが類似のイクイバレンシを持ちます。



# 5.4.1.3.7.5. リソース階層の情報

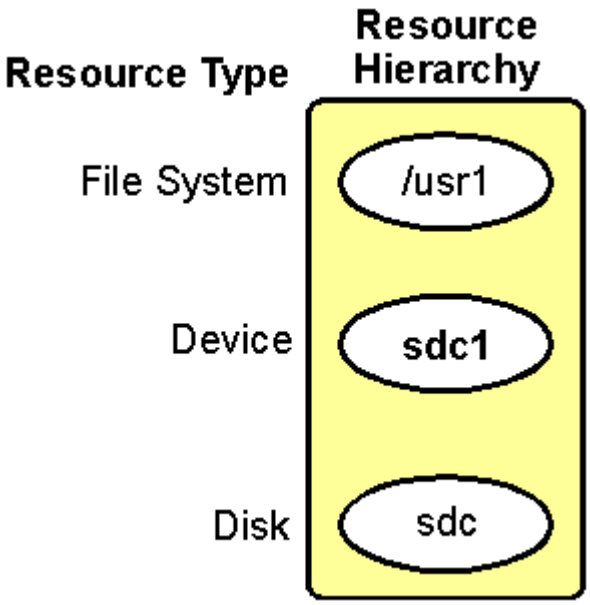
各リソースのステータスは、ステータスの詳細表示 とステータスの簡略表示 で表示されます。root リソースを表す LifeKeeper のタグ名は、[TAG] 列の左端から開始され、階層内のリソースのタグ名は適切にインデントされてリソース間の依存関係を表します。

以下の例は、ステータスの簡略表示のリソース階層セクションから取ったものです（デバイスとディスクの ID は、表示領域に収まるように切り詰められている）。

| LOCAL | TAG | ID | STATE | PRIO | PRIMARY |
|---|---|---|---|---|---|
| svr1 | app3910-on-svr1 | app4238 | ISP | 1 | svr2 |
| svr1 | filesys4083 | /jrl1 | ISP | 1 | svr2 |
| svr1 | device2126 | 000…300-1 | ISP | 1 | svr2 |
| svr1 | disk2083 | 000…300 | ISP | 1 | svr2 |

階層の図についてはリソース階層の例 のトピックを参照してください。詳細については、ステータスの詳細表示 とステータスの簡略表示 のトピックの「リソース階層の情報」セクションを参照してください。

# 5.4.1.3.7.6. リソース階層の例



# 5.4.1.3.7.7. ステータスの詳細表示

このトピックでは、 **lcdstatus** コマンドの出力例を使用してステータスの詳細表示で提供される情報のカテゴリについて説明します。この情報を表示する方法の詳細については、LCD (1M) のマニュアルページを参照してください。コマンドラインに、 **man lcdstatus** または **man LCD** を入力できます。LifeKeeper の GUI で使用できるステータス情報については、サーバーのステータスの表示 またはリソースのステータスの表示 を参照してください。

ステータスの詳細表示の例:

リソース階層の情報

```
Resource hierarchies for machine “wileecoyote”:

ROOT of RESOURCE HIERARCHY

apache-home.fred: id=apache-home.fred app=webserver type=apache
state=ISP

initialize=(AUTORES_ISP) automatic restore to IN-SERVICE by LifeKeeper

info=/home/fred /usr/sbin/httpd

reason=restore action has succeeded

depends on resources:
ipeth0-172.17.104.25,ipeth0-172.17.106.10,ipeth0-172.17.106.105

Local priority = 1

SHARED equivalency with “apache-home.fred” on “roadrunner”, priority =
10

FAILOVER ALLOWED

ipeth0-172.17.104.25: id=IP-172.17.104.25 app=comm type=ip state=ISP

initialize=(AUTORES_ISP) automatic restore to IN-SERVICE by LifeKeeper

info=wileecoyote eth0 172.17.104.25 fffffc00

reason=restore action has succeeded

these resources are dependent: apache-home.fred
```

```
Local priority = 1

SHARED equivalency with “ipeth0-172.17.104.25” on “roadrunner”,
priority = 10

FAILOVER ALLOWED

ipeth0-172.17.106.10: id=IP-172.17.106.10 app=comm type=ip state=ISP

initialize=(AUTORES_ISP) automatic restore to IN-SERVICE by LifeKeeper

info=wileecoyote eth0 172.17.106.10 fffffc00

reason=restore action has succeeded

these resources are dependent: apache-home.fred

Local priority = 1

SHARED equivalency with “ipeth0-172.17.106.10” on “roadrunner”,
priority = 10

FAILOVER ALLOWED

ipeth0-172.17.106.105: id=IP-172.17.106.105 app=comm type=ip state=ISP

initialize=(AUTORES_ISP) automatic restore to IN-SERVICE by LifeKeeper

info=wileecoyote eth0 172.17.106.105 fffffc00

reason=restore action has succeeded

These resources are dependent: apache-home.fred

Local priority = 1

SHARED equivalency with “ipeth0-172.17.106.105” on “roadrunner”,
priority = 10

FAILOVER ALLOWED
```

通信ステータスの情報

```
The following LifeKeeper servers are known:

machine=wileecoyote state=ALIVE

machine=roadrunner state=DEAD (eventslcm detected failure at Wed Jun 7
15:45:14 EDT 2000)

The following LifeKeeper network connections exist:

to machine=roadrunner type=TCP addresses=192.168.1.1/192.168.105.19

state="DEAD" priority=2 #comm_downs=0
```

LifeKeeper のフラグ

```
The following LifeKeeper flags are on:

shutdown_switchover
```

シャットダウンストラテジー

```
The shutdown strategy is set to: switchover.
```

# リソース階層の情報

LifeKeeper は、リソースのステータスを root リソースから表示します。表示には、リソースのすべての依存関係についての情報が含まれます。

複数のリソースに共通する要素は、最初の root リソースの下に 1 回のみ表示されます。各リソース記述の第 1 行には、リソースタグとその後に続くコロン (:) が表示されます (例: `device13557:`)。階層内でリソースの記述に使用できる情報要素を以下に示します。

- **id** - LifeKeeper が使用する一意のリソース識別文字列。
- **app** - アプリケーションのタイプを示します。例えば、サンプルリソースは *Web* サーバアプリケーションです。
- **type** - リソースのクラスタイプを示します。例えば、サンプルリソースは *Apache* アプリケーションです。
- **state** - リソースの現在の状態。
  - ISP — ローカルで In Service であり、保護されています。

    - ISU — In Service であり、保護されていません。
    - OSF — Out of Service であり、障害が発生しています。
    - OSU — Out of Service であり、障害はありません。

- **initialize** - リソースの初期化方法を指定します。例えば、LifeKeeper はアプリケーションのリソースをリストアしますが、ホストアダプタは LifeKeeper なしで初期化します。

- **info** - オブジェクトの  remove と restore のスクリプトが使用する、オブジェクトに固有の情報があります。

- **reason** - 存在する場合、リソースが現在の状態にある原因を示します。例えば、あるアプリケーションが OSU の状態になった原因は、別のサーバでそのアプリケーションがIn Service (ISP または ISU) になったからです。共有リソースは、グループ内の 1 台のサーバでのみ同時にアクティブにできます。

- **depends on resources -** 存在する場合、このリソースが依存するリソースのタグ名がリストされます。

- **these resources are dependent -** このオブジェクトに直接依存するすべての親リソースのタグ名が示されます。

- **Local priority** - このリソースについて、ターゲットサーバのフェイルオーバの優先順位の値を示します。

- **SHARED equivalency** - このリソースが同等として定義されたリモートリソースのリソースタグとサーバ名、およびこのリソースについてのフェイルオーバの優先順位の値を示します。

- **FAILOVER ALLOWED** - 存在する場合、上の行で同等と指定されたリモートサーバで LifeKeeper が動作可能であること、およびアプリケーションが障害に対して保護されていることを示します。`FAILOVER INHIBITED` は、LifeKeeper がシャットダウンされているかリモートサーバが停止していることにより、アプリケーションが保護されていないことを示します。

## 通信ステータスの情報

ステータス表示のこのセクションには、LifeKeeper が認識しているサーバとその現在の状態、および各コミュニケーションパスの情報がリストされます。

これらの通信情報の要素は、ステータス表示にあります。

- **state** -コミュニケーションパスのステータス。通信ステータスの値は以下の値をとります。
    - ALIVE - 通常の動作中。
    - DEAD - 通常の動作をしていません。

- **priority** -コミュニケーションパスに割り当てられた優先順位の値。この項目は TCP パスについてのみ表示されます。

- **#comm_downs** - ポートに障害が発生してフェイルオーバが発生した回数。パスの障害によりフェ

イルオーバが発生するのは、障害発生時に「ALIVE」のコミュニケーションパスが他にない場合のみです。

さらに、ステータス表示では、TTY コミュニケーションパスについてのみ維持されている以下の統計値を提供できます。

- **wrpid** - 個々の TTY コミュニケーションパスが、一意の読み取りプロセスと書き込みプロセスを持ちます。wrpid フィールドには、書き込みプロセスのプロセス ID があります。書き込みプロセスは、以下の 2 つの条件のうちいずれかが発生するまでスリープ状態です。
  - ハートビートタイマの期限が切れ、書き込みプロセスにメッセージを送信させる。
  - ローカルプロセスが、LifeKeeper のメンテナンスメッセージを他のサーバに送信するように書き込みプロセスに要求する。書き込みプロセスは、関連付けられた TTY ポートを使用して、メッセージを他のシステムの TTY ポート上にある読み取りプロセスに送信します。

- **rdpid** - 個々の TTY コミュニケーションパスが、一意の読み取りプロセスと書き込みプロセスを持ちます。rdpid フィールドには、読み取りプロセスのプロセス ID があります。読み取りプロセスは、以下の 2 つの条件のうちいずれかが発生するまでスリープ状態です。
  - ハートビートタイマの期限が切れ、定義済みのハートビート間隔が期限切れになったかどうかを読み取りプロセスが判断する必要がある場合。期限切れの場合、読み取りプロセスはコミュニケーションパスに DEAD 状態のマークを付けます。これにより、ALIVE とマークされた他のコミュニケーションパスがない場合はフェイルオーバイベントが開始されます。
  - リモートシステムの書き込みプロセスが LifeKeeper のメンテナンスメッセージを送信し、読み取りプロセスがメッセージの受信に必要なプロトコルを実行します。

- **#NAKs** - 書き込みプロセスが negative acknowledgment （NAK）を受信した回数。NAK メッセージは、他のシステム上にある読み取りプロセスが、書き込みプロセスが送信したメッセージを受け取らず、書き込みプロセスがメッセージパケットを再送信する必要があったことを意味します。#NAKs の統計値は、回線ノイズに起因して、長期間にわたって集計できます。ただし、急激に数値が増加した場合、通信サブシステムで診断手順を実行する必要があります。

- **#chksumerr** - サーバ間のチェックサムメッセージが一致しなかった回数。この統計値は、回線ノイズに起因して、長期間にわたって集計できます。ただし、急激に数値が増加した場合、通信サブシステムで診断手順を実行する必要があります。

- **#incmpltmes** - 受信メッセージパケットが予測サイズに一致しなかった回数。不一致の回数が多い場合、コミュニケーションパスに関連付けられたハードウェアポートで診断手順の実行が必要な可能性があります。

- **#noreply** - 書き込みプロセスが肯定応答の待機中にタイムアウトし、メッセージを再送信しなければならなかった回数。肯定応答がない場合、サーバの過負荷、またはサーバの障害を意味することがあります。

- **#pacresent** - 読み取りプロセスが同一パケットを受診した回数。これは、送信サーバの書き込みプロセスがタイムアウトし、同一メッセージを再送信する場合に発生することがあります。

- **#pacoutseq** - 読み取りプロセスが、順序が不正のパケットを受診した回数。このフィールドの値が大きい場合、メッセージパケットの脱落を示すことがあり、通信サブシステムで診断手順の実行が必要な可能性があります。

- **#maxretrys** - 特定のメッセージについて、再送信の最大回数を超えたときに増加する指標 (NAK と noreply のメッセージ)。#maxretrys フィールドの値が大きい場合、通信サブシステムで診断手順を実行する必要があります。

# LifeKeeper のフラグ

ステータスの詳細表示の後部近くに、システムのフラグセットがあります。共通タイプは、プロセスのロックが動作を完了するまで他のプロセスを確実に待機させるために使用する LCD のロックフラグです。LCD のロックの標準フォーマットは以下のとおりです。

```
!action!processID!time!machine:id.
```

一般的な LCD のロックフラグの例を示します。

- `!action!02833!701236710!server1:` ***filesys*** 。ファイルシステム階層を作成すると、このフォーマットでステータス表示にフラグが生成されます。 ***filesys*** の指定は、他のアプリケーションリソース階層では別のリソースタイプである場合も、一般的なアプリケーションやユーザ定義アプリケーションでは　app　である場合もあります。

- 他の代表的なフラグとして、`!nofailover!`*`machine`*、`!notarmode!`*`machine`*、`shutdown_switchover` などがあります。`!nofailover!`*`machine`*と `!notarmode!`*`machine`* のフラグは、LifeKeeper が作成と削除を行う内部の一時フラグで、サーバのフェイルオーバを制御します。`shutdown_switchover` フラグは、このサーバのシャットダウンストラテジーが ***switchover*** に設定されたことを示し、サーバのシャットダウンによりスイッチオーバが発生します。使用可能なフラグの詳細については、依存関係の作成方法については、LCDI-flag (1M) を参照してください。

# シャットダウンストラテジー

ステータスの詳細表示の最後の項目は、このシステム用に選択された LifeKeeper のシャットダウンストラテジーを示します。詳細については、サーバのシャットダウンストラテジーの設定 を参照してください。

# 5.4.1.3.7.8. ステータスの簡略表示

このトピックでは、 **lcdstatus -e** コマンドの出力例を使用して、ステータスの簡略表示で提供される情報のカテゴリについて説明します。この情報を表示する方法の詳細については、LCD (1M) のマニュアルページを参照してください。コマンドラインに、 **man lcdstatus** または **man LCD** を入力できます。LifeKeeper の GUI で使用できるステータス情報については、サーバーのステータスの表示 またはリソースのステータスの表示 を参照してください。

ステータスの簡略表示の例（lcdstatus -e コマンドの出力例）：

リソース階層の情報

```
BACKUP   TAG                 ID          STATE     PRIO  PRIMARY
svr1     appfs3910-on-svr1   appfs4238   ISP          1  svr2
svr1      filesys4083        /jrl1       ISP          1  svr2
svr1       device2126        000…300-1   ISP          1  svr2
svr1        disk2083         000…300     ISP          1  svr2
```

通信ステータスの情報

```
MACHINE  NETWORK ADDRESSES/DEVICE          STATE     PRIO
svr1     TCP     100.10.1.20/100.11.1.21   ALIVE        1
svr1     TTY     /dev/ttyS0                ALIVE       --
```

## リソース階層の情報

LifeKeeper は、各リソースのステータスを表示します。root リソースを表す LifeKeeper のタグ名は、[TAG] 列の左端から開始され、階層内のリソースのタグ名は適切にインデントされてリソース間の依存関係を表します。

**BACKUP** 列は、フェイルオーバの優先順序内で、このステータス表示の対象システムの次にあるシステムを示します。指定したリソースについて、ターゲットシステムが優先順位の最も低いシステムである場合、そのリソースの **BACKUP** 列にはダッシュ (------) が表示されます。

- **TAG 列** - リソースの root タグがあります。
- **ID 列** -各リソースの識別文字列があります。
- **STATE 列** -各リソースの現在の状態があります (リソースの状態 を参照)。
- **PRIO 列** -各リソースについて、ローカルサーバのフェイルオーバの優先順位の値があります。

- **PRIMARY** 列 -各リソースについて、優先順位が最高のサーバ名があります。

## 通信ステータスの情報

表示のこのセクションには、ターゲットシステムで定義された各コミュニケーションパスのリストがあります。各パスについて、以下の情報が表示されます。

- **MACHINE** - コミュニケーションパスのリモートサーバ名。
- **NETWORK** - コミュニケーションパスのタイプ (TCP または TTY)。
- **ADDRESSES/DEVICE** - コミュニケーションパスの IP アドレスまたはデバイス名のペア。
- **STATE** - コミュニケーションパスの状態 (ALIVE または DEAD)。
- **PRIO** - TCP パスの場合、パスに割り当てられた優先順位。TTY パスの場合、優先順位が割り当てられていないので、この列にはダッシュ (----) が表示されます。

# 5.4.1.4. 障害検出とリカバリのシナリオ

障害検出とリカバリーを実行するために、LifeKeeper のさまざまなコンポーネントがどのように連携しているかを調べるには、3つのタイプのリカバリーシナリオを説明する以下のトピックを参照してください。

---

IP ローカルリカバリー

リソースエラーのリカバリーシナリオ

サーバー障害のリカバリーシナリオ

# 5.4.1.4.1. IP ローカルリカバリー

## ローカルリカバリーのシナリオ

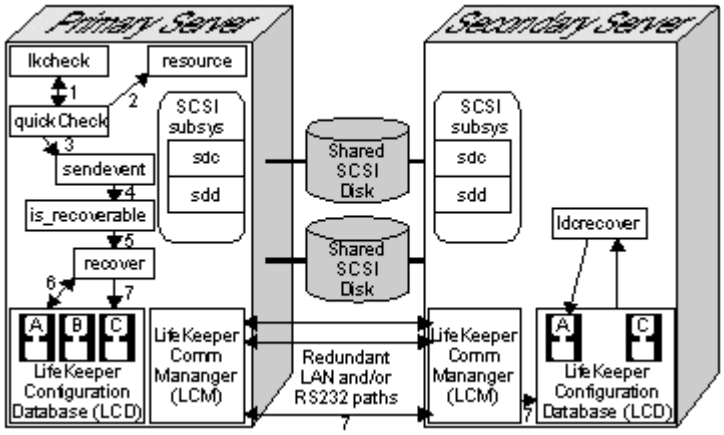
IP Recovery Kit によって IP アドレスの障害が検出されると、結果として生じる障害によって remove および restore を実行する **IP ローカルリカバリースクリプト** が起動されます。

> ✻ SIOS では、バックアップインターフェースが必要な場合、すべての LifeKeeper リリースに含まれる標準 Linux の NIC ボンディングメカニズムを使用してボンディングしたインターフェースを使用することを推奨しています。

# 5.4.1.4.2. リソースエラーのリカバリーシナリオ

LifeKeeper は、LifeKeeper が保護するリソースのステータスと健全性をチェックするリアルタイムデーモンモニター **lkcheck** を装備しています。 In Service の各リソースについて、 **lkcheck** が定期的にそのリソースタイプの quickCheck スクリプトを呼び出します。 **quickCheck** スクリプトがリソースのクイック健全性チェックを実行し、リソースが障害のある状態にあると判断すると、 **quickCheck** スクリプトはイベント通知メカニズム sendevent を呼び出します。

以下の図に、 **lkcheck** がプロセスを開始したときのリカバリープロセスの作業を示します。




1. **lkcheck** が実行されます。デフォルトでは、 **lkcheck** プロセスは 2 分ごとに実行されます。 **lkcheck** が動作すると、システムで In Service の各リソースについて適切は **quickCheck** スクリプトを呼び出します。

2. **quickCheck** スクリプトがリソースをチェックします。 **quickCheck** スクリプトが実行するチェックの内容は、各リソースタイプによって異なります。通常、スクリプトは、リソースのクライアントの動作をシミュレートして、予測した応答を受信するかどうかを確認することにより、目的の作業を実行するためにリソースが使用可能かどうかを単純に確認します。

3. **quickCheck** スクリプトが **sendevent** を呼び出します。 **quickCheck** スクリプトが、リソースが障害のある状態にあると判断した場合、 **sendevent** を呼び出して、適切なクラスとタイプを持つイベントを開始します。

4. リカバリー手順の検索。システムイベント通知メカニズム **sendevent** は、はじめに、イベントタイプまたはコンポーネントについて、LCD がリソースまたはリカバリーを持つかどうかを判断しようとします。この判断を行うために、is_recoverable プロセスは LCD のリソース階層をスキャンして、イベントに対応するリソースインスタンス (この例では filesys　の名前) を検索します。

   次の手順の動作は、スキャンでリソースレベルのリカバリー手順が検出されたかどうかによって異なります。

   - 検出されない場合：リカバリー手順が見つからない場合、is_recoverable は **sendevent** に戻り、 **sendevent** は基本イベント通知を続行します。

- 検出された場合：スキャンでリソースが検出された場合、is_recoverable はリカバリープロセスをバックグラウンドに運びます。is_recoverable プロセスが戻り、 **sendevent** が基本イベント通知を続行します。推奨フラグ「 *-A* 」を基本警告イベント応答スクリプトに渡し、LifeKeeper がリカバリーを実行することを示します。

5. リカバリープロセスが開始されます。リカバリーが続行していると仮定して、is_recoverable はリカバリープロセスを開始し、はじめにローカルリカバリーを試行します。

6. ローカルリカバリーが試行されます。インスタンスが検出された場合、リカバリープロセスは、LCD内のリソース階層にアクセスし、階層ツリーからイベントに応答する方法を知っているリソースを検索して、ローカルリカバリーを試行します。各リソースタイプについて、イベントクラスにちなむ名前を持つサブディレクトリ (そのイベントタイプのリカバリースクリプトを持つ) を含むリカバリーサブディレクトリを検索します。

   リカバリープロセスが、リソース階層で障害が発生しているリソースから上方向に最も離れたリソースに関連付けられている、リカバリースクリプトを実行します。リカバリースクリプトが正常に完了した場合、リカバリーは停止します。スクリプトが失敗した場合、次のリソースに関連付けられたスクリプトが実行され、リカバリースクリプトが正常に完了するか、障害が発生したインスタンスに関連付けられたリカバリースクリプトが試行されるまで続行されます。

   *ローカルリカバリーが正常に完了した場合、リカバリーは停止します。*

7. サーバー間のリカバリーが開始されます。ローカルリカバリーに失敗した場合、イベントはサーバー間のリカバリーにエスカレートします。

8. リカバリーが続行されます。ローカルリカバリーに失敗しているので、リカバリープロセスは失敗したインスタンスを *Out-of-Service-FAILED* (OSF) 状態にマークし、この失敗したリソースに依存するすべてのリソースを *Out-of-Service-UNIMPAIRED* (OSU) 状態にマークします。リカバリープロセスは次に、障害が発生したリソース、または障害が発生したリソースに依存するリソースが、他のシステム上にあるリソースとイクイバレンシー情報を持っているかどうかを判断し、優先順位が最高の動作可能なサーバーを選択します。同時にアクティブにできるイクイバレンシー情報を持つリソースは 1 つのみです。

   *イクイバレンシー情報が存在しない場合、リカバリープロセスは停止します。*

   イクイバレンシー情報が検出されて選択された場合、LifeKeeper はサーバー間のリカバリーを開始します。リカバリープロセスが LCM 経由で、イクイバレンシー情報を持つリソースを持つ選択されたバックアップシステムの LCD プロセスにメッセージを送信します。これは、LifeKeeper がサーバー間のリカバリーを試行することを意味します。

9. **lcdrecover** プロセスが転送を調整します。バックアップサーバーの LCD プロセスが **lcdrecover** プロセスを運び、同等リソースの転送を調整します。

10. バックアップサーバーのアクティブ化。 **lcdrecover** プロセスが同等のリソースを検出し、そのリソースが Out of Service のリソースに依存しているかどうかを判断します。 **lcdrecover** が、必要な各リソースについて restore スクリプト (リソースリカバリー動作スクリプトの一部) を実行し、リソースを in-service にします。

バックアップサーバーでリソースをリストアすることにより、プライマリーシステムからより多くの共有リソースを転送することが必要になる場合があります。プライマリーシステムとの間で、プライマリーサーバー上でのサービスから削除する必要があるリソースを示すメッセージが送受信され、次に選択したバックアップサーバーで in-service になり、重要なアプリケーションのすべての機能が提供されます。この動作は、転送する追加の共有リソースがなくなり、バックアップで必要なすべてのリソースインスタンスがリストアされるまで続行されます。

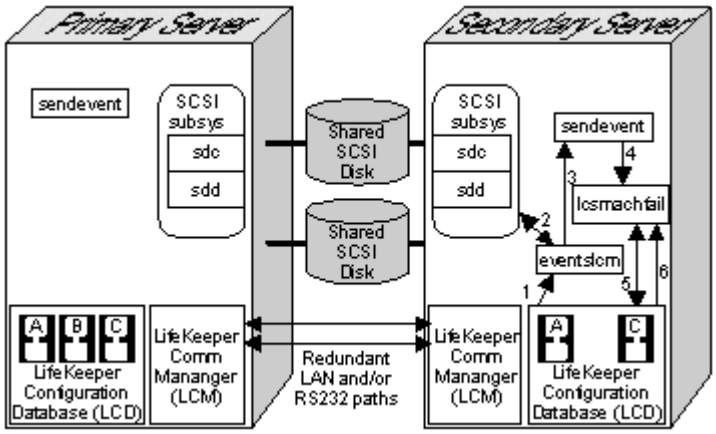
# 5.4.1.4.3. サーバー障害のリカバリーシナリオ

LifeKeeper Communications Manager (LCM) には、2 つの機能があります。

- メッセージング：LCM は、LifeKeeper がリカバリー、設定、または監査の実行を行うときに送信するメッセージの経路として機能します。
- 障害検出：LCM にはサーバーに障害が発生しているかどうかを検出する役割もあります。

LifeKeeper には、構成内の各サーバーに、ペアのサーバーが動作していることを定期的に通知する組み込みのハートビート信号があります。あるサーバーが、いずれかのコミュニケーションパス経由でハートビートメッセージを受信しなかった場合、LifeKeeper はそのパスを DEAD としてマークします。

以下の図に LCM ハートビートメカニズムがサーバーの障害を検出したときのリカバリー作業を示します。




以下の手順では、上の図で、LifeKeeper があるサーバのすべての通信接続を DEAD としてマークした場合のリカバリーシナリオを説明します。

1. LCM が **eventslcm** を起動します。LifeKeeper がすべてのコミュニケーションパスを DEAD としてマークすると、LCM は **eventslcm** プロセスを開始します。

   **eventslcm** プロセスを停止する活動は 1 つのみです。

   - ◦ コミュニケーションパスが、アクティブである。いずれかのコミュニケーションパスがハートビート信号の送信を再開した場合、LCM は **eventslcm** プロセスを停止します。

   通信障害に起因するフェイルオーバーやシステムの障害を防止するために、各ペアのサーバ間に、物理的に独立した冗長なコミュニケーションパスを 2 つ以上設定することが重要です。

2. sendevent へのメッセージ送信。 **eventslcm** は、 **sendevent** をイベントタイプ *machfail* で呼び出してシステム障害アラームを送信します。

3. sendevent がフェイルオーバリカバリを開始します。sendevent プログラムが、LifeKeeper がシステム障害イベントを処理できることを判断し、LifeKeeper フェイルオーバーリカバリープロセス **lcdmachfail** を実行します。

4. **lcdmachfail** のチェック。 **lcdmachfail** プロセスがはじめに、応答していないサーバがシャットダウンしていないことを確認します。システム障害の発生前に、他のシステムが正常にシャットダウンしている場合、フェイルオーバは禁止されます。次に、 **lcdmachfail** は、障害が発生したシステムと同等の共有を持つリソースをすべて特定します。これが、リカバリの関与ポイントです。

5. **lcdmachfail** がリソースをリストアします。 **lcdmachfail** が、障害が発生したシステムと同等のイクイバレンシ情報を持つ、バックアップサーバ上のリソースをすべて特定します。また、バックアップサーバが該当するリソースが構成されている、優先順位が最高のアクティブなサーバであるかどうかを判断します。すべてのバックアップサーバがこのチェックを実行するので、1 台のサーバのみが階層のリカバリを試行します。このチェックに合格した個々の同等リソースについて、 **lcdmachfail** が、関連付けられたリストアプログラムを呼び出します。次に、 **lcdmachfail** は、リストアしたリソースに依存する各リソースもリストアします。これは、バックアップサーバ上の階層全体がサービス中になるまで続行されます。

# 5.4.2. インストールと設定

## LifeKeeper for Linux のインストール

LifeKeeper for Linux ソフトウェアのインストールに関する詳細な手順については、LifeKeeper for Linux インストールガイド を参照してください。その他の情報については、 LifeKeeper for Linux リリースノート を参照してください。

## LifeKeeper for Linux の設定

LifeKeeper 環境がインストールされると、クラスタ内の各サーバ上で LifeKeeper ソフトウェアを設定することができます。以下の **LifeKeeper 設定手順** トピックに記載の手順に従ってください。ここには、詳細情報を含む各トピックへのリンクが記載されています。

---

設定手順

SNMP による LifeKeeper イベント転送

LifeKeeper イベントメール通知

任意の設定作業

Linux の設定

データレプリケーションの設定

ネットワーク設定

アプリケーションの設定

ストレージとアダプタの設定

LifeKeeper I-O フェンシングの概要

リソースポリシー管理

認証情報の設定

# 5.4.2.1. LifeKeeper の設定手順

LifeKeeperインストールガイドの説明に従って LifeKeeper 環境をインストールした場合、クラスタ内の各サーバーで LifeKeeperソフトウェアを起動し、設定することができます。

詳細を説明するトピックへのリンクを含む以下の手順を実行してください。以下の手順は、クラスタ内の各サーバーで実行します。

1. 次のコマンドを root として実行して LifeKeeper を起動します。
   ```
   $LKROOT/bin/lkcli start
   ```

   このコマンドによって、管理対象のサーバー上のまだ起動していないすべての LifeKeeper デーモンプロセスを起動します。

   LifeKeeper の起動および停止の詳細については、LifeKeeper の起動 および LifeKeeper の停止 を参照してください。

2. TTY 通信接続をセットアップ します。LifeKeeper のハートビート用に TTY 通信 パスを利用する場合は、ハートビート用の物理的な接続をセットアップする必要があります。

3. GUI を設定します。GUI の設定には多くのタスクが含まれます。GUI の実行準備 のLifeKeeper GUI - 概要 トピックから始めてください。詳細な手順については、GUIの実行準備 に網羅されたリンクの順番に従ってください。

   **注記:** LifeKeeper GUI を初めて実行すると、QuickStart ボタンが表示され、これを押すと LifeKeeper のリソースの設定を案内する手順とリンクを含むウィンドウが開きます。QuickStart Configuration Assistant は [Help] メニュー からいつでもアクセスできます。

4. コミュニケーションパスを作成 します。LifeKeeper の保護を有効にする前に、コミュニケーションパス (ハートビート) の定義を作成する必要があります。

5. 以下の設定作業を任意で実行します。

   - サーバーのシャットダウンストラテジーの設定
   - 手動フェイルオーバ確認オプションの設定
   - LifeKeeper ハートビートの調整
   - SNMP イベント転送の設定
   - イベントメール通知の設定
   - クラスタで STONITH デバイスを使用する場合は、STONITH デバイスを制御するスクリプトを作成し、LifeKeeper の適切なイベントディレクトリに配置します。

6. LifeKeeperでアプリケーションを保護する準備ができました。以降の手順は、オプションの

Recovery Kit のいずれかを使用するかどうかによって異なります。

- Recovery Kit を使用する場合、リソース階層の作成と拡張の手順については Recovery Kit に関連するドキュメンテーションを参照してください。
- 関連する Recovery Kit がないアプリケーションを使用する場合、2 通りの選択肢があります。
    - シンプルなアプリケーションの場合、アプリケーションと LifeKeeper との間のインターフェースの作成方法を慎重に検討してください。LifeKeeper Core に含まれる Generic Application Recovery Kit を使用して保護することもできます。
    - OS が提供しているサービスの場合、LifeKeeper Core に含まれる Quick Service Protection (QSP) Recovery Kit を使用して容易にリソースが作成できます。ただし、quickCheck では簡易なチェックしかしていませんので、注意が必要です。

# 5.4.2.1.1. TTY 接続のセットアップ

LifeKeeper のハートビート用に TTY コミュニケーションパスを利用する場合は、ハートビート用の物理的な接続をセットアップする必要があります。単一の通信障害による誤ったフェイルオーバを抑止するためには、複数のコミュニケーションパスが必要です。2 つ以上の LAN ベース (TCP) のコミュニケーションパスも使用する必要があります。

使用する各サーバのシリアルポートにシリアルハートビート用の TTY ケーブルを接続します。

1. 次のコマンドを実行してシリアルパスをテストします。

   **/opt/LifeKeeper/bin/portio -r -p port -b baud**

   ここで、

   - ◦ **baud** は、シリアルパス用に選択したボーレート (通常は 9600)
   - ◦ **port** は、サーバ 1 でテスト中のシリアルポート。例えば、 */dev/ttyS0*
     - これでサーバ 1 は、サーバ 2 からの入力を待っている状態です。

2. サーバ 2 で **portio** コマンドを実行します。ペアの 2 番目のシステムで次のコマンドを実行します。

   **echo Helloworld | /opt/LifeKeeper/bin/portio -p port -b baud**

   ここで、

   - ◦ **baud** は、サーバ 1 に合わせて選択した同じボーレート
   - ◦ **port** は、サーバ 2 でテスト中のシリアルポート。例えば、 */dev/ttyS0* 。

3. コンソールを確認します。コミュニケーションパスが正常に動作する場合、サーバ 1 のコンソールには「Helloworld」が表示されます。表示されない場合は、診断、修正作業を終えてから LifeKeeper の設定を続けてください。

# 5.4.2.2. SNMP による LifeKeeper イベント転送

SNMP による LifeKeeper イベント転送の概要

LifeKeeper イベント転送の設定

SNMP のトラブルシューティング

Recovery Kit固有のトラップについては、以下のドキュメントを参照してください。

DataKeeper イベントテーブル

EC2 イベントテーブル

# 5.4.2.2.1. SNMP による LifeKeeper イベント転送の概要

SNMP (Simple Network Management Protocol) は、ネットワークを管理するための、デバイスに依存しないフレームワークです。ネットワーク上のデバイスは、デバイスのベンダーが提供する MIB (Management Information Base) 変数によって記述されます。ネットワーク内の各ノード上では SNMP エージェントが実行され、ネットワークマネージャノードと通信を行います。ネットワークマネージャは、エージェントに対するクエリで MIB 変数の値を取得、設定することにより、エージェントノードを監視、制御します。エージェントは、トラップと呼ばれるメッセージを非同期に生成して例外イベントの発生をマネージャにしらせることもできます。SNMP (Simple Network Management Protocol) を使用してネットワークを監視および管理するアプリケーションは多数提供されています。

LifeKeeper のイベント通知機能では、特定のイベントが起きたときに通知を受信するアプリケーションを登録することができます (sendevent(5) マニュアルページを参照)。LifeKeeper は、LifeKeeper の動作を監視するサードパーティのネットワーク管理コンソールに向けて LifeKeeper の重要なイベントに関する SNMP トラップ通知を送信するように簡単に設定できます。

SNMP トラップを受信するリモート管理コンソールは、最初にそのシステムの管理用ソフトウェアを使用して設定する必要があります。LifeKeeper は、外部の SNMP の設定機能を提供していません。リモート管理サーバは、通常、LifeKeeper クラスタの外側に配置されます (つまり LifeKeeper のノードではありません)

## LifeKeeper イベントテーブル

以下の表では、LifeKeeper のイベントのリストと関連付けられているトラップ番号を示しています。オブジェクト ID (OID) は、プリフィックスとそれに続く個別のトラップ番号から、次のフォーマットで構成されます。

```
prefix.0.specific trap number
```

プリフィックスは「.1.3.6.1.4.1.7359」であり、これは MIB ツリーで **iso.org.dod.internet.private.enterprises.7359** に展開されます(7359 は、SteelEye (SIOS Technology) の企業番号です。LifeKeeper を表す「1」をこれに続けます)。例えば、「LifeKeeper Startup Complete」イベントは次の OID を生成します: **.1.3.6.1.4.1.7359.1.0.100** 。

| **LifeKeeper** イベント/説明 | トラップ番号 | オブジェクト **ID** |
|---|---|---|
| **LifeKeeper Startup Complete**<br><br>LifeKeeper が起動したノードから送信されます | 100 | .1.3.6.1.4.1.7359.1.0.100 |

| | | |
|---|---|---|
| **LifeKeeper Shutdown Initiated**<br><br>LifeKeeper のシャットダウンを開始したノードから送信されます | 101 | .1.3.6.1.4.1.7359.1.0.101 |
| **LifeKeeper Shutdown Complete**<br><br>LifeKeeper のシャットダウンを完了したノードから送信されます | 102 | .1.3.6.1.4.1.7359.1.0.102 |
| **LifeKeeper Manual Switchover Initiated on Server**<br><br>手動スイッチオーバを要求したノードから送信されます | 110 | .1.3.6.1.4.1.7359.1.0.110 |
| **LifeKeeper Manual Switchover Complete – recovered list**<br><br>手動スイッチオーバを完了したノードから送信されます | 111 | .1.3.6.1.4.1.7359.1.0.111 |
| **LifeKeeper Manual Switchover Complete – failed list**<br><br>手動スイッチオーバに失敗したクラスタ内の各ノードから送信されます | 112 | .1.3.6.1.4.1.7359.1.0.112 |
| **LifeKeeper Node Failure Detected for Server**<br><br>クラスタ内のノードに障害が発生したときに、クラスタ内の各ノードから送信されます | 120 | .1.3.6.1.4.1.7359.1.0.120 |
| **LifeKeeper Node Recovery Complete for Server – recovered list**<br><br>障害ノードからのリソースをリカバリしたクラスタ内の各ノードから送信されます | 121 | .1.3.6.1.4.1.7359.1.0.121 |
| **LifeKeeper Node Recovery Complete for Server – failed list**<br><br>障害ノードからのリソースのリカバリに失敗したクラスタ内の各ノードから送信されます | 122 | .1.3.6.1.4.1.7359.1.0.122 |
| **LifeKeeper Resource Recovery Initiated** | 130 | .1.3.6.1.4.1.7359.1.0.130 |

| | | |
|---|---|---|
| リソースをリカバリしているノードから送信されます。リカバリが完了したか失敗したかを示す 131 または 132 トラップが必ずこれに続きます。 | | |
| **LifeKeeper Resource Recovery Failed**<br><br>リカバリを試みたリソースを稼働できなかったときに、トラップ 130 を送信したノードから送信されます | 131* | .1.3.6.1.4.1.7359.1.0.131 |
| **LifeKeeper Resource Recovery Complete**<br><br>リソースのリカバリが完了したときに、トラップ 130 を送信したノードから送信されます | 132 | .1.3.6.1.4.1.7359.1.0.132 |
| **LifeKeeper Communications Path Up**<br><br>ノードへのコミュニケーションパスが確立されました | 140 | .1.3.6.1.4.1.7359.1.0.140 |
| **LifeKeeper Communications Path Down**<br><br>ノードへのコミュニケーションパスがダウンしました | 141 | .1.3.6.1.4.1.7359.1.0.141 |
| **LifeKeeper <Node monitoring> Failure**<br><br>Standby Node Health Check における Node監視で異常を検知したときに、異常を検知したノードから送信されます。<Node monitoring> には、検知した異常の内容が記載されます。 | 190 | .1.3.6.1.4.1.7359.1.0.190 |
| **LifeKeeper <OSUquickCheck> Failure**<br><br>Standby Node Health Check における OSUリソース監視で異常を検知したときに、異常を検知したノードから送信されます。<br><OSUquickCheck> には、異常が検知されたリソースのタグ名が記載されます。 | 200 | .1.3.6.1.4.1.7359.1.0.200 |
| **トラップ PDU に追加情報を含めるために以下の変数が使用されます。:** | | |
| Trap message | すべての | .1.3.6.1.4.1.7359.1.1 |

| | トラップ | |
|---|---|---|
| Resource Tag | 130 | .1.3.6.1.4.1.7359.1.2 |
| Resource Tag | 131 | .1.3.6.1.4.1.7359.1.2 |
| Resource Tag | 132 | .1.3.6.1.4.1.7359.1.2 |
| List of recovered resources | 111 | .1.3.6.1.4.1.7359.1.3 |
| List of recovered resources | 121 | .1.3.6.1.4.1.7359.1.3 |
| List of failed resources | 112 | .1.3.6.1.4.1.7359.1.4 |
| List of failed resources | 122 | .1.3.6.1.4.1.7359.1.4 |

* 複数のバックアップサーバでリカバリに失敗すると、このトラップは複数回表示されることがあります。

# 5.4.2.2.2. LifeKeeper イベント転送の設定

## 前提条件

SNMP のイベント転送機能は、LifeKeeper Core の機能の一部として含まれているので、LifeKeeper の追加パッケージをインストールする必要はありません。LifeKeeperはトラップを発行するためにsnmptrapコマンドを使用しているので、SNMP通知機能を使用するノードにsnmptrapコマンドをインストールしている必要があります。
snmptrapコマンドは以下のパッケージにて提供されています。
RHEL5以降および互換OS - net-snmp-utils
SLES11以降 - net-snmp

以前のバージョン (4.1 以前) の snmp の実装では、`defCommunity` ディレクティブがサポートされていないため、トラップは「public」コミュニティストリングを使用して送信されます。

LifeKeeper のノードで SNMP エージェント snmpd を起動しておく必要はありません。

ネットワーク管理コンソール上のトラップハンドラおよびトラップメッセージに対するハンドラの応答に関する設定は、LifeKeeper の本機能が提供する範囲ではありません。必要な手順については、お使いのシステム管理ツールが提供するドキュメンテーションを参照してください。

## 設定作業

LifeKeeper SNMP イベント転送を設定するには、以下の作業を実施します。SNMP トラップを生成する LifeKeeper クラスタの各ノードにおいて、最後の手順以外のすべてを 繰り返す必要があります。

1. 上述の `snmptrap` ユーティリティが利用できることを確認します。

2. SNMP トラップを受信するネットワーク管理ノードを指定します。指定するには、コマンドラインを使用するか、`/etc/default/LifeKeeper` ファイルを編集します。DNS の問題に影響されないように、ドメイン名ではなく IP アドレスを指定してください。

   - ◦ コマンドラインからは、`lk_configsnmp` を使用します (詳細については、`lk_configsnmp(1M)` のマニュアルページを参照)。このユーティリティでは、IP アドレスのみ使用できます。

   - ◦ または、デフォルトファイル`/etc/default/LifeKeeper` を編集して IP アドレスを追加します。`LK_TRAP_MGR=` エントリを見つけて「=」の右側に IP アドレスを１つまたは複数（カンマ区切り）入力します (「=」の前後やカンマの前後にはスペースを入れません)。

3. `defCommunity` をサポートとしない SNMP 実装の以前のバージョンをお使いの場合は、このステップを飛ばしてください。トラップは「public」コミュニティストリングを使用して送信されます。新しいバージョンの場合は次の手順を実行します。

   `/usr/share/snmp/snmp.conf` にデフォルトのコミュニティを指定してください。このファイルの保存場所は使用しているsnmpパッケージのバージョンに依存します。このファ

イルが存在しない場合は、十分な制限付きの権限で作成します。ディレクティブ defCommunity を値と共に追加します。これにより、トラップの送信時に SNMP バージョン 2c のコミュニティストリングが指定されます。例えば、以下のような行を追加します。

```
defCommunity myCommunityString
```

この設定ファイルの詳細については、snmp.conf マニュアルページを参照してください。

4. リモート管理コンソール上で、LifeKeeper のイベントから送られて来るトラップ OID を検出し応答するために必要な設定手順をすべて実行します。管理ノードが Linux サーバの場合、この機能の検証を開始するために最低限必要なことは、-f -Lo オプション (メッセージを stdout に出力) を指定して snmptrapd デーモンを開始することです。

## 設定の確認

設定が正常に動作することを確認するには、LifeKeeper の処理を実行します (例えば、LifeKeeper を開始または停止する、または LifeKeeper GUI を使用して、あるリソースを手動で in service にするなど)。管理コンソールでトラップメッセージを受信していることを確認します。トラップを受信していない場合は、管理システムの適切なログファイルを調査し、管理ソフトウェアが提供する標準のトラブルシューティング手順を実行してください。LifeKeeper のログを調べると、トラップメッセージの送信に問題があるかどうかを判断することができます。詳細については、SNMP のトラブルシューティング を参照してください。

## SNMP イベント転送の無効化

LifeKeeper による SNMP トラップの生成を無効にするには、ファイル /etc/default/LifeKeeper の LK_TRAP_MGR 環境変数から IP アドレスの割り当てを削除します。削除するには、コマンドラインで「disable」オプションを指定して lk_configsnmp ユーティリティを実行します (例については lk_configsnmp(1M) マニュアルページを参照)。または、/etc/default/LifeKeeper を編集して、LK_TRAP_MGR のエントリを LK_TRAP_MGR= に変更します (または行全体を削除する)。この手順は、トラップメッセージの送信を無効にしたい各ノードで実行する必要があります。

# 5.4.2.2.3. SNMP のトラブルシューティング

SNMP によるイベント転送に関連して予想される問題とその解決策を以下に説明します。具体的なエラーメッセージについては、 LifeKeeper メッセージカタログ を参照してください。

**問題:** LifeKeeper から SNMP のトラップメッセージが送信されない。

**解決策:** snmptrap ユーティリティがインストールされていることを確認します (通常は /bin/bin にあります)。インストールされていない場合は、適切な SNMP パッケージをインストールします (前提条件 を参照)。別の場所にインストールされている場合は、ファイル /etc/default/LifeKeeper の PATH 変数に適切なパスを追加します。

**問題:** SNMP のエラーメッセージがログに記録されない。LifeKeeper サーバから SNMP のトラップメッセージが送信されていないように見える。

**解決策:** トラップを受信するネットワーク管理サーバの IP アドレスが `LK_TRAP_MGR` に設定されていることを確認します。コマンドラインで、`lk_configsnmp`を「--query」オプション付きで使用して設定を確認します (`lk_configsnmp(1M)` マニュアルページの例を参照してください)。または、ファイル `/etc/default/LifeKeeper` の `LK_TRAP_MGR`のエントリを確認します。この変数は、SNMP トラップメッセージを生成する LifeKeeper の各ノードで設定する必要があります。

# 5.4.2.3. LifeKeeper イベントメール通知

LifeKeeper イベントメール通知の概要

LifeKeeper イベントメール通知の設定

メール通知のトラブルシューティング

# 5.4.2.3.1. LifeKeeper イベントメール通知の概要

LifeKeeper イベントメール通知は、特定のイベントが LifeKeeper クラスタで発生したときに 1 人以上のユーザがメールによる通知を受信する仕組みです。LifeKeeper のイベント通知機能では、特定のイベントが起きたときに通知を受信するアプリケーションを登録することができます (sendevent(5) マニュアルページを参照)。LifeKeeper は、LifeKeeper の動作を監視したいユーザのグループに向けて LifeKeeper の重要なイベントに関するメール通知を送信するように簡単に設定できます。

デフォルトでは、LifeKeeper イベントメール通知は無効になっています。この機能を有効にするには、/etc/default/LifeKeeper で指定する LK_NOTIFY_ALIAS 環境変数を設定する必要があります。LK_NOTIFY_ALIAS 環境変数には、メールアドレスまたはエイリアスを 1 つまたは複数個 (カンマ区切り) 設定できます。LK_NOTIFY_ALIAS を設定するには、コマンドラインから lk_confignotifyalias (lk_confignotifyalias(1M) マニュアルページで例を参照してください) を実行してイベントが発生したときにメールを受信するアドレスまたはアドレスリストを指定するか、デフォルトファイル /etc/default/LifeKeeper を編集してメールアドレスまたはアドレスリストを追加します。LK_NOTIFY_ALIAS= エントリを見つけて、アドレスまたはカンマ区切りのアドレスリストを入力します。選択した LifeKeeper イベントについてメールを送信する必要があるクラスタのすべてのノードで以上の手順を繰り返します。

メール通知を無効にするには、引数 *-disable* を付けて **lk_confignotifyalias** (lk_confignotifyalias(1M) マニュアルページで例を参照してください) を実行するか、デフォルトファイル /etc/default/LifeKeeper を編集して LK_NOTIFY_ALIAS の設定を削除します (この行を LK_NOTIFY_ALIAS= に変更)。

## メールが生成される LifeKeeper のイベント

以下の LifeKeeper イベントが発生するとメール通知が生成されます (LK_NOTIFY_ALIAS が設定されている場合)。

| **LifeKeeper** のイベント | イベントの説明 |
|---|---|
| LifeKeeper Startup Complete | LifeKeeper が起動したノードから送信されます。 |
| LifeKeeper Shutdown Initiated | LifeKeeper のシャットダウンを開始したノードから送信されます。 |
| LifeKeeper Shutdown Complete | LifeKeeper のシャットダウンを完了したノードから送信されます。 |
| LifeKeeper Manual Switchover Initiated on Server | 手動スイッチオーバを要求されたノードから送信されます。 |
| LifeKeeper Manual Switchover Complete - | 手動スイッチオーバが完了したノードから、リカバリに成功したリソースのリストと共に送信されます。 |

| | |
|---|---|
| recovered list | |
| LifeKeeper Manual Switchover Complete - failed list | 手動スイッチオーバが完了したノードから、切り替えに失敗したリソースのリストと共に送信されます。 |
| LifeKeeper Node Failure Detected | クラスタ内のノードに障害が発生したときに、クラスタ内の各ノードから送信されます。 |
| LifeKeeper Node Recovery Complete for Server - recovered list | 障害ノードからのリソースをリカバリしたクラスタ内の各ノードから、リカバリに成功したリソースのリストと共に送信されます。 |
| LifeKeeper Node Recovery Complete for Server - failed list | 障害ノードからのリソースのリカバリに失敗したクラスタ内の各ノードから、リカバリに失敗したリソースのリストと共に送信されます。 |
| LifeKeeper Resource Recovery Initiated | リソースをリカバリしているノードから送信されます。このメールに続いて、リカバリが完了したか失敗したかを示すメッセージ(「Resource Recovery Complete」または「Resource Recovery Failed」) が必ず送信されます。 |
| LifeKeeper Resource Recovery Complete | リソースのリカバリが成功した時点で、「LifeKeeper Resource Recovery Initiated」メッセージを送信したノードから、リカバリに成功したリソースのリストと共に送信されます。 |
| LifeKeeper Resource Recovery Failed | リソースが In Service の状態になることができない場合に、「LifeKeeper Resource Recovery Initiated」メッセージを送信したノードから、リカバリに成功したリソースのリストと共に送信されます。 |
| LifeKeeper Communications Path Up | ノードへのコミュニケーションパスが確立されました。 |
| LifeKeeper Communications Path Down | ノードへのコミュニケーションパスがダウンしました。 |
| LifeKeeper <Node monitoring> Failure Detected | Standby Node Health Check における Node監視で異常を検知したときに、異常を検知したノードから送信されます。<Node monitoring> には、検知した異常の内容が記載されます。 |
| LifeKeeper <OSUquickCheck> Failure Detected | Standby Node Health Check における OSUリソース監視で異常を検知したときに、異常を検知したノードから送信されます。<OSUquickCheck> には、異常が検知されたリソースのタグ名が記載されます。 |

# 5.4.2.3.2. LifeKeeper イベントメール通知の設定

## 前提条件

イベントメール通知機能は、LifeKeeper のコア機能の一部として含まれており、LifeKeeper の追加パッケージをインストールする必要はありません。ただし、LifeKeeper イベントのメール通知を生成する LifeKeeper の各ノードに電子メールソフトウェアがインストールされている必要があります。LifeKeeper は、mailx または s-nail パッケージによってインストールされるメールユーティリティを使用して通知を送信します。

メールの設定は、LifeKeeper の本機能が提供する範囲ではありません。デフォルトでは、LifeKeeper イベントメール通知は無効になっています。

## 設定作業

LifeKeeper イベントメール通知を設定するには、以下の作業を実施します。

1. 上述のメールユーティリティが利用できることを確認します。
2. LifeKeeper のイベントのメール通知を受信するユーザ (1 人以上) を特定し、LifeKeeper のデフォルトファイル `/etc/default/LifeKeeper` の `LK_NOTIFY_ALIAS` を設定します。これを行うには、コマンドラインを使用するか、ファイル `/etc/default/LifeKeeper` を編集して通知を受信するメールアドレスまたはエイリアスを指定します。

- ◦ コマンドラインからは、`lk_confignotifyalias` を使用します (詳細については、`lk_confignotifyalias (1M)` のマニュアルページを参照してください)。このユーティリティでは、カンマ区切りのメールアドレスまたはエイリアスのみ使用できます。
- ◦ または、デフォルトファイル `/etc/default/LifeKeeper` を編集してメールアドレスまたはエイリアスを追加します。`LK_NOTIFY_ALIAS=` エントリを見つけて「=」の右側にメールアドレスまたはエイリアス (1 つまたはカンマ区切りのリスト) を入力します (「=」の前後にはスペースを入れません)。

## 設定の確認

設定が正常に動作することを確認するには、LifeKeeper の処理を実行します (例えば、LifeKeeper を 開始 または 停止 する、または LifeKeeper GUI を使用して、あるリソースを手動で In Service の状態にするなど)。ファイル `/etc/default/LifeKeeper` の `LK_NOTIFY_ALIAS` で指定したユーザがメールを受信していること、LifeKeeper のログファイルにメッセージが記録されていることを確認します。メールを受信していない場合は、メール障害に対する通常のトラブルシューティング手順を実行してください。LifeKeeper のログを調べると、メール送信に問題があるかどうかを判断することができます。詳細については、メール通知のトラブルシューティング を参照してください。

# イベントメール通知の無効化

LifeKeeper によるメール通知の生成を無効にするには、ファイル /etc/default/LifeKeeper の LK_NOTIFY_ALIAS 環境変数からメールアドレスとエイリアスの割り当てを削除するだけです。コマンドラインで lk_confignotifyalias ユーティリティを「--disable」オプションを付けて実行します (k_confignotifyalias (1M) マニュアルページの例を参照してください)。または、/etc/default/LifeKeeper を編集して、LK_NOTIFY_ALIAS のエントリを LK_NOTIFY_ALIAS = に変更します。この手順は、メール送信を無効にしたい各ノードで実行する必要があります。

# 5.4.2.3.3. メール通知のトラブルシューティング

LifeKeeper イベントのメール通知に関連して予想される問題とその解決策を以下に説明します。具体的なエラーメッセージについては、 LifeKeeper メッセージカタログ を参照してください。

**問題:** LifeKeeper からのメールを受信しない。

**解決策:** メールユーティリティがインストールされていることを確認します (通常は /bin/mail にあります)。インストールされていない場合は、mailx または s-nail パッケージをインストールします。別の場所にインストールされている場合は、ファイル `/etc/default/LifeKeeper`PATH 変数にメールユーティリティのパスを追加します。

**問題:** LifeKeeper からのメールを受信しない。

**解決策:** メール設定を確認し、配信用のキューにメールメッセージが滞留していないことを確認します。メール設定の問題が原因でメッセージが滞留することがあります。`LK_NOTIFY_ALIAS` で指定しているメールアドレスが有効なアドレスであり、カンマで区切られていることを確認します。

**問題:** ログファイルに「mail returned」というエラーメッセージがある。

**解決策:** メールコマンドがエラー「X」を返す場合、LifeKeeper イベントがメールを生成、送信する際に問題 (「node failure」など) が発生しています。メール設定を確認し、`LK_NOTIFY_ALIAS` に含まれるメールアドレスが有効であり、アドレスのリストがカンマで区切られていることを確認します。また、`LK_NOTIFY_ALIAS` で指定しているメールアドレスのフォーマットを使用してコマンドラインからそれらのアドレスにメールを送信できることを確認します。

**問題:** メッセージや成功または失敗がログに何も記録されず、ノードのフェイルなどの LifeKeeper イベントが発生したときもメールを受信するはずのユーザがメールを受信しない。

**解決策:** `LK_NOTIFY_ALIAS` にメールアドレスが設定されており、複数の場合はカンマで区切られていることを確認します。コマンドラインで、`lk_confignotifyalias` を「--query」オプション付きで使用して設定を確認します (`lk_confignotifyalias(1M)` マニュアルページの例を参照してください)。または、ファイル `/etc/default/LifeKeeper` の `LK_NOTIFY_ALIAS` で確認します。この変数は、メール通知メッセージを生成する LifeKeeper の各ノードで設定する必要があります。また、LifeKeeper イベントメール通知の概要 で、その LifeKeeper イベントがメールメッセージを生成するのかどうかを確認します (すべてのイベントがメールメッセージを生成するわけではありません)。

# 5.4.2.4. 任意の設定作業

[Confirm Failover] と [Block Resource Failover] の設定

サーバのシャットダウンストラテジーの設定

LifeKeeper ハートビートの調整

SPS API でカスタム証明書を使用する

# 5.4.2.4.1. [Confirm Failover] と [Block Resource Failover] の設定

LifeKeeperの通常の動作として、ノード障害やリソース障害によるバックアップノードへの切り替えは自動的に行われるようになっています。しかし、利用する環境によっては、障害が検出されたシステムのフェイルオーバーリカバリーをLifeKeeper が実行する前にシステム管理者の手動による確認を必須とすることが望ましいこともあります。そのような場合には、この **[Confirm Failover]** や **[Block Resource Failover]** 設定の利用することができます。この機能を利用することによって、リソース障害やノード障害が発生した際に、自動的に行われるフェイルオーバーをブロックしたり、フェイルオーバーが行われるまでの待ち時間を設けたりすることができます。

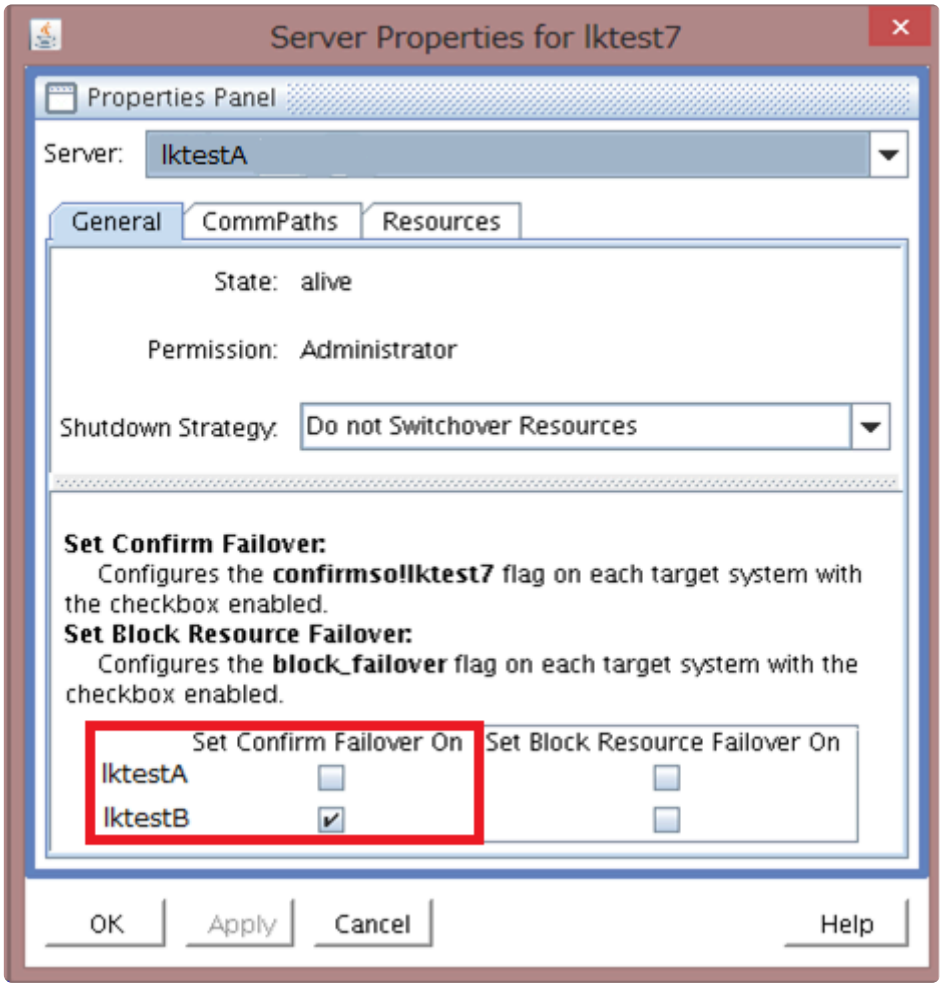
以下の説明、例、および考慮事項をよく読んで理解してから、お使いの SPS 環境で **[Confirm Failover]** または **[Block Resource Failover]** を設定してください。これらの設定は、コマンドライン、または **LifeKeeper の GUI の [Properties]** パネルから使用できます。

## [Set Confirm Failover On]

**[Confirm Failover]** を設定すると、LifeKeeper クラスターのノード障害によるフェイルオーバーが発生した時（注記: ノード障害は、システムへのすべてのLifeKeeperコミュニケーションパス障害によって識別されます。）、バックアップノードへの切り替えの実行に対して待ち時間を設けることができるようになります（後述のCONFIRMSODEF変数を参照して下さい）。また、その待ち時間の間にバックアップノードへの切り替えを実行するかしないかをユーザーが決定できるようになります。

> ✻ 注記: Set Confirm Failover On設定は、ノード障害が発生した場合にのみ有効です。一つ以上のコミュニケーションパスがアクティブであるリソース障害に対しては有効ではありません。

**[Confirm Failover]** 設定をGUIで有効にする場合は、サーバープロパティーのGeneralタブ画面を使用します。以下はサーバープロパティーのGeneral画面の例です。赤で囲まれた部分が **[Confirm Failover]** を設定する画面です。




> 注意： この設定は、SPS の管理者権限を持つユーザーのみが使用できます。

この画面の例では、lktestAという名前のホストから見た設定となります。上図の赤で囲まれた部分が **[Confirm Failover]** の設定部分で、HAクラスターを構成するノードの名前が縦に表示されています。この例では、lktestAの対向ノードがlktestBとなっています。

図の例ではlktestAを選択して設定状況を表示しており、lktestBの横にチェックが入った状態となっています。この場合、lktestB上にConfirm Failover フラグが作成され、lktestAからlktestBへのフェイルオーバーが行われる際に、フェイルオーバー実施までのタイムアウトと実施可否の確認が有効になります。このフラグの有無によって、 **[Confirm Failover]** の動作の実施がコントロールされます。

この時のフラグの作成状況については、コマンドで確認することができます。例にあるlktestAというホスト上でlktestBの欄にチェックを入れた場合には、lktestBに **[Confirm Failover]** フラグが作成されます。(例の通りの操作の場合lktestAにはフラグは作成されません)。具体的には以下のような出力になります。

```
[root@lktestB~]# flg_list

                confirmso!lktestA
```

「confirmso!lktestA」はflg_listコマンドの結果が出力され、 **[Confirm Failover]** フラグがlktestBノード上に設定されます。confirmsoフラグがセットされている状態で、フェイルオーバーが発生するとLifeKeeperのログファイルには以下のようなログが記録されます。

INFO:lcd.recover:::004113:chk_man_interv: Flag confirmso!hostname is set, issuing confirmso event and waiting for switchover instruction.

NOTIFY:event.confirmso:::010464:LifeKeeper: FAILOVER RECOVERY OF MACHINE lktestA requires manual confirmation! Execute '/opt/LifeKeeper/bin/lk_confirmso -y -s hostname ' to allow this failover, or execute '/opt/LifeKeeper/bin/lk_confirmso -n -s lktestA' to prevent it. If no instruction is provided, LifeKeeper will timeout in 600 seconds and the failover will be allowed to proceed.

このメッセージが出力されている間に、次のいずれかのコマンドを実行するとフェイルオーバーの可否をコントロールすることができます。

**フェイルオーバーを続行する場合**

#/opt/LifeKeeper/bin/lk_confirmso -y -s ホスト名

**フェイルオーバーをブロックする場合**

#/opt/LifeKeeper/bin/lk_confirmso -n -s ホスト名

コマンドを実行する際に指定するホスト名はConfirm Failoverフラグに書かれているホスト名、この例では、lktestAとなります。ログに具体的なコマンドの実行例が書かれていますので、内容を元にコマンドを実行してください。

設定された待ち時間を超えた場合、LifeKeeperのデフォルトの設定ではバックアップノードへのフェイルオーバー（あるいは、フェイルオーバーのブロック）を自動的に行うようになっています。タイムアウトを迎えた場合以下のようなログが記録されます。

lcdrecover[xxxx]: INFO:lcd.recover:::004408:chk_man_interv: Timed out waiting for instruction, using default CONFIRMSODEF value 0.

待ち時間を経過した際の動作は、/etc/default/LifeKeeperファイルの「CONFIRMSODEF=1 or 0」の設定で制御されます。デフォルトでは”0”が設定されており、待ち時間を超えた場合にはフェイルオーバーが継続されます。この値を”1”に変更した場合には、待ち時間を経過した場合にフェイルオーバーしないようにすることができます。

フェイルオーバー時の待ち時間を変更したい場合には、/etc/default/LifeKeeperファイルの「CONFIRMSOTO＝秒」の値を変更してください。上記CONFIRMSODEFの値によって決定されたフェイルオーバーの実行、あるいはブロックをする前に、この変数はユーザーからの手動確認を待つための秒数を指定します。この設定をこれらの変数の影響を反映するために、LifeKeeperやOSを再起動する必要はありません。CONFIRMSOTOに０秒を設定すると待ち時間無しで、CONFIRMSODEFの設定に基づいた動作をさせることができます。

## [Confirm Failover]設定を選択するタイミング

この設定は、コミュニケーションパスが冗長化されていない環境で場合のディザスターリカバリーやWAN構成で使用されます。

- 通常のサイト（非マルチサイトクラスターおよび非XenServer）では、あるサーバーで **[Properties]**

ページを開き、 **[Confirm Failover]** フラグをオンに設定するサーバーを選択してください。

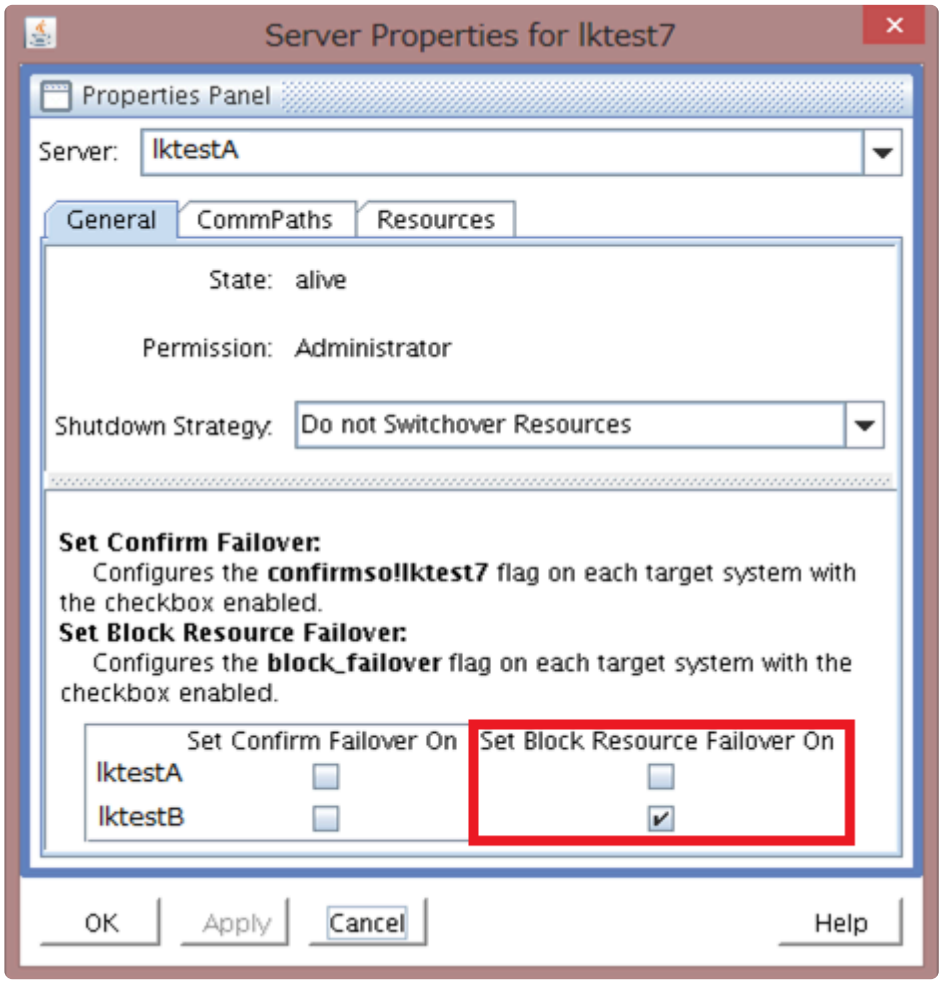
# [Block Resource Failover On]

**[Block Resource Failover]** 設定は、指定したシステムでのリソース障害に起因するフェイルオーバーをブロックします。

> 注記 : Block Resource Failover設定は、ノード障害が発生した場合のフェイルオーバー動作には影響しません。この設定は、ローカルリソースの回復に失敗し、クラスターで他ノードへリソースを転送する場合のみフェイルオーバーをブロックします。

デフォルトでは、リソース障害を検知した時、ローカルシステムでの障害リソースのリカバリー（ローカルリカバリー）を試行し、ローカルリカバリーが失敗した場合、または有効になっていない場合は、リソースが定義されている、優先順位が次に最も高いスタンバイノードにフェイルオーバーしようとします。Block Resource Failoverの設定は、この時のフェイルオーバーをブロックします。

Block Resource Failoverの設定をGUIで有効にする場合は、サーバープロパティーのGeneralタブ画面を使用します。以下はサーバープロパティーのGeneral画面の例です。赤で囲まれた部分が **[Block Resource Failover　On]** を設定する画面です。



> ✻ 注意 ：この設定は、SPS の管理者権限を持つユーザーのみが使用できます。

この画面の例は、lktestAという名前のホストから見た設定となります。上図の赤で囲まれた部分が **[Confirm Failover]** の設定部分で、HAクラスターを構成するノードの名前が縦に表示されています。この例では、lktestAの対向ノードがlktestBとなっています。

この場合、lktestB上にBlock Resource Failover フラグが作成されます。lktestBでflg_listコマンドを実行すると、「block_failover」というフラグが作成されていることを確認することができます。出力例は以下の通りです。

[root@lktestB~]# flg_list

block_failover

この結果lkestB上で、リソース障害が発生した場合には、他ノード （**lktestA**） へのフェイルオーバーがブロックされます。

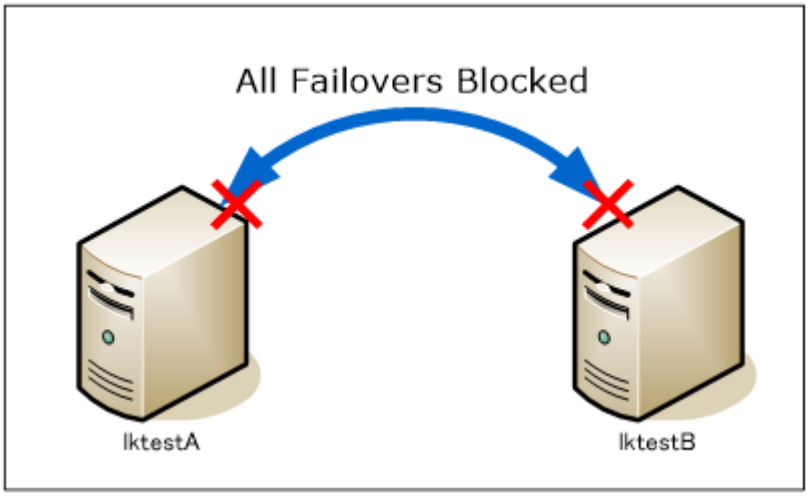
**block_failover**フラグは、それがセットされているノード上で発生したリソース障害によるフェイルオーバーをブロックします。 この設定によってフェイルオーバーがブロックされた場合には、以下のようなログが記録されます。

ERROR:lcd.recover:::004787:Failover is blocked by current settings. MANUAL INTERVENTION IS REQUIRED

# 設定例

いくつかの具体的な設定例を解説します。

## 全ての自動フェイルオーバーをすべてブロックする




この例ではlktestAとlktestBどちらでノード障害、リソース障害が発生してもフェイルオーバーしないようにします。この設定にはConfirm Failover設定とBlock Resource failover設定を利用します。LKCLI ではコマンド1つで設定可能です。詳細は lkcli server block-all-failovers を参照ください。GUI を使った設定例は

次の通りです。

1. lktestAサーバーを選択し、 **[Server Properties]** を表示してください。

   **[General]** タブで、lktestBの行にある **[Set Confirm Failover On]** ボックスと、LktestAとLktestB両方の行にある **[Set Block Resource Failover on]** ボックスをオンにしてください。

   GUIのlktestAのサーバープロパティー設定例

| | Set Confirm Failover On | Set Block Resource Failover On |
|---|---|---|
| lktestA | (チェックなし) | ✔ |
| lktestB | ✔ | ✔ |

   ※GUIでサーバープロパティーをみる場合、ノード名は、プロパティーパネルの上部近くに表示されます。

2. lktestBサーバーを選択し、 **[Server Properties]** を表示してください。

   **[General]** タブで、lktestAの行にある **[Set Confirm Failover On]** ボックスをオンにしてください。

   GUIのlktestBのサーバープロパティー設定例

| | Set Confirm Failover On | Set Block Resource Failover On |
|---|---|---|
| lktestB | (チェックなし) | ✔ |
| lktestA | ✔ | ✔ |

   ※GUIでは表示している自ノードのホスト名が列の上部に表示されます

   GUIで設定した後、lktestBでflg_listコマンドを使用してフラグ設定を確認すると、「confirmso!lktestA」フラグが作成されていることが確認できます。

   ここまでの結果それぞれのノードにconfirmso!hostnameとblock_failoverフラグがセットされていることを確認してください。confirmsoフラグについては、lktestA上のlktestBからのフェイルオーバーをブロックするために、フェイルオーバー確認が実行されることになっているhostnameが、フラグ名の内容の一部として記載されることを確認して下さい。lktestAは、lktestA上でlktestBをconfirmsoフラグ名の内容に記載する必要があります。本設定例ではそれぞれのノードで以下のようにフラグが作成されます。

| | Confirm Failover フラグ | Block Resource Failoverフラグ |
|---|---|---|
| lktestA側 | confirmso!lktestB | block_failover |
| lktestB側 | confirmso!lktestA | block_failover |

3. 両ノードの/etc/default/LifeKeeper のCONFIRMSOTOとCONFIRMSODEFの設定値を以下のように設定してください。（LifeKeeperやOSの再起動は必要ありません。）

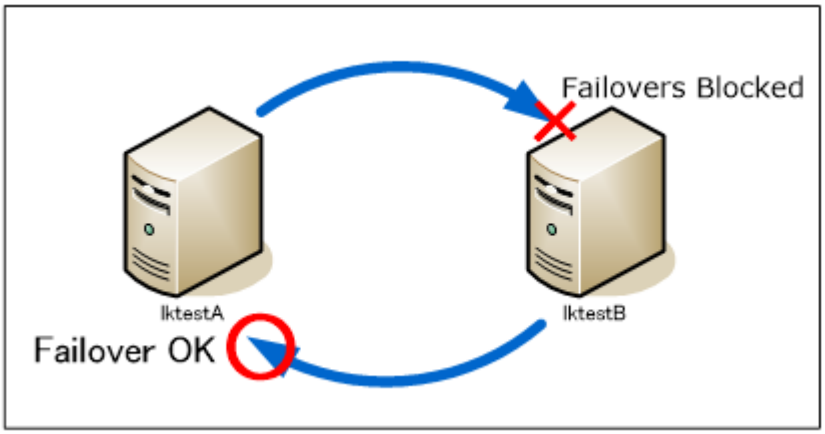
   CONFIRMSODEF = 1

   CONFIRMSOTO = 0

   CONFIRMSOTOの設定については待ち時間を設ける場合には時間を秒で指定することも可能です。ポイントはCONFIRMSODEFの設定が0(フェイルオーバーする)から1(フェイルオーバーしない)になっていることです。

   上記の設定により、オペレータが介在することなく、ノード障害はすぐにブロックされます。

## 一方向のフェイルオーバーをブロックする




この例ではlktestAで障害が検知された場合にはノード障害、リソース障害どちらの場合でもlktestBへのフェイルオーバーをブロックします。逆に、lktestBで障害が発生した場合にはノード障害、リソース障害どちらの場合でもServerAへフェイルオーバーします。

1. lktestAを選択し、 **[Server Properties]** を表示してください。

2. **[General]** タブで、lktestAの行にある **[Set Confirm Failover On]** ボックスと、lktestAの行にある **[Set Block Resource Failover On]** ボックスをオンにしてください。GUIでの設定状態は以下のようになります。

   GUIのlktestAのサーバープロパティー設定例

| | Set Confirm Failover On | Set Block Resource Failover On |
|---|---|---|
| lktestA | (チェックなし) | ✔ |
| lktestB | ✔ | (チェックなし) |

3. lktestBサーバーを選択し、 **[Server Properties]** を表示してください。

   **[General]** タブで、lktestAの行にある **[Set Confirm Failover On]** ボックスをオフ（チェックな

し）にしてください。GUIでの設定状態は以下のようになります。

GUIのlktestBのサーバープロパティー設定例

| | Set Confirm Failover On | Set Block Resource Failover On |
|---|---|---|
| lktestB | (チェックなし) | (チェックなし) |
| lktestA | (チェックなし) | ✔ |

※GUIでは表示している自ノードのホスト名が列の上部に表示されます。

この設定で、lktestB上に「confirmso!lktestA」フラグ、lktestA上に「Block failover」フラグが設定されていることを確認してください　(lktestAには、confirmsoフラグは設定されません)

| | Confirm Failover フラグ | Block Resource Failoverフラグ |
|---|---|---|
| lktestA | なし | block_failover |
| lktestB | confirmso!lktestA | なし |

4. lktestBの/etc/default/LifeKeeper を以下のように設定してください。LifeKeeperやOSの再起動は必要ありません。

   CONFIRMSODEF = 1

   CONFIRMSOTO = 0

   ここまでの設定によって、lktestAサーバーからlktestBのサーバーへのフェイルオーバーは許可されますが、lktestBからlktestAへのフェイルオーバーは許可されません。

# 5.4.2.4.2. サーバーのシャットダウンストラテジーの設定

シャットダウンストラテジーは、サーバーがシャットダウンするときにバックアップサーバーにリソースをスイッチオーバーするかどうかを制御する LifeKeeper の設定オプションです。以下のオプションがあります。

| | |
|---|---|
| Do not Switchover Resources (デフォルト) | LifeKeeper は、正常なシャットダウンではバックアップサーバーのリソースを起動しません。 |
| Switchover Resources | LifeKeeper は、正常なシャットダウンでバックアップサーバーのリソースを起動します。 |

シャットダウンストラテジーは、デフォルトでは「Do not Switchover Resources」に設定されています。クラスター内の各サーバーでどちらのストラテジーを使用するかを決定し、必要に応じてシャットダウンストラテジーを「Switchover Resources」に変更してください。

クラスター内の各サーバーで次のようにします。

1. [Edit] メニュー で **[Server]** を選択し、次に **[Properties]** をクリックします。
2. 修正するサーバーを選択します。
3. **[Server Properties]** ダイアログの [General] タブ で、 **[Shutdown Strategy]** を選択します。

**注意：** シャットダウンストラテジーが有効に機能するには、正常なシャットダウン時に LifeKeeper のプロセスが起動している必要があります。

# 5.4.2.4.3. LifeKeeper ハートビートの調整

## ハートビート設定項目の概要

LifeKeeper のハートビートは、各サーバーが「生存」していることを確認するためにコミュニケーションパスを通じて LifeKeeper のサーバー間で送信される信号です。ハートビートに関しては、LifeKeeper が障害を検知する速さを決定する要素が 2 つあります。

- **間隔**: ハートビート信号を送信する時間の間隔（単位は秒）。また、この時間内に対向サーバーからハートビート信号を含むLCM信号を受信できなかった場合、ハートビート停止と判断する。
- **ハートビート回数**: コミュニケーションパスが切断していると判定するハートビートの連続失敗回数。

これらのハートビートの値は、LifeKeeper デフォルトファイル /etc/default/LifeKeeper 内の以下の 2 つの設定項目で指定します。デフォルト値を使用した場合よりも早期にサーバーの障害を検知したい場合は、設定項目を変更することができます。

- LCMHBEATTIME (間隔)
- LCMNUMHBEATS (ハートビート回数)

次の表は、TCP および TTY 経由のハートビートの設定項目についてのデフォルト値と最小値の一覧です。TTY コミュニケーションパスは、媒体として通信速度が遅いため、間隔を 2 秒未満にすることはできません。

| 設定項目 | デフォルト値 | 最小値 |
|---|---|---|
| LCMHBEATTIME | 5 | 1 (TCP) 2 (TTY) |
| LCMNUMHBEATS | 3 | 2 (TCP、TTY) |

! 設定値については、クラスター内のすべてのサーバーで同じ値にする必要があります。

! LCMHBEATTIME は、デフォルトから変更しないでください。ネットワークやシステムに大きな負荷がかかると、LCMBEATTIME が5秒未満の場合、予期せぬハートビートの欠落が発生することがあります。各サーバーが「生きている」と判断するためのチューニングは、LCMNUMHBEATS を変更して行うことを強く推奨します。

### 例

LifeKeeper のクラスターで両方の間隔がデフォルト値に設定されていると仮定します。LifeKeeper は、サーバー間で 5 秒ごとにハートビートを送信します。通信障害によって 2 回のハートビートが途絶し、3 回目のハートビートで再開した場合、LifeKeeper はアクションを実行しません。コミュニケーションパスの

切断継続時間がハートビート 3 回分になった場合は、LifeKeeper はそのコミュニケーションパスを切断と判定します。ただし、他方の冗長的なコミュニケーションパスも切断と判定されるまではフェイルオーバーを開始しません。

## ハートビートの設定

設定項目とその値を追加するには、/etc/default/LifeKeeper ファイルを手動で編集する必要があります。通常、デフォルトファイルにはこれらの設定項目のエントリーが含まれていません。設定する値を含む次の行を最終行以降に追加してください。

```
LCMHBEATTIME=x

LCMNUMHBEATS=y
```

最小値を下回る値を設定した場合、LifeKeeper はその値を無視して代わりに最小値を採用します。

## 設定上の考慮事項

- 間隔を 5 秒未満に設定すると、ネットワークの中断による誤ったフェイルオーバーを発生させるリスクが高くなるため、5 秒未満に設定する場合は、コミュニケーションパスをプライベートネットワーク上で構成してください。

- 検証によると、ハートビート回数を 2 未満にした場合に誤ったフェイルオーバーの発生リスクが高まります。このため、この値は 2 以上に制限されています。

- 誤ったフェイルオーバーを回避するため、間隔およびハートビート回数の値はどちらもクラスタ内のすべてのサーバーで必ず同じ値にする必要があります。このため、これらの値を編集する前に両方のサーバーで LifeKeeper を停止しておく必要があります。LifeKeeper の稼働開始後、アプリケーションを保護している状態でハートビートの設定項目を編集する場合は、コマンド "`/opt/LifeKeeper/bin/lkstop -f`" を使用できます。このコマンドは LifeKeeper を停止しますが、保護下のアプリケーションは停止しません。

- LCMHBEATTIME および LCMNUMHBEATS の値に上限値はありません。ただし、非常に大きい数字に値を設定すると、LifeKeeper の障害検知能力は著しく損なわれます。例えば、両方の値を 25 に設定した場合、サーバー障害を検知するまで LifeKeeper は 625 秒間 (10 分間以上) 待機します。これはサーバーをリブートしてクラスタに再参加させるのに十分な時間です。

**注記:** TTY および TCP コミュニケーションパスの両方を使用する場合、各設定項目の値は両方のコミュニケーションパスに適用されます。唯一の例外は、TTY コミュニケーションパスの最小値である 2 未満の値が間隔に設定された場合です。

例えば、障害をできるだけ早く検知するために、LifeKeeper で許容される最小値を指定したとします。

```
LCMHBEATTIME=1

LCMNUMHBEATS=2
```

このとき LifeKeeper は TCP コミュニケーションパスの間隔に 1 秒を採用し、TTY の間隔には 2 秒を採用します。サーバー障害が発生すると、LifeKeeper は間隔の短い TCP の障害 (1 秒間隔の 2 回のハートビート後) を先に検知します。ただし、TTY の障害 (2 秒間隔の 2 回のハートビート後) を検知するまでは何もしません。

# 5.4.2.4.4. LifeKeeper API でカスタム証明書を使用する

LifeKeeper API のリリース 7.5 以降では、異なるシステムとの通信に SSL/TLS が使用されます。現在、この API は一部のみ使用され、内部使用のみとして予約されていますが、将来のリリースではお客様とサードパーティが使用できるように公開される可能性があります。デフォルトでは、ノード間で一定の身元確認が可能なデフォルト証明書が SPS と共にインストールされます。このドキュメントでは、デフォルト証明書を組織独自の認証局 (CA) が作成した証明書に置き換える方法を説明します。

注記: 通常の SPS 通信では、これらの証明書は使用されません。

## 証明書の使用方法

データ転送を保護するために SPS サーバ間の通信に SSL/TLS が使用されている場合、システムがそれ自体を識別するために証明書を提供します。また、システムは、CA 証明書を使用して、SSL 接続経由で提示された証明書を検証します。

以下の 3 種類の証明書が使用されます。

- `/opt/LifeKeeper/etc/certs/LK4LinuxValidNode.pem` (server certificate)
- `/opt/LifeKeeper/etc/certs/LK4LinuxValidClient.pem` (client certificate)
- `/opt/LifeKeeper/etc/certs/LKCA.pem` (certificate authority)

最初の 2 つの証明書は、サーバが実行する検証に合格するために CA 証明書による署名が必要です。証明書の共通名は検証されません。証明書は CA によって署名されるのみということに注意してください。

## 独自の証明書の使用

運用環境によっては、デフォルト証明書を組織内部の CA または商用 CA が作成した証明書に置き換える必要がある場合があります。そのような場合は、上記の 3 種類の証明書を、 *同じ証明書ファイル名* を持つ新しい証明書に置き換えます。これらの証明書は PEM 形式です。`LK4LinuxValidNode.pem` および `LK4LinuxValidClient.pem` はそれぞれ、キーと証明書の両方を含んでいます。`LK4LinuxValidNode.pem` 証明書は、サーバタイプの証明書です。`LK4LinuxValidClient.pem` は、クライアントタイプの証明書です。

デフォルトの証明書を置換した場合、変更を反映すために LifeKeeperを再起動する必要があります。証明書の設定を間違えると、`steeleye-lighttpd` デーモンが起動に失敗し、LifeKeeper のログファイルにエラーが記録されます。問題が発生した場合、このログファイルを参照すると実行すべき完全なコマンドを見ることができます。

# 期限切れの証明書の更新

注意：v9.5.1およびそれ未満のバージョンの証明書は、2021年9月に期限切れとなります。上記すべての古い証明書から、同じファイル名の新しい証明書への置き換えを実施してください。最新の証明書につきましては 弊社LifeKeeper / DataKeeper ユーザーポータル をご参照の上、ダウンロードしてください。

# 5.4.2.5. Linux の設定

<table>
<tr><td>オペレーティングシステム</td><td>必要なすべてのパッケージをインストールするためには、オペレーティングシステムはデフォルトでインストールしてください。最小構成のオペレーティングシステムでは必要なすべてのパッケージが含まれないため、LifeKeeper で使用することはできません。</td></tr>
<tr><td>カーネルのアップデート</td><td>LifeKeeper クラスタの可用性を最大限に引き出すには、システムで使用するカーネルのバージョンが非常に重要です。次の表に、LifeKeeper 認定テストに合格した、サポート対象のディストリビューション、バージョン、およびカーネルを示します。<br><br>注記:カーネルのアップグレードを実行する際、インストールイメージから setup スクリプト(./setup) の再実行が必要になる場合があります。カーネルアップグレード後に DataKeeper が起動しない場合には、setup スクリプトを実行して適切なカーネルモジュールをインストールしてください。<br><br>
<table>
<tr><th>ディストリビューション/バージョン</th><th>サポート対象のバージョン</th><th>サポート対象のカーネル</th></tr>
<tr><td>Red Hat Enterprise Linux for AMD64/EM64T<br><br>(*DataKeeper 非同期ミラーは、RHEL 7.4～7.6 の一部のカーネルではサポートされていません 詳細については、ここをクリックしてください。 )</td><td>7<br>7.1<br>7.2<br>7.3<br>7.4*<br>7.5*<br>7.6*<br>7.7<br>7.8<br>7.9</td><td>3.10.0-123.el7<br>3.10.0-229.el7<br>3.10.0-327.el7<br>3.10.0-514.el7<br>3.10.0-693.el7<br>3.10.0-862.el7<br>3.10.0-957.el7<br>3.10.0-1062.el7<br>3.10.0-1127.el7<br>3.10.0-1160.el7</td></tr>
<tr><td>Red Hat Enterprise Linux for AMD64/EM64T<br><br>(RHEL 7 から RHEL 8 のように Kernel をメジャーバージョンアップすることはサポートされません)<br>(*kernel の問題に起因してシンプロビジョニングディスクでDataKeeperの非同期モードは利用できません。詳細については 既知の問題と制限 をご参照ください。 )</td><td>8.0<br>8.1<br>8.2<br>8.3<br>8.4<br>8.5<br>8.6*<br>8.7</td><td>4.18.0-80.el8.x86_64<br>4.18.0-147.el8.x86_64<br>4.18.0-193.el8.x86_64<br>4.18.0-240.el8.x86_64<br>4.18.0-305.el8.x86_64<br>4.18.0-348.el8.x86_64<br>4.18.0-372.9.1.el8.x86_64*<br>4.18.0-425.3.1.el8.x86_64</td></tr>
</table>
</td></tr>
</table>

| ディストリビューション/バージョン | サポート対象のバージョン | サポート対象のカーネル |
|---|---|---|
| Red Hat Enterprise Linux for AMD64/EM64T<br><br>(RHEL 8 から RHEL 9 のように Kernel をメジャーバージョンアップすることはサポートされません)<br><br>*(*kernel の問題に起因してシンプロビジョニングディスクで DataKeeper の非同期モードは利用できません。詳細については 既知の問題と制限 をご参照ください。 )* | 9.0*<br>9.1 | 5.14.0-70.13.1.el9_0<br>5.14.0-162.6.1.el9_1 |
| SUSE Linux Enterprise Server 12 for x86_64<br><br>*(*SP3はカーネルを4.4.82-6.9.1にアップグレードしてください)* | 12 SP1<br>12 SP2<br>12 SP3*<br>12 SP4<br>12 SP5<br>※SLES12.0はサポート対象外 | 3.12.49-11.1<br>4.4.21-69.1<br>4.4.82-6.9.1<br>4.12.14-94.41.1<br>4.12.14-120.1 |
| SUSE Linux Enterprise Server 15 for x86_64<br><br>*(*DataKeeper にセクターサイズが奇数のディスクは利用できません)*<br>*(*SLES12 から SLES15 のように Kernel をメジャーバージョンアップすることはサポートされません)*<br>*(**SP3 はカーネルを 5.3.18-59.5 にアップグレードしてください)* | 15*<br>15 SP1<br>15 SP2<br>15 SP3**<br>15 SP4 | 4.12.14-23.1<br>4.12.14-195.1<br>5.3.18-14.1<br>5.3.18-59.5<br>5.14.21-150400.24.21.2 |
| Oracle Linux<br><br>*(*DataKeeper 非同期ミラーは、OL 7.4～7.6 の一部のカーネルではサポートされていませ* | 7<br>7.1<br>7.2<br>7.3<br>7.4*<br>7.5* | 3.10.0-123.el7<br>3.10.0-229.el7<br>3.10.0-327.el7<br>3.10.0-514.el7<br>3.10.0-693.el7<br>3.10.0-862.el7 |

| ディストリビューション/バージョン | サポート対象のバージョン | サポート対象のカーネル |
|---|---|---|
| ん 詳細についてはここをクリックしてください )<br>*(**UEK6はカーネルを5.4.17-2102.202.5にアップグレードしてください)* | 7.6*<br>7.7<br>7.8<br>7.9<br>UEK R3<br>UEK R4<br>UEK R5<br>UEK R6** | 3.10.0-957.el7<br>3.10.0-1062.el7<br>3.10.0-1127.el7<br>3.10.0-1160.el7<br>3.8.13-16.2.1.el7uek<br>4.1.12-37.3.1.el7uek<br>4.14.35-1818.3.3.el7uek<br>5.4.17-2102.202.5.el7uek |
| Oracle Linux<br>*(Oracle Linux 7 から Oracle Linux 8 のように Kernel をメジャーバージョンアップすることはサポートされません。)*<br>*(*kernel の問題に起因してシンプロビジョニングディスクで DataKeeper の非同期モードは利用できません。詳細については 既知の問題と制限 をご参照ください。 )*<br>*(**UEK6はカーネルを5.4.17-2102.202.5にアップグレードしてください)* | 8.0<br>8.1<br>8.2<br>8.3<br>8.4<br>8.5<br>8.6*<br>8.7<br>UEK R6**<br>UEK R7 | 4.18.0-80.el8<br>4.18.0-147.el8.x86_64<br>4.18.0-193.el8.x86_64<br>4.18.0-240.el8.x86_64<br>4.18.0-305.el8.x86_64<br>4.18.0-348.el8.x86_64<br>4.18.0-372.9.1.el8.x86_64*<br>4.18.0-425.3.1.el8.x86_64<br>5.4.17-2102.202.5.el8uek<br>5.15.0-3.60.5.1.el8uek.x86_64 |
| Oracle Linux<br>*(Oracle Linux 8 から Oracle Linux 9 のように Kernelをメジャーバージョンアップすることはサポートされません)*<br>*(*kernel の問題に起因して シンプロビジョニングディスクでDataKeeperの非同期モードは利用できません 詳細については 既知の問題と制限 をご参照ください )* | 9.0*<br>9.1<br>UEK R7 | 5.14.0-70.13.1.el9_0<br>5.14.0-162.6.1.el9_1<br>5.15.0-0.30.19.el9uek |
| Miracle Linux<br>*kernel の問題に起因してシンプロビジョニングディスクで DataKeeper の非同期モードは利用できません。 詳細については 既知の問題と制限 をご参照ください。* | 8.4<br>8.6* | 4.18.0-305.el8<br>4.18.0-372.9.1.el8 |

| ディストリビューション/バージョン | サポート対象のバージョン | サポート対象のカーネル |
|---|---|---|
| Miracle Linux<br><br>*( Miracle Linux 8 から Miracle Linux 9 のようにKernelをメジャーバージョンアップすることはサポートされません)*<br><br>**kernel の問題に起因してシンプロビジョニングディスクで DataKeeper の非同期モードは利用できません。 詳細については 既知の問題と制限 をご参照ください。* | 9.0* | 5.14.0-70.13.1.el9_0 |
| CentOS<br><br>*(*DataKeeper 非同期ミラーは、CentOS 7.4～7.6 の一部のカーネルではサポートされていません 詳細についてはここをクリックしてください )* | 7<br>7.1<br>7.2<br>7.3<br>7.4*<br>7.5*<br>7.6*<br>7.7<br>7.8<br>7.9 | 3.10.0-123.el7<br>3.10.0-229.el7<br>3.10.0-327.el7<br>3.10.0-514.el7<br>3.10.0-693.el7<br>3.10.0-862.el7<br>3.10.0-957.el7<br>3.10.0-1062.el7<br>3.10.0-1127.el7<br>3.10.0-1160.el7 |
| CentOS<br><br>*(CentOS 7 から CentOS 8 のように Kernel をメジャーバージョンアップすることはサポートされません)* | 8.0<br>8.1<br>8.2<br>8.3 | 4.18.0-80.el8<br>4.18.0-147.el8.x86_64<br>4.18.0-193.el8.x86_64<br>4.18.0-240.el8.x86_64 |
| Rocky Linux<br><br>**kernel の問題に起因してシンプロビジョニングディスクで DataKeeper の非同期モードは利用できません。詳細については 既知の問題と制限 をご参照ください。*<br><br>*詳細については サポートマトリックス を参照してください。* | 8.4<br>8.5<br>8.6*<br>8.7 | 4.18.0-305.3.1.el8_4<br>4.18.0-348.el8.0.2<br>4.18.0-372.9.1.el8.<br>4.18.0-425.3.1.el8 |

<table>
<tr><td></td><td>
<table>
<tr><th>ディストリビューション/バージョン</th><th>サポート対象のバージョン</th><th>サポート対象のカーネル</th></tr>
<tr><td>Rocky Linux<br><br>*kernel の問題に起因して シンプロビジョニングディスクで DataKeeper の非同期モードは利用できません 詳細については 既知の問題と制限 をご参照ください。<br><br>詳細については サポートマトリックス を参照してください。</td><td>9.0*<br>9.1<br>UEK R7</td><td>5.14.0-70.13.1.el9_0<br>5.14.0-162.6.1.el9_1<br>5.15.0-0.30.19.el9uek</td></tr>
</table>
<br>注記: このリストのサポート対象のディストリビューションおよびカーネルは、LifeKeeper のみを考慮したものです。お使いのサーバーおよびストレージハードウェアについては、各メーカーがサポートするディストリビューションおよびカーネルに従ってください。</td></tr>
<tr><td>LUN のサポート</td><td>Linux の SCSI ドライバには、論理ユニット (LUN) の検索対象とするデバイスを制御するいくつかのパラメータがあります。<br><br>• LUN をサポートしないデバイスのリスト – このリストのデバイスは LUN をサポートしないことが既知であるため、SCSI ドライバはこれらのデバイスに対して LUN を検索することを許可しません。<br><br>• LUN をサポートするデバイスのリスト – このリストのデバイスは LUN をサポートすることが既知であるため、必ず LUN を検索します。<br><br>• Probe all LUNs on each SCSI device – デバイスがどちらのリストにも存在しない場合、検索するかどうかを指定します。このパラメータは、make config を使用して SCSI モジュールセクションで設定します。<br><br>(SUSE を含む) ほとんどのディストリビューションでは、Probe all LUNs 設定はデフォルトで有効になっていますが、Red Hat ではデフォルトで無効に設定されています。LifeKeeper の構成でデータ保護を目的として通常使用される外部 RAID コントローラには、多くの場合、複数の LUN (論理ユニット) が設定されます。LUN のサポートを有効にするには、このフィールドを選択してカーネルを再構築する必要があります。<br><br>カーネルやモジュールを再構築せずに Probe all LUNs を有効にするには、変数 max_scsi_luns を 255 に設定します (これによって最大 255 個の LUN をスキャンするようになります)。SCSI ドライバがモジュールになっているカーネル (Red Hat など) で max_scsi_luns を設定するには、/etc/modules.conf に以下のエントリを追加し、初期 RAM ディスクを再構築して、再起動してからその RAM ディスクを読み込みます。<br><br><code>options scsi_mod max_scsi_luns=255</code></td></tr>
</table>

| | |
|---|---|
| | SCSI ドライバがコンパイルされるカーネル (SUSE など) で max_scsi_luns を設定するには、*/etc/lilo.conf* に以下のエントリを追加します。<br><br>`append="max_scsi_luns=255"`<br><br>**注記**: 255 個の LUN をスキャンすると、デバイスによってはブートのパフォーマンスに悪影響を与える可能性があります (特に、BLIST_SPARSELUN が指定されたデバイス)。Dell PV650F というアレイではそのような状況が発生しました。このパフォーマンスの問題を回避するには、アレイ上で設定した LUN の最大数 (16 または 32 など) を max_scsi_luns に設定します。例えば、以下のようになります。<br><br>`append="max_scsi_luns=16"` |
| **チャンネルボンディング、ネットワークチーミングの動作確認環境** | LifeKeeperでは、以下の設定のチャンネルボンディング、またはネットワークチーミングを用いた環境での動作確認を行なっています。<br><br>• チャンネルボンディングにおけるボンディングポリシー<br><br>+ balance-rr<br><br>+ active-backup<br><br>• ネットワークチーミングにおけるランナー<br><br>+ round-robin<br><br>+ active-backup<br><br>RHEL8 以降では NetworkManager で connection.autoconnect-slaves を true で使用してください。 |

# 5.4.2.6. データレプリケーションの設定

| 項目 | 説明 |
| --- | --- |
| **SIOS DataKeeper の機能** | SIOS DataKeeper は、Linux カーネルの機能を利用しています。そのため、最低要件としてカーネルバージョン 2.6 以降である必要があります。<br>利用可能な OS は サポートマトリックス を参照してください。 |
| **SIOS DataKeeper ドキュメンテーション** | SIOS DataKeeper のドキュメンテーションは、SIOS Technology Corp. の Web サイトにある「SIOS Protection Suite テクニカルドキュメンテーション」の中に収録されています。 |

# 5.4.2.7. ネットワーク設定

| 項目 | 説明 |
| --- | --- |
| ルーティングテーブルに対する IP Recovery Kit の影響 | LifeKeeper が保護する IP アドレスは、論理インターフェースとして、Linux 上で実装されます。Linux 上で論理インターフェースを設定すると、その論理インターフェースに関連付けられたサブネットへのルートが自動的にルーティングテーブルに追加されます。例えば、物理インターフェースによってそのサブネットへのルートがすでに存在する場合も同様です。この追加により、同じサブネットに対して複数のルーティングテーブルエントリが作成される可能性があります。<br><br>接続元のアドレスを検査して確認するアプリケーションの場合、複数のルーティングテーブルエントリがあると、LifeKeeper システムが (LifeKeeper がインストールされていない) 他のシステム上のそのようなアプリケーションに接続しようとしたときに問題が発生することがあります。複数のルーティングテーブルエントリによって、物理インターフェースからではなく論理インターフェースから接続が張られているように見えます。 |
| IP サブネットマスク | LifeKeeper 保護下の IP 設定では、物理インターフェースの IP アドレスと、LifeKeeper が保護するエイリアス IP アドレスのサブネットを同じにする場合、2 つのアドレスのサブネットマスクを同じにする必要があります。サブネットマスクの設定を間違えると、LifeKeeper GUI のクライアントとサーバ間の接続に遅延や障害が発生します。 |
| EEpro100 ドライバの初期化 | Intel Ethernet インターフェースを搭載するシステムでは、eepro100 ドライバの初期化の問題を解決するために、Intel e100 ドライバをインストールする必要があります。eepro100 ドライバを使用すると、ブート時にインターフェースが起動したときに以下のエラーが発生し、インターフェースをシャットダウンするまでエラーを出し続けることがあります。<br><br>eth0: card reports no Rx buffers<br><br>eth0: card reports no resources |

# 5.4.2.8. アプリケーションの設定

| 項目 | 説明 |
| --- | --- |
| データベース初期設定ファイル | データベースの初期設定ファイルは、共有デバイス上に置いてローカルファイルシステムの指定場所にシンボリックリンクを作成するか、または個別のシステム上に保持して変更を適用する必要がある場合に手動で両方のシステムを更新するかのいずれかに必要があります。 |
| **Oracle** のローカルマウントポイント | Oracle のローカル環境は、 *internal* として接続するか、 *sysdba* として接続するかによって異なります。データベースを LifeKeeper の保護下に置く場合、「`connect / as sysdba`」を使用してローカルマウントポイント上にデータベースを作成する必要があります。 |
| **Apache** のアップデート | Linux オペレーティングシステムのアップグレードの一環として、SPS が保護する Apache アプリケーションをアップグレードするには、起動時にデフォルトサーバインスタンスを無効にする必要があります。<br><br>設定ファイル *(httpd.conf)* がデフォルトのディレクトリ *(/etc/httpd/conf)* にある場合、Red Hat のアップグレードにより設定ファイルが上書きされます。したがって、アップグレードする前にファイルのコピーを作成し、アップグレードした後にファイルをリストアする必要があります。<br><br>また、*Apache Web Server Recovery Kit 管理ガイド* の「Apache Web Server の設定に関する考慮事項」セクションも参照してください。 |

# 5.4.2.9. ストレージとアダプターの設定

| 項目 | 説明 |
| --- | --- |
| Azure 共有ディスク | **アプリケーションとファイル システムの保護:** LifeKeeper で Azure 共有ディスクを適切に構成するには、steeleye-lkSCSI3 リカバリーキットをインストールする必要があります。 LifeKeeper 9.6.2 以降では、steeleye-lkSCSI3 リカバリーキットが自動的に選択され、Microsoft Azure にインストールされます。<br><br>**SCSI-3パーシステントリザベーションの使用:** SCSI3 キットは “Write Exclusive” のリザベーションタイプで SCSI-3 持続的リザベーションを使用します。これは、クラスターのあるノードによってリザベーションされたデバイスは、クラスターの他のノードから読み取り可能なままですが、これらの他のノードはデバイスに書き込むことができないことを意味します。これは、他のノードでファイルシステムをマウントして読み取り専用でアクセスできることを意味するものではないことに注意してください。<br><br>LifeKeeper は sg_persist ユーティリティを使用して、持続的なリザベーションを発行し、監視します。必要であれば、LifeKeeper は sg_persist(8) ユーティリティをインストールします。<br><br>構成の詳細については Microsoft の Azure Shared Disk を参照してください。 |
| マルチパス I/O 冗長コントローラ | マルチパス I/O のソリューションには数種類あり、すでに利用可能なものや Linux 環境向けに開発中のものなどがあります。SIOS Technology Corp. は、多くのサーバベンダ、アダプタベンダ、およびドライバー開発者と積極的に協力することで、LifeKeeper とマルチパス I/O ソリューションとの協調動作を実現しています。データーの整合性を保護するために LifeKeeper が使用する SCSI リザベーションは、特殊な要件を必要とするため、マルチパス I/O ソリューションの最初の実装では多くの場合要件が満たされません。<br><br>ディスクアレイサポートに関する以下の技術情報を参照し、個別のアレイがマルチパスおよび特定のマルチパスソリューションでサポートされているかを判断してください。マルチパスおよび特定のマルチパスソリューションと共に動作する LifeKeeper のサポート対象として一覧に指定されていないアレイは、サポート対象ではないと考えてください。 |
| マルチパス構成での大量の I/O | マルチパス構成では、パスの操作中に大量の I/O を実行すると、システムが応答しなくなったように見えることがあります。マルチパスのソフトウェアが LUN のアクセスをあるパスから別のパスに移動する場合、処理中の I/O も新しいパスに移動させる必要があります。この I/O の経路変更は、その I/O の応答時間の遅延を発生させます。この間にさらに I/O が発 |

| | |
|---|---|
| | 行されると、それらはシステム内のキューとなり、システムはプロセス用のメモリを使い果たしてしまう可能性があります。非常に高負荷の I/O の下では、これらの遅延と低メモリ状態によってシステムが無応答になり、LifeKeeper がこれをサーバのダウンとして検知し、フェイルオーバを開始することがあります。<br><br>この問題が発生する頻度には多くの要因が影響を及ぼします。<br><br>• プロセッサの速度は、I/O がキューに保持される速さに影響します。高速なプロセッサでは、障害が検知される頻度が高くなります。<br>• システムメモリの搭載量は、システムが無応答になるまでにキューに保持できる I/O の数に影響します。 メモリが多いシステムでは、障害が検知される頻度が低くなります。<br>• 使用する LUN の数は、キューに保持できる I/O の量に影響します。<br>• I/O の特性は、キューに保持される I/O の量に影響します。問題が発生したテストケースでは、ディスクにデータを無制限に書き込んでいました。ほとんどのアプリケーションは、データの読み取りと書き込みの両方を行うはずです。フェイルオーバを待って読み取りがブロックされることで書き込みも抑制され、結果的に I/O 速度が減少して障害検知の頻度が低くなります。<br><br>例えば、RDAC を使用した IBM DS4000 のマルチパス構成のテストでは、DS4000 への I/O スループットを毎秒 190 MB 以上にしてパス障害をシミュレーションした場合に LifeKeeper は約 12 回に 1 回サーバの障害を (誤) 検出しました。このテストでは、サーバとして IBM x345 (デュアル Xeon 2.8GHz プロセッサとメモリ 2 GB を搭載) を使用し、DS4400 に接続して使用サーバ当たり 8 ボリューム (LUN) にしました。フェイルオーバを抑止するために、LifeKeeper の LCMNUMHBEATS パラメータ (`/etc/default/LifeKeeper` 内) を 16 に増やしました。このパラメータの変更により、無応答のシステムが生存していないと判定するまでに LifeKeeper は、デフォルトの約 15 秒ではなく、約 80 秒間待機するようになります。 |
| 大規模ストレージ構成の場合のスイッチオーバに関する特別な考慮事項 | いくつかの大規模ストレージ構成 (例えば、複数の論理ボリュームグループがあり、各ボリュームグループ内に 10 以上の LUN を持つ構成) では、LifeKeeper は障害を検出したときにデフォルトの 300 秒のタイムアウト時間内に sendevent を完了することができない場合があります。その結果、バックアップシステムへのスイッチオーバが失敗します。サービス状態にならないリソースが生じ、LifeKeeper のログにエラーメッセージが記録されます。<br><br>大規模ストレージ構成では、/etc/default/LifeKeeper ファイルの SCSIER |

| | |
|---|---|
| | ROR を「event」から「halt」に変更することを推奨します。これにより LifeKeeper は SCSI エラーの発生時に「halt」を実行します。LifeKeeper はバックアップシステムへのフェイルオーバに成功するようになります。 |
| HP 3PAR StoreServ 7200 FC | HP 3PAR StoreServ 7200 は、以下の構成で SIOS Technology Corp. のパートナーによってテスト済みです。<br><br>QLogic QMH2572 8Gb FC HBA for HP BladeSystem c-Class(ファームウェアバージョン5.06.02 (90d5) を使用した)HP 3PAR StoreServ 7200 (ファームウェア (HP 3PAR OS) バージョン 3.1.2), ドライバーバージョン 8.03.07.05.06.2-k (RHELにバンドル) + DMMP (device-mapper-1.02.66-6, device-mapper-multipath-0.4.9-46.el6)<br><br>テストはRHEL 6.2 (x86_64) を使用したLifeKeeper for Linux v8.1.1 で実施されました。<br><br>**注記:** 3PAR StoreServ 7200 は、デフォルトのパスチェッカとのリザベーション競合を返します。この競合を回避するには、/etc/default/LifeKeeper 内に次のパラメータを設定 (追加) してください。<br><br>**DMMP_REGISTRATION_TYPE=hba** |
| HP 3PAR StoreServ 7400 FC | HP 3PAR StoreServ 7400は、以下の構成で SIOS Technology Corp. のパートナーによってテスト済みです。<br><br>HP 3PAR StoreServ 7400 (ファームウェア (HP 3PAR OS) バージョン 3.1.2) + HP DL380p Gen8 + Emulex LightPulse Fibre Channel SCSI HBA (ドライバーバージョン 8.3.5.45.4p) + DMMP (device-mapper-1.02.66-6, device-mapper-multipath-0.4.9-46.el6)<br><br>テストはRHEL 6.2 (x86_64) を使用したLifeKeeper for Linux v8.1.1 で実施されました。<br><br>**注記:** 3PAR StoreServ 7400 は、デフォルトのパスチェッカとのリザベーション競合を返します。この競合を回避するには、/etc/default/LifeKeeper 内に次のパラメータを設定 (追加) してください。<br><br>**DMMP_REGISTRATION_TYPE=hba**<br><br>およびユーザーフレンドリーデバイスマッピングはサポートされません。 “multipath.conf” に次のパラメータを設定してください。<br><br>**“user_friendly_names no”** |

| | |
|---|---|
| HP 3PAR StoreServ 7400 iSCSI (DMMP Recovery Kit を使用したマルチパス構成) | HP 3PAR StoreServ 7400は、以下の構成で SIOS Technology Corp. のパートナーによってテスト済みです。<br><br>HP 3PAR StoreServ 7400 (ファームウェア (HP 3PAR OS) version 3.1.3) + HP Ethernet 10Gb 2-port 560SFP+ Adapter (Networkdriver ixgbe-3.22.0.2) + iSCSI (iscsi-initiator-utils-6.2.0.873-10.el6.x86_64)、DMMP (device-mapper-1.02.79-8.el6、device-mapper-multipath-0.4.9-72.el6).<br><br>注記 : 3PAR StoreServ 7400 iSCSI は、リザベーション競合を返す場合があります。この競合を回避するため、/etc/default/LifeKeeper 内に次のパラメータを設定 (追加) してください。<br><br>**DMMP_REGISTER_IGNORE=TRUE** |
| HP 3PAR StoreServ 7400 iSCSI(Quorum/Witness Kit　使用) | HP 3PAR StoreServ 7400は、以下の構成で SIOS Technology Corp. のパートナーによってテスト済みです。<br><br>iSCSI (iscsi-initiator-utils-6.2.0.872-21.el6.x86_64) + DMMP (device-mapper-multipath-0.4.9-41 + device-mapper-1.02.62-3) + nx_nic v4.0.588.<br><br>DMMP と DMMP Recovery Kit を RHEL 6.1 で利用する場合には、Quorum/Witness Server Kit と STONITH 機能を利用する必要があります。SCSI リザベーションを無効にするため、“/etc/default/LifeKeeper“に RESERVATIONS=none を追記する必要があります。<br><br>サーバーは IPMI 2.0 に準拠したインタフェースを備えている必要があります。 |
| HP 3PAR StoreServ 10800 FC | HP 3PAR StoreServ 10800 FC は、以下の構成で SIOS Technology Corp. のパートナーによってテスト済みです。<br><br>ファームウェア (HP 3PAR OS) バージョン3.1.2 + HP DL380p Gen8 + Emulex LightPulse Fibre Channel HBA (ドライバーバージョン 8.3.5.45.4p) + DMMP (device-mapper-1.02.66-6, device-mapper-multipath-0.4.9-46 el6)<br><br>テストはRHEL 6.2 (x86_64) を使用したLifeKeeper for Linux v8.1.2 で実施されました。<br><br>注記: 3PAR StoreServ 10800 FC は、デフォルトのパスチェッカーとのリザベーション競合を返します。この競合を回避するには、/etc/default/LifeKeeper 内に次のパラメータを設定 (追加) してください。 |

| | |
|---|---|
| | **DMMP_REGISTRATION_TYPE=hba**<br><br>およびユーザーフレンドリーデバイスマッピングはサポートされません。 “multipath.conf” に次のパラメータを設定してください。<br><br>**“user_friendly_names no”** |
| HP MSA1040/2040fc | HP MSA 2040 Storage FC は、以下の構成で SIOS Technology Corp. のパートナーによってテスト済みです。<br><br>HP MSA 2040 Storage FC (ファームウェアGL101R002) + HP SN1000Q 16Gb 2P FC HBA QW972A (ファームウェアバージョン 6.07.02, ドライバーバージョン 8.04.00.12.06.0-k2 (RHELにバンドル)) + DMMP (device-mapper-1.02.74-10, device-mapper-multipath-0.4.9-56).<br><br>テストは RHEL 6.3 (X86_64) を使用した LifeKeeper for Linux v8.1.2 で実施されました。 |
| HP P9500/XP | SIOS LifeKeeper for Linux v7.2 以降を使用する場合について、Hewlett-Packard 社により認定。テストに使用されたモデルは HP P9500/XP です。以下の環境の LifeKeeper で動作することが認定されています。<br><br>• Red Hat Enterprise for 32-bit, x64 (64-bit; Opteron および Intel EMT64)<br>RHEL 5.3, RHEL 5.4, RHEL 5.5<br>• SuSE Enterprise Server for 32-bit, x64 (64-bit; Opteron および Intel EMT64)<br>SLES 10 SP3, SLES 11, SLES 11 SP1<br>• ネイティブまたは内蔵のクラスタリングソリューション RHCS および SLE HA |
| HP StoreVirtual 4330 iSCSI (DMMP Recovery Kit を使用したマルチパス構成) | HP StoreVirtual 4330 は、以下の構成で SIOS Technology Corp. のパートナーによってテスト済みです。<br><br>HP StoreVirtual 4330 (ファームウェアHP LeftHand OS 10.5) + HP Ethernet 1Gb 4-port 331FLR (Networkdriver tg3-3.125g) + iSCSI (iscsi-initiator-utils-6.2.0.872-41.el6.x86_64)、 DMMP (device-mapper-1.02.74-10.el6,device-mapper-multipath-0.4.9-56.el6) |
| HP StoreVirtual(LeftHand) シリーズ OS (SAN/iQ) バージョン 11.00 iSCSI | HP 社 StoreVirtual(LeftHand) ストレージにおいて、OS(SAN/iQ) バージョン 11.00 をサポートします。本バージョンが使われている全ての Stor |

| | |
|---|---|
| | eVirtual シリーズがサポート対象であり、仮想ストレージ・アプライアンスである StoreVirtual VSA も含まれます。<br><br>テスト構成は次の通りです。<br><br>StoreVirtual VSA(11.0.00.1263.0) + RHEL 6.4(x86_64) + DMMP (device-mapper-1.02.77-9.el6.x86_64, device-mapper-multipath-0.4.9-64.el6.x86_64) |
| HP StoreVirtual 4730 iSCSI (DMMP Recovery Kit を使用したマルチパス構成) | HP StoreVirtual 4730 は、以下の構成で SIOS Technology Corp. のパートナーによってテスト済みです。<br><br>HP StoreVirtual 4730 (ファームウェアHP LeftHand OS 11.5) + HP FlexFabric 10Gb 2-port 536FLB Adapter (Networkdriver bnx2x-1.710.40) + iSCSI (iscsi-initiator-utils-6.2.0.873-10.el6.x86_64)、 DMMP (device-mapper-1.02.79-8.el6,device-mapper-multipath-0.4.9-72.el6) |
| HP StoreVirtual(LeftHand) シリーズ LeftHand OS バージョン 11.5 iSCSI (DMMP Recovery Kitを使用したマルチパス構成) | HP 社 StoreVirtual(LeftHand) ストレージにおいて、LeftHand OS バージョン 11.5 をサポートします。本バージョンが使われている全ての StoreVirtual シリーズがサポート対象であり、仮想ストレージ・アプライアンスである StoreVirtual VSA も含まれます。<br><br>テスト構成は次の通りです。<br><br>StoreVirtual 4730(11.5.00.0673.0) + RHEL 6.5(x86_64) + DMMP (device-mapper-1.02.79-8.el6.x86_64, device-mapper-multipath-0.4.9-72.el6.x86_64) |
| HP StoreVirtual 4330 iSCSI (Quorum/Witness Kit　使用) | HP StoreVirtual 4330 は、 以下の構成で SIOS Technology Corp. のパートナーによってテスト済みです。<br><br>HP StoreVirtual 4330 (ファームウェア HP LeftHand OS 10.5) + iSCSI (iscsi-initiator-utils-6.2.0.872-41.el6.x86_64)、 bonding(バージョン: 3.6.0)、tg3(バージョン: 3.125g) SCSI リザベーションを無効にするには、”/etc/default/LifeKeeper” に RESERVATIONS=noneを設定します。 |
| IBM San Volume Controller (SVC) | シングルパス構成において、パートナーテストにより認定。SDD Recovery Kit および Device Mapper Multipath Recovery Kit を使用するマルチパス設定において、SIOS Technology Corp. により認定。 |

| | |
|---|---|
| IBM Storwize V7000 iSCSI | パートナーテストによりiSCSI (iscsi-initiator-utils-6.2.0.872-34.el6.x86_64) と DMMP (device-mapper-1.02.66-6.el6、device-mapper-multipath-0.4.9-46.el6) を使用する IBM Storwize V7000 (Firmware Version 6.3.0.1)を 認定しています。テストは、LifeKeeper for Linux v7.5 と RHEL 6.2 を使用して行われました。<br><br>**制限事項:** IBM Storwize V7000 は、Quorum/Witness Server Kit および STONITH と組み合わせて使用する必要があります。/etc/default/LifeKeeper 内で以下の設定により、SCSI リザベーションを無効にしてください。<br><br>RESERVATIONS=none |
| IBM Storwize V7000 FC | パートナテストにより Red Hat Enterprise Linux Server Release 6.2 (Tikanga)、HBA: QLE2562 DMMP: 0.4.9-46 を組み合わせたマルチパス構成で認定済み。 |
| IBM Storwize V3700 FC | パートナーテストにより Red Hat Enterprise Linux Server Release 6.5 (Santiago)、HBA: QLE2560、DMMP: 0.4.9-72 を組み合わせたマルチパス構成で認定済み。 |
| IBM XIV Storage System | パートナテストにより Red Hat Enterprise Linux Sever Release 5.6、HBA: NEC N8190-127 Single CH 4Gbps (Emulex LPe1150 相当品)、XIV Host Attachement Kit: バージョン 1.7.0 を組み合わせたマルチパス構成でのみ認定済み。<br><br>**注記:** LifeKeeper を使用する IBM XIV Storage System で 32 個以上の LUN を作成する必要がある場合、詳細については IBM の営業担当者にお問い合わせください。 |
| Dell EqualLogic PS4000/4100/4110/6000/6010/6100/6110/6500/6510 | Dell EqualLogic は、SIOS Technology Corp. パートナーによってテスト済みです。テスト構成は次の通りです。Dell EqualLogic PS4000/4100/6000/6100/6110/6500/6510 + DMMP + DMMP Recovery Kit + RHEL 5.3 + iscsi-initiator-utils-6.2.0.868-0.18.el5。LUN 数が大きい場合 (20 以上) は、/etc/default/LifeKeeper の REMOTETIMEOUT 設定を REMOTETIMEOUT=600 に変更してください。 |
| 富士通<br><br>ETERNUS DX60 S2 / DX80 S2 / DX90 S2<br><br>• iSCSI、シングルパスおよびマルチパス構成 | マルチパス環境にてDMMP ARKを利用する際、 /etc/multipath.conf に以下の設定が必要となります。<br><br>`prio alua`<br><br>`path_grouping_policy group_by_prio` |

| | |
|---|---|
| ETERNUS DX410 S2 / DX440 S2I<br>• iSCSI、シングルパスおよびマルチパス構成<br><br>ETERNUS DX8100 S2/DX8700 S2<br>• iSCSI、シングルパスおよびマルチパス構成<br><br>ETERNUS DX100 S3/DX200 S3<br>• iSCSI、シングルパスおよびマルチパス構成 | |
| ETERNUS DX500 S3/DX600 S3<br>• iSCSI、シングルパスおよびマルチパス構成 | failback immediate<br><br>no_path_retry 10<br><br>Path_checker tur |
| ETERNUS DX200F<br>• iSCSI、シングルパスおよびマルチパス構成<br><br>ETERNUS DX60 S3<br>• iSCSI、シングルパスおよびマルチパス構成<br><br>ETERNUS AF250 / AF650<br>• iSCSI、シングルパスおよびマルチパス構成<br><br>ETERNUS DX60 S4 / DX100 | |

| | |
|---|---|
| S4 / DX200 S4<br>• iSCSI、シングルパスおよびマルチパス構成<br><br>ETERNUS DX500 S4 / DX600 S4 / DX8900S4<br>• iSCSI、シングルパスおよびマルチパス構成<br><br>ETERNUS AF250 S2 / AF650 S2<br>• iSCSI、シングルパスおよびマルチパス構成<br><br>ETERNUS DX60 S5 / DX100 S5 / DX200 S5<br>• iSCSI、シングルパスおよびマルチパス構成<br><br>ETERNUS DX500 S5 / DX600 S5 / DX900S5<br>• iSCSI、シングルパスおよびマルチパス構成<br><br>ETERNUS AF150 S3 / AF250 S3 / AF650 S3<br>• iSCSI、シングルパスおよびマルチパス構成 | |
| 富士通<br><br>ETERNUS DX60 S2 / DX80 S2 / DX90 S2<br>• ファイバチャネル、シングルパスおよびマルチパス構成 | マルチパス環境にてDMMP ARKを利用する際、 /etc/multipath.conf に以下の設定が必要となります。<br><br>`prio alua`<br><br>`path_grouping_policy group_by_prio` |

| | |
|---|---|
| ETERNUS DX410 S2 / DX440 S2<br><br>• ファイバチャネル、シングルパスおよびマルチパス構成<br><br>ETERNUS DX8100 S2/DX8700 S2<br><br>• ファイバチャネル、シングルパスおよびマルチパス構成 | |
| ETERNUS DX100 S3/DX200 S3/DX500 S3/DX600S3<br><br>• ファイバチャネル、シングルパスおよびマルチパス構成 | `failback immediate`<br><br>`no_path_retry 10`<br><br>`Path_checker tur` |
| ETERNUS DX200F<br><br>• ファイバチャネル、シングルパスおよびマルチパス構成 | マルチパス環境にてETERNUSマルチパスドライバーを利用する際、設定ファイルにパラメータを設定する必要はありません。 |
| ETERNUS DX60 S3<br><br>• ファイバチャネル、シングルパスおよびマルチパス構成<br><br>ETERNUS DX8700 S3 / DX8900 S3<br><br>• ファイバチャネル、シングルパスおよびマルチパス構成 | |

| | |
|---|---|
| ETERNUS AF250 / AF650<br>• ファイバチャネル、シングルパスおよびマルチパス構成<br><br>ETERNUS DX60 S4 / DX100 S4 / DX200 S4<br>• ファイバチャネル、シングルパスおよびマルチパス構成<br><br>ETERNUS DX500 S4 / DX600 S4 / DX8900S4<br>• ファイバチャネル、シングルパスおよびマルチパス構成<br><br>ETERNUS AF250 S2 / AF650 S2<br>• ファイバチャネル、シングルパスおよびマルチパス構成<br><br>ETERNUS DX60 S5 / DX100 S5 / DX200 S5<br>• ファイバチャネル、シングルパスおよびマルチパス構成<br><br>ETERNUS DX500 S5 / DX600 S5 / DX900S5<br>• ファイバチャネル、シングルパスおよびマルチパス構成 | |

| | |
|---|---|
| ETERNUS AF150 S3 / AF250 S3 / AF650 S3<br><br>• ファイバチャネル、シングルパスおよびマルチパス構成 | |
| NEC iStorage M10e iSCSI (LifeKeeper Recovery Kit を使用したマルチパス構成) | NEC iStorage M10e iSCSI は、以下の構成で SIOS Technology Corp. のパートナーによってテスト済みです。<br><br>NEC iStorage M10e iSCSI + 1GbE NIC + iSCSI (iscsi-initiator-utils-6.2.0.873-10.el6.x86_64)、SPS (sps-utils-5.3.0-0.el6,sps-driver-E-5.3.0-2.6.32.431.el6) |
| NEC iStorage Storage Path Savior Multipath I/O | **マルチパスデバイスを使用するアプリケーションとファイルシステムの保護:** LifeKeeperデバイスを使用するアプリケーションやファイルシステムを LifeKeeper によって設定し保護するには、LifeKeeper Recovery Kit をインストールする必要があります。<br><br>LifeKeeperKit のインストール後は、1 つ以上のマルチパスデバイスノードを使用するアプリケーション階層を作成するだけで、LifeKeeper Kit が提供する新しいリソースタイプが自動的に組み込まれます。<br><br>**マルチパスデバイスノード:** LifeKeeper Kit を使用するには、すべてのファイルシステムおよび Raw デバイスをネイティブの /dev/sd* デバイスノードではなく、マルチパスデバイスノード (/dev/dd*) 上にマウントまたは設定する必要があります。<br><br>**SCSI-3 Persistent Reservations の使用:** LifeKeeperKit は、リザベーションタイプを「書き込み専用」とする SCSI-3 Persistent Reservations を使用します。この場合、クラスタのあるノードが予約したデバイスは、クラスタの他のノードから読み取り可能のままですが、他のノードからデバイスへの書き込みはできなくなります。このことは、それらの他のノード上で進行中の読み取り専用アクセスのためにファイルシステムをマウントできるという意味ではないことに注意してください。<br><br>LifeKeeper では、sg_persist ユーティリティを使用してパーシステントリザベーションを発行し、監視します。必要であれば、LifeKeeper は sg_persist(8) ユーティリティをインストールします。<br><br>**テスト環境:** LifeKeeper Kit は、Emulex HBA および Emulex lpfc ドライバーを使用する NEC iStorage ディスクアレイにおいて、テストおよび認定済みです。LifeKeeper Kit は、LifeKeeper がサポートする他の NEC iStorage D、S、および M でも同様に問題なく動作すると考えられま |

| | |
|---|---|
| | す。<br><br>**[テストされた Emulex HBA]**<br><br>iStorage D-10<br>==========<br>LP952<br>LP9802<br>LP1050<br>LP1150<br>==========<br><br>iStorage M100<br>==========<br>LPe1150<br>LPe11002<br>LPe1250<br>LPe12002<br>LPe1105<br>LPe1205<br>==========<br><br>**マルチパスソフトウェアの要件:** LifeKeeper Kit は、LifeKeeper for Linux 3.3.001 を使用してテスト済みです。インストールされている LifeKeeper パッケージに対する既知の依存関係はありません。<br><br>**インストール要件:** LifeKeeper for Linux Recovery Kit をインストールする前に LifeKeeper ソフトウェアをインストールする必要があります。<br><br>**LifeKeeper パスの追加または修復:** LifeKeeper は、LifeKeeper リソース起動する場合に、パーシステントリザベーションを確立してその時点でアクティブなパスに登録します。最初のリザベーションの後に新しいパスが追加されるか、障害が起きたパスが修復されて LifeKeeper がそのパスを自動的に再度アクティブにした場合、そのパスは、LifeKeeper が LifeKeeperリソースに対する次の quickCheck を実行するまでリザベーションの一部として登録されません。その時点までに LifeKeeper がそのパスに対する書き込みを許可した場合、リザベーションコンフリクトが発生し、システムのメッセージファイルに競合が記録されます。LifeKeeper ドライバーは、登録されたパスでそれらの I/O を再試行するため、アプリケーションにとっては検出可能な障害になりません。quickCheck によるパスの登録が完了すると、その後の書き込みは成功します。 |
| Pure Storage FA-400 シリーズ FC (DMMP Recovery Kit を使用したマルチパス構成) | パートナーテストにより、DMMP Recovery Kit を利用したFC接続のマルチパス構成で認定済みです。 |

| | |
|---|---|
| QLogic ドライバー | QLogic アダプタを使用するサポート対象の他のファイバチャネルアレイについては、qla2200 または qla2300 ドライバーのバージョン 6.03.00 以降を使用してください。 |
| Emulex ドライバー | サポート対象の Emulex のファイバチャネル HBA については、lpfc ドライバー v8.0.16 以降を使用してください。 |
| Adaptec 29xx ドライバー | Adaptec 29xx を使用するサポート対象の SCSI アレイについては、OS ディストリビューションに付属の aic7xxx ドライバーのバージョン 6.2.0 以降を使用してください。 |

## HP のマルチパス I/O 設定

| 項目 | 説明 |
|---|---|
| Secure Path を使用するマルチパスクラスタのインストール | Secure Path を使用するマルチパスクラスタを新規にインストール場合は、次の手順を実行します。<br><br>1. 選択した OS を各サーバにインストールします。<br>2. クラスタハードウェア (FCA2214 アダプタ、ストレージ、スイッチ、およびケーブル) をインストールします。<br>3. HP Platform Kit をインストールします。<br>4. HP Secure Path ソフトウェアをインストールします。ここでシステムをリブートする必要があります。Secure Path からストレージへのパスを適切に設定したことを確認してください。詳細については、Secure Path のドキュメンテーションを参照してください。<br>5. LifeKeeper をインストールします。 |
| Secure Path による永続的デバイスノード | Secure Path は、/dev/spdev/spXX (XX はデバイス名) の形式の「永続的な」デバイスノードをサポートします。これらのノードは、特定の SCSI デバイスノード /dev/sdXX へのシンボリックリンクです。LifeKeeper はこれらのノードを「通常の」SCSI デバイスノード /dev/sdXX であるかのように認識します。LifeKeeper は、デバイスが /dev/sda1 か /dev/sdq1 かを直接検出し、その後正しいデバイスノードを直接使用することにより、リブートおよびクラスタノードをまたがってデバイス名の永続性を独自に維持しています。<br><br>**注記:** SCSI デバイスノードへのシンボリックリンクのサポートは、LifeKeeper v4.3.0 で追加されました。 |
| アクティブ / パッシブコントローラおよびコントローラスイッチオーバ | MSA1000 では、一方のコントローラをアクティブに、他方のコントローラをスタンバイモードにすることによってマルチパスを実装しています。アクティブなコントローラまたはアクティブなコントローラへのパスのいずれかに問題が起きた場合、スタンバイコントローラがアクティブ化されて処理を引き継ぎます。コントローラをアクティブにする場合、コントローラの準備ができるまでにある程度の時間がかかります。アレイ上で設定されている LUN の数に応じて、30 ~ 90 秒の時間を必要とします。この間、ストレージへの I/O は、新しくアクティブになるコントローラに経路変更できるようになるまでブロックさ |

| | |
|---|---|
| | れます。 |
| 起動時にシングルパスでも通知が発生しない | システムがロードされたときに、サーバがシングルパスでしかストレージにアクセスできない場合でも、この問題に関する通知が発生しません。この問題は、システムがリブートしたときに上記のような物理的なパスの障害が起きると発生しますが、一時的なパス障害でも発生しています。システムをロードするときは、管理者はストレージへのすべてのパスが正しく構成されたことを必ず確認し、構成されていない場合は、ハードウェアの問題を修復するか、システムをリロードして一時的な問題を解決するかいずれかのアクションを取ることを推奨します。 |

## 日立 HDLM のマルチパス I/O 設定

| 項目 | 説明 |
|---|---|
| マルチパスデバイスを使用するアプリケーションとファイルシステムの保護 | HDLM デバイスを使用するアプリケーションやファイルシステムを LifeKeeper によって設定し保護するには、HDLM Recovery Kit をインストールする必要があります。<br><br>HDLM Kit のインストール後は、マルチパスデバイスノードを 1 つ以上使用するアプリケーション階層を作成するだけで、HDLM Kit が提供する新しいリソースタイプが自動的に組み込まれます。 |
| マルチパスデバイスノード | HDLM Kit を使用するには、すべてのファイルシステムおよび Raw デバイスをネイティブの /dev/sd* デバイスノードではなく、マルチパスデバイスノード (/dev/sddlm*) 上にマウントするか、設定する必要があります。 |
| SCSI-3 Persistent Reservations の使用 | HDLM Kit は、リザベーションタイプを「書き込み専用」とする SCSI-3 Persistent Reservations を使用します。この場合、クラスタのあるノードが予約したデバイスは、クラスタの他のノードから読み取り可能のままですが、他のノードからデバイスへの書き込みはできなくなります。このことは、それらの他のノード上で進行中の読み取り専用アクセスのためにファイルシステムをマウントできるという意味ではないことに注意してください。<br><br>LifeKeeper では、sg_persist ユーティリティを使用してパーシステントリザベーションを発行し、監視します。必要であれば、LifeKeeper は sg_persist(8) ユーティリティをインストールします。 |
| ハードウェア要件 | HDLM Kit は、QLogic qla2432 HBA と 8.02.00-k5-rhel5.2-04 ドライバー、および Silkworm3800 FC スイッチを使用する 日立 SANRISE AMS1000 ディスクアレイにおいて、テストおよび認定済みです。HDLM Kit は、他の日立ディスクアレイでも同様に問題なく動作すると考えられます。HDLM Kit は、SANRISE AMS シリーズ、SANRISE USP、および日立の VSP においても、テストおよび認定済みです。HBA および HBA ドライバーは HDLM がサポートするものを使用してください。 |

| | |
|---|---|
| | BR1200 は、Hitachi Data Systems により認定済みです。シングルパスとマルチパスの両方の設定で、RDAC ドライバーが必要です。RDAC ドライバーを使用する BR1200 設定のみをサポートします。HDLM (HDLM ARK) を使用する BR1200 設定はサポートしません。 |
| マルチパスソフトウェアの要件 | HDLM Kit は、以下の HDLM for Linux でテスト済みです。<br><br>8.0.1, 8.1.0, 8.1.1, 8.1.2, 8.1.3, 8.1.4, 8.2.0, 8.4.0, 8.5.0, 8.5.1, 8.5.2, 8.5.3, 8.5.4, 8.6.0, 8.6.1, 8.6.2, 8.6.4, 8.6.5, 8.7.0, 8.7.1, 8.7.2, 8.7.3, 8.7,4, 8.7.6, 8.7.7, 8.7.8, 8.8.0, 8.8.1, 8.8.3, 8.8.4<br><br>インストールされている HDLM パッケージに対する既知の依存関係はありません。<br><br>**注記:** LVM と HDLM をともに使用する場合、HDLM がサポートするバージョンが必要です。また、フィルタ設定を /etc/lvm/lvm.conf に追加して、システムが /dev/sddlm* に対応する /dev/sd* を検出しないようにする必要があります。詳細については、HDLM のマニュアルの「LVM Configuration」を参照してください。 |
| Linux ディストリビューションの要件 | Linux ディストリビューションの要件<br><br>HDLM は以下のディストリビューションでサポートされています。<br><br>RHEL 7、7.1、7.2、7.3、7.4、7.5、7.6、7.7、7.8、7.9 (x86_64(*1)国内、海外共)<br><br>RHEL 8.1、8.2、8.3 、8.4 (x86_64(*1)国内、海外共)<br><br>(*1) AMD Opteron(シングルコア、デュアルコア) あるいは Intel EM64T アーキテクチャ CPU + x86_64 カーネル<br><br>(*2) 下記のカーネルを示します。<br>x86_64:3.10.0-123.13.2.el7.x86_64, 3.10.0-123.20.1.el7.x86_64<br><br>(*3) 下記のカーネルを示します。<br>x86_64:3.10.0-229.4.2.el7.x86_64,3.10.0-229.20.1.el7.x86_64, 3.10.0-229.34.1.el7.x86_64<br><br>(*4) 下記のカーネルを示します。<br>x86_64:3.10.0-327.4.4.el7.x86_64, 3.10.0-327.4.5.el7.x86_64, 3.10.0-327.10.1.el7.x86_64, 3.10.0-327.18.2.el7.x86_64,<br>3.10.0-327.22.2.el7.x86_64, 3.10.0-327.36.1.el7.x86_64, 3.10.0-327.36.3.el7.x86_64,<br>3.10.0-327.44.2.el7.x86_64, 3.10.0-327.46.1.el7.x86_64, 3.10.0-327.49.2.el7.x86_64，3.10.0-327.55.2.el7.x86_64，3.10.0-327.55.3.el7.x86_64，3.10.0-327.58.1.el7.x86_64, 3.10.0-327.62.1.el7.x86_64，3.10.0-327.62.4.el7.x86_64, 3.10.0-327.64.1.el7.x8 |

| | |
|---|---|
| | 6_64, 3.10.0-327.93.1.el7.x86_64, 3.10. 0 327. 9 6 .1 .el 7 .x86_64, 3.10.0-327.98.2.el7.x86_64，3.10.0-327.102.1.el7.x86_64<br><br>(*5)下記のカーネルを示します。<br>x86_64：3.10.0-514.6.1.el7.x86_64，3.10.0-514.10.2.el7.x86_64，3.10.0-514.16.1.el7.x86_64，3.10.0-514.21.1.el7.x86_64，3.10.0-514.26.2.el7.x86_64，3.10.0-514.36.5.el7.x86_64，3.10.0-514.44.1.el7.x86_64, 3.10.0-514.51.1.el7.x86_64, 3.10.0-514.55.4.el7.x86_64<br><br>(*6)下記のカーネルを示します。<br>x86_64：3.10.0-693.1.1.el7.x86_64, 3.10.0-693.5.2.el7.x86_64, 3.10.0-693.11.1.el7.x86_64, 3.10.0-693.11.6.el7.x86_64, 3.10.0-693.21.1.el7.x86_64, 3.10.0-693.43.1.el7.x86_64, 3.10.0-693.33.1.el7.x86_64, 3.10.0-693.46.1.el7.x86_64, 3.10.0-693.62.1.el7.x86_64<br><br>(*7)下記のカーネルを示します。<br>x86_64：3.10.0-862.3.2.el7.x86_64, 3.10.0-862.14.4.el7.x86_64<br><br>(*8)下記のカーネルを示します。<br>x86_64：3.10.0-957.10.1.el7.x86_64，3.10.0-957.12.2.el7.x86_64,<br>3.10.0-957.21.3.el7.x86_64，3.10.0-957.27.2.el7.x86_64<br><br>(*9)下記のカーネルを示します。<br>x86_64：3.10.0-1062.1.1.el7.x86_64, 3.10.0-1062.9.1.el7.x86_64, 3.10.0-1062.18.1.el7.x86_64, 3.10.0-1062.56.1.el7.x86_64, 3.10.0-1062.60.1.el7.x86_64, 3.10.0-1062.63.1.el7.x86_64, 3.10.0-1062.67.1.el7.x86_64<br><br>(*10)下記のカーネルを示します。<br>x86_64：4.18.0-147.5.1.el8_1.x86_64，4.18.0-147.8.1.el8_1.x86_64<br><br>(*11)下記のカーネルを示します。<br>x86_64：4.18.0-193.13.2.el8_2.x86_64，4.18.0-193.28.1.el8_2.x86_64，4.18.0-193.79.1.el8_2.x86_64, 4.18.0-193.95.1.el8_2.x86_64<br><br>(*12) iSCSI 環境のみがサポートされています。<br><br>(*13)下記のカーネルを示します。<br>x86_64：4.18.0-240.22.1.el8_3.x86_6<br><br>(*14) 下記のカーネルを示します。<br>x86_64：3.10.0-1160.15.2.el7.x86_64, 3.10.0-1160.31.1.el7.x86_64 ×86_64, 3.10.0-1160.32.2.el7.x86_64, 3.10.0-1160.59.1.el7.x86_64, 3.10.0-1160.76.1.el7.x86_64, 3.10.0-1160.80.1.el7.x86_64， 3.10.0-1160.83.1.el7.x86_64<br><br>(*15) 下記のカーネルを示します。<br>x86_64：4.18.0-305.3.1.el8_4.x86_64，4.18.0-305.19.1.el8_4.x86_64，4.18.0-305.2 |

| | |
|---|---|
| | 5.1.el8_4.x86_64 |
| インストール要件 | HDLM Recovery Kit をインストールする前に HDLM ソフトウェアをインストールする必要があります。また、SCSI デバイスから HDLM デバイスに環境を転送したい場合は、HDLM 環境を設定した後、インストールセットアップスクリプトを実行する必要があります。そのようにしないと、sg3_utils がインストールされません。 |
| HDLM パスの追加または修復 | LifeKeeper は、HDLM リソースを起動する場合、パーシステントリザベーションを確立してその時点でアクティブなパスに登録します。最初のリザベーションの後に新しいパスが追加されるか、障害が起きたパスが修復されて HDLM がそのパスを自動的に再度アクティブにした場合、そのパスは、LifeKeeper が HDLM リソースに対する次の quickCheck を実行するまでリザベーションの一部として登録されません。その時点までに HDLM がそのパスに対する書き込みを許可した場合、リザベーションコンフリクトが発生し、システムのメッセージファイルに競合が記録されます。HDLM ドライバーは、登録されたパスでそれらの I/O を再試行するため、アプリケーションにとっては検出可能な障害になりません。quickCheck によるパスの登録が完了すると、その後の書き込みは成功します。quickCheck がリザベーションコンフリクトを検出すると、ステータスが「Offline(E)」に変更されます。ステータスが「Offline(E)」の場合、ユーザはオンラインの HDLM コマンドを使用して手動でステータスを「Online」に変更する必要があります。 |
| RHEL 7.xの追加設定 | HDLM RecoveryKitをRed Hat Enterprise Linux 7.0以降で使用する場合、 /etc/default/LifeKeeper に HDLM_DLMMGR=.dlmmgr_exe を追記する必要があります。 |

| | | RHEL7 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 7.0 セキュリティフィックス (*2) | 7.1 セキュリティフィックス (*3) | 7.2 セキュリティフィックス (*4) | 7.3 セキュリティフィックス (*5) | 7.4 セキュリティフィックス (*6) | 7.5 セキュリティフィックス (*7) | 7.6 セキュリティフィックス (*8) | 7.7 セキュリティフィックス (*9) | 7.8 セキュリティフィックス | 7.9 セキュリティフィックス(*14) |
| | | x86/x86_64 | | | | | | | | | |
| HDLM | 8.0.1 | X | X | | | | | | | | |
| | 8.1.0 | X | X | | | | | | | | |
| | 8.1.1 | X | X | | | | | | | | |
| | 8.1.2 | X | X | | | | | | | | |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 8.1.3 | X | X | | | | | | | | |
| | 8.1.4 | X | X | | | | | | | | |
| | 8.2.0 | X | X | | | | | | | | |
| | 8.2.1 | X | X | | | | | | | | |
| | 8.4.0 | X | X | X | | | | | | | |
| | 8.5.0 | X | X | X | | | | | | | |
| | 8.5.1 | X | X | X | X | X | | | | | |
| | 8.5.2 | X | X | X | X | X | | | | | |
| | 8.5.3 | X | X | X | X | X | | | | | |
| | 8.5.4 | X | X | X | X | X | | | | | |
| | 8.6.0 | X | X | X | X | X | | | | | |
| | 8.6.1 | X | X | X | X | X | X | | | | |
| | 8.6.2 | X | X | X | X | X | X | X（注意: 3） | | | |
| | 8.6.4 | X | X | X | X | X | X | X | | | |
| | 8.6.5 | X | X | X | X | X | X | X | | | |
| | 8.7.0 | X | X | X | X | X | X | X | X | | |
| | 8.7.1 | X | X | X | X | X | X | X | X | | |
| | 8.7.2 | X | X | X | X | X | X | X | X | | |
| | 8.7.3 | X | X | X | X | X | X | X | X | X | X |
| | 8.7.4 | X | X | X | X | X | X | X | X | X | X |
| | 8.7.6 | X | X | X | X | X | X | X | X | X | X |
| | 8.7.7 | X | X | X | X | X | X | X | X | X | X |
| | 8.7.8 | X | X | X | X | X | X | X | X | X | X |
| | 8.8.0 | X | X | X | X | X | X | X | X | X | X |
| | 8.8.1 | X | X | X | X | X | X | X | X | X | X |
| | 8.8.3 | X | X | X | X | X | X | X | X | X | X |
| | 8.8.4 | X | X | X | X | X | X | X | X | X | X |
| LifeKeeper (HDML ARK) | v9.5.0 | X | X | X | X | X | X | X | X | X | |
| | v9.5.1 | X | X | X | X | X | X | X | X | X | X |
| | v9.5.2 | X | X | X | X | X | X | X | X | X | X |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | v9.6.0 | X | X | X | X | X | X | X | X | X | X |
| | v9.6.1 | X | X | X | X | X | X | X | X | X | X |
| | v9.6.2 | X | X | X | X | X | X | X | X | X | X |
| | v9.7.0 | X | X | X | X | X | X | X | X | X | X |
| | X = サポートあり、空白 = サポートなし | | | | | | | | | | |

(注): Lifekeeper v9.0.x と RHEL7/7.1/7.2 の組み合わせは、Bug7205 のパッチの適用が必要です。

(注 2): RHEL7/7.1/7.2/7.3/7.4/7.5/7.6/7.7/7.8/7.9 では、Raw デバイスを使用した構成は未サポートです。

(注 3): HDLM 8.6.2-02 以降でのサポートです。

| | | **RHEL8** | | | |
|---|---|---|---|---|---|
| | | 8.1 セキュリティフィックス (*10) | 8.2 セキュリティフィックス (*11)(*12) | 8.3 セキュリティフィックス(*13) | 8.4 セキュリティフィックス(*15) |
| | | **x86/x86_64** | | | |
| HDLM | 8.7.2 | X | | | |
| | 8.7.3 | X | | | |
| | 8.7.4 | X | X | X | |
| | 8.7.6 | X | X | X | |
| | 8.7.7 | X | X | X | |
| | 8.7.8 | X | X | X | |
| | 8.8.0 | X | X | X | |
| | 8.8.1 | X | X | X | X |
| | 8.8.3 | X | X | X | X |
| | 8.8.4 | X | X | X | X |
| LifeKeeper(HDLM ARK) | v9.5.1 | X | X | X | |
| | v9.5.2 | X | X | X | X |
| | v9.6.0 | X | X | X | X |
| | v9.6.1 | X | X | X | X |

| | | | | | |
|---|---|---|---|---|---|
| | v9.6.2 | X | X | X | X |
| X = サポートあり、空白 = サポートなし | | | | | |

(注):RHEL8.1, RHEL8.2, RHEL8.3，RHEL8.4 では，Raw デバイスを使用した構成は未サポートです。

## Device Mapper のマルチパス I/O 設定

| | |
|---|---|
| Device Mapper Multipath デバイスを使用するアプリケーションとファイルシステムの保護 | Device Mapper Multipath デバイスを使用するアプリケーションやファイルシステムを LifeKeeper によって設定し保護するには、Device Mapper Multipath (DMMP) Recovery Kit をインストールする必要があります。<br><br>DMMP Kit のインストール後は、1 つ以上のマルチパスデバイスノードを使用するアプリケーション階層を作成するだけで、DMMP Kit が提供する新しいリソースタイプが自動的に組み込まれます。 |
| マルチパスデバイスノード | DMMP Kit を使用するには、すべてのファイルシステムおよび Raw デバイスをネイティブの /dev/sd* デバイスノードではなく、マルチパスデバイスノード上にマウントまたは設定する必要があります。ディスク全体を利用できるサポート対象のマルチパスデバイスノードは、/dev/dm-#、/dev/mapper/<uuid>、/dev/mapper/<user_friendly_name> および /dev/mpath/<uuid> です。ディスクのパーティションに対応するには、/dev/mapper ディレクトリに作成される各パーティション用のデバイスノードを使用します。 |
| SCSI-3 Persistent Reservations の使用 | Device Mapper Multipath Recovery Kit は、リザベーションタイプを「書き込み専用」とする SCSI-3 Persistent Reservations を使用します。この場合、クラスタのあるノードが予約したデバイスは、クラスタの他のノードから読み取り可能のままですが、他のノードからデバイスへの書き込みはできなくなります。このことは、それらの他のノード上で進行中の読み取り専用アクセスのためにファイルシステムをマウントできるという意味ではないことに注意してください。<br><br>LifeKeeper では、sg_persist ユーティリティを使用してパーシステントリザベーションを発行、監視します。必要であれば、LifeKeeper は sg_persist(8) ユーティリティをインストールします。<br><br>EMC Symmetrix (VMAX を含む) アレイをマルチパスソフトウェアおよび LifeKeeper と組み合わせて使用する場合は、SCSI-3 Persistent Reservations を LUN 単位で有効にする必要があります。これは、DMMP と PowerPath の両方に当てはまります。 |
| ハードウェア要件 | Device Mapper Multipath Kit は、EMC CLARiiON CX300、HP EVA 8000、HP MSA1500、HP P2000、IBM SAN Volume Controller (SVC)、IBM DS8100、IBM DS6800、IBM ESS、DataCore SANsymphony、HDS 9980V を使用して SIOS Technology Corp. によりテスト済みです。Device Mapper Multipath のサポートについては、ストレージベンダにお問い合わせください。 |

| | |
|---|---|
| | CX300 およびVNXシリーズでリザベーションのサポートを有効にするには、リザベーションに従うようにハードウェアハンドラに通知する必要があります。このアレイ用に */etc/multipath.conf* 内に次のパラメータを設定してください。<br><br>`hardware_handler "3 emc 0 1"`<br><br>HP MSA1500 の場合、デフォルトのパスチェッカ (tur) とのリザベーションコンフリクトを返します。これによりスタンバイノードは、すべてのパスを障害であると判定します。この状態を回避するには、このアレイ用に */etc/multipath.conf* 内に次のパラメータを設定してください。<br><br>`path_checker readsector0`<br><br>HP 3PAR F400 は、デフォルトのパスチェッカとのリザベーション競合を返します。この競合を回避するには、このアレイ用に */etc/default/LifeKeeper* 内に次のパラメータを設定 (追加) してください。<br><br>`DMMP_REGISTRATION_TYPE=hba`<br><br>HDS 9980V の場合、以下の設定が必要です。<br><br>• Host mode: 00<br>• System option: 254 (有効にする必要がある。すべてのサーバに影響を与えるグローバルな HDS 設定)<br>• Device emulation: OPEN-V<br><br>HDS の DMMP 設定の詳細については、HDS ドキュメンテーション「Suse Linux Device Mapper Multipath for HDS Storage」または「Red Hat Linux Device Mapper Multipath for HDS Storage」の v1.15 以降を参照してください。このドキュメンテーションでは、互換性のある multipath.conf ファイルも提供しています。<br><br>EVAストレージでDMMPリカバリキットを使用するには、ファームウェアバージョン6以上が必要です。 |
| マルチパスソフトウェアの要件 | ・SUSE の場合、multipath-tools-0.4.5-0.14 以降が必要です。<br><br>・Red Hat の場合、device-mapper-multipath-0.4.5-12.0.RHEL4 以降が必要です。 |
| 一時的なパス障害 | Device Mapper Multipath デバイスで I/O テストを実行中に、サーバのリブートなどの SAN 上の操作によって一時的なパスの障害が報告されることは珍しくありません。ほとんどの場合、結果として単に 1 つのパスだけが障害となり他のパスは I/O を送信するため、パフォーマンスへのわずかな影響以外に検出される障害はありません。ただし一部の |

ケースでは、複数のパスが障害として報告され、機能するパスがまったくない状態になることがあります。この状態では、ファイルシステムやデータベースなどのアプリケーションからは I/O エラーが発生しているように見えます。このような障害を排除する上で、Device Mapper Multipath およびベンダのサポートはこれまでに大きく改善されました。ただし、まだ問題が発生することはあります。このような状況を回避するため、以下の措置を検討してください。

1. ディスクアレイベンダの手順に従ってマルチパス構成が正しく設定されていることを確認します。
2. 「failback」機能の設定を確認します。この機能は、パスの障害および修復後にパスを再度アクティブにするまでの時間を指定します。「immediate」に設定した場合、パスがオンラインに戻るとすぐに使用を再開することを意味します。整数に設定した場合、パスがオンラインに戻ってから使用を再開するまでの秒数を意味します。10 ~ 15 に設定すると、一般的に SAN 上のスラッシングを回避するのに十分な処理時間が得られます。
3. 「no_path_retry」機能の設定を確認します。この機能は、すべてのパスに障害が発生したときに Device Mapper Multipath がやるべきことを指定します。10 ~ 15 に設定することを推奨します。この機能によって、すべてのパスに障害が起きた一時的なイベントを「乗り切る」方法が提供され、復旧に必要な妥当な時間を稼ぐことができます。LifeKeeper の DMMP Kit はストレージへの I/O を監視しており、4 分以内に応答がなかった場合、LifeKeeper はスタンバイサーバにリソースをスイッチオーバします。注記: 簡単には削除できない I/O が発生するため、LifeKeeper では「no_path_retry」設定を「queue」に設定することは推奨されません。それらを削除するためのメカニズムは、新しいバージョンの DM に含まれており、デバイスの設定を次のように変更できます。

```
/sbin/dmsetup message -u 'DMid' 0 fail_if_no_path
```

これによって no_path_retry の設定が一時的に「fail」に変更され、未処理の I/O がすべて失敗します。ただし、multipathd は、no_path_retry をいつでもデフォルトにリセットできます。失敗した I/O を消去するために設定が fail_if_no_path に変更されたときは、デバイスにアクセスする前に (手動または LifeKeeper によって) 設定をデフォルトにリセットする必要があります。

「no_path_retry」が「queue」に設定されていて障害が発生した場合、LifeKeeper はリソースをスタンバイサーバにスイッチオーバします。ただし、LifeKeeper は失敗した I/O を削除しません。この I/O を消去するための推奨の方法はリブートですが、上記の dmsetup コマンドを使用して管理者が消去することもできます。I/O を消去しておかないと、他方のサーバでリソースがサービス休止状態になってロックを解放した場合にこの「古い」I/O が発行される事態になってデータの破損が発生します。

# 5.4.2.10. LifeKeeper I/O フェンシングの概要

I/Oフェンシングは、障害ノードをデータから切り離すことにより、共有ストレージへの非協調的なアクセスを防止する機能です。複数のサーバが同じデータにアクセスできる環境では、データの破損を防ぐためにすべての書き込みを制御された方法で行うことが不可欠です。障害検知メカニズムが破綻した場合、この破綻によってノード障害に似た状況になるため問題が発生します。例えば、2 ノードクラスタで 2 つのノード間の接続に障害が発生した場合、各ノードは相手側に障害が発生したと「思い込む」ため、両方のノードがデータに対する制御を獲得しようと試みてデータの破損につながります。I/O フェンシングは、特定のノードからのデータアクセスをブロックすることによりこのデータ破損のリスクを排除します。

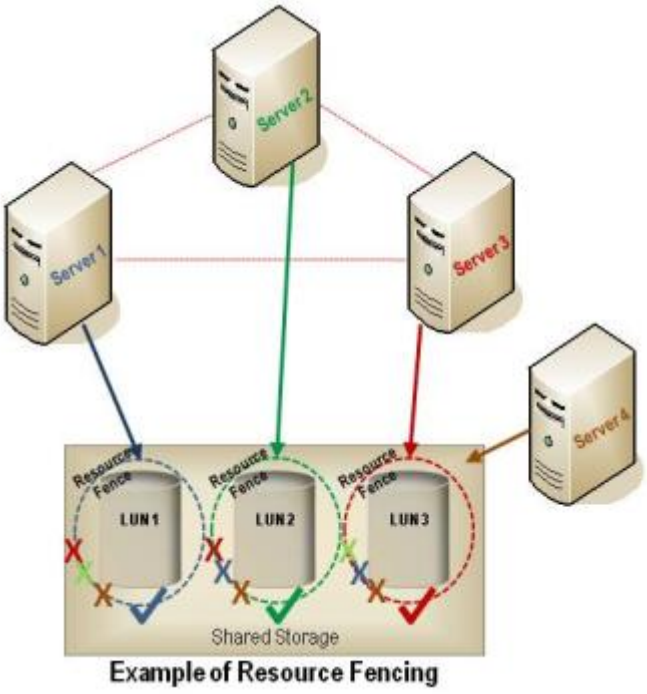
なお、DataKeeper 構成では、DataKeeper 構成での I/O フェンシング を参照して下さい。

# 5.4.2.10.1. SCSI リザベーション

## SCSI リザベーションを利用したストレージフェンシング

LifeKeeper for Linux は、リソースフェンシングとノードフェンシングの両方をサポートしますが、主要なフェンシングメカニズムは、SCSI リザベーションによるストレージフェンシングです。共有ストレージに対する最高レベルのデータ保護を提供するこのフェンシングを使用すると、非常に粒度の高い LUN レベルのロックによって最大限の柔軟性と最大限のセキュリティが可能になります。このアーキテクチャでベースとなる共有リソース (LUN) は、プライマリ quorum デバイスです。 quorum は、共有ストレージに対する排他的なアクセスと定義できます。つまり、この共有ストレージは 1 度に 1 台のサーバーからしかアクセスできません。 quorum (排他的アクセス) を持つサーバーは、「プライマリ」の役割を持ちます。quorum の確立 (排他的アクセスをどのサーバーに与えるか) は、「quorum デバイス」によって行われます。

上述の通り、リザベーションが有効の場合、quorum デバイスはその共有リソースです。共有リソースは、共有リソースに対するリザベーションを持つサーバーを判断して quorum を確立します。これにより、ある 1 つのサーバーがその LUN にアクセスできる限り、クラスターは実質的には単一のサーバーで運用されることになります。



Example of Resource Fencing

SCSI リザベーションは、共有のユーザデータを保護し、LifeKeeper が指定するシステムのみデータを変更できるようにします。クラスター内外の他のシステムがそのデータを変更することは許可されません。さらに SCSI リザベーションによって、クラスター内の複数のサーバーで障害が起きた場合に、LifeKeeper 保護下のアプリケーションは共有のユーザデータに安全にアクセスできます。サーバーの多数派 quorum は必要ありません。唯一の要件は、共有データの所有権の帰属が確立していることです。

quorum/witness 機能を追加すると、 quorum のメンバーシップを確立することができます。このメンバーシップがない場合、スプリットブレインの状況で複数のサーバー (場合によっては全サーバー) がお互いを終了させることも考えられます。リザベーションが有効になっている構成に Watchdog を追加すると、部

分的にサーバーがハングしている状態からリカバリするメカニズムが提供されます。ハングしたサーバーが LifeKeeper に検出されないような場合に、Watchdog はリカバリを開始します。また、サーバーがハングしてさらにリザベーションが奪われた場合に、Watchdog はそのサーバーをリブートしてリカバリを開始することができます。

## I/O フェンシングのための代替方式

SCSI リザベーションを利用したリソースフェンシングに加えて、LifeKeeper for Linux はリザベーションの無効化もサポートします。リザベーションの有効無効にかかわらず、以下の 2 つの点に注意すべきです。

- ストレージへのアクセスは LifeKeeper が制御する必要があります。
- ストレージへの意図しないアクセス (ファイルシステムのマウント、手動の fsck など) が発生しないように細心の注意を払う必要があります。

以上の 2 つのルールを順守してリザベーションを有効にすると、LifeKeeper はたいていのエラーを防止できます。リザベーションが (単独で) 無効になった状態は、保護がない状態です。したがって、この保護を実現するには、他の選択肢を検討する必要があります。以降のセクションでは、SCSI リザベーションなしでも LifeKeeper で非常に信頼性の高い構成を実現できる各種のフェンシングオプションと代替方式を説明します。

# 5.4.2.10.2. リザベーションの無効化

リザベーションを使用すると、共有ストレージに対する最高レベルのデータ保護が可能になりますが、場合によっては、リザベーションを使用できず LifeKeeper 内で無効にしなければならないことがあります。リザベーションを無効にすると、複数のシステムが意図的または意図せずにストレージにアクセスしようとする場合にストレージが調停役として動作することがなくなります。

そのため、システムハング、システムビジー、またはサーバーが停止したように見えるあらゆる状況に対応できるように、クラスターメンバーシップによってストレージをフェンシングする別の方法を採用することを検討する必要があります。

リザベーションがなくても信頼性の高い構成を実現する鍵は、フェイルオーバが発生したとき、「他の」サーバーの電源がオフになったこと、または電源が再投入されたことを「知る」ことです。この要件を満たすために利用可能なフェンシングオプションは 4 つあり、これらは SCSI リザベーションなしでも LifeKeeper で非常に信頼性の高い構成を実現できます。オプションは以下の通りです。

- STONITH (Shoot the Other Node in the Head) (高信頼性のインターコネクト、すなわちサーバーと STONITH デバイスとの間のシリアル接続を使用) - STONITH は、サーバーがクラスターの一部とみなされなくなったときに、そのサーバーを物理的に停止させたり、電源を切断したりする技術です。LifeKeeper フェイルオーバのイベント時にサーバーの電源を切断する機能をサポートしています。これにより共有データへの安全なアクセスを保証します。このオプションはリザベーションと同様の信頼性を提供できますが、利用できるのは物理的に同じ場所に配置された 2 つのノードに限定されます。

- Quorum/Witness - Quorum/Witness サーバーは、特にクラスターサーバーが異なる場所に配置されている場合に、クラスター内でのメンバーシップを確認するために使用されます。このオプションはスプリットブレインに対応できるもののシステムハングに対応できないため、単独での使用は推奨されません。

- Watchdog - Watchdog は、サーバーの状態を監視します。問題が検出されると、問題のあるサーバーは再起動または電源を切断されます。このオプションではサーバーのハングからのリカバリはできますが、スプリットブレインには対応できません。したがって、このオプションもまた単独での使用は推奨されません。

- CONFIRM_SO - このオプションでは自動フェイルオーバを無効にする必要があります。そのため、信頼性が非常に高い (管理者のスキルによって異なります) 一方で、可用性はあまり高くありません。

これらの代替フェンシング方式はどれも単独では十分とは言えませんが、組み合わせて使用することで非常に信頼性の高い構成を実現できます。

## 非共有ストレージ

非共有ストレージ環境で LifeKeeper を使用する計画の場合、共有ストレージに存在するデータ破損のリスクは問題にならないため、リザベーションは不要です。ただし、データの部分的または完全な再同期およびマージが必要な場合があります。信頼性と可用性を最適化するには、非共有ストレージでも上記のオプ

ションを検討する必要があります。

> **注記:** 各オプションの信頼性と可用性の比較の詳細については、I/O フェンシング比較表を参照してください。

完全なデータ保護を実現するオプションはないことを理解することは重要です。ただし、以下のように組み合わせるとリザベーションとほぼ同等レベルの保護を実現できます。

# リザベーションを使用しない I/O フェンシングの設定

ノードフェンシングをサポートするクラスターを構成するには以下の手順を実行します。

1. LifeKeeper を停止します。
2. LifeKeeper 内での SCSI リザベーションの使用を無効にします。無効にするには、クラスターのすべてのノードで LifeKeeper のデフォルトファイル */etc/default/LifeKeeper* を編集します。Reservations 変数を追加または修正して「none」にします ( 例：`RESERVATIONS=none` )。このオプションはリザベーションを利用できない場合のみ使用することに注意してください。
3. I/O フェンシングを提供する STONITH デバイスを用意して設定します。この設定では、STONITH デバイスが reboot コマンドではなく、poweroff コマンドをシステムに対して実行するようにします。LifeKeeper の通信が何らかの理由で中断したとき、手動操作によって同時に両ノード上のデバイス階層を In Service の状態にしないように注意してください。
4. 必要に応じて、quorum/witness サーバーを用意して設定します。quorum/witness サーバーを設定、使用する詳細な手順と情報については、Quorum/Witness Server Support Package トピックを参照してください。

   注記: サイトの障害時に最良の保護機能を提供するため、quorum/witness サーバーはクラスター内の別サーバーから離れた場所にある必要があります。
5. 必要に応じて、Watchdog を設定します。詳細については、 Watchdog トピックを参照してください。

# 共有ストレージへのアクセス

リザベーションの有効無効にかかわらず、以下の 2 つの点に注意すべきです。

- ストレージへのアクセスは LifeKeeper が制御する必要があります。
- ストレージへの意図しないアクセス (ファイルシステムのマウント、手動の fsck など) が発生しないように細心の注意を払う必要があります。

## STANDBY NODE WRITE PROTECTION 機能

リザベーションを無効 ( `RESERVATIONS=none` ) にした場合においても、不注意な意図しないアクセスによるデータ破損を防ぐために、スタンバイノードからディスクデバイスへの書き込みを禁止することがで

きます。この機能を有効にするには、 */etc/default/LifeKeeper* に以下の設定を追加してください。

```
STANDBY_NODE_WRITE_PROTECTION=enable
SNHC=1
SNHC_DISKCHECK=1
```

この機能は Standby Node Health Check を利用しています。そのため、OSU リソース監視の OSUquickCheck が実行されるまでは、書き込み禁止にならないことがあります。スイッチオーバーしてから約 LKCHECKINTERVAL（デフォルト120）秒間は書き込み可能になっていることがあるためご注意ください。

この機能は、スプリットブレインの発生を抑止することはできません。また、スプリットブレインが発生すれば、両ノードで書き込みを許可します。リザベーションを無効 ( `RESERVATIONS=none` ) に設定するときは、クラスター内のデータの信頼性を保証するために、別の方法 で I/O フェンシングを構成する必要があります。

> 注記: Device Mapper Multipath (DMMP) RK、Hitachi Dynamic Link Manager Software (HDLM) RK、SCSI-3 Persistent Reservations (SCSI3) RK、PowerPath RK、NEC iStorage StoragePathSavior (NECSPS) RK で保護された共有ストレージは、STANDBY NODE WRITE PROTECTION 機能に対応していません。

# 5.4.2.10.2.1. I/O フェンシング表

| | スプリットブレイン | ハングしたサーバ |
|---|---|---|
| **Reservations オン** | | |
| **単体** | | |
| **Quorum/Witness** | | |
| **Watchdog** | | |
| **Watchdog & Quorum/ Witness** | | |
| **STONITH (シリアル)** | | |
| **Reservations オフ** | | |
| **なし** | | |
| **STONITH (シリアル)** | | |
| **CONFIRM_SO*** | | |
| **Quorum/Witness** | | |
| **Watchdog** | | |
| **非共有ストレージ** | | |
| **デフォルトの機能** | | |
| **Quorum/Witness** | | |
| **CONFIRM_SO*** | | |
| **Watchdog** | | |
| **STONITH (シリアル)** | | |

* *CONFIRM_SO は信頼性が高いが（管理者の知識による）、自動フェイルオーバがオフになっているという事実により可用性が低い。*

# 5.4.2.10.3. Quorum/Witness

## Quorum/Witness Server Support Package for LifeKeeper

### 機能の概要

LifeKeeper Core の既存のフェイルオーバープロセスに Quorum/Witness Server Support Package for LifeKeeper（steeleye-lkQWK、以下　Quorum/Witness　パッケージ）を組み合わせることにより、ネットワーク全体にわたる障害の恐れがある環境において、より高い信頼度でシステムフェイルオーバーを実行できます。つまり、「スプリットブレイン」の発生リスクを大幅に軽減しながら、ローカルサイトのフェイルオーバーと WAN 越しのノードへのフェイルオーバーを実行することができます。

ネットワーク分断を考慮した分散システムにおいて、クラスター全体での合意を得るために Quorum と呼ばれる概念があります。Quorum を持っているノードはクラスター全体での合意が得られるノードであり、リソースを起動することが許可されます。一方、Quorumを持っていないノードはクラスター全体での合意が得られないノードで、リソースを起動することは許可されません。これにより、スプリットブレインを抑止します。そして、この Quorum を持っているかどうかを確認することを Quorum チェックと呼び、Quorum を持っている場合を Quorum チェック成功、Quorum を持っていない場合を Quorum チェック失敗と表現します。

通信障害が発生した時、障害が発生したノードと別の１台以上のノード（もしくは他のデバイス）を使って障害ノードのステータスについて「セカンドオピニオン」を取得することができます。このセカンドオピニオンを得るために使用されるノードを Witness ノード（もしくは Witness デバイス）と呼びます。また、このセカンドオピニオンを得ることを Witness チェックと呼びます。フェイルオーバー先となることができるノードは、Witness ノード（Witness デバイス）が障害となったノードのステータスに関して同じ意見である場合のみ、リソース起動が許可されます。これにより、ノード間で発生する単純な通信障害から発生するフェイルオーバーを回避し、全体のアクセスや、パフォーマンス、サービス中のノードに影響を与えないようにします。実際の運用では、LifeKeeperが起動した時やコミュニケーションパスが復旧した時にも、Witness ノード（Witness デバイス）に問い合わせを行います。なお、Witness チェックを行うのは、Quorum を持っていることが前提です。

> ✻ SCSI リザベーションによる排他制御を利用できない DataKeeper や NAS などの共有ストレージを使用する場合、Quorum/Witness 機能を使用することでスプリットブレイン状態の発生を抑制してより安全な HA クラスターを構築することができます。

> ✻ Witnessノードのシステム要件は LifeKeeper のシステム要件 と同じです。Witness ノードはレプリケーションに参加せず、保護するリソースもないため、最低要件を満たしていれば動作します。

### パッケージのインストールと設定

Quorum/Witness 機能を利用するクラスターに参加するすべてのノード（Witness 専用ノードを含む）に Quorum/Witness パッケージをインストールする必要があります。Witness ノードに必要な唯一の設定

は、適切なコミュニケーションパスを作成する ことです。Quorum モードを *tcp_remote* で使用する場合、/etc/default/LifeKeeper 設定ファイルの QUORUM_HOSTS に設定したホストには LifeKeeper をインストールする必要はありません。

Quorum/Witness 機能をセットアップする一般的なプロセスには以下の手順が含まれます。

1. サーバーをセットアップし、他のサーバーとネットワーク通信ができることを確認します。
2. サーバーに LifeKeeper をインストールします。その際、setupコマンドで「Use Quorum / Witness functions」を有効にして、Quorum/Witness パッケージもインストールしてください。
3. Witness 専用ノードを含め、すべてのノード間でコミュニケーションパスを作成します。
4. Quorum/Witnessの設定 を行います。

上記の手順が完了すると、そのクラスターでは Quorum/Witness 機能が動作するようになり、フェイルオーバーが許可される前に、Quorum チェックおよび Witness チェックが行われます。
詳細な設定オプションについては、下の「設定可能なコンポーネント」セクションを参照してください。

**注記:** Quorum/Witness パッケージをインストールしたノードであれば、Quorum/Witness 機能のクラスターに参加できます。Witness 専用ノードとは、保護対象のリソースを持たず、他のすべてのノードとコミュニケーションパスが設定されているノードのことを指しています。

# 設定可能なコンポーネント

Quorum/Witness パッケージでは、Quorum モードと Witness モードの 2 つのモードを設定できます。デフォルトでは、Quorum/Witness パッケージをインストールすると、Quorum モードと Witness モードの両方のモードが有効になります。

これらのモードの動作は、/etc/default/LifeKeeper 設定ファイルでカスタマイズすることが可能です。Quorum モードと Witness モードを個別に調整することもできます。パッケージがインストールされると設定ファイルにはデフォルト設定が書き込まれ、 *majority* がデフォルトの Quorum モードに、*remote_verify* がデフォルトの Witness モードになります。以下はその例です。

```
QUORUM_MODE=majority
WITNESS_MODE=remote_verify
```

# 使用可能な Quorum モード

Quorum チェックのモードとして、次の 4 種類のモードが用意されています。これらは、/etc/default/LifeKeeper 設定ファイルの QUORUM_MODE で設定できます。

| QUORUM_MODE | 説明 |
|---|---|
| *majority*<br>（デフォルト） | コミュニケーションパスを通じて疎通確認を行い、Quorum チェックを行います。クラスター内の過半数のノードと疎通ができることで、Quorum チェック成功と判断します。 本モードは 3 ノード以上のクラスターで動作可能です。 2 ノード構成の場合は、 |

| | |
|---|---|
| | Witness 専用のノードを追加する必要があります。 詳細は「majority モード 」を参照してください。 |
| *tcp_remote* | コミュニケーションパスから独立したホストに対して、指定されたポート上の TCP/IP サービスに接続確認を行い、Quorum チェックを行います。指定されたホストの過半数に接続できることで、Quorum チェック成功と判断します。 別途、接続確認用のホストが必要です。 詳細は「tcp_remote モード 」を参照してください。 |
| *storage* | 共有ストレージを Witness デバイスとして用いる Quorum モードです。クラスター内のすべてのノードからアクセスできる共有ストレージを用いた合意システムで、共有ストレージを介してノードの情報交換を行います。共有ストレージにアクセスできることで、Quorum チェック成功と判断します。 本モードは 2 ノードのクラスターから動作可能で、最大 4 ノードまでサポートされます。別途、共有ストレージが必要です。<br>詳細は「storage モード 」を参照してください。<br>なお、Quorum モードに *storage* を選んだ場合、後述の Witness モードも *storage* を選択しなければなりません。 |
| *none* または *off* | Quorum チェックが無効になっています。この設定では、常に Quorum チェック成功と判断されます。 |

## 使用可能な Witness モード

Witness チェックのモードとして、次の 3 種類のモードが用意されています。これらは、`/etc/default/LifeKeeper` 設定ファイルの `WITNESS_MODE` で設定できます。

| WITNESS_MODE | 説明 |
|---|---|
| *remote_verify*（デフォルト） | クラスター内の他のすべてのノードに対して障害が疑われるノードのステータスに関する意見を求めます。 1 つのノードでも障害なしと判断した場合は、Witness チェックの結果は障害なしと判断します。すべてのノードが障害と判断した場合は、Witness チェックの結果は障害が発生していると判断します。 |
| *storage* | 共有ストレージを Witness デバイスとして用いる Witness モードです。共有ストレージに書き込まれた情報を定期的に確認し、更新されていれば障害なし、更新が止まれば障害が発生していると判断します。 詳細は「storage モード 」を参照してください。<br>なお、Witness モードに *storage* を選んだ場合、先述の Quorum モードも *storage* を選択しなければなりません。 |
| *none* または *off* | Witness チェックが無効になっています。この設定では、常に障害が発生していると判断されます。 |

**注記:** リソースを持たない Witness 専用ノードは Witness チェックを実行する必要がありません。そのため、そのようなノードでは Witness モードを *none* または *off* に設定する必要があります。

# 使用可能な Quorum モードと Witness モードの組み合わせ

以下の表の組み合わせが可能です。

| | | QUORUM_MODE | | | |
|---|---|---|---|---|---|
| | | ***majority*** | ***tcp_remote*** | ***storage*** | ***none または off*** |
| **WITNESS_MODE** | ***remote_verify*** | 利用可能<br>３ノード以上 | 利用可能<br>３ノード以上 | 利用不可 | 利用可能<br>３ノード以上 |
| | ***storage*** | 利用不可 | 利用不可 | 利用可能<br>２ノード以上<br>４ノード以下 | 利用不可 |
| | ***none または off*** | 利用可能<br>３ノード以上 | 利用可能<br>２ノード以上 | 利用不可 | 利用可能 |

# Quorum を喪失したときに利用可能なアクション

Quorum/Witness パッケージでは、Quorum を喪失したときにシステムがどのように応答すべきかについて３種類のオプションを提供しています。“*fastboot*”、“*fastkill*” および “*osu*” です。これらのオプションは、`/etc/default/LifeKeeper` 設定ファイルの `QUORUM_LOSS_ACTION` 設定によって選択できます。３つのオプションはすべてそのシステムのリソースをサービス休止状態にしますが、それぞれ異なる動作をします。

| オプション | 説明 |
|---|---|
| *fastboot* | （コミュニケーションパスの障害により）Quorum の喪失が検出されると、システムは **直ちに** リブートします。これは過激な方法ですが、確実にシステムを外部のリソースから素早く切り離すことができます。ストレージレベルのレプリケーションなど、多くの場合にリソースをこのように即座にリリースすることが望まれます。 このオプションには、以下の 2 つの重要な注意点があります。<br>1. システムは、最初にシャットダウン手順を実行することなく **直ちに** ハードリブートを実行します。（ディスクの同期などの）シャットダウンタスクは一切実行されません。<br>2. システムは、ストレージとのネゴシエーションやリソースへのアクセスなどを含む通常の起動ルーチンを実行しながら復帰します。 |
| *fastkill*（デフォル | *fastboot* オプションに非常に似ていますが、システムは Quorum を喪失したときにハードリブートするのではなく、 **直ちに** 停止します。 *fastboot* オプションと同様に（ディスクの同期などの）シャットダウンタスクは一切実行されません。システムは手動で起動する必要があります。 |

| | |
|---|---|
| ト) | その後システムは、ストレージとのネゴシエーションやリソースへのアクセスなどを含む通常の起動ルーチンを実行しながら復帰します。 |
| *osu* | 最も穏健なオプションです。Quorum を喪失したシステムはそのまま稼働しますが、システム上のリソースはサービス休止状態にされます。一部のクラスター構成ではこの方法で十分ですが、他のクラスター構成では保護能力不足だったり応答が遅すぎる場合があります。 |

# 5.4.2.10.3.1. majority モード

コミュニケーションパスを通じて疎通確認を行い、Quorum チェックを行います。クラスター内の過半数のノードと疎通ができることで、Quorum チェック成功と判断します。
本モードは 3 ノード以上のクラスターで動作可能です。 2 ノード構成の場合は、Witness 専用のノードを追加する必要があります。

> **注記:** 多数決ベースのため、クラスターを構成するノード数は奇数にすることを推奨します。

**quorum を使用してクラスターを停止する:**

1. ターゲットを停止
2. ソースを停止
3. witness を停止

**quorum を使用してクラスターを起動する:**

1. witness を起動
2. ターゲットを起動
3. ソースを起動

## majority モードの設定

`/etc/default/LifeKeeper` 設定ファイルで `QUORUM_MODE` の設定を *majority* に設定してください。
本モードでは、他の設定はありません。

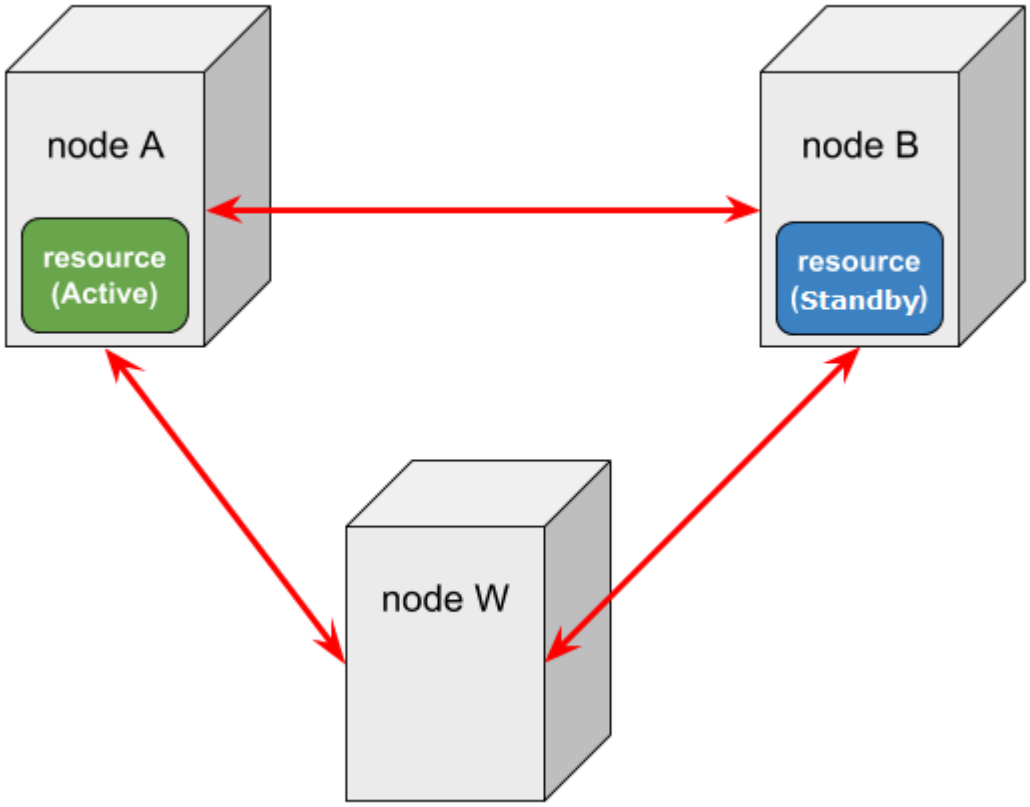
## majority モードで使用可能な Witness モード

本モードでは以下の Witness モードを使用できます。各モードの詳細は「使用可能な Witness モード」を参照してください。

- *remote_verify*
- *none/off*

# majority モードの期待される動作（標準の設定を仮定）

ノード A（リソースはサービス状態）、ノード B（リソースは待機状態）、そしてノード W（保護対象のリソースを持っていない Witness 専用ノード）の 3 ノード構成のクラスターの振る舞いについて示します。




なお、ノード障害に関するリソースの状態を変更し得るイベントは以下の 3 つです。

- COMM_DOWN イベント
  ノード間のコミュニケーションパスが全て切断した時に呼び出されるイベント。

- COMM_UP イベント
  COMM_DOWN 状態からコミュニケーションパスが復旧した際に呼び出されるイベント。

- LCM_AVAIL イベント
  LCM の初期化が終わった時に呼び出されるイベントで、LifeKeeper 起動時に 1 度だけ呼び出される。起動時に他のノードとコミュニケーションパス通信ができる場合は、必ず COMM_UP イベントの処理より前に LCM_AVAIL イベントの処理が行われる。

## シナリオ 1

**ノード A とノード B の間のコミュニケーションパス通信に障害が発生**

この場合、以下のように動作します。

1. ノード A とノード B は、COMM_DOWN イベントの処理を開始します。ただし、全く同時とは限りません。

2. 両方のノードは Quorum チェックを実行し、両方共自身が Quorum を持っていると判断します（ノード A とノード B の両方からノード W が見えているため、既知の 3 ノードのうちの 2 ノードと疎通があるため多数派にいると判断します）。

3. 各ノードは、まだ通信可能な別のノードに対し、自ノードと通信できなくなったノードの状態について問い合わせ（Witness チェック）を行います。このシナリオでは、ノード A がノード B のステータスについてノード W に問い合わせ、ノード B がノード A のステータスについてノード W に問い合わせることになります。

4. ノード A とノード B は共に ノード W への問い合わせによって他方のノードがまだ生存していると判断し、フェイルオーバー処理は発生しません。リソースはノード A でサービス状態のままになります。

## シナリオ 2

**ノード A とノード W の間のコミュニケーションパス通信に障害が発生**

Quorum/Witness パッケージがインストールされていると、すべてのノードが Witness ノードとして動作するため、このシナリオは前のシナリオと同じになります。この場合、ノード A とノード W は共にノード B へ問い合わせを行い、それにより他方のノードがまだ生存していると判断します。

## シナリオ 3

**ノード A に障害が発生してノード A が停止**

この場合、ノード B は以下の動作をします。

1. ノード A との COMM_DOWN イベントの処理を開始します。

2. ノード W と通信が可能であるので、Quorum を持っていると判断します。

3. ノード A が見えないことをノード W に確認した後、通常のフェイルオーバー処理を開始します。

4. これにより保護対象のリソースはノード B でサービス中になります。

**リソースがノード B でサービス中に、他のノードと通信可能な状態でノード A のサーバーが電源 ON**

この場合、ノード A は LCM_AVAIL イベントを処理します。ノード A は、Quorum を持っていると判断し、ノード B でリソースがサービス中のためリソースを起動させません。

また、その直後に、ノード A とノード B、ノード A とノード W との COMM_UP イベントが各ノードで発生しします（ノード A では２回発生する）。各ノードの COMM_UP イベントの処理では、Quorum を持っていると判断し、またノード B でリソースがサービス中のためリソースは起動させません。

**リソースがノード B でサービス中に、他のノードと通信不能な状態でノード A のサーバーが電源ON**

この場合、ノード A は LCM_AVAIL イベントを処理し、ノード B とノード W はノード A と通信できないので何もしません。ノード A は、３ノードのうちの１ノード（自分自身）としか通信できないため Quorum を持っていないと判断します。Quorum を持っていないため、ノード A はリソースを起動させません。

## シナリオ４

**ノード A のネットワークに障害が発生（ノード A は稼働しているが、他のノードと通信不能）**

この場合、ノード A は以下の動作をします。

1. ノード B との COMM_DOWN イベントの処理を開始します（ほぼ同時に、ノード W との COMM_DOWN イベントの処理も開始されます）。
2. ノード B ともノード W とも通信ができないため、Quorum を持っていないと判断します。
3. LifeKeeper に設定されている QUORUM_LOSS_ACTION パラメータの設定に基づいてノードの強制再起動や強制停止などを行います。QUORUM_LOSS_ACTION のデフォルトは「fastkill」に設定されています。詳細は Quorum/Witness を参照してください。

また、ノード B は以下の動作をします。

1. ノード A との COMM_DOWN イベントの処理を開始します。
2. ノード W とまだ通信が可能であるため、Quorum を持っていると判断します。
3. ノード A が見えないことをノード W に確認（Witness チェック）した後、通常のフェイルオーバー動作を開始します。
4. これにより保護対象のリソースはノード B でサービス中になります。

**リソースがノード B でサービス中に、ノード A の通信が回復**

この場合、ノード B は、 COMM_UP イベントを処理し、Quorum を持っている（３ノードすべてが見える）ことと、サービス中のリソースを持っていることを判断します。ノード A は、COMM_UP イベントを処理し、Quorum を持っていることと、リソースが別のノードでサービス中であることを判断します。この時点では、ノード A はリソースを起動させません。

## シナリオ５

**3つのノードすべてが他のノードと通信不能**

この場合、ノード A は以下の動作をします。

1. ノード B との COMM_DOWN イベントの処理を開始します（ほぼ同時に、ノード W との COMM_DOWN イベントの処理も開始されます）。

2. ノード B ともノード W とも通信ができないため、Quorum を持っていないと判断します。

3. LifeKeeper に設定されている QUORUM_LOSS_ACTION パラメータの設定に基づいてノードの強制再起動や強制停止などを行います。QUORUM_LOSS_ACTION のデフォルトは「fastkill」に設定されています。詳細は Quorum/Witness を参照してください。

また、ノード B は以下の動作をします。

1. ノード A との COMM_DOWN イベントの処理を開始します。

2. ノード A ともノード W とも通信ができないため、Quorum を持っていないと判断します。

3. サービス状態のリソースがないので、強制停止はしません。

コミュニケーションパスが復旧した場合、ノード A 上でリソースが起動します。なお、本動作となるためには、以下の 2条件を満たす必要があります。

- ノード A のリソースに設定された初期化設定値が AUTORES_ISP であること。
- リソースにセットされた優先度がノード A で最も高い値であること。

# 5.4.2.10.3.2. tcp_remote モード

コミュニケーションパスから独立したホストに対して、指定されたポート上の TCP/IP サービスに接続確認を行い、Quorum チェックを行います。指定されたホストの過半数に接続できることで、Quorum チェック成功と判断します。
本モードは２ノードのクラスターでも動作可能です。ただし、Witness モードを *remote_verify* で使用する場合は３ノード必要になります。また、別途、接続確認用のホストが必要です。

**注記:** 多数決ベースのため、指定するホスト数は奇数にすることを推奨します。
**注記:** このモードは本質的に柔軟性と複雑さを備えるため、使用するには、LifeKeeper および関連する特定のネットワーク／クラスター設定の両方に対する十分な理解と注意が必要です。

## tcp_remote モードの設定

`/etc/default/LifeKeeper` 設定ファイルで QUORUM_MODE の設定を *tcp_remote* に設定してください。また、他に以下の設定があります。詳細は「Quorumパラメーター一覧」を参照してください。

- QUORUM_HOSTS
- QUORUM_TIMEOUT_SECS

## tcp_remote モードで使用可能な Witness モード

本モードでは以下の Witness モードを使用できます。各モードの詳細は「使用可能な Witness モード」を参照してください。

- *remote_verify*
- *none* または *off*

# 5.4.2.10.3.3. storage モード

> *リソースを作成および拡張する前に、Quorum が正しく設定されていることを確認してください。Quorum の構成が誤っていると、サーバーが誤って停止する可能性があります。*

クラスター内のすべてのノードからアクセスできる共有ストレージを用いた合意システムで、共有ストレージを介してノードの情報交換を行います。各ノードは自ノード情報を定期的に共有ストレージに書き込み、また定期的に他ノードが書き込んだノード情報を読み込みます。この共有ストレージに書き込まれたノード情報を QWK オブジェクトと呼びます。QWK オブジェクトはクラスターを構成するノード数だけ必要です。
Quorum チェックでは、共有ストレージにアクセスできることで Quorum チェック成功と判断します。また、Witness チェックでは、障害が疑われるノードの QWK オブジェクトを確認し、一定期間内に更新されていると障害なし、更新が止まっていれば障害が発生していると判断します。なお、実装上は、Quorum チェック時に Witness チェックも実施しています。
なお、Quorum モードに storage を選んだ場合、後述の Witness モードも storage を選択しなければなりません。
本モードは 2 ノードのクラスターから動作可能で、最大 4 ノードまでサポートされます。別途、クラスターを構成するノード数分の QWK オブジェクトを格納するための共有ストレージが必要です。なお共有ストレージにアクセスできなくなると、リソースの起動に影響します。すべてのノードから常時アクセス可能な共有ストレージを選定してください。

**注意:** 共有ストレージは各ノードからアクセス可能なストレージのことです。この共有ストレージは Quorum/Witness 機能専用であり、LifeKeeper がリソースとして保護することはできません。

**注意:** 本モードを利用するには、設定後に初期化が必要です（後述の「storage モードの使用手順」参照）。また、初期化後にクラスター内のノードの追加･削除や設定の変更を行う場合は、再初期化が必要です。

**注意:** クラスター内に”-”と”.”だけ異なるホスト名のノードが存在する場合（例：lifekeeper-sios と lifekeeper.sios）、本モードを利用することができません。本モードを利用する場合はホスト名を変更してください。

## 利用可能な共有ストレージ

Quorum/Witness 機能の目的であるスプリットブレイン回避を確実なものにするために、Quorum 分割（例：ノード A の QWK オブジェクトにアクセス可能であるが、ノード B の QWK オブジェクトにアクセス不能）が発生しない構成としなければなりません。そのため、クラスター内のすべての QWK オブジェクトが同一共有ストレージ上の同一種のブロックデバイス･レギュラーファイル･EFS または S3 オブジェクトに配置されていることを条件とします。

利用可能な共有ストレージは以下の通りです。/etc/default/LifeKeeper 設定ファイルの QWK_STORAGE_TYPE で指定します。

| QWK_STORAGE_TYPE | QWK オブジェクトの配置場所 |
|---|---|

<table>
<tr><td rowspan="2">block</td><td>共有ストレージに物理ストレージやRDM（物理互換）、iSCSI（VM 内イニシエーター）を使う場合、１つのQWK オブジェクトは以下のいずれかに配置してください。<br><br>(a) 1 QWK オブジェクト = 1 パーティション<br><br>(b) 1 QWK オブジェクト = 1 LU<br><br>(a) の場合、複数のホストが１つの LU に書き込みを行いますので、パーティションの LU 内のオフセットは 4K（ストレージ装置のセクターサイズ）でアラインしてください。また、他用途のパーティションを混在させないでください。<br>(b) の場合、LU 内部にパーティションは作成しないでください。<br>また、(a) と (b) いずれの場合も、ファイルシステムを作成する必要はありません。</td></tr>
<tr><td>共有ストレージにVMDKを使う場合、１つのQWK オブジェクトは以下に配置してください。<br><br>1 QWK オブジェクト = 1 VMDK<br><br>パーティションは作成しないでください。また、ファイルシステムを作成する必要はありません。<br>VMDKのプロビジョニングオプションは以下の通り設定してください。<br><br>thick (eager zeroed)</td></tr>
<tr><td>file (または EFS)</td><td>共有ストレージにNFS (または EFS) を使う場合、１つのQWK オブジェクトは以下に配置してください。<br><br>1 QWK オブジェクト = NFS (または EFS) ファイルシステム内の 1 レギュラーファイル<br><br>EFSを使用した場合も、/etc/fstabにファイルシステム情報の記述が必要です。一方でパラメーターはNFSとEFSで体系が異なり、以下に記載のオプションはNFSにのみ適用されます。<br><br>NFSサーバーの export オプションは以下の通り設定してください。<br><br>rw,no_root_squash,sync,no_wdelay<br><br>NFSの mount オプションは以下の通り設定してください。</td></tr>
</table>

| | |
|---|---|
| | soft,timeo=20,retrans=1,noac<br><br>OS再起動後、自動でマウントするように /etc/fstab 等の設定を行ってください。<br><br>EFS の設定については、こちらの AWS の記事 を参照してください。 |
| aws_s3 | 共有ストレージに Amazon Simple Storage Service (S3) を使う場合、１つの QWK オブジェクトは以下に配置してください。<br><br>**1 QWK オブジェクト = 1 S3 オブジェクト**<br><br>LifeKeeper が動作しているインスタンスのリージョンと別のリージョンの S3 を利用してください。これは、同一リージョンの S3 を利用した場合に AZ(Availability Zone) 間の接続が分断されると、複数 AZ にレプリケーションが配置されている S3 も同時に分断し、S3 への一貫性のあるアクセスが期待できないためです。<br>クラスターを構成するすべてのノードで以下の要件を満たしている必要があります。<br><br>• AWS Command Line Interface (AWS CLI) がインストールされていて、rootユーザで使用可能であること。インストール方法は、「AWS Command Line Interface のインストール 」を参照してください。<br>• Amazon S3 におけるエンドポイント（AWS のリージョンとエンドポイント ）にプロトコル HTTP および HTTPS を使用してアクセスできること。ただし、インターネットに接続できない環境で、別のリージョンの S3 にアクセスするにはエンドポイントの指定が必要ですが、現在 LifeKeeper はエンドポイントの指定に対応していません。<br>• EC2 の IAM ロール や AWS CLI の設定 を適切に行い、root ユーザから S3 オブジェクトにアクセスできること。<br><br>注記: 設定ファイル /etc/default/LifeKeeper のパラメーター PATH に AWS CLI 実行ファイルのパスが設定されていない場合、PATH に AWS CLI 実行ファイルのパスを追加してください。 |

1 QWK オブジェクトのサイズは4096バイトです。

自ノード QWK オブジェクトに対しては read / write を行います。他ノードの QWK オブジェクトに対しては read のみ行います。適切にアクセス権限を設定してください（Persistent Reservation などのアクセス権制限を伴う場合は要注意）。

**注記:** vSphere の4K ネイティブディスクは QWK オブジェクトの共有ストレージとして利用することはで

きません。

## storage モードの設定

/etc/default/LifeKeeper 設定ファイルで QUORUM_MODE と WITNESS_MODE の設定を storage に設定してください。また、他に以下の設定があります。詳細は「Quorumパラメーター一覧」を参照してください。

- QWK_STORAGE_TYPE
- QWK_STORAGE_HBEATTIME
- QWK_STORAGE_NUMHBEATS
- QWK_STORAGE_OBJECT_<ホスト名>
- HTTP_PROXY, HTTPS_PROXY, NO_PROXY

## storage モードの使用手順

本モードで使用する場合は、初期化が必要です。以下の手順に従って初期化してください。

1. クラスターを構成するすべてのサーバーをセットアップして、他のサーバーとネットワーク通信ができることを確認します。
2. クラスターを構成するすべてのサーバーに LifeKeeper をインストールします。その際、setup コマンドで「Use Quorum / Witness functions」を有効にして、Quorum/Witness パッケージもインストールしてください。
3. すべてのノード間にコミュニケーションパスを作成します。
4. すべてのノードで /etc/default/LifeKeeper 設定ファイルを編集し、storage モードの設定を行います。

> ✻ Storage Quorum を使用するには、LCMNUMHBEATS を増やして、パスが失敗としてマークされるまでの時間を長くすることを **推奨します** 。LCMNUMHBEATS を 9 にすることで、タイムアウト時間がデフォルトの15秒から45秒に変更されます。なお、LCMNUMHBEATS を変更した場合は、LifeKeeper を再起動してください。

> ! qwk_storage_init を実行する **前** に、すべてのコミュニケーションパスが稼働していてAliveであることを確認してください。

5. すべてのノードで qwk_storage_init コマンドを実行します。このコマンドは、すべてのノードで QWK オブジェクトの初期化が終わるまで待ち状態になります。このコマンドがすべてのノード

で終了すると、storage モードとして Quorum/Witness 機能が利用できる状態となります。

初期化後にクラスターを構成するノードの追加･削除をする、または /etc/default/LifeKeeper 設定ファイルのパラメーターを変更する場合、再初期化が必要です。以下の手順に従って再初期化してください。

1. すべてのノードで qwk_storage_exit コマンドを実行します。
2. ノードを削除する場合、削除するノードと他すべてのノード間のコミュニケーションパスを削除します。
   ノードを追加する場合、追加するノードと他すべてのノード間にコミュニケーションパスを作成します。
3. すべてのノードで /etc/default/LifeKeeper 設定ファイルを再設定します。
4. すべてのノードで qwk_storage_init コマンドを実行します。

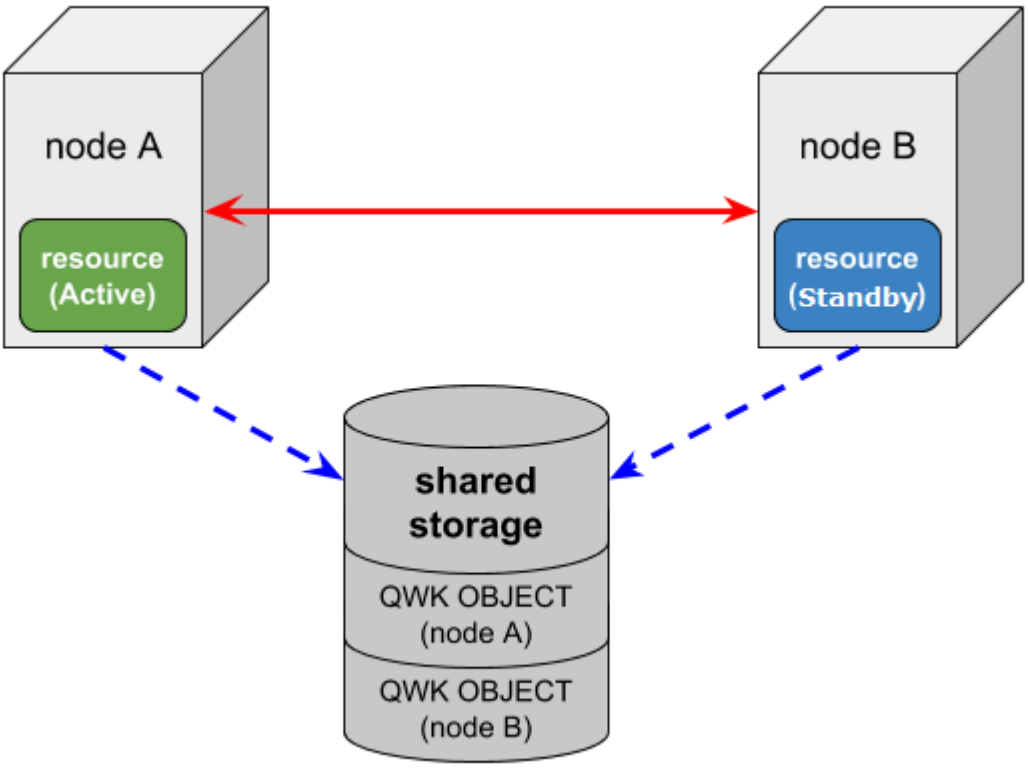
## トラブルシューティング

lifekeeper.log に 135802 メッセージが頻繁に出力されている場合、定期的に行っている QWK オブジェクトの読み書きの負荷が高すぎて、処理が間に合っていません。クラスター内のノード数が多い場合、共有ストレージにS3 を使用している場合（特に 2 リージョン）、QWK オブジェクトの読み書きの負荷が高くなります。
以下の対策を実施して、135802 メッセージが出力されないようにしてください。

- QWK_STORAGE_HBEATTIME の値を増やす（フェイルオーバーに要する時間が長くなります）。
- 共有ストレージが S3 の場合は、1 リージョンにして、ネットワーク的に近いリージョンの S3 を利用する。
- 処理能力を増やす（Amazon EC2 インスタンスの場合、インスタンスタイプを変更するなど）。

# storage モードの期待される動作（標準の設定を仮定）

ノード A（リソースはサービス状態）、ノード B（リソースは待機状態）の 2 ノード構成のクラスターの振る舞いについて示します。




なお、ノード障害に関するリソースの状態を変更し得るイベントは以下の３つです。

- COMM_DOWN イベント
  ノード間のコミュニケーションパスが全て切断した時に呼び出されるイベント。

- COMM_UP イベント
  COMM_DOWN 状態からコミュニケーションパスが復旧した際に呼び出されるイベント。

- LCM_AVAIL イベント
  LCM の初期化が終わった時に呼び出されるイベントで、LifeKeeper 起動時に１度だけ呼び出される。起動時に他のノードとコミュニケーションパス通信ができる場合は、必ず COMM_UP イベントの処理より前に LCM_AVAIL イベントの処理が行われる。

## シナリオ１

**ノード A とノード B の間のコミュニケーションパス通信に障害が発生（ノード A とノード B とも共有ストレージにアクセス可能）**

この場合、以下のように動作します。

1. ノード A とノード B は、COMM_DOWN イベントの処理を開始します。ただし、全く同時とは限りません。

2. 両方のノードは Quorum チェックを実行し、両方共自身が Quorum を持っていると判断します（共有ストレージアクセスが可能であるため）。

3. 各ノードは、通信できなくなったノードの QWK オブジェクトの更新を確認します（Witness チェック）。ノード A とノード B は共に稼働しているため定期的に QWK オブジェクトを更新され、Witness チェックでその更新を確認します。

4. Witness チェックの結果、他方のノードがまだ生存しているためフェイルオーバー処理は発生しません。リソースはノード A でサービス状態のままになります。

## シナリオ 2

**ノード A に障害が発生してノード A が停止**

この場合、ノード B は以下の動作をします。

1. ノード A との COMM_DOWN イベントの処理を開始します。

2. 共有ストレージにアクセス可能であるので、Quorum を持っていると判断します。

3. ノード A の QWK オブジェクトの更新が停止していることを確認します（Witness チェック）。

4. Witness チェックの結果、ノード A で障害が発生していると判断したため通常のフェイルオーバー処理を開始します。これにより保護対象のリソースはノード B でサービス中になります。

**リソースがノード B でサービス中に、通信可能かつ共有ストレージにアクセス可能な状態でノード A のサーバーが電源 ON**

この場合、ノード A は LCM_AVAIL イベントを処理します。ノード A は、Quorum を持っていると判断し、ノード B でリソースがサービス中のためリソースを起動させません。
また、その直後に、COMM_UP イベントが各ノードで発生しします。各ノードの COMM_UP イベントの処理では、Quorum を持っていると判断し、またノード B でリソースがサービス中のためリソースは起動させません。

**リソースがノード B でサービス中に、通信不能かつ共有ストレージにアクセス可能な状態でノード A のサーバーが電源 ON**

この場合、ノード A は LCM_AVAIL イベントを処理します。ノード A は、Quorum を持っていると判断し、通信できないノード B に障害が発生しているか確認するためノード B の QWK オブジェクトの更新を確認します。ノード B は稼働しているため、QWK オブジェクトは定期的に更新されます。ノード A では QWK オブジェクトの更新を確認し、ノード B が生存しているが通信できないためにリソースは起動させません。
ノード B はノード A と通信できないので何もしません。

## シナリオ3

**ノード A のネットワークに障害が発生（ノード A は稼働しているが、他のノードや共有ストレージにアクセス不能）**

この場合、ノード A は以下の動作をします。

1. ノード B との COMM_DOWN イベントの処理を開始します。
2. 共有ストレージにアクセス不能であるため、Quorum を持っていないと判断します。
3. 直ちに 強制停止します（QUORUM_LOSS_ACTION のデフォルト「fastkill」の振る舞い）。

また、ノード B は以下の動作をします。

1. ノード A との COMM_DOWN イベントの処理を開始します。
2. 共有ストレージにアクセス可能であるため、Quorum を持っていると判断します。
3. ノード A の QWK オブジェクトの更新が停止していることを確認します（Witness チェック）。
4. Witness チェックの結果、ノード A で障害が発生していると判断したため通常のフェイルオーバー処理を開始します。これにより保護対象のリソースはノード B でサービス中になります。

**リソースがノード B でサービス中に、共有ストレージにアクセス可能な状態でノード A のサーバーが電源 ON**

シナリオ 2 と同じです。

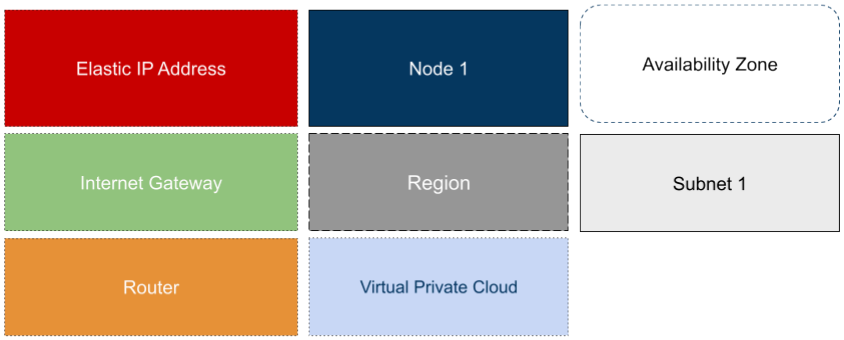
**リソースがノード B でサービス中に、共有ストレージにアクセス不能な状態でノード A のサーバーが電源 ON**

この場合、ノード A は LCM_AVAIL イベントを処理します。ノード A は、Quorum を持っていないと判断し、リソースを起動させません。
ノード B とのコミュニケーションパス通信が可能であれば、その後に COMM_UP イベントが発生します。しかし、Quorum を持っていないので、ノード A ではリソースを起動させません。

# 5.4.2.10.3.4. AWS 環境の Quorum/Witness クラスターの推奨事項

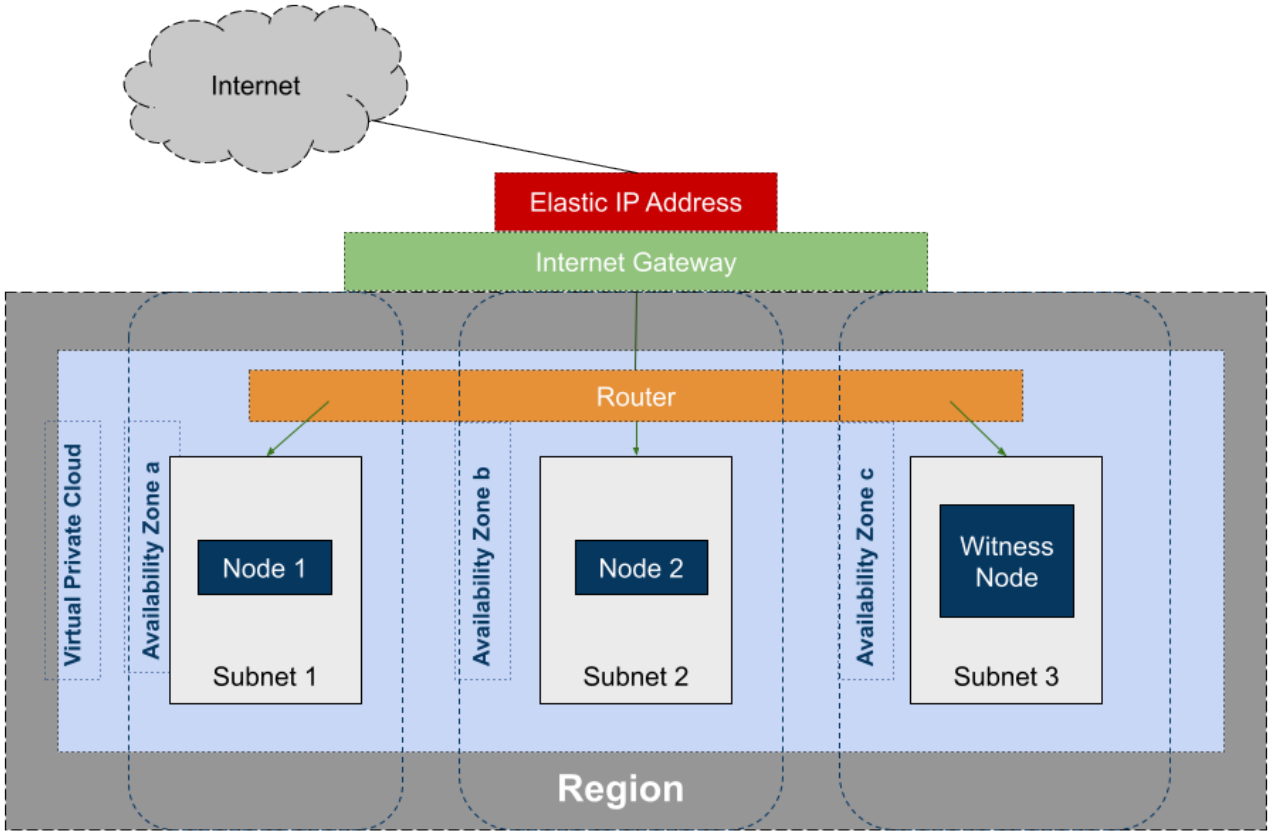
このページでは、AWS のクラウド環境において既存のノードと併せて **Quorum/Witness** を設定するのに役立つクラスター構成を紹介します。

## 概略図:



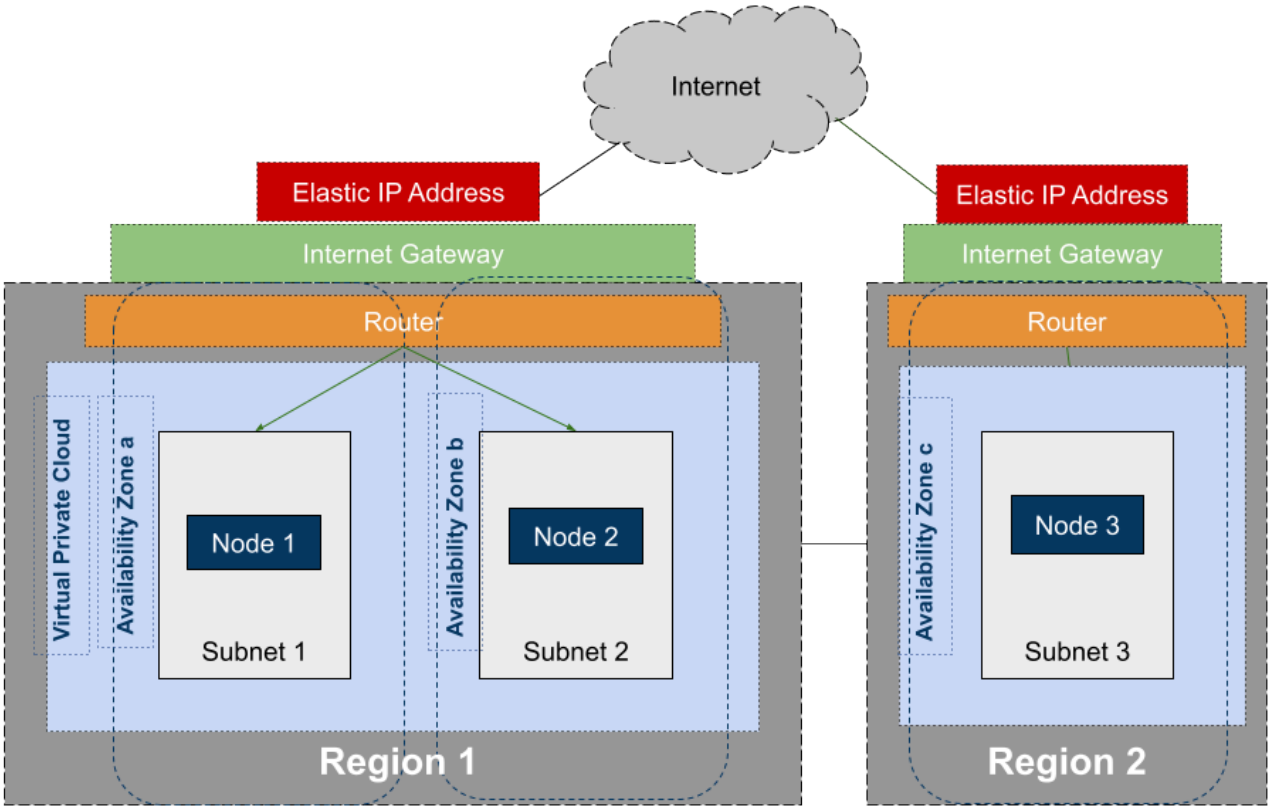

## 2ノード、単一リージョンのデプロイメント

両方のノードがすべて同じリージョンにある場合、Witness ノードも同じリージョンに配置する必要があります。リージョン内の Availability Zone に障害が発生した場合でも、Witness ノードが残ります。リージョン全体で障害が発生した場合、ノードは存在しなくなるので、フェイルオーバーは無意味になります。



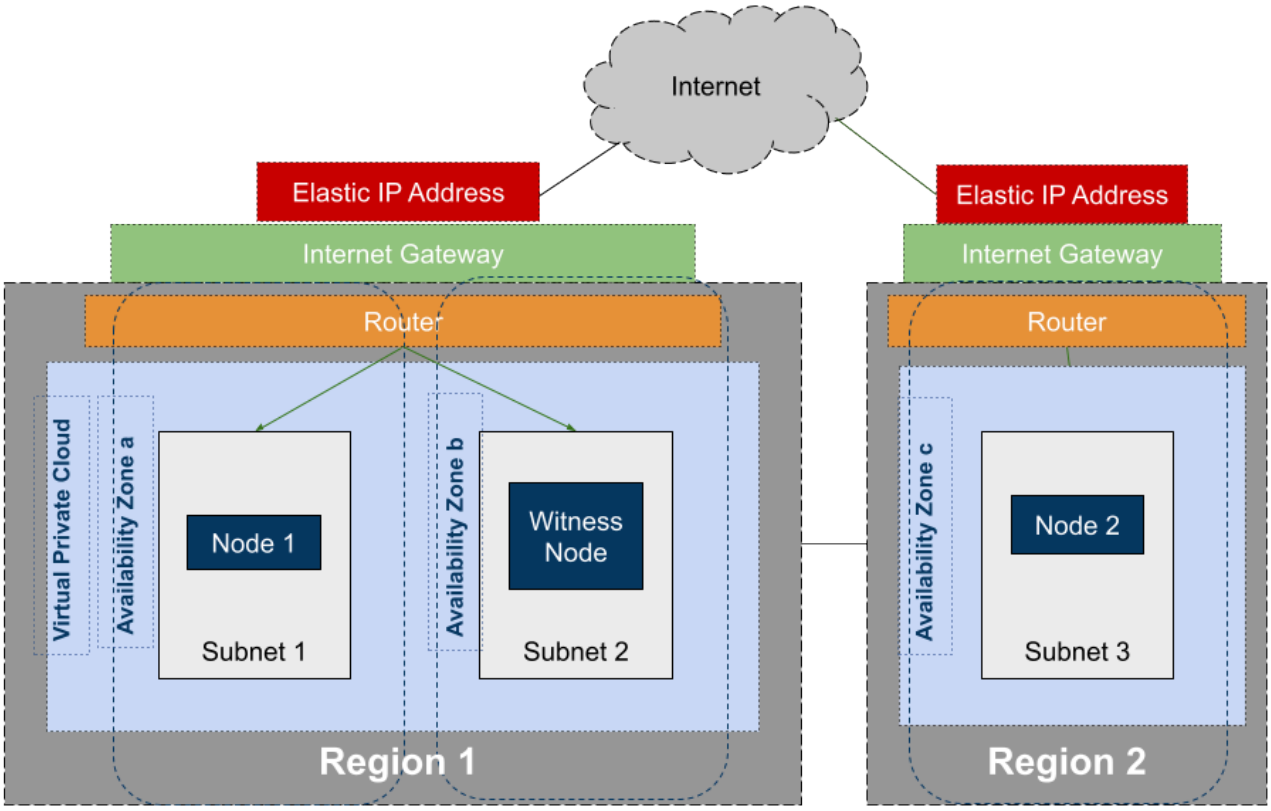

## 3ノード、2リージョンのデプロイメント

2つのノードが1つのリージョンに存在し、3番目のノードが別のリージョンに存在する3ノードのクラスターでは、SIOS Protection Suite for Linux ではWitness ノードは無意味になります。これは、すでに奇数の票があるためです。リージョン1全体がフェイルオーバーした場合、DR リージョンへのフェイルオーバーは常に手動のプロセスになります。

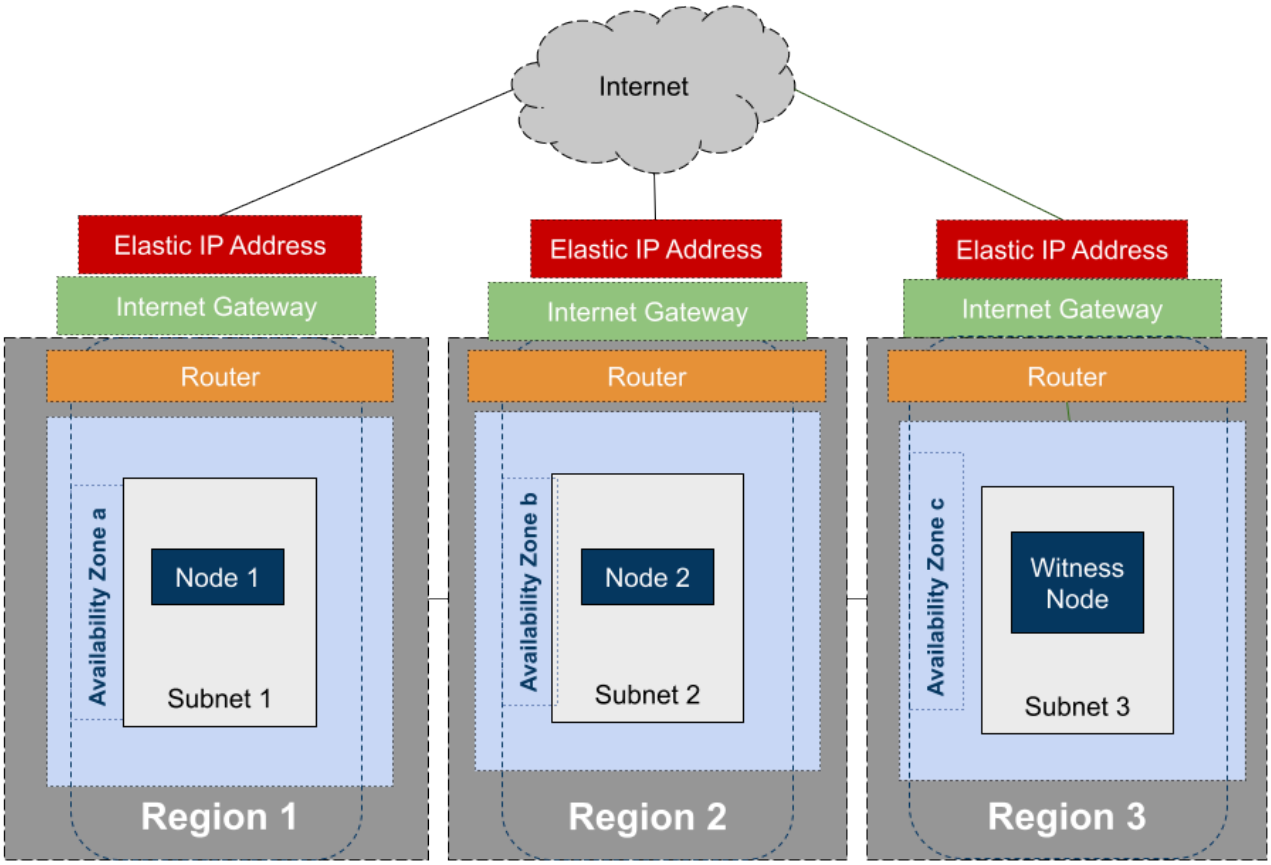



## 2ノード + Witness ノード、マルチリージョンのデプロイメント

1つのノードがリージョン1に、ノード2がリージョン2にある2ノード構成のクラスターでは、リージョン3がある場合は Witness ノードはリージョン3に配置する必要があります。リージョン3がない場合はリージョン1に配置します。リージョン2に配置しない理由は、リージョン1とリージョン2の間のネットワークに障害が発生した場合、リージョン2へのフェイルオーバーが不必要に発生するためです。3番目のリージョンがあると、その可能性はなくなります。3番目のリージョンを持つことが現実的でない場合は、誤ったフェイルオーバーを排除するために、Witness ノードはプライマリーリージョンに配置することをお勧めします。ただし、リージョン1が完全にダウンした場合は、バックアップノードを手動でオンラインにする必要があります。 この事象が発生した場合、障害が発生した日時と、サーバーをオンラインにするために何を行ったかを具体的に記録しておいてください。

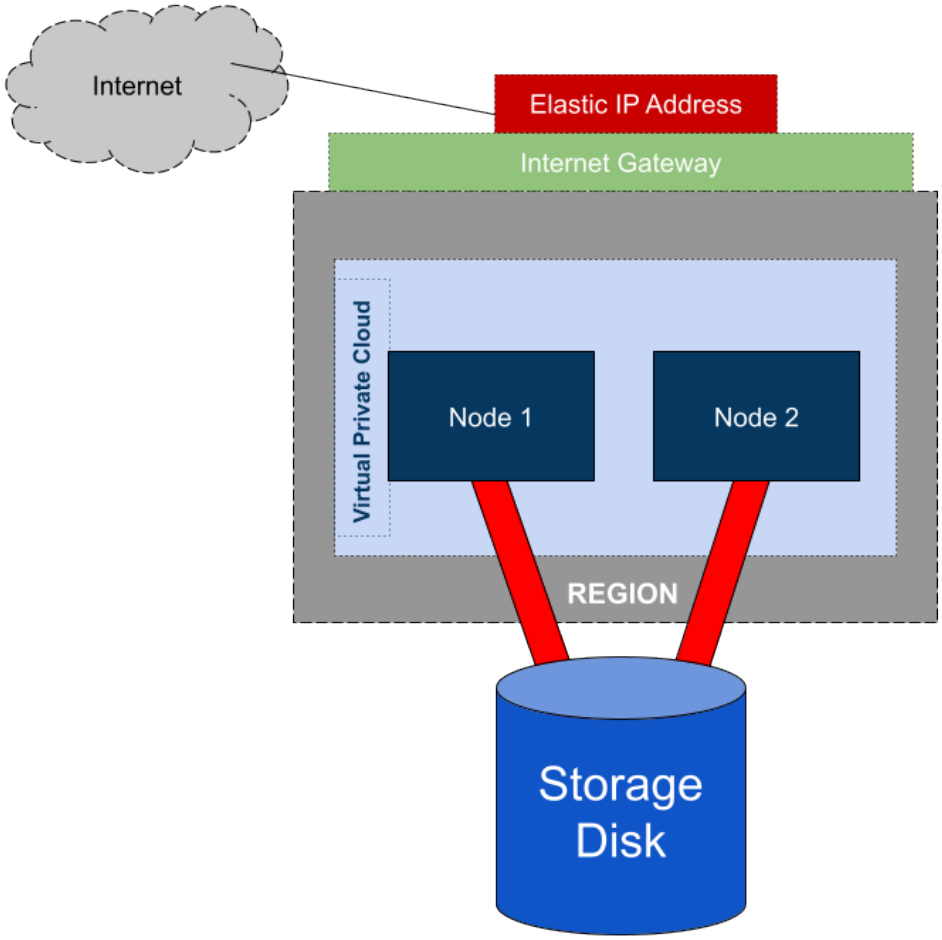

Internet
Elastic IP Address
Elastic IP Address
Internet Gateway
Internet Gateway
Router
Router
Virtual Private Cloud
Availability Zone a
Availability Zone b
Availability Zone c
Node 1
Witness Node
Node 2
Subnet 1
Subnet 2
Subnet 3
Region 1
Region 2

### 3リージョンの場合




## 2ノードとストレージディスク

Amazon S3 にはグローバルという概念があり、S3 バケットはグローバルにアクセス可能です。Amazon S3 を使用する場合は LifeKeeper が動作しているインスタンスのリージョンと別のリージョンの S3 を利用してください。ただし、インターネットに接続できない環境で、別のリージョンの S3 にアクセスするにはエンドポイントの指定が必要ですが、現在 LifeKeeper はエンドポイントの指定に対応していません。NFS または EFS は同じリージョンにセットアップできますが、インスタンスとは別の AZ(Availability Zone) に配置する（EFS の場合は 1 ゾーンストレージクラス）ことで AZ 障害に対応できます。




**注記:** ブロックストレージに加えて **NFS**、**EFS**、**S3** バケット を使用することができます。

# 5.4.2.10.4. STONITH

STONITH (Shoot the Other Node in the Head) は、クラスター内のノードの電源をリモートでオフにするためのフェンシング手法です。LifeKeeper は、外部の電源スイッチ制御、IPMI 対応のマザーボード制御、ハイパーバイザーが提供する電源機能、クラウドベンダーのツールを使用して、クラスター内の他のノードの電源をオフにすることで、STONITH 機能を提供します。各 STONITH メソッドを使用すると、クラスターソフトウェアは、停止したように見えるクラスターノードの電源をオフにして、異常なノードが共有データにアクセスしたり破損したりしないようにします。

> 注意： フェンシングループを回避するために、再起動よりも電源オフを推奨します（2台のマシンが通信不能になっても、お互いに STONITH を実行でき、相互に電源をオフにして再起動します）。

## IPMI を使用した STONITH

IPMI（Intelligent Platform Management Interface）は、システムの状態監視と管理に使用できる、コンピューターシステムへの一連の共通インターフェースを定義します。STONITH と一緒に使用することで、クラスターソフトウェアはシリアル接続またはネットワーク接続を介してスイッチに指示を出して停止したように見えるクラスターノードの電源をオフにし、異常なノードが共有データにアクセスしたり破損したりしないようにします。

### パッケージ要件

- クラスター内の **各サーバー** 上の IPMI ツールパッケージ (例：ipmitool-1.8.11-6.el6.x86_64.rpm)

### ベースボードマネジメントコントローラー（BMC）の設定

クラスター内の各サーバーで BIOS または ipmitool コマンドを使用します（例：ipmitool を使用）。

- **静的 IP を使用：** ipmitool lan set 1 ipsrc static
- **IP アドレスを追加：** ipmitool lan set 1 ipaddr 192.168.0.1
- **サブネットマスクを設定：** ipmitool lan set 1 netmask 255.0.0.0
- **ユーザー名を設定：** ipmitool user set name 1 root
- **パスワードを設定：** ipmitool user set password 1 secret
- **ユーザーに管理者特権レベルを追加：** ipmitool user priv 1 4
- **ユーザーのネットワークアクセスを有効にする：** ipmitool user enable 1
  （詳細については、ipmitoolのマニュアルページを参照してください。）

### STONITH のインストール

次のコマンドを実行して、LifeKeeper がインストールされ、すべてのサーバーへのコミュニケーションパスが構成されている **各サーバー** に LifeKeeper STONITH スクリプトをインストールします。

```
# /opt/LifeKeeper/samples/STONITH/stonith-install
```

### /opt/LifeKeeper/config/stonith.conf の更新

クラスター内の各サーバーの stonith.conf ファイルにエントリーを追加します。

```
# LifeKeeper STONITH configuration
#
# Example: <host> ipmitool -I <interface> -H <ip> -U root -P secret power off
minute-maid ipmitool -I lanplus -H 192.168.0.1 -U root -P secret power off
kool-aid ipmitool -I lanplus -H 192.168.0.2 -U root -P secret power off
```

## VMware vSphere 環境での STONITH

vCLI (vSphere Command-Line Interface) は、ESXi ホストや仮想マシンなどの仮想インフラを管理するために VMware がサポートしているコマンドラインインターフェースです。ニーズに合った vCLI コマンドを選択して、VMware 仮想マシン間での LifeKeeper STONITH の使用に適用できます。

### パッケージの要件

#### STONITH サーバー

VMware vSphere SDK パッケージまたは VMware vSphereCLI（vSphere CLI は、vSphere SDK と同じインストールパッケージに含まれています）。

#### 監視対象の各仮想マシン

VMware Tools

### 設定

vSphere CLI コマンドは、vSphere SDK for Perl 上で実行されます。

- vCLI-esxcli
  - esxcli —-server=10.0.0.1 —-username=root —-password=secret vms vm kill —-type='hard' —-world-id=1234567
- vCLI-vmware_cmd
  - vmware-cmd -H 10.0.0.1 -U root -P secret **<vm_id>** stop hard

### <vm_id> の決定

vSphere CLI コマンドは、vSphere SDK for Perl 上で実行されます。 <vm_id> は VM の識別子として使用されます。この変数は、構成中の VM の構成ファイルを指している必要があります。

1. 利用可能なホストのリストを取得します。

```
vmware-cmd -H <vmware host> -l
```

2. 出力例：

```
/vmfs/volumes/4e08c1b9-d741c09c-1d3e-0019b9cb28be/lampserver/
lampserver.vmx
/vmfs/volumes/4e1e1386-0b862fae-a859-0019b9cb28bc/oracle10/oracle.vmx
/vmfs/volumes/4e08c1b9-d741c09c-1d3e-0019b9cb28be/lampserver02/
lampserver02.vmx
```

3. 太字で出力されている最初の VM を参照するコマンド：

```
vmware-cmd -H 10.0.0.1 -U root -P secret
/vmfs/volumes/4e08c1b9-d741c09c-1d3e-0019b9cb28be/l
ampserver/lampserver.vmx stop hard
```

### STONITH のインストール

次のコマンドを実行して、LifeKeeper がインストールされ、すべてのサーバーへのコミュニケーションパスが構成されている **各サーバー** に LifeKeeper STONITH スクリプトをインストールします。

```
# /opt/LifeKeeper/samples/STONITH/stonith-install
```

### /opt/LifeKeeper/config/stonith.conf の更新

上記の出力にリストされている stonith.conf ファイルの3つのホストのエントリー（他のすべてのエントリーはコメントアウトまたは削除してください）：

```
# LifeKeeper STONITH configuration
#
# Example: vmware-cmd -H 10.0.0.1 -U root -P secret stop hard
lampserver vmware-cmd -H 10.0.0.1 -U root -P secret
/vmfs/volumes/4e08c1b9-d741c09c-1d3e-0019b9cb28be/lampserver/lampserver.vmx
stop hard
oracle vmware-cmd -H 10.0.0.1 -U root -P secret
/vmfs/volumes/4e1e1386-0b862fae-a859-0019b9cb28bc/oracle10/oracle.vmx stop hard
lampserver02 vmware-cmd -H 10.0.0.1 -U root -P secret /
vmfs/volumes/4e08c1b9-d741c09c-1d3e-0019b9cb28be/lampserver02/lampserver02.vmx
stop hard
```

## AWS 環境での STONITH

AWS 上での STONITH については、こちら を参照ください。

## Microsoft Azure 環境でのSTONITH

Azure CLI は、仮想マシンなどの Azure リソースを管理するため Microsoft でサポートされているコマン

ドラインインターフェースです。STONITHと共に使用すると、クラスターソフトウェアは、停止したように見えるクラスターノードの電源をオフにして、異常なノードが共有データにアクセスしたり破損したりしないようにします。

## 要件

パッケージ

Azure CLI – install Linux Azure CLI on **each server** in the cluster.

カスタムロール

Azure Fencing に用いる Microsoft Azure 上の仮想マシン及びユーザーに割り当てられるカスタムロールには、仮想マシンを powerOff できる権限が最低限必要です。

```
Microsoft.Compute/*/read
Microsoft.Compute/virtualMachines/powerOff/action
Microsoft.Compute/virtualMachines/start/action
```

## 事前確認

各サーバー で以下のコマンドを実行し、Microsoft Azure 上の認証が機能していることを確認してください。

```
# az vm show –resource-group <group name> –name <vm name>
```

## STONITH のインストール

次のコマンドを実行して、LifeKeeper がインストールされ、すべてのサーバーへのコミュニケーションパスが構成されている 各サーバー に、LifeKeeper for Microsoft Azure STONITH スクリプトをインストールします。

```
# /opt/LifeKeeper/samples/STONITH/azure-stonith-install
```

上記コマンドは対話的に動作しますので、グループ名を入力し、表示されるクラスターノードの Microsoft Azure 上での仮想マシン名を確認してください。

コマンド出力例

```
STONITH script install…
Please enter the Resource Group name in Azure: rg-Group1

Please enter the System name in Azure[vm-HostA]:
Enable Stonith on node vm-HostA [Yes]:
s
Please enter the System name in Azure[vm-HostB]:
Enable Stonith on node vm-HostB [Yes]:
```

```
Configuration file /opt/LifeKeeper/config/stonith.conf was saved.
```

### /opt/LifeKeeper/config/stonith.conf の更新

インストールが完了すると、仮想マシンの電源をオフにする設定が以下のファイルに追加されます。

/opt/LifeKeeper/config/stonith.conf

```
# LifeKeeper STONITH configuration
#
# Example: <host> az vm restart -g <resource group> -n <node name>
vm-HostA az vm stop -g rg-Group1 -n vm-HostA --skip-shutdown
vm-HostB az vm stop -g rg-Group1 -n vm-HostB --skip-shutdown
```

✻ 注意： フェンシングループを回避するために、再起動よりも「 **stop** 」を推奨します（2台のマシンが通信不能になっても、お互いに STONITH を実行でき、相互に電源をオフにして再起動します）。速やかにノードの電源を切るには、「 **skip-shutdown** 」の使用をお勧めします。

## 期待される動作

LifeKeeper がノードとの通信障害を検出すると、そのノードの電源がオフになり、フェールオーバーが発生します。問題が修復されたら、ノードの電源を手動でオンにする必要があります。

## STONITHの削除

1. */opt/LifeKeeper/events/prefailover* の下にあるすべてのフォルダーを削除します。

```
rm -r /opt/LifeKeeper/events/prefailover/*
```

2. */opt/LifeKeeper/config/stonith.conf* にある設定ファイルを削除します。

```
rm /opt/LifeKeeper/config/stonith.conf
```

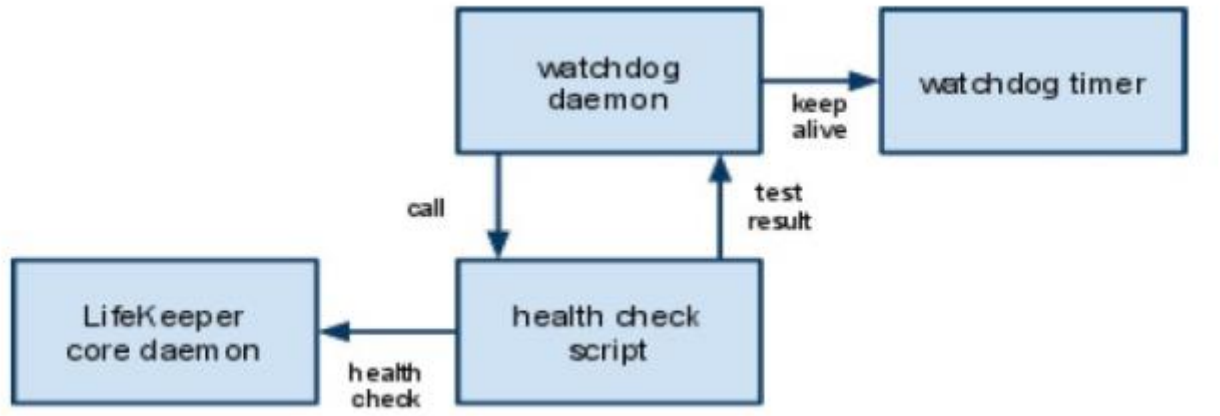
# 5.4.2.10.5. Watchdog

Watchdog は、サーバーが正常に動作しない場合に、問題の発生を予防する修正処置 (リブート) を確実に実行できるようにサーバーを監視する方法です。 Watchdog は、特別な Watchdog ハードウェアを使用して実装する場合と、ソフトウェアのみのオプションを使用して実装する場合があります。

> **注記:** この構成は、Red Hat Enterprise Linux Versions 7系とRed Hat Enterprise Linux Versions 8系 で検証されています。他のオペレーティングシステムでは検証されていないため、サポートされません。

## コンポーネント

- Watchdog timer - ソフトウェアドライバまたは外部ハードウェアコンポーネント
- Watchdog daemon - 該当する Linux ディストリビューションを通じて rpm が入手可能
- LifeKeeper Core daemon - LifeKeeper のインストールに付随
- health cehck script - LifeKeeper core の動作状況を確認するスクリプト




**LifeKeeper と Watchdog の相互運用性**

次のセクションを注意深く読んでください。デーモンは、エラーからリカバリするように設計されており、注意深く設定しないとデーモンはシステムをリセットします。インストールおよび設定を行う前に慎重に計画してください。このセクションの目的は、 Watchdog についての説明や設定をすることではありません。ここでは、 Watchdog 構成での LifeKeeper との相互運用についての説明や設定のみ行います。

## 設定

以下の手順は、root ユーザ権限を持つ管理者が行う必要があります。管理者は、 Watchdog のリスクおよび問題についてすでに熟知しているものとします。

ヘルスチェックスクリプト (LifeKeeper 監視スクリプト) は、LifeKeeper の設定と Watchdog の設定 (`/opt/LifeKeeper/samples/watchdog/LifeKeeper-watchdog`) を関連付けるコンポーネントです。このスクリプトは、LifeKeeper コアコンポーネントの基本部分を監視できます。

1. 以前に Watchdog を設定していた場合は、次のコマンドを入力して Watchdog を停止します。そうでない場合は、手順 2 に進みます。
   `systemctl stop watchdog`

2. Watchdog ソフトウェアのインストールで供給される Watchdog 設定ファイル (*/etc/watchdog.conf*) を編集します。

   a. test-binary を次のように修正します。

   ```
   test-binary = /opt/LifeKeeper/samples/watchdog/LifeKeeper-watchdog
   ```

   b. test-timeout を次のように修正します。

   ```
   test-timeout = 5
   ```

   c. interval を次のように修正します。

   ```
   interval = 7
   ```

   interval は、LifeKeeper のコミュニケーションパスのタイムアウト (15 秒) よりも小さい値にしてください。約半分の値が妥当です。

3. LifeKeeper が起動していることを確認します。まだの場合は、LifeKeeper の起動 トピックを参照してください。

4. 次のコマンドを入力して Watchdog を起動します。
   `systemctl start watchdog`

5. 今後の再起動の際に Watchdog を自動的に起動させるには、次のコマンドを入力します。
   `systemctl enable watchdog`

✻ **注記:** Watchdog を設定すると、予想外のリブートがときどき発生する可能性があります。これは、 Watchdog の仕組みから来る一般的な性質です。正常に応答しないプロセスがあると、 Watchdog 機能は LifeKeeper (またはオペレーティングシステム) がハングしていると判断し、(警告なしに) システムをリブートします。

## アンインストール

LifeKeeper をアンインストールする場合は、慎重に行ってください。以下に列記の通り、上記の手順を逆の順で実行します。

! **警告:** LifeKeeper を構成する RPM パッケージを削除する方法で LifeKeeper をアンインストールする場合は、 **先に Watchdog を停止してください** 。上記の手順 2 では、

LifeKeeper の Watchdog スクリプトを呼び出すように Watchdog 設定ファイルを修正しています。したがって、先に Watchdog を停止しておかないと、存在しないスクリプトを呼び出すことになります。リブートを実行するこのスクリプトが見つからない場合は、エラーが発生します。この状態は Watchdog を停止するまで継続します。

1. 次のコマンドを入力して Watchdog を停止します。
   `systemctl stop watchdog`

2. Watchdog ソフトウェアのインストールで供給される Watchdog 設定ファイル (*/etc/watchdog.conf*) を編集します。

   a. test-binary および interval の両エントリをコメントアウトします (各行の先頭に # を追加します)。

   ```
   #test-binary =
   #interval =
   ```

   **注記:** interval が他の機能によって以前から使用されていた場合は、そのままにしておくこともできます。

3. LifeKeeper をアンインストールします。LifeKeeper の削除 トピックを参照してください。

4. これで Watchdog を起動し直すことができます。LifeKeeper のみが Watchdog を使用していた場合は、次のコマンドを入力すると Watchdog を永続的に無効にできます。
   `systemctl disable watchdog`

# 5.4.2.10.6. I/O フェンシング機能組み合わせ情報

LifeKeeper for Linux では様々なフェンシング機能を用意しておりますが、LifeKeeper がインストールされるサーバーの構成や、保護対象となるサービスで使用されるストレージの構成により、使用できる適切なフェンシング機能やこれらの組み合わせは多岐にわたります。

各サーバー構成ごとに利用可能なフェンシング機能と適切な組み合わせについては、以下の情報を参照してください。またストレージ構成については各サーバー構成のページに詳細を記載しています。

- 物理サーバー構成の場合は「フェンシング機能利用可否リスト（物理構成）」を参照してください。
- 仮想サーバー構成（VMware）の場合は「フェンシング機能利用可否リスト（仮想構成 VMware編）」を参照してください。

## 各フェンシング機能概要

上述の情報で示される各 I/O フェンシング機能の概要を以下に記載します。

- **SCSI-2 Reserve** リソースがアクティブであるノードからストレージに対し SCIS-2 のリザーブコマンドを発行することにより保護対象の LU をロックし、他ノードからの同時アクセスを抑止します。
- **SCSI-3 Reserve** リソースがアクティブであるノードからストレージに対し SCSI-3 のリザーブコマンドを発行することにより保護対象の LU をロックし、他ノードからの同時アクセスを抑止します。
- **SCSI Fencing** 自ノードのリソースがスタンバイである状態においてコミュニケーションパスの全断を検知した際、他ノードから行われた LU のロックを無効化し、再び自ノードから SCSI-2 または SCSI-3 のリザーブコマンドを発行することにより、保護対象の LU をロックし、他ノードからの同時アクセスを抑止します。
- **IPMI STONITH** コミュニケーションパスの全断を検知した際、通信が途絶した他ノードの IPMI インタフェースに対して、物理サーバーの停止命令を発行することにより、保護対象サービスが両ノードで起動することを防止します。こちらの機能は IPMI ツールを用いるため、IPMI インタフェースを搭載した物理サーバーに対してのみ使用できます。
- **VMware STONITH** コミュニケーションパスの全断を検知した際、通信が途絶した他ノードが稼働する仮想ホストまたは vCenter に対して、仮想マシンの停止命令を発行することにより、保護対象サービスが両ノードで起動することを防止します。こちらの機能は VMware が提供するツールを用いるため、VMware 上に構成した仮想マシンに対してのみ使用できます。
- **Quorum Witness** 事前に調停役を設け、コミュニケーションパスの切断または復旧を検知した際、自ノードへのフェイルオーバーがサービス継続に有効であるかを判断します。LifeKeeper では通常、前述のリザーブコマンドにより複数ノードから LU への同時アクセスを抑止しますが、これらのリザーブを利用しない環境では本機能の利用が必須となります。調停役には3種類のモードがあ

り、Witness 専用ノード、独立ホスト、共有ストレージのいずれかをご利用環境にあわせて選択いただけます。詳しくは Quorum/Witness のページを参照してください。

# 5.4.2.10.6.1. フェンシング機能利用可否リスト（物理構成）

本ページでは、2台の物理サーバーをクラスターノードとする構成を例に、使用できる適切なフェンシング機能の組み合わせをストレージ構成ごとに記載します。

## 共有ディスク構成（SCSI リザーブを用いる単一または複数パスの構成）

本構成は、サポートストレージ一覧 の表に認定済み共有ストレージとして記載されているストレージで SCSI リザーブによる排他制御機能を利用する場合が該当します。
SCSI-3 Reserve および複数パスの制御のためにマルチパスドライバを必要とするストレージを利用する場合、それぞれのマルチパスドライバに対応したマルチパスキットが必要となります。

### 使用できる機能と適切な組み合わせ

以下の表では、本構成で使用できるフェンシング機能と適切な組み合わせパターンを示します。

### 各記号の意味

● – 利用必須
◯ – オプションとして利用可能
? – 利用不可

表中で「オプションとして利用可能」としている機能については、組み合わせの中で当該機能の利用は必須ではありません。表中で「利用不可」としている機能については、表下の注記を参照してください。

| | 組み合わせパターン | | | | | |
|---|---|---|---|---|---|---|
| **SCSI-2 Reservations** | ● | ● | ● | ?[1] | ?[1] | ?[1] |
| **SCSI-3 Reservations** | ?[1] | ?[1] | ?[1] | ● | ● | ● |
| **IPMI STONITH** | ?[2] | ?[2] | ?[2] | ?[2] | ?[2] | ?[2] |
| **Quorum/Witness (tcp_remote)** | ◯ | ?[3] | ?[3] | ◯ | ?[3] | ?[3] |
| **Quorum/Witness (majority)** | ?[3] | ◯ | ?[3] | ?[3] | ◯ | ?[3] |
| **Quorum/Witness (storage)** | ?[3] | ?[3] | ◯ | ?[3] | ?[3] | ◯ |

[1] LifeKeeper の仕組み上、単一の共有ディスクに対して SCSI-2 および SCSI-3 のリザーブは共存することができません。

[2] 競合によるサービス停止の可能性がある為、SCSI Fencing と IPMI STONITH は共存することができません。

[3] LifeKeeper の仕組み上、単一のクラスターに対して Quorum Witness の各モードは共存することができ

ません。

### 構成上使えない機能について

- 仮想環境やクラウド環境向けに用意されたフェンシング機能は、本構成では使用できません。

## 共有ディスク構成（SCSIリザーブを用いない単一または複数パスの構成）

本構成は、SCSI / FC / iSCSI / SAS などの方式（NASを除く）により接続されるストレージで、SCSIリザーブによる排他制御機能を利用できない場合が該当します。
複数パスの制御のためにマルチパスドライバを必要とするストレージを利用する場合、それぞれのマルチパスドライバに対応したマルチパスキットが必要となります。

### 使用できる機能と適切な組み合わせ

以下の表では、本構成で使用できるフェンシング機能と適切な組み合わせパターンを示します。

### 各記号の意味

● – 利用必須
◯ – オプションとして利用可能
? – 利用不可

表中で「オプションとして利用可能」としている機能については、組み合わせの中で当該機能の利用は必須ではありません。表中で「利用不可」としている機能については、表下の注記を参照してください。

| | 組み合わせパターン | | |
|---|---|---|---|
| **IPMI STONITH** | ◯ | ◯ | ◯ |
| **Quorum/Witness (tcp_remote)** | ● | ?[1] | ?[1] |
| **Quorum/Witness (majority)** | ?[1] | ● | ?[1] |
| **Quorum/Witness (storage)** | ?[1] | ?[1] | ● |

[1] LifeKeeper の仕組み上、単一のクラスタに対して Quorum Witness の各モードの共存は不可

### 構成上使えない機能について

- SCSI-2 Reserve および SCSI-3 Reserve は、SCSIリザーブを用いない前提の本構成では使用できません。
- SCSI Fencing は、SCSI リザーブを用いない前提の本構成では使用できません。
- 仮想環境やクラウド環境向けに用意されたフェンシング機能は、本構成では使用できません。

## データレプリケーション構成

本構成は、DataKeeper を用いて各ノードに接続されたローカルストレージをノード間でレプリケーションする場合が該当します。

### 使用できる機能と適切な組み合わせ

以下の表では、本構成で使用できるフェンシング機能と適切な組み合わせパターンを示します。本構成において必須となるフェンシング機能はありませんが、データ保護の観点からフェンシング機能を使用されることを推奨します。

### 各記号の意味

● – 利用必須
◯ – オプションとして利用可能
? – 利用不可

表中で「オプションとして利用可能」としている機能については、組み合わせの中で当該機能の利用は必須ではありません。表中で「利用不可」としている機能については、表下の注記を参照してください。

| | 組み合わせパターン | | |
|---|---|---|---|
| **IPMI STONITH** | ◯ | ◯ | ◯ |
| **Quorum/Witness (tcp_remote)** | ◯ | ?[1] | ?[1] |
| **Quorum/Witness (majority)** | ?[1] | ◯ | ?[1] |
| **Quorum/Witness (storage)** | ?[1] | ?[1] | ◯ |

[1] LifeKeeper の仕組み上、単一のクラスターに対して Quorum Witness の各モードは共存することができません。

### 構成上使えない機能について

- SCSI リザーブを用いるフェンシング機能は本構成では使用できません。
- 仮想環境やクラウド環境向けに用意されたフェンシング機能は、本構成では使用できません。

## Network Attached Storage (NAS) 構成

本構成は、Network File System (NFS) の方式により接続される NAS ストレージを利用する場合が該当します。

### 使用できる機能と適切な組み合わせ

以下の表では、本構成で使用できるフェンシング機能と適切な組み合わせパターンを示します。本構成において必須となるフェンシング機能はありませんが、データ保護の観点からフェンシング機能を使用されることを推奨します。

### 各記号の意味

● – 利用必須
◯ – オプションとして利用可能
? – 利用不可

表中で「オプションとして利用可能」としている機能については、組み合わせの中で当該機能の利用は必須ではありません。表中で「利用不可」としている機能については、表下の注記を参照してください。

| | 組み合わせパターン | | |
|---|---|---|---|
| **IPMI STONITH** | ◯ | ◯ | ◯ |
| **Quorum/Witness (tcp_remote)** | ◯ | ?[1] | ?[1] |
| **Quorum/Witness (majority)** | ?[1] | ◯ | ?[1] |
| **Quorum/Witness (storage)** | ?[1] | ?[1] | ◯ |

[1] LifeKeeper の仕組み上、単一のクラスターに対してQuorum Witness の各モードは共存することができません。

### 構成上使えない機能について

- 共有ディスクの使用を前提としたフェンシング機能は、本構成では使用できません。
- 仮想環境やクラウド環境向けに用意されたフェンシング機能は、本構成では使用できません。

# 5.4.2.10.6.2. フェンシング機能利用可否リスト（仮想構成 VMware編）

本ページでは、VMware 上の仮想マシンとして作成された2台のサーバーをクラスターノードとする構成を例に、使用できる適切なフェンシング機能の組み合わせをストレージ構成ごとに記載します。

## 共有ディスク構成（SCSI リザーブを用いる単一または複数パスの構成）

本構成は、サポートストレージ一覧 の表に認定済み共有ストレージとして記載されているストレージで SCSI リザーブによる排他制御機能を利用する場合が該当します。
SCSI-3 Reserve および複数パスの制御のためにマルチパスドライバを必要とするストレージを利用する場合、それぞれのマルチパスドライバに対応したマルチパスキットが必要となります。

### 使用できる機能と適切な組み合わせ

以下の表では、本構成で使用できるフェンシング機能と適切な組み合わせパターンを示します。

### 各記号の意味

● – 利用必須
◯ – オプションとして利用可能
? – 利用不可

表中で「オプションとして利用可能」としている機能については、組み合わせの中で当該機能の利用は必須ではありません。表中で「利用不可」としている機能については、表下の注記を参照してください。

| | 組み合わせパターン | | | | | |
|---|---|---|---|---|---|---|
| **SCSI-2 Reservations** | ● | ● | ● | ?[1] | ?[1] | ?[1] |
| **SCSI-3 Reservations** | ?[1] | ?[1] | ?[1] | ● | ● | ● |
| **VMware STONITH** | ?[2] | ?[2] | ?[2] | ?[2] | ?[2] | ?[2] |
| **Quorum/Witness (tcp_remote)** | ◯ | ?[3] | ?[3] | ◯ | ?[3] | ?[3] |
| **Quorum/Witness (majority)** | ?[3] | ◯ | ?[3] | ?[3] | ◯ | ?[3] |
| **Quorum/Witness (storage)** | ?[3] | ?[3] | ◯ | ?[3] | ?[3] | ◯ |

[1] LifeKeeper の仕組み上、単一の共有ディスクに対して SCSI-2 および SCSI-3 のリザーブは共存することができません。

[2] 競合によるサービス停止の可能性がある為、SCSI Fencing と VMware STONITH は共存することができません。

[3] LifeKeeper の仕組み上、単一のクラスターに対して Quorum Witness の各モードは共存することができ

ません。

### 構成上使えない機能について

- VMware以外の仮想環境向けに用意されたフェンシング機能は、本構成では使用できません。
- クラウド環境や物理サーバー向けに用意されたフェンシング機能は、本構成では使用できません。

## 共有ディスク構成（SCSIリザーブを用いない単一または複数パスの構成）

本構成は、SCSI / FC / iSCSI / SAS などの方式（NASを除く）により接続されるストレージで、SCSIリザーブによる排他制御機能を利用できない場合が該当します。
複数パスの制御のためにマルチパスドライバを必要とするストレージを利用する場合、それぞれのマルチパスドライバに対応したマルチパスキットが必要となります。

### 使用できる機能と適切な組み合わせ

以下の表では、本構成で使用できるフェンシング機能と適切な組み合わせパターンを示します。

### 各記号の意味

● – 利用必須
◯ – オプションとして利用可能
? – 利用不可

表中で「オプションとして利用可能」としている機能については、組み合わせの中で当該機能の利用は必須ではありません。表中で「利用不可」としている機能については、表下の注記を参照してください。

| | 組み合わせパターン | | |
|---|---|---|---|
| **VMware STONITH** | ◯ | ◯ | ◯ |
| **Quorum/Witness (tcp_remote)** | ● | ?[1] | ?[1] |
| **Quorum/Witness (majority)** | ?[1] | ● | ?[1] |
| **Quorum/Witness (storage)** | ?[1] | ?[1] | ● |

[1] LifeKeeper の仕組み上、単一のクラスタに対して Quorum Witness の各モードの共存は不可

### 構成上使えない機能について

- SCSI-2 Reserve および SCSI-3 Reserve は、SCSIリザーブを用いない前提の本構成では使用できません。
- SCSI Fencing は、SCSI リザーブを用いない前提の本構成では使用できません。
- 仮想環境やクラウド環境向けに用意されたフェンシング機能は、本構成では使用できません。

## データレプリケーション構成

本構成は、DataKeeper を用いて各ノードに接続されたローカルストレージをノード間でレプリケーションする場合が該当します。

### 使用できる機能と適切な組み合わせ

以下の表では、本構成で使用できるフェンシング機能と適切な組み合わせパターンを示します。本構成において必須となるフェンシング機能はありませんが、データ保護の観点からフェンシング機能を使用されることを推奨します。

### 各記号の意味

● – 利用必須
◯ – オプションとして利用可能
? – 利用不可

表中で「オプションとして利用可能」としている機能については、組み合わせの中で当該機能の利用は必須ではありません。表中で「利用不可」としている機能については、表下の注記を参照してください。

| | 組み合わせパターン | | |
|---|---|---|---|
| **VMware STONITH** | ◯ | ◯ | ◯ |
| **Quorum/Witness (tcp_remote)** | ◯ | ?[1] | ?[1] |
| **Quorum/Witness (majority)** | ?[1] | ◯ | ?[1] |
| **Quorum/Witness (storage)** | ?[1] | ?[1] | ◯ |

[1] LifeKeeper の仕組み上、単一のクラスターに対して Quorum Witness の各モードは共存することができません。

### 構成上使えない機能について

- SCSIリザーブを用いるフェンシング機能は本構成では使用できません。
- 仮想環境やクラウド環境向けに用意されたフェンシング機能は、本構成では使用できません。

## Network Attached Storage (NAS) 構成

本構成は、Network File System (NFS) の方式により接続される NAS ストレージを利用する場合が該当します。

### 使用できる機能と適切な組み合わせ

以下の表では、本構成で使用できるフェンシング機能と適切な組み合わせパターンを示します。本構成において必須となるフェンシング機能はありませんが、データ保護の観点からフェンシング機能を使用されることを推奨します。

### 各記号の意味

● – 利用必須
◯ – オプションとして利用可能
? – 利用不可

表中で「オプションとして利用可能」としている機能については、組み合わせの中で当該機能の利用は必須ではありません。表中で「利用不可」としている機能については、表下の注記を参照してください。

| | 組み合わせパターン | | |
|---|---|---|---|
| **VMware STONITH** | ◯ | ◯ | ◯ |
| **Quorum/Witness (tcp_remote)** | ◯ | ?[1] | ?[1] |
| **Quorum/Witness (majority)** | ?[1] | ◯ | ?[1] |
| **Quorum/Witness (storage)** | ?[1] | ?[1] | ◯ |

[1] LifeKeeper の仕組み上、単一のクラスターに対してQuorum Witness の各モードは共存することができません。

### 構成上使えない機能について

- 共有ディスクの使用を前提としたフェンシング機能は、本構成では使用できません。
- 仮想環境やクラウド環境向けに用意されたフェンシング機能は、本構成では使用できません。

## VMDK as Shared Storage 構成

本構成は、VMDK as Shared Storage の方式により構成される VMware仮想ハードディスクを利用する場合が該当します。

### 使用できる機能と適切な組み合わせ

以下の表では、本構成で使用できるフェンシング機能と適切な組み合わせパターンを示します。

### 各記号の意味

● – 利用必須
◯ – オプションとして利用可能
? – 利用不可

表中で「オプションとして利用可能」としている機能については、組み合わせの中で当該機能の利用は必須ではありません。表中で「利用不可」としている機能については、表下の注記を参照してください。

| | 組み合わせパターン | | |
|---|---|---|---|
| **VMware STONITH** | ?[1] | ?[1] | ?[1] |
| **Quorum/Witness (tcp_remote)** | ○ | ?[2] | ?[2] |
| **Quorum/Witness (majority)** | ?[2] | ○ | ?[2] |
| **Quorum/Witness (storage)** | ?[2] | ?[2] | ○ |

[1] 競合によるサービス停止の可能性がある為、VMDK as Shared Storage と VMware STONITH は共存することができません。

[2] LifeKeeper の仕組み上、単一のクラスターに対してQuorum Witness の各モードは共存することができません。

### 構成上使えない機能について

- 共有ディスクの使用を前提としたフェンシング機能は、本構成では使用できません。

# 5.4.2.11. リソースポリシー管理

## 概要

LifeKeeper for Linux のリソースポリシー管理では、リソースのローカルリカバリーとフェイルオーバーの動作管理機能が提供されます。リソースポリシーは、 **lkpolicy** コマンドラインツール (CLI) を使用して管理できます。

## LifeKeeper

LifeKeeperには、個々のアプリケーションおよび関連し合うアプリケーションのグループを監視する機能があり、定期的にローカルリカバリーを実行したり、保護下のアプリケーションに障害が発生したときに通知することができます。関連し合うアプリケーションの例としては、主アプリケーションが下位のストレージまたはネットワークリソースに依存する階層などがあります。アプリケーションまたはリソースに障害が発生した場合のデフォルトの動作は以下の通りです。

1. **ローカルリカバリー**: 最初に、リソースまたはアプリケーションの **ローカル** でリカバリーを試みます。このときは、外部の介入なしにローカルサーバー上でリソースまたはアプリケーションをリストアしようとします。ローカルリカバリが成功した場合、LifeKeeper for Linux は追加のアクションを実行しません。

2. **フェイルオーバー**: 次に、ローカルリカバリーでリソースまたはアプリケーションのリストアに失敗した (またはリソースを監視するリカバリキットがローカルリカバリーをサポートしていない) 場合、 **フェイルオーバー** は開始されます。フェイルオーバー処理では、クラスタ内の別のサーバー上で該当アプリケーション (および依存するすべてのリソース) を起動しようと試みます。

リカバリ動作の詳細については、LifeKeeper for Linux 障害検出とリカバリのシナリオ を参照してください。

## ポリシーによるカスタム動作およびメンテナンスモード動作

LifeKeeper for Linux Version 7.5 以降では、デフォルトのリカバリ動作を変更する追加ポリシーを設定する機能をサポートします。リソース単位またはサーバ単位で、4 つのポリシーが設定可能です (リソース単位のポリシーに関する注意については下のセクションを参照してください)。 **サーバーレベルでポリシーを変更する方法を推奨します。**

利用可能なポリシーは以下の通りです。

### 標準ポリシー

- **Failover** - このポリシー設定を使用すると、リソースフェイルオーバーを有効 / 無効にできます。(**Note**: リザベーションが適切に処理されるには、 **フェイルオーバー** は、個々の SCSI リソースで無効にすることはできません。)

- **LocalRecovery** - LifeKeeper for Linux は、デフォルトでは、フェイルオーバーを実行する前に、個々のリソースまたは保護対象アプリケーション全体を再起動することにより、保護対象リソースの

リカバリーを試みます。このポリシー設定を使用すると、ローカルリカバリを有効 / 無効にできます。

- **TemporalRecovery** - 通常、LifeKeeper for Linux は、障害リソースのローカルリカバリーを実行します。ローカルリカバリーに失敗すると、LifeKeeper は、リソース階層を別ノードにフェイルオーバーします。ローカルリカバリーに成功した場合は、フェイルオーバーは実行されません。

  ローカルリカバリーに成功した場合でも、サーバーの何らかの異常によって短時間の間にローカルリカバリーが再試行される場合があり、結果として何度も連続してローカルリカバリーが試行されることになります。これが発生すると、問題のアプリケーションは可用性が悪化します。

  この反復的なローカルリカバリー / 障害サイクルを回避するために、時間的リカバリポリシーを設定できます。時間的リカバリーポリシーを使用すると、管理者は指定した時間内に試行するローカルリカバリーの回数を (成功かどうかにかかわらず) 制限することができます。

    *例:* リソースが試行するローカルリカバリーの回数を 30 分間で 3 回に限定するポリシー定義をユーザが設定した場合、30 分以内に 3 回目のローカルリカバリーが試行されると、LifeKeeper for Linux はフェイルオーバを実行します。

  定義した時間的リカバリーポリシーは有効または無効にできます。時間的リカバリーポリシーが無効の場合、時間的リカバリ処理は継続して実行され、ポリシーが適用されるはずの時間に通知がログに表示されますが、実際のアクションは実行されません。

**注記:** 時間的リカバリーポリシーを設定した状態で、フェイルオーバーとローカルリカバリーの一方または両方を無効にすることは可能です。フェイルオーバーまたはローカルリカバリーを無効にした場合に、時間的リカバリポリシーは実行されることが **ない** ため、この状態は非論理的です。

### メタポリシー

「メタ」ポリシーは、他の複数のポリシーに影響を与える可能性があるポリシーです。通常これらのポリシーは、標準ポリシーであれば複数個の設定が必要になるような特定のシステム動作を実現するためのショートカットとして使用します。

- **NotificationOnly** - このモードでは、管理者は LifeKeeper を「監視専用」状態にすることができます。 **1 つのリソース (または、サーバ単位のポリシーの場合はすべてのリソース) のローカルリカバリーおよびフェイルオーバーの両方が影響を受けます。** 障害が検知されると、ユーザーインターフェースには **Failure** 状態が表示されます。 ただし、リカバリーもフェイルオーバーも実行されません。 **注記:** 管理者は、障害の原因となった問題を手動で修正し、障害が起きたリソースを復帰させて通常の LifeKeeper の運用を継続する必要があります。

## リソースレベルのポリシーに関する重要な考慮事項

リソースレベルのポリシーとは、リソース階層全体またはサーバーレベルのポリシーとは異なり、特定のリソースにのみ適用されるポリシーです。

例：

アプリケーション

- IP

- file system

上記のリソース階層では、アプリケーションは IP とファイルシステムの両方に依存しています。ポリシーは、特定のリソースのローカルリカバリーまたはフェイルオーバーを無効にするように設定できます。これは、例えば、IP リソースのローカルリカバリーが失敗し、IP リソースのフェイルオーバーが無効に設定されていた場合、IP リソースはフェイルオーバーを実行せず、他のリソースのフェイルオーバーも発生させないことを意味します。ただし、ファイルシステムリソースのローカルリカバリーが失敗し、ファイルシステムリソースのポリシーのフェイルオーバーが無効化されていない場合、階層全体がフェイルオーバーを実行します。

**注記:** 重要事項として、リソースレベルのポリシーは設定対象の特定のリソースにのみ適用されることに注意してください。

上記は単純な例です。複雑な階層を構成することもできるため、リソースレベルのポリシーを設定するときは注意してください。

既知の問題を参照してください。 *(“フェイルオーバー時に誤った順序でリソースが停止されます“の項目)*

# lkpolicy ツール

**lkpolicy** ツールは、LifeKeeper が稼働するサーバのポリシーを管理 (参照、設定、削除) するためのコマンドラインツールです。`lkpolicy` は、ポリシーの設定および修正、ポリシーの削除、利用可能なポリシーと現在の設定値の表示をサポートします。さらに、設定したポリシーは有効または無効に設定できるため、リカバリー動作に影響を与えながらリソース / サーバー設定を保持できます。

全体的な使用方法は次の通りです。

```
lkpolicy [--list-policies | --get-policies | --set-policy | --remove-policy]
<name value pair data…>
```

`<name value pair data…>`は、運用方法および対象のポリシーによって異なります (特にポリシーを設定する場合)。例： 有効 / 無効タイプのポリシーのほとんどでは、必要なのは `[--on]` または `[--off]` のスイッチのみですが、時間的ポリシーの場合は、しきい値を設定するための値も必要です。

## lkpolicy の使用方法の例

### ローカルおよびリモートサーバーとの認証

**lkpolicy** ツールは、サーバが公開する API を通じて LifeKeeper サーバと通信します。この API は、

lkpolicy ツールなどのクライアントに対して認証を要求します。 lkpolicy ツールで LifeKeeper サーバーに最初にアクセスしようとしたときに、そのサーバーに対する認証情報がまだ保存されていない場合、ユーザーは認証情報を求められます。認証情報はユーザー名とパスワードの形式であり、さらに以下の条件があります。

1. クライアントには LifeKeeper の管理者権限が必要です。したがって、そのユーザー名は、( *PAM* による) オペレーティングシステムの認証設定によって *lkadmin グループ* に属する必要があります。必ずしも ***root*** で実行する ***必要はありません*** が、root ユーザはデフォルトで適切なグループに属しているため、root を使用することもできます。
2. 認証情報は認証情報ストアに保存されるため、ツールを使用してこのサーバにアクセスするたびに手動で認証情報を入力する必要はありません。

認証情報ストアと credstore ユーティリティによる管理の詳細については、LifeKeeper の認証情報の設定を参照してください。

lkpolicy によるセッションの例は以下のようになります。

```
[root@thor49 ~]# lkpolicy -l -d v6test4
Please enter your credentials for the system 'v6test4'.
Username: root
Password:
Confirm password:
Failover
LocalRecovery
TemporalRecovery
NotificationOnly
[root@thor49 ~]# lkpolicy -l -d v6test4
Failover
LocalRecovery
TemporalRecovery
NotificationOnly
[root@thor49 ~]#
```

## ポリシーのリスト表示

lkpolicy --list–policy–types

## 現在のポリシーの表示

lkpolicy --get–policies

lkpolicy --get–policies tag=*

lkpolicy --get–policies --verbose tag=mysql* # all resources starting with mysql

lkpolicy --get–policies tag=mytagonly

## ポリシーの設定

lkpolicy --set–policy Failover --off

lkpolicy --set–policy Failover --on tag=myresource

lkpolicy --set–policy Failover --on tag=*

lkpolicy --set–policy LocalRecovery --off tag=myresource

lkpolicy --set–policy NotificationOnly --on

lkpolicy --set–policy TemporalRecovery --on recoverylimit=5 period=15

lkpolicy --set–policy TemporalRecovery --on --force recoverylimit=5 period=10

## ポリシーの削除

lkpolicy --remove–policy Failover tag=steve

注記: *NotificationOnly* はポリシーのエイリアスです。 *NotificationOnly* を有効にすることは、対応する *LocalRecovery* および *Failover* ポリシーを無効にすることと等価です。

# 5.4.2.12. 認証情報の設定

他のシステムと通信するための認証情報は、認証情報ストアを使用して管理されています。このストアは、必要に応じて /opt/LifeKeeper/bin/credstore ユーティリティで管理できます。このユーティリティを使用すると、サーバアクセスに必要な認証情報をサーバごとに設定、変更、削除することができます。

## 認証情報の追加または変更

認証情報の追加と変更は同じ方法で実行できます。代表的な例として、サーバー server.mydomain.com に対する資格情報を追加または変更する場合は次のようになります。

```
/opt/LifeKeeper/bin/credstore -k server.mydomain.com myuser
```

この例では、server.mydomain.com へのアクセスに使用するユーザ名として *myuser* を指定して しています。パスワードを入力 / 確認するプロンプト (*passwd* の様に) が表示されます。

> 注記: LifeKeeper サーバの 認証情報 を格納するために使用されるキー名は、lkpolicy などのコマンドで使用するホスト名と完全に一致する必要があります。コマンドで使用するホスト名が FQDN であれば、認証情報のキーも FQDN でなければなりません。コマンドで使用するホスト名がショートネームであれば、認証情報のキーもショートネームでなければなりません。

認証情報ストアに デフォルト キーを作成することもできます。サーバキーが存在しない場合に デフォルト の認証情報が認証に使用されます。 デフォルト キーを追加、変更するには以下のコマンドを実行してください。

```
/opt/LifeKeeper/bin/credstore -k default myuser
```

## ストア内の認証情報のリスト表示

現在格納されている認証情報をリスト表示するには、以下のコマンドを実行します。

```
/opt/LifeKeeper/bin/credstore -l
```

これにより、認証情報ストア内に格納されているキーが表示されます。この場合の「キー」は、認証情報を使用する対象のサーバを示しています (認証情報自体は秘密情報のため、このコマンドが表示するのは、実際の認証情報の内容ではなくキー名のみです)。

## サーバの認証情報の削除

特定のサーバに対する認証情報を削除するには、以下のコマンドを実行します。

```
/opt/LifeKeeper/bin/credstore -d -k myserver.mydomain.com
```

この例では、サーバ myserver.mydomain.com に対する認証情報がストアから削除されます。

# 追加情報

credstore ユーティリティの詳細については、以下のコマンドを実行してください。

```
/opt/LifeKeeper/bin/credstore --man
```

コマンドのマニュアルページがすべて表示されます。

# 5.4.2.13. Standby Node Health Check

## 機能の概要

LifeKeeper Core の既存の監視機能に Standby Node Health Check を組み合わせることにより、Standby ノードにおける CPU や Memory の使用率の監視や Out of service のリソースの健全性の監視を行い、Standbyノードの異常を検知できます。これにより、事前に問題を解消することができ、Activeノードで異常が発生した際のフェイルオーバーに失敗するリスクを軽減できます。この監視処理は、通常の LifeKeeper の監視実行間隔（ LKCHECKINTERVAL の設定値）と同じ間隔で実行されます。

Standby Node Health Check は、主に以下の二つの機能に分類されます。

### Node監視

ノード上ですべてのリソースが Out of service の場合、Standbyノードと判断して Node監視のスクリプトを呼び出します。Node監視のスクリプトが CPU や Memory の使用率の監視を行い、ノードが正常に切り替えが出来ない可能性が高い状態にあると判断すると、異常ありと判断され、予め設定されたメール通知機能、またはSNMPのイベント転送機能を通じて、検知した内容を通知します。詳細は「 Node監視 」を参照してください。

### OSUリソース監視

Out of service の各リソースについて、lkcheck が定期的にそのリソースタイプの OSUquickCheck スクリプトを呼び出します。OSUquickCheck スクリプトがリソースのクイック健全性チェックを実行し、リソースが正常に起動できない状態にあると判断すると、該当リソースのステータスを OSF に変更し、異常ありと判断され、予め設定されたメール通知機能、またはSNMPのイベント転送機能を通じて、検知した内容を通知します。詳細は「 OSUリソース監視 」を参照してください。

## インストールと設定

特別なインストールは必要ありません。
Standby Node Health Check 機能をセットアップする一般的なプロセスには以下の手順が含まれます。

1. イベントメール通知 や SNMPによるイベント転送 の設定を行います。
2. Standby Node Health Check の設定を行います。（詳細は次項の「Standby Node Health Check の設定」を参照してください。）
3. 既に LifeKeeper が起動している場合は、設定を反映させるため lkcheck プロセスを再起動します。lkcheck プロセスの再起動には次のコマンドを実行してください。

```
killall lkcheck
```

上記の手順が完了すると、そのノードでは Standby Node Health Check 機能が動作するようになります。設定に応じてNode監視や Out of service の各リソースの監視が行われます。詳細は「 Node監視 」と「 OSUリソース監視 」のページを参照してください。

## Standby Node Health Check の設定

/etc/default/LifeKeeper 設定ファイルで SNHC の設定を有効に設定してください。詳細は「 Standby Node Health Check パラメータ一覧 」を参照してください。

# 5.4.2.13.1. Node監視

ノード上ですべてのリソースが Out of service の場合、Standbyノードと判断して Node監視のスクリプトを呼び出します。Node監視のスクリプトが CPU や Memory の使用率の監視を行い、ノードが正常に切り替えが出来ない可能性が高い状態にあると判断すると、異常ありと判断され、予め設定されたメール通知機能、またはSNMPのイベント転送機能を通じて、検知した内容を通知します。この監視処理は、通常のLifeKeeperの監視実行間隔（ LKCHECKINTERVAL の設定値）と同じ間隔で実行されます。

## 監視可能な項目

Node監視で監視できる項目は以下です。

| リソース名 | 監視内容 |
|---|---|
| CPU使用率 | /proc/stat ファイルから、監視実行時のCPU使用率を確認します。 |
| Memory使用率 | /proc/meminfo ファイルから、監視実行時のMemory使用率を確認します。 |

## Node監視の設定

/etc/default/LifeKeeper 設定ファイルで SNHC_CPUCHECK や SNHC_MEMCHECK の設定を有効に設定してください。また、他に以下の設定があります。詳細は「 Standby Node Health Check パラメータ一覧 」を参照してください。

- SNHC_CPUCHECK_THRESHOLD
- SNHC_CPUCHECK_TIME
- SNHC_MEMCHECK_THRESHOLD
- SNHC_MEMCHECK_TIME

# 5.4.2.13.2. OSU リソース監視

Out of service（ステータスが OSU） の各リソースについて、lkcheck が定期的にそのリソースタイプの OSUquickCheck スクリプトを呼び出します。OSUquickCheck スクリプトがリソースのクイック健全性チェックを実行し、リソースが正常に切り替えが出来ない可能性が高い状態にあると判断すると、該当リソースのステータスを OSF に変更し、異常ありと判断され、予め設定されたメール通知機能、または SNMP のイベント転送機能を通じて、検知した内容を通知します。この監視処理は、通常の LifeKeeper の監視実行間隔（LKCHECKINTERVAL の設定値）と同じ間隔で実行されます。

## 監視可能なリソース

現在は、以下の表に記載されているリソースが OSU リソース監視の対象となっています。

| リソース名 | 監視内容 |
|---|---|
| IPリソース | IPリソースの対象となる NIC リンクの状態を確認します（IP_NOLINKCHECK=1 の場合は確認しません）。<br>ping が可能かつ ping list が設定されている場合は、登録されているアドレスへの到達性を確認します。実 IP リソースは監視対象外です。 |
| Disk または DMMP リソース | 各リソースに対応したコマンドを用いて対象の Disk へのパスを確認します。 |
| NASリソース | NFSサーバーに対してNFS接続ができるかを確認します。NFS接続のタイムアウト値に関しては、 設定上の考慮事項 をご参照ください。 |

## OSUリソース監視の設定

/etc/default/LifeKeeper 設定ファイルで SNHC_IPCHECK や SNHC_DISKCHECK の設定を有効に設定してください。また、他に以下の設定があります。詳細は「 Standby Node Health Check パラメータ一覧 」を参照してください。

- SNHC_IPCHECK_SLEEPTIME

## 障害からの復帰

OSU リソース監視で異常が検知された場合、該当リソースのステータスが OSF（Out of service かつ障害あり）に変更されます。ステータスが変更されると、該当のリソースに対して OSU リソース監視は実行されなくなります。障害検知の通知の内容を確認して障害を取り除いた後、該当リソースのステータスを OSU に復帰させる必要があります。

以下のコマンドを実行してステータスを OSF から OSU に復帰できます。

```
/opt/LifeKeeper/lkadm/bin/retstate <resource tag>
```

# 5.4.3. LifeKeeper 管理

## 概要

LifeKeeper は操作時に管理を必要としません。LifeKeeper は、保護されたリソースを監視し、障害が発生した場合に指定されたリカバリアクションを実行するように、自動的に機能します。以下のケースでは LifeKeeper GUI を使用します。

- **リソースおよび階層の定義。** LifeKeeper は次のインターフェースオプションを提供します。
  - LifeKeeper GUI。
  - LifeKeeper コマンドラインインターフェース。
- **リソース監視。** LifeKeeper GUI は、リソースステータス情報および LifeKeeper ログへのアクセスを提供します。
- **手動での処理。** メンテナンスやその他の管理アクションのために、サーバまたは特定のリソースを停止することが必要になる場合があります。LifeKeeper GUI には、特定のリソースを稼動させたり停止させることができるメニュー機能が用意されています。アプリケーションが LifeKeeper の保護下に置かれると、これらの LifeKeeper のインターフェースを介してのみアプリケーションを起動および停止させることができます。LifeKeeper の起動および停止は、コマンドラインを介してのみ行われます。

LifeKeeper の管理、設定、およびメンテナンス操作を実行する詳細な手順については、GUI の作業 およびメンテナンス作業 を参照してください。

> 注記 : LifeKeeperが提供しているコマンド（実行可能なスクリプトやプログラム）を実行するには、スーパーユーザ権限が必要です。

su コマンドや sudo でスーパーユーザ権限を付与したユーザで、LifeKeeper のコマンドを実行することは可能ですが、SIOS Technology Corp では、root ユーザ以外で LifeKeeper のコマンドのテストはしていません。

---

エラーの検出および通知

N-way リカバリ

管理作業

サーバプロパティの編集

コミュニケーションパスの作成

コミュニケーションパスの削除

サーバのプロパティ - フェイルオーバ

リソース階層の作成

ファイルシステムリソース階層の作成

Generic Application リソース階層の作成

Raw デバイスリソース階層の作成

Quick Service Protection (QSP) Recovery Kit

リソースプロパティの編集

リソース優先順位の編集

リソース階層の拡張

ファイルシステムリソース階層の拡張

Generic Application リソース階層の拡張

Raw デバイスリソース階層の拡張

階層の拡張解除

リソース依存関係の作成

リソース依存関係の削除

すべてのサーバからの階層の削除

# 5.4.3.1. エラーの検出および通知

アプリケーション内の問題を検出して通知する機能は、最適な総合的耐障害性ソリューションを構築する上で非常に重要です。すべての個々のアプリケーションは、障害発生のメカニズムと形式によって異なるため、一般的なメカニズムを示すことはできません。ただし、一般的に、多くのアプリケーションの設定は、LifeKeeper に用意されている Core システムのエラー検出機能を利用することができます。リソースエラー回復シナリオ および サーバ障害回復シナリオ の各トピックでは、共通する 2 つの障害状況を使用して、LifeKeeper のコア機能を示しています。

LifeKeeper には、エラー、アラーム、およびリカバリ手順を起動するイベントを定義するための完全な環境も用意されています。このインターフェースは、通常、システムエラーログ用のパターンマッチ定義 (`/var/log/messages`)、またはカスタムビルドのアプリケーション固有の監視プロセスが必要になります。

# 5.4.3.2. N-Way リカバリ

N-Way リカバリを使用すると、異なる複数のリソースを、クラスタ内の異なるバックアップサーバにフェイルオーバすることができます。

保護されたリソース に戻る

# 5.4.3.3. 管理作業

サーバプロパティの編集

コミュニケーションパスの作成

コミュニケーションパスの削除

サーバのプロパティ - フェイルオーバ

リソース階層の作成

リソースプロパティの編集

リソース優先順位の編集

リソース階層の拡張

階層の拡張解除

リソース依存関係の作成

リソース依存関係の削除

すべてのサーバからの階層の削除

# 5.4.3.3.1. サーバプロパティの編集

1. サーバプロパティを編集するには、サーバプロパティを表示する 場合と同様に、[Server Properties] ダイアログを表示してください。

2. 該当のサーバに適切な権限でログインした場合は、次の項目が編集可能になります。
   - シャットダウン方法
   - フェイルオーバ確認

3. 変更が加えられると、 **[Apply]** ボタンが有効になります。このボタンをクリックすると変更が適用されます。ウィンドウは閉じられません。

4. 完了したら、 **[OK]** をクリックし、変更内容を保存してウィンドウを閉じるか、または **[Cancel]** をクリックして、変更内容を適用せずにウィンドウを閉じます。

# 5.4.3.3.2. コミュニケーションパスの作成

サーバ間の LifeKeeper コミュニケーションパスを設定する前に、ハードウェアおよびソフトウェアのセットアップを検証します。詳細については、SPS for Linuxリリースノート を参照してください。

サーバのペア間にコミュニケーションパスを作成するには、両方のサーバに個別にパスを定義する必要があります。LifeKeeper では、サーバのペア間に TCP (TCP/IP) と TTY の両方のコミュニケーションパスを作成することができます。TTY パスは、所定のペア間に 1 つだけ作成できます。これに対し、TCP コミュニケーションパスは、パスの終点となるローカルおよびリモートのアドレスを指定することで、サーバのペア間に複数作成することができます。所定のリモートサーバへの TCP パスを使用する順序を LifeKeeper に設定するには、優先値を使用します。

**重要:** 単一のコミュニケーションパスを使用した場合、互いに通信するクラスタ内のサーバの機能に支障をきたす可能性があります。単一のコミュニケーションパスを使用しているときに、そのコミュニケーションパスで障害が発生した場合、複数のサーバ上で同時に LifeKeeper の階層が使用可能になることがあります。これは、「偽のフェイルオーバ」と呼ばれます。また、TCP コミュニケーションパス上のネットワークトラフィックが大きくなると、偽のフェイルオーバや LifeKeeper の初期化の問題など、予期せぬ動作が生じる可能性があります。

1. 開始するには、次の 4 つの方法があります。
   - サーバアイコンを右クリックし、サーバコンテキストメニュー が表示されたら **[Create Comm Path]** をクリックしてください。
   - グローバルツールバー で、 **[Create Comm Path]** ボタンをクリックしてください。
   - サーバコンテキストツールバー で (表示されている場合)、 **[Create Comm Path]** ボタンををクリックしてください。
   - [Edit] メニュー で、 **[Server]** 、 **[Create Comm Path]** の順に選択してください。

2. **[Create Comm Path]** というタイトルのダイアログが表示されます。表示されるオプションのそれぞれについて、 **[Help]** をクリックすると、選択した項目の説明が表示されます。

3. リストボックスから **[Local Server]** を選択し、 **[Next]** をクリックしてください。

4. リストボックスで 1 つ以上の **[Remote Servers]** を選択してください。リストボックスにリモートサーバが表示されていない (つまり、クラスタにまだ接続されていない) 場合は、 **[Add]** を使用して入力してください。ローカルとリモートの両方のサーバに対するネットワークアドレスが解決可能であることを確認する必要があります (たとえば、DNS で解決するか、 */etc/hosts* ファイルに追加します)。 **[Next]** をクリックしてください。

5. **[Device Type]** に対して **[TCP]** または **[TTY]** を選択して、 **[Next]** をクリックしてください。

6. **[Device Type]** が [TCP] に設定されている場合は、1 つ以上の **[Local IP Addresses]** を選択してください。 **[Device Type]** が [TTY] に設定されている場合は、 **[Local TTY Device]** を選択してください。 **[Next]** をクリックしてください。

7. **[Device Type]** が **[TCP]** に設定されている場合は、 **[Remote IP Address]** を選択してください。

**[Device Type]** が **[TTY]** に設定されている場合は、 **[Remote TTY Device]** を選択してください。 **[Next]** をクリックしてください。

8. **[Device Type]** が **[TCP]** に設定されている場合は、このコミュニケーションパスに対して **[Priority]** を入力または選択してください。 **[Device Type]** が **[TTY]** に設定されいる場合は、このコミュニケーションパスに対して **[Baud Rate]** を入力または選択してください。 **[Next]** をクリックしてください。

9. **[Create]** をクリックしてください。ネットワーク接続が正常に作成されたことを示すメッセージが表示されます。 **[Next]** をクリックしてください。

10. 複数のローカル IP アドレスまたは複数のリモートサーバを選択したときに、 **[Device Type]** が [TCP] に設定されている場合は、手順 6 に戻り、次のコミュニケーションパスの設定を続けます。複数のリモートサーバを選択したときに、 **[Device Type]** が [TTY] に設定されている場合は、手順 5 に戻り、次のコミュニケーションパスの設定を続けます。

11. 終了メッセージが表示されたら、 **[Done]** をクリックしてください。

[Server Properties] ダイアログ を表示するか、またはコマンド **lcdstatus -q** を入力することで、コミュニケーションパスを確認することができます。 **lcdstatus** の使用方法については、LCD マニュアルページを参照してください。 **[ALIVE]** ステータスが表示されます。

さらに、GUI の右ペインのサーバアイコンをチェックしてください。これが、作成済みの 1 つ目のコミュニケーションパスである場合は、1 つのコミュニケーションパスが **[ALIVE]** であるが、冗長コミュニケーションパスがないことを示す黄色のハートビートがサーバアイコンに表示されます。

**[ALIVE]** のコミュニケーションパスが 2 つ以上ある場合は、緑色のハートビートがサーバアイコンに表示されます。

> **重要:** IPv6 アドレスを使用してコミュニケーションパスを作成する場合は、自動設定/ステートレスアドレスではなく、静的に割り当てられたアドレスを使用してください。自動設定/ステートレスアドレスは、時間がたつと変更され、コミュニケーションパスが使用できなくなる可能性があります。

数分たってもコミュニケーションパスが使用可能にならない場合は、ペアのサーバのコンピュータ名が正しいことを確認してください。TTY コミュニケーションパスを使用している場合は、2 つのサーバ間のケーブル接続が正しく、緩んでいないことを確認してください。必要に応じて `portio(1M)` コマンドを使用して、TTY 接続の動作を確認してください。

# 5.4.3.3.3. コミュニケーションパスの削除

1. 開始するには、次の 4 つの方法があります。
   - サーバアイコンを右クリックして、サーバコンテキストメニュー が表示されたら **[Delete Comm Path]** をクリックしてください。
   - グローバルツールバー で、 **[Delete Comm Path]** ボタンをクリックしてください。
   - サーバコンテキストツールバー で (表示された場合)、 **[Delete Comm Path]** ボタンをクリックしてください。
   - [Edit] メニュー で、[Server]、 **[Delete Comm Path]** の順に選択してください。

2. [Delete Comm Path] というタイトルのダイアログが表示されます。表示されるオプションのそれぞれについて、 **[Help]** をクリックすると、選択した項目の説明が表示されます。

3. リストから **[Local Server]** を選択し、 **[Next]** をクリックしてください。このダイアログが表示されるのは、グローバルツールバー の **[Delete Comm Path]** ボタンを使用するか、[Edit] メニュー の [ **Server** ] を選択して、削除を選択した場合のみです。

4. 削除するコミュニケーションパスを選択して、 **[Next]** をクリックしてください。

5. **[Delete Comm Path(s)]** をクリックしてください。出力パネルが有効の場合、ダイアログが閉じ、コミュニケーションパスを削除するコマンドの結果が出力パネル に表示されます。有効でない場合は、ダイアログが表示されたままこれらの結果が表示されます。 すべての結果が表示されたら、 **[Done]** をクリックして終了してください。ネットワーク接続が正常に削除されたことを示すメッセージが表示されます。

6. **[Done]** をクリックして、ダイアログを閉じ、GUI ステータス表示に戻ってください。

# 5.4.3.3.4. サーバのプロパティ - フェイルオーバ

プライマリサーバがローカルリカバリを試行して失敗した場合、または完全に機能停止した場合、ほとんどのサーバ管理者は、LifeKeeper で保護されたリソースをバックアップサーバに復元することが必要になります。これがデフォルトの LifeKeeper の動作になります。ただし管理者によっては、保護されたリソースをリカバリサイトで自動的に稼動するようにしたくない場合もあります。例えば、ディザスタリカバリ状況においてサーバ間のネットワーク接続の信頼性が低くなるような WAN 環境に LifeKeeper がインストールされている場合です。

デフォルトでは、保護されたすべてのリソースに対して自動フェイルオーバが有効になっています。保護されたリソースに対する自動フェイルオーバを無効にしたり、バックアップサーバへの自動フェイルオーバを行わないようにするには、[Server Properties] の **[General]** タブにある **[Failover]** セクションを使用して、以下の通り設定してください。

クラスタ内の各サーバで次のようにします。

1. サーバのプロパティを表示する 場合と同様に、 **[Server Properties]** ダイアログを表示してください。

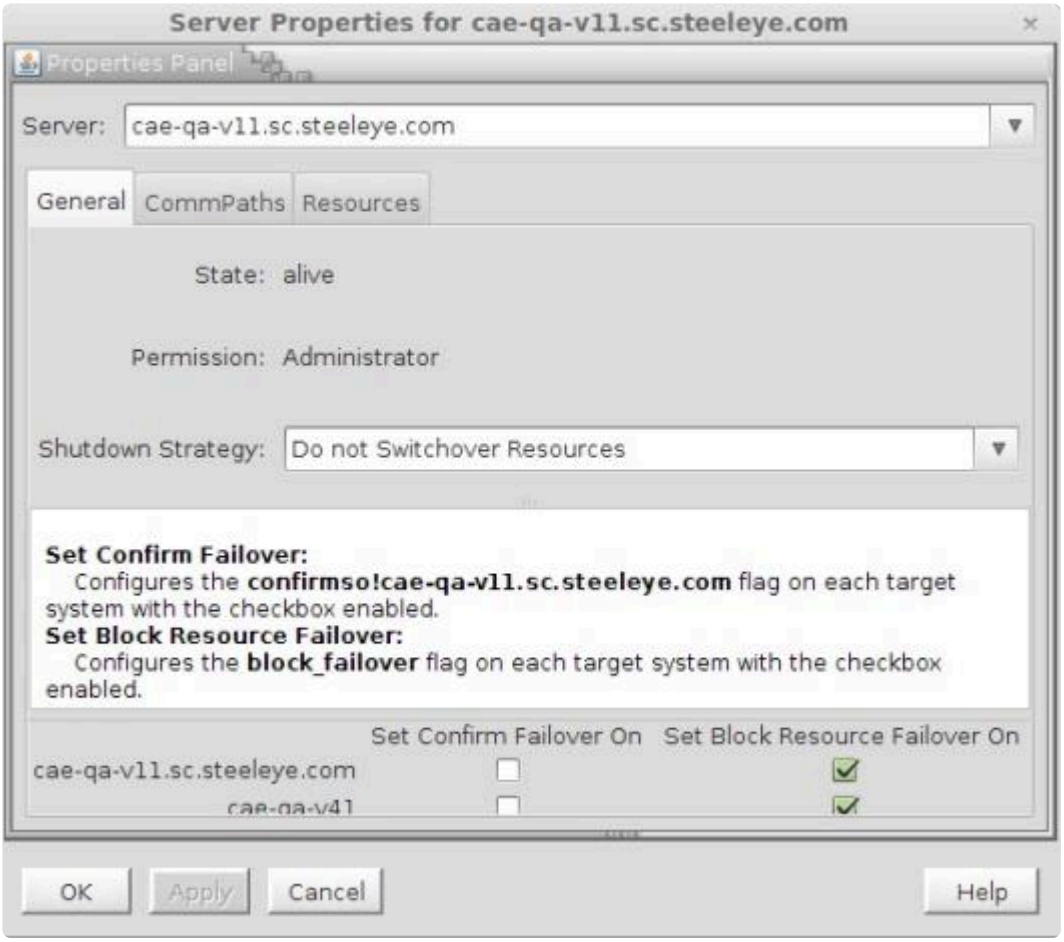
2. **[General]** タブを選択してください。[Server Properties] ダイアログの **[Failover]** セクションで、システムおよびリソースのフェイルオーバ機能を無効にするサーバをチェックしてください。デフォルトでは、LifeKeeper のすべてのフェイルオーバ機能が有効になっています。

   **[Set Confirm Failover On]** 列で、ローカルサーバの完全な機能停止に対応するバックアップサーバとしての資格を喪失させるサーバを選択してください。

   **[Set Block Resource Failover On]** 列で、このローカルサーバ上でリソース階層に障害が発生した場合に対応するバックアップサーバとしての資格を喪失させるサーバを選択してください。最初にシステムのフェイルオーバ機能を無効にしなければ、リソースのフェイルオーバを無効にすることはできません。

   選択内容を適用するには、 **[Apply]** ボタンを押してください。

設定の詳細につきましては、 **[Confirm Failover] と [Block Resource Failover] の設定** のページを参照してください。


Server Properties for cae-qa-v11.sc.steeleye.com
Properties Panel
Server: cae-qa-v11.sc.steeleye.com
General
CommPaths
Resources
State: alive
Permission: Administrator
Shutdown Strategy: Do not Switchover Resources
Set Confirm Failover:
Configures the confirmso!cae-qa-v11.sc.steeleye.com flag on each target system with the checkbox enabled.
Set Block Resource Failover:
Configures the block_failover flag on each target system with the checkbox enabled.
Set Confirm Failover On
Set Block Resource Failover On
cae-qa-v11.sc.steeleye.com
cae-qa-v41
OK
Apply
Cancel
Help

# 5.4.3.3.5. リソース階層の作成

1. リソース階層の作成を開始するには、次の 4 つの方法があります。

> 各種 ARK のライセンスを持たない Quorum ホストでリソース階層を作成することはできません。

- **サーバアイコン** を右クリックして、サーバコンテキストメニュー が表示されたら、 **[Create Resource Hierarchy]** をクリックしてください。
- グローバルツールバー で、 **[Create Resource Hierarchy]** ボタンをクリックしてください。
- サーバコンテキストツールバー で (表示された場合)、 **[Create Resource Hierarchy]** ボタンをクリックしてください。
- [Edit]メニュー で、 **[Server]** を選択して、 **[Create Resource Hierarchy]** をクリックしてください。

2. [Create Resource Wizard] というタイトルのダイアログが表示され、クラスタ内にインストールされているすべての認識されたリカバリキットのリストが示されます。 アプリケーションを保護するためにリソース階層を構築する Recovery Kit を選択して、[Next] をクリックしてください。
3. [Switchback Type] を選択して、[Next] をクリックしてください。
4. [Server] を選択して、[Next] をクリックしてください。注記: サーバコンテキストメニューから開始した場合、クリックしたサーバアイコンから自動的にサーバが決定されるので、この手順はスキップされます。
5. 続いて表示されるダイアログを使用して、作成しているリソース階層の種類に必要なデータを入力してください。

## LifeKeeper アプリケーションリソース階層

LifeKeeper をリカバリキット無しでインストールした場合、デフォルトでは [Select Recovery Kit] リストに、ファイルシステムまたは Generic Application 用のオプションが含まれています。Generic Application のオプションは、関連付けられたリカバリキットがないアプリケーションに使用できます。

Raw I/O Recovery Kit または IP Recovery Kit (これらは両方とも、別個にパッケージ化され、LifeKeeper Core メディアに含まれている Core Recovery Kit です) をインストールした場合、[Select Recovery Kit] リストはこれらの Recovery Kit に対する追加のオプションを提供します。

これらの利用可能なオプションについては、以下のトピックを参照してください。

- ファイルシステムリソース階層の作成
- Generic Application リソース階層の作成

- Raw デバイスリソース階層の作成

IP Recovery Kit については、IP Recovery Kit テクニカルドキュメンテーション を参照してください。

# Recovery Kit のオプション

インストールしたオプションの各リカバリキットは、[Select Recovery Kit] リストにエントリを追加します。たとえば、Oracle、Apache、および NFS の Recovery Kit がリストに表示されます。必要なリソース階層を作成する手順については、各リカバリキットに付属の管理ガイドを参照してください。

> 注記: ファイルシステムまたは論理ボリューム上に構築されるその他のアプリケーションリソース階層を作成する場合、最初にLogical Volume Manager (LVM) Recovery Kit をインストールする必要があります。

# 5.4.3.3.5.1. ファイルシステムリソース階層の作成

このオプションは、SCSIプロトコル（SCSIバスまたはiSCSI経由）またはNetwork Attached Storage（NAS Recovery Kitが必要）を使用して、2台以上のサーバーから直接アクセス可能なストレージ上のファイルシステムを保護するために使用します。SIOS DataKeeperを使用して複製ファイルシステム階層を作成するには、DataKeeperリソース階層を作成するを参照してください。

1. ファイルシステムリソース階層の作成を開始するには、次の 4 つの方法があります。

   • サーバーアイコンを右クリックして、サーバーコンテキストメニュー が表示されたら、 **[Create Resource Hierarchy]** をクリックしてください。

   • グローバルツールバー で 、 **[Create Resource Hierarchy]** ボタンをクリックしてください。

   • サーバーコンテキストツールバー で (表示された場合)、 **[Create Resource Hierarchy]** ボタンをクリックしてください。

   • [Edit] メニュー で、 **[Server]** を選択して、 **[Create Resource Hierarchy]** をクリックしてください。

2. *[Create Resource Wizard]* というタイトルのダイアログが表示され、 ***[Recovery Kit]*** リストが示されます。 *[File System Resource]* を選択して、 ***[Next]*** をクリックしてください。

3. **[Switchback Type]** を選択して、[Next] をクリックしてください。

4. **[Server]** を選択して、 **[Next]** をクリックしてください。注意:サーバーコンテキストメニューから開始した場合、クリックしたサーバアイコンから自動的にサーバーが決定されるので、この手順はスキップされます。

5. *[Create gen/filesys Resource]* ダイアログが表示されます。ファイルシステムリソース階層に対して **[Mount Point]** を選択し、 **[Next]** をクリックしてください。選択したマウントポイントが、クラスタ内の別のサーバと共有されていることを確認するには、各ストレージキットをチェックして、マウントされたデバイスを共有として認識しているかどうかを確認します。ストレージキットがマウントされたデバイスを認識していない場合は、次のエラーダイアログが表示されます。

   < ***file system*** > is not a shared file system

   **[OK]** を選択すると、 *[Create gen/filsys Resource]* ダイアログに戻ります。

   注記:

   • マウントポイントを選択リストに表示するには、そのマウントポイントがマウントされている必要があります。マウントポイントに対するエントリーが `/etc/fstab` ファイルに存在する場合、階層の作成および拡張時にこのエントリーが削除されます。NAS Recovery Kit を使用する前に (特にマウント設定が複雑な場合)、`/etc/fstab` のバックアップを作成することをお勧め

します。/etc/default/LifeKeeper で調整可能な REPLACEFSTAB=true|TRUE を設定することにより、/etc/fstab のエントリーが削除されても元に戻すように指定することができます。

- これらのリソースの多く (SIOS DataKeeper、LVM、Device Mapper Multipath など) は、ファイルシステムリソースを作成するためにクラスタ内の各サーバで LifeKeeper リカバリキットを必要とします。 これらのキットが適切にインストールされていない場合、クラスタ内で共有されるファイルシステムが表示されません。

6. ファイルシステムリソース階層に対するデフォルトの **[Root Tag]** が作成されます。(これは、ステータス表示でこのリソースに使用されるラベルです)。このルートタグを選択するか独自のルートタグを作成して、 **[Next]** をクリックしてください。

7. **[Create Instance]** をクリックしてください。インスタンス作成のステータスを示すメッセージが、ウィンドウに表示されます。

8. **[Next]** をクリックしてください。ファイルシステム階層が正常に作成されたというメッセージが、ウィンドウに表示されます。

9. この時点で、ファイルシステムリソース階層の拡張 に移動するには **[Continue]** をクリックし、GUI に戻るには **[Cancel]** をクリックします。 **[Cancel]** をクリックすると、階層が 1 つのサーバにしか存在しないという警告メッセージが表示され、この時点で保護されなくなります。

# 5.4.3.3.5.2. Generic Application リソース階層の作成

このオプションは、関連付けられたリカバリーキットがないユーザー定義のアプリケーションを保護する場合に使用します。 /opt/LifeKeeper/lkadm/subsys/gen/app/templates/ には、ユーザー向けのスクリプト用のテンプレートが用意されています。 これらのテンプレートを、保護するアプリケーション用にカスタマイズしてテストする前に、別のディレクトリにコピーします。 Generic Application リソース階層の情報を lksupport データで提供するには、/opt/LifeKeeper/bin/lksupport.d のテンプレートを一意の名前にコピーし、それをカスタマイズしてすべての関連ファイルを取得します。 lksupport ユーティリティは、lksupport.d 内のすべての実行可能ファイルを実行します。

> **注意:** ファイルシステム、ディスクパーティション、IP アドレスなどの他のリソースに依存するアプリケーションについては、これらの各リソースを個別に作成し、Create Dependency を使用して適切な依存関係を作成してください。

1. Generic Application リソース階層の作成を開始するには、次の 4 つの方法があります。
   - サーバアイコンを右クリックして、サーバーコンテキストメニュー が表示されたら、 **[Create Resource Hierarchy]** をクリックしてください。
   - グローバルツールバー で、[Create Resource Hierarchy] ボタンをクリックしてください。
   - サーバーコンテキストツールバー で (表示された場合)、 **[Create Resource Hierarchy]** ボタンをクリックしてください。
   - [Edit] メニュー で、 **[Server]** を選択して、 **[Create Resource Hierarchy]** をクリックしてください。

2. [Create Resource Wizard] というタイトルのダイアログが表示され、 **[Recovery Kit]** リストが示されます。 Generic Application を選択して、 **[Next]** をクリックしてください。

3. **[Switchback Type]** を選択して、 **[Next]** をクリックしてください。

4. **[Server]** を選択して、 **[Next]** をクリックしてください。 **注記:** サーバコンテキストメニューから開始した場合、クリックしたサーバアイコンから自動的にサーバが決定されるので、この手順はスキップされます。

5. 次のダイアログで、アプリケーションに対する **[Restore Script]** へのパスを入力し、 **[Next]** をクリックしてください。これは、アプリケーションを起動するコマンドです。テンプレートディレクトリに、テンプレート起動スクリプト restore.template が用意されています。この restore スクリプトが、既に起動されているアプリケーションに影響を与えてはなりません。

6. アプリケーションに対する **[Remove Script]** へのパスを入力し、 **[Next]** をクリックしてください。これは、アプリケーションを停止するコマンドです。テンプレートディレクトリに、テンプレート停止スクリプト remove.template が用意されています。

7. アプリケーションに対する **[quickCheck Script]** へのパスを入力し、 **[Next]** をクリックしてください。これは、アプリケーションを監視するコマンドです。テンプレートディレクトリに、テンプレート監視スクリプト quickCheck.template が用意されています。

8. アプリケーションに対する **[Local Recovery Script]** へのパスを入力し、 **[Next]** をクリックしてください。これは、ローカルサーバ上の障害が発生したアプリケーションの復元を試みるコマンドです。テンプレートディレクトリに、テンプレート回復スクリプト recover.template が用意されています。

9. **[Application Information]** を入力し、 **[Next]** をクリックしてください。これは、起動、停止、回復、監視の各スクリプトで必要になる可能性のあるアプリケーションに関するオプションの情報です。

10. **[Bring Resource In Service]** に対して [Yes] または [No] を選択し、 **[Next]** をクリックしてください。[No] を選択すると、リソース状態が作成後に OSU に設定されます。[Yes] を選択すると、あらかじめ用意された restore スクリプトが実行されます。 ファイルシステム、ディスクパーティション、IP アドレスなどの他のリソースに依存するアプリケーションについては、適切な依存リソースをまだ作成していない場合には、[No] を選択してください。

11. **[Root Tag]** を入力してください。これは、リソースインスタンスに対する一意の名前です。(これは、ステータス表示でこのリソースに対して表示されるラベルです。)

12. **[Create Instance]** をクリックして、作成プロセスを起動してください。インスタンス作成のステータスを示すメッセージが、ウィンドウに表示されます。

13. **[Next]** をクリックしてください。階層が正常に作成されたというメッセージが、ウィンドウに表示されます。

14. この時点で、Generic Application リソース階層の拡張 に移動するには **[Continue]** をクリックし、GUI に戻るには **[Cancel]** をクリックします。[Cancel] をクリックすると、階層が 1 つのサーバにしか存在しないという警告が表示され、この時点で保護されなくなります。

**注記:** リソース階層作成時に指定したrestore、remove、quickCheckなどの各種スクリプトは、LKROOT/subsys/gen/resource/app/以下の各ディレクトリに配置されています。

- **restore - actions/!restore/<** タグ名 **>**
- **remove - actions/!remove/<** タグ名 **>**
- **quickCheck - actions/!quickCheck/<** タグ名 **>**
- **recover - recovery/!recover/<** タグ名 **>**

# 5.4.3.3.5.3. Raw デバイスリソース階層の作成

このオプションは、Raw デバイスリソースを保護する場合に使用します。たとえば、既存のデータベース階層に追加する必要がある Raw デバイスに対して追加のテーブルスペースを作成する場合、このオプションを使用して Raw デバイスリソースを作成します。

**注記:** LifeKeeper では、ディスク論理ユニット (または LUN) レベルのディスクパーティションリソースを、一度に 1 つのクラスタの 1 つのシステムにロックします。

1. Raw デバイスリソース階層の作成を開始するには、次の 4 つの方法があります。

- º サーバアイコンを右クリックして、

サーバコンテキストメニュー が表示されたら、 **[Create Resource Hierarchy]** をクリックしてください。
- º

グローバルツールバー で、 **[Create Resource Hierarchy]** ボタンをクリックしてください。
- º

サーバコンテキストツールバー で (表示された場合)、 **[Create Resource Hierarchy]** ボタンをクリックしてください。
- º

[Edit] メニュー で、 **[Server]** を選択して、 **[Create Resource Hierarchy]** をクリックしてください。

2. [Create Resource Wizard] というタイトルのダイアログが表示され、 **[Recovery Kit]** リストが示されます。Raw デバイスを選択して、 **[Next]** をクリックしてください。

3. **[Switchback Type]** を選択して、 **[Next]** をクリックしてください。

4. **[Server]** を選択して、 **[Next]** をクリックしてください。

   **注記:** サーバコンテキストメニューから開始した場合、クリックしたサーバアイコンから自動的にサーバが決定されるので、この手順はスキップされます。

5. このリソースが配置される共有ストレージデバイスで **[Raw Partition]** を選択して、 **[Next]** をクリックしてください。

6. **[Root Tag]** を入力してください。これは、リソースインスタンスに対する一意の名前です。(これは、ステータス表示でこのリソースに対して表示されるラベルです。)

7. **[Create Instance]** をクリックして、作成プロセスを起動してください。作成中に何が発生したかを示すテキストが、[Creating scsi/raw resource] というタイトルのウィンドウに表示されます。

8. **[Next]** をクリックしてください。階層が正常に作成されたというメッセージが、ウィンドウに表示されます。

9. この時点で、Raw リソース階層の拡張 に移動するには **[Continue]** をクリックし、GUI に戻るには

**[Cancel]** をクリックします。[Cancel] をクリックすると、階層が 1 つのサーバにしか存在しないということを警告するメッセージが表示され、この時点で保護されなくなります。

# 5.4.3.3.5.4. Quick Service Protection (QSP) Recovery Kit

## はじめに

QSP Recovery Kit は、OS のサービスを簡易的に保護する仕組みを提供します。OS の service コマンドまたは systemctl コマンドで起動 (start) ･停止 (stop) ができるサービスであれば、容易にリソースが作成できます。Generic Application を使っても同じことが実現できますが、QSP を使えばコードの開発が不要です。また、依存関係を作成することで、他のリソースで保護しているアプリケーションと連携してサービスを起動･停止させることができます。

ただし、QSP の quickCheck では簡易なチェックしか行っていません（service コマンドの status を利用しています）。実際にそのサービスが提供できているか、プロセスが動作しているかは保証されません。複雑な起動や停止処理、堅牢なチェックが必要な場合は、Generic Application の利用を検討してください。

## 要件

QSPで保護するサービスは、以下の要件を満たしている必要があります。

- OS の service コマンドまたは systemctl コマンドで保護したいサービスの起動 (start) と停止 (stop) ができること。また、コマンド成功時は0を返すこと。
- quickCheck 機能を有効にする場合は、OS の service コマンドまたは systemctl コマンドで保護したいサービスの状態取得 (status) ができること。また、コマンド成功時は 0 を返すこと。
- 保護したいサービス名が 256 文字以内で、英数記号以外の文字が含まれていないこと。

> ✻ 補足 : service コマンドの互換性を利用して、実装上は systemd 環境でも service コマンドを実行する場合があります。

リソース作成時は、プライマリサーバで保護したいサービスを起動しておく必要があります。

なお、専用の Recovery Kit があるサービスやアプリケーションではないサービス等については（以下、これらのサービスを「QSP 保護対象外サービス」と呼ぶ）、QSP 保護対象外となり QSP Recovery Kit では保護できません。

## QSP リソース階層の作成

このオプションは、OS のサービスを保護する場合に使用します。

1. QSPリソース階層の作成を開始するには、次の 4 つの方法があります。

- サーバアイコンを右クリックして、サーバコンテキストメニュー が表示されたら、[Create Resource Hierarchy] をクリックしてください。

- グローバルツールバー で、[Create Resource Hierarchy] ボタンをクリックしてください。
- サーバコンテキストツールバー で (表示された場合)、[Create Resource Hierarchy] ボタンをクリックしてください。
- [Edit] メニュー で、[Server] を選択して、[Create Resource Hierarchy] をクリックしてください。

2. [Create Resource Wizard] というタイトルのダイアログが表示され、[Recovery Kit] リストが示されます。Quick Service Protection を選択して、[Next] をクリックしてください。

3. [Switchback Type] を選択して、[Next] をクリックしてください。

4. [Server] を選択して、[Next] をクリックしてください。

   注記： サーバコンテキストメニューから開始した場合、クリックしたサーバアイコンから自動的にサーバが決定されるので、この手順はスキップされます。

5. 次のダイアログで、[Service Name] にサービスの一覧が示されます。保護したいサービスを選択して [Next] をクリックしてください。

   注記： サービスが起動していない場合は、リストに表示されないことがあります。その場合は、[Cancel] をクリックして処理を中断し、サービス起動後に最初からやり直してください。また、QSP 保護対象外サービスは表示されません。

6. quickCheck 機能を有効にする場合は [enable] を選択、無効にする場合は [disable] を選択し、[Next] をクリックしてください。この設定は後から変更することも可能です。

   注記： service コマンドで状態取得 (status) がサポートされていないサービスでは、QSP の監視機能を使用できませんので必ず “disable” を選択してください。

7. [Resource Tag] を入力してください。これは、リソースインスタンスに対する一意の名前です。（これは、ステータス表示でこのリソースに対して表示されるラベルです。）

8. [Create Instance] をクリックして、作成プロセスを起動してください。インスタンス作成のステータスを示すメッセージがウィンドウに表示されます。

9. [Next] をクリックしてください。続けて拡張を行うダイアログが表示されます。GUI に戻るには [Cancel] をクリックしてください。[Cancel] をクリックすると、階層が１つのサーバにしか存在しないという警告が表示され、この時点では LifeKeeper による保護は行われません。

## QSP リソース階層の拡張

この操作は、リソース階層の拡張 に関するセクションで説明されているように、QSP リソース階層の作成を終了した後に自動的に開始したり、既存の QSP リソースから開始することができます。それが済んだら、次に以下の手順を完了します。

1. LifeKeeper が提供する [Resource Tag] を選択するか、またはターゲットサーバ上のリソース階層に対する独自のタグを入力してください。

2. [Extend] をクリックして、拡張プロセスを起動してください。ダイアログに拡張操作のステータス

が表示され、階層が正常に拡張されたことを示すメッセージが表示されて終了します。同じリソース階層を別のサーバに拡張する場合は、[Next Server] をクリックしてください。その場合は、拡張の操作が繰り返されます。または、[Finish] をクリックして、この操作を完了してください。

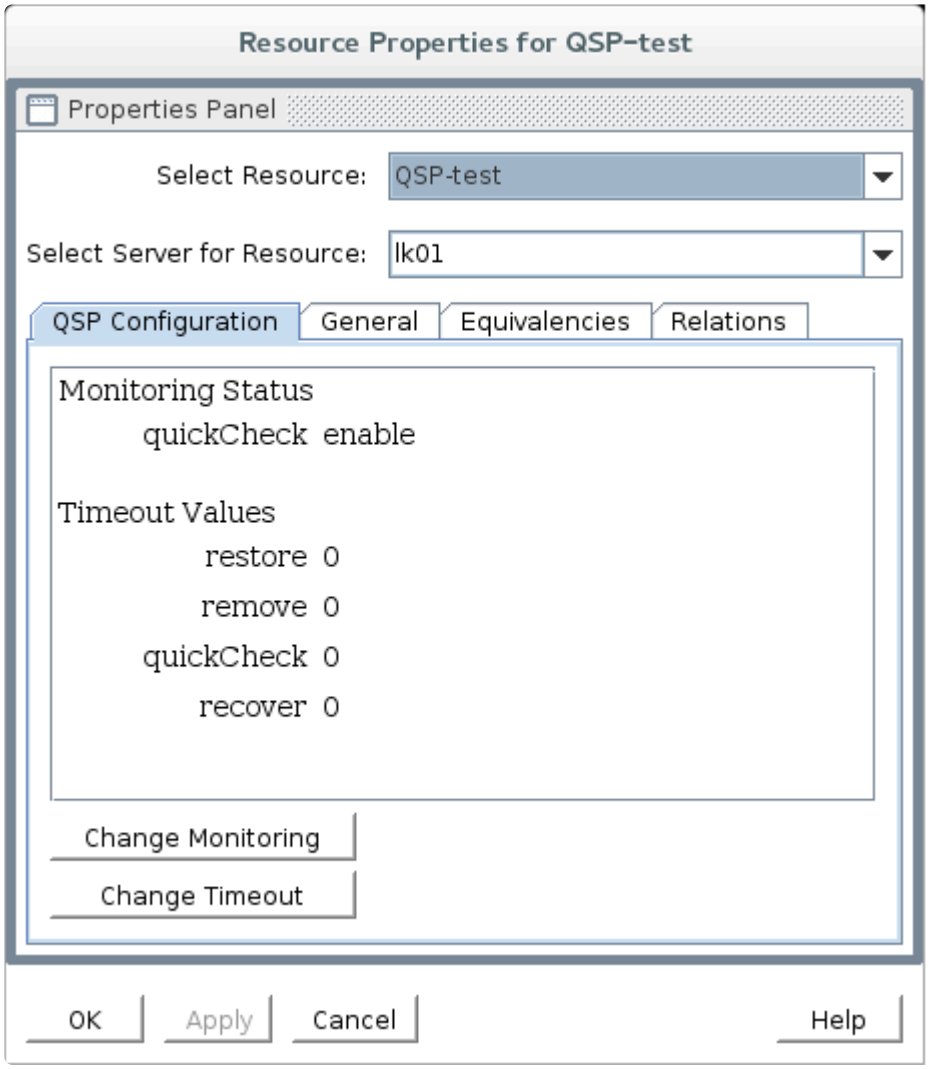
3. 拡張された階層が確認されると、確認情報がダイアログに表示されます。これが終了すると、[Done] ボタンが有効になります。[Done] をクリックして終了してください。

## QSPリソースのコンフィグレーション

QSPリソース固有の設定として以下の項目があります。

| 設定項目 | | デフォルト値 | 意味 |
|---|---|---|---|
| 監視機能 | quickCheck | リソース作成時に指定 | 監視機能の有効･無効の設定。 |
| タイムアウト | restore | 0 | restore のタイムアウト（単位：秒）を指定する。0が指定された場合はタイムアウトしない。 |
| | remove | 0 | remove のタイムアウト（単位：秒）を指定する。0が指定された場合はタイムアウトしない。 |
| | quickCheck | 0 | quickCheck のタイムアウト（単位：秒）を指定する。0が指定された場合はタイムアウトしない。 |
| | recover | 0 | recover のタイムアウト（単位：秒）を指定する。0が指定された場合はタイムアウトしない。 |

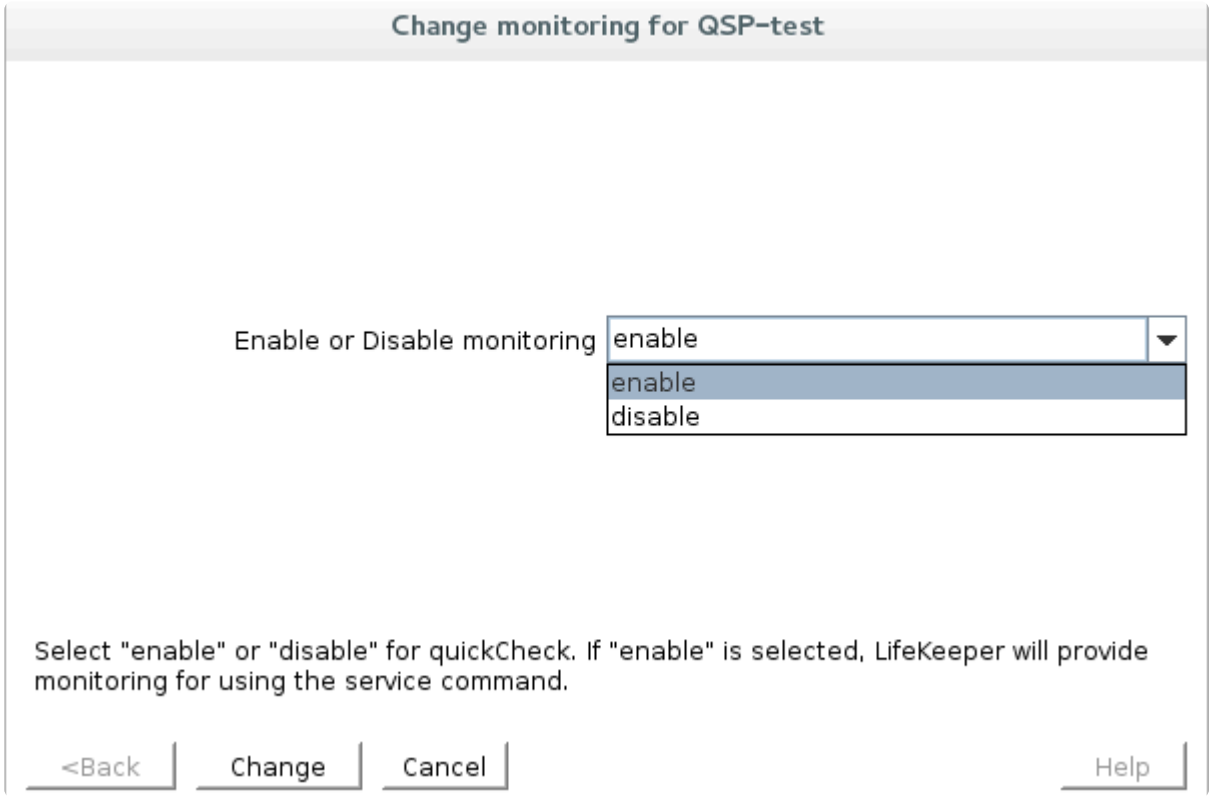
設定値の確認･変更はリソースのプロパティを表示 させ [QSP Configuration] タブから行うことができます。また、設定はノード毎に行う必要があります。 なお、quickCheck 機能を無効にした場合、タイムアウトの quickCheck と recover は表示されません。また変更もできません。




## 監視機能の変更方法

1. リソースプロパティの [QSP Configuration] タブを表示して、[Change quickCheck] をクリックしてください。
2. quickCheck を有効にする場合は [enable] を選択、無効にする場合は [disable] を選択してください。
3. [Change] をクリックすると変更処理が起動して、変更処理のメッセージが表示されます。
4. [Done] をクリックして終了してください。

   注記： 全ノードの設定を変更することはできません。変更した場合は、他のノードでも変更してください。



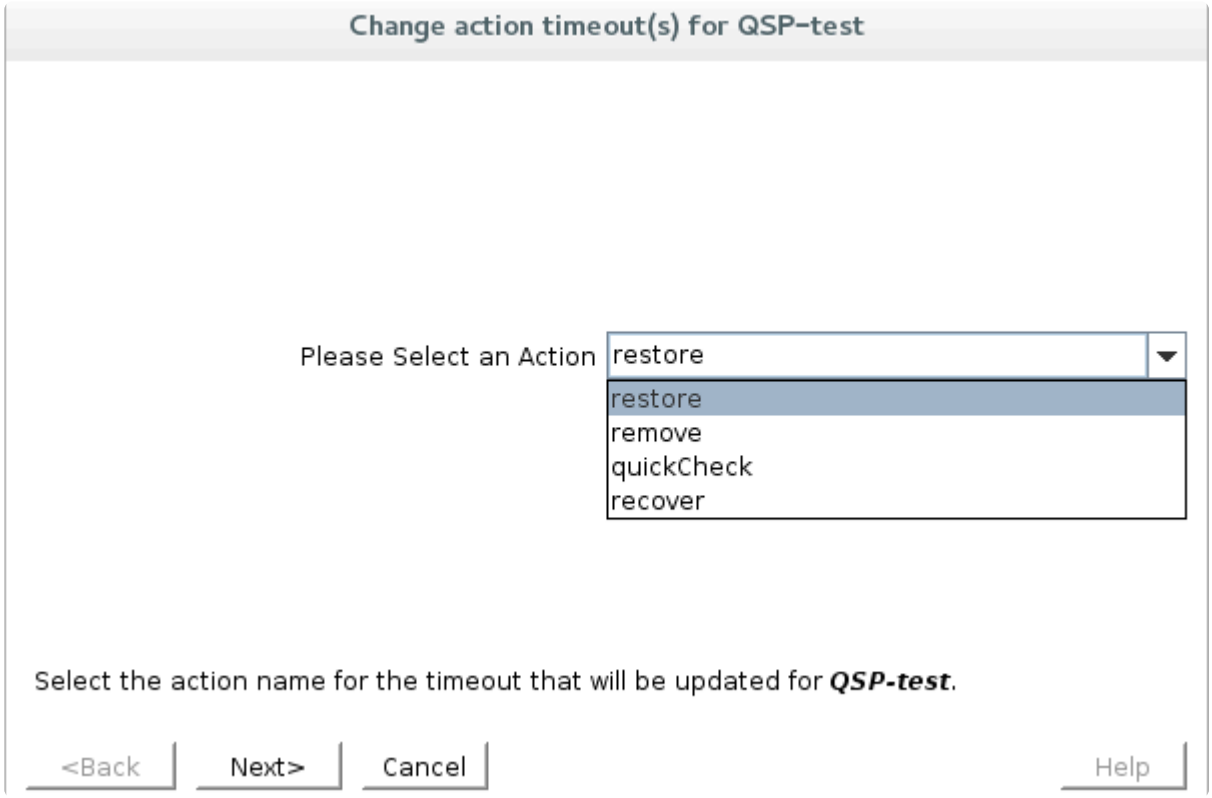

## タイムアウト値の変更方法

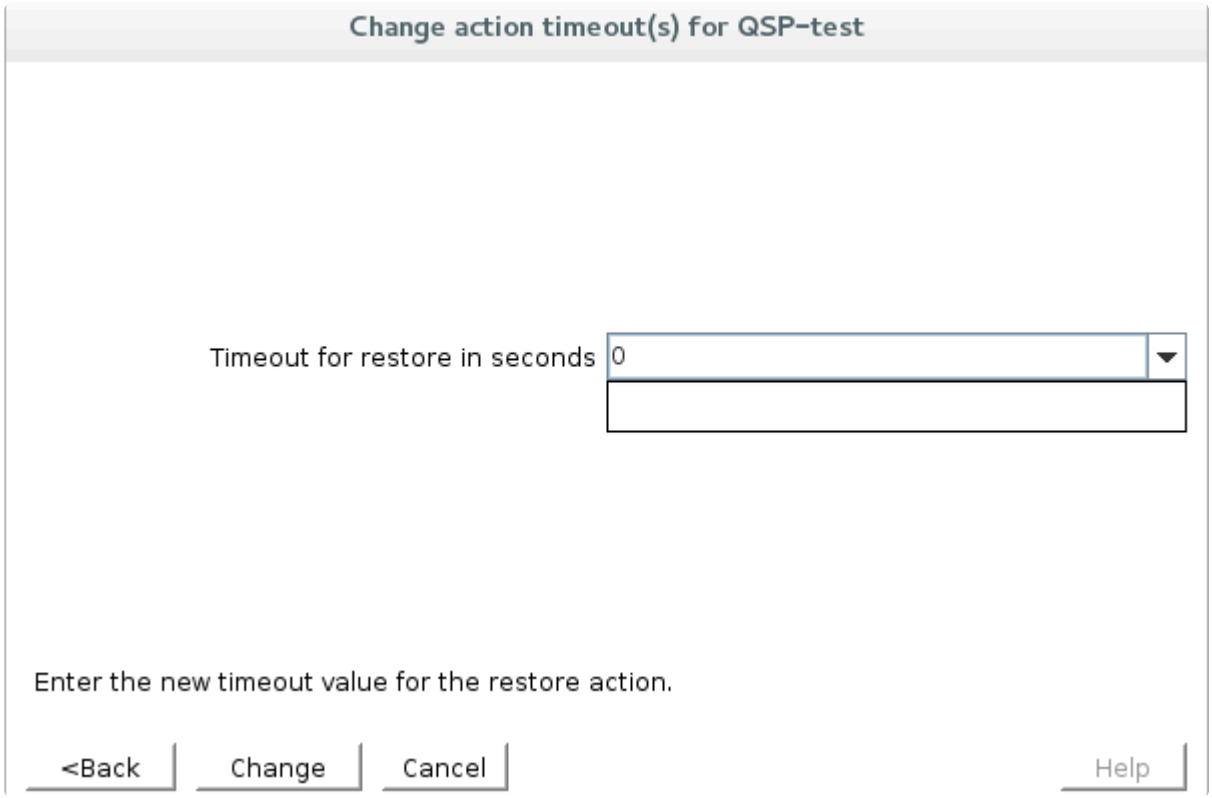
1. リソースプロパティの [QSP Configuration] タブを表示して、[Change Timeout] をクリックしてください。
2. 変更したいタイムアウトのアクションを選択して、[Next] をクリックしてください。

   注記： 監視機能が無効の場合は、[quickCheck] と [recover] は選択肢に表示されません。
3. 秒単位でタイムアウト値を入力してください。

   注記： 10進数で入力してください。数字以外の文字は無効です。
4. [Change] をクリックすると変更処理が起動して、変更処理のメッセージが表示されます。
5. [Done] をクリックして終了してください。

   注記： 全ノードの設定を変更することはできません。変更した場合は、他のノードでも変更してください。


Change action timeout(s) for QSP-test
Please Select an Action
restore
restore
remove
quickCheck
recover
Select the action name for the timeout that will be updated for QSP-test.
<Back
Next>
Cancel
Help


Change action timeout(s) for QSP-test
Timeout for restore in seconds
0
Enter the new timeout value for the restore action.
<Back
Change
Cancel
Help

# 5.4.3.3.6. リソースのプロパティの編集

1. リソースのプロパティを編集するには、リソースのプロパティを表示する 場合と同様に、[Resource Properties] ダイアログを表示してください。

2. 該当のサーバに適切な権限でログインした場合は、次の項目が編集可能になります。

- º スイッチバック
- º リソース設定 (特殊な設定を持つリソースの場合のみ)
- º

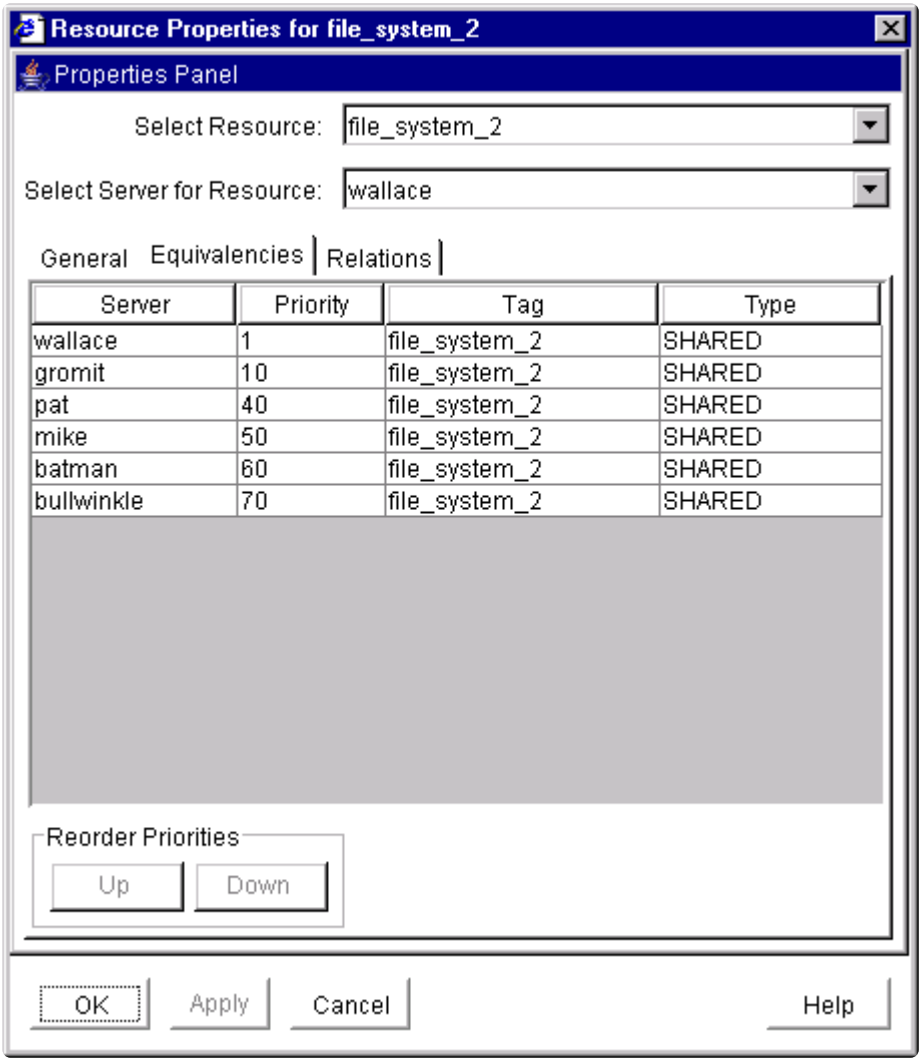
リソースの優先順位

3. 変更が加えられると、 **[Apply]** ボタンが有効になります。このボタンをクリックすると変更が適用されます。ウィンドウは閉じられません。

4. 完了したら、 **[OK]** をクリックし、変更内容を保存してウィンドウを閉じるか、または **[Cancel]** をクリックして、変更内容を適用せずにウィンドウを閉じます。

# 5.4.3.3.7. リソースの優先順位の編集

リソース階層が定義されているサーバの優先順位は、編集または変更することができます。最初に、リソースのプロパティを表示する 場合と同様に、Resource Properties ダイアログを表示してください。下に示すように Resource Properties ダイアログの [Equivalencies] タブに、サーバ上の特定のリソースに対する優先順位が表示されます。




優先順位を変更するには、次の 2 つの方法があります。

- **[Up]** / **[Down]** ボタンを使用してイクイバレンシを移動することにより、優先順位を変更してください。または
- 優先順位の値を直接編集してください。

## [Up] および [Down] ボタンの使用

1. Equivalencies 表で行をクリックして、イクイバレンシを選択してください。選択したイクイバレンシに応じて、**[Up]** または **[Down]** ボタンが有効になります。最も優先順位が高いサーバを選択した場合以外は、**[Up]** ボタンが有効になります。最も優先順位が低いサーバを選択した場合以外は、**[Down]** ボタンが有効になります。

2. **[Up]** または **[Down]** をクリックして、優先順位リストのイクイバレンシを移動してください。

数値の優先順位の列は変更されませんが、イクイバレンシがリスト内で上下に移動します。

## 優先順位の値の編集

1. Equivalencies 表の Priority 列で優先順位の値をクリックすることにより、優先順位を選択してください。優先順位の値の周りにボックスが表示され、値が強調表示されます。

2. 必要な優先順位を入力して、 **Enter** を押してください。

   **注記:** 有効なサーバの優先順位は、1 ～ 999 です。

優先順位を編集した後、Equivalencies 表が再ソートされます。

## 変更の適用

Equivalencies 表で必要な優先順位を設定したら、 **[Apply]** (または **[OK]** ) をクリックして変更を適用します。 **[Apply]** ボタンをクリックすると、加えられた変更内容が適用されます。 **[OK]** ボタンをクリックすると、加えられた変更内容が適用され、ウィンドウが閉じられます。 **[Cancel]** ボタンをクリックすると、 **[Apply]** が直前にクリックされているので、加えられた変更内容を保存せずにウィンドウが閉じられます。

# 5.4.3.3.8. リソース階層の拡張

LifeKeeper の **[Extend Resource Hierarchy]** オプションは、あるサーバから既存の階層をコピーして、別の LifeKeeper サーバ上に同様の階層を作成します。階層が他のサーバに拡張されると、そのリソースに対してカスケーディングフェイルオーバが使用可能になります。既存の階層が現在存在するサーバは、テンプレートサーバと呼ばれます。新たに拡張された階層が配置されるサーバは、ターゲットサーバと呼ばれます。

ターゲットサーバは、拡張された階層をサポートすることができ、他のリモートサーバ上の同等の階層と (アクティブな LifeKeeper コミュニケーションパスを介して) 通信できなければなりません。つまり、既存の階層内のリソースに関連付けられているすべてのリカバリキットが、ターゲットサーバだけでなく、階層が現在存在する他のすべてのサーバに既にインストールされている必要があります。

1. GUI を介してリソース階層を拡張するには、次の 5 つの方法があります。

   • 新しいリソース階層を作成 してください。階層が作成されたことがダイアログに表示されたら、**[Continue]** ボタンをクリックして、Pre-Extend Wizard を介して新しい階層の拡張を開始してください。

   • グローバルまたはサーバ固有のリソースアイコンを右クリックして、リソースコンテキストメニュー を表示し、次に [Extend Resource Hierarchy] をクリックして、Pre-Extend Wizard を介して選択したリソースを拡張してください。

   • グローバルツールバー で、 **[Extend Resource Hierarchy]** ボタンをクリックしてください。[Pre-Extend Wizard] ダイアログが表示されたら、 **[Template Server]** および **[Tag to Extend]** を選択し、それぞれ選択した後に **[Next]** をクリックしてください。

   • リソースコンテキストツールバー で (表示された場合)、 **[Extend Resource Hierarchy]** ボタンをクリックして、Pre-Extend Wizard を表示してください。

   • [Edit] メニュー で、 **[Resource]** を選択して、 **[Extend Resource Hierarchy]** をクリックしてください。[Pre-Extend Wizard] ダイアログが表示されたら、 **[Template Server]** および **[Tag to Extend]** を選択し、それぞれ選択した後に **[Next]** をクリックしてください。

2. デフォルトの **[Target Server]** を選択するか、または選択リストの中から 1 つ入力して、 **[Next]** をクリックしてください。

3. **[Switchback Type]** を選択して、 **[Next]** をクリックしてください。

4. デフォルト値を選択するか、または独自の **[Template Priority]** を入力して、 **[Next]** をクリックしてください。

5. 独自の **[Target Priority]** を選択するか入力して、 **[Next]** をクリックしてください。

6. ダイアログに、次に実行される拡張前のチェックが表示されます。これらのテストが成功した場合、LifeKeeper は、拡張している特定の種類のリソースに必要な手順の実行を開始します。

**[Extend Resource Hierarchy]** オプションの **[Accept Defaults]** ボタンは、LifeKeeper の **[Extend Resource Hierarchy]** のデフォルト値を熟知して、値の入力や確認をしないで素早く LifeKeeper リソース階層を拡張したいユーザ向けです。GUI ダイアログを使用して対話的に段階を追って LifeKeeper リソース階層を拡張する場合は、 **[Next]** ボタンを選択します。

✻ **注記:** マルチルート階層のすべてのルートは、まとめて拡張する必要があります。つまり、単一ルート階層として拡張することはできません。

✻ **注記:** コマンドラインによる手順については、SAP ドキュメントのコマンドラインからの SAP リソースの拡張 を参照してください。

# 5.4.3.3.8.1. ファイルシステムリソース階層の拡張

この操作は、リソース階層の拡張 に関するセクションで説明されているように、ファイルシステムリソース階層の作成 を完了した後に自動的に開始したり、既存のファイルシステムリソースから開始することができます。それが済んだら、次に以下の手順を完了します。これらの手順は、ファイルシステムリソースに固有のものです。

1. *[Extend gen/filesys Resource Hierarchy]* ダイアログボックスが表示されます。ファイルシステム階層に対して **[Mount Point]** を選択し、 **[Next]** をクリックしてください。

2. LifeKeeper が提供する **[Root Tag]** を選択するか、またはターゲットサーバ上のリソース階層に対する独自のタグを入力して、 **[Next]** をクリックしてください。

3. ダイアログに拡張操作のステータスが表示され、階層が正常に拡張されたことを示すメッセージが表示されて終了します。同じリソース階層を別のサーバに拡張する場合は、 **[Next Server]** をクリックしてください。その場合は、拡張の操作が繰り返されます。または、 **[Finish]** をクリックして、この操作を完了してください。

4. 拡張された階層が検証されると、確認情報がダイアログに表示されます。これが終了すると、 **[Done]** ボタンが有効になります。 **[Done]** をクリックして終了してください。

# 5.4.3.3.8.2. Generic Application リソース階層の拡張

この操作は、リソース階層の拡張 に関するセクションで説明されているように、Generic Application リソース階層の作成 を終了した後に自動的に開始したり、既存の Generic Application リソースから開始することができます。それが済んだら、次に以下の手順を完了します。これらの手順は、Generic Application リソースに固有のものです。

1. LifeKeeper が提供する **[Root Tag]** を選択するか、またはターゲットサーバ上のリソース階層に対する独自のタグを入力して、 **[Next]** をクリックしてください。

2. 次に **[Application Information]** (オプション) を入力し、 **[Next]** をクリックしてください。

3. ダイアログに拡張操作のステータスが表示され、階層が正常に拡張されたことを示すメッセージが表示されて終了します。同じリソース階層を別のサーバに拡張する場合は、 **[Next Server]** をクリックしてください。その場合は、拡張の操作が繰り返されます。または、 **[Finish]** をクリックして、この操作を完了してください。

4. 拡張された階層が確認されると、確認情報がダイアログに表示されます。これが終了すると、 **[Done]** ボタンが有効になります。 **[Done]** をクリックして終了してください。

# 5.4.3.3.8.3. Raw デバイスリソース階層の拡張

この操作は、リソース階層の拡張 に関するセクションで説明されているように、Raw デバイスリソース階層の作成 を終了した後に自動的に開始したり、既存の Raw デバイスリソースから開始することができます。それが済んだら、次に以下の手順を完了します。これらの手順は、Raw デバイスリソースに固有のものです。

1. LifeKeeper が提供する **[Root Tag]** を選択するか、またはターゲットサーバ上のリソース階層に対する独自のタグを入力して、 **[Next]** をクリックしてください。

2. ダイアログに拡張操作のステータスが表示され、階層が正常に拡張されたことを示すメッセージが表示されて終了します。同じリソース階層を別のサーバに拡張する場合は、 **[Next Server]** をクリックしてください。その場合は、拡張の操作が繰り返されます。または、 **[Finish]** をクリックして、この操作を完了してください。

3. 拡張された階層が検証されると、確認情報がダイアログに表示されます。これが終了すると、 **[Done]** ボタンが有効になります。 **[Done]** をクリックして終了してください。

# 5.4.3.3.9. 階層の拡張解除

LifeKeeper の **[Unextend Resource Hierarchy]** オプションは、単一サーバから階層全体 (すべてのリソースを含む) を削除します。これは、すべてのサーバから 1 つの階層を削除する **[Delete Resource Hierarchy]** 選択項目とは異なります。

**[Unextend Resource Hierarchy]** を使用する場合、既存の階層を削除するサーバは、ターゲットサーバと呼ばれます。

**[Unextend Resource Hierarchy]** 選択項目は、ターゲットへのアクティブな LifeKeeper コミュニケーションパスを持つ LifeKeeper サーバから使用することができます。

1. 開始するには、次の 5 つの可能な方法があります。

   • 拡張解除したいリソース階層/サーバの組み合わせに対するアイコンを右クリックしてください。リソースコンテキストメニュー が表示されたら、 **[Unextend Resource Hierarchy]** をクリックしてください。

   • 拡張解除したいグローバルリソース階層のアイコンを右クリックしてください。リソースコンテキストメニュー が表示されたら、 **[Unextend Resource Hierarchy]** をクリックしてください。ダイアログが表示されたら、リソース階層の拡張を解除するサーバを **[Target Server]** リストで選択し、 **[Next]** をクリックしてください。

   • グローバルツールバー で、 **[Unextend Resource Hierarchy]** ボタンをクリックしてください。ダイアログが表示されたら、リソース階層の拡張を解除するサーバを **[Target Server]** リストで選択し、 **[Next]** をクリックしてください。次のダイアログで、 **[Hierarchy to Unextend]** リストから拡張解除したいリソース階層を選択し、再度 **[Next]** をクリックしてください。

   • リソースコンテキストツールバー で (表示された場合)、 **[Unextend Resource Hierarchy]** ボタンをクリックしてください。

   • [Edit] メニュー で、 **[Resource]** をポイントして、 **[Unextend Resource Hierarchy]** をクリックしてください。ダイアログが表示されたら、リソース階層の拡張を解除するサーバを **[Target Server]** リストで選択し、 **[Next]** をクリックしてください。次のダイアログで、 **[Hierarchy to Unextend]** リストから拡張解除したいリソース階層を選択し、再度 **[Next]** をクリックしてください。

2. 拡張解除するように指定したサーバおよびリソース階層を確認するメッセージが、ダイアログに表示されます。 **[Unextend]** をクリックしてアクションを実行してください。

3. 出力パネル が有効な場合、ダイアログが閉じて、リソース階層の拡張を解除するコマンドの結果が出力パネルに表示されます。有効でない場合は、ダイアログが表示されたままこれらの結果が表示されます。 すべての結果が表示されたら、 **[Done]** をクリックして終了します。

# 5.4.3.3.10. リソース依存関係の作成

ほとんどの Recovery Kits では、元のリソース階層作成タスク中にそれらの依存関係が作成されますが、特定の条件下では、新規または追加のリソース依存関係を作成したり、既存のリソース依存関係を削除することが必要になる場合があります。一例として、既存の IP 依存関係を別の IP アドレスに変更する場合が挙げられます。リソース階層全体を削除して、新しいリソース階層を作成する代わりに、既存の IP 依存関係を削除して、異なる IP アドレスを持つ新しい依存関係を作成することができます。

1. 開始するには、次の 4 つの可能な方法があります。

   • 親子依存関係を追加したい、サーバの下の親サーバ固有のリソース、または親グローバルリソースに対するアイコンを右クリックしてください。リソースコンテキストメニュー が表示されたら、**[Create Dependency]** をクリックしてください。

   **注記:** 右ペインでサーバ固有のリソースを右クリックした場合、**[Server]** の値はそのサーバになります。左ペインでグローバルリソースを右クリックした場合、**[Server]** の値は、リソースが最も高い優先度を持つサーバになります。

   • **グローバルツールバー** で、**[Create Dependency]** ボタンをクリックしてください。ダイアログが表示されたら、リソース依存関係の作成を開始するサーバを **[Server]** リストで選択し、**[Next]** をクリックしてください。次のダイアログで、**[Parent Resource Tag]** リストから親リソースを選択し、再度 **[Next]** をクリックしてください。

   • リソースコンテキストツールバー で (表示された場合)、**[Create Dependency]** ボタンをクリックしてください。

   • [Edit] メニュー で、**[Resource]** をポイントして、**[Create Dependency]** をクリックしてください。ダイアログが表示されたら、リソース依存関係の作成を開始するサーバを **[Server]** リストで選択し、**[Next]** をクリックしてください。

   • 次のダイアログで、**[Parent Resource Tag]** リストから親リソースを選択し、再度 **[Next]** をクリックしてください。

2. サーバ上の既存の有効なリソースのドロップダウンボックスから、**[Child Resource Tag]** を選択してください。以下の例外を持つサーバ上で利用可能なすべてのリソースが、ダイアログに表示されます。

   • 親リソース、その先祖、およびその子。

   • 親リソースと同じサーバに拡張されていないリソース。

   • 親リソースと同じ相対優先度を持たないリソース。

   • 親リソースが稼働中の場合に、親と同じサーバ上で稼働していないリソース。

   **[Next]** をクリックして、次のダイアログに進んでください。

3. このダイアログで、依存関係の作成に対して適切な親および子のリソースタグが選択されていることを確認できます。 **[Create Dependency]** をクリックして、親を拡張したクラスタ内のすべてのサーバで依存関係を作成してください。

4. 出力パネル が有効な場合、ダイアログが閉じて、依存関係を作成するコマンドの結果が出力パネルに表示されます。有効でない場合は、ダイアログが表示されたままこれらの結果が表示されます。すべての結果が表示されたら、 **[Done]** をクリックして終了します。

# 5.4.3.3.11. リソース依存関係の削除

1. 開始するには、次の 4 つの可能な方法があります。

   • 親子依存関係を削除したい、サーバの下の親サーバ固有のリソース、または親グローバルリソースに対するアイコンを右クリックしてください。リソースコンテキストメニュー が表示されたら、**[Delete Dependency]** をクリックしてください。

   • グローバルツールバー で、**[Delete Dependency]** ボタンをクリックしてください。ダイアログが表示されたら、リソース依存関係の削除を開始するサーバを **[Server]** リストで選択し、**[Next]** をクリックしてください。次のダイアログで、**[Parent Resource Tag]** リストから親リソースを選択し、再度 **[Next]** をクリックしてください。

   • リソースコンテキストツールバー で (表示された場合)、**[Delete Dependency]** ボタンをクリックしてください。

   • [Edit] メニュー で、**[Resource]** をポイントして、**[Delete Dependency]** をクリックしてください。ダイアログが表示されたら、リソース依存関係の削除を開始するサーバを **[Server]** リストで選択し、**[Next]** をクリックしてください。次のダイアログで、**[Parent Resource Tag]** リストから親リソースを選択し、再度 **[Next]** をクリックしてください。

2. ドロップダウンボックスから **[Child Resource Tag]** を選択してください。これは、削除したい依存関係における子のタグ名である必要があります。**[Next]** をクリックして、次のダイアログボックスに進んでください。

3. このダイアログで、依存関係の削除に対して適切な親および子のリソースタグが選択されていることを確認できます。**[Delete Dependency]** をクリックして、クラスタ内のすべてのサーバ上の依存関係を削除してください。

4. 出力パネル が有効な場合、ダイアログが閉じて、依存関係を削除するコマンドの結果が出力パネルに表示されます。有効でない場合は、ダイアログが表示されたままこれらの結果が表示されます。すべての結果が表示されたら、[Done] をクリックして終了します。

# 5.4.3.3.12. すべてのサーバからの階層の削除

1. 開始するには、次の 5 つの可能な方法があります。

- 削除を開始するサーバの下の、削除したい階層にあるリソースのアイコンを右クリックしてください。リソースコンテキストメニュー が表示されたら、 **[Delete Resource Hierarchy]** をクリックしてください。

> **注記:** リソースをすべてのノードで out of service にせず先にリソース階層を削除してしまうといくつかのシステムリソースが使用できなくなる場合があります。このような状況をできるだけ避けるため、リソース階層を削除する前にすべてのノードでリソースを out of service にしておくことを推奨します。

- 削除したい階層にあるグローバルリソースのアイコンを右クリックしてください。リソースコンテキストメニュー が表示されたら、 **[Delete Resource Hierarchy]** をクリックしてください。ダイアログが表示されたら、リソース階層の削除を開始するサーバを **[Server]** リストで選択し、 **[Next]** をクリックしてください。

- グローバルツールバー で、 **[Delete Resource Hierarchy]** ボタンをクリックしてください。ダイアログが表示されたら、リソース階層の削除を開始するサーバを **[Target Server]** リストで選択し、 **[Next]** をクリックしてください。次のダイアログで、 **[Hierarchy to Delete]** リストから削除したい階層内のリソースを選択し、再度 **[Next]** をクリックしてください。

- プロパティーパネル のリソースコンテキストツールバー で (表示された場合)、 **[Delete Resource Hierarchy]** ボタンをクリックしてください。

- メニュー で、 **[Resource]** をポイントして、 **[Delete Resource Hierarchy]** をクリックしてください。ダイアログが表示されたら、リソース階層の削除を開始するサーバを **[Target Server]** リストで選択し、 **[Next]** をクリックしてください。次のダイアログで、 **[Hierarchy to Delete]** リストから削除したい階層内のリソースを選択し、再度 **[Next]** をクリックしてください。

2. 削除するために指定した階層を確認するメッセージが、ダイアログに表示されます。 **[Delete]** をクリックしてアクションを実行してください。

3. 出力パネル が有効な場合、ダイアログが閉じて、階層を削除するコマンドの結果が出力パネルに表示されます。有効でない場合は、ダイアログが表示されたままこれらの結果が表示されます。 すべての結果が表示されたら、 **[Done]** をクリックして終了します。

# 5.4.4. LifeKeeper User Guide

User Guide は、検索可能な総合リソースで、LifeKeeper の GUI で実行できる多くの作業の詳細情報があります。このドキュメンテーションにアクセスするには、User Guide をクリックしてください。

GUI から実行できる作業は 3 つの分野に分類できます。

共通の作業 - これらはどのユーザでも実行できる基本的な作業で、クラスタへの接続、サーバやリソースのプロパティの表示、ログファイルの表示、GUI の設定の変更などがあります。

オペレータの作業 - これらは Operator (オペレータ) の権限を必要とする高度な作業で、リソースを in service および out of service にする操作などがあります。

管理者の作業 - これらは、Administrator (管理者) の権限を必要とする作業です。サーバのプロパティの編集、リソースの作成、コミュニケーションパスの作成や削除などのサーバレベルの作業、およびリソースの編集、拡張、削除などのリソースレベルの作業があります。

以下の表に、それぞれのユーザ権限で使用できるデフォルトの作業を示します。特定のリソースタイプについてその他の作業が使用できることがあります。これらの作業については、関連するリソースキットのドキュメントで説明しています。

| 作業 | 権限 | | |
|---|---|---|---|
| | ゲスト | オペレータ | 管理者 |
| サーバとリソースの表示 | X | X | X |
| サーバへの接続と切断 | X | X | X |
| サーバのプロパティとログの表示 | X | X | X |
| サーバのプロパティの変更 | | | X |
| リソース階層の作成 | | | X |
| コミュニケーションパスの作成と削除 | | | X |
| リソースのプロパティの表示 | X | X | X |
| リソースのプロパティの変更 | | | X |
| リソースのサービス中とサービス休止の切り替え | | X | X |
| リソース階層の拡張と拡張解除 | | | X |
| リソースの依存関係の作成と削除 | | | X |
| リソース階層の削除 | | | X |

# 5.4.4.1. LifeKeeper for Linux の使用

以下のトピックでは、LifeKeeper のグラフィカルユーザインターフェース (GUI)、および LifeKeeper の GUI から実行できる多数の作業について詳しく説明しています。

---

GUI

ステータスの表

プロパティパネル

出力パネル

メッセージバー

GUI の終了

共通の作業

オペレータの作業

高度な作業

メンテナンス作業

テクニカルノート

# 5.4.4.1.1. GUI

GUI のコンポーネントは、LifeKeeper Core のインストールの一部として、すでにインストールされています。

---


- GUI の概要 - 全般
  - LifeKeeper GUI ソフトウェアパッケージ
- メニュー
  - リソースのコンテキストメニュー
  - サーバのコンテキストメニュー
  - [File] メニュー
  - [Edit] メニュー - [Resource]
  - [Edit] メニュー - [Server]
  - [View] メニュー
  - [Help] メニュー
- ツールバー
  - GUI のツールバー
  - リソースのコンテキストツールバー
  - サーバのコンテキストツールバー
- GUI の実行準備
  - LifeKeeper の GUI - 概要
  - LifeKeeper の GUI の設定
  - GUI サーバの開始 / 停止
  - Java のセキュリティポリシー
  - LifeKeeper サーバでの GUI の実行

# 5.4.4.1.1.1. GUI の概要 – 全般

クラスタマシンの必要なグループメンバシップをユーザが持っている限り、GUI を使用することで、任意のマシンから、任意のクラスタ内のサーバとリソースの管理、操作、または監視ができます (詳細については、GUI のユーザの設定 を参照)。GUI のサーバとクライアントのコンポーネントについて説明します。

## GUI サーバ

GUI サーバは、ハイパーテキスト転送プロトコル (HTTP) とリモートメソッド呼び出し (RMI) を使用して GUI クライアントと通信します。デフォルトでは、GUI サーバは LifeKeeper の起動時に初期化されますが、任意に設定することもできます。GUI サーバの開始 / 停止 を参照してください。

## GUI クライアント

GUI クライアントは、任意の LifeKeeper サーバ上のアプリケーション として実行できます。

クライアントには、以下のコンポーネントがあります。

- 左上のステータスの表 には、接続しているサーバとそのリソースの上位のステータスが表示されます。
- 右上のプロパティパネル には、ステータスの表で直前に選択したオブジェクトの詳細情報が表示されます。
- 下部の出力パネル には、コマンドの出力が表示されます。
- ウィンドウの最下部にあるメッセージバー には、処理のステータスメッセージが表示されます。
- コンテキストツールバー (プロパティパネル内) とグローバルツールバー を使用すると、頻繁に使用する作業に即座にアクセスできます。
- コンテキストメニュー (ポップアップ) とグローバルメニュー から、すべての作業にアクセスできます。

## GUI クライアントの終了

[File] メニュー から **[Exit]** を選択すると、すべてのサーバから切断され、クライアントが終了します。

# 5.4.4.1.1.1.1. LifeKeeper GUI ソフトウェアパッケージ

LifeKeeper GUI は、LifeKeeper Core パッケージクラスタにバンドルされている **steeleye-lkGUI** ソフトウェアパッケージに含まれています。 **steeleye-lkGUI** パッケージは、以下の動作を実行します。

- Java アーカイブフォーマットの LifeKeeper GUI クライアントをインストールする。
- LifeKeeper GUI サーバをインストールする。
- LifeKeeper 管理 Web サーバをインストールする。

  **注記:** LifeKeeper 管理 Web サーバは、パブリック Web サーバとは異なるポート 81 を使用するように設定されます。

- ディレクトリ */opt/LifeKeeper/htdoc/* に Java ポリシーファイルをインストールします。このファイルには、LifeKeeper GUI の実行に必要な最小限の権限があります。LifeKeeper GUI アプリケーションは、この場所にある java.policy ファイルを使用して、アクセスを制御します。
- GUI 管理用に LifeKeeper を準備する。

続行する前に、LifeKeeper サーバに LifeKeeper GUI パッケージがインストール済みであることを確認する必要があります。コマンド **`rpm -qi steeleye-lkGUI`** を入力して、このパッケージがインストール済みであるかどうかを確認できます。GUI パッケージがインストール済みである場合、出力にパッケージ名 **`steeleye-lkGUI`** が表示されます。

# 5.4.4.1.1.2. メニュー

## SIOS LifeKeeper for Linux のメニュー

---

リソースのコンテキストメニュー

サーバのコンテキストメニュー

[File] メニュー

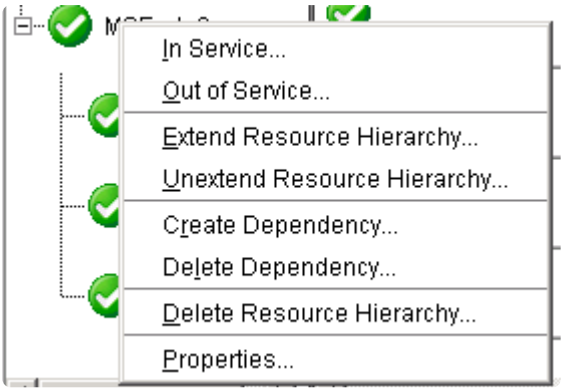
[Edit] メニュー - [Resource]

[Edit] メニュー - [Server]

[View] メニュー

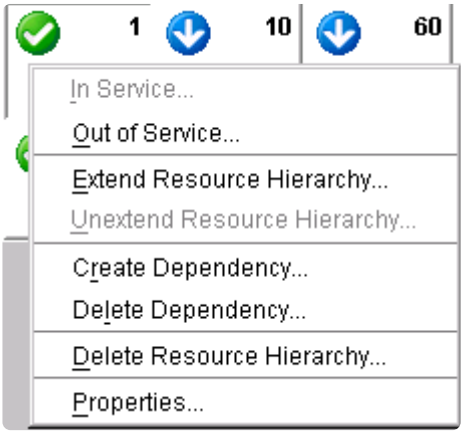
[Help] メニュー

# 5.4.4.1.1.2.1. リソースのコンテキストメニュー




リソースのコンテキストメニューは、ステータスの表 内にあるグローバル (クラスタ全体の) リソース (上図)、またはサーバ固有のリソースインスタンス (下図) を右クリックしたときに表示されます。デフォルトのリソースコンテキストメニューについて、ここで説明しますが、このメニューは特定のリソースタイプについてカスタマイズされていることがあります。この場合、メニューは該当するリソースキットのドキュメンテーションで説明されています。

選択したリソースについて、動作が呼び出されます。特定のサーバ上にあるリソースインスタンスを選択した場合、そのサーバについて動作が呼び出されます。一方、グローバル (クラスタ全体の) リソースを選択した場合は、サーバを選択する必要があります。




In Service -リソース階層を in service にします。

Out of Service -リソース階層を out of service にします。

Extend Resource Hierarchy -フェイルオーバをサポートするために、リソース階層を別のサーバに拡張します。

Unextend Resource Hierarchy -1 台のサーバから拡張リソース階層を削除します。

Create Dependency -2 つのリソース間に親 / 子の関係を作成します。

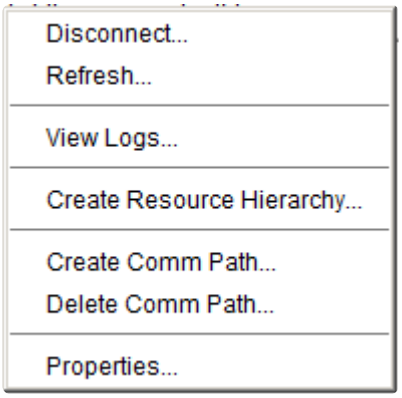
Delete Dependency -2 つのリソース間にある親 / 子の関係を削除します。

Delete Resource Hierarchy -リソース階層を LifeKeeper クラスタ内のすべてのサーバから削除します。

Properties -[Resource Properties] ダイアログ を表示します。

# 5.4.4.1.1.2.2. サーバのコンテキストメニュー

サーバのコンテキストメニューは、ステータスの表 内にあるサーバアイコンを右クリックしたときに表示されます。このメニューは [Edit] メニューの [Server] サブメニューと同じですが、動作は常に、最初に選択したサーバ上で呼び出される点が異なります。




Disconnect -クラスタから切断します。

Refresh -GUI を最新情報に更新します。

View Logs -接続しているサーバについて、LifeKeeper のログメッセージを表示します。

Create Resource Hierarchy -リソース階層を作成します。

Create Comm Path -サーバ間にコミュニケーションパスを作成します。

Delete Comm Path -サーバからコミュニケーションパスを削除します。

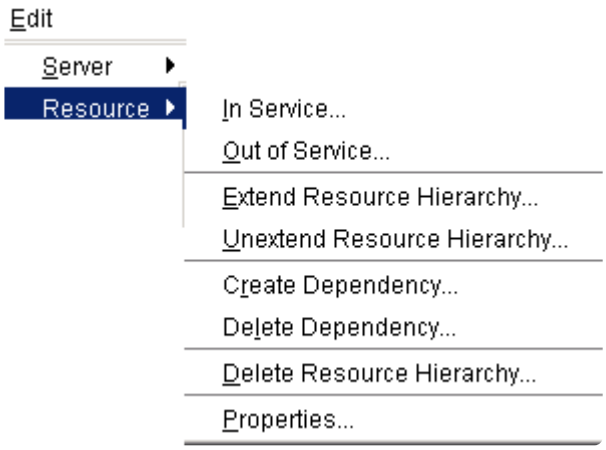
Properties -[Server Properties] ダイアログ を表示します。

# 5.4.4.1.1.2.3. [File] メニュー

**Connect** - LifeKeeper クラスタに接続します。LifeKeeper クラスタ内の各サーバに接続するには、そのサーバでログイン認証が必要です。

**Exit** - すべてのサーバから切断し、GUI のウィンドウを閉じます。

# 5.4.4.1.1.2.4. [Edit] メニュー - [Resource]




In Service - リソース階層を in service にします。

Out of Service - リソース階層を out of service にします。

Extend Resource Hierarchy - フェイルオーバをサポートするために、リソース階層を別のサーバに拡張します。

Unextend Resource Hierarchy - 1 台のサーバから拡張リソース階層を除去します。

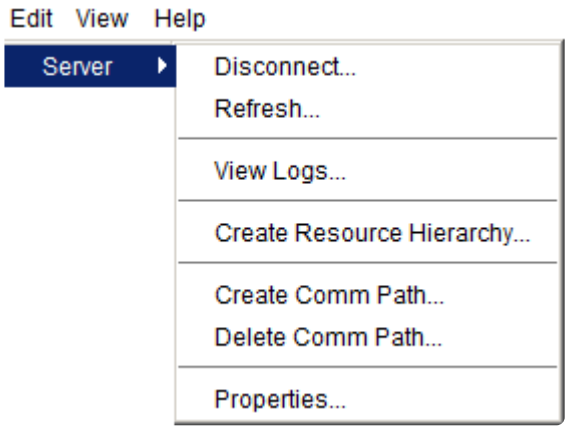
Create Dependency - 2 つのリソース間に親 / 子の関係を作成します。

Delete Dependency - 2 つのリソース間にある親 / 子の関係を削除します。

Delete Resource Hierarchy - リソース階層を LifeKeeper クラスタ内のすべてのサーバから削除します。

Properties - [Resource Properties] ダイアログ を表示します。

# 5.4.4.1.1.2.5. [Edit] メニュー - [Server]




Disconnect - クラスタから切断します。

Refresh - GUI を最新情報に更新します。

View Logs - 接続しているサーバについて、LifeKeeper のログメッセージを表示します。

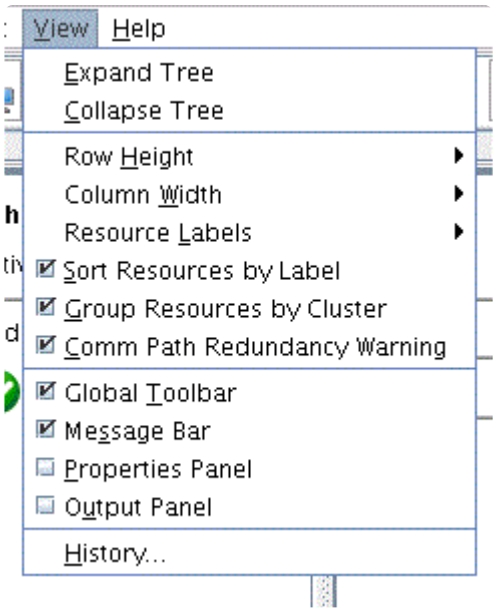
Create Resource Hierarchy - リソース階層を作成します。

Create Comm Path - サーバ間にコミュニケーションパスを作成します。

Delete Comm Path - サーバからコミュニケーションパスを削除します。

Properties -[Server Properties] ダイアログ を表示します。

# 5.4.4.1.1.2.6. [View] メニュー




Expand Tree -リソース階層ツリー全体を展開します。

Collapse Tree -リソース階層ツリー全体を折り畳みます。

**Row Height** -リソース階層ツリーおよびリソーステーブルの行の高さを修正します。表示されたリソースの数により大、中、小を選択してください。

**Column Width** -リソース階層ツリーおよびリソーステーブルの列幅を修正します。表示されたリソースにより 自動 、大、中、小を選択してください。

Resource Labels -このオプショングループを使用すると、リソース階層ツリー内のリソースを、タグ名別と ID 別のいずれで表示するかを指定できます。

**Sort Resources by Label** -リソースラベルのみでリソースを分類します。

**Group Resources by Cluster** -クラスタサーバおよびリソースラベルによって分類します。同じクラスタ内に属するリソースが同じグループになります。

**Comm Path Redundancy Warning** -サーバステータスグラフィックでコミュニケーションパスの状態についての説明を明記します。

- 選択した場合は、一組のサーバ間のコミュニケーションパスが冗長化されていない場合、サーバ警告が表示されます。
- 選択しない場合は、一組のサーバ間のコミュニケーションパスが冗長化されていない場合は無視しますが、コミュニケーションパスが切れた場合には、サーバ警告が表示されます。

Global Toolbar -チェックボックスがオンの場合、このコンポーネントを表示します。

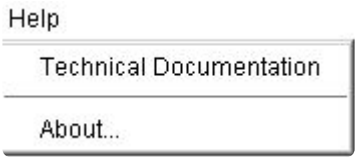
Message Bar -チェックボックスがオンの場合、このコンポーネントを表示します。

Properties Panel -チェックボックスがオンの場合、このコンポーネントを表示します。

Output Panel -チェックボックスがオンの場合、このコンポーネントを表示します。

History -メッセージバーに表示された最新メッセージを、LifeKeeper の GUI の [Message History] ダイアログボックスに表示します (最大 1000 行)。

# 5.4.4.1.1.2.7. [Help] メニュー




**Technical Documentation -** SIOS Technology Corp. のテクニカルドキュメンテーションの開始ページを表示します。

**About...** - LifeKeeper GUI のバージョン情報を表示します。

# 5.4.4.1.1.3. ツールバー

## SIOS LifeKeeper for Linux のツールバー

---

GUI のツールバー

リソースのコンテキストツールバー

サーバのコンテキストツールバー

# 5.4.4.1.1.3.1. GUI のツールバー

このツールバーは、プロパティパネル に表示されるデフォルトのサーバ とリソース のコンテキストツールバーを組み合わせたものですが、このツールバーから動作を実行するときには、サーバとリソースを選択する必要があります。

| | |
|---|---|
| | Connect -LifeKeeper クラスタに接続します。 |
| | Disconnect -LifeKeeper クラスタから切断します。 |
| | Refresh -GUI を最新情報に更新します。 |
| | View Logs -接続しているサーバについて、LifeKeeper のログメッセージを表示します。 |
| | Create Resource Hierarchy -リソース階層を作成します。 |
| | Delete Resource Hierarchy -リソース階層を LifeKeeper クラスタ内のすべてのサーバから削除します。 |
| | Create Comm Path -サーバ間にコミュニケーションパスを作成します。 |
| | Delete Comm Path -サーバからコミュニケーションパスを削除します。 |
| | In Service -リソース階層を in service にします。 |
| | Out of Service -リソース階層をサービス停止にします。 |
| | Extend Resource Hierarchy -フェイルオーバをサポートするために、リソース階層を別のサーバに拡張します。 |
| | Unextend Resource Hierarchy -1 台のサーバから拡張リソース階層を削除します。 |

| | |
|---|---|
| | Create Dependency -2 つのリソース間に親 / 子の関係を作成します。 |
| | Delete Dependency -2 つのリソース間にある親 / 子の関係を削除します。 |

# 5.4.4.1.1.3.2. リソースのコンテキストツールバー

ステータスの表 からサーバ固有のリソースインスタンスを選択すると、プロパティパネル にリソースのコンテキストツールバーが表示されます。

選択したサーバとリソースについて、動作が呼び出されます。灰色表示のリソースについて、動作を選択することはできません。

| | |
|---|---|
| | In Service -リソース階層を in service にします。 |
| | Out of Service -リソース階層を out of service にします。 |
| | Extend Resource Hierarchy -フェイルオーバをサポートするために、リソース階層を別のサーバに拡張します。 |
| | Unextend Resource Hierarchy -1 台のサーバから拡張リソース階層を削除します。 |
| | Add Dependency - 2 つのリソース間に親 / 子の関係を作成します。 |
| | Remove Dependency - 2 つのリソース間にある親 / 子の関係を削除します。 |
| | Delete Resource Hierarchy -リソース階層をすべてのサーバから削除します。 |

# 5.4.4.1.1.3.3. サーバのコンテキストツールバー

ステータスの表 からサーバを選択すると、プロパティパネル にサーバのコンテキストツールバーが表示されます。選択したサーバについて、動作が呼び出されます。

| | |
|---|---|
| | Disconnect -LifeKeeper クラスタから切断します。 |
| | Refresh -GUI を最新情報に更新します。 |
| | View Logs -接続しているサーバについて、LifeKeeper のログメッセージを表示します。 |
| | Create Resource Hierarchy -リソース階層を作成します。 |
| | Delete Resource Hierarchy -リソース階層を LifeKeeper クラスタ内のすべてのサーバから削除します。 |
| | Create Comm Path -サーバ間にコミュニケーションパスを作成します。 |
| | Delete Comm Path -サーバからコミュニケーションパスを削除します。 |

# 5.4.4.1.1.4. GUI の実行の準備

LifeKeeper の GUI - 概要

LifeKeeper の GUI の設定

GUI サーバの開始 / 停止

Java のセキュリティポリシー

LifeKeeper サーバでの GUI の実行

# 5.4.4.1.1.4.1. LifeKeeper の GUI の設定

## GUI 管理用の LifeKeeper サーバーの設定

各 LifeKeeper サーバーについて、以下の手順を実行してください。各手順には、詳細手順の参照先またはリンクがあります。

1. 各サーバーに、Java 実行時環境 (JRE) または Java ソフトウェア開発キット (JDK) をインストールする必要があります。必要な Java のバージョンについては、SPS for Linux リリースノート を参照してください。

2. 各サーバーで、LifeKeeper GUI サーバーを開始してください (GUI サーバーの開始/停止 を参照)。
   **注記:** GUI サーバーが後続の初期インストールを開始した後、LifeKeeper の開始と停止は、GUI サーバーを含む LifeKeeper のすべてのデーモンプロセスの開始と停止を行います。

3. root 以外のユーザに GUI の使用を許可するように計画している場合は、GUI ユーザの設定 が必要です。

## GUI の実行

LifeKeeper の GUI は、クラスター内の LifeKeeper サーバーで実行できます。

LifeKeeper クラスター内のサーバーで GUI の設定と実行を行う方法については、LifeKeeper サーバーでの GUI の実行 を参照してください。

## GUI の設定

| 項目 | 説明 |
|---|---|
| **GUI** のクライアントとサーバーの通信 | LifeKeeper GUI のクライアントとサーバーは、通信に Java のリモートメソッド呼び出し (RMI) を使用します。RMI が正しく動作するためには、クライアントとサーバーは解決可能なホスト名または IP アドレスを使用する必要があります。DNS が実装されていない場合 (または、他の名前のルックアップメカニズムを使用して名前が解決できない場合) は、クライアントとサーバーのそれぞれについて、*/etc/hosts* ファイルを編集し、他のすべての LifeKeeper サーバーの名前とアドレスを含めてください。 |
| **GUI** サーバーの **Java** プラットフォーム | LifeKeeper GUI サーバーには、Java 実行環境 (JRE) - Java 仮想マシン、Java プラットフォームのコアクラス、およびサポートするファイル - をインストールする必要があります。LifeKeeper のGUI は、OpenJDK をサポートしています。LifeKeeper インストール時に実行する setup スクリプトが OS 同梱の OpenJDK をインストールします。ただし、以下の環境では Linux ディストリビューターがサポートしている OpenJDK を提供していないため、LifeKeeper のインストールイメージに同梱している OpenJDK をインス |

| | |
|---|---|
| | トールします。サポートしている OpenJDK のバージョンは、リリースノートを参照してください。<br><br>• RedHat Enterprise Linux/CentOS/Oracle Linux 7.0<br><br>**注記:** LifeKeeper インストール時に、GUI が使用する JRE のパスを LifeKeeper のデフォルトファイル `/etc/default/LifeKeeper` の PATH に設定します。使用する JRE を変更したい場合はこの PATH を編集してください。このファイルの編集時に LifeKeeper が実行中である場合は、変更内容を認識させるために LifeKeeper GUI サーバーと GUI クライアントを停止し、再起動する必要があります。<br><br>**注記:** 以下の LifeKeeper と OS のバージョンの組み合わせでは、LifeKeeper と OS 同梱の OpenJDK の間でメモリ管理の不整合があります。それを回避するため、LifeKeeper のインストールイメージに同梱している OpenJDK を `/opt/LifeKeeper/lib64/java` 以下にインストールして使用しています。メモリ管理の不整合は既に修正済みですが、LifeKeeper をアップグレードした場合は `/opt/LifeKeeper/lib64/java` にインストールされた JRE をそのまま使用し続けます。<br>OS 同梱の JRE を使用したい場合は、JRE をインストールして、上記の手順に従って /etc/default/LifeKeeper の PATH を変更してください。`/opt/LifeKeeper/lib64/java` は不要ですので削除しても問題ありません。<br><br>• LifeKeeper<br>◦ v9.4.1 ～ v9.5.1<br>• OS<br>◦ RedHat Enterprise Linux/CentOS/Oracle Linux 8.0 ～ 8.2<br>◦ SUSE Linux Enterprise Server 15.0 ～ SP2 |
| **Java Runtime Environment** のアンインストール | • LifeKeeperのインストールイメージに同梱しているOpenJDKパッケージをインストールした環境<br>LifeKeeper のアンインストール時にOpenJDKパッケージも同時にアンインストールされます。<br>• Linuxディストリビューターが提供しているOpenJDKをインストールした環境<br>LifeKeeper のアンインストール時にOpenJDKパッケージはアンインストールされません。必要であれば手動でアンインストールを行ってください。 |
| **Java** リモートオブジェクトレジストリーのサーバーポート | LifeKeeper GUI サーバーは、各 LifeKeeper サーバー上の Java リモートオブジェクトレジストリー用にポート 82 を使用します。これにより、サーバーは、典型的なファイアーウォールの後にあるクライアントからの RMI 呼び出しをサポートできます。 |
| **LifeKeeper** の管理 **Web** サーバー | LifeKeeper GUI サーバーには、クライアントのブラウザーの通信用に管理 Web サーバーが必要です。現在、LifeKeeper GUI サーバーは、管理 Web サーバーとして lighttpd Web サーバーのプライベートコピーを使用しています。この Web サーバーは、steeleye-lighttpd パッケージによりインストールと設定が実行され、他の Web サーバーとの競合を避けるためにポート 81 を使用します。 |
| **GUI** クライア | LifeKeeper GUI クライアントには、LifeKeeper クラスター内のすべてのホストへのネット |

| | |
|---|---|
| ントのネットワークアクセス | ワークアクセスが必要です。LifeKeeper GUI クライアントをブラウザー内で実行する場合、アプレットへのネットワークアクセスを可能にするためにセキュリティーレベルを低下させる必要があります。セキュリティーを低い値に設定した状態で、他のサイトを閲覧しないように注意してください (つまり、イントラネットまたは信頼できるサイトについてのみセキュリティー設定を変更する)。 |

## GUI の制限

| 項目 | 説明 |
|---|---|
| **GUI の相互運用性の制限** | LifeKeeper for Linux クライアントは、Linux サーバー上の LifeKeeper の管理にのみ使用できます。LifeKeeper for Linux の GUI と LifeKeeper for Windows は、同時には使用できません。 |

# 5.4.4.1.1.4.2. GUI サーバの開始 / 停止

## LifeKeeper GUI サーバを開始するには

LifeKeeper GUI サーバが動作していない場合は、 *root* として以下のコマンドを入力してください。

```
/opt/LifeKeeper/bin/lkGUIserver start
```

このコマンドは、管理しているサーバで LifeKeeper GUI サーバのデーモンプロセスが現在動作していない場合、それらのデーモンプロセスをすべて開始します。以下のようなメッセージが表示されます。

```
# Installing GUI Log
# LK GUI Server Startup at:
# Mon May 8 14:14:46 EDT 2006
# LifeKeeper GUI Server Startup completed at:
# Mon May 8 14:14:46 EDT 2006
```

LifeKeeper GUI サーバが開始した後、以降の LifeKeeper の開始操作はすべて、LifeKeeper GUI サーバのプロセスを自動的に開始します。

### トラブルシューティング

LifeKeeper GUI は、各サーバのポート 81 を管理 Web サーバ用に、ポート 82 を Java リモートオブジェクトレジストリに使用します。他のアプリケーションがそれらのポートを使用している場合、LifeKeeper GUI は正しく機能しません。これらの値は、ファイル ***/etc/default/LifeKeeper*** の以下のエントリを編集することにより変更できます。

```
GUI_WEB_PORT=81 GUI_RMI_PORT=82
```

**注記：**これらのポートの値は、起動時に GUI サーバで初期化されます。ポートの値を変更した場合、steeleye-lighttpdプロセスを停止し、再起動する必要があります。これらの値は、接続するすべてのクラスタ全体で同一である必要があります。

## LifeKeeper GUI サーバを停止するには

LifeKeeper GUI サーバが動作している場合は、 *root* として以下のコマンドを入力してください。

```
/opt/LifeKeeper/bin/lkGUIserver stop
```

このコマンドは、管理しているサーバで LifeKeeper GUI サーバのデーモンプロセスが現在動作している場合、それらのデーモンプロセスをすべて停止します。以下のメッセージが表示されます。

```
# LifeKeeper GUI Server Shutdown at:
# Fri May 19 15:37:27 EDT 2006
# LifeKeeper GUI Server Shutdown Completed at:
```

```
# Fri May 19 15:37:28 EDT 2006
```

## LifeKeeper GUI サーバのプロセス

LifeKeeper GUI サーバが動作していることを確認するには、以下のコマンドを入力してください。

```
ps -ef | grep runGuiSer
```

以下のような出力が表示されます。

```
root 2805 1 0 08:24 ? 00:00:00 sh/opt/LifeKeeper/bin/runGuiSer
```

現在動作している他の GUI サーバのデーモンプロセスのリストを表示するには、以下のコマンドを入力してください。

```
ps -ef | grep S_LK
```

以下のような出力が表示されます。

```
root 30228 30145 0 11:20 ? 00:00:00 java -Xint -Xss3M
-DS_LK=true -Djava.rmi.server.hostname=thor48 ...
```

# 5.4.4.1.1.4.3. Java のセキュリティポリシー

LifeKeeper の GUI は、ポリシーベースのアクセス制御を使用します。GUI クライアントのロード時に、現在有効なセキュリティポリシーに基づいて権限がGUI クライアントに割り当てられます。ポリシーはさまざまな署名者 / 場所からのコードに提供される権限を指定し、外部から設定可能なポリシーファイルから初期化されます。

デフォルトでは、システム全体のポリシーファイルとオプションのユーザポリシーファイルが 1 つずつあります。システム全体でコードに権限を付与するシステムポリシーファイルが先にロードされ、次にユーザポリシーファイルが追加されます。LifeKeeper GUI がアプリケーションとして起動される場合は、これらのポリシーファイルに加えて、LifeKeeper GUI のポリシーファイルもロードされることがあります。

## ポリシーファイルの場所

デフォルトでは、システムポリシーファイルは以下の場所にあります。

*<JAVA.HOME>/lib/security/java.policy (Linux)*

*<JAVA.HOME>\lib\security\java.policy (Windows)*

**注記:** JAVA.HOME は、システムのプロパティ「JAVA.HOME」の値を指し、JRE または JDK がインストールされたディレクトリの場所を指定します。

ユーザポリシーファイルは「.」の文字で始まり、デフォルトでは以下の場所にあります。

*<USER.HOME>\.java.policy*

**注記:** USER.HOME は、システムのプロパティ「user.home」の値を指し、ユーザのホームディレクトリを指定します。例えば、Windows NT ワークステーション上にあるユーザ Paul のホームディレクトリは、「paul.000」です。

Windows システムの場合、user.home のプロパティ値のデフォルト値は以下のとおりです。

*C:\WINNT\Profiles\<USER> (マルチユーザ **Windows NT** システム)*

*C:\WINDOWS\Profiles\<USER> (マルチユーザ **Windows 95/98** システム)*

*C:\WINDOWS (シングル – ユーザ **Windows 95/98** システム)*

デフォルトでは、LifeKeeper GUI のポリシーファイルは以下の場所にあります。

*/opt/LifeKeeper/htdoc/java.policy **(Linux)***

## ポリシーファイルの作成と管理

デフォルトでは、LifeKeeper GUI がアプリケーションとして起動される場合に LifeKeeper GUI のポリシ

ーファイルが使用されます。LifeKeeper GUI をアプレットとして実行する場合、ホームディレクトリにユーザポリシーファイルを作成する必要があります (存在しない場合)。ユーザポリシーファイルは、LifeKeeper GUI を実行するために必要な最低限の権限を指定する必要があります。このトピックの「ポリシーファイルの例」セクションで後述します。

ポリシーファイルの作成と管理は、単純なテキストエディタ、または Java 実行時環境 (JRE) や Java 開発キット (JDK) に含まれるグラフィカルな **Policy Tool** ユーティリティから行うことができます。Policy Tool を使用すると、入力が簡略化され、ポリシーファイルに必要な構文の知識が不要になります。Policy Tool の使用方法の詳細については、http://docs.oracle.com/javase/8/docs/technotes/tools/ にある Policy Tool のドキュメンテーションを参照してください。

LifeKeeper GUI を実行するために必要な最低限の権限を持つ **ユーザポリシーファイルを作成する最も簡単な方法** は、***/opt/LifeKeeper/htdoc/java.policy*** にある LifeKeeper GUI のポリシーファイルをホームディレクトリにコピーし、ファイル名を *.java.policy* に変更することです (ファイル名の前にあるドットは必須)。Windows システムでは、ファイル *http://<server name>:81/java.policy* (<server name> は LifeKeeper サーバのホスト名) を開いてホームディレクトリに *.java.policy* の名前を付けて保存することで、LifeKeeper GUI のポリシーファイルをコピーできます。ユーザポリシーファイルの正しい場所を特定する必要がある場合は、Java のコントロールパネルを使用して Java コンソールを有効にし、LifeKeeper GUI をアプレットとして起動します。ユーザポリシーファイルのホームディレクトリのパスが、Java コンソールに表示されます。

## ポリシーファイルでの権限の付与

権限は、システムリソースへのアクセスを表します。アプレットにリソースへのアクセスを許可するには、対応する権限を、アクセスを試行するコードに明示的に付与する必要があります。権限は通常、名前を持ち (「ターゲット名」として参照される)、場合によっては、1 つ以上の動作を含むカンマ区切りリストを持ちます。例えば、以下のコードは、*/tmp* ディレクトリのファイル *abc* に対する読み取りアクセスを表す FilePermission オブジェクトを作成します。

```
perm = new java.io.FilePermission("/tmp/abc","read");
```

この例では、ターゲット名は「*/tmp/abc* 」、動作文字列は「*read* 」です。

ポリシーファイルは、指定したコードソースからのコードに許可する権限を指定します。この例で、*/home/sysadmin* ディレクトリのコードにファイル */tmp/abc* への読み取りアクセスを付与するポリシーファイルのエントリは以下のとおりです。

```
grant codeBase "file:/home/sysadmin/" { permissionjava.io.FilePermission
"/tmp/abc", "read"; };
```

## ポリシーファイルの例

このポリシーファイルの例には、LifeKeeper GUI の実行に必要な最小限の権限があります。このポリシーファイルは、LifeKeeper GUI パッケージにより ***/opt/LifeKeeper/htdoc/java.policy*** にインストールされます。

```
/*
```

```
* LifeKeeper GUI に必要な権限。コードベースで
* これを制限することもできます。ただし、そのようにする場合は、リカバリキットが
* 任意のコードベース付きの任意の jar コンポーネントを持つことができることに
* 注意してください。したがって、これらも含めるには
* grant 文を変更する必要があります。
*/
grant {

/*
* LifeKeeper クラスタ内のすべてのマシンに対して
* これを行うことができなければなりません。それに合わせてネットワーク仕様を
* 制限することもできます。
*/
permission java.net.SocketPermission"*", "accept,connect,resolve";
/*
* リモートプロパティファイルおよび
* jar ファイルを取得するには、URLClassLoaders を使用してください。
*/
permission java.lang.RuntimePermission"createClassLoader";
/*
* GUI をアプリケーションとして実行する場合のみ以下が必要です
* (デフォルトの RMI セキュリティマネージャは、
* ブラウザがアプレット用にインストールするセキュリティマネージャよりも
* 制限あり)。
*/
permission java.util.PropertyPermission "*","read";
permission java.awt.AWTPermission "*";
permission java.io.FilePermission "<<ALL FILES>>","read,execute";
};
```

# 5.4.4.1.1.4.4. LifeKeeper サーバでの GUI の実行

LifeKeeper GUI を実行する最も簡単な方法は、LifeKeeper サーバでアプリケーションとして実行することです。これは、実際には、同一システム上で GUI のクライアントとサーバを実行することです。

1. GUI 管理用の LifeKeeper サーバを設定した後、root として以下のコマンドを入力することにより、サーバ上で GUI をアプリケーションとして実行できます。

   ```
   /opt/LifeKeeper/bin/lkGUIapp
   ```

2. lkGUIapp スクリプトが適切な環境変数を設定して、アプリケーションを開始します。アプリケーションのロード時に、LifeKeeper のアプリケーション指定ダイアログまたはスプラッシュ画面が表示されます。

3. アプリケーションのロード後、LifeKeeper の GUI が表示され、[Cluster Connect] ダイアログが自動的に表示されます。接続先のサーバ名、およびログインとパスワードを入力してください。

4. クラスタへの接続が確立した後、GUI のウィンドウに、接続しているサーバにより保護されているリソースとステータスがグラフィックで表示されます。GUI のメニューとツールバーのボタンから、管理機能を使用できます。

# 5.4.4.1.1.4.5. LifeKeeper GUI – 概要

LifeKeeper GUI は Java テクノロジーを使用して、LifeKeeper とその設定データにグラフィカルなステータスのインターフェイスを提供します。 LifeKeeper GUI を使用すると、任意のマシンで作業しているユーザーは、クラスターマシンで必要なグループメンバーシップを持っている限り、任意のクラスター内のサーバーとリソースを管理、操作、または監視できます。 詳しくはGUI ユーザーの設定 をご覧ください。 LifeKeeper GUI サーバーについて以下に説明します。

## GUI サーバー

LifeKeeper GUI サーバーは、システム起動時に LifeKeeper クラスター内の各サーバー上で初期化されます。それは Java Native Interface (JNI)、 .NET Framework を介して LifeKeeper のCore ソフトウェアと通信します。

# 5.4.4.1.2. ステータスの表

ステータスの表には、接続しているサーバとそのリソースのステータスがグラフィック表示されます。以下の項目が表示されます。

- 最も上の行に、各サーバの状態。
- 左端の列に、各リソースのグローバル (サーバ全体での) 状態と親 / 子の関係。
- 残りのセルに、各サーバの各リソースの状態。

サーバとリソースの状態は、グラフィックス、テキスト、および色を使用して表示されます。サーバのテーブルの空白セルは、特定のリソースがそのサーバで定義されていないことを示します。

ステータスの表でサーバまたはリソースを選択した場合、その項目の詳細な状態の情報とコンテキスト依存ツールバーがプロパティパネル に表示されます。また、任意の項目のセルを右クリックすることで、該当するサーバのコンテキストメニュー またはリソースのコンテキストメニュー をポップアップ表示できます。

ステータスの表は 2 つのセクションに分かれています。左右のセクションの境界を移動して、それらのセクションの相対サイズを変更できます。また、ステータスの表を折り畳んで、階層ツリーの上位項目のみを表示できます。ツリーのリソース項目の折り畳み / 展開 を実行すると、表内にリストされる階層に対しても折り畳み / 展開が適用されます。

# 5.4.4.1.3. プロパティパネル

プロパティパネルには、ステータスの表から選択されたサーバまたはリソースのプロパティが表示されます。プロパティパネルは、[Server Properties] ダイアログ または [Resource Properties] ダイアログ と同じ機能を持ち、さらに一般的に使用するコマンドに即座にアクセスできるコンテキスト依存ツールバーがあります。このパネルの上部には、サーバを選択した場合は ***server_name*** 、リソースを選択した場合は ***server_name: resource_name*** がキャプションとして表示されます。

プロパティパネルに表示されるコンテキスト依存ツールバーは、サーバのコンテキストツールバー とリソースのコンテキストツールバー です。サーバまたはリソースのツールバーもカスタマイズできます。ツールバーのカスタマイズの詳細については、該当するApplication Recovery Kit のドキュメンテーション を参照してください。

プロパティパネル下部にあるボタンは、以下の機能を持ちます。

- **[Apply]** ボタンは、パネルの編集可能なプロパティに対する変更内容を適用します。このボタンが有効になるのは、編集可能なプロパティを変更した場合のみです。
- **[Reset]** ボタンは、サーバにすべてのプロパティの現在の値を照会し、これまで変更した内容を消去します。このボタンは常に有効です。
- **[Help]** ボタンは、プロパティパネルのコンテキスト依存ヘルプを表示します。このボタンは常に有効です。

プロパティパネルのサイズを増減するには、パネルの左端にある境界を左右にスライドしてください。このパネルを開閉するには、[View] メニュー の **[Properties Panel]** チェックボックス を使用してください。

# 5.4.4.1.4. 出力パネル

出力パネルは、LifeKeeper GUI クライアントが送出したコマンドの出力を収集します。コマンドの実行開始時に、タイムスタンプ付きのラベルが出力パネルに追加され、そのラベルの下に、そのコマンドの出力がすべて追加されます。複数のコマンドを同時に実行する場合 (通常は異なるサーバ上)、各コマンドの出力が対応するセクションに送られ、各コマンドの結果が見やすくなります。

出力パネルのサイズを増減するには、パネル上部にある境界を上下にスライドしてください。このパネルを開閉するには、[View] メニュー の **[Output Panel]** チェックボックスを使用してください。出力パネルを閉じているときには、各コマンドを開始するダイアログが表示されたままになり、このダイアログを閉じるまで出力がこのダイアログに表示されます。そして、このダイアログを閉じた後はコマンドの出力を確認できなくなります。出力パネルを再び開いた後は、LifeKeeper の GUI はデフォルトの動作に戻ります。

# 5.4.4.1.5. メッセージバー

メッセージバーは、[Status] ウィンドウの下に表示されます。メッセージが 1 行のテキストで表示されます。「Connecting to Server X」や「Failure to connect to Server X」などのメッセージが表示されます。

メッセージバーを非表示にするには、[View] メニュー の **[Message Bar]** チェックボックスをオフにします。

メッセージバーを表示するには、[View] メニューの **[Message Bar]** チェックボックスをオンにします。

メッセージバーに表示されたメッセージの履歴を表示する方法については、メッセージ履歴の表示 を参照してください。

# 5.4.4.1.6. GUI の終了

[File] メニュー から **[Exit]** を選択すると、すべてのサーバから切断され、GUI のウィンドウが閉じます。

# 5.4.4.1.7. 共通の作業

以下に、すべてのユーザが実行できる基本作業を示します。

---


LifeKeeper の起動

LifeKeeper の停止

LifeKeeper プロセスの表示

LifeKeeper GUI サーバプロセスの表示

LifeKeeper の制御プロセスの表示

サーバのクラスタへの接続

クラスタからの切断

接続サーバの表示

サーバのステータスの表示

サーバプロパティの表示

サーバログファイルの表示

リソースのタグと ID の表示

リソースのステータスの表示

リソースのプロパティの表示

Resource Labels

メッセージ履歴の表示

リソース階層ツリーの展開および折り畳み

[Cluster Connect] ダイアログ

[Cluster Disconnect] ダイアログ

[Resource Properties] ダイアログ

[Server Properties] ダイアログ

# 5.4.4.1.7.1. LifeKeeper の起動

SPS ソフトウェアはすべて、 */opt/LifeKeeper* ディレクトリにインストールされます。

すべての確認作業 が完了すると、両方のサーバで LifeKeeper を起動する準備が整います。このセクションでは、LifeKeeper サーバデーモンプロセスの起動について説明します。LifeKeeper GUI アプリケーションは、別個のコマンドを使用して起動され、LifeKeeper GUI の設定 に説明されています。LifeKeeper には、LifeKeeper デーモンプロセスの起動と停止を行うコマンドラインインターフェース が用意されています。これらのデーモンプロセスは、LifeKeeper GUI を起動する前に実行する必要があります。

## LifeKeeper サーバプロセスの起動

> **!** LifeKeeperが起動し、クラスター内の他のサーバーとの通信を確立する際、クラスター内のすべてのサーバーとの通信が確立されるまで、LifeKeeperはデータレプリケーション（DataKeeper）リソースがIn-Service（手動または自動）になるのを許可しません。すべてのサーバーとの通信が確立される前にデータレプリケーションリソースをIn-Serviceにしようとすると、データレプリケーションリソースはOSFとマークされます（詳細については、システムのログファイルを参照してください）。

LifeKeeper がシステムで現在実行されていない場合は、すべてのサーバで root ユーザとして次のコマンドを入力してください。

```
/opt/LifeKeeper/bin/lkstart
```

このコマンドを実行すると LifeKeeper サービスが開始され、LifeKeeper はシステムの起動時に自動的に開始するように設定されます。
数秒の遅延の後、情報メッセージが表示されます。

> **✻** 注記: LifeKeeper を起動するときに **LifeKeeper Distribution Enabling Package** を参照するエラーメッセージが表示された場合は、LifeKeeper インストールイメージファイル をインストール/再インストールする必要があります。

lkstartコマンドの詳細については、コマンドラインに man LCD と入力して、LCD ヘルプページを参照してください。

また、自動起動を有効にせずにLifeKeeperプロセスのみを起動させる場合には、以下のコマンドを実行してください。

```
service lifekeeper start (または, systemctl start lifekeeper)
```

## LifeKeeper の自動起動の有効化

上述のコマンドで LifeKeeper が起動しますが、システムを再起動するたびに毎回同じコマンドを実行する必要があります。サーバの起動時に LifeKeeper が自動的に起動するようにするには、次のコマンドを入力してください。

```
chkconfig lifekeeper on (または, systemctl enable lifekeeper)
```

詳細については、chkconfig マニュアルページを参照してください。

# 5.4.4.1.7.2. LifeKeeper の停止

LifeKeeper を停止する必要がある場合は、root ユーザーで次のコマンドから用途に合ったものを入力して停止してください。

## systemctl コマンドを使用する

- **systemctl set-environment LKSTOP_MODE=stop-nofailover; systemctl stop lifekeeper; systemctl unset-environment LKSTOP_MODE**
  このコマンドを使用すると、ローカルシステム上の LifeKeeper をシャットダウンします (実行されている場合)。まず、すべての保護されたリソースをローカルシステム上のサービスから取り除いてから、LifeKeeper デーモンをシャットダウンします。保護されたリソースは、クラスタ内の別のシステムにフェイルオーバされません。

- **systemctl set-environment LKSTOP_MODE=stop-daemons; systemctl stop lifekeeper; systemctl unset-environment LKSTOP_MODE**
  このコマンドを使用すると、リソースをサービスから取り除くセクションを実行しません。 リソースはローカルシステム上で引き続き実行されますが、LifeKeeper の保護はなくなります。リソースを正常にシャットダウンしない場合、SCSI ロックなどの項目などを取り除くことができなくなるため、このコマンドの使用には注意が必要です。このコマンドを実行した後、そのシステムで障害が発生するかシステムがシャットダウンされた場合、システムは適切なリソースにフェイルオーバを開始できません。

- **systemctl stop lifekeeper**
  このコマンドを使用すると、サービスからリソースを取り除きますが、!nofailover! フラグ [LifeKeeper のフラグ を参照] を通信可能などのシステムにも設定しません。これは、shutdown_switchover フラグが設定されたらフェイルオーバが実行されることを意味します。shutdown_switchover が設定されていない場合、このコマンドは、LKSTOP_MODE に stop-nofailover を指定した時と同様に動作します。

- **systemctl set-environment LKSTOP_MODE=stop-failover; systemctl stop lifekeeper; systemctl unset-environment LKSTOP_MODE**
  このコマンドを使用すると、リソースをサービスから取り除き、フェイルオーバが実行されます。このコマンドは、shutdown_switchover フラグを設定して systemctl stop lifekeeper コマンドを実行した場合と同様の動作をします。

> 注記: 上記のコマンドは、lifekeeper.service の自動起動設定を変更しません。従って、システム再起動時に LifeKeeper が自動起動する/しないはシステム終了時の lifekeeper.service の自動起動設定に従います。

## lkstop コマンドを使用する

> 注記: リソースの停止が失敗した場合でも lkstop の プロセスは続行 され、lkstopが失敗することは ありません 。

- **/opt/LifeKeeper/bin/lkstop**
  このコマンドを実行すると、現在実行されている場合ローカルシステムの LifeKeeper はシャットダウンされます。まず保護されているすべてのリソースをローカルシステムのサービスから停止し、次に LifeKeeper デーモンをシャットダウンします。

- **/opt/LifeKeeper/bin/lkstop -f**
  このコマンドは、サービスからリソースを停止するセクションをスキップします。リソースは引き続きローカルシステムで実行されますが、LifeKeeper には保護されなくなります。

その他のオプションと使用上のベストプラクティスについては、コマンド を参照してください。

注記: 上記のコマンドは lifekeeper.service の自動起動設定を無効に変更します（-r オプション指定時は除く）。従って、実行後に lifekeeper.service の自動起動設定を有効にしない限り、システムを再起動しても LifeKeeper は自動起動しません。

「lkstop -f」を使用してLifeKeeperを停止すると、データレプリケーションリソースを含め、In-Serviceのリソースは構成された状態で実行されたままとなります。「lkstop -f」の後、階層/リソースが正しいサーバーでIn-Serviceに戻れることが重要です。すべてのサーバーとの通信が確立されると、LifeKeeperはすべてのリソースを自動的にIn-Serviceにします。LifeKeeperがすべてのサーバーの再起動とクラスターへの再参加を待機している間、データレプリケーションリソースは一時的にOSFとマークされる場合があります（詳細については、システムのログファイルを参照してください）。

## 各停止用コマンドでの挙動の違い

各コマンドの挙動をまとめると、以下のようになります。

| コマンド | 自動起動設定 | リソース | フェイルオーバー |
|---|---|---|---|
| `systemctl set-environment LKSTOP_MODE=stop-nofailover; \`<br>`systemctl stop lifekeeper; \`<br>`systemctl unset-environment LKSTOP_MODE` | 変更しない | 停止する | しない |
| `systemctl set-environment LKSTOP_MODE=stop-daemons; \`<br>`systemctl stop lifekeeper; \`<br>`systemctl unset-environment LKSTOP_MODE` | 変更しない | 停止しない | しない |
| `systemctl stop lifekeeper` | 変更しない | 停止する | その時のshutdown_switchoverフラグのON/OFFに従う |

| | | | |
|---|---|---|---|
| `systemctl set-environment LKSTOP_MODE=stop-failover; \`<br>`systemctl stop lifekeeper; \`<br>`systemctl unset-environment LKSTOP_MODE` | 変更しない | 停止する | する |
| `lkstop` | 無効化 | 停止する | しない |
| `lkstop -f` | 無効化 | 停止しない | しない |

## LifeKeeper の自動起動の無効化

システムの再起動時に LifeKeeper が自動で再起動しないようにするには、次のコマンドを入力してください。

```
chkconfig lifekeeper off
```

*または*

```
systemctl disable lifekeeper
```

詳細については、chkconfig （または systemctl）マニュアルページを参照してください。

# 5.4.4.1.7.3. LifeKeeper プロセスの表示

現在実行されているすべての LifeKeeper Core デーモンプロセスのリストを表示するには、次のコマンドを実行してください。

```
ps -ef | grep LifeKeeper | grep -w bin | grep -v lklogmsg
```

出力の例を以下に示します。

```
root 11880 11873 0 14:03 ? 00:00:00 /opt/LifeKeeper/bin/lk_logmgr -l/opt/LifeK
eeper/out -d/etc/default/LifeKeeper
root 12240 11877 0 14:04 ? 00:00:00 /opt/LifeKeeper/bin/lcm
root 12247 11879 0 14:04 ? 00:00:00 /opt/LifeKeeper/bin/ttymonlcm
root 12250 11876 0 14:04 ? 00:00:00 /opt/LifeKeeper/bin/lcd
root 12307 11874 0 14:04 ? 00:00:00 /opt/LifeKeeper/bin/lkcheck
root 12311 11875 0 14:04 ? 00:00:00 /opt/LifeKeeper/bin/lkscsid
root 12325 11871 0 14:04 ? 00:00:00 /opt/LifeKeeper/bin/lkvmhad
root 12335 12330 0 14:04 ? 00:00:00 /opt/LifeKeeper/bin/perl /opt/LifeKeeper/h
tdoc/cgi-bin/DoRequest.fcgi
```

LifeKeeper の実行状態は次のコマンドで確認できます。

```
/opt/LifeKeeper/bin/lktest
```

LifeKeeper が実行中の場合は次のように出力されます。

```
F S UID PID PPID C CLS PRI NI SZ STIME TIME CMD
4 S root 12240 11877 0 TS 39 -20 6209 14:04 00:00:00 lcm
4 S root 12247 11879 0 TS 39 -20 30643 14:04 00:00:00 ttymonlcm
4 S root 12250 11876 0 TS 29 -10 9575 14:04 00:00:00 lcd
```

LifeKeeper が実行中でない場合は何も出力されず、コマンドは 1 で終了します。

✻ グラフィカルユーザーインターフェース (GUI) に必要なプロセスとともに、LifeKeeper Core デーモンを開始、停止、および監視する別の LifeKeeper プロセスも実行されています。プロセスのリストについては、LifeKeeper 制御プロセスの表示 および LifeKeeper GUI サーバプロセスの表示 を参照してください。また、ほとんどの LifeKeeper プロセスには子プロセス lklogmsg があり、予期しない出力をキャプチャして記録します。

# 5.4.4.1.7.4. LifeKeeper GUI サーバプロセスの表示

LifeKeeper GUI サーバが動作していることを確認するには、以下のコマンドを入力してください。

```
ps -ef | grep runGuiSer
```

以下のような出力が表示されます。

```
root 2805 1 0 8:24 ?00:00:00 sh /opt/LifeKeeper/bin/runGuiServer
```

現在動作している他の GUI サーバのデーモンプロセスのリストを表示するには、以下のコマンドを入力してください。

```
ps -efw | grep S_LK
```

以下のような出力が表示されます。

```
root 819 764 0 Oct16 ? 00:00:00 java -Xint -Xss3M -DS_LK=true
-Djava.rmi.server.hostname=wake -Dcom.steeleye.LifeKeeper.rmiPort=82
-Dcom.steeleye.LifeKeeper.LKROOT=/opt/LifeKeeper
-DGUI_RMI_REGISTRY=internal -DGUI_WEB_PORT=81
com.steeleye.LifeKeeper.beans.S_LK
```

LifeKeeper GUI サーバ管理 Web サーバが動作していることを確認するには、以下のコマンドを入力してください。

```
ps -ef|grep steeleye-light | egrep -v "lklogmsg|runsv"
```

以下のような出力が表示されます。

```
root 12330 11872 0 14:04 ? 00:00:00 /opt/LifeKeeper/sbin/steeleye-
lighttpd -D -f/opt/LifeKeeper/etc/lighttpd/lighttpd.conf
```

# 5.4.4.1.7.5. LifeKeeper の制御プロセスの表示

LifeKeeper の制御プロセスが動作していることを確認するには、以下のコマンドを入力してください。

```
ps -ef | grep runsv
```

以下のような出力が表示されます。

```
root 29093 1 0 11:35 ? 00:00:00 /opt/LifeKeeper/sbin/runsvdir -P /opt/
LifeKeeper/etc/service

root 29097 29093 0 11:35 ? 00:00:00 /opt/LifeKeeper/sbin/runsv lcd

root 29098 29093 0 11:35 ? 00:00:00 /opt/LifeKeeper/sbin/runsv lcm

root 29099 29093 0 11:35 ? 00:00:00 /opt/LifeKeeper/sbin/runsv lk_logmgr

root 29100 29093 0 11:35 ? 00:00:00 /opt/LifeKeeper/sbin/runsv lkcheck

root 29101 29093 0 11:35 ? 00:00:00 /opt/LifeKeeper/sbin/runsv lkscsid

root 29102 29093 0 11:35 ? 00:00:00 /opt/LifeKeeper/sbin/runsv lkvmhad

root 29103 29093 0 11:35 ? 00:00:00 /opt/LifeKeeper/sbin/runsv steeleye-
lighttpd

root 29104 29093 0 11:35 ? 00:00:00 /opt/LifeKeeper/sbin/runsv ttymonlcm

root 29105 29093 0 11:35 ? 00:00:00 /opt/LifeKeeper/sbin/runsv
lkguiserver

root 29465 2894 0 11:36 pts/0 00:00:00 grep --color=auto runsv
```

これらのプロセスは LifeKeeper Core のデーモンプロセスの開始、停止、および監視を行います。LifeKeeper を開始するにはこれらのプロセスが動作している必要があります。これらのプロセスは、デフォルトでシステムの起動時に開始するように設定されています。この動作は変更しないでください。

# 5.4.4.1.7.6. サーバのクラスタへの接続

1. 開始するには、以下の 2 つの方法があります。
   - ◦ グローバルツールバー の **[Connect]** ボタンをクリックする。
   - ◦ [File] メニュー の **[Connect]** をクリックする。
2. [Cluster Connect] ダイアログ の **[Server Name]** フィールドに、接続するクラスタ内のサーバ名を入力してください。

> 注記: **IPv6** アドレスを使用する場合は、このアドレスを大かっこ [ ] で囲む必要があります。これにより、マシンの IPv6 アドレス経由で接続を確立できます。別の方法として、名前をアドレスに割り当てることができ、その名前を使用して接続できます。

3. **[Login]** と **[Password]** のフィールドに、指定のサーバ上で LifeKeeper が認証に使用するユーザのログイン名とパスワードを入力してください。
4. **[OK]** をクリックしてください。

GUI が正常に指定サーバに接続した場合、GUI は、新しいサーバが検出されなくなるまで、クラスタ内にあるすべての既知のサーバへの接続 (およびステータス表示への追加) を継続します。

> 注記 : 最初のログイン名とパスワードが、クラスタ内のサーバ上にあるクライアントで認証に失敗した場合、そのサーバでの別のログイン名とパスワードを入力するように要求されます。[Password] ダイアログ で **[Cancel]** を選択した場合、サーバへの接続は中止され、GUI はクラスタ内の残りのコンポーネントへの接続を継続します。

# 5.4.4.1.7.7. クラスタからの切断

この作業は、選択したサーバ経由で、GUI クライアントをクラスタ内のすべてのサーバから切断します。

1. 開始するには、以下の 3 つの方法があります。
   - ◦ グローバルツールバー の **[ Disconnect ]** ボタンをクリックする。
   - ◦ [Edit] メニュー の **[Server]** を選択し、 **[Disconnect]** をクリックする。
   - ◦ サーバのコンテキストツールバー が表示される場合は、そこにある **[Disconnect]** ボタンをクリックする。
2. [Cluster Disconnect] ダイアログ の **[Select Server in Cluster]** リストから、切断するクラスタ内のサーバ名を選択してください。
3. **[OK]** をクリックしてください。クラスタ内の全サーバのリストを持つ **[Confirmation]** ダイアログが表示されます。
4. **[Confirmation]** ダイアログの **[OK]** をクリックして、クラスタ内の全サーバからの切断を確定してください。

クラスタからの切断後、そのクラスタ内にあるすべてのサーバが、GUI のステータス表示から消去されます。

# 5.4.4.1.7.8. 接続サーバの表示

サーバの状態は、下図に示すように、表内のサーバのグラフィック表示で表されます。サーバアイコンが視覚的に示すサーバの状態の詳細については、サーバの状態の表示 を参照してください。

# 5.4.4.1.7.9. サーバのステータスの表示

サーバの状態は、下図に示すように、表内のサーバのグラフィック表示で表されます。

| サーバの状態 | 状態のシンボル | 意味 |
|---|---|---|
| ALIVE | | クライアントはサーバに有効な接続を行うことができます。<br><br>このサーバから ALIVE のリモートサーバへのコミュニケーションパスが ALIVE です。<br><br>DEAD とマークされたコミュニケーションパス、および DEAD のサーバをターゲットとするコミュニケーションパスは無視されます。これは、DEAD のサーバには DEAD のグラフィックで表されるからです。 |
| ALIVE | | クライアントはサーバに有効な接続を行うことができます。<br><br>このサーバから指定リモートサーバへの 1 つ以上のコミュニケーションパスが DEAD です。<br><br>このサーバから指定リモートサーバへの間には、冗長コミュニケーションパスが存在しません。 |
| DEAD | | クラスタ内の他のサーバから DEAD として報告されました。 |
| UNKNOWN | | ネットワーク接続が失われました。最後に分かっている LifeKeeper の状態が ALIVE です。 |

# 5.4.4.1.7.10. サーバのプロパティの表示

1. 開始するには、以下の 2 つの方法があります。
   - プロパティを表示するサーバのアイコンを右クリックします。サーバのコンテキストメニューが表示されたら、 **[Properties]** をクリックします。サーバのプロパティは、サーバをクリックすると プロパティパネル (有効になっている場合) にも表示されます。
   - [Edit] メニュー の **[Server]** をポイントし、 **[Properties]** をクリックします。ダイアログが表示されたら、表示するサーバを **[Server]** リストから選択します。

2. 別のサーバのプロパティを表示する場合は、サーバを **[Server]** リストから選択してください。

3. 確認が完了したら、 **[OK]** をクリックしてウィンドウを閉じてください。

# 5.4.4.1.7.11. サーバのログファイルの表示

1. 開始するには、以下の 4 つの方法があります。

   - ◦ サーバのアイコンを右クリックしてサーバのコンテキストメニュー を表示し、次に **[View Log]** をクリックして [LifeKeeper Log Viewer] ダイアログを表示する。
   - ◦ グローバルツールバー の **[View Log]** ボタンをクリックし、[LifeKeeper Log Viewer] ダイアログの [Server] リストから、表示するサーバを選択する。
   - ◦ サーバのコンテキストツールバー が表示される場合は、その **[View Log]** ボタンをクリックする。
   - ◦ [Edit] メニュー の **[Server]** をポイントし、 **[View Log]** をクリックする。次に、**[LifeKeeper Log Viewer]** ダイアログの [Server] リストから、表示するサーバを選択する。

2. グローバルツールバー または **[Edit]** メニュー から捜査を開始して、別のサーバのログを表示する場合は、[LifeKeeper Log Viewer] ダイアログの **[Server]** リストから、そのサーバを選択します。 **サーバのコンテキストメニュー** または **サーバのコンテキストツールバー** から **[View Log]** を選択した場合は、この機能は使用できません。

3. 確認が完了したら、 **[OK]** をクリックして **[Log Viewer]** ダイアログを閉じてください。

# 5.4.4.1.7.12. リソースのタグと ID の表示

リソースのタグと ID を即座に表示するには、[Status] ウィンドウのリソースアイコンにカーソルを合わせて、マウスの左ボタンを 1 回押します (シングルクリック)。優先順位が最も低いサーバのリソースのタグと ID がメッセージバーに表示されます。特定サーバ上にあるリソースのタグと ID を表示する場合は、表内のリソースインスタンスセルを左クリックしてください。

メッセージバーに表示されるメッセージは、以下のようになります。

```
Resource Tag = ipdnet0-153.98.87.73, Resource ID = IP-153.98.87.73
```

特定の状況では、GUI がリソース ID を特定できないことがあります。この場合は、リソースタグのみがメッセージバーに表示されます。

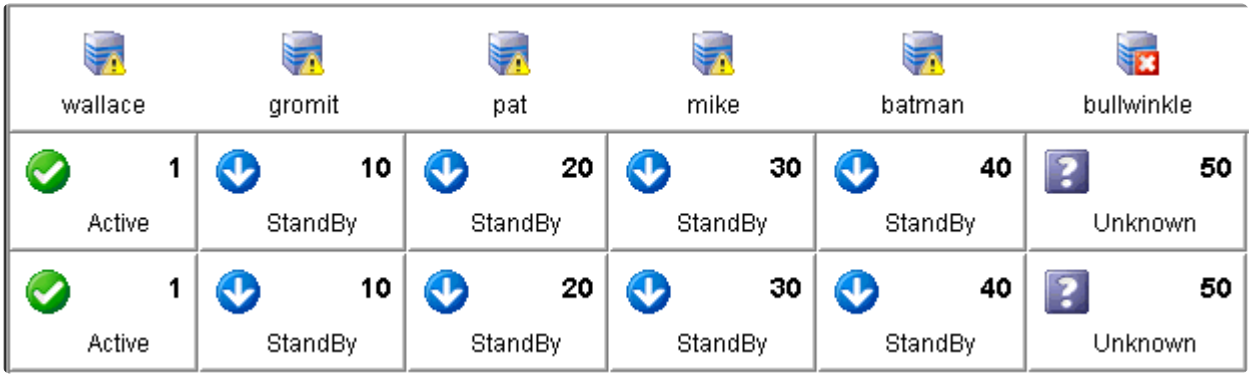
# 5.4.4.1.7.13. リソースのステータスの表示

リソースのステータスつまり状態は、 **グローバルリソースのステータス** (すべてのサーバについて) と **サーバリソースのステータス** (1 台のサーバ上) の 2 つの形式で表示されます。グローバルリソースのステータスは、 **[Status]** ウィンドウの左ペインにある **リソース階層ツリー** に表示されます。サーバリソースのステータスは、リソースの列とサーバ行の交点にある表のセルにあります。

## サーバリソースのステータス

下図に、アクティブ、スタンバイ、および不明のリソースステータスを持つサーバを示します。

- 「wallace」上のリソースはすべて、アクティブです。
- 「gromit」、「pat」、「mike」、および「batman」上のリソースはすべてスタンバイです。
- 「bullwinkle」上のリソースはすべて、不明です。




| サーバリソースの状態 | 状態のシンボル | 意味 |
|---|---|---|
| アクティブ | | このサーバ上でリソースは動作可能であり、保護されています。 (ISP) |
| 可用性の低下 | | このサーバ上でリソースは動作可能ですが、バックアップリソースによる保護はされていません。 (ISU) |
| スタンバイ | | サーバは、リソースの動作を引き継ぐことができます。(OSU) |
| 障害 | | このサーバ上のリソースに問題が検出されました。例えば、リソースを In Service にする試行が失敗しました。(OSF) |
| 不明 | | リソースが初期化されていないか (ILLSTATE)、このサーバで LifeKeeper が動作中でありません。 |
| | 空のパネル | サーバのリソースが定義されていません。 |

# グローバルリソースのステータス

| 状態のシンボル | 説明 | 意味 / 原因 |
|---|---|---|
| | 正常 | リソースがアクティブ (ISP) で、バックアップがアクティブです。 |
| | 警告 | リソースがアクティブ (ISP) です。1 つ以上のバックアップが、不明または障害 (OSF) としてマークされています。 |
| | 障害。リソースが、いずれのサーバでもアクティブでありません (OSF)。 | リソースが、通常の原因により Out of Service になりました。<br><br>リソースが、通常ではない方法により動作が停止しました。<br><br>リカバリは完了していないか、失敗しました。 |
| | 不明。利用可能な情報からは、状態を特定できませんでした。 | 複数のサーバがアクティブであることを告げています。<br><br>サーバへの接続が遮断されました。<br><br>サーバのリソースインスタンスがすべて、不明の状態です。 |

# 5.4.4.1.7.14. リソースのプロパティの表示

1. 開始するには、以下の 3 つの方法があります。

   - º プロパティを表示するリソース / サーバの組み合わせのアイコンを右クリックします。

リソースのコンテキストメニュー が表示されたら、[ **Properties** ]をクリックします。リソースのプロパティは、プロパティパネル (有効になっている場合) にも表示されます。

   - º プロパティを表示するグローバルリソースのアイコンを右クリックします。

リソースのコンテキストメニュー が表示されたら、 **[Properties]** をクリックします。ダイアログが表示されたら、表示するリソースが存在するサーバを **[Server]** リストから選択します。

   - º

[Edit] メニュー の **[ Resource ]** をポイントし、**[ Properties ]** をクリックします。ダイアログが表示されたら、プロパティを表示するリソースを **[Resource]** リストから選択し、表示するリソースが存在するサーバを **[Server]** リストから選択します。

2. 別のリソースのプロパティを表示する場合は、リソースを **[Resource]** リストから選択してください。

3. 別のサーバ上にあるリソースのプロパティを表示する場合は、サーバを **[Server]** リストから選択してください。

4. 確認を終了したら、 **[OK]** をクリックしてウィンドウを閉じてください。

# 5.4.4.1.7.15. Resource Labels

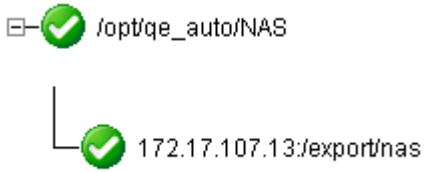
## Resource Labels

このオプショングループを使用すると、リソース階層ツリー内のリソースを、タグ名別 と ID 別のいずれで表示するかを指定できます。

注記：リソース階層ツリーに表示されるリソースタグ /ID は、優先順位が最も低い番号を持つサーバに属します。特定サーバ上にあるリソースのタグ /ID を表示する場合は、表内のリソースインスタンスセルを左クリックしてください。メッセージバーにそのタグ /ID が表示されます。

**By tag name:**

**By ID:**



# 5.4.4.1.7.16. メッセージ履歴の表示

1. [View] メニュー の [ **History** ] をクリックしてください。LifeKeeper GUI の [Message History] ダイアログが表示されます。

2. 履歴のメッセージをすべて消去する場合は、 **[Clear]** をクリックしてください。

3. ダイアログを閉じるには、 **[OK]** をクリックしてください。

**[Message History]** ダイアログには、メッセージバーからの最新のメッセージが表示されます。履歴リストには、最大 1000 行を表示できます。最大行数を超えた場合、新しいメッセージにより最も古いメッセージが「押し出され」ます。

これらのメッセージは、クライアントとサーバとの間の動作のみを表し、時系列で表示されます。最新のメッセージがリストの上部に表示されます。

## メッセージ履歴の解釈

**<—** は、メッセージがサーバから受信したことを示し、通常は以下の形式をとります。

```
<--"server name":"action"

<--"server name":"app res": "action"

<--"server name":"res instance":"action"
```

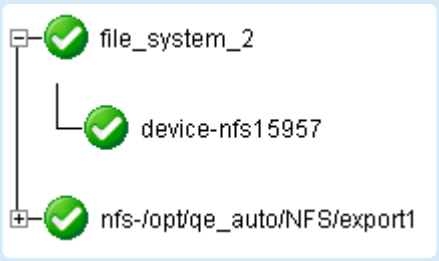
**—>** はメッセージがクライアントから送信されたことを示し、通常は以下の形式をとります。

```
-->"server name":"action"

-->"server name":"app res": "action"

-->"server name":"res instance":"action"
```

**[Clear]** ボタンをクリックすると、履歴が消去されますが、ダイアログは閉じません。

**[OK]** ボタンをクリックすると、履歴を消去せずにダイアログが閉じます。

# 5.4.4.1.7.17. リソース階層ツリーの展開と折り畳み

| | |
|---|---|
|  | このツリーのセグメントでは、リソース *file_system_2* が展開されており、リソース *nfs-/opt/qe_auto/NFS/export1* が折り畳まれています。<br><br>展開されているリソースアイコンの左には、⊟ が表示されます。<br><br>折り畳まれているリソースアイコンの左には、⊞ が表示されます。 |

リソース階層ツリーを **展開** するには、

- ◦ ⊞ をクリックするか、
- ◦ ⊞ の右側にあるリソースアイコンをダブルクリックしてください。

リソース階層ツリーを **すべて展開** するには、

- ◦ **[ View ]** メニュー の **[ Expand Tree ]** をクリックするか、
- ◦ **[Status]** ウィンドウの左ペインにある列ヘッダの **[Resource Hierarchy Tree]** ボタンをダブルクリックしてください。

> **注記:** リソース階層ツリーに表示されるリソースタグ /ID は、優先順位が最も低い番号を持つサーバに属します。特定サーバ上にあるリソースのタグ /ID を表示する場合は、表内のリソースインスタンスセルを左クリックしてください。メッセージバーにそのタグ /ID が表示されます。

リソース階層ツリーを折り畳むには、

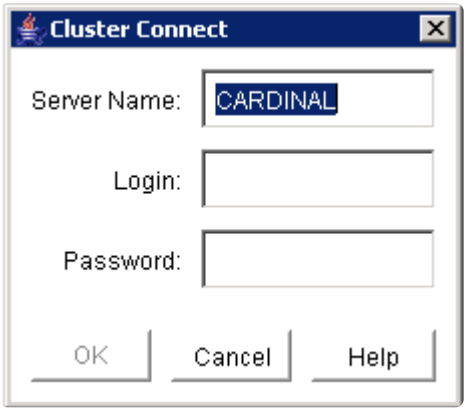
- ◦ ⊟ をクリックするか、
- ◦ ⊟ の右側にあるリソースアイコンをダブルクリックしてください。

リソース階層ツリーをすべて折り畳むには、

- ◦ **[View]** メニュー の **[Collapse Tree]** をクリックするか、
- ◦ [Status] ウィンドウの左ペインにある列ヘッダの **[Resource Hierarchy Tree]** ボタンをダブルクリックしてください。

**注記:**「9」と「0」のキーは、すべてのリソース階層ツリーに対して即座に展開 / 折り畳みを実行するホットキー / アクセラレータキーとして指定されています。

# 5.4.4.1.7.18. [Cluster Connect] ダイアログ

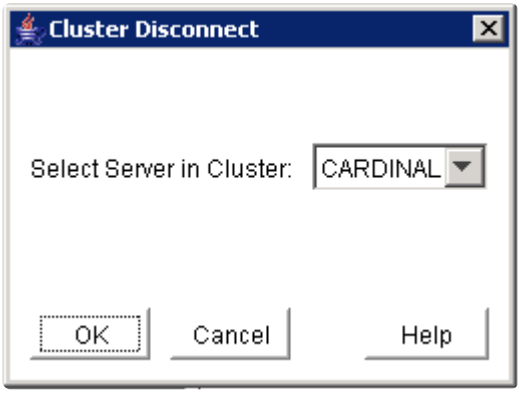



**Server Name** - 接続先のサーバ名。

**Login** - 接続先のサーバに LifeKeeper 認証情報を持つユーザのログイン名。

**Password** - 接続先のサーバで指定ログインを認証するパスワード。

# 5.4.4.1.7.19. [Cluster Disconnect] ダイアログ




**Select Server in Cluster** -

接続しているサーバの名前がドロップダウンリストボックスに表示されます。リストから、切断するクラスタのサーバを選択してください。切断されるクラスタ内のすべてのサーバが、確認ダイアログに表示されます。

# 5.4.4.1.7.20. [Resource Properties] ダイアログ

[Resource Properties] ダイアログは、[Edit] メニュー やリソースコンテキストメニュー から使用できます。このダイアログには、サーバ上にある特定のリソースのプロパティが表示されます。[Edit] メニューからアクセスした場合は、リソースとサーバを選択できます。リソースコンテキストメニューからアクセスした場合、はサーバを選択できます。

## [General] タブ

- ◦ Tag - リソースインスタンスの名前。システムに対して一意で、管理者にリソースを示します。
- ◦ ID - リソースインスタンスに関連する文字列であり、リソースタイプのすべてのインスタンス間で一意です。関連するアプリケーションソフトウェアに対して、リソースインスタンスの内部特性のいくつかを示します。
- ◦ Switchback (管理者権限を持つユーザは編集可能) - サービス起動中のリソースが存在するサーバに障害が発生した場合に、サーバのリカバリ動作を管理する設定。この設定が [Intelligent] の場合、指定リソースの可能なバックアップとしてサーバが動作します。この設定が [Automatic] の場合、サーバはアクティブにリソースの再取得を試行します (以下の条件が満たされる場合)。
  - º サーバがクラスタから離れるときには、リソース階層のサービスが既に起動している必要があります。
  - º リソース階層がすべてサービス起動している場合は、低プライオリティのサーバでサービスを起動している必要があります。

  **注記:** 自動スイッチバックのチェックは、LifeKeeper を起動したとき、またはクラスタに新しいサーバを追加したときにのみ実行されます。通常のクラスタ動作中には実行されません。
- ◦ State -リソースインスタンスの現在の状態。
  - º

*Active* - ローカルで in service であり、保護されています。
- º

*Warning* - ローカルで in service ですが、ローカルリカバリは試行されません。
- º

*Failed* - out of service、障害。
- º

*Standby* - サービス停止、障害なし。

- ◦

*LLSTATE* - LifeKeeper の起動シーケンスの一部として実行されるリソース初期化プロセスにより適切に初期化されていません。この状態のリソースは、LifeKeeper で保護されていません。

- ◦

*UNKNOWN* - リソースの状態を特定できませんでした。GUI サーバが使用できない可能性があります。

- ◦ Reason -存在する場合、リソースが現在の状態にある原因 (つまり、最後の状態変化の原因) を示します。例えば、galahad 上にあるアプリケーションの状態が OSU である原因は、tristan 上にある共有プライマリリソース *ordbfsaa-on-tristan* の状態が ISP か ISU であることです。共有リソースは、グループ内の 1 つのシステムでのみ同時にアクティブにできます。

- ◦ Initialization - 起動時のリソースの初期化動作を決定する設定であり、AUTORES_ISP、INIT_ISP、INIT_OSU などがあります。

## [Relations] タブ

- ◦ Parent - このリソースに直接依存するリソースのタグ名を示します。

- ◦ Child - このリソースが依存するすべてのリソースのタグ名を示します。

- ◦ Root - このリソース階層で、親を持たないリソースのタグ名。

## [Equivalencies] タブ

- ◦ Server - リソースが定義済みの同等性を持つサーバ名。

- ◦ Priority (管理者権限を持つユーザは編集可能) - このリソースについて、ターゲットサーバのフェイルオーバの優先順位の値。

- ◦ Tag -同等のサーバ上にあるこのリソースのタグ名。

- ◦ Type -同等性のタイプ (SHARED、COMMON、COMPOSITE)。

- ◦ Reorder Priorities (管理者権限を持つユーザは編集可能) - [Up]/[Down] ボタンを使用して、選択した同等リソースの優先順位を並べ替えることができます。

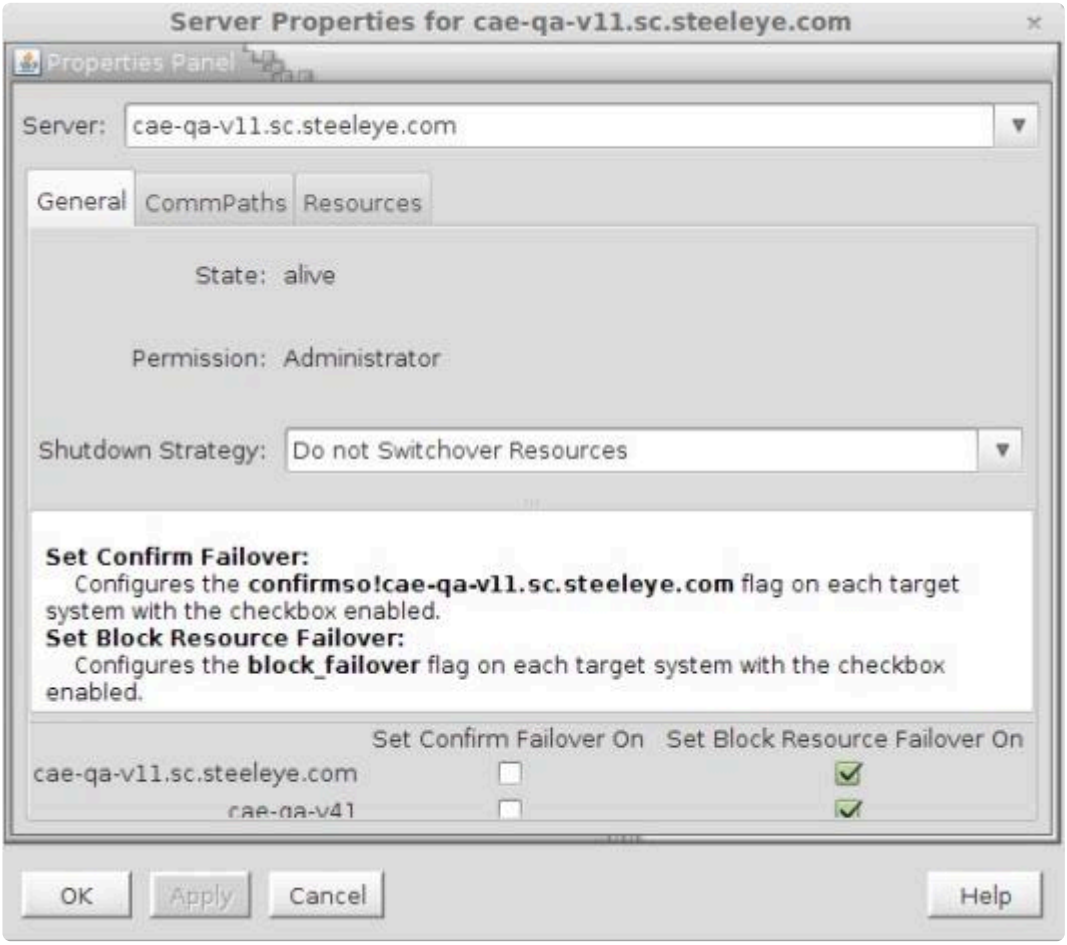
[OK] ボタンをクリックすると、変更内容が適用されてウィンドウが閉じます。[Apply] ボタンをクリックすると、変更内容が適用されます。[Cancel] ボタンをクリックすると、最後に [Apply] をクリックして以降の変更内容を保存せずに、ウィンドウが閉じます。

# 5.4.4.1.7.21. [Server Properties] ダイアログ

[Server Properties] ダイアログは、サーバのコンテキストメニューや[Edit] メニュー から使用できます。このダイアログには、特定のサーバのプロパティが表示されます。サーバのプロパティは、プロパティパネル (有効になっている場合) にも表示されます。

このダイアログの 3 つのタブについて説明します。[OK] ボタンをクリックすると、変更内容が適用されてウィンドウが閉じます。[Apply] ボタンをクリックすると、変更内容が適用されます。[Cancel] ボタンをクリックすると、最後に [Apply] をクリックして以降の変更内容を保存せずに、ウィンドウが閉じます。

## [General] タブ




- **Name** - 選択したサーバの名前。
- **State** -サーバの現在の状態。 サーバの状態は以下の値をとります。
  - *ALIVE* - サーバが使用可能。
  - *DEAD* - サーバが使用不可。
  - *UNKNOWN* - リソースの状態を特定できませんでした。GUI サーバが使用できない可能性があります。
- **Permission** - そのサーバに現在ログインしているユーザの権限レベル。権限は以下の値をとります。
  - *Administrator* - ユーザは LifeKeeper のすべての作業を実行できます。
  - *Operator* - ユーザは、LifeKeeperのリソースとサーバのステータスを監視でき、リソースのサービス起動やサービス停止ができます。

    - *Guest* - ユーザは LifeKeeper のリソースとサーバのステータスを監視できます。
- **Shutdown Strategy** (管理者権限を持つユーザは編集可能) - サーバがシャットダウンしたときに、リソースがクラスタ内のバックアップサーバにスイッチオーバするかどうかを制御する設定。設定「*Switchover Resources*」は、リソースがクラスタ内のバックアップサーバでサービスが起動することを示します。設定「*Do not Switchover Resources*」は、リソースがクラスタ内にある別のサーバでサービスが起動しないことを示します。
- **Failover Strategy** - この設定を使用して、LifeKeeper のクラスタ内にある特定システムからのフェイルオーバをユーザに確定するよう要求できます。この設定は、LifeKeeper の管理者のみが使用できます。オペレータとゲストには、この設定は表示されません。デフォルトでは、フェイルオーバはすべて、ユーザの操作を必要とせず自動実行されます。ただし、confirm failover フラグが設定されると、指定システムからフェイルオーバするには、以下のコマンドを実行して確定することが必要です。`lk_confirmso -y system` 。以下のコマンドを実行して、フェイルオーバをブロックできます。`lk_confirmso -n system` 。指定期間内にこれらのコマンドのいずれかが実行されない限り、システムは事前プログラミングされたデフォルト動作を実行します。 `/etc/default/LifeKeeper` ファイル内にある 2 つのフラグが、この自動動作を制御します。

        - º

CONFIRMSODEF

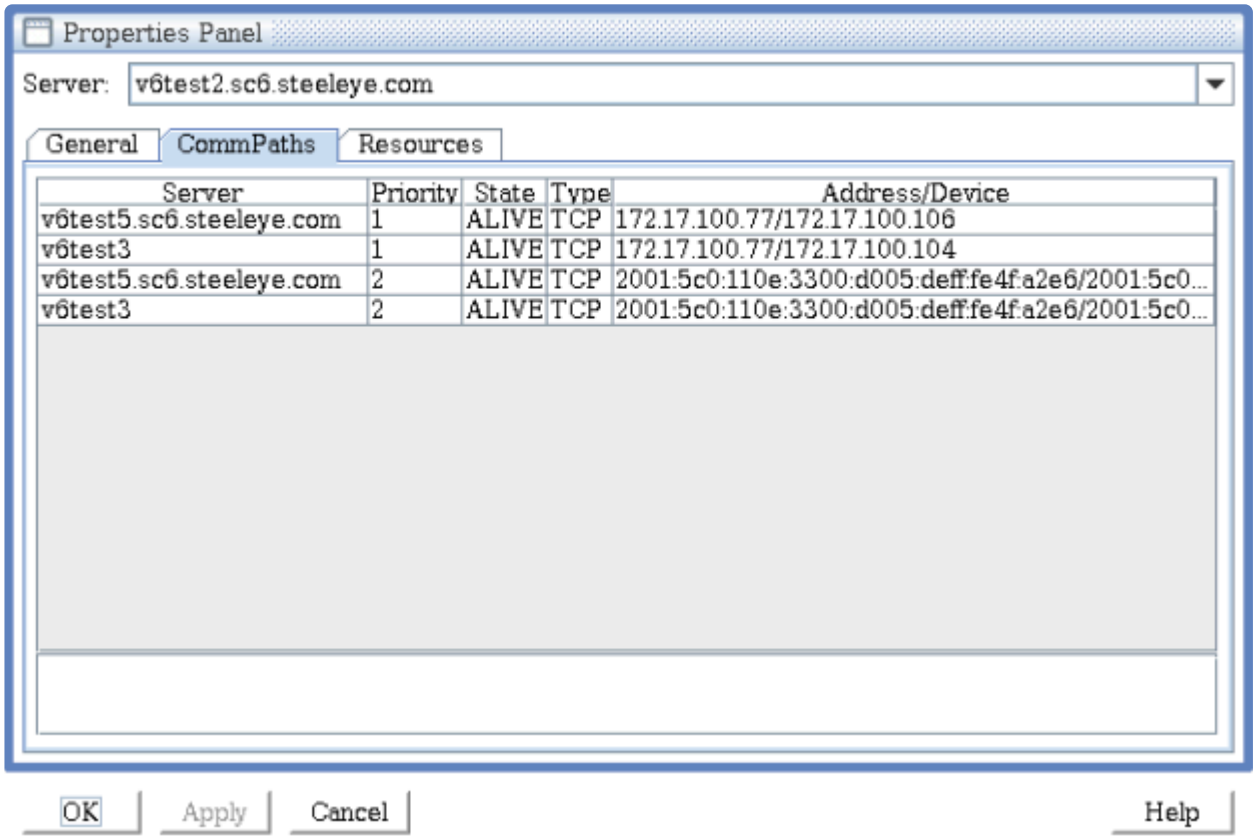
            これは、デフォルト動作を指定します。「0」に設定されている場合、デフォルト動作はフェイルオーバを実行します。「*1*」に設定されている場合、デフォルト動作はフェイルオーバをブロックします。

        - º

CONFIRMSOTO

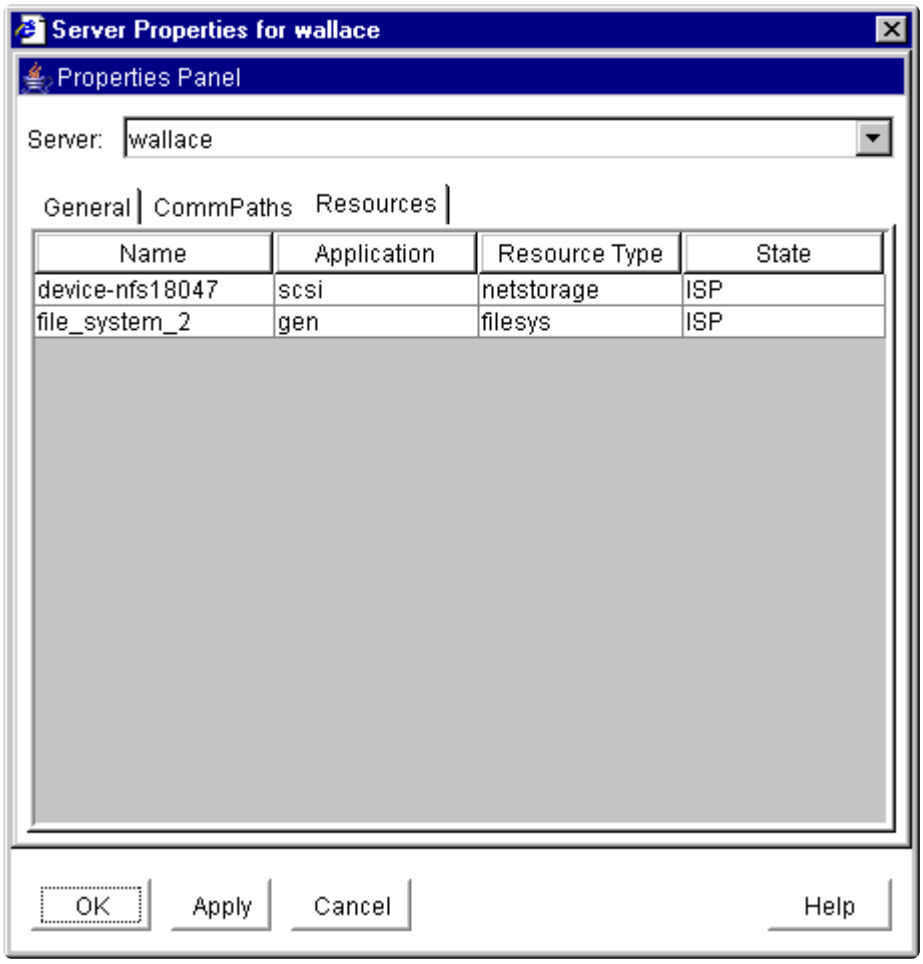
            これは秒単位で設定され、デフォルト動作を実行する前に LifeKeeper が待機する時間を示します。

# [CommPaths] タブ




- **Server** -LifeKeeper のクラスタ内で、コミュニケーションパスが接続している他のサーバのサーバ名。
- **Priority** - 2 台のサーバ間でコミュニケーションパスを使用する順序を定義する優先順位。1 が最高の優先順位で、99 が最低の優先順位です。
- **State** -LifeKeeper の設定データベース (LCD) のコミュニケーションパスの状態。コミュニケーションパスの状態は以下の値をとります。
  - *ALIVE* - 通常の動作をしています。
  - *DEAD* - 通常の動作をしていません。
  - *UNKNOWN* - 状態を特定できませんでした。GUI サーバが使用できない可能性があります。
- **Type -** リスト内のサーバと、[Server] フィールドに指定されたサーバとの間のコミュニケーションパスの種類。TCP (TCP/IP)、または TTY。
- **Address/Device** - コミュニケーションパスが使用する IP アドレスまたはデバイス名。
- **Comm Path Status** - LifeKeeper の設定データベース (LCD ) 内のコミュニケーションパスの状態に基づいて、GUI が判定したコミュニケーションパスのステータスの概要。以下に、コミュニケーションパスのステータスの値を示します。これらの値は、下のパネルの詳細テキストの下に表示されます。
  - *NORMAL* - すべてのコミュニケーションパスが通常の動作をしています。
  - *FAILED* - 指定サーバに対するすべてのコミュニケーションパスが動作していません。
  - *UNKNOWN* - コミュニケーションパスのステータスを特定できませんでした。GUI サーバが使用できない可能性があります。
  - *WARNING* - 指定サーバに対する 1 つ以上のコミュニケーションパスが動作していません。
  - *DEGRADED* - 指定サーバに対する 1 つ以上の冗長コミュニケーションパスが動作していません。
  - *NONE DEFINED* - コミュニケーションパスが定義されていません。

# [Resources] タブ




- **Name** -選択したサーバ上にあるリソースインスタンスのタグ名。
- **Application** - リソースタイプのアプリケーション名 (gen、scsi など)。
- **Resource Type** - サービスを提供するリソースタイプ、ハードウェアのクラス、ソフトウェアのクラス、またはシステムのエンティティのクラス (app、filesys、nfs、device、disk など)。
- **State** -リソースインスタンスの現在の状態。
  - *ISP* — ローカルでサービス起動中であり、保護されています。
  - *ISU* - ローカルでサービス起動中ですが、ローカルリカバリは試行されません。
  - OSF - サービス停止、障害。
  - *OSU* - サービス停止、障害なし。
  - *LLSTATE* - LifeKeeper の起動シーケンスの一部として実行されるリソース初期化プロセスにより、リソースの状態が適切に初期化されていません。この状態のリソースは、LifeKeeper で保護されていません。
  - *UNKNOWN* - リソースの状態を特定できませんでした。GUI サーバが使用できない可能性があります。

# 5.4.4.1.8. オペレータの作業

以下のトピックは、オペレータの権限を必要とする高度な作業です。

---

リソースを In Service にする

リソースを Out of Service にする

# 5.4.4.1.8.1. リソースを In Service にする

注記: LifeKeeper は、リソースを階層の最下位から最上位へ順に In Service にします。1 つの階層内のすべてのリソースを In Service にする場合は、階層内の最上位（親）リソースを選択してください。

1. 開始するには、以下の 5 つの方法があります。

- in service にするリソース / サーバの組み合わせのアイコンを右クリックします。リソースのコンテキストメニュー が表示されたら、 **[In Service]** をクリックします。

- in service にするグローバルリソースのアイコンを右クリックします。 **リソースのコンテキストメニュー** が表示されたら、 **[In Service]** をクリックします。ダイアログが表示されたら、in service にするリソースが存在するサーバを **[Server]** リストから選択し、 **[Next]** をクリックします。

- グローバルツールバー の **[In Service]** ボタンをクリックします。ダイアログが表示されたら、in service にするリソースが存在するサーバを **[Server]** リストから選択し、 **[Next]** をクリックします。次のダイアログで、in service にするリソースを 1 つ以上 **[Resouce(s)]** リストから選択し、 **[Next]** をもう一度クリックします。

- リソースのコンテキストツールバー が 表示される場合は、その **[In Service]** ボタンをクリックします。

- [Edit] メニュー の **[Resource]** をポイントし、 **[In Service]** をクリックします。ダイアログが表示されたら、in service にするリソースがあるサーバを **[Server]** リストから選択し、 **[Next]** をクリックします。次のダイアログで、in service にするリソースを 1 つ以上 [Resouce(s)] リストから選択し、 **[Next]** をもう一度クリックします。

2. 選択したサーバとリソースを In Service にすることを示すダイアログボックスが表示されます。親リソースの in service にせずに依存する子リソースを in service にしようとする場合、このダイアログには警告も表示されます。 **[In Service]** をクリックして、依存する子リソースと共にリソースを in service にしてください。

3. 出力パネル が有効の場合は、ダイアログが閉じ、リソースを In Service にするコマンドの結果が **出力パネル** に表示されます。出力パネルが無効の場合は、これらの結果を表示するダイアログが表示されたままになり、結果がすべて表示されたら **[Done]** をクリックします。in service になった追加の依存 (子) リソースが、ダイアログまたは **出力パネル** に表示されます。

4. リソースを in service にする動作で発生したエラーは、in service にするリソースがあるサーバの LifeKeeper ログに記録されます。

# 5.4.4.1.8.2. リソースを Out of Service にする

> 注記: LifeKeeper は、リソースを階層の最上位のリソースから下位のリソースへ、順に Out of Service にします。階層全体（子リソースを持つ親）を Out of Service にしたい場合は、最下位レベルのリソースを Out of Service にしてください。

1. 開始するには、以下の 4 つの方法があります。

   - Out of Service にするグローバルリソース、またはリソース / サーバの組み合わせのアイコンを右クリックします。リソースのコンテキストメニュー が表示されたら、 **[Out of Service]** をクリックします。

   - グローバルツールバー の [Out of Service] ボタンをクリックします。[Out of Service] ダイアログが表示されたら、 Out of Service にするリソースを 1 つ以上 [Resouce(s)] リストから選択し、 **[Next]** をクリックします。

   - リソースのコンテキストツールバー が表示される場合は、 **[Out of Service]** ボタンをクリックします。

   - [Edit] メニュー の **[ Resource]** をポイントし、 **[Out of Service]** をクリックします。 **[Out of Service]** ダイアログが表示されたら、 Out of Service にするリソースを 1 つ以上 [Resouce(s)] リストから選択し、 **[Next]** をクリックします。

2. 選択したリソースが Out of Service になることを示す **[Out of Service]** ダイアログボックスが表示されます。親リソースを Out of Service にせずに依存する子リソースを Out of Service にしようとする場合、このダイアログには警告も表示されます。 **[Out of Service]** をクリックして、次のダイアログボックスに進みます。

3. 出力パネル が有効の場合は、ダイアログが閉じ、リソースを Out of Service にするコマンドの結果が出力パネルに表示されます。出力パネルが無効の場合は、これらの結果を表示するダイアログが表示されたままになり、結果がすべて表示されたら **[Done]** をクリックします。

4. リソースを Out of Service にする動作で発生したエラーは、 Out of Service にするリソースが存在するサーバの LifeKeeper ログに記録されます。

# 5.4.4.1.9. 高度な作業

LCD

LCM

LifeKeeper API for Monitoring

# 5.4.4.1.9.1. LCD

## LifeKeeper 設定データベース

LifeKeeper 設定データベース (LCD) は、LifeKeeper が既知のすべてのリソースタイプについて、オブジェクト指向のリソース階層情報を管理し、リカバリ方向の情報を保存します。データは共有メモリにキャッシュされ、ファイルに保存されるので、システムの再起動後もデータが保持されます。LCD には、リカバリが必要なリソースインスタンスについての状態の情報、および特定の詳細情報もあります。

LCD のディレクトリ構造、保存されるデータタイプ、使用できるリソースタイプ、およびアプリケーションスクリプトの使用の詳細については、以下の関連トピックを参照してください。

### 関連トピック

- LCDI のコマンド
- LCD の設定データ
- LCD のディレクトリ構造
- LCD のリソースタイプ
- LifeKeeper のフラグ
- リソースのサブディレクトリ
- /opt/LifeKeeper の LCD のディレクトリ構造

# 5.4.4.1.9.1.1. LCDI のコマンド

**注：以下の文章では、独自のリカバリキットを定義してリソースを作成する方法を案内しています。既存のリカバリキットを使用する場合は、GUI を使用してリソースを作成してください。**

LifeKeeper には、アプリケーションのリソース階層を定義するためのメカニズムが 2 つ用意されています。

- LifeKeeper の GUI
- LifeKeeper 設定データベースのインターフェース (LCDI) コマンド

LCDI は LifeKeeper が提供するインターフェースコマンドのセットで、使用するアプリケーションのニーズに合わせてリソース階層の設定の作成とカスタマイズができます。アプリケーションが複数のリソース (例: 2 つ以上のファイルシステム) に依存する場合、コマンドインターフェースを使用します。

コマンドの詳細については、LCDI のマニュアルページを参照してください。このトピックでは、開発シナリオを示し、GUI とコマンドの両方の機能を使用してリソース階層を作成できる方法を説明します。

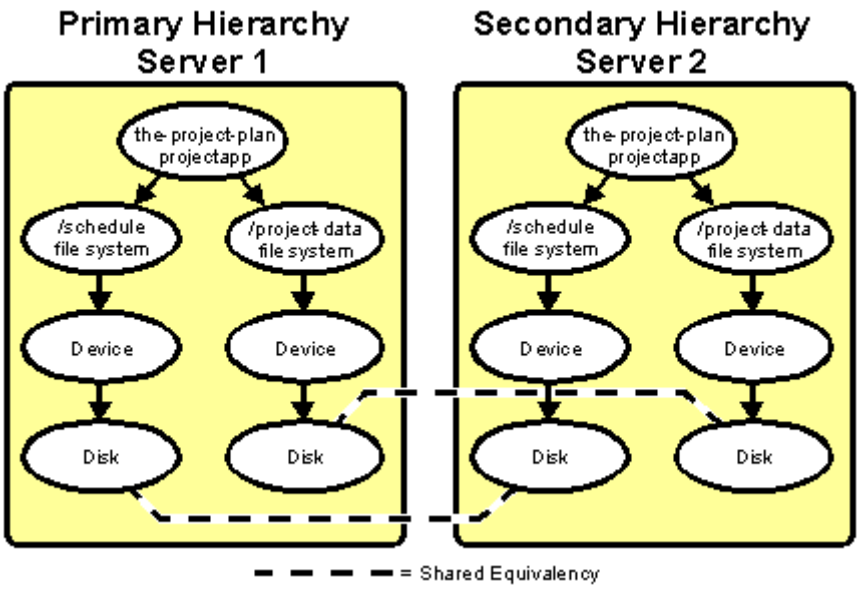
## シナリオの状況

アプリケーションの例である ProjectPlan は、サーバ 1 とサーバ 2 が共有する SCSI ファイルシステムにデータを保存しています。サーバ 1 が、アプリケーションのプライマリ階層にあります。アプリケーションには、 */project-data* と */schedule* の 2 つのファイルシステムがあります。階層定義の最初の手順では、依存関係を指定します。

このアプリケーションの例には、以下の依存関係があります。

- **共有ファイルシステム。** アプリケーションは、 */project-data* と */schedule* のファイルシステムに依存します。
- **SCSI ディスクサブシステム。** .次に、ファイルシステムは、SCSI ディスクサブシステム (デバイス、ディスク、およびホストアダプタのリソースを含む) に依存します。

結果として、階層を作成する作業は以下の図のようになります。




# 階層の定義

この例のアプリケーション階層を作成するために必要な作業を示します。

1. **ファイルシステムリソースの作成。** LifeKeeper の GUI には、ファイルシステムリソースを作成するメニューがあります。ファイルシステムリソース階層の作成 を参照してください。

   この定義作業の最後で、LCD では 2 つのファイルシステムリソースが以下のように定義されます。

| ID | タグ | サーバ |
|---|---|---|
| /project-data | project-data-on-Server1 | Server1 |
| /project-data | project-data-from-Server1 | Server2 |
| /schedule | schedule-on-Server1 | Server1 |
| /schedule | schedule-from-Server1 | Server2 |

   **注記:** LifeKeeper で使用されるタグ名には意味はありません。単なるラベルです。表内のタグ名は LifeKeeper のデフォルト値です。

2. **リソースの定義。** この例では、以下の項目を定義する必要があります。

| | |
|---|---|
| アプリケーション: | projectapp |
| リソースタイプ: | plan |
| インスタンス ID: | 1yrplan |

| タグ: | the-project-plan |
|---|---|

**注記:** LifeKeeper の GUI を使用して定義の大部分を作成できますが、この例の以降ではコマンドインターフェースの操作を説明します。

3. **ディレクトリの作成。** 各システムで以下のコマンドを使用して、ディレクトリ */opt/LifeKeeper/subsys* の下に必要なアプリケーションリカバリディレクトリを作成します。

```
mkdir -p /opt/LifeKeeper/subsys/projectapp/Resources/plan/actions
```

4. **アプリケーションの定義。** 以下のコマンドで、アプリケーション *projectapp* を作成します。

```
app_create -d Server1 -a projectapp

app_create -d Server2 -a projectapp
```

5. **リソースタイプの定義。** 以下のコマンドで、リソースタイプ *plan* を作成します。

```
typ_create -d Server1 -a projectapp -r plan

typ_create -d Server2 -a projectapp -r plan
```

6. **リカバリスクリプトのインストール。** restore と remove のスクリプトを各サーバの以下のディレクトリにコピーします。

   */opt/LifeKeeper/subsys/projectapp/Resources/plan/actions*

7. **インスタンスの定義。** 以下のコマンドで、リソースのインスタンスタイプが plan、ID が *1yrplan* のリソースを定義します。

```
ins_create -d Server1 -a projectapp -r plan -I\

AUTORES_ISP -t the-project-plan -i 1yrplan

ins_create -d Server2 -a projectapp -r plan -I\

SEC_ISP -t the-project-plan -i 1yrplan
```

   Server1 に作成したインスタンスの -I AUTORES_ISP 命令は、LifeKeeper の再起動時にそのリソースを自動的に in service にするように LifeKeeper に指示します。この例では、リソースの restore スクリプトが実行され、正常に実行された場合はリソースが ISP 状態になります。この動作は、ペアのリソースがすでにサービス起動している場合は実行されません。

   Server2 に作成したインスタンスの -I SEC_ISP 命令は、LifeKeeper の再起動時にそのリソースを in service にしないように LifeKeeper に指示します。その代わり、Server2 は Server1 上にあるリソースのバックアップとして機能し、プライマリのリソースまたはサーバに障害が発生したときにローカルリソースを in service にします。

8. **依存関係の定義。** 以下のコマンドは、アプリケーションとファイルシステムの依存関係を定義します。

```
dep_create -d Server1 -p the-project-plan -c project-data-on-System1

dep_create -d Server2 -p the-project-plan -c project-data-from-
Server1

dep_create -d Server1 -p the-project-plan -c schedule-on-Server1

dep_create -d Server2 -p the-project-plan -cschedule-from-Server1
```

9. **lcdsync の実行。** 以下の `lcdsync` コマンドを実行して、設定のコピーを更新するように LifeKeeper に通知します。

```
lcdsync -d Server1

lcdsync -d Server2
```

10. **リソースを In Service にする。** プライマリサーバで LifeKeeper の GUI にアクセスし、**[Edit] > [Resource] > [In-Service]** をクリックしてリソースを In Service にします。もしくはプライマリサーバー上で以下のコマンドを実行し In-Service にします。

```
perform_action -t the-project-plan -a restore
```

11. **共有関係の作成。** 以下のコマンドで各ノードで登録したリソースの共有関係を作成し、リソースを切り替えることができるようにします。

```
eqv_create -d Server1 -t the-project-plan -p 1 -S Server2 -o the-
project-plan -r 10 -e SHARED

eqv_create -d Server2 -t the-project-plan -p 10 -S Server1 -o the-
project-plan -r 1 -e SHARED
```

# 5.4.4.1.9.1.2. LCD の設定データ

LCD には、以下の関連するデータタイプが保存されます。

- ◦ 依存関係の情報
- ◦ リソースのステータス情報
- ◦ サーバ間のイクイバレンシ情報

## 依存関係の情報

定義した各リソースについて、LifeKeeper は依存関係のリスト、および依存物 (あるリソースに依存するリソース) のリストを保持します。詳細については、`LCDI_relationship (1M)` と `LCDI_instances (1M)` のマニュアルページを参照してください。

## リソースのステータス情報

LifeKeeper は、各リソースインスタンスのステータス情報をメモリに保持します。LCD が認識するリソースの状態 は、 **ISP** 、 **ISU** 、 **OSF** 、 **OSU** 、および **ILLSTATE** です。システムイベントが発生した場合、または管理者が特定の操作を行った場合に、リソースがある状態から別の状態に変化することがあります。リソースの状態が変化した場合、ステータスの変化が、ローカルサーバの LCD、およびそのリソースのダイアログサーバ上にあるデータベースに反映されます。

## サーバ間のイクイバレンシ情報

さまざまなサーバ上にある複数のリソース間に関係が存在することがあります。イクイバレンシ情報 とは、別のサーバ上にある 2 つのリソースが同一の物理エンティティであることを示す関係です。2 台のサーバが、イクイバレンシ情報の関係にある 1 つのリソースを持つ場合、LifeKeeper はその動作により、2 台のサーバ上にあるリソースの 1 つのみが同時に In Service、保護 (ISP) になるようにします。両方のサーバでそのリソースインスタンスを Out of Service ( **OSU** または **OSF** ) にすることができますが、データの整合性の理由から、同時に In Service にできるリソースは 1 つのみです。

SCSI バス上にある複数のディスクが、同等なリソースの一例です。SCSI のロック (または予約) メカニズムにより、任意の時点でディスクデバイスのロックを所有できるのは 1 台のサーバのみです。このロック所有機能により、同時に複数のサーバによる同一ディスクリソースへのアクセスが防止されます。

さらに、階層内の依存関係により、ファイルシステムのようにディスクに依存するリソースはすべて、同時に 1 台のサーバでのみ In Service になります。

# 5.4.4.1.9.1.3. LCD のディレクトリ構造

*/opt/LifeKeeper* の下にある主なサブディレクトリを示します。

**config** - LifeKeeper の設定ファイル。イクイバレンシ情報を含みます。 **bin** - LifeKeeper の実行可能プログラム。is_recoverable などがあります。詳細については、障害検出とリカバリのシナリオ を参照してください。 **subsys** - リソースとタイプ。LifeKeeper は、共有 SCSI ディスクサブシステムのリソースとタイプの定義を scsi で、汎用アプリケーションのメニュー機能を gen で提供します。アプリケーションのインターフェースを定義する場合は、subsys の下にディレクトリを作成してください。

- 

**events** - 警報イベント。詳細については、LifeKeeper の警報とリカバリ を参照してください。

*/opt/LifeKeeper* 内の LCD ディレクトリの構造については、*/opt/LifeKeeper* 内の　LCD ディレクトリの構造 のトピックを参照してください。

# 5.4.4.1.9.1.4. LCD のリソースタイプ

LCD は共有メモリ、および */opt/LifeKeeper* ディレクトリの両方に保持されます。ディレクトリ構造の図に示すように、subsys には、アプリケーションインターフェースの指定に使用できるアプリケーションリソースセットが 2 つあります。

- gen - 汎用アプリケーションとファイルシステムの情報
- scsi - SCSI に固有のリカバリ情報

これらのサブディレクトリについてはリソースのサブディレクトリ を参照してください。

# 5.4.4.1.9.1.5. LifeKeeper のフラグ

ステータスの詳細表示 の後部近くに、システムのフラグセットがあります。共通タイプは、プロセスのロックが動作を完了するまで他のプロセスを確実に待機させるために使用する LCD のロックフラグです。LCD のロックの標準フォーマットは以下のとおりです。

```
!action!processID!time!machine:id.
```

一般的な LCD のロックフラグの例を示します。

- ◦ **!action!02833!701236710!<servername>:filesys** .ファイルシステム階層を作成すると、このフォーマットでステータス表示にフラグが生成されます。 *filesys* の指定は、他のアプリケーションリソース階層では別のリソースタイプである場合も、一般的なアプリケーションやユーザ定義アプリケーションでは *app* である場合もあります。

- ◦ 他の代表的なフラグとして、 **!nofailover!machine** and **shutdown_switchover** があります。 **!nofailover!machine** フラグは、LifeKeeper が作成と削除を行う内部の一時フラグで、サーバのフェイルオーバを制御します。 **shutdown_switchover** フラグは、このサーバのシャットダウン方針がスイッチオーバに設定されたことを示し、サーバのシャットダウンによりスイッチオーバが発生します。使用可能なフラグの詳細については、`LCDI-flag(1M)` を参照してください。

# 5.4.4.1.9.1.6. リソースのサブディレクトリ

***scsi*** と ***gen*** のディレクトリはそれぞれ、リソースのサブディレクトリを持ちます。これらのディレクトリの内容は、LifeKeeper が提供するリソースタイプのリストです。

**scsi のリソースタイプ。** これらのリソースタイプは、*/opt/LifeKeeper/subsys/scsi/resources* ディレクトリにあります。実際の設定によっては、その他のディレクトリが存在する場合があります。

- **device** - ディスクパーティション、または仮想ディスクデバイス
- **disk** - 物理ディスク、または LUN
- **hostadp** - ホストアダプタ

**gen のリソースタイプ。** これらのリソースタイプは、*/opt/LifeKeeper/subsys/gen/resources* ディレクトリにあります。

- **filesys** - ファイルシステム
- **app** - 汎用またはユーザ定義のアプリケーションであり、他のリソースに依存することがある

各リソースタイプのディレクトリには、以下のものが 1 つ以上あります。

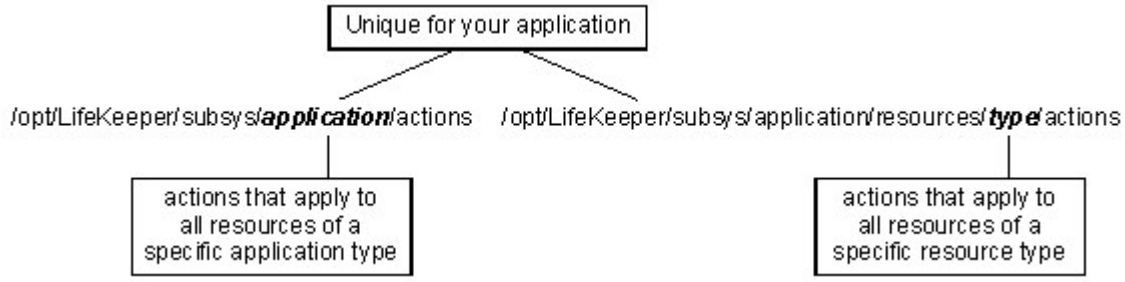
- **インスタンス。** このファイルは、LCD に保存されている、リソースインスタンスに関する恒久的な情報を反映します。このリソースタイプに関連付けられたリソースインスタンスの記述的な情報があります。

**警告:** インスタンスファイル (または LCD ファイル) を直接変更しないでください。リソースインスタンスの作成や操作を行うには、LifeKeeper の GUI の機能、または ins_create、ins_remove、ins_gettag、ins_setas、ins_setinfo、ins_setinit、ins_setstate、および ins_list の LifeKeeper の LCDI_instances コマンドのみを使用してください。これらのコマンドの詳細については、LCDI_instances (1M) のマニュアルページを参照してください。

- ***recovery*** 。このオプションのディレクトリには、障害が検出されたリソースのローカルリカバリの試行に使用されるプログラムがあります。recovery ディレクトリには、sendevent に渡されるイベントクラスに対応するディレクトリがあります。ディレクトリの名前は、sendevent プログラムに渡されるクラスパラメータ (-C) と一致する必要があります。 (LifeKeeper の警告とリカバリ を参照)。

各サブディレクトリに、アプリケーションは対応するイベントタイプを処理するリカバリプログラムを入れることができます。これらのプログラムの名前は、sendevent の -E パラメータで渡される文字列と一致する必要があります。このオプションのディレクトリは、複数のアプリケーションに使用されるように存在することはできません。

- ***actions*** 。このディレクトリには、特定のリソースタイプのリソースインスタンスについてのみ動作するリカバリ実行プログラムのセットがあります。使用するアプリケーションについて、アプリケーション内のすべてのリソースに適用する動作がある場合は、その動作を、***resource type*** ディレクトリではなく、アプリケーションディレクトリの ***actions*** サブディレクトリに入れてください。




リカバリ指示ソフトウェアが、リソースインスタンスの変更や復旧に使用されます。各リソースタイプの ***actions*** ディレクトリに、 **remove** と **restore** の 2 つの動作が必要です。

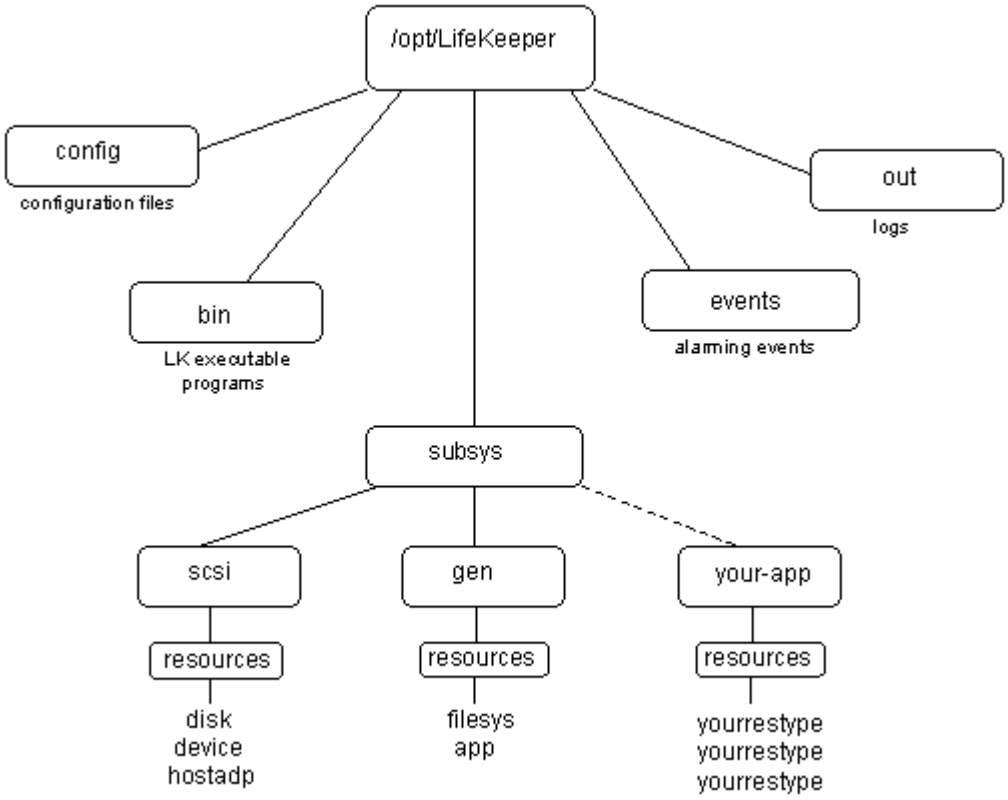
## リソースの動作

リソースタイプの ***actions*** ディレクトリには、特定のアプリケーションの動作を記述するプログラム (多くの場合は shell スクリプト) があります。各リソースタイプについて、restore と remove の 2 つの動作が必要です。

remove と restore のプログラムは、正反対の機能を実行する必要があります。つまり、相互の動作を元に戻す必要があります。これらのスクリプトは、絶対に手動で実行しないでください。これらのスクリプトは、LifeKeeper のリカバリ動作と制御のインターフェース (LRACI) の `perform_action` shell プログラムのみが実行する必要があります (`LRACI-perform_action (1M)` マニュアルページを参照)。

# 5.4.4.1.9.1.7. /opt/LifeKeeper の LCD のディレクトリ構造

以下の図に、 ***/opt/LifeKeeper*** のディレクトリ構造を示します。



# 5.4.4.1.9.2. LCM

The LifeKeeper Communications Manager (LCM) は、1 台以上の LifeKeeper サーバ上にあるプロセス間に信頼性の高い通信を提供します。このプロセスは、システム間の冗長コミュニケーションパスを使用できるので、1 つのコミュニケーションパスに障害が発生しても、LifeKeeperやそれが保護するリソースには障害が発生しません。LCM は、RS-232 (TTY) と TCP/IP の接続を含む多様な通信方法をサポートしています。

LCM は以下の機能を提供します。

- ◦ **LifeKeeper のハートビート。** 接続している他の LifeKeeper システムと定期的に通信して、他のシステムが動作を継続しているかどうかを判断します。LifeKeeper はハートビート信号がないことを認識することにより、他の方法では検出されないシステム全体の障害を検出できます。

- ◦ **管理サービス。** LifeKeeper の管理機能は、LCM の機能を使用してリモート管理を実行します。この機能は、シングルポイントの管理、設定の検証、および管理動作の正常性チェックに使用されます。

- ◦ **設定とステータスの通信。** LifeKeeper 設定データベース (LCD) は、リソースのステータス、可用性、および設定を LCM 機能経由で記録します。LCM の機能により、LCD はプライマリとセカンダリのシステム間で整合性のあるリソース情報を保持できます。

- ◦ **フェイルオーバリカバリ。** あるシステム上のリソースに障害が発生すると、LCM は LifeKeeper に、バックアップシステム上にリソースを復旧するように通知します。

LCM が提供する LifeKeeper のサービスに加えて、shell コマンドセットによりアプリケーションによる信頼性の高いシステム間通信が可能です。これらのコマンドとして、`snd_msg`, `rcv_msg`、`can_talk` などがあります。これらのコマンドの詳細については、`LCMI_mailboxes (1M)` のマニュアルページを参照してください。LCM はシステム上でリアルタイムプロセスとして動作し、システムのハートビートが送信されるなどの重要な通信が確実に実行されるようにします。

## 関連トピック

通信ステータスの情報

LifeKeeper の警報とリカバリ

# 5.4.4.1.9.2.1. 通信ステータスの情報

ステータス表示の通信ステータスの情報のセクションには、LifeKeeper が認識しているサーバとその現在の状態、および各コミュニケーションパスの情報がリストされます。

以下の例は、ステータスの簡略表示の通信ステータスのセクションのものです。

```
MACHINE NETWORK ADDRESSES/DEVICE STATE PRIO

tristan TCP 100.10.100.100/100.10.100.200 ALIVE 1

tristan TTY /dev/ttyS0 ALIVE --
```

詳細については、ステータスの詳細表示 とステータスの簡略表示 のトピックの「通信ステータスの情報」セクションを参照してください。

# 5.4.4.1.9.2.2. LifeKeeper の警報とリカバリ

LifeKeeper のエラー検出と通知は、イベント警報メカニズム sendevent をベースにしています。**sendevent** メカニズムの重要な概念は、独立したアプリケーションが重要なコンポーネントについて警報を受信できるように登録できることです。警報を開始する側のコンポーネントと受信する側のアプリケーションのいずれも、他のアプリケーションの存在を知るように変更する必要はありません。アプリケーションに固有のエラーが、 **sendevent** 機能経由で LifeKeeper のリカバリメカニズムをトリガできます。

このセクションでは、警報クラス、警報の処理、および警報ディレクトリのレイアウトを含む警報に関連するトピックを説明し、次に警報の概念を示す処理シナリオを示します。

## 警報クラス

*/opt/LifeKeeper/events* ディレクトリには、アラームクラスのセットがリストされます。これらのクラスは、イベントを生成するシステムの特定サブコンポーネントに対応します (例: *filesys* )。各警報クラスのサブディレクトリには、可能性のある警報のセットがあります (例: *badmount* 、 *diskfull* )。shell スクリプトまたはプログラムを適切なディレクトリに入れることで、これらの警報を受信するようにアプリケーションを登録できます。

LifeKeeper は基本的な警報通知機能を使用しています。この警報機能により、イベントについて登録されたすべてのアプリケーションで、該当する警報の発生時に sendevent により処理プログラムが非同期で実行されます。LifeKeeper が存在する場合、 **sendevent** プロセスははじめに、LifeKeeper のリソースオブジェクトがクラスとイベントを処理できるかどうかを判断します。LifeKeeper がクラス/イベントの一致を検出した場合、適切な復旧シナリオが実行されます。

**sendevent** 警報機能の追加スクリプトを定義することは任意です。LifeKeeper リソースを定義すると、LifeKeeper が基本的な警報機能を提供します。その詳細は、この章の処理シナリオで後述します。

注記: リソースインスタンスのローカルリカバリは、LifeKeeper の制御下にあるアプリケーションが、中断されたリソースサービスを、イベントが発生したシステムのエンドユーザに返そうとする試行です。サーバ間リカバリでは、アプリケーションはバックアップシステムに移行できます。この種のリカバリは、ローカルリカバリが失敗したか、ローカルリカバリが不可能である場合に試行されます。

## 警報の処理

LifeKeeper の注意が必要な可能性のあるイベントを検出するアプリケーションまたはプロセスは、**sendevent** プログラムを実行し、各エラークラス、エラー名、および障害のあるインスタンスの引数を渡すことにより、イベントを報告できます。必須の詳細、オプションのパラメータ、および構文については、sendevent (5) のマニュアルページを参照してください。

## 警報ディレクトリのレイアウト

*/opt/LifeKeeper/events* ディレクトリには、2 種類の内容があります。

- ◦ **LifeKeeper の指定クラス** 。LifeKeeper は、 *events* ディレクトリの下に *lifekeeper* と *filesys*

の 2 つの警報クラスを用意しています。警報イベントの例としては、 *diskfull* などがあります。警報クラスは、 **sendevent** コマンドの -C オプションで渡される文字列に対応し、警報イベントは -E オプションで渡される文字列に対応します。Lifekeeper の警報クラスは、LifeKeeper のサブシステム内のイベント報告用に内部的に使用されます。

- ◦ アプリケーションに固有のクラス 。特定のアプリケーションで警報クラスの定義が必要な場合、 *events* ディレクトリに他のサブディレクトリが追加されます。アプリケーションは shell スクリプトまたはバイナリプログラムをそのサブディレクトリに入れることで、これらの警報を受信するように登録します。これらのプログラムの名前は、属するアプリケーションパッケージの名前に由来します。

# 5.4.4.1.9.3. LifeKeeper API for Monitoring

## はじめに

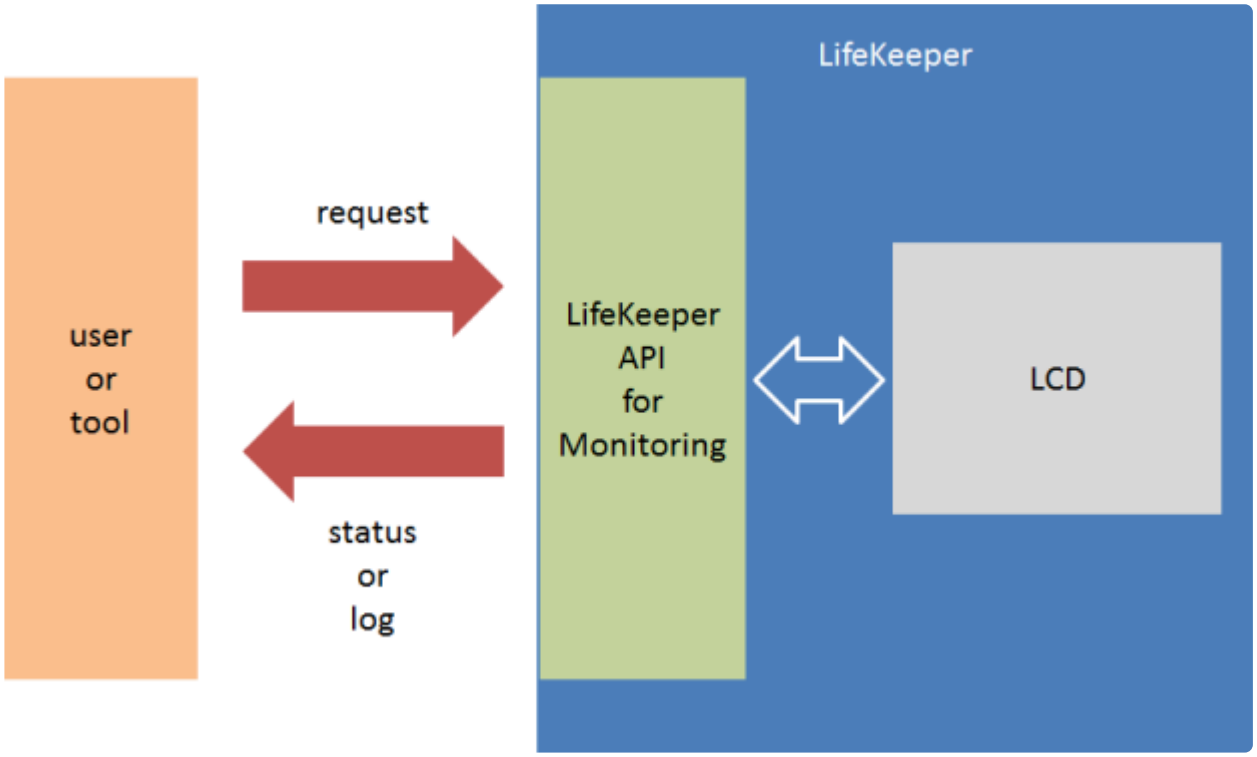
LifeKeeper API for Monitoring は、稼働中の LifeKeeper に対して問い合わせを行うことで、LifeKeeper ノードの稼働状況や保護しているリソースの状況を取得できます。

## 概要

本APIはlighttpd上でCGIとして動作し、lcdstatusコマンドで取得できる情報を、API経由で取得できるようにしたものです。本APIで取得できる情報の概要は以下のとおりです。

- ◦ LifeKeeper ノードの死活
- ◦ コミュニケーションパスの死活
- ◦ リソースの保護状況

これにより、ユーザは LifeKeeper サーバにログインせずとも異常の発生を知ることができます。しかし、具体的にどのような異常が発生しているのかを確認するためには、LifeKeeperGUI にログインしたり、LifeKeeper のログを確認したりする必要があります。




## 取得できる情報

LifeKeeper が稼働している１台のノードに対して、ユーザが問合せを行った場合、以下の情報が取得できます。なお、LifeKeeper は通常複数のサーバで構成されることが多いですが、取得できる情報は問合せを

行ったサーバのもののみです。

- ◦ ステータス
  - º 各サーバの稼働状況
    - ▪ノード名
    - ▪ 稼働状況（ALIVE/DEAD)
  - º コミュニケーションパスの稼働状況
    - ▪ ノード名
    - ▪ 稼働状況（ALIVE/DEAD）
    - ▪ アドレス/デバイス名
  - º リソースの保護状況
    - ▪ ノード名
    - ▪ タグ
    - ▪ 稼働状況（ALIVE/DEAD）
    - ▪ ステータス（ISP、OSU、OSF、…）
    - ▪ 依存関係
    - ▪ DRリソースのミラー情報(ステータスがISPの場合のみ取得可能)
      - ▪ タグ
      - ▪ Mirror status configuration (Sync, Pause, …)
      - ▪ Replication status (75%, 100%, …)
- ◦ ログ
  - º /var/log/lifekeeper.log ※ログファイルのパスが変更されている場合はサポートしない
    - ▪ 最大1000行分（データ出力フォーマットがHTMLの場合）
    - ▪ 全て（データ出力フォーマットがplain textの場合）
  - º /var/log/lifekeeper.err ※ログファイルのパスが変更されている場合はサポートしない
    - ▪ 最大1000行分（データ出力フォーマットがHTMLの場合）

- ▪ 全て（データ出力フォーマットがplain textの場合）

# 通信形式

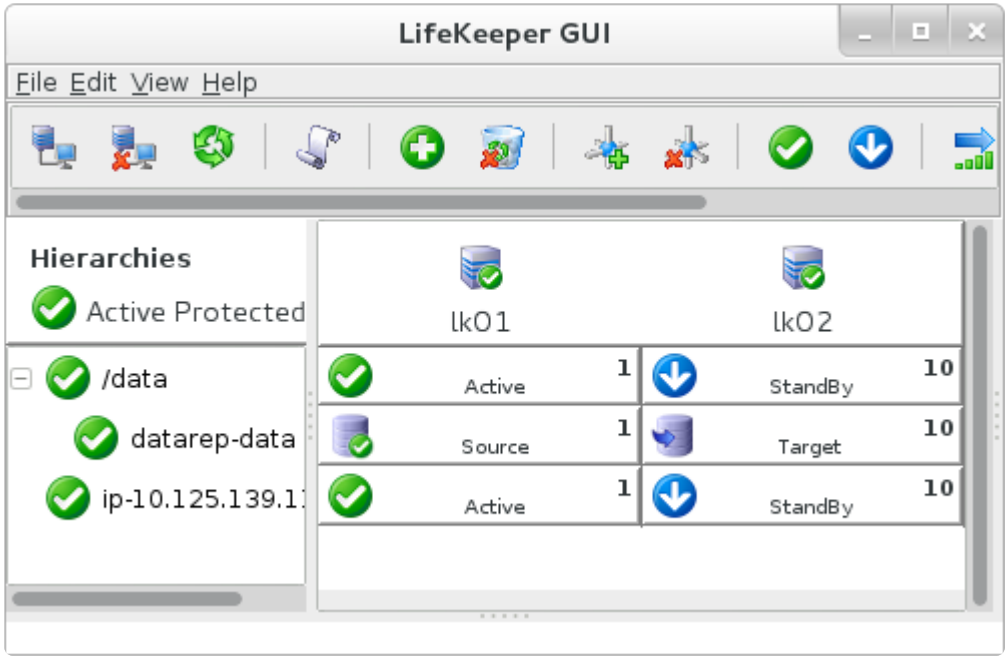
本 API で情報を取得するための通信は、HTTP で行います。
ユーザが対象サーバ上で動作する lighttpd 上で動作する CGI に対してHTTP GET リクエストを送信すると、前述の情報を取得することができます。

# データフォーマット

データフォーマットは以下の３つがあります。

- ◦ JSON
  - º 外部ツールにてステータスを解析するために利用されることを想定
  - º ステータスの確認が可能
  - º ログの出力はできない
- ◦ HTML
  - º ブラウザでの目視確認などに利用されることを想定
  - º ステータスの確認が可能
  - º ログは 1000 行分のみ取得可能
- ◦ plain text
  - º ステータスの確認はできない
  - º ログの取得ができる
  - º /var/log/lifekeeper.log や /var/log/lifekeeper.err の全ての内容を取得可能

LifeKeeper のクラスタ設定が以下の図の状態であるとき、取得できる JSON フォーマットと HTML フォーマットを示します。





```
{
        "resource": [
                {
                        "replication": {},
                        "child": [
                                {
                                        "tag": "datarep-data"
                                }
                        ],
                        "server": {
                                "status": "ISP",
                                "name": "lk01"
                                },
                        "tag": "/data"
                },
                {
                        "replication": {
                                "percent": "100%",
                                "mirror": "Fully Operational"
                        },
                        "child": [],
                        "server": {
                                "status": "ISP",
                                "name": "lk01"
                        },
                        "tag": "datarep-data"
                },
                {
                        "replication": {},
                        "child": [],
```

```
                        "server": {
                                "status": "ISP",
                                "name": "lk01"
                        },
                        "tag": "ip-10.125.139.118"
                }
        ],
        "compath": [
                {
                        "status": "ALIVE",
                        "server": [
                                {
                                        "name": "lk01",
                                        "term": "192.168.139.18"
                                },
                                {
                                        "name": "lk02",
                                        "term": "192.168.139.19"
                                }
                        ]
                },
                {
                        "status": "ALIVE",
                        "server": [
                                {
                                        "name": "lk01",
                                        "term": "172.20.139.18"
                                },
                                {
                                        "name": "lk02",
                                        "term": "172.20.139.19"
                                }
                        ]
                }
        ],
        "server": [
                {
                        "status": "ALIVE",
                        "name": "lk01"
                },
                {
                        "status": "ALIVE",
                        "name": "lk02"
                }
        ]
}
```

RESOURCEs

| tag | lk01 |
|---|---|
| /data | ISP |
| datarep-data | ISP |
| ip-10.125.139.118 | ISP |

DATA REPLICATIONs

| tag | nodes | mirror status | replication status |
|---|---|---|---|
| datarep-data | lk01 -> lk02 | Fully Operational | 100% |

COMMUNICATION PATHs

| communication path | status |
|---|---|
| 192.168.139.18/192.168.139.19 | ALIVE |
| 172.20.139.18/172.20.139.19 | ALIVE |

# 使用方法

## 本機能の有効化

本機能はデフォルトでは無効になっています。有効にする場合、/etc/default/LifeKeeper に以下の設定を行う必要があります。

```
LKAPI_MONITORING=true
```

なお、本機能の有効･無効を切り替える場合、LifeKeeper の再起動は不要です。

クラスタを構成している場合、設定の有効化は、問い合わせを行うLifeKeeperノードにのみ必要です。問い合わせを行わないノードは本機能の設定が無効であっても問題ありません。

## ポート番号

本機能は、デフォルトでは 779 番ポートを使用します。ポート番号をデフォルトから変更する場合、/etc/default/LifeKeeper に以下の設定を行う必要があります。

```
LKAPI_WEB_PORT=<ポート番号>
```

## 使用例

LifeKeeper が稼働しているサーバに以下のように問い合わせることで情報を取得できます。ここでは、curl を用いた例を示します。

```
curl http://<IPADDR>:779/Monitoring.cgi
```

このように引数を与えない場合、デフォルトでは JSON 形式にて現在のステータスが取得できます。HTML 形式でログを取得する場合、以下のように引数を指定します。

```
curl http://<IPADDR>:779/Monitoring.cgi?format=html&show=log
```

使用できる引数一覧は以下の通りです。

| 名前 | 説明 | 値 | 備考 |
|---|---|---|---|
| show | 取得したい情報を指定する | status, log, log-err | デフォルトは status |
| format | データフォーマットを指定する | json, html, plain | デフォルトでは json<br>show=log,log-err を指定した時に、format=json を指定した場合はエラーとなる |

## セキュリティ

本機能を用いると、情報取得のリクエストを行った全てのユーザが LifeKeeper のステータスを取得することができます。
そのため、ユーザのセキュリティ設定に基づき、ステータスを取得できるユーザの制限や、SSL の設定、情報の暗号化を行うことができます。

### Basic 認証

本APIで情報を取得するために、Basic認証を使用することができます。

lighttpd 設定ファイルの修正例は以下の通りです。赤字で示す箇所を追加、変更してください。修正後、設定を反映させるため、コマンド「/opt/LifeKeeper/sbin/sv restart steeleye-lighttpd」を実行し、lighttpd を再起動してください。

なお htpasswd コマンドが存在しない場合、ご使用のディストリビューション提供の httpd-tools をインストールしてください。

/opt/LifeKeeper/etc/lighttpd/lighttpd.conf

```
server.modules              = (
                                ...
                                "mod_auth", # uncommenting
```

/opt/LifeKeeper/lib64/steeleye-lighttpd/include_server_bind.pl

```
print qq/\$SERVER["socket"] == "$addr:$port" {\n/;
print qq/    server.document-root = "\/opt\/LifeKeeper\/api"\n/;
print qq/    auth.backend = "htpasswd"\n/;
print qq/    auth.backend.htpasswd.userfile = "\/opt\/LifeKeeper\/etc\/lighttp
d\/lighttpd.user.htpasswd"\n/;
print qq/    auth.require = ( "\/" =>\n/;
print qq/        (\n/;
```

```
print qq/                    "method"  => "basic",\n/;
print qq/                    "realm"   => "LifeKeeperAPI",\n/;
print qq/                    "require" => "valid-user"\n/;
print qq/        )\n/;
print qq/    )\n/;
print qq/ }\n/;
```

htpasswdファイル作成手順

```
htpasswd -c /opt/LifeKeeper/etc/lighttpd/lighttpd.user.htpasswd
<USERNAME>
```

## SSL/TLS 対応

本 API での通信において、SSL/TLS を用いることができます。lighttpd 設定ファイルの修正例は以下の通りです。修正後、設定を反映させるため、コマンド「/opt/LifeKeeper/sbin/sv restart steeleye-lighttpd」を実行し、lighttpd を再起動してください。

また、この対応により、本 API が使用するデフォルトポートの設定が不要となった場合は、次の「779 番ポートの無効化」を参照し無効化をおこなってください。

/opt/LifeKeeper/etc/lighttpd/include_ssl_port.pl

```
configAPI("0.0.0.0", 443);
if(socket($sock, AF_INET6, SOCK_STREAM, 0)) {
    configAPI("[::]", 443);
}
sub configAPI {
    my $addr = shift;
    my $port = shift;

    print qq/\$SERVER["socket"] == "$addr:$port" {\n/;
    print qq/    server.document-root = "\/opt\/LifeKeeper\/api"\n/;
    print qq/    ssl.engine  = "enable"\n/;
    print qq/    ssl.pemfile = "\/opt\/LifeKeeper\/etc\/certs\/LK4LinuxValidNo
de.pem"\n/;
    print qq/ }\n/;
}
```

### 779 番ポートの無効化

SSL/TLS の対応を行った場合、お客様の環境によっては本 API が従来使用していたポート（デフォルトでは 779 番ポート）の設定が不要となる場合があります。その場合は、以下の設定を削除、あるいはコメントアウトを行い、本 API が使用するデフォルトポートの設定を無効化してください。修正後、設定を反映させるため、コマンド「/opt/LifeKeeper/sbin/sv restart steeleye-lighttpd」を実行し、lighttpd を再起動して

ください。

/opt/LifeKeeper/lib64/steeleye-lighttpd/include_server_bind.pl

```
#lkapi_config("0.0.0.0", $lkapi_port);
#if(socket($sock, AF_INET6, SOCK_STREAM, 0)) {
#     lkapi_config("[::]", $lkapi_port);
#}
```

### IP アドレスアクセス制限

アクセス可能な IP アドレスを制限することができます。

IP アドレス 192.168.10.1 以外の接続を拒否する場合の lighttpd 設定ファイルの修正例は以下の通りです。修正後、設定を反映させるため、コマンド「/opt/LifeKeeper/sbin/sv restart steeleye-lighttpd」を実行し、lighttpd を再起動してください。

/opt/LifeKeeper/etc/lighttpd/conf.d/lkapi_user.conf

```
$HTTP["remoteip"] != "192.168.10.1" {
url.access-deny = ( "" )
}
```

## エラー

本 API 実行時にエラーが発生することがあります。その際、エラーの概要が出力されます。以下に JSON フォーマットでのエラー例を示します。 なお、ここでは lighttpd が返す HTTP ステータスコードについては触れません。

```
{
"error" : {
        id : -1,
        message : "Failed to get LCD status"
        }
}
```

出力フォーマットが HTML 版の場合も、同様の内容のメッセージを出力します。

# 5.4.4.1.10. メンテナンス作業

以下に、LifeKeeper のメンテナンス作業を示します。


- LifeKeeper の設定値の変更
- ファイルシステムの健全性の監視
- LifeKeeper が保護するシステムのメンテナンス
- リソース階層のメンテナンス
- フェイルオーバー後の復旧
- LifeKeeper の削除
- ファイアーウォールを使用した状態での LifeKeeper の実行
- ファイアーウォール経由での LifeKeeper GUI の実行
- リソース階層の転送

# 5.4.4.1.10.1. LifeKeeper の設定値の変更

LifeKeeper には、設定と設定を行った後に変更を要する場合がある値が多数あります。変更を要する場合がある値の例として、LifeKeeper サーバーの uname、コミュニケーションパスの IP アドレス、IP リソースのアドレス、タグ名などがあります。これらの値を変更するには、注意して以下の手順に従ってください。

1. 以下のコマンドを使用して、すべてのサーバーで LifeKeeper を停止してください。
   `$LKROOT/bin/lkcli stop`
   コミュニケーションパスを削除したり、サーバーからリソース階層を拡張解除したりする必要はありません。

2. LifeKeeper サーバーの uname を変更する場合は、`hostname(1)` コマンドを使用してサーバーのホスト名を変更してください。

3. 先に進む前に、新しいホスト名がクラスター内のすべてのサーバーで解決可能であることを確認してください。コミュニケーションパスのアドレスを変更する場合は、新しいアドレスが設定され、動作していることを確認してください (**ping** と **telnet** のユーティリティーをこの確認に使用可能)。

4. LifeKeeper の複数の値を変更する必要がある場合は、クラスター内の各サーバー上のファイルで、古い値と新しい値を以下のフォーマットで指定する必要があります。

   ```
   old_value1=new_value1
   ...
   old_value9=new_value9
   ```

5. クラスター内の **すべて** のサーバーで **lk_chg_value** コマンドを実行し、出力を確認して、予測しなかった変更内容による副作用が発生していないことを確認してください。

   a. 変更する値が複数ある場合は、以下のコマンドを実行してください。
      `$LKROOT/bin/lk_chg_value -Mvf file_name`

      注記: ***file_name*** は、手順 4 で作成したファイルの名前です。

   b. 変更する値が 1 つのみの場合は、以下のコマンドを実行してください。
      `$LKROOT/bin/lk_chg_value -Mvo old_value -n new_value`

      **-M** オプションは、LifeKeeper のすべてのファイルに対して変更を行わないことを指定します。

6. クラスター内のすべてのサーバーで、 **-M** オプションを指定せずに `lk_chg_value` コマンドを実行して、LifeKeeper のファイルを変更してください。

   a. 変更する値が複数ある場合は、以下のコマンドを実行してください。
      `$LKROOT/bin/lk_chg_value -vf file_name`

      注記: ***file_name*** は、手順 4 で作成したファイルの名前です。

b. 変更する値が 1 つのみの場合は、以下のコマンドを実行してください。
```
$LKROOT/bin/lk_chg_value -vo old_value -n new_value
```

7. 以下のコマンドを使用して、LifeKeeper を再起動してください。
```
$LKROOT/bin/lkcli start
```
LifeKeeper の GUI を使用してクラスターを表示する場合は、GUI を閉じてから再起動しなければならないことがあります。

## 例

*Server1* と *Server2* は、2 ノードクラスター内にある LifeKeeper サーバーの uname です。 *Server1* は、アドレス 172.17.100.48 のコミュニケーションパスを持ちます。 *Server2* はアドレス 172.17.100.220 の IP リソースを持ち、この IP リソースは *Server1* に拡張されています 。 *Server1* について以下の値を変更します。

| 値 | 旧 | 新 |
|---|---|---|
| uname | Server1 | Newserver1 |
| コミュニケーションパスのアドレス | 172.17.100.48 | 172.17.105.49 |
| IP リソースのアドレス | 172.17.100.220 | 172.17.100.221 |

これらの変更を行うには、以下の手順を実行する必要があります。

1. 以下のコマンドを使用して、 *Server1* と *Server2* の両方で LifeKeeperを停止してください。
```
$LKROOT/bin/lkcli stop
```

2. 以下のコマンドを使用して、 *Server1* の uname を *Newserver1* に変更してください。
```
hostname Newserver1
```

3. *Newserver1* と *Server2* の両方に、以下の内容を持つファイル */tmp/subs* を作成してください。

```
Server1=Newserver1
172.17.100.48=172.17.105.49
172.17.100.220=172.17.100.221
```

4. 両方のサーバーで以下のコマンドを実行し、出力を確認して、予測しなかった変更内容による副作用が発生していないことを確認してください。
```
$LKROOT/bin/lk_chg_value -Mvf /tmp/subs
```

5. 両方のサーバーで、 **-M** オプションを指定せずに **lk_chg_value** コマンドを実行して、LifeKeeper のファイルを変更してください。
```
$LKROOT/bin/lk_chg_value -vf /tmp/subs
```

6. 以下のコマンドを使用して、両方のサーバーで LifeKeeper を再起動してください。
```
$LKROOT/bin/lkcli start
```

注記: LifeKeeper のファイルを変更せずに **lk_chg_value** による変更内容を表示するには、 **-M** オプションを使用してください。 **lk_chg_value** が調べるファイルを表示するには、 **-v** を使用してください。タグ名を変更しない場合は、 **-T** オプションを使用してください。リソース ID を変更しない場合は、 **-I** オプションを使用してください。

# 5.4.4.1.10.2. ファイルシステムの健全性の監視

ファイルシステムの健全性の監視機能は、LifeKeeper が保護する、ファイルシステム依存のアプリケーションで障害が発生する原因となる条件を検出します。監視は、アクティブ / サービス中のリソース (つまりファイルシステム) でのみ実行されます。監視する条件は、以下の 2 つです。

- ファイルシステムがフル (またはほぼフル) の状態になる。
- ファイルシステムが不適切にマウント (またはアンマウント) された。

これら 2 つの条件のいずれかが検出されると、いくつかの動作のいずれかが実行されることがあります。

- 警告メッセージがログ記録され、システム管理者に電子メールを送信できる。
- リソースインスタンスのローカルリカバリを試行できる。
- リソースをバックアップサーバーにフェイルオーバできる。

## 条件の定義

### フル (もしくは空き容量が少なくなっている) のファイルシステム

ディスクフルの条件は検出できますが、ローカルリカバリまたはフェイルオーバの実行で解決することはできません。管理者の操作が必要です。デフォルトでは、メッセージがログ記録されます。さらに、通知スクリプトを使用して警告メッセージを管理者あてにメール送信するなど、任意の追加処理を行うことが可能です。このスクリプトはLifeKeeperインストールディレクトリ内の "events/filesys/diskfull/notify" にあります。

ディスクフルの条件に加えて、ディスクの使用率が指定のしきい値を超えている場合、警告メッセージを LifeKeeper のログに記録できます。

ディスクフルのしきい値は以下のとおりです。

```
FILESYSFULLERROR=95
```

警告を行うしきい値は以下のとおりです。

```
FILESYSFULLWARN=90
```

デフォルト値は上記のとおりそれぞれ 90% と 95% ですが、 */etc/default/LifeKeeper* ファイルの調整可能なパラメータを使用して設定できます。これら 2 つのしきい値の意味は以下のとおりです。

`FILESYSFULLWARN` – ファイルシステムがこの割合まで使用されると、メッセージが LifeKeeper のログに表示されます。

`FILESYSFULLERROR` – ファイルシステムがこの割合まで使用されると、メッセージが LifeKeeper のログ、およびシステムログに表示されます。ファイルシステムの通知スクリプトも呼び出されます。

### アンマウントされた、または不適切にマウントされたファイルシステム

LifeKeeper は */etc/mtab* ファイルをチェックして、LifeKeeper が保護するサービス中のファイルシステムが実際にマウントされているかどうかを調べます。さらに、 *filesys* のリソース情報フィールドに保存されているマウントオプションに対してマウントオプションをチェックし、階層の作成時に使用されていた元のマウントポジションと一致するかどうかを確認します。

ファイルシステムがアンマウントされているか、不適切にマウントされていることを検出した場合、ローカルリカバリが起動され、正しいマウントオプションを使用してファイルシステムの再マウントが試行されます。

再マウントに失敗した場合、条件を解消するためにフェイルオーバが試行されます。以下のリストに、フェイルオーバに進行する場合がある再マウントの障害の一般的な原因を示します。

- ファイルシステムが破損している (fsck の障害)
- マウントポイントディレクトリの作成失敗
- マウントポイントがビジー
- マウントの失敗
- LifeKeeper の内部エラー
- OS のアップグレード後に新しいマウントオプションを追加

## カーネルアップグレード後の新しいマウントオプション、または非推奨のマウントオプション

Linux カーネルをアップグレードすると、既存のファイルシステムマウントオプションの一部が新しいカーネルで非推奨になったり、新しいカーネルで、新しいデフォルトマウントオプションを既存のマウントに追加する可能性があります。例えば、「nobarrier」マウントオプションは RedHat Enterprise Linux 8 で廃止され、一部のカーネル バージョンでは「logbufs=8」や「logbsize=32k」などの新しいデフォルトマウントオプションが追加されました。

LifeKeeper で保護されたファイルシステムリソースに、カーネルのアップグレード後に廃止されるマウントオプションが含まれている場合、廃止されたオプションは、クラスター内のすべてのサーバーで LifeKeeper リソースのマウントオプションのリストから削除する必要があります。詳細については、LifeKeeper ファイルシステムリソースのマウントオプションの変更 セクションを参照してください。

カーネルのアップグレード後に、LifeKeeper で保護された既存のマウントポイントに、カーネルによって新しいデフォルトのマウントオプションが追加された場合、クラスター内のすべてのサーバーで LifeKeeper リソースのマウントオプションのリストに、新しいオプションを追加する必要があります。詳細については、LifeKeeper ファイルシステムリソースのマウントオプションの変更 セクションを参照してください。

# LifeKeeper ファイルシステムリソースのマウントオプションの変更

保護されたファイルシステムのマウント時に LifeKeeper が使用するマウントオプションを変更するには

1. LifeKeeper でファイルシステムリソースを Out-of-Service にします。 これにより、保護されたファイルシステムがアンマウントされます。

2. クラスター内の各サーバーのマウントオプションを更新します。

   a. LifeKeeper GUI を使用する場合

      i. マウントオプションを変更する各サーバーのファイルシステムリソースを右クリックし、[ Change Mount Options ] を選択します。

      ii.表示されるダイアログで、LifeKeeper がファイルシステムをマウントするときに使用するオプションのカンマ区切りリストを指定して、マウントオプションを変更します。 必要なマウントオプションを入力したら、[ Set Value ] をクリックします。 [ Finish ]をクリックして確認ダイアログを終了します。

   b. LifeKeeper コマンドラインインターフェイス を使用する場合

      i. `lkcli resource config fs --tag <tag> --mountopts "comma-separated list"`

      ii.このコマンドは、クラスター内の各サーバーで実行する必要があります。

3. LifeKeeper でファイルシステムリソースを In-Service にします。 これにより、保護されたファイルシステムがマウントされます。

# 5.4.4.1.10.3. LifeKeeper が保護するシステムのメンテナンス

LifeKeeper が保護するサーバーをシャットダウンしてメンテナンスを行うときには、メンテナンスの前にバックアップサーバーでシステムのリソース階層を in service にする必要があります。このプロセスにより、メンテナンスが必要なシステム上にある共有ディスクの動作がすべて停止します。

2 ノードの LifeKeeper クラスター（Server A とServer B で構成）の典型的な保守プロセスを以下に説明します。

1. **Server Aの監視を無効にします。** Server Aの/etc/default/LifeKeeperファイルのLKCHECKINTERVALの値を0に設定し、`killall lkcheck`を実行してください。この時、0に変更する前の LKCHECKINTERVAL の値を記録しておいてください。 ステップ8 で使用します。
2. **ServerAですべてのリソースをIn Service (起動)してください。** LifeKeeper GUI を使用してServer B で起動しているリソースをプライマリサーバーであるServer A で起動します。手順は、リソースを In Service にする を参照してください。
3. **Server BのLifeKeeperを停止します。** Server Bで `$LKROOT/bin/lkcli stop` を実行してLifeKeeperを停止します。 リソースは保護されていない状態になります。
4. **Server BのLinuxをシャットダウンして電源を落とします。** Server BのLinuxオペレーティングシステムをシャットダウンしてから、電源を停止します。
5. **Server Bのメンテナンスを実施します。** Server B で必要なメンテナンスを実施してください。
6. **Server Bの電源を入れてLinuxを起動します。** Server B の電源を入れてLinuxオペレーティングシステムを再起動してください。
7. **Server BのLifeKeeperを起動します。** Server Bで `$LKROOT/bin/lkcli start` コマンドを実行してLifeKeeperを起動します。LifeKeeperが起動するとリソースは保護された状態になります。
8. **Server A のリソース監視を有効にします。** Server Aの/etc/default/LifeKeeperファイルのLKCHECKINTERVALの値をステップ1で変更する前の値に変更し `killall lkcheck` を実行してください。
9. **Server Bの監視を無効にします。** Server Aの/etc/default/LifeKeeperファイルのLKCHECKINTERVALの値を0に変更し、`killall lkcheck`を実行してください。この時、 0 に変更する前のLKCHECKINTERVAL の値を記録しておいてください。 ステップ16 で使用します。
10. **Server Bですべてのリソースを In Service (起動)してください。** LifeKeeper GUI を使用してServer Aで起動しているリソースをServer Bで起動します。手順は、リソースを In Service にする を参照してください。
11. **Server AのLifeKeeperを停止します。** Server Bで `$LKROOT/bin/lkcli stop` を実行してLifeKeeperを停止します。 リソースは保護されていない状態になります。
12. **Server AのLinuxをシャットダウンして電源を落とします。** Server BのLinuxオペレーティングシステムをシャットダウンしてから、電源を停止します。
13. **Server Aのメンテナンスを実施します。** Server B で必要なメンテナンスを実施してください。
14. **Server Aの電源をいれてLinuxを起動します。** Server B の電源を入れてLinuxオペレーティングシステムを再起動してください。
15. **Server AのLifeKeeperを起動します。** Server Bで `$LKROOT/bin/lkcli start` コマンドを実行してLifeKeeperを起動します。LifeKeeperが起動するとリソースは保護された状態になります。
16. **Server B のリソース監視を有効にします。** Server Aの/etc/default/LifeKeeperファイルのLKCHECKINTERVALの値をステップ9で変更する前の値に変更し `killall lkcheck` を実行して

ください。

17. **Server A**ですべてのリソースを**In Service (起動)**してください。 プライマリサーバーであるServer A で、LifeKeeper GUI を使用してサーバ B で現在サービス中のリソース階層を起動します。詳細な手順は リソースを In Service にする を参照してください。

# 5.4.4.1.10.4. リソース階層のメンテナンス

システム上のその他すべての階層を LifeKeeper で保護した状態で、あるリソース階層のメンテナンスを実行できます。このためには、メンテナンスが必要な階層を Out of Service にし、メンテナンス作業の完了後にその階層を In Service にします。

リソース階層のメンテナンスを実行するには、以下の手順に従ってください。

1. **階層を Out of Service にしてください。** LifeKeeper の GUI を使用して、メンテナンスを実行する必要があるリソース階層をすべて Out of Service にします。詳細については、リソースを Out of Service にする を参照してください。

2. **メンテナンスを実行してください。** リソース階層で必要なメンテナンスを実行します。

3. **階層をリストアしてください。** LifeKeeper の GUI を使用して、リソース階層を In Service にします。詳細については、リソースを In Service にする を参照してください。

# 5.4.4.1.10.5. フェイルオーバ後の復旧

LifeKeeper がプライマリサーバー (*Server A*) からバックアップサーバー (*Server B*) にフェイルオーバリカバリを実行した後、以下の手順を実行してください。

1. **ログを確認してください。** *Server B* の LifeKeeper が *Server A* からフェイルオーバリカバリを実行すると、フェイルオーバ中にステータスメッセージが表示されます。

   実際の出力は、設定によって異なります。マウントやアンマウントの失敗に関するいくつかのメッセージが表示されることが予測されますが、これらのメッセージはリカバリの失敗を示唆しません。これらのメッセージ、および *Server B* でリソースを in service にするときに発生したエラーは、LifeKeeper のログに記録されます。

2. **メンテナンスを実行してください。** *Server A* の障害の原因を特定し、解決します。メンテナンスを実行するために、 *Server A* の電源をオフにすることが必要な場合があります。

3. **必要に応じて、 *Server A* を再起動してください。** メンテナンスが完了したら、必要に応じて *Server A* を再起動します。

4. **必要に応じて、LifeKeeper を開始してください。** *Server A* で LifeKeeper が動作していない場合は、コマンド `$LKROOT/bin/lkcli start` を使用して、LifeKeeper を開始してください。

5. **アプリケーションを *Server A* に戻してください。**都合のよい時点で、LifeKeeper の GUI を使用して、 *Server A* でアプリケーションを in service にします。詳細については、リソースを In Service にする を参照してください。 *Server A* でアプリケーションが **[Automatic Switchback]** に設定されている場合は、この手順は不要なことがあります。

# 5.4.4.1.10.6. LifeKeeper の削除

Linux 環境での LifeKeeper パッケージのアンインストールはコマンドラインから行います。以下のコマンドを実行してください。

```
/opt/LifeKeeper/bin/rmlk
```

これによりすへ゛ての LifeKeeper パッケージがアンインストールされ、システムから /opt/LifeKeeper ディレクトリが削除されます。このコマンドは、システム上で LifeKeeper が実行されているかどうかにかかわらず実行できます。
クラスタ内のすべてのノードで LifeKeeper を実行している状態でコマンドを実行すると、階層の拡張が解除され、すべてのコミュニケーションパスが削除されます。これにより、コマンドが実行されたノードに残っているものがすべて削除され、ノードを効果的にクラスタから削除できます。
LifeKeeper がコマンドの実行時に実行されていない場合、別のノードには削除されずに残るものが発生し、実行中のシステムに影響を与える可能性があります。クラスタ内の他のノードで次のコマンドを実行して（LifeKeeperを実行せずに）これらを削除すると、アンインストールが完了します。

```
/opt/LifeKeeper/bin/rmlk -l
```

このコマンドでは、次のオプションを使用できます。

- -l ：すべての LifeKeeper ライセンスを削除します。

**このコマンドは慎重に使用してくた゛さい。**

> **注意:** `lkbackup` コマンドによる LifeKeeper 構成の定期的なバックアップは、結果を自動的に /opt/LifeKeeper/config/ にアーカイブします。 `rmlk` コマンドは /opt/LifeKeeper ディレクトリを削除するため、コマンドを実行する前にアーカイブをバックアップすることを推奨します。

# 5.4.4.1.10.7. ファイアーウォールを使用した状態での LifeKeeper の実行

以下のネットワークアクセス要件を満たす場合、LifeKeeper for Linux は、同一サーバー上にファイアーウォールを設定した状態で実行できます。

> **注記:** ファイアーウォールの設定を変更する場合や無効にする場合は、ご使用のディストリビューションのマニュアルに従い設定してください。

## LifeKeeper のコミュニケーションパス

コミュニケーションパスは、特定の IP アドレスを使用して、LifeKeeper クラスター内にあるサーバーペアの間に設定されます。TCP ポート 7365 は、作成時にデフォルトで各通信のリモート側により使用されますが、通信の開始側の TCP ポートは任意です。推奨方法は、そのシステムが既知のコミュニケーションパスでローカルとリモートの IP アドレスの各指定ペアについて、受信と送信の両方のトラフィックを許可するように、各 LifeKeeper サーバーにファイアーウォールを設定することです。

## LifeKeeper GUI の接続

LifeKeeper GUI は、デフォルトの初期接続ポートであるポート 81 と 82 を含めて、特定の TCP ポートを多数使用します。また、LifeKeeper GUI は、ポート 1024 以降をオブジェクトの送受信に使用するリモートメソッド呼び出し (RMI)  も使用します。これらすべてのポートが、各 LifeKeeper サーバーのファイアーウォールで、少なくとも GUI クライアントが動作する外部システムに対して開いている必要があります。

## LifeKeeper の IP アドレスリソース

IP アドレスに関連するアプリケーションにアクセスする必要があるクライアントシステムから、LifeKeeper の階層にある IP アドレスリソースにアクセスできるように、ファイアーウォールを設定する必要があります。IP アドレスリソースは LifeKeeper クラスター内のあるサーバーから別のサーバーに移動できるので、すべての LifeKeeper サーバー上のファイアーウォールを適切に設定する必要があります。

また、LifeKeeper は、ブロードキャスト ping のテストを使用して、IP アドレスリソースの健全性を定期的にチェックします。このテストでは、仮想 IP アドレスからブロードキャスト ping パケットを送信し、ローカルサブネット上の他のいずれかのシステムが最初に応答するまで待ちます。このテストが失敗しないようにするには、各 LifeKeeper サーバー上のファイアーウォールが以下のタイプのネットワーク動作を許可するように設定する必要があります。

- 仮想 IP アドレスからのインターネット制御メッセージプロトコル (ICMP) パケットの送信 (アクティブな LifeKeeper サーバーがブロードキャスト ping を送信できる)
- 仮想 IP アドレスからの ICMP パケットの受信 (他の LifeKeeper サーバーがブロードキャスト ping を受信できる)

- 任意のローカルアドレスからの ICMP 応答パケットの送信 (他の LifeKeeper サーバーがブロードキャスト ping に応答できる)
- 仮想 IP アドレスでの ICMP 応答パケットの受信 (アクティブな LifeKeeper サーバーがブロードキャスト ping への応答を受信できる)

# LifeKeeper Data Replication

LifeKeeper Data Replication を使用する場合は、複製に ndb を使用する任意のポートへのアクセスを許可するように、ファイアーウォールを設定する必要があります。nbd が使用するポートは、以下の式で計算できます。

```
10001 + <mirror number> + <256 * i>
```

**i** は 0 から始まり、使用されていないポート番号が計算されるまで加算されます。 */etc/services* に定義されているポート、 `netstat -an --inet --inet6` の出力に含まれるポート、または LifeKeeper Data Replication の他のリソースが使用中としてすでに定義されているポートは、使用中です。

**例:** LifeKeeper Data Replication リソースのミラー番号が 0 である場合、式は当初、使用するポートを 10001 として計算しますが、この番号は、一部の Linux ディストリビューションでは SCP 設定ポートとして */etc/services* に定義されています。この場合、 ***i*** が 1 だけ増分されてポート番号 10257 が得られます。この番号は、これらの Linux ディストリビューションの */etc/services* には定義されていません。

## その他ノード間通信

LifeKeeper の各サーバーは、ポート 778 の SSL 接続を使用して通信を行います。このポートは、/etc/default/LifeKeeper 内の設定変数 API_SSL_PORT を使用して変更できます。

# 5.4.4.1.10.8. ファイアーウォール経由での LifeKeeper GUI の実行

場合によっては、LifeKeeper クラスターが会社のファイアーウォール内に配置され、管理者はファイアーウォールの外側にあるリモートシステムから LifeKeeper GUI を実行します。

LifeKeeper は、GUI のサーバーとクライアントとの通信にリモートメソッド呼び出し (RMI) を使用します。RMI クライアントは、それぞれの方向に通信を確立できる必要があります。RMI クライアントは動的ポートを使用するので、クライアントには推奨ポートを使用できません。

解決法としては、以下のように ssh を使用して、ファイアーウォールを通過する方法があります。

1. 社内の IT 部門が、ファイアーウォール内にアクセスするために十分にセキュリティの高い shell ポートを社内ファイアーウォールに開けていることを確認します。多くの場合、IT 部門がアクセスを許可するマシンは、実際にはクラスター内のマシンではなく、そこからクラスター内にアクセスできる中間マシンです。このマシンは、Unix または Linux が動作するマシンである必要があります。
2. 中間マシンと LifeKeeper サーバーの両方が、sshd (Secure Shell デーモン) を実行していること、および X11 ポート転送が有効になっていること (これは通常、 *etc/ssh/sshd_config* の `X11Forwarding yes' 行にある) を確認してください。不明の場合は、IT 部門に依頼してください。
3. X の Unix クライアントから以下のコマンドを使用して、中間マシンにトンネルを作成します。

   ```
   ssh -X -C <intermediate machine>
   ```

   **-C** は「トラフィックの圧縮」を意味し、低速のインターネットリンクから受信する場合に役立つことが多々あります。

4. 中間マシンから以下のコマンドを使用して、LifeKeeper サーバーにトンネルを作成します。

   ```
   ssh -X <LifeKeeper server>
   ```

   中間マシンは LifeKeeper サーバーとの間にかなり高い帯域幅の接続をもつはずなので、このコマンドには圧縮は不要です。

5. すべての操作が良好に完了した場合、以下のコマンドを実行してください。

   ```
   echo $DISPLAY
   ```

   「*localhost:10.0*」のような値に設定されます。値が設定されない場合、X11 の転送がいずれかの sshd 設定ファイルで無効になっています。

6. 以下のコマンドを実行して、LifeKeeper サーバーから単純な *xterm* をポップアップ表示できることを確認してください。

   ```
   /usr/X11R6/bin/xterm
   ```

7. *xterm* が表示された場合、以下のコマンドを使用して、LifeKeeper で **lkGUIapp** を実行できます。

```
/opt/LifeKeeper/bin/lkGUIapp
```

8. GUI コンソールが表示されるまで待ってください。Java は多くのグラフィックス動作を使用し、低速リンクで伝播するには時間がかかります (圧縮している場合でも)。しかし、最終的には GUI コンソールが表示されます。

# 5.4.4.1.10.9. リソース階層の転送

LifeKeeper サーバで定期的なメンテナンスやその他の作業を実行する必要がある場合、LifeKeeper の GUI を使用して In Service のリソースを別のサーバに移動できます。 サーバ *A* から サーバ *B* に In Service のリソース階層を転送するには、GUI を使用して サーバ *B* でリソース階層を in service にします。 サーバ *A* のリソースがすべて、対応するバックアップサービスで In Service になるまで、操作を繰り返します。手順については、リソースを In Service にする を参照してください。

サーバ *A* のリソースがすべて、バックアップサーバでアクティブになった後、アプリケーションの処理に影響を与えることなく、 サーバ *A* をシャットダウンできます。ただし、メンテナンスの期間中、クラスタ内にあるサーバ数によっては、リソースが LifeKeeper で保護されないことがあります。

# 5.4.4.1.11. テクニカルノート

お使いの LifeKeeper 環境に関する設定と動作上の問題に関する以下のテクニカルノートをお読みになることを強く推奨します。

## LifeKeeper の機能

| 項目 | 説明 |
|---|---|
| ライセンス | LifeKeeper を使用するには、各サーバーに一意の実行時ライセンスキーが必要です。これは物理サーバーおよび仮想サーバーの両方に適用されます。ライセンスキーは、LifeKeeper Core ソフトウェア、および LifeKeeper リカバリーキットの各パッケージにそれぞれ必要です。インストールスクリプトで、サーバーの Host ID を取得して表示するラインセンスユーティリティパッケージをインストールします。ライセンスがインストールされた後、ユーティリティが、使用可能な Entitlement ID、または Host ID (Entitlement ID が使用できない場合) を返します。ホスト ID およびソフトウェアに付属のアクティベーション ID を使用して **SIOS Technology Corp.** の **Web サイト** からライセンスキーを取得してください。 |
| 大型クラスターのサポート | LifeKeeper は、最大 32 台のサーバーを持つ大型クラスターの設定をサポートします。ただし、LifeKeeper 以外の多くの要因が、クラスター内でサポートされるサーバーの台数に影響することがあります。この要因として、ストレージの相互接続、オペレーティングシステム、ストレージソフトウェアの制限などがあります。サポートされる最大クラスターサイズを調べるには、ベンダ固有のハードウェアとソフトウェアの設定情報を参照してください。 |
| 国際化とローカライズ | LifeKeeper for Linux v5.2 以降は、リソース名とタグ名でのワイド / マルチバイト文字の使用をサポートしていますが、ネイティブの言語メッセージサポートは含まれていません。Java のプロパティファイルのロケール固有バージョンを作成することにより、LifeKeeper の GUI をローカライズできますが、現在フルにローカライズされているのは英語バージョンのみです。ただし、LifeKeeper の GUI に表示される多くのメッセージは LifeKeeper Core から来ているので、GUI のローカライズは、ユーザーにとって、Core ソフトウェアがフルにローカライズされるまでの単なる部分的な解決法です。<br>追加情報については、制限または既知の問題 の「**言語環境の影響**」も参照してください。 |
| **LifeKeeper の MIB ファイル** | LifeKeeper は、LifeKeeper クラスター内で発生するイベントを記述する SNMP トラップを送出するように設定できます。この機能の設定に関する詳細については、lk_configsnmp(8) のマニュアルページを参照してください。LifeKeeper のトラップを記述する MIB ファイルは、*/opt/LifeKeeper/include/LifeKeeper-MIB.txt* に記載されています。 |
| **Watchdog** | LifeKeeper は、Watchdog 機能をサポートしています。この機能は、SIOS Technology Corp. により Red Hat EL 6 + softdog 、および Red Hat EL 7 + softdog でテスト済みです。 |

| | |
|---|---|
| **STONITH** | LifeKeeper は、STONITH 機能をサポートしています。この機能は、SIOS Technology Corp. により IBM x3550 x86_64 アーキテクチャ上の SLES 11、および RHEL 5.5 の 64-ビットでテスト済みです。 |
| **XFS ファイルシステム** | XFS ファイルシステムは、ファイルシステムのチェックと修正に fsck ユーティリティを使用しません。その代わりに、ログの再生をマウントに依存します。整合性の問題についての懸念がある場合は、システム管理者がファイルシステムを out of service にしてシステムをアンマウントし、`xfs_check(8)` と `xfs_repair(8)` を実行して問題を解決する必要があります。 |
| **IPv6** | SIOS は、ip コマンドの使用に移行し、`ifconfig` コマンドを使用しなくなりました (詳細については IPv6 の既知の問題 を参照)。 |

## チューニング

| 項目 | 説明 |
|---|---|
| **IPC セマフォと IPC 共有メモリ** | LifeKeeper には、プロセス間通信 (IPC) セマフォと IPC 共有メモリが必要です。以下の Linux カーネルオプションの Red Hat のデフォルト値は、*/usr/include/linux/sem.h* にあり、LifeKeeper の多数の設定をサポートするのに十分な値です。<br><br>**注記:** 以下は各オプションにおいてLifeKeeperに必要な最小値です。これらの値は LifeKeeper 専用であり、他のアプリケーションセマフォ要件に基づいて調整する必要がありますが、LifeKeeper の最小値を下回ってはなりません。<br><br>オプション 最小値 **Red Hat 7** のデフォルト値<br><br>SEMOPM 14 32<br><br>SEMMNI 25 128<br><br>SEMMSL 20 32000 |
| システムファイルテーブル | LifeKeeper がバックアップシステムに正常にフェイルオーバーするためには、システムリソースが使用可能である必要があります。例えば、システムファイルテーブルがフルの場合、LifeKeeper が新しいプロセスを開始してリカバリーを実行することができない可能性があります。エンタプライズパッチを持つカーネル (LifeKeeper がサポートするものを含む) では、file-max、つまりシステムで開いているファイルの最大数は、デフォルトでシステムメモリサイズの 1/10 に設定されます。これは、LifeKeeper の多数の設定をサポートするのに十分な値です。file-max 値をデフォルト値よりも低く設定すると、予期しない LifeKeeper の障害が発生することがあります。<br><br>file-max 値は、以下のコマンドで取得できます。<br><br>`cat /proc/sys/fs/file-nr` |

| | |
|---|---|
| | このコマンドは、3 つの値を返します。1 番目の値はファイルテーブルのエントリのこれまでの最大値 (システムがこれまでに検出した最大値)、2 番目の値は現在のファイルテーブルのエントリ数、3 番目の値は file-max 値です。<br><br>file-max を調整するには、/etc/sysctl.conf の「fs.file-max」値を追加 (または変更) し (フォーマットについては sysctl.conf(5) を参照)、<br><br>`sysctl -p`<br><br>次にこのファイルを実行して、システムを更新します。/etc/sysctl.conf の値は、再起動後も保持されます。 |

# LifeKeeper の動作

| 項目 | 説明 |
|---|---|
| **カーネルデバッガ (kdb)** | LifeKeeper が保護するサーバーでカーネルデバッガ ( **kdb** ) を使用する前に、そのサーバーで LifeKeeper をシャットダウンするか、LifeKeeper が保護するリソースをバックアップサーバーにスイッチオーバーする必要があります。LifeKeeper の SCSI 予約デーモン ( **lkscsid** ) を有効にした状態で (デフォルトで有効)、 **kdb** を使用すると、予期しないパニックが発生することがあります。 |
| **ロックしている共有デバイスでのシステムパニック** | LifeKeeper はロックを使用して、共有 SCSI バス上にある他のサーバーがアクセスしないように共有データを保護します。他のサーバーがデバイスをロックしたことにより LifeKeeper がデバイスにアクセスできない場合、致命的なエラーが発生し、即座に対処する必要があります。対処しない場合、データが破損するおそれがあります。この条件が検出された場合、LifeKeeper はシステムにパニックを発生させる機能を有効にします。<br><br>共有デバイスが予約された状態で、LifeKeeper が「`lkcli stop -f`」の様にリソースを停止させずに LifeKeeper のデーモンのみを停止した場合、他のサーバーがリソースを復旧するときに LifeKeeper のロックメカニズムによりカーネルのパニックがトリガされることがあります。この方法で LifeKeeper を停止する前に、リソースをすべて out-of-service にする必要があります。 |
| **nolock オプション** | NFS 領域でロックを使用するアプリケーションを使用し、LifeKeeper の推奨するマウントオプションを使用する場合は、nolock を追加する必要があります。 `rw,nolock,bg,hard,tcp,nfsvers=3,timeo=600,rsize=32768,wsize=32768,actimeo=0` |
| **Out-of-Service 階層の復旧** | LifeKeeper サーバーの障害発生後のリカバリーの一部として、障害が発生したサーバーに設定されているリソース階層のうち、障害発生時にいずれかのサーバーで *in-service* ではないものは、その時点で優先順位が最高の *alive* のサーバーで復旧されます。これは、障害が発生したサーバー、復旧中のサーバー、クラスター内の他のサーバーを含め、 *out of service* の階層が最後にどこで in service だったかには |

| | |
|---|---|
| | 無関係です。 |
| **Linux ファイアーウォールとの共存** | LifeKeeper はコミュニケーションパス、GUI、IP、およびデータレプリケーションに特定のポートを使用することに注意してください。Linux のファイアーウォール機能を使用する場合、LifeKeeper が使用している特定のポートを開放する必要があります。詳細については、ファイアーウォールを使用した状態での LifeKeeper の実行 を参照してください。<br><br>ファイアーウォールを無効･変更するには、OS ディストリビューションのドキュメントを参照してください。 |
| **Linuxセキュリティソフトとの共存** | SELinux が enforcing mode で実行されている場合、LifeKeeper は動作しません。また、SAP環境で必要な場合を除き、permissive mode を使用することはお勧めできません。 詳細については、 SELinux を無効にする を参照してください。AppArmor (このセキュリティモデルを使用するディストリビューションの場合) が有効になっている可能性があります。 |
| **Suid マウントオプション** | suid マウントオプションは、 *root* としてマウントするときのデフォルトであり、マウントコマンドにより */etc/mtab* に書き込まれることはありません。LifeKeeper 環境では、suid マウントオプションは不要です。 |

## サーバーの設定

| 項目 | 説明 |
|---|---|
| **BIOS のアップデート** | 使用可能な最新の BIOS を常にすべての LifeKeeper サーバーにインストールする必要があります。 |

# LifeKeeper 8.2.0 以降の GUI 要件

LifeKeeper GUI クライアントでユーザーを正常に認証するには、64 ビットバージョンの PAM 関連のパッケージがすべて必要です。

# [Confirm Failover] と [Block Resource Failover] の設定

以下の説明、例、および考慮事項をよく読んで理解してから、お使いの LifeKeeper 環境で **[Confirm Failover]** または **[Block Resource Failover]** を設定してください。これらの設定は、コマンドライン、または LifeKeeper の GUI の **[Properties]** パネルから使用できます。

## Confirm Failover On:

定義 – システム *A* から システム *B* へのフェイルオーバーの手動確認を有効にします ( システム *A* はプロパティが [Properties] パネル に表示されるサーバーで、 システム *B* はチェックボックスの左にあるシステム)。あるシステムでこのオプションをオンに設定した場合、障害発生が検出されたシステムについて LifeKeeper がフェイルオーバーリカバリーを実行するには、システム管理者による手動確認が必要になり

ます。

フェイルオーバーを確認するには、**lk_confirmso** コマンドを使用してください。デフォルトでは、このコマンドを実行するまで管理者には 10 分の猶予時間があります。この時間は、 */etc/default/LifeKeeper* の **CONFIRMSOTO** 設定で変更できます。管理者が 10 分以内に **lk_confirmso** コマンドを実行しない場合、フェイルオーバーは続行されるか、ブロックされます。デフォルトでは、フェイルオーバーが続行されます。この動作は、 */etc/default/LifeKeeper* の **COMFIRMSODEF** 設定で変更できます。

例：自動フェイルオーバーをすべてブロックする場合は、 **[Properties]** パネルの **[Confirm Failover On]** オプションを設定し、さらに **CONFIRMSODEF** を **1** (フェイルオーバーをブロック)、 **CONFIRMSOTO** を **0** (フェイルオーバー動作が決定されるまで待機しない) に設定してください。

この設定を選択するタイミング:

この設定は、設定に冗長ハートビートコミュニケーションパスを含まない多くのディザスタリカバリー、その他の WAN 設定で使用されます。

あるサーバーで **[Properties]** ページを開き、 **[Confirm Failover]** フラグ をオンに設定するサーバーを選択してください。

## Block Resource Failover On:

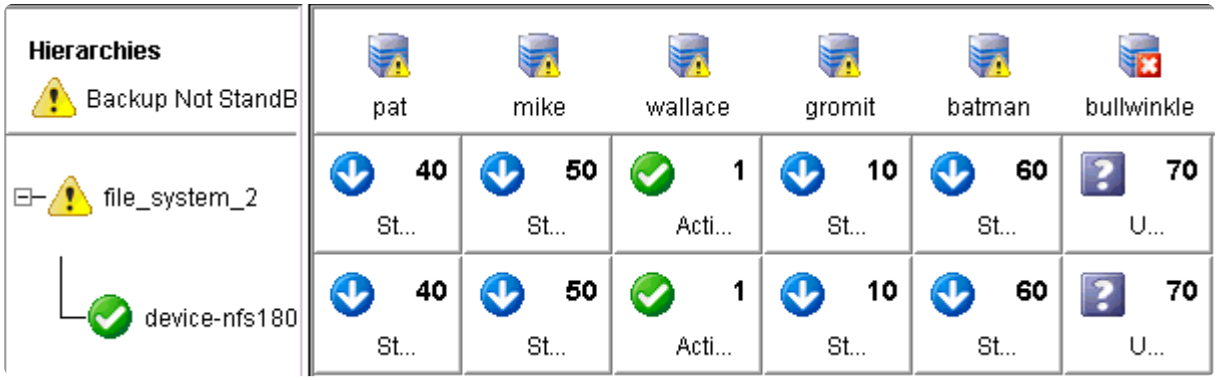
定義 - デフォルトでは、リソースのすべての障害について復旧イベントが発生し、ローカルシステムの障害リソースの復旧が試行されます。ローカルリカバリーが失敗した場合、または有効になっていない場合は、リソースが定義されている、優先順位が次に最も高いシステムに、LifeKeeper がローカル履歴を転送します。ただし、宛先として指定したシステムでこの設定を選択している場合、リソース障害に起因するリソースの転送はすべてブロックされます。

この設定が有効の場合、以下のメッセージがログに記録されます。

Local recovery failure, failover blocked, MANUAL INTERVENTION REQUIRED

# 5.4.4.2. クラスターの例

## 拡張したマルチクラスターの例



# 5.4.4.3. ダイアログ

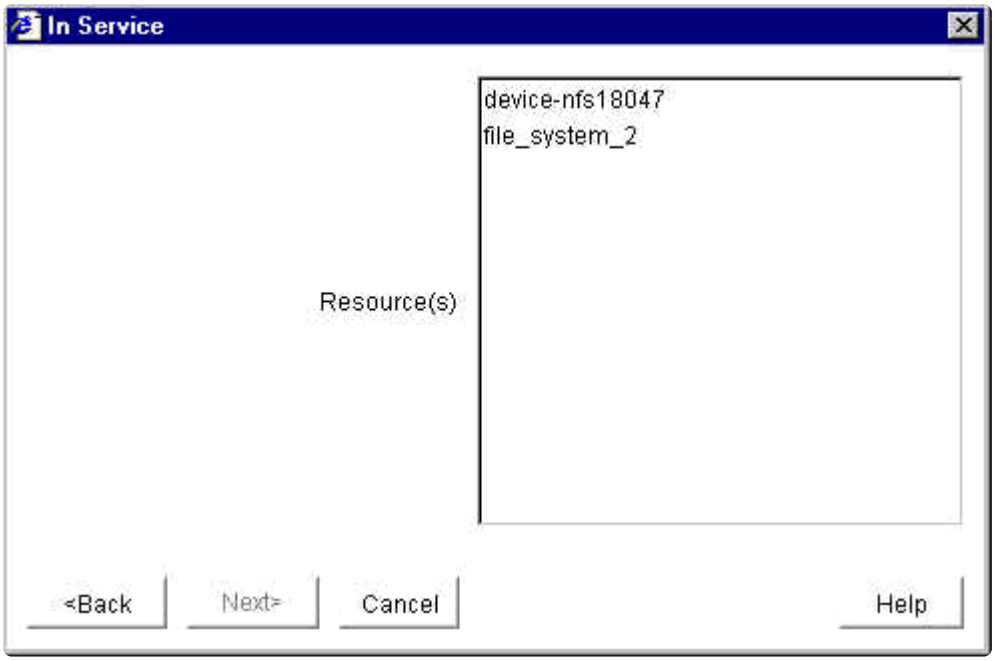
- [In Service] ダイアログ
- [Out-of-Service Resource Properties] - [EquivalenciesResource] ダイアログ
- [Password] ダイアログ
- [Resource Properties] - [Equivalencies]
- [Resource Properties] - [General]
- [Resource Properties] - [Relations]
- [Server Properties] - [Commpath]
- [Server Properties] - [General]
- [Server Properties] - [Resource]

# [In Service] ダイアログ

**Select a Server** - 最初のダイアログのドロップダウンリストボックスには、LifeKeeper クラスタ内にあるサーバ名があります。 in service にするリソースインスタンスが存在する サーバ を選択してください。 **[Next]** ボタンをクリックして、次のダイアログに進んでください。

**注記:** 右側のペインにあるサーバ固有のリソースを右クリックして [In Service] の作業を開始した場合、このダイアログと次のダイアログは表示されません。これは、 in service にするサーバとリソースを指定したからです。

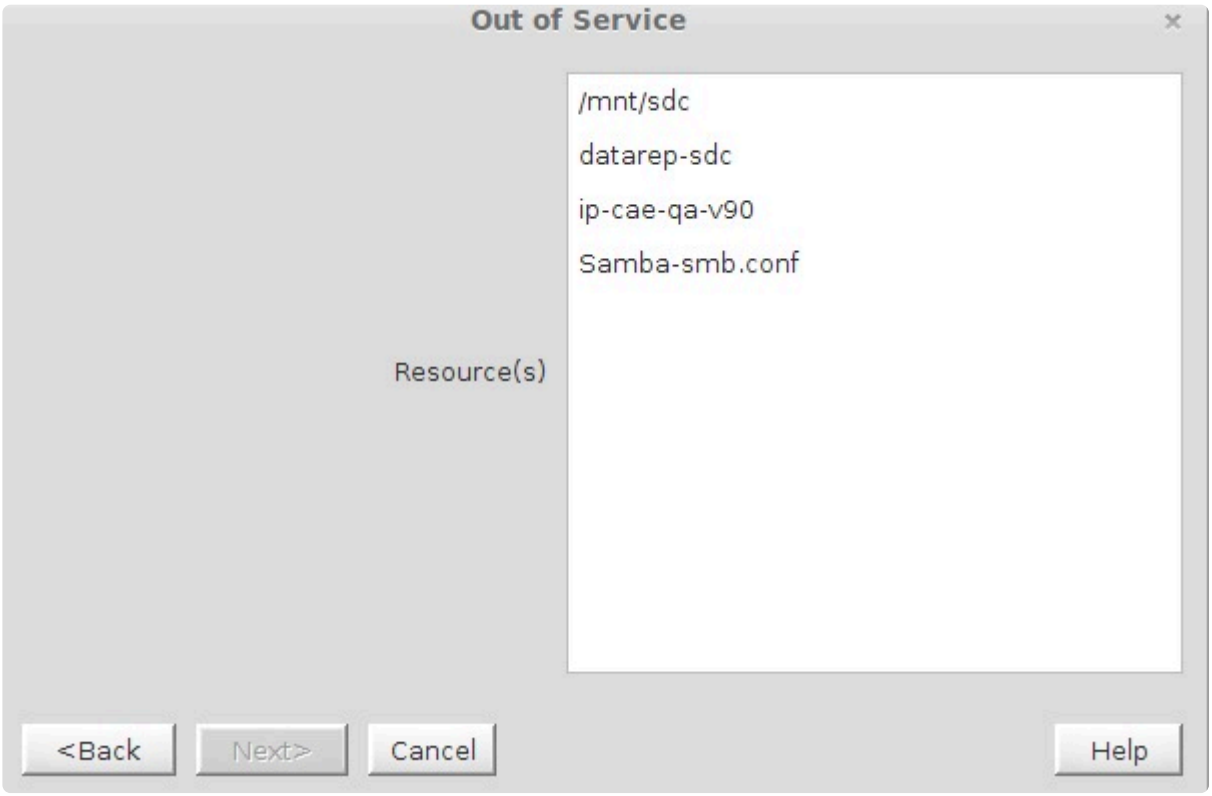
**Select a Resource** - 2 番目のダイアログのドロップダウンリストボックスには、前のダイアログで選択したサーバ上で使用できるすべてのリソースの名前があります。 in service にするリソースを選択してください。




**注記:** 左側のペインにあるグローバルリソースを右クリックして [In Service] の作業を開始した場合、このダイアログは表示されません。これは、 in service にするリソースを指定したからです。

# [Out-of-Service Resource] ダイアログ

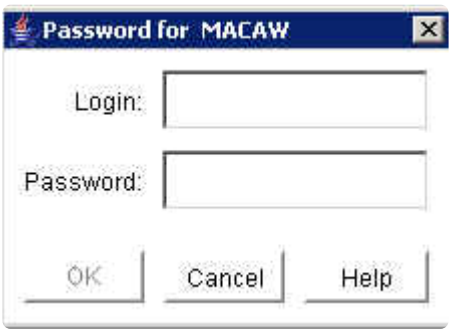
**Select a Resource** - このダイアログのリストには、LifeKeeper クラスタ内で in service になっているすべてのリソースの名前があります。out of service にするリソースを選択してください。




注記 : 左側のペインにある in service のグローバルリソース、または右側のペインにある in service のサーバに固有のリソースインスタンスを右クリックして [Out-of-Service] の作業を開始した場合、このダイアログは表示されません。これは、out of service にするリソースを指定したからです。

# [Password] ダイアログ

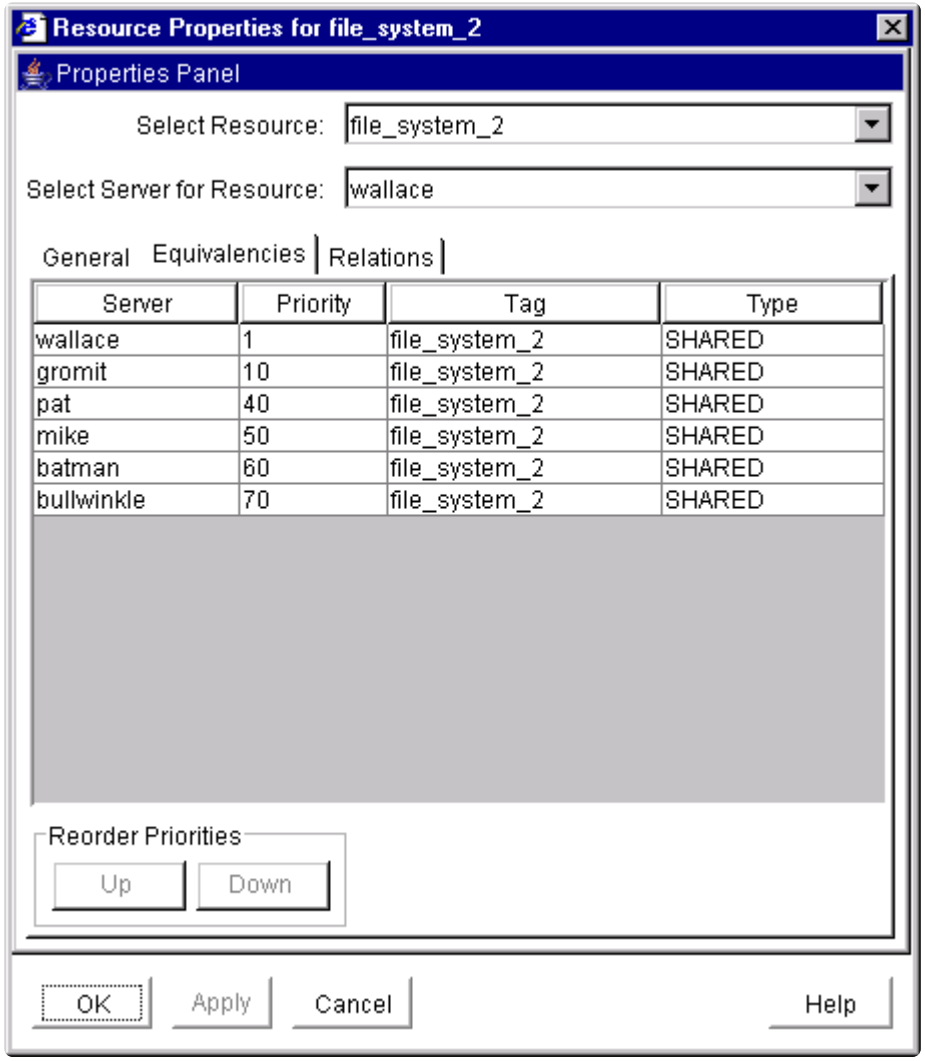
**注記:** このダイアログは、[Cluster Connect] ダイアログ に最初に入力したログイン名またはパスワードが正しくない場合に表示されます。



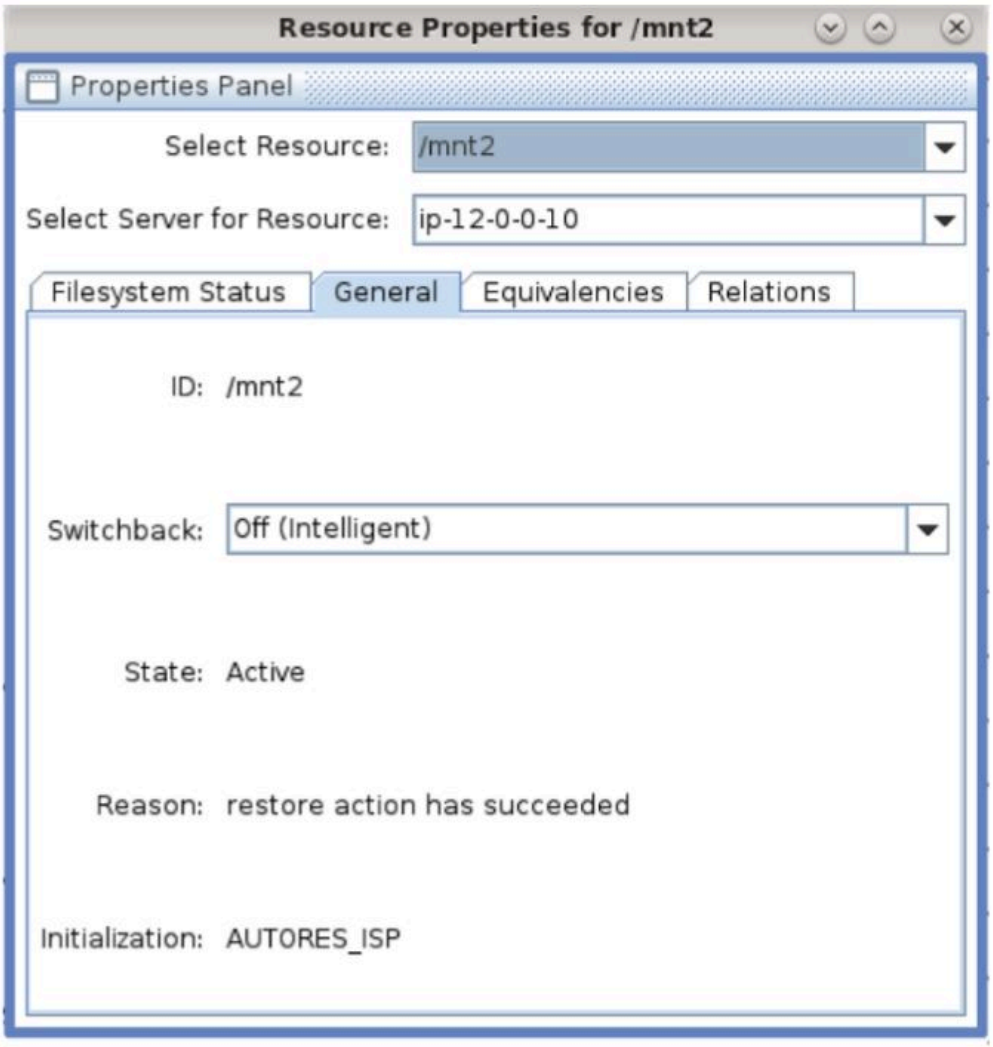

**Login** - 指定サーバに LifeKeeper 認証情報を持つユーザのログイン名。

**Password** - サーバで指定ログインを認証するパスワード。

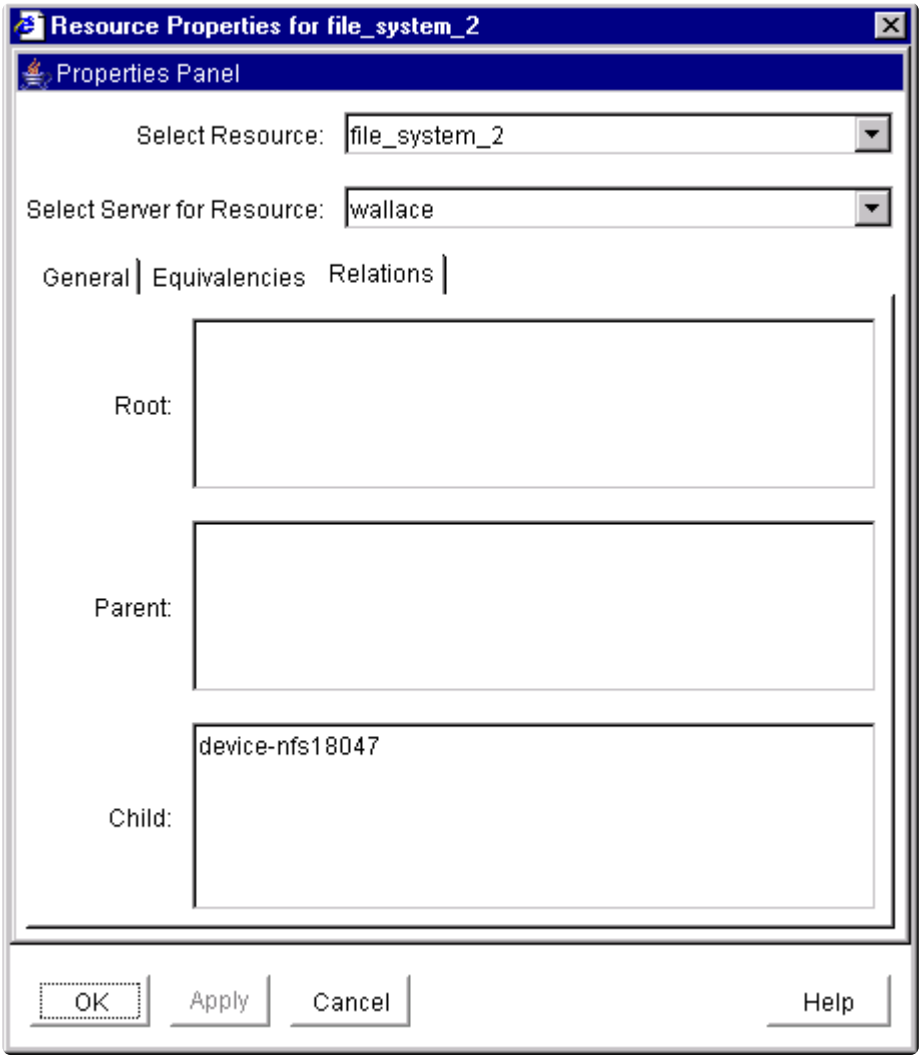
# [Resource Properties] - [Equivalencies]



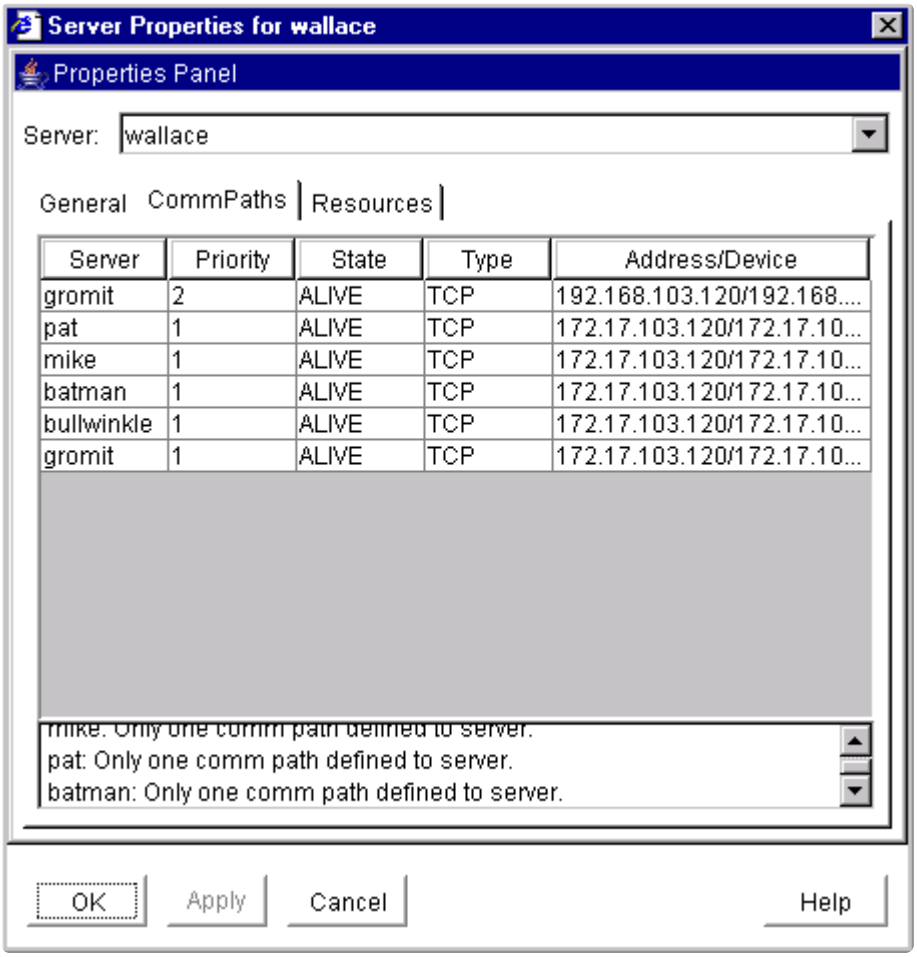
# [Resource Properties] - [General]



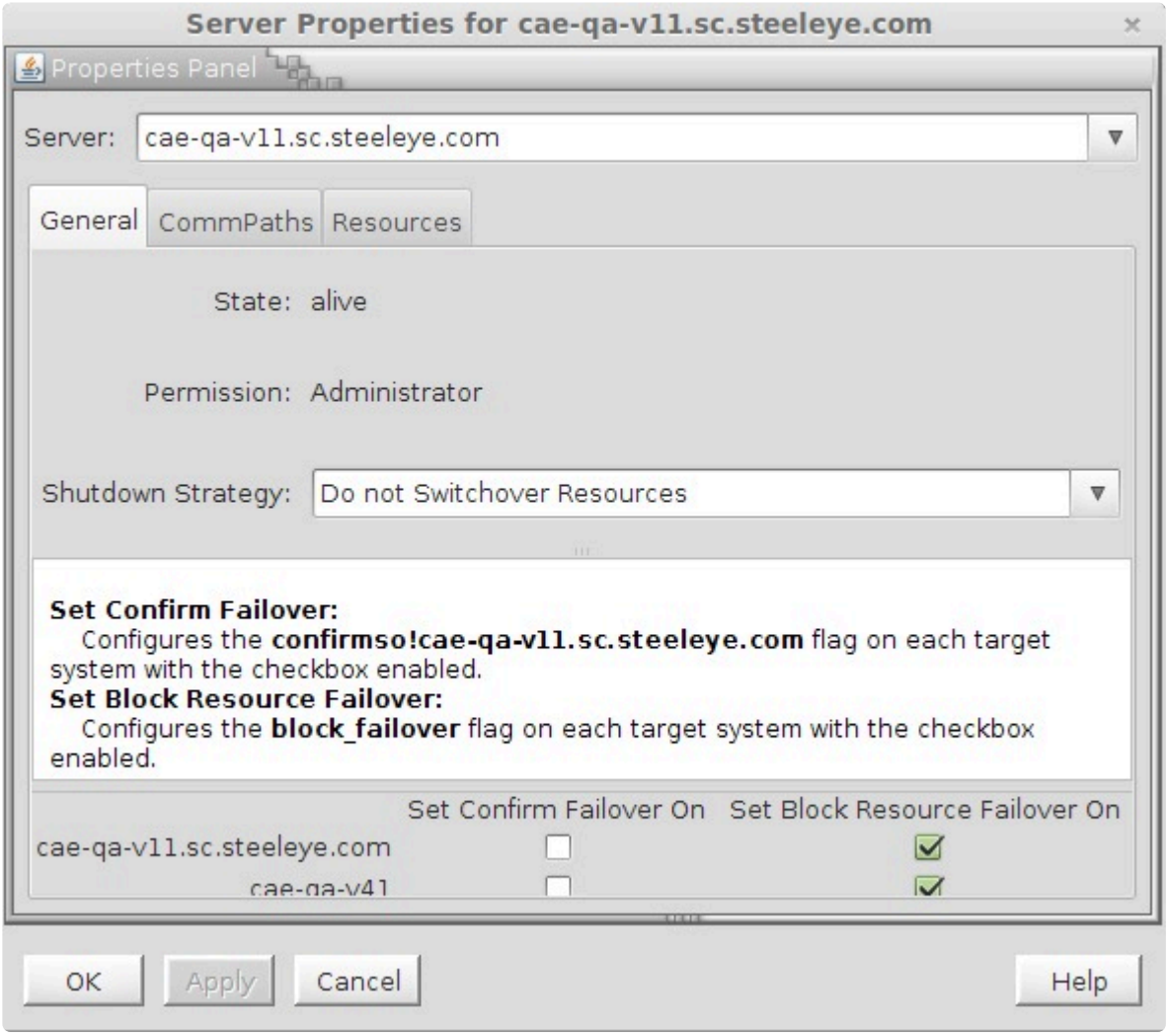
# [Resource Properties] - [Relations]



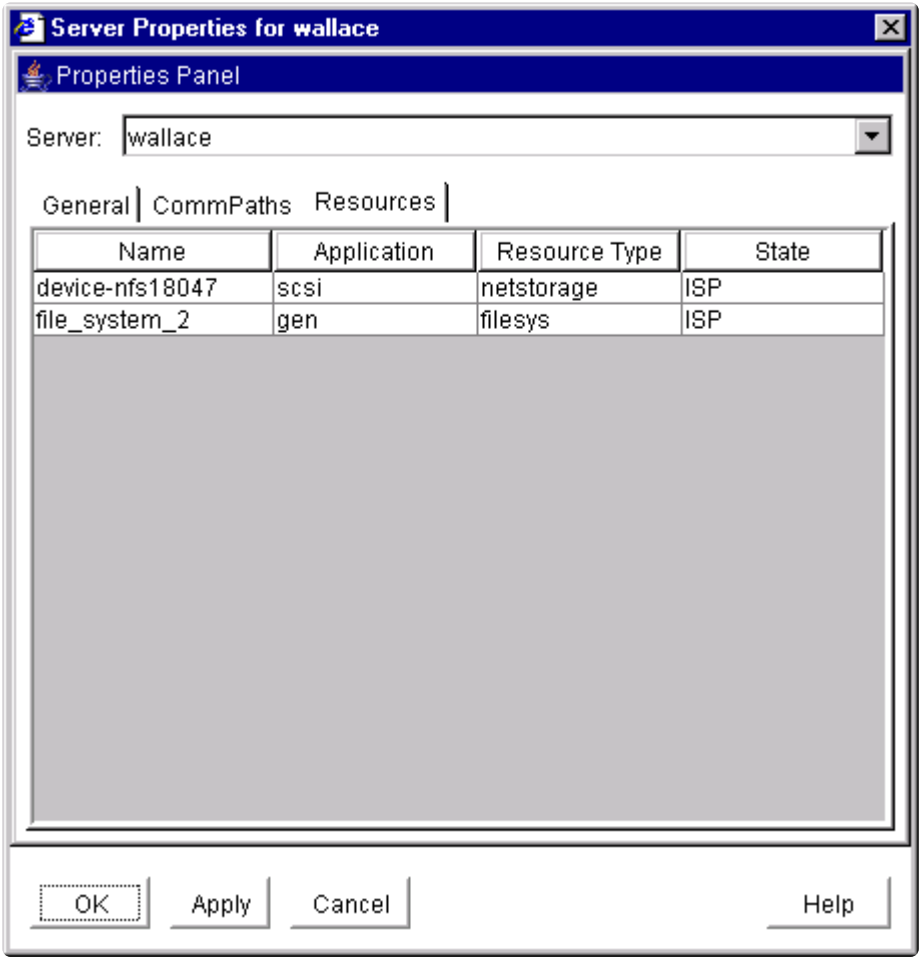
# [Server Properties] - [Commpath]



# [Server Properties] - [General]



# [Server Properties] - [Resource]



# 5.4.5. トラブルシューティング

メッセージカタログ には、操作、管理、GUI など、LifeKeeper for Linux の使用中に遭遇する可能性のあるすべてのエラーコードが列挙されています。また、エラーコードの原因に関する追加の説明や、問題解決のために必要な処置についても、必要に応じて記載されています。この完全なリストを検索すると、受信したエラーコードを見つけることができます。また、以下の個別のメッセージカタログに直接アクセスすることもできます。

- Core メッセージカタログ
- DB2 メッセージカタログ
- DMMP Kit メッセージカタログ
- EC2 メッセージカタログ
- ファイルシステムキットメッセージカタログ
- Gen/App Kit メッセージカタログ
- IP Kit メッセージカタログ
- Oracle Listener Kit メッセージカタログ
- Oracle Kit メッセージカタログ
- SCSI Kit メッセージカタログ
- DataKeeper Kit メッセージカタログ
- QSP Kit メッセージカタログ
- GUI メッセージカタログ

上記のメッセージカタログに加え、以下のトピックでも、直面する可能性がある問題や制限事項のトラブルシューティングについて詳細を説明します。

---

- LifeKeeper が開始するフェイルオーバーの一般的な原因
- 既知の問題と制限
- GUI トラブルシューティング
- コミュニケーションパスの稼働と停止
- 不完全なリソースの作成

不完全なリソースの優先順位の変更

階層の設定中に共有ストレージが見つからない

LifeKeeper サーバ障害からの復旧

停止できないプロセスからの復旧

手動リカバリ時のパニックからの復旧

Out-of-Service 階層の復旧

リソースタグ名の制限

シリアル (TTY) コンソールの警告

システムが init 状態 S に遷移しているという警告

共有ストレージでスレッドがハングしているというメッセージ

# 5.4.5.1. ソリューション

## - LifeKeeper Core -

**アクションを実行する前に LifeKeeper が起動するのに十分な時間を与える**

ソリューションの詳細

**問題:**

LifeKeeper を介してアクションを実行する前に起動するための十分な時間が許可されていない場合、特定のプロセスの初期化が完了していない可能性があるため、アクションが失敗するべきではないときにアクションが失敗するという特定のシナリオが発生する可能性があります。LifeKeeper の起動とプロセスの初期化に十分な時間を確保することが重要です。

同じアクティビティの一部として複数のノードで LifeKeeper を開始する場合、1 つのノードで lkstart を実行するときは、次のノードで lkstart を実行する前に終了するのに十分な時間を与えることが重要です。

**ソリューション:**

起動時に実行されるいくつかのプロセスにはさまざまな時間がかかるため、他の場所で LifeKeeper を起動したり、アクションを実行したりする前に待機すべき時間は決まっていません。同様に、起動のたびに同じ時間がかかるわけではないので、一度測定した時間が常に正確であると仮定しないでください。

代わりに、LifeKeeper のログを使用して、起動と初期化がいつ完了したかを判断してください。LifeKeeper が起動時に最後に行うことは、LifeKeeper リソースの初期化です。 LifeKeeper の最新の起動後に「RESOURCE INITIALIZATION FINISHED」というログ メッセージが表示されたら、クラスタ内の次のノードで起動するか、LifeKeeper GUI または CLI を介してアクションを実行しても問題ありません。

## - SAP/Oracle パッチ適用 -

**DataKeeper を使用した Oracle ノード（SAP/Oracle）へのパッチ適用**

ソリューションの詳細

ほとんどの Oracle のアップデート/パッチ適用にはシステムテーブルドライブへのアクセスのみが必要であり、データードライブ（保護されたDataKeeperのデータードライブ）へのアクセスは必要ありません。

一般的には以下の手順で行うことができます。

手順を開始する前にプライマリーでブロックフェイルオーバーを設定し、ターゲットでフェイルオーバーを確認します。

1. スタンバイ/ターゲットサーバーで、以下を実施します。
   a. 適切なアップグレード、パッチなどを適用します。
   b. サーバーを再起動します。
2. 次に、リソースをアクティブ/ソースサーバーからスタンバイ/ターゲットサーバーにスイッチオーバーします。
   a. Oracle データベースにアクセスできることを確認します。
3. もう一方のノード（この時点ではスタンバイノード、手順の当初はアクティブ/ソースサーバー）で、以下を実施します。
   a. 適切なアップグレード、パッチなどを適用します。
   b. サーバーを再起動します。
4. （オプション）再度スイッチオーバーを実行し、元のアクティブ/ソースサーバーの Oracle への接続とアクセスを確認します。

パッチが DataKeeper ボリュームへのアクセスを必要とする場合（例：catsbp の実行など）は、ミラーを一時停止/ロック解除して、スタンバイ/ターゲットシステムでアップグレードを実行する必要があります。

バックアップで Oracle を起動できることを確認してから、バックアップノードで Oracle を停止し、ミラーリングを続行します。次に、ソースシステムでアップグレードを繰り返します。

手順の最後に、プライマリーの「block failover」フラグとターゲットの「confirm failover」フラグを削除します。

## - Storage Quorum -

**SPS for Linux: Storage Quorumを使用する際のハートビートに関する推奨事項**

ソリューションの詳細

問題:
Storage Quorumを使用する際のハートビートに関する推奨事項について

ソリューション:

**Storage Quorum** を使用する 場合、 **SIOS** では **LCMNUMHBEATS** を増やして 、パスが障害と判定されるまでの時間を長くすることを推奨しています。これにより、タイムアウト期間がデフォルトの15秒から45秒に変更されます。

**LCMNUMHBEATS** を **9** に変更します

```
LCMNUMHBEATS=9
```

コアパラメータを変更することになるので、LifeKeeperを再起動する必要があります。

**サービスの停止時間を最小限に抑える** には, “`lkstop -f`“ を使用するとリソースを実行したままにすることができます。
LifeKeeperが停止している間は、保護されたリソースの障害は検出または対処されません。

```
# lkstop -f

# lkstart
```

> 注意: これらの変更は、クラスター内の各ノードで行う必要があります。

**SPS for Linux:** ノード間の通信が失われたときに、**Storage Quorum**はフェイルオーバーを防止できない

ソリューションの詳細

問題:
Storage Quorumの構成が不完全なため、通信処理が失われたときに障害が発生する

ソリューション:

qwk_storage_init をクラスター内の各ノードで実行する前に、クラスターノード間のすべてのコミュニケーションパスを作成し、**“ALIVE”** にする必要があります。

**これがうまくいかない場合** は、 すべてのコミュニケーションパスが **ALIVE** になってから 、 以下のコマンドを実行して Storage Quorum 構成を再初期化してください。

```
# /opt/LifeKeeper/bin/qwk_storage_exit

# /opt/LifeKeeper/bin/qwk_storage_init
```

> 注意: これにより、Storage Quorum が再初期化されます。

**SPS for Linux: Amazon S3** ストレージを使用する場合の **storage quorum** パラメーターの推奨値

ソリューションの詳細

問題:
Amazon S3 ストレージを使用する場合の storage quorum パラメーターの推奨値について

ソリューション:

ドキュメントに従って storage quorum を設定したら、ハートビートのパラメーターを設定します。
詳細はこちら

**いくつかの方法** で、あなたの環境においてデフォルトの設定が十分なものになっているか確認することができます。

2つの主要なパラメーターがあります。

- QWK_STORAGE_HBEATTIME (デフォルト値: 6) – QWK オブジェクトを読み書きする間隔を **秒**

単位 で指定し ます。
- QWK_STORAGE_NUMHBEATS (デフォルト値 :4) – Witness チェックで対象ノードに障害が発生していると判断するための値を指定します。QWK オブジェクトの読み込みにおいて、このパラメーターに指定した **回数以上更新が停止** していると、対象ノードに障害が発生していると判断します。

Amazon S3 bucketを使用してQWK オブジェクトを保存する場合 (QWK_STORAGE_TYPE=aws_s3)、ご利用の環境において以下のコマンドを実行して接続に **問題がないこと** を確認することを **推奨します** 。

1. **ping s3.amazonaws.com** を実行し、応答時間が1秒以下であることを確認してください。この操作により、EC2 ノードからグローバル AWS ドメインへの正常な **接続性が確保** されます。

> ✻ 注意: S3 はグローバルサービスであっても、S3 バケットはリージョンにあります。

2. **ping <bucketname>.s3.amazonaws.com** を実行することで、ホストするS3 サービスの IP アドレスを解決します。こちらも1秒以下である必要があります。

   その他に検討する事は、このノードでのS3のアクティビティ全体に対して渡される **データの量** です。ファイル転送で確認することもできます。ping を使用して応答時間の測定をすることも可能です。(上記の ping フォーマット) を参照してください。

> ✻ 注意: AWS においては、S3 はグローバルサービスですが、EC2 ノードはリージョンサービスです。S3 URI (Amazon S3 storage Uniform Resource Identifier またはそのウェブアドレス) は、 **特定のリージョン** にあることがあります。

トラフィックがある場合とない場合を比較し、上記のハートビートの回数 (QWK_STORAGE_NUMHBEATS) および ハートビートの時間 (QWK_STORAGE_HBEATTIME) を調整することが可能です。

上記 パラメーター は **qwk_storage_init** を実行する前に指定する必要があります。

**ほとんどの場合** また ping 応答が1秒以下の場合はデフォルト値で十分ですが、S3 が遅延するもしくはデグレードしているように見える場合は、 QWK_STORAGE_HBEATTIME の値を増やすことをお勧めします。デフォルトの 6 から 7 へ変更してください。 これによりタイムアウト値が 24 秒 (6 × 4) から 28 秒 (7 × 4) に増加します。 タイムアウト値を 30 秒以上にすることは推奨しません。

デフォルト値を変更する場合は、 **クラスター内の各ノード** で変更を行い、すべてのコミュニケーションパスが **ALIVE** になっている状態で、クラスター内の各ノード で以下のコマンドを実行して Storage Quorum 構成を再初期化してください。

```
# /opt/LifeKeeper/bin/qwk_storage_exit

# /opt/LifeKeeper/bin/qwk_storage_init
```

## - Majority Mode -

**Witnessノードを使用したLifeKeeperのシャットダウンと再起動の手順**

1. lkstop を使用してバックアップ（ターゲット）サーバーを停止します。

lkstop コマンドが完了するのを待ちます。lkstop が完了したことをログエントリーで確認します。

NOTIFY:shutdown:::010055:LifeKeeper stopped

Quorum メッセージ:

NOTIFY:event.comm_down:::010469:We do have quorum on comm_down, continuing

通常は2分未満で完了します

2. lkstop を使用してプライマリー（ソース）サーバーを停止します。

lkstop コマンドが完了するのを待ちます。lkstop が完了したことをログエントリーで確認します。

NOTIFY:shutdown:::010055:LifeKeeper stopped

Quorum メッセージ:

NOTIFY:event.comm_down:::010469:We do have quorum on comm_down, continuing

通常は2分未満で完了します

3. lkstop を使用して、WitnessノードでLifeKeeperを停止します。

lkstop コマンドが完了するのを待ちます。lkstop が完了したことをログエントリーで確認します。

NOTIFY:shutdown:::010055:LifeKeeper stopped

通常は2分未満で完了します

4. lkstart を使用してWitnessサーバーを起動します。

lkstart コマンドが完了するのを待ちます。lkstart が完了したことをログエントリーで確認します。

INFO:event.lcm_avail:::010479:RESOURCE INITIALIZATION FINISHED

通常は2分未満で完了します

5. `lkstart` を使用してバックアップ（ターゲット）サーバーを起動します。

`lkstart` コマンドが完了するのを待ちます。`lkstart` が完了したことをログエントリーで確認します。

INFO:event.lcm_avail:::010479:RESOURCE INITIALIZATION FINISHED

Quorum メッセージ:

INFO:event.comm_up:::010490:We do have quorum on comm_up to <node>, putting resources into service if needed

通常は2分未満で完了します

6. `lkstart` を使用してプライマリー（ソース）サーバーを起動します。

`lkstart` コマンドが完了するのを待ちます。`lkstart` が完了したことをログエントリーで確認します。

INFO:event.lcm_avail:::010479:RESOURCE INITIALIZATION FINISHED

Quorum メッセージ:

INFO:event.comm_up:::010490:We do have quorum on comm_up to <node>, putting resources into service if needed

通常は2分未満で完了します

この順序でサーバーを停止することにより、リソースはプライマリーサーバーでIn-Serviceのままになります（フェイルオーバーは開始されません）。バックアップサーバーが停止したときにミラーはIn-Serviceではなかったため、サーバーが復旧してもミラーはIn-Serviceにならず、Out-of-Service（OSU）状態のままになります。

LifeKeeperがプライマリーサーバーで起動すると、LKはミラーをIn-Serviceにします。これは、LifeKeeperがダウンしたときにミラーがISPであり、バックアップサーバーではIn-Serviceになっていないためです。LifeKeeperは、バックアップサーバーの準備ができていると判断すると、プライマリーサーバーからバックアップサーバーへの再同期を開始します。

> ! 以下の順序でシステムを起動しないと、data_corruptフラグが立ってしまい、ユーザーの介入が必要になる場合があります。

## - SAP HANA -

**HANA** データベースにパッチを適用する手順

ソリューションの詳細

問題:
**HANA** データベースのパッチ適用手順について教えてください。

ソリューション:

**Node 1** = source
**Node 2** = backup

1. Node 1で quickCheck を無効にします。 /etc/default/LifeKeeperファイルの LKCHECKINTERVAL の値を 0 に設定し、killall コマンドで lkcheck プロセスをkillしてください。
2. Node 2で “lkstop -f” を実行して LifeKeeper を停止してください。
    a. a.HSRがNode 1と2で実行されていることを確認してください。
3. Node 2で HANA を停止してパッチを適用してください。
4. Node 2で HANA を起動してください。
    a. a.HSR が引き続き実行中であることを確認してください。
5. Node 2で “lkstart” を実行して LifeKeeper を起動してください。
6. Node 2へ HANA に関連するリソースをスイッチオーバーします。
7. 上記の手順を繰り返し、Node 1にパッチを適用します
8. 必要に応じてNode 1にリソースをスイッチオーバーしてください。
9. quickCheck を有効にします。
    /etc/default/LifeKeeperファイルの LKCHECKINTERVAL を変更前の値に設定してください。

## - ミラーの Quickcheck が常に失敗と回復を繰り返す -

ミラーの **Quickcheck** が常に失敗と回復を繰り返す

ソリューションの詳細

問題:
**lifekeeper.log** を見ると、ミラーが常にクイックチェックに失敗していることがわかりますが、**recover** は常に機能しています。

メッセージの例:

NOTIFY:lcd.recmain:recover:datarep-data:011115:BEGIN recover of “datarep-data” (class=netraid event=recover)
INFO:dr:recover:datarep-data:104008:/dev/md0: merging bitmap from target “SV-GCS-LIVEB”*
*Oct 5 05:57:07 SP-GCS-LIVEA recover [11754]: INFO:dr:recover:datarep-data:104009:/dev/md0: bitmap merged, resyncing 2.3
*Oct 5 05:57:12 SP-GCS-LIVEA recover [11754]:
INFO:dr:recover:datarep-data:104095:Partial resynchronization of component “/dev/nbd1” has begun for mirror “/dev/md0”*

これは通常、nbd とメッセージ ログのエラーが一致します。

**Oct 5 05:56:59 SP-GCS-LIVEA kernel: [5499433.410998] nbd (pid 7278: nbd-client) got signal 9**
**Oct 5 05:56:59 SP-GCS-LIVEA kernel: [5499433.411003] nbd1: shutting down socket**
**Oct 5 05:56:59 SP-GCS-LIVEA kernel: [5499433.411015] nbd1: Receive control failed (result -4)**
**Oct 5 05:56:59 SP-GCS-LIVEA kernel: [5499433.411039] nbd1: queue cleared**

**Oct 5 05:57:11 SP-GCS-LIVEA nbd-client: Begin Negotiation**
**Oct 5 05:57:11 SP-GCS-LIVEA nbd-client: size = 268434407424**
**Oct 5 05:57:11 SP-GCS-LIVEA nbd… (truncated, see original email for full text)**

解決方法:

これらのメッセージは、レプリケーション接続を切断させる nbd の問題を示しているようです。LifeKeeper ログ内の recover による再同期は、切断された接続を再確立するための反応です。

ミラーが作成されると、その状態は quickCheck とソースの mdadm プロセスによって監視されます。quickCheck プロセスはいくつかの項目をチェックし、(/proc/mdstat の情報から) ミラーが同期していないことを検出した場合、ターゲットに ping できない場合、ターゲットの状態がアクティブでない場合、または nbd-client/nbd-server プロセスが実行されていないことを検出した場合に、recover イベントを発行します。

recover は、mdadm 監視プロセスを介した md ドライバーからのイベントに基づいても開始されます。これには、Fail、FailSpare、および DegradedArray イベントが含まれます (これらは mdadm のマニュアルページに記載されています) 。

これらのイベントは、md ドライバーが検出した問題を示しており、LifeKeeper はドライバーと同じ状態になるように対応する必要があります。 さらに、ミラーが使用しているコミュニケーションパスがダウンした場合も、recover イベントが発生します。

1 つのチューニング オプションがあります。

NBD_XMIT_TIMEOUT パラメータの値を大きくします。NBD_XMIT_TIMEOUT のデフォルト値は 6 秒です。

この値をあまり大きくしないように注意してください。この値によりパケット転送での実際のハング状態が検出され、接続をリセットしてミラーリングを再開します。待機時間が長すぎると、ソースでの書き込みがハングし、最終的にはシステムがハングする可能性があります。

## - EC2 -

**Curl 呼び出しが失敗しました (err=7)(output=curl: (7) 接続に失敗しました (EC2 Kit)**

ソリューションの詳細

**ISSUE:**
AWS CLI と EC2 Recovery Kit で問題が発生します。

ログには、Curl 呼び出し (AWS CLI から IMDS) が err=7 で失敗し、その後、quickcheck スクリプトがチェックを誤りフェイルオーバーがトリガーされたことが示されています。 フェイルオーバーは、curl 呼び出しのタイムアウトが 5 秒と短すぎて、AWS の断続的なネットワーク問題によってトリガーされたため、フェイルオーバーは必要ありませんでした。curl 呼び出しにもう少し時間があれば、フェイルオーバーは防止できます。

エラーメッセージ：
ERROR:ec2:quickCheck:ec2-1.1.1.1:123456:curl call failed
(err=7)(output=curl: (7) Failed to connect to 169.254.169.254 port 80:
Connection refused)

169.254.169.254 port 80 is the AWS Instance MetaData Service

ソリューション:
このような不必要なフェイルオーバーの発生を防ぐために、/etc/defaults/LifeKeeper で変更できる調整可能なパラメーターがあります。 変更するパラメーターは次のとおりです。
**LK_CURL_TIMEOUT=X**

- このパラメーターのデフォルト値は 5 です。
- /etc/defaults/LifeKeeper ファイルに調整可能な行が存在しない場合、値は 5 に設定されます。

変更するには、数値を変更するか、パラメーターが存在しない場合は次の行を追加します。
**LK_CURL_TIMEOUT=10**

ご利用の環境で発生する可能性のある、断続的なネットワーク障害の長さに基づいて、タイムアウト値を少しずつ増やしてください。
一般的に、タイムアウト値を 10 にすると、ほとんどのネットワーク障害の問題が回避されるようです。

**注意:**
10 に変更しても問題が再び発生する場合は、値を数秒増やしてください。

# 5.4.5.1.1. ビデオソリューション

ライセンスポータルの概要

https://fast.wistia.net/embed/iframe/rdljg3wui5

## 概要:

このビデオでは、当社のライセンスポータルのナビゲーション概要を説明します。このビデオで言及されているトピックは次のとおりです。

- License Portal へのログイン
- License and Delivery Portal の概要
- List Entitlements の概要
- Licenses のアクティベート概要
- Licenses 概要
- List Licenses の概要
- Rehost と Repair の実行
- Organizations の概要
- List Users の概要
- Users 作成の概要
- User ビュー

**License Portal** へのアクセス

- License Portal にアクセスしてください。

# 5.4.5.2. LifeKeeper が開始するフェイルオーバーの一般的な原因

障害が発生した場合、LifeKeeper には2つのリカバリー方式があります。ローカルリカバリーとサーバー間リカバリー（フェイルオーバー）です。ローカルーリカバリーが失敗した場合、「フェイルオーバー」が実行されます。フェイルオーバーは、アクティブだったアプリケーション、サーバー、システム、ハードウェアコンポーネント、ネットワークの障害や異常終了が発生したときのバックアップサーバーへの自動的な切り替えと定義されます。フェイルオーバーとスイッチオーバーは基本的には同じ動作ですが、スイッチオーバーが人の介入を必要とするのに対し、フェイルオーバーは自動で、通常は警告なしで実行されます。この自動的なフェイルオーバーは、さまざまな理由で発生します。以下は、LifeKeeper が開始するフェイルオーバーの最も一般的な例の一覧です。

## サーバーレベルでの原因

### サーバーの障害

LifeKeeperには設定内の各サーバーに、ペアのサーバーが動作していることを定期的に通知する組み込みのハートビート信号があります。サーバーがハートビートメッセージを受信しなかった場合に、障害として検出されます。

- プライマリーサーバーが電源を喪失するか、電源がオフになる。
- 過負荷による CPU 使用率 – 非常に高負荷の I/O の下では、これらの遅延と低メモリ状態によってシステムが無応答になり、LifeKeeper がこれをサーバーのダウンとして検知し、フェイルオーバーを開始することがあります。
- Quorum/Witness – プライマリーサーバーが Quorum を失った場合、Quorum/Witness の I/O フェンシングメカニズムの一環として、( 設定に基づいて )「 *fastboot* 」、「 *fastkill* 」、または「 *osu* 」が実行され、フェイルオーバーが開始されます。フェイルオーバー時を決定する際、プライマリーサーバーで障害が発生しクラスターを構成できなくなったことが確認できた場合にのみ、witness サーバーによりバックアップサーバー上でリソースを in service にすることができます。全体のアクセスや、パフォーマンス、in-service のノードが影響を受けない場合は、これによってノード間で発生する単純な通信障害から発生するフェイルオーバーを回避します。

| 関連トピック |
| --- |
| サポートストレージ一覧 |
| サーバー障害リカバリーのシナリオ |
| LifeKeeper ハートビートの調整 |
| Quorum/Witness |

## 通信障害 / ネットワーク障害

LifeKeeperは、サーバー間で 5 秒ごとにハートビートを送信します。通信障害によって 2 回のハートビートが途絶し3 回目のハートビートで再開した場合、LifeKeeperは一切処置を行いません。コミュニケーションパスの切断継続時間がハートビート 3 回分になった場合は、LifeKeeperはコミュニケーションパスを切断と判定します。ただし、他方の冗長的なコミュニケーションパスも切断と判定されるまではフェイルオーバーを開始しません。

- プライマリーサーバーへのネットワーク接続の喪失。
- ネットワークの遅延。
- TCP コミュニケーションパス上のネットワークトラフィックが大きくなると、偽のフェイルオーバーや LifeKeeper の初期化の問題など、予期せぬ動作が生じる可能性があります。
- STONITH を使用しているときに LifeKeeper がノードとの通信障害を検出すると、そのノードの電源が切断され、フェイルオーバーが発生します。
- NIC の障害。
- ネットワークスイッチの障害。
- 手動によるネットワーク接続の解除。

| 関連トピック |
| --- |
| コミュニケーションパスの作成 |
| LifeKeeper ハートビートの調整 |
| ネットワーク設定 |
| ネットワーク設定の確認 |
| SNMP 経由の LifeKeeper イベントの転送 |
| ネットワーク関連のトラブルシューティング |
| ファイアーウォールを使用した状態での LifeKeeper の実行 |
| STONITH |

## スプリットブレイン

単一のコミュニケーションパスを使用しているときに、そのコミュニケーションパスで障害が発生した場合、複数のシステム上で同時に LifeKeeper の階層が in service になることがあります。これは、偽のフェイルオーバーまたは「 **スプリットブレイン** 」シナリオと呼ばれます。「スプリットブレイン」シナリオでは、各サーバーが、アプリケーションを制御できると認識しているため、データにアクセスしようとしたり共有ストレージデバイスにデータを書き込もうとする場合があります。スプリットブレインシナリオを解決

するために、LifeKeeper では、サーバーの電源をオフにしたり、再起動したり、階層を out-of-service にすることで、すべての共有データに対するデータの整合性を保証することができます。また、TCP コミュニケーションパス上のネットワークトラフィックが大きくなると、偽のフェイルオーバーや LifeKeeper が適切に初期化できなくなるなど、予期しない動作が生じる可能性があります。

以下に、スプリットブレインが発生する可能性のあるシナリオを示します。

- 上記のいずれかの通信障害
- LifeKeeper の不正なシャットダウン
- サーバーリソースの枯渇
- すべてのネットワークパスの喪失
- DNS またはその他のネットワークの問題
- システムのロックアップ / 解除

# リソースレベルでの原因

LifeKeeper には、個々のアプリケーションおよび関連アプリケーションのグループを監視する機能があり、定期的にローカルリカバリーを実行したり、保護下のアプリケーションに障害が発生したときに通知することができます。関連し合うアプリケーションの例としては、主アプリケーションが下位のストレージまたはネットワークリソースに依存する階層などがあります。LifeKeeper は、これらの保護下のリソースのステータスと健全性を監視します。LifeKeeper は、リソースが障害状態にあると判断すると、外部の介入なしに現在のシステム (in-service のノード) でリソースまたはアプリケーションをリストアしようとします。このローカルリカバリーが失敗した場合、リソースのフェイルオーバーが開始されます。

## アプリケーションの障害

- アプリケーションの障害が検出されたが、ローカルリカバリープロセスが失敗。
- 削除の失敗 - リソースのフェイルオーバープロセスでは、該当する重要なアプリケーションのすべての機能を提供するために、プライマリーサーバー上でのサービスから特定のリソースを削除し、選択したバックアップサーバーでそのリソースを in service にする必要があります。 **この削除プロセスが失敗した場合は、プライマリーサーバーの再起動が実行され、** その結果、完全なサーバーフェイルオーバーに移行します。

  削除の失敗の例

  - ファイルシステムをアンマウントできない
  - 保護対象のアプリケーション (Oracle、mySQL、Postgres など) をシャットダウンできない

| 関連トピック |
| --- |
| ファイルシステムの健全性監視 |
| リソースエラーリカバリーのシナリオ |

## ファイルシステム

- ディスクフル - LifeKeeper のファイルシステムの健全性監視は、ファイルシステムリソースのフェイルオーバーになる可能性のある、ファイルシステムのディスクフル状態を検出できます。
- アンマウントされた、または不適切にマウントされたファイルシステム - in-service 状態で LK 保護下のファイルシステムをユーザが手動でアンマウントまたはオプションを変更。
- 再マウントの障害 - 以下のリストに、フェイルオーバーに進行する場合がある再マウントの障害の一般的な原因を示します。
    - ファイルシステムが破損している (fsck の障害)
    - マウントポイントディレクトリの作成失敗
    - マウントポイントがビジー
    - マウントの失敗
    - LifeKeeper の内部エラー

| 関連トピック |
| --- |
| ファイルシステムの健全性監視 |

## IP アドレスの障害

IP Recovery Kit によって IP アドレスの障害が検出されると、結果として生じる障害によって IP ローカルリカバリースクリプトが起動されます。LifeKeeper は最初に、その IP アドレスを現在のネットワークインターフェース上で in service に戻そうとします。ローカルリカバリーの試みが失敗すると、LifeKeeper はその IP アドレスと依存関係を持つすべてのリソースのバックアップサーバーへのフェイルオーバーを実行します。フェイルオーバー中に、削除プロセスによって現在のサーバー上の該当する IP アドレスの設定が解除され、バックアップサーバー上でそのアドレスを設定できるようになります。 **この削除プロセスに失敗すると、システムは再起動します。**

- IP 競合
- IP コリジョン

- DNS の名前解決の障害
- NIC やスイッチの障害

| 関連トピック |
| --- |
| 切り替え可能な IP アドレスの作成 |
| IP ローカルリカバリー |

### リザベーションコンフリクト

- 保護されたデバイスへのリザベーションの喪失または盗難
- 保護されたリソースデバイスへのリザベーションまたは制御の回復が不可能 (手動のユーザ介入、HBA またはスイッチの障害が原因)

| 関連トピック |
| --- |
| SCSI リザベーション |
| リザベーションの無効化 |

## SCSI デバイス

- 保護された SCSI デバイスを開くことができない。デバイスに障害が発生しているか、システムからデバイスが削除されている可能性があります。

# 5.4.5.3. 既知の問題と制限

下記に、LifeKeeper for Linux で明らかになっている制限または既知の問題を機能領域ごとに示します。

インストール - 既知の問題と制限

LifeKeeper Core - 既知の問題と制限

インターネット / IP ライセンス - 既知の問題と制限

GUI - 既知の問題と制限

データレプリケーション - 既知の問題と制限

IPv6 - 既知の問題と制限

Apache Recovery Kit - 既知の問題と制限

Oracle Recovery Kit - 既知の問題と制限

MySQL Recovery Kit - 既知の問題と制限

NFS Server Recovery Kit - 既知の問題と制限

SAP Recovery Kit - 既知の問題と制限

LVM Recovery Kit - 既知の問題と制限

Multipath Recovery Kits (DMMP / HDLM / PPATH / NECSPS) - 既知の問題と制限

DMMP Recovery Kit - 既知の問題と制限

DB2 Recovery Kit - 既知の問題と制限

Sybase ASE Recovery Kit - 既知の問題と制限

WebSphere MQ Recovery Kit - 既知の問題と制限

EC2 Recovery Kit - 既知の問題と制限

# 5.4.5.3.1. インストール - 既知の問題と制限

| 説明 |
| --- |
| **LifeKeeper を正しくインストールするには、yum または zypper の機能構成が必要です**<br><br>設定が正しくない場合や非機能構成の yum または zypper では、LifeKeeper のインストールスクリプトが失敗する可能性があります。次のような出力が表示されます。<br><br>`Install LifeKeeper and dependent packages done.`<br>`Setup high availability data replication features.. done.`<br>`Setup NFS high availability features…`<br>`Configure LifeKeeper management group`<br><br>`Setup failed.`<br>`Fix the problem and try again.`<br>`sed: can't read /etc/default/LifeKeeper: No such file or directory`<br>`/tmp/mnt/setuplibs/install.sh: line 118: /config/system_information: No such file or directory`<br>`/tmp/mnt/setuplibs/install.sh: line 122: /config/system_information: No such file or directory`<br>`/tmp/mnt/setuplibs/install.sh: line 122: /config/system_information: No such file or directory`<br>`/tmp/mnt/setuplibs/install.sh: line 122: /config/system_information: No such file or directory`<br><br>さらに、/var/log/LK_install.log ファイルに次のように表示されることがあります。<br><br>`Abort, retry, ignore? [a/r/i/? shows all options] (a): a`<br>`done.` |
| **リリース 7.4 以降では、SIOS 製品 RPM パッケージの再配置はサポートされません** |
| **Linux の依存関係**<br><br>オプションの Recovery Kit を含めて SIOS Protection Suite for Linux のインストールには、いくつかの依存関係を持つパッケージが必要になります。パッケージマネージャが適切に設定されていれば、必要なパッケージはパッケージマネージャによって自動的にインストールされます。<br>自動でインストールが行えなかった場合には setup スクリプトが中断されるので、手動で依存関係のあるパッケージ (詳細は Linux の依存関係 を参照してください) をインストールした後に、setup スクリプトを再実行してください。<br><br>**注記**：依存関係にあるこれらのパッケージのインストールが正常に完了していない場合、SIOS |

Protection Suite for Linux を起動する機能、および SIOS Protection Suite for Linux の GUI をロードする機能にも影響します。

**新しいデバイスがスキャンされているときに nbd ドライバがロードされると、multipathd デーモンはエラーログにエラーを記録します**

**解決方法**：これらのエラーをログに記録しないようにするには、*/etc/multipath.conf* の blacklist に **devnode "^nbd"** を追加します。

**LK のインストール前に mksh が導入されていると、setup 実行時にパッケージ競合を検出して失敗する**

`mksh` パッケージがインストール済みの場合、 SIOS Protection Suite for Linux のセットアップスクリプトはパッケージの競合を検出して失敗します。また、システムの要件で `KornShell(ksh)` が必要になる場合は `KornShell(ksh)` パッケージを導入してください。

**対応策:** RHEL、CentOS、または Oracle Linux では、`mksh` パッケージをアンインストールし、必要に応じて `KornShell(ksh)` パッケージをインストールしてください。その後、SIOS Protection Suite for Linux のセットアップスクリプトを再実行してください。

例:

1. mksh パッケージをアンインストールしてください。
   ```
   yum remove mksh
   ```
2. ksh パッケージをインストールしてください。 (※KornShell(ksh) が必要な場合)
   ```
   yum install ksh
   ```
3. セットアップスクリプトを再実行してください。

**デーモンの予期しない終了**

Red Hat Enterprise Linux 7.2 および Red Hat 7.2 の派生システムへのアップデートを行った環境で、IPC を使用するデーモンが予期せず終了することがあります。 Red Hat Enterprise Linux 7.2 へ導入された新機能である systemd が、最後のユーザセッション終了時に全てのプロセス間通信 (IPC) に割り当てられた資源の回収を行います。 このセッションとは管理用の cron ジョブもしくは対話式のものです。このふるまいにより、同一ユーザでかつ同一資源を使用しているデーモンが意図せず終了することがあります。

この問題を回避するためには、/etc/systemd/logind.conf を編集して次の行を追加してください。
`RemoveIPC=no`

その後、次のコマンドを実行し、変更を反映させてください。

```
systemctl restart systemd-logind.service
```

これらの手順により、デーモンは上記のような状況でも予期せず終了することはなくなります。共有メモリとセマフォを使用しているアプリケーション（MQ、Oracle、SAP など）はこの問題の影響を受ける可能性があるため、この手順が必要となります。

**Kernel のアップデート後、LifeKeeper の "setup" の再実行が必要になるケースがあります**

Red Hat Enterprise Linux 7.x / CentOS 7.x / Oracle Linux 7.x 環境において、カーネルを7.3 相当以降にアップデートした場合に、DataKeeper が正常に動作しないという不具合が生じます。

**回避策:**
アップデート後のシステムで、LifeKeeper をインストールする際に実行した "setup" スクリプトを再実行することで問題は解消します。

**説明**
OSのカーネルモジュールの互換性に問題があり、アップデート前に組み込まれていたカーネルモジュールが、アップデート後のカーネルでは利用出来ません。
DataKeeper は、 nbd.ko という、ネットワーク経由でディスクアクセスを行うカーネルモジュールを使用しています。このモジュールは、LifeKeeper インストール時に行う setup スクリプトにより、適切なものがインストールされます。
カーネルのアップデート後に、setup スクリプトを再度実行することで、新しいカーネル用の nbd.ko が導入されます。

**setup スクリプトの不要な警告**

LifeKeeperパッケージのインストール状況によって、setup スクリプトを実行した際に以下のような警告が表示されます。
Found changes in following files.
These files are overwritten in install process.
If you want to keep changes, please backup these files.

/opt/LifeKeeper/lkadm/subsys/scsi/DEVNAMEが見つかりません。
/opt/LifeKeeper/lkadm/subsys/scsi/DEVNAME/binが見つかりません。
/opt/Lifekeeper/subsys/scsi/resources/DEVNAMEが見つかりません。

これはパッケージ管理上の問題により発生する警告であり、setup スクリプト及び LifeKeeper の動作には影響しません。

将来のバージョンで修正予定です。

# 5.4.5.3.2. LifeKeeper Core – 既知の問題と制限

| 説明 |
| --- |
| **アンチウィルスソフトウェア – LifeKeeper for Linuxでの除外事項**<br>注意：SIOSではSentinelOneを使用してテストを実施しています。<br><br>ウイルス対策ソフトが動作している環境でLifeKeeper for Linuxを使用すると、LifeKeeperが正常に動作しない場合があります。<br><br>**推奨される対応**<br><br>LifeKeeperに次のパス除外を追加します。<br>/opt/LifeKeeper/*<br>/var/log/lifekeeper*<br>/etc/default/LifeKeeper<br>/var/log/messages<br>/tmp/*<br><br>SAP HANAを使用している場合は、SAP HANAログディレクトリの場所を追加します。<br>LifeKeeperがNBDモジュールをインストールするシステムでDataKeeperを使用する場合は、NBDモジュールを含めます。<br><br>オプションのパス除外：<br>/var/LifeKeeper |
| **アンマウントの時間を増やすか、強制アンマウントが完了するまでの時間を減らす**<br><br>特定の調整可能な値は、アンマウントにかかる時間を増やしたり、強制アンマウントが完了するまでの時間を減らしたりします。 このタイマーに寄与する3つの調整可能なパラメータがあります。<br><br>`FS_UMOUNT_RETRIES` (defaults to 1)<br>`FS_KERNEL_RETRIES` (defaults to 60). (This is the recommended value to change to modify timing)<br>`FS_UMOUNT_TIMEOUT` (defaults to 30)<br><br>Forceumount スクリプトの最大実行時間は、FS_KERNEL_RETRIES 調整パラメーターを操作することで短縮できます。 値を1つ下げると、(最大) スクリプトの実行時間が3秒節約されます。 絶対最小実行時間は30秒、最大実行時間は次のように定義されます。<br><br>最大実行時間 = (`$FS_KERNEL_RETRIES` * 3) + 3*(`$FS_UMOUNT_RETRIES` + 3) |
| **SLESを使用して9.6.2にアップグレードすると失敗します: パッケージ'net-tools-deprecated'が見つかりません。**<br><br>回避策 ：sle-module-legacy rpmを追加して、セットアップを再実行します。その後、アップグレードが機 |

能します。

**Quorum loss actionで“fastboot“を使用すると、Quorum喪失後にAutomatic switchbackが成功しない場合がある**

回避策 ：Quorum loss actionの“fastboot”(`QUORUM_LOSS_ACTION=fastboot`)を使用する場合、サーバーはQuorumを失うと強制的に再起動されます。(詳細については、Quorum/Witness を参照してください。) サーバーがオンラインに戻ったときに、Automatic switchbackを使用するように設定されたリソース階層が、LifeKeeperによって優先度の高いサーバーで自動的にIn-Serviceに戻されない場合があります。 この場合、ユーザーは優先度の高いサーバーでリソース階層を手動でIn-Serviceに戻す必要があります。

**カーネルのアップグレード後の新しいマウントオプションまたは非推奨のマウントオプション**

Linuxカーネルをアップグレードすると、既存のファイルシステムマウントオプションの一部が新しいカーネルで非推奨になったり、新しいカーネルが新しいデフォルトマウントオプションを、既存のマウントに追加したりする可能性があります。 たとえば、“nobarrier“マウント オプションはRedHat Enterprise Linux 8で廃止され、一部のカーネルバージョンでは「logbufs=8」や「logbsize=32k」などの新しいデフォルトマウントオプションが追加されました。

LifeKeeperで保護されたファイルシステムリソースに、カーネルのアップグレード後に廃止されるマウントオプションが含まれている場合、廃止されたオプションは、クラスター内のすべてのサーバーでLifeKeeperリソースのマウントオプションのリストから削除する必要があります。詳細については LifeKeeper ファイルシステムリソースのマウントオプションの変更 を参照してください。

カーネルのアップグレード後に、LifeKeeperで保護された既存のマウントポイントに、カーネルによって新しいデフォルトのマウントオプションが追加された場合、クラスター内のすべてのサーバーでLifeKeeperリソースのマウントオプションのリストに、新しいオプションを追加する必要があります。詳細についてはLifeKeeperファイルシステムリソースのマウントオプションの変更 を参照してください。

**シャットダウンストラテジーを「Do not Switchover Resources」(デフォルト) に設定している場合、LifeKeeperを停止した直後に起動しないでください。LifeKeeper停止から起動までの時間が短いと、スプリットブレインが発生する可能性があります。特にstorageモードのQuorum構成では、注意が必要です。**

LifeKeeperの停止処理と起動処理で競合が発生し、スプリットブレインが発生する可能性があります。LifeKeeperを停止してから起動まで、数秒間の間隔を空けてください。storageモードのQuorum構成は停止処理に時間がかかるので、QWK_STORAGE_HBEATTIME * QWK_STORAGE_NUMHBEATS（デフォルト24秒）より長い時間を空ける必要があります。

**ネットワークの接続などに問題が生じた場合、ネットワークを自動構成するサービスは停止してください**

LifeKeeperを用いてIPアドレスを保護している環境では、avahi-daemonなどのネットワークを自動構成

するデーモンやサービスとIPリソースが競合する場合があります。コミュニケーションパスの復旧やIPリソースの起動などに問題が生じた場合は、ネットワークを自動構成するサービスを停止してください。

**ifconfig downやip link downコマンドによるネットワークの切断はしないでください**

仮想IPリソースの保護対象となっているネットワークインタフェースがコミュニケーションパスでも使用されている場合、ネットワークインタフェースを ifconfig downやip link downコマンドを実行して切断すると、再接続後もコミュニケーションパスが復帰しないことがあります。そのような操作は行わないでください。

**マルチユーザーに設定されたsystemd ターゲットではLifeKeeperは起動しません**

LifeKeeperを適切に機能させるためには、「`systemctl set-default`」または「`systemctl isolate`」を実行する時に、「`lifekeeper-graphical.target`」（グラフィカルモードの場合）または「`lifekeeper-multi-user.target`」（コンソールモードの場合）を使用する必要があります。通常の「`graphical.target`」および「`multi-user.target`」systemd ターゲットを使用しないでください。

**ディスクのUUIDに関するDataKeeperの制限事項**

バージョン9.5.0以降、DataKeeperはオペレーティングシステムにUUIDを提示しないディスクをミラーリングできなくなりました。このようなディスクをミラーリングする一番良い方法は、GPT（GUID Partition Table）でパーティションに分割することです。これは「parted」ツールを使用して実施できます。

**注意:** ディスクをパーティションに分割すると、ディスクにすでに保存されているデータはすべて破壊されます。

**回避策:** DataKeeper for Linuxトラブルシューティング を参照してください。

**SLES 15では、ログのローテーション後にLifeKeeperのログがLifeKeeperログファイルに表示されない場合があります**

コマンドラインで `logrotate`を実行した場合、またはログのサイズが原因でバックグラウンドでログのローテーションが発生した場合、LifeKeeperはロギングを停止します。

**対応策**: `systemctl reload rsyslog`を実行して、LifeKeeperのロギングを再開してください。

**ファイルシステムラベルは、大規模な設定で使用しないことを推奨します**

ファイルシステムラベルを使用すると、大きなクラスタの場合、起動時にパフォーマンスが低下する可能性があります。ラベルを使用するには、システムに接続されるすべてのデバイスをスキャンする必要があり、通常はその結果として問題が生じます。SAN に接続されているシステム、特にデバイスへのアクセスがブロックされているLifeKeeperが導入されているシステムの場合、このスキャニングは非常に遅くなる可能性があります。

Red Hat システムでこのパフォーマンスの問題を防ぐには、/etc/fstabを編集し、ラベルをパス名に置き換えます。

**lkscsid は、一部環境においてディスク障害時にシステムを停止します**

lkscsidがディスク障害を検出すると、デフォルトではsendeventをLifeKeeperに発行し、障害から復旧しようとします。sendevent はまず、ローカルで障害から復旧しようとします。それに失敗すると、ディスクの階層を別のサーバーに切り替えて障害から復旧しようとします。一部のバージョンのLinux（SLES11）では、lkscsidはsendeventを発行できないため、代わりにすぐにシステムを停止します。これは、/dev/sdaなどのSCSIデバイスノードを使用している階層にのみ影響します。

**DataKeeperのCreate Resourceが失敗します**

特定の環境（IDE ディスクエミュレーションを使用する仮想環境、HP CCISS ストレージを装備したサーバーなど）でDataKeeperを使用している場合、ミラーの作成時にエラーが発生することがあります。

```
ERROR 104052: Cannot get the hardware ID of the device “dev/hda3”
```

この原因は、LifeKeeperが該当ディスクを認識せず、デバイスに関連付けられている一意のIDを取得できないからです。

対応策: GUIDパーティションを使用することで該当ディスクを認識できるようになります。

**API アクセスに対するホスト名の指定**

LifeKeeperサーバー認証情報の格納に使用するキー名は他のLifeKeeperサーバーのホスト名と *完全に* 一致する必要があります（そのサーバーに対するhostnameコマンドで表示されます)。ホスト名がFQDNの場合、認証キーはFQDNである必要があります。ホスト名が短縮名の場合、キーも短縮名にする必要があります。

対応策: credstore によって格納されたホスト名がホスト名と完全に一致していることを確認します。

**リソースの作成後にlkbackupをリストアすると、破損したイクイバレンシが残されます**

作成したリソースの設定ファイルはlkbackup中に保存されます。lkbackupでバックアップした後に初めて作成したリソースは、このバックアップからリストアする際に適切に把握されない可能性があります。

**回避策**: 新しいリソースを初めて追加する前にlkbackupからリストアしてください。新しいリソースがlkbackupの後で追加された場合、リストアの前に削除するか、リソースの階層のインスタンスを削除し、リストアの後で階層を再拡張してください。 **注意** ：特定のリソースを初めて作成する際にlkbackupを実行することを推奨します。

**フェイルオーバー中にリソースが間違った順序で停止される**

リソース階層が別のルート階層と共通のリソースインスタンスを共有している場合、カスケードフェールオーバーまたはリソースフェールオーバー中に、リソースが間違った順序で停止されることがあります。

**回避策:** 共通ルートを作成すると、階層内のリソースの停止が上から下に確実に行われます。

1. restoreとremoveが常に成功するgen/appリソースを作成します。
2. 現在のすべてのルートをgen/appリソースの子リソースとして依存関係を作成します。

**注意:** restoreおよびremoveスクリプトに */bin/true* を使用することもできます。

**ネストされたファイルシステム階層を削除すると、メッセージ「Object does not exist」が生成されます**

**回避策** ：これは問題ではないので、メッセージを無視してかまいません。

**ネストされたマウントの作成時に`filesyshier` が正しくないタグを返します**

ネストされたファイルシステムのリソースがデータベースに存在する場合、ファイルシステムキットは、親とネストされた子の両方にファイルシステムを作成します。ただし、`filesyshier` は子のタグのみを返します。これにより、アプリケーションは子の依存関係を作成しますが、親の依存関係は作成しません。

**回避策** ：1つのマウントポイント内で複数のファイルシステムがネストされている場合、`dep_create` または UI の [Create Dependency]を使用して、親アプリケーションタグの追加の依存関係を手動で作成しなければならないことがあります。

**DataKeeper ：ネストされているファイルシステムの作成はDataKeeperで失敗します**

既存のファイルシステムを複製するためにDataKeeperミラーを作成するときに、ファイルシステムがこの構造内でネストされている場合、ファイルシステムをアンマウントしてからファイルシステムリソースを作成する必要があります。

**対応策** ：ネストされているファイルシステムを手動でアンマウントしてから、個々のネストされているマウントの再マウント / 作成を行ってください。

**ファイルシステムリソースに保護されたデバイスのマウントポイントの変更は、データ破損を引き起こす可能性があります**

ファイルシステムリソースに保護されたデバイスのマウントポイントは変更しないでください。変更した場合は、ファイルシステムリソースが仕様通りにデバイスを処理することができず、デバイスが２つのノードにマウントされて、データ破損を引き起こす可能性があります。

**XFSファイルシステム使用時に、quickCheckに失敗することがあります**

Red Hat Enterprise Linux7 / Oracle Linux7/ CentOS7にインストールしたLifeKeeperでCHECK_FS_QUOTAS設定を有効としたとき、保護するXFSファイルシステムリソースにuquota、gquotaオプションを設定すると、quickCheckに失敗します。

**回避策** ：XFSファイルシステムのマウントオプションにuquota、gquotaの代わりにusrquota、grpquotaをご使用ください。もしくは、CHECK_FS_QUOTAS設定を無効にしてください。

**Btrfsはサポートしません**

LifeKeeperをインストールするファイルシステム（/opt/LifeKeeper）にBtrfs（またはサポートされていないその他のLifeKeeper for Linuxファイルシステム）を使用することはできません。またビットマップファイルやその他 LifeKeeper に関連するファイルを置くファイルシステムにBtrfs（またはサポートされていないその他のLifeKeeper for Linuxファイルシステム）を使用することはできません。Btrfsファイルシステムをリソース保護することもできません。

**対応策:** Btrfsファイルシステムに /opt/LifeKeeper を配置するための簡単な回避策は、インスタンスに小さなディスクを追加し、ext4またはxfsでそのディスクをフォーマットし、このファイルシステムを /opt/LifeKeeper としてマウントすることです。

1. /opt/LifeKeeper に使用する小さなディスクを作成します
   - ソフトウェアのインストールには最低110MBが必要です
   - 注意：Azureでは、最低1GBのデータディスクを作成できます。
   - 注意：追加のARKとミラーの数により、必要な合計スペースが増加する場合があります。

2. ディスクがノードに追加されて表示されたら、ディスクをパーティションに分割します（または lvmを使用します）。
   - 例：`gdisk /dev/sdb`
   - 注意：システムにディスクを追加する方法は、このKBAの対象範囲外です（お使いの環境については、自社のシステム管理者にお問い合わせください）

3. サポートされているファイルシステムでパーティションをフォーマットします（詳細については、リリースノート を参照してください）。
   • 例：`mkfs.ext4 /dev/sdb1`（sdb1は手順2で作成）

4. 新しく作成され、フォーマットされたパーティションを /etc/fstab に追加し、システム起動時に自動的にマウントされるように設定します。

5. 新しいパーティションを /opt/LifeKeeper としてマウントします
   • 例：`mount /dev/sdb1 /opt/LifeKeeper`
   • ファイルシステムがマウントされていることを確認します

6. LifeKeeperをインストールします

7. インストール後、/etc/fstab を編集してエントリを追加し、再起動時にディスクをマウントできるようにします
   • 例: /dev/sdb /opt/LifeKeeper ext4

**AWS上でSLES12SP1以降を用いた場合に発生する問題**

・静的ルーティング設定ができない

/etc/sysconfig/network/routesに静的ルーティングが設定されている場合、DHCPにより自動で割り当てられたIPアドレスを用いた通信に失敗することがあります。

回避策: /etc/sysconfig/network/ifroute-ethXのROUTEを正しく設定し、ルーティング情報を更新してください。

・"Change Hostname via DHCP"が無効になっているにもかかわらず、ホスト名が変更される

LifeKeeperはホスト名が変更された場合正しく動作しません。AWS上で動作するSLES12SP1以降において、"Change Hostname via DHCP"が無効になっているにもかかわらず、ホスト名が変更されることがあります。

回避策:

- /etc/cloud/cloud.cfgにてupdate_hostnameをコメントアウトしてください
- /etc/cloud/cloud.cfg にて preserve_hostname にtrueを設定してください
- /etc/sysconfig/network/dhcpにてDHCLIENT_SET_HOSTNAMEをnoに設定してください

**Quorum/Witness KitをWitnessモードで利用している環境において、シャットダウンストラテジーを”Switchover Resources”に設定した場合**

Quorum/Witness KitをWitnessモードで利用している環境において、シャットダウンストラテジー

を”Switchover Resources”に設定した場合に、正しくスイッチオーバーが行われない場合があります。

**回避策**: シャットダウン前に手動での切り替えを行ってください。

**/etc/service ファイルの編集**

/etc/service にある以下の行を削除するとLifeKeeperが起動できなくなります。

```
lcm_server 7365/tcp
```

このファイルを編集する場合は、この行を削除しないように注意してください。

**SCSI ID にスペースを含む文字列を返すストレージユニットはLifeKeeperで保護できません**

**bind マウントはサポートしません**

LifeKeeperで保護するファイルシステムには、bindマウント(mount --bind)を使用できません。

AWSやAzure上で動作するバージョン12以降のSLES環境において、クラウドネットワークプラグインによって仮想IPアドレスが削除されるのを防ぐために、ネットワークインターフェイスの構成ファイルを変更する必要があります。

詳細は こちら を参照してください。

**$id の無効なサブディレクトリキャッシュが存在します**

これにより、$idがマウント解除されず、リソースの削除が失敗する可能性があります。これは、NFSv3エクスポートのサブディレクトリをマウントするときの既知の問題が原因である可能性があります。

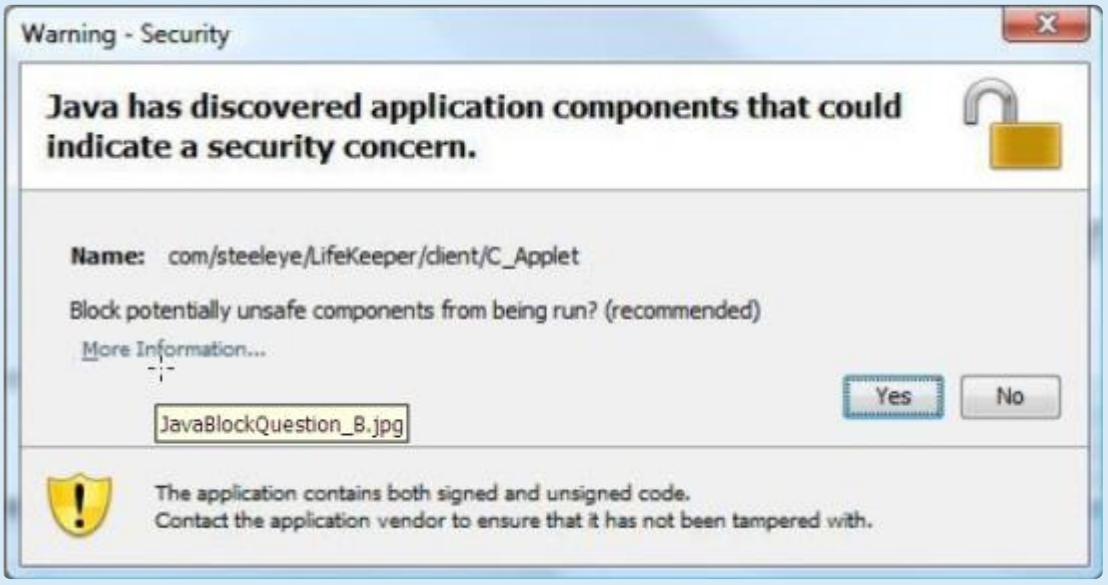
**回避策:** サブディレクトリがマウントされないようにリソースを再構成します。エクスポートされたディレクトリのみをマウントする必要があります。

# 5.4.5.3.3. インターネット / IP ライセンス - 既知の問題と制限

| 説明 |
| --- |
| ***/etc/hosts*** **の設定の依存関係**<br><br>**/etc/hosts 設定:**<br>インターネットベースのライセンス (IPv4 アドレス) を使用している場合、 */etc/hosts* の設定はライセンスの検証に悪影響を与える可能性があります。LifeKeeper の起動に失敗した場合は、以下のようなメッセージが出力されます。<br><br>`Error in obtaining LifeKeeper license key:`<br>`Invalid host.`<br>`The hostid of this system does not match the hostid specified in the license file.`<br><br>リストされているインターネットホスト ID が正しい場合、 */etc/hosts* の設定が原因の可能性があります。 */etc/hosts* エントリを正しく一致させるには、IPv6 エントリの前に IPv4 エントリを記載する必要があります。 */etc/hosts* 設定が原因かどうかを確認するには、次のコマンドを実行します。<br><br>`/opt/LifeKeeper/bin/lmutil lmhostid -internet -n`<br><br>記載された IPv4 アドレスが、インストールされたライセンスファイルの IPv4 アドレスと一致しない場合、正しいアドレスを返すために、 */etc/hosts* を変更し、IPv4 エントリを IPv6 エントリの前に配置する必要があります。 |

# 5.4.5.3.4. GUI – 既知の問題と制限

| 説明 |
| --- |
| **IP リソースでソースアドレスルーティングを使用すると、GUI に影響を与える可能性があります**<br><br>IP リソースでソースアドレスルーティングを使用すると、スイッチオーバーの実行時にリソースステータスを適切に表示する GUI の機能に影響を与える可能性があります。 実際には問題なく切り替えが完了しているのに、リソースの状態が「failed」または「unknown」と表示されることがあります。<br><br>**解決方法:**<br><br>この問題を解決するには、GUI を終了して再起動します。 |
| **ネットワーク切断から復旧後すぐはGUI で IP リソースの状態を更新できません**<br><br>クラスタ内のサーバ間のプライマリネットワークが切断され、再接続されると、RMI/TCP レイヤーの問題のため、リモート GUI クライアントの IP リソースの状態が更新されるまで 1 分 25 秒かかる場合があります。 |
| **Java 署名/未署名混合コードの警告 – LifeKeeper Java GUI クライアントアプレットをリモートシステムからロードすると、以下のセキュリティ警告が表示されることがあります**<br><br> <br><br><br>[Run] をクリックすると、以下のダイアログが表示されます。 |




ブロックするかどうかを確認するメッセージが表示されます。[No] をクリックすると、LifeKeeper GUI の動作が可能になります。

**推奨される処置:** セキュリティ警告の数を減らすには、2 つのオプションがあります。

1. **[Always trust content from this publisher]** ボックスをチェックし、 **[Run]** をクリックします。次に LifeKeeper GUI Java クライアントを読み込むときには、警告メッセージが表示されません。

   または

2. ブロックに関する 2 番目のダイアログが表示されないようにするには、以下のエントリを Java の「 **deployment.properties** 」ファイルに追加します。Java クライアントをロードしたときにセキュリティ警告は表示されますが、アプレットはブロックされず、ブロックするかどうかを **[Yes]** または **[No]** で確認する上記のダイアログは表示されなくなります。この設定はすべての Java アプレットに適用されることに注意してください。

   ```
   deployment.security.mixcode=HIDE_RUN
   ```

   上記のメッセージが両方とも表示されないようにするには、1 と 2 を実行してください。

**ポート 778･779 が使用中の場合、steeleye-lighttpd プロセスの開始に失敗します**

steeleye-lighttpd の起動時にプロセスがポート 778 もしくは 779 を使用している場合、steeleye-lighttpd の起動に失敗し、GUIへの接続障害が発生する。

**解決方法**: クラスタ内のすべてのノードで以下の設定を行い、LifeKeeper をすべてのノードで再起動します。

以下の行を */etc/default/LifeKeeper* に追加します。

```
API_SSL_PORT=port_number
LKAPI_WEB_PORT=port_number
```

port_number は使用する新しいポートです。

# 5.4.5.3.5. データレプリケーション – 既知の問題と制限

| 説明 |
| --- |
| **特定バージョンのカーネルでDataKeeperの同期に失敗する。**<br><br>*/var/log/messages* に以下のログが繰り返し出力されます。<br><br>`Apr 18 13:05:59 node1 nbd-client: Begin Negotiation`<br>`Apr 18 13:05:59 node1 nbd-client: size = 53684994048`<br>`Apr 18 13:05:59 node1 nbd-client: Negotiation Complete`<br>`Apr 18 13:05:59 node1 nbd-client: Ioctl/2 failed: Device or resource busy`<br>`Apr 18 13:05:59 node1 kernel: block nbd1: Device being setup by another task`<br><br>RHEL8.0~8.2, Oracle Linux UEK R5 (kernel-4.14.35-*) をご利用の場合、それぞれRHEL8.3以降、UEK R6以降にアップデートしてください。 |
| **再同期中に 3 番目のノードへの拡張が失敗する**<br><br>DKリソース (ミラー) では、再同期の進行中にミラーを 3 番目のノードに拡張しようとすると、再同期の進行中にミラーを「拡張」できないため、3番目のノードへの拡張は失敗します。<br><br>次のようなエラーが発生します。<br><br>removing hierarchy remnants<br>getId: /opt/LifeKeeper/lkadm/subsys/scsi/CPQARRAY/bin/getId -i “/dev/sdb1” returned “”<br>getId: /opt/LifeKeeper/lkadm/subsys/scsi/device/bin/getId -i “/dev/sdb1” returned “360022480e2671ceb01246bb1c5d67ebd-1”<br>**mdadm: /dev/md0 is performing resync/recovery and cannot be reshaped**<br>Failed to grow array (1) |
| **RHEL 8.6 / 9.0 のカーネルではシンプロビジョニングされたディスクで DataKeeper 非同期モードをサポートしません**<br><br>RHEL 8.6 / 9.0 のカーネルではカーネルパニックが発生することが確認されています。アップストリーム修正を含む更新されたカーネルを提供するために Red Hat と協力していますが、ディスクがシンプロビジョニングされている場合、RHEL 8.6 / 9.0 のカーネルでは非同期ミラーを構成しないことを推奨します。RHEL 8.6 / 9.0 のカーネルで LifeKeeper をインストールまたは更新すると、これについて警告が表示されます。 |
| **非同期モードで作成し同期モードで拡張した DataKeeper リソースの構成はサポートされません**<br><br>非同期モードで作成し同期モードで拡張した DataKeeper リソースでは、ミラーを読み書きしているプ |

ロセスがカーネル内でハングアップすることがあります。

DataKeeper リソースの同期モードの判定方法は、以下のコマンドを各ノードで実行してください。0 は同期モードで、非 0 は非同期モードです。全ノードで、全て同期モードもしくは全て非同期モードであれば問題ありません。いずれかのノードで、同期モードと非同期モードが混在している場合は、本問題が発生し得る構成です。

```
perl -nle 'my @x = split(/\x01/, $_); print "$x[0]:$x[3]";' /opt/LifeKeeper/subsys/scsi/resources/netraid/mirrorinfo_<md num>
```

**回避策**：直接回避する方法はありません。DataKeeper リソース階層を再作成して、作成･拡張時にどちらも同期モードを選択してください。

**カーネル 4.12 以降を実行している場合、総セクタ数が奇数になるパーティションはサポートされません**

総セクタ数が奇数になるパーティションの使用は、カーネル 4.12 以降を実行している環境の DataKeeper ミラーではサポートされていません。これは、ディスクの終端を超えて書き込みをしようとした場合に再同期が失敗するという問題に起因します。

**LVM 構成における DataKeeper for Linux 非同期モードに関する注意事項**

複数の非同期ミラー上に LVM を構成すると、データの整合性が保てず、さらには kernel panic が発生する場合があります。

DataKeeper 上に LVM を構成する場合には、DataKeeper ミラー 1 つだけ、または複数の同期ミラー、いずれかの構成でなければなりません。

**両方のサーバに重要な I/O トラフィックを持つシンメトリックなアクティブ SDR 設定で、netraid デバイス (ミラー) にマウントされたファイルシステムが応答を停止し、結果的にシステム全体がハングします**

Linux バッファキャッシュの単一スレッドの特性により、バッファキャッシュフラッシングデーモンは、リモートでコミットする必要があるバッファをフラッシュアウトしようとしてハングする可能性があります。フラッシングデーモンがハングすると、クリアされていないバッファの数がシステムで許容されている上限 (*/proc/sys/kernel/vm/bdflush* で設定) を超えると、クリアされていないバッファを持つ Linux システムのすべてのアクティビティが停止します。

これは、リモートシステムがリモートバッファを消去できなくなるような事態でないかぎり、通常は深刻な問題ではありません。LifeKeeper はネットワークの障害を検出し、そのときにレプリケーションを停止するので、ハング状態は消去されます。ただし、リモートシステムがローカルシステムにもレプリケートされた場合 (つまり、相互がシンメトリカルにレプリケートされた場合)、両方のシステムでこのフラッシングデーモンのハング状態が発生すると、永久にデッドロックする可能性があります。

デッドロックを解除するには、両方のシステムの nbd-client デーモンを手動で停止します (これにより、ミラーが切断されます)。ただし、このデッドロックを完全に防止する場合は、シンメトリックアクティブレプリケーションを推奨しません。

**大きなミラーサイズを備えた md_raid1 プロセスではトップで高い CPU 使用率がレポートされます**

`mdX_raid1` プロセス *(X はミラー番号)* では、非常に大きなミラー (500GB 以上) を操作している際に、一部の OS ディストリビューションで高い CPU 使用率がトップで報告されることがあります。

**解決方法**: CPU の使用率を減らすには、LifeKeeper 設定 `LKDR_CHUNK_SIZE` でチャンクサイズを 1024 に変更し、この新しい設定を使用するためにミラーを削除して再作成します。

**DataKeeper リソースで `lkbackup` を使用する場合は、全同期が必要です**

`lkbackup` では *`instance`* と *`mirror_info`* ファイルを保存しますが、リソースが存在しないときにソースおよびターゲットの状態は保証されないため、`lkbackup` から restore した後で、DataKeeper ミラーの全同期を実行することがベストプラクティスです。

**SLES12 SP1 以降でデータ圧縮機能を使用することはサポートしていません**

パフォーマンスの問題があるため、SLES12 SP1 以降でデータ圧縮機能を使用することはサポートしていません。

**特定バージョンのカーネルでは DataKeeper の非同期モードはサポートしていません。**

LifeKeeper for Linux において DataKeeper リソースの非同期モード設定を使用すると、特定バージョンのカーネルでカーネルパニックが発生することが確認されています。本事象はカーネルに依存した問題であるため、LifeKeeper による根本的な解決方法はありません。DataKeeper の非同期モード設定をご利用いただくには、カーネルのアップデート、またはダウングレードが必要となります。

DataKeeper の非同期モードを利用できないカーネルバージョンは次の通りです。

Red Hat Enterprise Linux / Oracle Linux / CentOS において、以下のバージョンのカーネルを使用している場合
- 3.10.0-693.24.1.el7.x86_64 以降の 3.10.0-693. 系カーネル
- 3.10.0-862.el7.x86_64 から 3.10.0-862.26.x.el7.x86_64 まで
- 3.10.0-957.el7.x86_64 から 3.10.0-957.3.x.el7.x86_64 まで

上記に該当する環境をご利用で、DataKeeper リソースを非同期モードでご利用の場合には以下のカーネル バージョンへのアップデート（またはダウングレード）を実施してください。

Red Hat Enterprise Linux / Oracle Linux / CentOS において、以下のバージョンのカーネルを使用している場合
3.10.0-693.24.1.el7.x86_64 未満 の3.10.0-693. 系カーネル
3.10.0-862.29.1.el7.x86_64 以降
3.10.0-957.4.1.el7.x86_64 以降

カーネルのアップデート（またはダウングレード）が不可能である場合には、DataKeeper では非同期モードを利用しないようにしてください。

---

一部のカーネルバージョンでは **Secure Boot** 機能をサポートしていません

RHEL7 以降、CentOS7 以降、Oracle Linux7(非 UEK カーネルのみ) 以降で Secure Boot を有効にしていると、nbd モジュールのロードに失敗します。また、SUSE Linux Enterprise Server、Oracle Linux UEK kernel においても、一部のカーネルバージョンでは、Secure Boot を有効にしていると、md/raid1 カーネルモジュールのロードに失敗します。

**解決方法:** 以下のいずれかの対応を行ってください。(1) を推奨します。また、いずれの方法も OS 再起動が必要です。
(1) Secure Boot を無効にする。
UEFI の設定で、Secure Boot を無効にしてください。
(2) 署名の検証を無効にする。
“mokutil --disable-validation” コマンドを実行して、署名の検証を無効にしてください。詳細は mokutil のマニュアルを参照ください。

---

コミュニケーションパス復旧時にミラーリングが停止することがあります

通信障害から回復し再同期が行われる際、同期のためのプロセスが複数起動します。タイミングによっては、一方のプロセスの処理により、もう一方のプロセスの初期化がブロックされるため、再同期が停止状態となります。

通信障害から回復後、以下のような状況を全て満たす場合、本事象に該当します。

- DataKeeperリソースの同期状態が「Out of Sync」のままである
- /var/log/lifekeeper.logに、以下のログが quickCheck の間隔で出現し続けている
  `INFO:lkcheck:::006041:recover is being in progress: [DataKeeperリソースタグ名]`
- recover プロセスが1つ以上起動したままになっている。例えば、以下のコマンドで確認ができます。
  `top | grep recover`

**対処策 :**

DataKeeper リソースが In Service になっているノードを再起動する。

それでも直らない場合は、サポートにお問い合わせください。

# 5.4.5.3.6. IPv6 - 既知の問題と制限

| 説明 |
| --- |
| SIOS は、 ifconfig コマンドから ip コマンドの使用に移行しました。この変更のため、外部スクリプトを使用するお客様も、同様の変更を行うことを推奨します。 ifconfig コマンドを発行し、結果を解析して特定のインターフェースを探す代わりに、スクリプトは「ip -o addr show」を使用し、結果を解析して、「inet」および「secondary」という語を含む行を検索します。<br><br>`# ip -o addr show`<br>`1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN`<br>`\ link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00`<br>`1: lo inet 127.0.0.1/8 scope host lo`<br>`1: lo inet6 ::1/128 scope host`<br>`\ valid_lft forever preferred_lft forever`<br>`2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state U P qlen 1000`<br>`\ link/ether d2:05:de:4f:a2:e6 brd ff:ff:ff:ff:ff:ff`<br>`2: eth0 inet 172.17.100.77/22 brd 172.17.103.255 scope global eth0`<br>`2: eth0 inet 172.17.100.79/22 scope global secondary eth0`<br>`2: eth0 inet 172.17.100.80/22 scope global secondary eth0`<br>`2: eth0 inet6 2001:5c0:110e:3364::1:2/64 scope global`<br>`\ valid_lft forever preferred_lft forever`<br>`2: eth0 inet6 2001:5c0:110e:3300:d005:deff:fe4f:a2e6/64 scope global dynam ic`<br>`\ valid_lft 86393sec preferred_lft 14393sec`<br>`2: eth0 inet6 fe80::d005:deff:fe4f:a2e6/64 scope link`<br>`\ valid_lft forever preferred_lft forever`<br><br>ip コマンドの上記の結果では、以下の行に eth0 インターフェースの仮想 IP アドレスが含まれます。<br><br>`2: eth0 inet 172.17.100.79/22 scope global secondary eth0`<br>`2: eth0 inet 172.17.100.80/22 scope global secondary eth0` |
| ***/etc/sysconfig/network-scripts/ifcfg-<nicName>*** **の「IPV6_AUTOCONF = No」が再起動または起動の際に考慮されません**<br><br>起動時に、自動設定されたステートレスな IPv6 アドレスがネットワークインターフェースに割り当てられます。IPV6_AUTOCONF=No が設定されているインターフェースのステートレス IPv6 アドレスでコミュニケーションパスが作成された場合、任意のシステムリソースが ifdown <nicName>;ifup <nicName> などのインターフェースを管理する際にアドレスが削除されます。<br><br>IPV6_AUTOCONF が No に設定されていたため、プライマリサーバを再起動した後で、自動設定された |

IPv6 アドレスを使用しているコミュニケーションパスは復旧せず、無効のままでした。

**解決方法**: スタティックな IPv6 アドレスのみを使用してください。自動設定された IPv6 アドレスを使用すると、再起動後に通信が失われたり、NIC が変更されたりする可能性があります。

自動設定された IPv6 アドレスをコミュニケーションパスの作成に使用できますが、システム管理者は以下の条件を認識する責任があります。

- 自動設定されたステートレス IPv6 アドレスがネットワークインターフェース (NIC) の MAC アドレスに準拠していること。コミュニケーションパスが作成され、関連 NIC が後で置き換えられた場合、自動設定された IPv6 アドレスは異なるものになり、LifeKeeper はコミュニケーションパスが無効になっていることを正しく表示します。コミュニケーションパスを再作成する必要があります。
- Red Hat Enterprise Linux では、ホスト操作のあらゆる側面で一貫した IPv6 自動設定を確保するための動作を実行するには、sysctl.conf、net.ipv6.* 命令 (‘if/ip’ ユーティリティで参照される ifcfg-<nic> の明示的な IPV6_AUTOCONF 設定、およびシステムが起動して init レベルを切り替える際に NIC 制御に影響する /etc/sysctl.conf) に加え、個々のインターフェース設定ファイルを正確に設定するために、詳細かつ具体的なドメインの知識が必要になります。

**IP:IPv6 のソースアドレス変更設定ではソースアドレスが設定されません**

IPv6 IP リソースのソースアドレスを設定しようとすると、何も変更されていない場合でも成功となります。

**対応策**: 現在のところ、対応策はありません。今後のリリースで対応する予定です。

**IP:無効な IPv6 アドレス設定が IP リソース作成で許可されます**

オクテットに 4 文字を超える文字が含まれている場合、2001:5c0:110e:3368:000000:000000001:61:14 という形式の IPv6 アドレスが許容されます。

**対応策**: 正しい形式の IPv6 アドレスを入力してください。

**bonding NIC に割り当てられているものの、「暫定的な」状態のアドレスでは、IPv6 リソースが ISP としてレポートされます**

LifeKeeper の IPv6 保護リソースは、IPv6 リソースが bonding インターフェース上にある SLES システムでは「In Service Protected」(ISP: in service の保護) と不正に識別されます。これは、‘active-backup’ (1) および 2.6.21 以前の Linux カーネルとは別のモードです。IPv6 の bonding リンクは、解決できないアドレスを持つ「暫定的な」状態のままになります。

**対応策:** bondingインターフェースモードを ‘active-backup’ (1) に設定します。または、‘active-backup’ (1) 以外のモードの場合、リンク状態を「tentative (暫定的)」から「valid (有効)」に設定する更新したカーネルで操作します。

# 5.4.5.3.7. Apache - 既知の問題と制限

| 説明 |
| --- |
| **Apache** キットは **IPv6** をサポートしていません。 ***httpd.conf*** で **IPv6** を識別しません<br><br>*httpd.conf* ファイルの「Listen」命令エントリに IPv6 アドレスを割り当てると、問題が発生します。<br><br>**解決方法**: Apache Recovery Kit で IPv6 がサポートされるまで、リソース作成後に IPv6 アドレスを *httpd.conf* ファイルに指定できません。 |

# 5.4.5.3.8. Oracle Recovery Kit – 既知の問題と制限

| 説明 |
| --- |
| **Oracle Recovery Kit には Connection Manager および Oracle Names 機能のサポートが含まれていません**<br><br>LifeKeeper Oracle Recovery Kit には、Oracle Connection Manager と Oracle Names という Oracle Net 機能のサポートが含まれていません。Oracle Connection Manager は、同じサービスにアクセスする必要がある多数の接続を管理するルーティングプロセスです。Oracle Names はサービスアドレスの一括格納を管理する Oracle 固有の命名サービスです。<br><br>LifeKeeper Oracle Recovery Kit は、送信されてきたクライアント接続要求をリスニングし、サーバへのトラフィックを管理する Oracle Net Listener プロセスを保護します。Oracle Listener に固有の LifeKeeper 設定情報については、『 *LifeKeeper for Linux Oracle Recovery Kit 管理ガイド* 』を参照してください。 |
| **Oracle Recovery Kit は ASM またはグリッドコンポーネント機能をサポートしていません**<br><br>Oracle で提供されている Oracle Automatic Storage Manager (ASM) 機能は現在、LifeKeeper ではサポートされていません。 また、グリッドコンポーネントは LifeKeeper Oracle Recovery Kit によって保護されません。Raw デバイス、ファイルシステム、論理ボリュームに対するサポートは、現在の LifeKeeper for Linux Oracle Recovery Kit に含まれています。グリッドコンポーネントに対するサポートは、gen/app リカバリキットを使用して LifeKeeper の保護機能に追加できます。 |
| **Oracle Recovery Kit は NFS バージョン 4 をサポートしていません**<br><br>Oracle Recovery Kit は共有データベースストレージ用に NFS バージョン 3 をサポートしています。NFSv4 ファイルロッキングメカニズムのため、NFS バージョン 4 は、現時点ではサポートされていません。 |
| **フェイルオーバ後にプライマリサーバで Oracle Listener が in service のままになります**<br><br>ネットワーク障害により、アプリケーションがバックアップサーバにフェイルオーバした後、プライマリサーバで Listener プロセスがアクティブなままになります。正しいデータベースへの接続は影響を受けませんが、その Listener プロセスを終了したいと考えるかもしれません。 |
| **DataKeeperでファイルシステムのネスト作成に失敗する**<br><br>既存のファイルシステムを複製するためにDataKeeperミラーを作成する場合、この構造内にファイルシステムがネストされている場合は、ファイルシステムリソースを作成する前に、まずそれをアンマウントする必要があります。<br><br>**適切な手順**: oracle/BPP ファイルシステムを作成してから、その下に /oracle/BPP/mirrorlogA のようなサブファイル システムを作成します。 (すなわち、/oracle/BPP ファイルシステムが GUI リストの一番下に表示されるはずです。)<br><br>**回避策:** ネストしたファイルシステムを手動でアンマウントし、各ネストしたマウントを再マウント/作成します。 |

# 5.4.5.3.9. MySQL - 既知の問題と制限

| 説明 |
| --- |
| **include 文をサポートしません**<br><br>MySQL Recovery Kit は、my.cnf 設定ファイルにおいて include 文をサポートしておりません。 |
| クラッシュリカバリについて<br><br>MySQL リカバリキットは、MySQL がクラッシュリカバリ処理を行っている間、それを異常と判断して回復処理を起動します。 そのため、両者の処理が競合してしまい、MySQL のクラッシュリカバリが正常に行われない場合があります。<br>この現象は、リカバリ中にクライアントからの接続を受け付けなくなるために、LifeKeeper の監視処理で行われる接続確認が失敗するために発生します。 |

# 5.4.5.3.10. NAS Recovery Kit – 既知の問題と制限

| 説明 |
| --- |
| **autofs** はサポートされていません<br><br>autofs によって管理されるファイルシステムを保護することはできません。 |

# 5.4.5.3.11. NFS Server Recovery Kit – 既知の問題と制限

| 説明 |
| --- |
| **$id の無効なサブディレクトリ キャッシュがまだ存在します**<br><br>詳細については LifeKeeper Core – 既知の問題 を参照してください。 |
| **最上位の NFS リソース階層は hanfs リソースのスイッチバックタイプを使用します**<br><br>障害からサービス状態に復旧する際に NFS リソース階層がプライマリーサーバーにスイッチバックするかどうかを制御するスイッチバックタイプは、hanfs リソースで定義されます。 |
| **クライアントの指定に IPv6 アドレスを使用することはできません**<br><br>/etc/exports ファイルにおいて IPv6 アドレスを使用してクライアントを指定した場合、リソース作成に失敗します。<br><br>**解決方法**: クライアントの指定にはホスト名やワイルドカードを使ってください。 |
| **ファイルロックのスイッチオーバーが失敗します**<br><br>リソースのスイッチオーバーまたはフェイルオーバーの際、ファイルロックのスイッチオーバーが機能しません。これは、NFS v3/v4 のどちらのバージョンでマウントしても同じです。<br>クライアントのアプリケーションで、ファイルロックを使用しないでください。 |
| **gssproxy が起動していない場合、リソースの作成処理がハングアップします**<br><br>gssproxy デーモンが起動していないと、リソース作成時に LifeKeeper が gssproxy デーモンを起動します。しかし、gssproxy が LifeKeeper の期待する動作を行わないために、リソース作成の処理が終了しません。本問題は、lkGUIapp でも lkcli どちらでも発生します。<br>なお、gssproxy がインストールされておらず、rpc.svcgssd がインストールされている場合は、rpc.svcgssd を使用するため本問題は発生しません。<br><br>**解決方法**: gssproxy デーモンを起動してください。また、常に起動させておくために、自動起動を有効にしてください。以下、RHEL7 の場合のコマンド例です。<br>`systemctl start gssproxy.service`<br>`systemctl enable gssproxy.service` |

**LifeKeeper Single Server Protection で rpc.idmapd および rpc.svcgssd が起動していない場合、restore の処理が失敗します**

LifeKeeper Single Server Protection では rpc_pipefs をマウントするためのディレクトリーが作成されません。rpc.idmapd および rpc.svcgssd デーモンを起動する前に rpc_pipefs をマウントしますが、ディレクトリーがないために失敗します。

**解決方法**: rpc.idmapd および rpc.svcgssd デーモンを起動してください。また、常に起動させておくために、自動起動を有効にしてください。以下、SLES15 の場合のコマンド例です。
なお、gssproxy がインストールされている場合は、rpc.svcgssd デーモンを起動しておく必要はありません。

```
systemctl start nfs-idmapd.service
systemctl enable nfs-idmapd.service
systemctl start rpc-svcgssd.service
systemctl enable rpc-svcgssd.service
```

# 5.4.5.3.12. SAP Recovery Kit – 既知の問題と制限

## SAP Recovery Kit

| 説明 |
| --- |
| **コミュニケーションパス復旧時に ERS v2 リソースに対してスプリットブレインが発生する可能性があります**<br><br>SAP クラスターで Quorum が構成されていて、ASCS と ERS v2 リソースの両方が同じノードに存在する場合、コミュニケーションパスが復旧する際に ERS v2 リソースの切り替えが発生するとスプリットブレインになる場合があります。本件に関するパッチについては製品サポートへお問い合わせください。 |
| **RHEL 8 の csh バグのため、SAP リソースが In Service になりません**<br><br>RHEL 8 で利用可能な tcsh パッケージのバグにより、特定の状況では、SAP管理ユーザーの .cshrc および .login ファイルを正しく読み込めません。このため、SAP Recovery Kit が依存する重要な環境変数が適切にエクスポートされず、SAP リソースがRHEL 8 で In Service にならない可能性があります。詳細とに解決方法ついては、RedHat Bug 1714267 を参照してください。 |
| **SAP デュアルスタック環境の制限**<br><br>SPS-L 9.4.0（対応するセントラルサービスインスタンスとは別の階層で動作）で実装された新しい設計のERSリソースタイプ は、同じSIDでセントラルサービスとエンキューレプリケーションサーバーインスタンスのペアが2組インストールされている（例：ASCS00/ERS10 および SCS01/ERS11SAPなど）SAPデュアルスタック環境（ABAP + Java）をサポートしません。<br><br>同じSIDでインストールされたSAPデュアルスタック（ABAP + Java）環境を利用しているユーザーは、9.4.0より前のERSリソース設計（対応するASCS/SCSリソースに依存するSAP階層の最上位に位置）を引き続き使用する必要があります。 |
| **SAP 階層の削除または拡張解除に失敗します**<br><br>同じ IP リソースを階層内の複数の場所に格納している SAP 階層を削除または拡張解除すると、リソースが削除されず、コアダンプが発生することもあります。<br><br>この問題を修正するには、拡張解除または削除操作に失敗した後で、残ったリソースを LifeKeeper GUI から手動で削除します。サーバからコアファイルを削除するという方法もあります。 |
| **[Handle Warnings] により -e 行 1 で構文エラーが発生します**<br><br>**[Handle Warnings]** のデフォルトの動作 **[No]** を **[Yes]** に変更すると、エラーを受信します。<br><br>**解決方法**: このオプションをデフォルト設定の **[No]** のままにしてください。 **注記**: 黄は過渡的な状態であり、ほとんどの場合は障害を表すものではないため、この設定はデフォルト選択の **[No]** のままにすること |

| |
|---|
| を強くお勧めします。 |
| **同じ設定を選択すると、Update Wizard のボタンが非表示になります**<br><br>現在の設定を変更せずに **[Handle Warnings]** を更新しようとすると、戻る必要があることを示す次の画面で **[Done]** ボタンが表示されません。 |
| **res_state に変更を加えると、監視が無効になります**<br><br>**[Protection Level]** を **[BASIC]** に設定し、SAP を手動でダウンさせた場合 (メンテナンスなどの目的で)、「FAILED」とマークされ、監視が停止します。<br><br>**解決方法**: 監視を再開する場合、手動で起動するのではなく LifeKeeper がリソースを起動する必要があります。 |
| **ERS が Core/CI の親ではない場合、サービス中の ERS がリモートホストで失敗します**<br><br>**注記**: これは、9.4.0 より前の ERS リソース設計（ ERS リソースが、対応する ASCS/SCS リソースに依存する SAP階層の最上位に存在）にのみ適用されます。詳細については、 LifeKeeperのERSリソースタイプ を参照してください。<br><br>追加 SAP リソースの依存関係なしで ERS リソースを作成すると、スイッチオーバ時に初期のサービス中状態が失敗します。<br><br>**解決方法**: CI/Core インスタンス (SCS または ASCS) の親として ERS を作成してから、in-service を再試行します。 |
| **SAPインスタンスプロセスの状態が一致しなくなる**<br><br>SAPリソースの移行中に並行して管理コマンドを発行すると、SAPインスタンスプロセスの状態が一致しなくなる可能性があります。これを解決するには、手動による介入が必要になる場合があります。 |

# 5.4.5.3.13. LVM Recovery Kit – 既知の問題と制限

| 説明 |
| --- |
| **DataKeeper上のLVMを保護している場合、ボリュームグループの再設定 で案内している手法によるリソース階層の自動修正はできません。**<br>詳細は ボリュームグループの再設定 をご覧ください。 |
| **LVM 構成における DataKeeper for Linux 非同期モードに関する注意事項**<br><br>複数の非同期ミラー上に LVM を構成すると、データの整合性が保てず、さらには kernel panic が発生する場合があります。<br>DataKeeper 上に LVM を構成する場合には、DataKeeper ミラー 1 つだけ、または複数の同期ミラー、いずれかの構成でなければなりません。 |
| **lkID の使用はディスク全体で LVM を上書きした状態と互換性がありません**<br><br>lkID を使用して、LVM 物理ボリュームとして設定されているディスクで一意のディスク ID を生成すると、lkID および LVM 情報が格納されているディスク上の場所で競合が発生します。これにより、lkID および pvcreate が使用された順番に従って、lkID または LVM 情報のどちらかが上書きされます。<br><br>**対応策**: lkID を LVM と組み合わせて使用する必要がある場合は、ディスクをパーティション化し、ディスク全体ではなくディスクパーティションを LVM 物理ボリュームとして使用します。 |

# 5.4.5.3.14. Multipath Recovery Kits (DMMP / HDLM / PPATH /NECSPS) - 既知の問題と制限

| 説明 |
| --- |
| Multipath Recovery Kit（DMMP / HDLM / PPATH /NECSPS）では、マルチパスリソースの状態がOSFの場合、LifeKeeperをスタンバイ（OSU）ノードで停止した時に、アクティブ（ISP）ノードでシステム停止が発生することがあります。<br><br>回避策：<br><br>a）LifeKeeperを停止する前に階層をスタンバイノードに切り替える<br><br>または<br><br>b）LifeKeeperを停止する前にスタンバイノードで `ins_setstate`を実行し、マルチパスリソースの状態をOSUに設定する |

# 5.4.5.3.15. DMMP Recovery Kit – 既知の問題と制限

| 説明 |
| --- |
| **DMMP:スタンバイサーバーで発行された write がハングすることがあります**<br><br>別のサーバーでリザーブされている DMMP デバイスに write が発行されると、IO が永久に (またはデバイスがもう一方のサーバーでリザーブされなくなるまで) ハングすることがあります。 もう一方のサーバーでデバイスが解放され、write が発行されると、データが破損することがあります。<br>この問題の原因は、DMMP での IO 再試行に従ってパス確認が実行される方法にあります。<br>「no_path_retry」が 0 (失敗) に設定されている場合、このハングは発生しません。 別のサーバーでパスがリザーブされているときにデバイスの path_checker が失敗しても (MSA1000)、この問題は発生しません。<br><br>**対応策**:「no_path_retry」を 0 (失敗) に設定します。 しかしこれにより、一時的なパスの障害が原因で、IO の障害が発生する可能性もあります。 |
| **DMMP:複数のイニシエータが ATP_C をサポートする SAS アレイで正しく登録されていません**<br><br>LifeKeeper は、複数の SAS イニシエータが SAS アレイに接続される設定をネイティブにサポートしていません。 こうした設定では、各イニシエータが正常に登録されないので、1 つのイニシエータのみが IO を発行できるようになります。 マルチパスドライバ (DMMP など) が未登録のイニシエータに IO を発行すると、エラーが発生します。<br><br>**解決方法**: SAS ストレージ情報に基づいてパス ID が設定されるように、`/etc/default/LifeKeeper` の設定値を以下のように設定してください。<br><br>`MULTIPATH_SAS=TRUE` |
| **DMMP リカバリーキットのパラメータ設定が必要な場合、複数の異なるストレージを同時に使用することはできません** |
| **末尾が "p<数字>" になっている LV と併用できません**<br><br>LVM を併用する場合、末尾が "p<数字>"になっている LV が存在すると DMMP RK が正常に動作しません。 このような名前を持つ LV は作成しないでください。 |

# 5.4.5.3.16. DB2 Recovery Kit – 既知の問題と制限

| 説明 |
| --- |
| **DB2 Recovery Kit** が不要なエラーをレポートします<br><br>DB2 が共有ディスクにインストールされている場合、DB2 リソースを拡張する際に以下のメッセージが表示されることがあります。<br><br>`LifeKeeper was unable to add instance "%s" and/or its variables to the DB2 registry.`<br><br>このメッセージは、DB2 リソースの拡張動作には悪影響を及ぼしません。 |

# 5.4.5.3.17. Sybase ASE Recovery Kit – 既知の問題と制限

| 説明 |
| --- |
| ユーザー名 / パスワードの問題:<br><br>• デフォルトのユーザー名がパスワードで保護されている場合、すべての検証が完了するまで **create UI** はこれを検出しません<br><br>Sybase リソースを作成する際に、ユーザー名の入力が要求されます。フロントのヘルプには、ユーザーが指定されない場合はデフォルトの「sa」が使用されるというメッセージが表示されます。ただし、この時点ではデフォルトのアカウントに対するパスワード検証は実行されません。LifeKeeper for Linux が Sybase リソースを作成しようとする際に、パスワードの検証 / 入力が行われていないためにリソース作成が失敗します。パスワード検証は [user/password] ダイアログで発生しますが、ユーザープロンプトに有効なユーザーが実際に入力された場合のみです。デフォルトのユーザー名を使用する場合でも、 **create** 操作時にそのユーザー名を指定する必要があります。<br><br>• ユーザー名が指定されない場合、パスワードの入力要求が省略されます<br><br>ユーザー名を入力しない場合、[user/password] ダイアログはパスワードの入力要求を省略します。UI オプションを使用してユーザー / パスワードを更新する際に Sybase のユーザー名を入力しない場合、デフォルトの「sa」が使用され、そのアカウントのパスワード検証は実行されません。これにより、無効な認証情報のエラーが発生して、データベースの監視が失敗します。デフォルトのユーザー名を使用する場合でも、update 操作時にそのユーザー名を指定する必要があります。この問題を解決するには、以下の手順を実行してください。<br><br>1. 必要な Sybase データファイルが、目的のサーバーから現在アクセス可能であることを確認します。多くの場合、プライマリ上のローカルリカバリの障害を監視するため、これはバックアップサーバーです。<br><br>2. このサーバーのコマンドラインから Sybase データベースインスタンスを開始してください (データベースを手動で開始する方法の詳細は、Sybase 製品のドキュメンテーションを参照)。<br><br>3. コマンドラインを使用して、LKROOT/bin ディレクトリ (/opt/LifeKeeper/bin) に移動 (cd) してください。<br><br>4. bin ディレクトリで次のコマンドを実行してください。<br>./ins_setstate -t <SYBASE_TAG> -S ISP<br><SYBASE_TAG>は Sybase リソースのタグ名です。<br><br>5. コマンドの実行が完了した後、ただちに UI から **Update User/Password Wizard** を実行 |

し、有効なユーザー名を入力してください (Sybase のデフォルトの「sa」を使用する場合も同様)。
**注意**：**Update User/Password Wizard** にアクセスするには、Sybase リソースインスタンスを右クリックし、 **[Change Username/Password]** を選択してください。

6. ローカルサーバー上で階層が更新された後、リソースがすべてのノードで in service にすることができることを確認してください。

- **Sybase のローカルユーザー名が 8 文字以上の場合、バックアップサーバーの保護が失敗します**

  Sybase のユーザー名は 8 文字未満にする必要があります。Sybase のローカルユーザー名が 8 文字以上の場合、リソース作成に使用されるプロセスとユーザー ID のチェック、および監視が失敗します。また、これにより、有効な Sybase Backup Server のインスタンスを保護の対象として選択できなくなります。この問題は、オペレーティングシステムが 8 文字以上のユーザー名を各種コマンド (ps など) の UID に変換することに起因します。7 文字以下のユーザー名を使用する必要があります。

**リソース作成問題:**

- Sybase のデフォルトのインストールプロンプトは、16.0 SP02(/opt/sybase) に基づきます。LifeKeeper for Linux でリソースを作成する際に、Sybase のインストール場所のデフォルトプロンプトが、Sybase バージョン 16.0 SP02(/opt/sybase) からの相対パスとして表示されます。リソース作成のプロンプトには、Sybase の正しいインストール場所を手動で入力するか、参照する必要があります。

**拡張の問題:**

- **拡張時に Sybase のタグプロンプトが編集可能ですが変更しないでください** 拡張時に Sybase のタグプロンプトが編集可能ですが、これは推奨されません。サーバーごとに異なるタグを使用すると、コマンドラインによるリモート管理で問題が発生するおそれがあります。

**[Properties] ページの問題:**

- **[Properties] ペインに [update user/password] の画像が表示されません** 正しい画像の代わりに小さい四角がツールバーに表示されます。この四角を選択すると、 **User/Password Update Wizard** が起動されます。

**Sybase Monitor Server** は、Sybase 15.7 と LifeKeeper for Linux の組み合わせではサポートされていません。Sybase 15.7 で Sybase Monitor Server プロセスを設定する場合、Generic Application (gen/app) リソースを使用してこのサーバープロセスを保護する必要があります。

**Sybase RK を使用してリソースを作成できない、または SLES 15 を使用している場合に LifeKeeper のアップグレードで問題が発生する**

**症状:**
SLES 15（SLES 15 SP1 または SP2 を含む）を設定する場合、LifeKeeper のアップグレードまたは新規

インストール時に Sybase RK が動作しない場合があります。

LifeKeeperの新規インストール中は、Sybaseリソースを作成できません。また、LifeKeeperのアップグレード中に、LifeKeeperからのリソース制御で問題が発生する可能性があります。この問題の元になっているエラーは、「An error occurred when attempting to allocate localization-related structures（ローカリゼーション関連の構造を割り当てようとした際にエラーが発生しました）」です。

**解決方法:**
解決方法は2つあります。

1. ファイル */sybase/<SID>/locales/locales.dat* の [Linux] セクションに、次の行を追加します。
   `locale = POSIX, us_english, utf8`

   または、

2. 次の行を、 *syb<SID>* ユーザープロファイルの Sybase プロファイルに追加します。このプロファイルは通常、 */sybase//SYBASE.sh* にあります。
   `export LANG=en_US.UTF-8`

   <SID> は 3文字のシステム ID です。

   これらの変更により強制的に正しい言語が使用されるようになり、米国英語のロケールに適したものになります。日本や中国で別のロケールを使用している場合は、現地の国やシステムのロケールを確認してください。

   また、 syb<SID> ユーザーに su して `echo $LANG` コマンドを実行することもできます。cshell パラメーターは、適切なロケールの正しい言語を指します。

   一方の方法ではシェルスクリプトを変更し、他方ではロケールに POSIX サポートを追加しています。Sybase (SAP ASE) データベース管理者のやりやすいほうで変更してください。

**Sybase ARK の稼働を検出できません**

**症状：** Sybase ARK の稼働を検出できません

**原因：** Sybase ARK はデフォルトの sql interface tool (isql) を使用します。一部の 64 ビットシステム上では、この isql toolは、isql ではなくisql64 としてインストールされています。

**解決方法：** isql64 ツールを同一パスにおいて isql としてコピーしてください。もしくは isql64 に対して isql という名前でリンクを作成してください。

# 5.4.5.3.18. WebSphere MQ Recovery Kit – 既知の問題と制限

| 説明 |
| --- |
| **lksupport**コマンド実行時のエラー<br><br>MQ RK の保護下にある MQ キューマネージャーが NFS 共有されているディスク上に配置されている場合、 lksupportコマンドを実行すると、以下のエラーが出力されることがあります。<br><br>`cat: <PATH>/mqm/qmgrs/tkqmgr/qm.ini: Operation not permitted`<br><br>これは、NFS 領域への root アクセスが禁止されていることにより発生するものです。 このエラーが出力されることにより発生する問題はありません。 |
| **quickCheck** がモニタしているキューに長いメッセージ**(101**文字以上**)**を書き込まれた場合、異常と判定してしまいます |
| **MQ** 標準のディレクトリ以外にインストールされている場合、リカバリキットのインストールに失敗します |
| **MQ** のパッケージ名を変更した場合、リカバリキットのインストールに失敗します |
| **MQ** が標準のディレクトリ以外にインストールされている場合、サンプルのビルドに失敗します |
| 複数のリスナーを一つのインスタンスに定義し、自動起動と手動起動が混在している場合、リソース作成および **quickCheck** に失敗することがあります |

# 5.4.5.3.19. SAP HANA – 既知の問題と制限

| 説明 |
| --- |
| **v9.7.0 より古い LifeKeeper for Linux を実行している場合、リソースはセカンダリーノードでサービスを開始できません。**<br><br>**問題:** セカンダリーノードが v9.7.0 以降にアップデートされ、フェイルオーバーまたはスイッチオーバーが発生してクラスター内のプライマリーノードになると、他のすべてのノードの構成が更新されます。v9.7.0 より前のバージョンの LifeKeeper for Linux を別のノードで実行している場合、以前のバージョンでは新しい構成フォーマットがサポートされていないため、サービスを開始できません。<br><br>**回避策 / 解決策:** セカンダリーノードを v9.7.0 以降にアップグレードして、通常のフェイルオーバー / スイッチオーバー操作を可能にします。 古いリリースでリソースを稼働させるには、バックアップから構成を復元します。詳細については、 lkbackup を参照してください。 |
| **ローカルデータベースが sapcontrol Stop/StopWait コマンドで停止された場合、ローカルリカバリーが失敗し、SAP HANA リソース階層のフェイルオーバーが発生する場合がある**<br><br>**問題**: ユーザーが sapcontrol ユーティリティーを使用して HDB インスタンスに対して **Stop/StopWait** 要求を発行すると (これは、ユーザーが「HDB stop」コマンドを発行したときに内部的に使用されるものでもあります)、sapstartsrv はすべてのインスタンスを適切に停止する非同期プロセスを開始します。 データベースが完全にシャットダウンされるか、プロセスがタイムアウトするまで、HANA データベースはプロセスを停止しません。 したがって、sapstartsrv がデータベースを正常にシャットダウンしている間に sapcontrol を介して発行された他のアクションは、すでに進行中の停止アクションと競合し、最終的に失敗してタイムアウトになります。<br><br>特に、次の一連のイベントでは、保護されたデータベースに対してローカルリカバリーが有効になっている場合でも、SAP HANA リソース階層のフェイルオーバーが発生する可能性があります。<br><br>1. ユーザーは、「sapcontrol Stop/StopWait」または「HDB stop」コマンドを発行して、プライマリーノード上で実行されている HANA データベースの正常なシャットダウンを開始します。<br><br>2. SAP HANA Recovery Kitの「quickCheck」スクリプトは、少なくとも1つのデータベースプロセスが実行されていないことを検出し、その結果、ローカルでデータベースを再起動しようとします。<br><br>3. SAP HANA Recovery Kitの「recover」スクリプトは、「sapcontrol StartWait」コマンドを発行して、保護された HDB インスタンスの再起動を試みます。<br><br>4. ステップ 1 で発行された「sapcontrol Stop/StopWait」コマンドはまだ HANA データベースプロセスをアクティブに停止しているため、SAP HANA Recovery Kitが発行する「sapcontrol Start」コマンドは失敗してタイムアウトします。<br><br>5. SAP HANA Recovery Kit はローカルでデータベースを再起動できないため、ローカルリカバリーは失敗し、SAP HANAリソース階層はスタンバイサーバーにフェイルオーバーします。<br><br>**回避策 / 解決策**: 本番前のクラスターテストの一環として、プライマリーデータベースの障害後のローカル |

| |
|---|
| 復旧をシミュレートするためにデータベースを手動で停止する場合、プライマリーデータベースのクラッシュをより正確にシミュレートするためにデータベースプロセスを強制終了（たとえば 'HDB kill-9' など）することを検討してみてください。詳細については SAP HANA リソース階層のテスト を参照してください。 |
| **SAP HANA** のライセンス名が正しく表示されない<br><br>ライセンスが以下のように表示されます。<br><br>Unknown (lklce database hana) Eval 13 Mar 2022 (19 days) |

# 5.4.5.3.20. Recovery Kit for EC2 – 既知の問題と制限

| 説明 |
| --- |
| **EC2**インスタンスに十分な権限が付与されているにも関わらず、**EC2**リソース（ルートテーブルシナリオ）作成時に権限の不足の警告が表示され、リソースの作成に失敗する。<br><br>**原因:**<br>以下の2つの原因が関係して本現象が発生します。<br>1. EC2リソース（ルートテーブルシナリオ）作成時、ロールチェックで権限が付与されていないと判断されることがあります。これは権限が付与されていないRouteTableに対してLifeKeeperがロールチェックをしていることが原因です。<br>2. v9.6.0で送信元/送信先チェックを無効化する処理が追加されました。本機能のデフォルトの挙動として、権限が不足している場合はEC2リソース（ルートテーブルシナリオ）の作成に失敗するようになりました。<br>**対処:**<br>*/etc/default/LifeKeeper* に `EC2_NO_SRC_DEST_CHECK=0` を設定することで、権限の不足による警告が表示されてもリソースの作成自体は成功するように変更できます。（これはv9.5.1以前と同じ挙動となります。） |
| インスタンスの**UTC**時刻を**AWS**の**UTC**時刻に合わせてください。<br><br>インスタンスの時刻が合っていなかった場合、EC2 リソースのリストアに失敗します。 |

# 5.4.5.3.21. Oracle Cloud Infrastructure (OCI) 上で LifeKeeper を使用する際の既知の問題/制限事項

OCI 上では Recovery Kit for Oracle Cloud Infrastructure を利用する必要があります。

Oracle 社が提供している プラットフォーム・イメージ を利用する場合は、LifeKeeper は以下の OS をサポートします。
OS のバージョンについては サポートマトリックス に従います。

- Oracle Linux (UEK, RHCK)
- CentOS 7

BYOI (Bring Your Own Image) の場合は、 サポートマトリックス に従います。ただし、以下の OS はサポート対象外です。

- SLES
- CentOS 8

以下の Recovery Kit はサポート対象外です。

- WebSphere MQ Recovery Kit
- SAP Recovery Kit
- SAP HANA Recovery Kit
- SAP MaxDB Recovery Kit
- Sybase Recovery Kit

また以下の Recovery Kit は OCI 上で利用できません。

- DB2 Recovery Kit
- Recovery Kit for EC2
- Recovery Kit for Route53
- VMDK as Shared Storage Recovery Kit
- Multipath Recovery Kits

# 5.4.5.4. コミュニケーションパスの稼働と停止

コミュニケーションパスの停止と稼働が繰り返される場合（LifeKeeper GUI でAlive、Dead、Aliveというように表示される場合）、ハートビートの設定がクラスタ内のすべてのサーバで同じ値に設定されていない可能性があります。

この状態は、いずれか一方のサーバにある LifeKeeper デフォルトファイル **/etc/default/LifeKeeper** で設定名に誤記がある場合にも発生する可能性があります。

## 推奨される対策

1. クラスタ内のすべてのサーバで LifeKeeper を停止します。

2. クラスタ内の各サーバで、 **/etc/default/LifeKeeper** にある **LCMHBEATTIME** 設定と **LCMNUMHBEATS** 設定の値とスペルを確認します。設定値、スペルミスの無いことを各ノードで確認します。

3. クラスタ内のすべてのサーバで LifeKeeper を再起動します。

# 5.4.5.5. 不完全なリソースの作成

インスタンスの一部のみが作成された状態で、リソース設定プロセスが中断された場合、階層を再設定する前に、手動でクリーンアップする必要があります。LifeKeeper GUI を使用し、一部が作成されたリソースを削除してください。手順については、すべてのサーバからの階層の削除 を参照してください。階層リストにこれらのリソースが含まれていない場合、ins_remove（LCDI-instances(1M) を参照）および dep_remove（LCDI-relationship(1M)）を使用し、部分的な階層をクリーンアップしなければならない可能性もあります。

# 5.4.5.6. 不完全なリソースの優先順位の変更

LifeKeeper の階層は、親子の関係によって関連付けられたすべてのリソースとして定義されています。複数の親を持つリソースの場合、GUI と階層のすべてのリソースを区別することは一概に簡単とも言えなくなります。階層の整合性を保持するには、サーバごとに階層内のすべてのリソースに対して優先順位を変更する必要があります。[OK] または [Apply] ボタンを押した後で選択される階層のすべてのルートリソースを表示することで、GUI はこの要件を強制します。この時点で、すべてのルートを受け付けるか、操作をキャンセルするかを選択できます。ルートのリストを受け付けた場合、新しい優先順位の値が階層内のすべてのリソースに割り当てられます。

その階層の [Resource Properties] ダイアログが表示されている間、他の変更を階層に加えていることを確認する必要があります。[Resource Properties] ダイアログの優先順位を編集する前に、LifeKeeper に加えられた変更が動的にダイアログで更新されます。ただし、変更を加えると、基本的な変更が LifeKeeper で加えられた場合でも、ダイアログの値は凍結されます。[Apply] または [OK] ボタンをクリックした後でのみ、変更が加えられたことが通知されるので、優先順位の変更操作は要求どおりに進みません。

複数の優先順位の変更を伴う優先順位の変更操作時に、復旧できないエラーの可能性を最低限に抑えるには、プログラムは、一度に 1 つのサーバに対して個別に行われる一連の変更として、複数の優先順位の変更操作を実行します。また、操作時に優先順位の競合を防ぐために、必要に応じて一時的な値がプロパティに割り当てられます。この一時的な値は、最大許容値 999 を超えるもので、優先順位の変更中に一時的に GUI に表示されることもあります。操作が完了すると、一時的な値はすべて、要求された値に置き換えられます。エラーが発生し、優先順位の値をロールバックできない場合、一時的な優先順位の値の一部がそのまま残る可能性もあります。この場合は、下記の推奨手順に従って階層を修復してください。

## 一貫した状態への階層のリストア

優先順位の変更操作の間にエラーが発生し、操作を完了できない場合、優先順位は不整合の状態のまま残る可能性があります。エラーは、システムやコミュニケーションパスの障害を含め、さまざまな理由で発生します。操作が開始された後や完了する前にエラーが発生し、プログラムが前の優先順位にロールバックできなかった場合、操作中にエラーがあったこと、および前の優先順位を restore できなかったことを示すメッセージが表示されます。この場合、以下の処置を実行し、階層を一貫性のある状態に restore する必要があります。

1. 可能であれば、問題の原因を特定します。システムまたはコミュニケーションパスの障害を確認します。優先順位管理プログラムの実行中に、その他の操作が行われていないことを確認します。

2. 可能であれば、問題の原因を修正してから先に進みます。たとえば、階層を修復する前に、障害が発生したシステムまたはコミュニケーションパスを restore する必要があります。

3. [Resource Properties] ダイアログから操作を再試行します。

4. [Resource Properties] ダイアログから変更できない場合は、コマンドライン `hry_setpri` を使用して階層を修復するとより簡単かもしれません。このスクリプトを使用すると、一度に 1 つのサーバに対して優先順位を変更できます。このスクリプトは、GUI からは実行できません。

5. 修復を実行したら、階層が存在するすべてのサーバに対して `eqv_list` コマンドを実行し、階層の

すべてのリソースに対して返された優先順位の値を調べ、LifeKeeper データベースがすべてのサーバで一貫していることを確認します。

6. 最終的に、階層を修復できない場合は、階層を削除して再作成する必要がある可能性もあります。

# 5.4.5.7. 階層の設定中に共有ストレージが見つからない

リソースの階層を設定中に、LifeKeeper が「No shared storage」(共有ストレージがありません) というメッセージをレポートする状況がいくつかあります。

**考えられる原因**：ストレージを共有するサーバー間でコミュニケーションパスが定義されていません。共有ストレージデバイスで階層が設定されている場合、LifeKeeper は、クラスタ内の別のサーバーを少なくとも 1 つ検証し、その共有ストレージにアクセスできることを確認します。

**推奨される対策:** LifeKeeper GUI または `lcdstatus (1M)` を使用し、コミュニケーションパスが設定されており、アクティブになっていることを確認します。

**考えられる原因:** ストレージを共有するサーバー間でコミュニケーションパスが機能していません。

**推奨される対策:** LifeKeeper GUI または `lcdstatus (1M)` を使用し、コミュニケーションパスが設定されており、アクティブになっていることを確認します。

**考えられる原因:** Linux が共有ストレージにアクセスできない。この原因としては、ドライバがロードされていないことや、ドライバがロードされたときにストレージの電源が入っていないこと、あるいはストレージデバイスが正しく設定されていないことなどが考えられます。

**推奨される対策:** `/proc/scsi/scsi` でデバイスが正しく定義されていることを確認します。

**考えられる原因:** LifeKeeper を起動する前にストレージが Linux で設定されていない。LifeKeeper の起動時に、すべての SCSI デバイスがスキャンされ、デバイスのマッピングが判別されます。LifeKeeper の起動後にデバイスが設定された (電源がオンにされた、接続された、またはドライバがロードされた) 場合、デバイスを設定して使用できるようにするには、LifeKeeper を停止してから再起動する必要があります。

**推奨される対策:** `$LKROOT/subsys/scsi/resources/hostadp/device_info` にデバイスがリストされていることを確認してください。`$LKROOT` は、`/opt/LifeKeeper`です。デバイスがこのファイルにリストされていない場合、LifeKeeper はそのデバイスを使用しません。

**考えられる原因:** ストレージがサポートされていない。サポートストレージ一覧には、LifeKeeper で動作がテストされ、サポートされている具体的な SCSI デバイスが列挙されています。ただし、このリストに含まれているのは、すでに知られているデバイスなので注意してください。LifeKeeper の要件を満たしているものの、SIOS Technology Corp. がテストしていないデバイスが存在する可能性もあります。

**推奨される対策:** `$LKROOT/subsys/scsi/resources/hostadp/device_info` にデバイスがリストされていることを確認してください。`$LKROOT` は、`/opt/LifeKeeper` です。デバイスがこのファイルにリストされているものの、デバイス名の後に来る ID が「NU-」で始まる場合、LifeKeeper はデバイスから一意の ID を取得できなかったことを示します。一意の ID がない場合、LifeKeeper はデバイスが共有されているかどうかを判別できません。

**考えられる原因:** ストレージでは、デバイスを LifeKeeper で使用できるようにする前に、特定の LifeKeeper ソフトウェアをインストールする必要があります。たとえば、Raw I/O サポートを有効にするための **steeleye-lkRAW** キット、データレプリケーションを有効にするための **steeleye-lkDR** ソフトウェアなどです。

**推奨される対策:** 必要な LifeKeeper パッケージが各サーバーにインストールされていることを確認します。ソフトウェアの要件については、 LifeKeeper for Linux リリースノート を参照してください。

補足のヒント：

test_lk(1M)ツールを使用すると、ストレージおよび通信の問題のデバッグに役立ちます。

# 5.4.5.8. LifeKeeper サーバ障害からの復旧

LifeKeeper クラスタ内のサーバに、オペレーティングシステムの再インストールを (したがって LifeKeeper の再インストールも) 必要とする障害が発生した場合、クラスタの各サーバからリソース階層を再拡張する必要があります。ただし、再インストールしたサーバとの共有イクイバレンシ関係がクラスタのサーバにある場合、LifeKeeper は、再インストールしたサーバへ既存のリソース階層を拡張することを許可しません。また、再インストールされたサーバには階層が実際には存在していないため、再インストールしたサーバから階層を拡張解除することもできません。

## 推奨される対策:

1. リソース階層が設定されている各サーバで、eqv_list コマンドを使用してすべての共有イクイバレンシのリストを取得します (詳細については、LCDI-relationship を参照してください).

   下記の例では、server1 および server2 に対する IP リソースの iptag のコマンドおよび結果の出力を示します。ここでは、server2 が再インストールされたサーバ、server1 が設定された階層です。

   ```
   eqv_list -f:

   server1:iptag:server2:iptag:SHARED:1:10
   ```

2. リソース階層が設定された各サーバで、eqv_remove を使用して、階層の各リソースのイクイバレンシ関係を手動で削除します (詳細については、LCDI-relationship を参照してください)。

   たとえば、上記の手順 1 の例を基に、server1 に対して以下のコマンドを実行します。

   ```
   eqv_remove -t iptag -S server2 -e SHARED
   ```

3. 2つ以上のサーバがあるクラスタでは、これらのリソース階層のイクイバレンシ関係が定義されているクラスタ内の各サーバに対して手順 1 と 2 を繰り返します。

4. 最後に、GUI を使用し、リソース階層が in-service になっているサーバから再インストールされたサーバに各リソース階層を拡張します。

# 5.4.5.9. 停止できないプロセスからの復旧

プロセスが停止不可の場合、LifeKeeper は共有ディスクパーティションをアンマウントできない可能性があります。そのため、リソースを別のシステムで In Service にすることができません。停止できないプロセスから復旧する唯一の方法は、システムを再起動することです。

# 5.4.5.10. 手動リカバリ時のパニックからの復旧

手動スイッチオーバ時に PANIC になると、リカバリが不完全に終わる可能性があります。PANIC またはその他の大きなシステム障害が手動スイッチオーバ時に発生した場合、バックアップシステムへの完全自動リカバリは保証できなくなります。In Service になる必要があるすべてのリソースが In Service であることをバックアップシステムで確認してください。In Service ではないリソースがあった場合は、LifeKeeper GUI を使用して、そのリソースを手動で In Service にします。手順については、リソースを In-Service する を参照してください。

# 5.4.5.11. Out-of-Service 階層の復旧

LifeKeeper サーバの障害からの復旧の一環として、障害が発生したサーバで設定されているものの、サーバの障害時にどのサーバでも In Service ではなかったリソース階層が、障害時に最優先で Alive になったサーバで復旧されます。これは、障害が発生したサーバ、復旧中のサーバ、階層内の他のサーバを含め、Out of Service の階層が最後にどこで In Service だったかには無関係です。

# 5.4.5.12. リソースタグ名の制限

## タグ名の長さ

LifeKeeper 内のすべてのタグは、256 文字以内にする必要があります。

## 有効な特殊文字

| |
|---|
| - _ . / |

タグの最初の文字に「.」および「/」を使用することはできません。

## 無効な文字

| |
|---|
| + ; : ! @ # $ * = 「スペース」 |

# 5.4.5.13. シリアル (TTY) コンソールの警告

シリアルコンソールデータパスの一部が信頼できない場合、または Out of Service になった場合、シリアル (RS-232 TTY) コンソールを使用するユーザは LifeKeeper サービスで深刻な問題に直面する可能性があります。操作中に、LifeKeeper はコンソールメッセージを生成します。設定に (標準的な VGA コンソールではなく) シリアルコンソールがある場合、これらのコンソールメッセージが確実に配信されるようにするために、LifeKeeper からエンドユーザターミナルへのデータパス全体が機能している必要があります。

ターミナルの電源オフ、モデムの未接続、ケーブルのゆるみなど、データパスがつながっていない場合、Linux STREAMS ファシリティは、コンソールメッセージをキューに入れます。STREAMS キューがいっぱいになった場合、Unix カーネルは、STREAMS バッファキューにメッセージを入れる余地ができるまで LifeKeeper を保留にします。これにより、LifeKeeper がハングすることもあります。

**注記:** LifeKeeper 環境のシリアルコンソールは可能なかぎり避け、VGA コンソールを使用することを推奨します。シリアルコンソールを使用する必要がある場合、シリアルコンソールがオンになっていること、ケーブルとオプションのモデムが正しく接続されていること、メッセージが表示されていることを必ず確認してください。

# 5.4.5.14. システムが init 状態 S に遷移しているという警告

LifeKeeper が動作している場合、システムを直接、init 状態 S に切り替えないでください。Linux の init システムの操作が原因で、こうした遷移が全 LifeKeeper プロセスの即時停止につながり、突発的な障害を発生させる可能性があります。この場合は、代わりに LifeKeeper を (`lkstop` で) 手動停止するか、システムを最初に init 状態 1 にしてから init 状態 S にしてください。

# 5.4.5.15. 共有ストレージでスレッドがハングしているというメッセージ

デバイス確認スレッドがそれほど迅速に処理を完了していない場合、スレッドがハングしているというメッセージが LifeKeeper ログに記録されることがあります。これにより、リソースがあるサーバから別のサーバに移動し、さらに悪いケースでは、サーバが停止する可能性があります。

## 説明

(/etc/default/LifeKeeperの) FAILFASTTIMER は、各デバイスが正常に動作していること、および特定のシステムによって所有されているすべてのリソースがそのシステムからアクセス可能で、そのシステムに所有されていることを確認するための秒数を定義します。FAILFASTTIMER は、この所有権を確定し、データの信頼性を最大限に確保するために、可能なかぎり小さくする必要があります。ただし、デバイスがビジー状態で、負荷がピークの場合、指定した時間内で応答できない可能性もあります。デバイスの操作が FAILFASTTIMER よりも長くかかっている場合、LifeKeeper はデバイスがハングしている可能性を検討します。FAILFASTTIMER の時間を 3 回繰り返してもデバイスが応答しない場合、LifeKeeper は、デバイスに障害が発生したものとみなして、リカバリを実行します。リカバリプロセスは、SCSIERROR の設定で定義します。SCSIERROR の設定によっては、ローカルリカバリを実行し、失敗した場合はスイッチオーバを実行するために sendevent が発行されることもあります。この操作がない場合、システムが停止するおそれもあります。

## 推奨される対策:

ハングメッセージがまれにエラーログに出力され、もうハングしていないというメッセージがそれに続く場合、さらに括弧の数が常に 1 つの場合、それほど警戒する理由はありません。ただし、このメッセージが頻繁にログに記録され、数が 2 または 3 の場合、以下の 2 つの処置が必要になる可能性があります。

- ◦ ストレージの負荷を減らすことを試みる。ストレージの処理に FAILFASTTIMER (デフォルトでは、5 秒または 15 秒を 3 回) の 3 倍の時間がかかっている場合、ストレージに対する負荷を考慮し、I/O の長い遅延を避けるために負荷を分散する必要があります。これにより、LifeKeeper は、デバイスを頻繁に確認できるようになり、さらにそのデバイスを使用しているアプリケーションのパフォーマンスも向上します。

- ◦ 負荷を減らすことができない場合、FAILFASTTIMER をデフォルトの 5 秒から増やすことができます。この値は、できる限り低く抑える必要があります。そのため、メッセージがまったく表示されなくなるか、まれにしか表示されなくなるまで、少しずつ値を増やしてください。

注記: FAILFASTTIMER の値が変更された場合、新しい値を有効にするために、LifeKeeper を終了し、再起動する必要があります。

# 5.5. SIOS DataKeeper for Linux

## はじめに

SIOS DataKeeper for Linux は、LifeKeeper 環境に統合データミラーリング機能を提供します。この機能により、LifeKeeper リソースが共有 / 非共有ストレージ環境で動作可能になります。

SIOS DataKeeper for Linux によるミラーリング

SIOS DataKeeper の仕組み

# 5.5.1. SIOS DataKeeper for Linux によるミラーリング

SIOS DataKeeper for Linux は、共有ストレージを使用せずに可用性の高いクラスタ（SIOS LifeKeeper を使用）を構築したいお客様や、ビジネスに不可欠なデータをサーバ間でリアルタイムに複製したいお客様に別の方法を提供します。

SIOS DataKeeper は、同期または非同期のボリュームレベルのミラーリングを使用して、プライマリサーバ（ミラーソース）から 1 台以上のバックアップサーバ（ミラーターゲット）にデータを複製します。

## DataKeeper の特長

SIOS DataKeeper には、以下の特長があります。

- ◦ TCP/IP ベースのローカルエリアネットワーク（LAN）またはワイドエリアネットワーク（WAN）経由で、リモートの場所に高い信頼性、効率、整合性でデータをミラーリングできます。
- ◦ 同期と非同期のミラーリングをサポートします。
- ◦ 複製はファイルシステムの下のブロックレベルで実行されるので、関与するアプリケーションに対して透過的です。
- ◦ LifeKeeper と共に使用した場合、複数ターゲットへのカスケーディングフェイルオーバも含めて、複数ターゲットへの同時ミラーリングをサポートします。
- ◦ 内蔵のネットワーク圧縮ににより、ワイドエリアネットワークでの最大スループットが向上します。
- ◦ 主要なファイルシステムをすべてサポートします（ファイルシステムのジャーナリングサポートの詳細については、 SPS for Linux リリースノート の製品説明を参照してください）。
- ◦ ミラーリングしたデータにフェイルオーバの保護を提供します。
- ◦ LifeKeeper のグラフィカルユーザインターフェースに統合されています。
- ◦ 他の LifeKeeper Application Recovery Kit をフルにサポートします。
- ◦ システムリカバリ時に、プライマリサーバとバックアップサーバとの間でデータを自動的に再同期します。
- ◦ 障害発生時には、仮想のシステムコンポーネントの健全性を監視し、ローカルリカバリを実行します。
- ◦ I/O フェンス用の STONITH デバイスをサポートします。詳細については、STONITH のトピックを参照してください。

# 同期ミラーリングと非同期ミラーリングの違い

同期ミラーリングと非同期ミラーリングの違いを理解すると、アプリケーション環境に適切なミラーリング方法を選択することができます。

## 同期ミラーリング

SIOS DataKeeper は、プライマリサーバとバックアップサーバに同時にデータを書き込む同期ミラーリング方法を使用して、リアルタイムミラーを実現します。書き込み動作のたびに、DataKeeper は書き込みをターゲットデバイスに転送し、リモート確認を受信してから I/O 完了を通知します。同期ミラーリングの長所は、データの保護レベルが高いことです。これは、常にデータのすべてのコピーを確実に同一にしているからです。ただし、リモート確認を待つために、パフォーマンスが低下することがあり、これは特に WAN 環境で発生します。

## 非同期ミラーリング

非同期ミラーリングでは、それぞれの書き込みがソースデバイスに対して行われ、次に、コピーがターゲットデバイスに送信されるキューに入れられます。これはつまり、任意の時点で、ソースからターゲットデバイスへの送信を待っている多数の書き込みトランザクションが存在する可能性があります。非同期ミラーリングの長所は、書き込みがプライマリディスクに到達した時点で確認されるため、パフォーマンスが高いことです。ただし、プライマリディスクに障害が発生した場合、非同期書き込みキュー内にある書き込みはターゲットに送信されないため、信頼性が低くなります。この問題を緩和するために、SIOS DataKeeper はプライマリディスクに対する個々の書き込みについてインテントログファイルにエントリを作成します。また、多量の書き込みが継続的に行われた場合、キューに書き込まれたデータの送出を優先するため一時的にI/Oパフォーマンスが低下する場合があります。

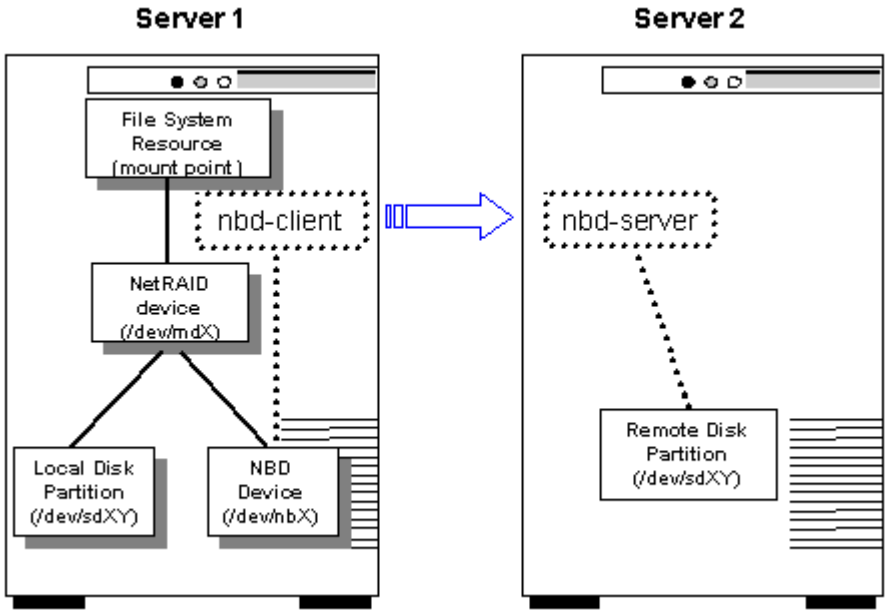
インテントログとは、プライマリとターゲットのミラー間で同期していないデータブロックを示すビットマップです。サーバの障害発生時にインテントログを使用すると、データ全体の再同期を回避できます。

> 全同期は、ディスクパーティション全体を同期させます

> **補足:** 非同期ミラーの場合、DataKeeperでは、最大4096個の未処理のターゲット書き込みをキューに入れることができます。/etc/default/LifeKeeper の `LKDR_ASYNC_LIMIT` で、さらに多くの書き込みをキューに入れられるように設定してください。
>
> ターゲットへの未処理の書き込みの数が上限に達すると、その数が設定値を下回るまで、ミラーは同期モードに戻ります。
>
> ブロックサイズが4Kであると想定すると、値がデフォルト（書き込みの非同期制限が4096）のままになっている場合、最大16MBのデータが転送されます。

# 5.5.2. SIOS DataKeeper の仕組み

SIOS DataKeeper は、NetRAID デバイスを作成して保護します。NetRAID デバイスは RAID1 のデバイスであり、下図に示すようにローカルのディスクまたはパーティション、およびネットワークブロックデバイス（NBD）で構成されます。




LifeKeeper がサポートするファイルシステムは、その他すべてのストレージデバイスと同様に、NetRAID デバイスにマウントできます。この場合、ファイルシステムは複製されたファイルシステムと呼ばれます。LifeKeeper は、NetRAID デバイスと複製されたファイルシステムの両方を保護します。

ファイルシステムは、DataKeeper [リソース階層を作成することにより作成されます。NetRAID デバイスを別のサーバに拡張すると NBD デバイスが作成され、2 台のサーバ間にネットワーク接続が確立されます。NBD 接続が確立されるとただちに、SIOS DataKeeper がデータの複製を開始します。

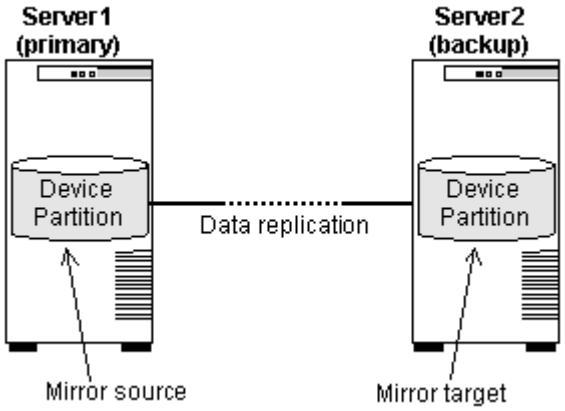
nbd-client プロセスがプライマリサーバで実行され、バックアップサーバで動作している nbd-server プロセスと接続します。

## 同期（および再同期）

DataKeeper リソース階層は作成されてから拡張されるまでの間、デグレードモードです。つまり、データはローカルのディスクまたはパーティションにのみ書き込まれます。階層をバックアップ（ターゲット）システムに拡張すると、SIOS DataKeeper が 2 つのシステム間でデータを同期し、以降の書き込みはすべてターゲットに複製されます。どの時点でもデータが「非同期」になった場合（システムまたはネットワークの障害が発生した場合）、SIOS DataKeeper はソースとターゲットのシステムでデータを自動的に再同期します。インテントログ（ビットマップファイル）を使用するようにミラーが設定されている場合、SIOS DataKeeper はインテントログを使用して非同期のデータを特定するので、全体の再同期は不要です。インテントログ（ビットマップファイル）を使用するようにミラーが設定されていない場合は、データ複製の中断後に全体の再同期が実行されます。

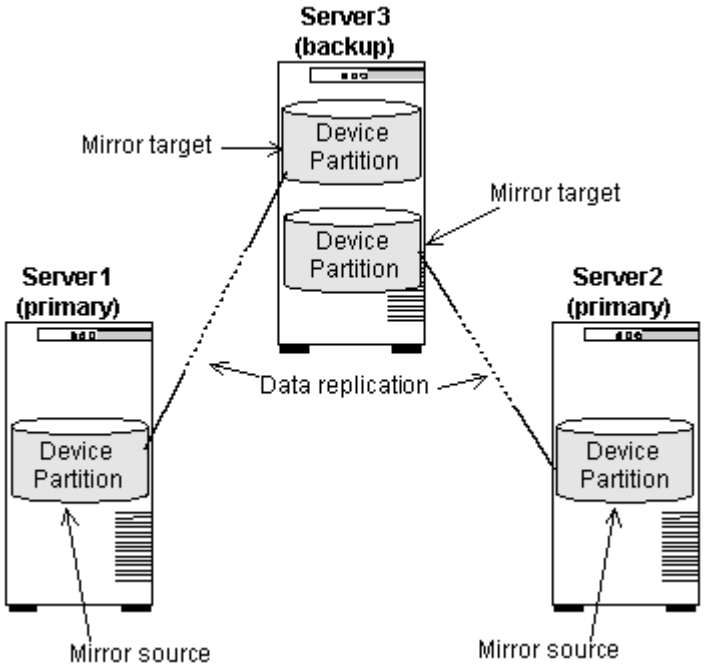
# 標準ミラーの構成

最も一般的なミラーの構成では、下図に示すように 2 台のサーバがあり、各サーバのローカルのディスクまたはパーティションとの間にミラーが確立されます。サーバ 1 は、ミラーソースを持つプライマリサーバです。サーバ 2 は、ミラーターゲットを持つバックアップサーバです。



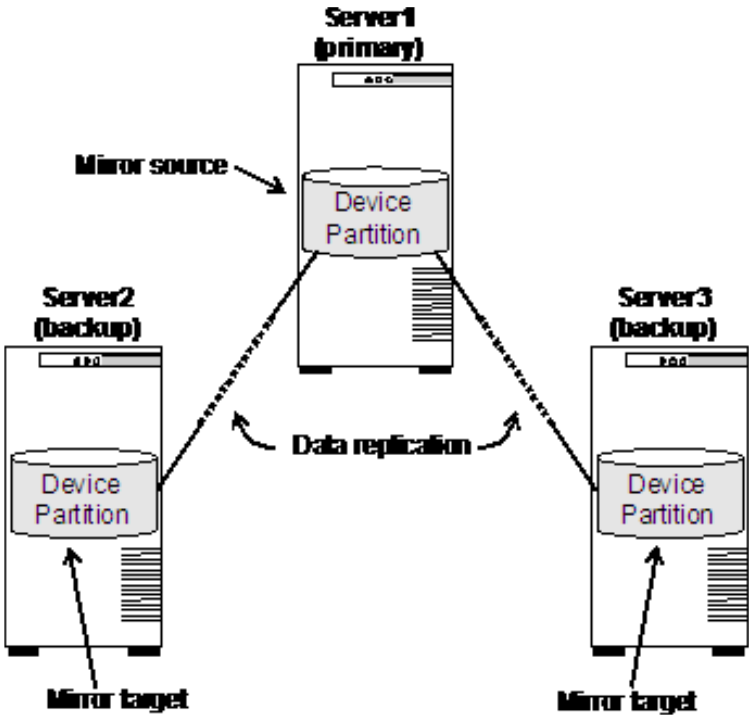

# N+1 Configuration

前述した標準ミラーの構成の変形として一般的に使用される構成では、クラスタ内にある 2 台以上のサーバが共通のバックアップサーバにデータを複製します。この場合は、下図に示すように、各ミラーソースがバックアップサーバの個別のディスクまたはパーティションに複製する必要があります。



# 複数ターゲットの構成

適切な Linux のディストリビューションとバージョン 2.6.7 以降のカーネルと共に使用した場合、下図に示すように、SIOS DataKeeper は、プライマリサーバの 1 つのディスクまたはパーティションから複数のバックアップシステムにデータを複製することもできます。



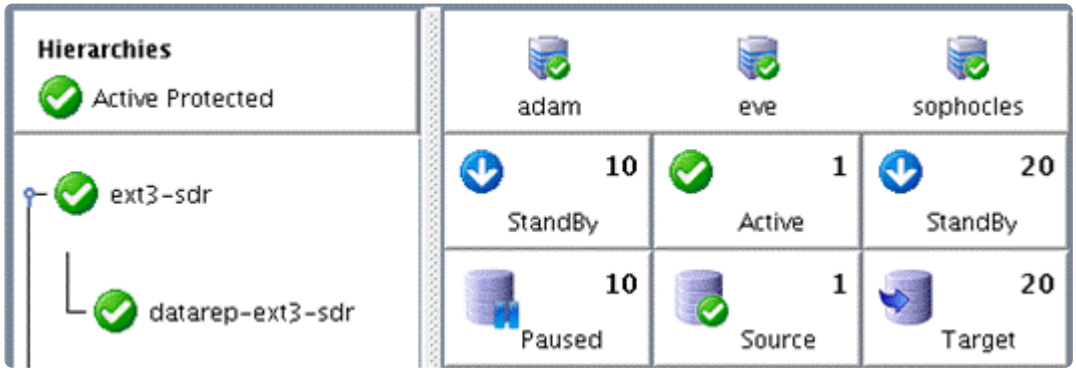

特定のソースのディスクまたはパーティションを最大 7 つのミラーターゲットに複製でき、各ミラーターゲットは別のシステムに存在する必要があります。つまり、ソースのディスクまたはパーティションを、同一ターゲットシステム上にある複数のディスクまたはパーティションにミラーリングすることはできません。

このタイプの構成では、LifeKeeper のカスケーディングフェイルオーバ機能を使用でき、保護するアプリケーションとそのデータに対して複数のバックアップシステムを提供できます。

ミラーリングの開始時にすべてのターゲットに完全再同期されるのを回避するには、クラスター内の残りのターゲットを再接続する前に、まず前のソースのビットマップをマージする必要があります。 9.3.2 以前のバージョンでは、ミラーリングの開始時に前のソースが利用できなかった場合は、各ターゲットに対して完全再同期が自動的に行われます。 9.3.2 からは、ミラーがシステム上で起動されると、ターゲットに接続する前に、前のソースがクラスターに参加するのを待ちます。前のソースがクラスターに参加すると、そのビットマップがマージされ、すべてのターゲットが部分的な再同期で参加できるようになります。ミラーが停止し、ターゲットが同期中の場合、ミラーを起動してターゲットに複製するために前のソースは必要ありません。前のソースがクラスターへの再参加に利用できない場合は、「mirror_action fullresync」コマンドを使用して、ターゲットを完全再同期で手動で再同期することができます。 /etc/default/LifeKeeper の変数 LKDR_WAIT_FOR_PREVIOUS_SOURCE_TIMEOUT によって、再同期の動作が決まります。（詳細は「DataKeeper パラメーターリスト 」を参照）。

# SIOS DataKeeper リソース階層

以下の例に、LifeKeeper の GUI に表示される典型的な DataKeeper リソース階層を示します。




リソース *datarep-ext3-sdr* は NetRAID リソースであり、親リソース *ext3-sdr* はファイルシステムリソースです。本書の以降の部分では、「DataKeeper リソース」は両方のリソースを合わせたものを指すことに注意してください。ファイルシステムリソースは NetRAID リソースに依存するので、NetRAID リソースに対する動作はその上にあるファイルシステムにも影響します。

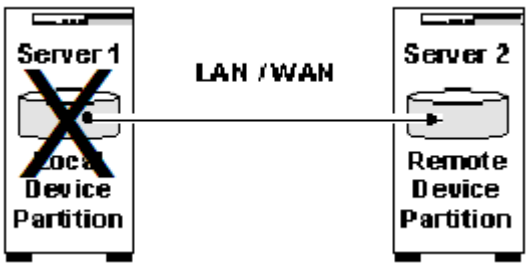
# フェイルオーバのシナリオ

## フェイルオーバのシナリオ – 2 ノード

以下の 4 つの例で、SIOS DataKeeper を使用するフェイルオーバで何が起きるかを説明します。これらの例では、LifeKeeper for Linux クラスタは、サーバ 1（プライマリサーバ）とサーバ 2（バックアップサーバ）の 2 台のサーバで構成されます。

### シナリオ 1

サーバ 1 からサーバ 2 へのミラーが正常に完了し、その後サーバ1が動作不能に陥る。




**結果:** フェイルオーバが発生します。サーバ 2 がプライマリサーバの役割を担当し、サーバ 1 が再び動作可能になるまでデグレードモード（バックアップなし）で動作します。サーバ 1 が再び動作可能になると、SIOS DataKeeper がサーバ 2 からサーバ 1 への再同期を開始します。2.6.18 以前のカーネルでは、全体の再同期が実行されます。2.6.19 以降のカーネル、または Red Hat Enterprise Linux 5.4 の 2.6.18-164 以降のカーネル（または Red Hat 5.4 以降のサポートする派生カーネル）では、部分的な再同

期が実行されます。つまり、ソースとターゲットにあるビットマップファイルに記録された変更部分についてのみ同期が必要です。

**注記:** SIOS DataKeeper は、現在ミラーソースとして動作しているサーバに以下のフラグをセットします。

```
$LKROOT/subsys/scsi/resources/netraid/$TAG_last_owner
```

サーバ 1 がサーバ 2 にフェイルオーバすると、このフラグがサーバ 2 にセットされます。このため、サーバ 1 が動作を再開すると、SIOS DataKeeper はこの最終オーナフラグをサーバ 1 から削除します。その後、サーバ 2 からサーバ 1 にデータの再同期を開始します。

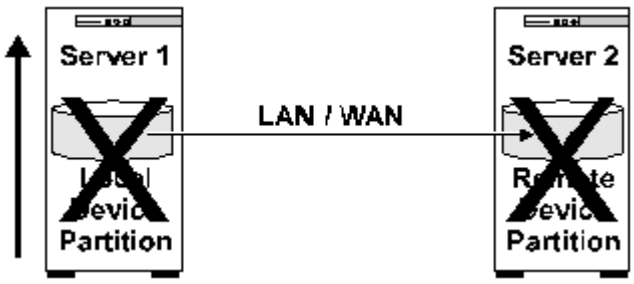
### シナリオ 2

シナリオ 1 で、サーバ 2（プライマリサーバである状態）が、サーバ 1（この時点ではバックアップサーバ）との再同期中に動作不能になる。

**結果:** 再同期プロセスが正常に完了しなかったので、データが破損している可能性があります。この結果、LifeKeeper は DataKeeper リソースをサーバ 1 にフェイルオーバしません。サーバ 2 が動作可能になった場合にのみ、LifeKeeper はサーバ 2 で DataKeeper リソースをサービス中（ISP）にします。

### シナリオ 3

サーバ 1（プライマリ）とサーバ 2（ターゲット）の両方が動作不能になる。サーバ 1（プライマリ）が最初に動作可能になる。




**結果:** サーバ 1 は、DataKeeper リソースを in service にしません。この理由は、停止してからオンラインに戻ったソースサーバは、ターゲットと通信できないからです。ソースサーバは以下のタグをセットします。

```
$LKROOT/subsys/scsi/resources/netraid/$TAG_data_corrupt
```

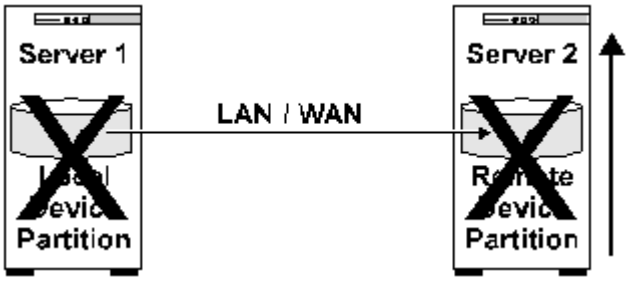
これは、正しくない方向へのデータ同期を防止する安全策です。この場合、サーバ 1 でミラーを強制的にオンラインにする必要があります。つまり、サーバ 1 の *data_corrupt* フラグを削除し、リソースを in service にします。ミラーを強制的にオンラインにする を参照してください。

**注記:** *$TAG_data_corrupt* フラグを削除する前に、サーバ 1 が最終のプライマリサーバであることを確認する必要があります。サーバ 1 が最終のプライマリサーバでない場合、データが破損する可能性があります

す。これは、 *last_owner* フラグの有無で確認できます。

### シナリオ 4

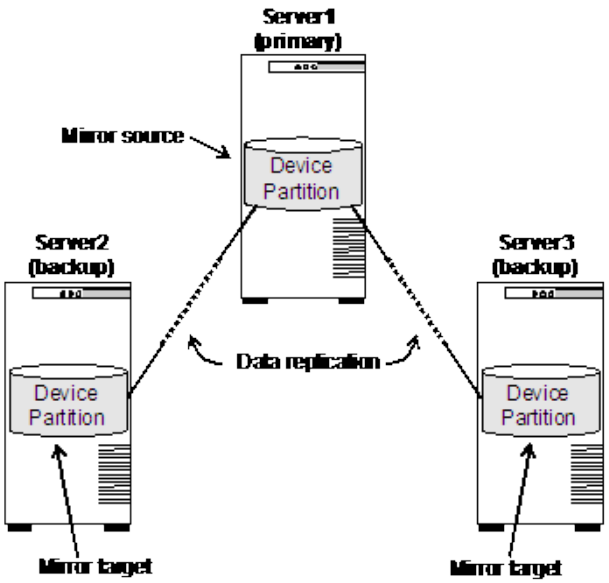
サーバ 1（プライマリ）とサーバ 2（ターゲット）の両方が動作不能になる。サーバ 2（ターゲット）が最初に動作可能になる。



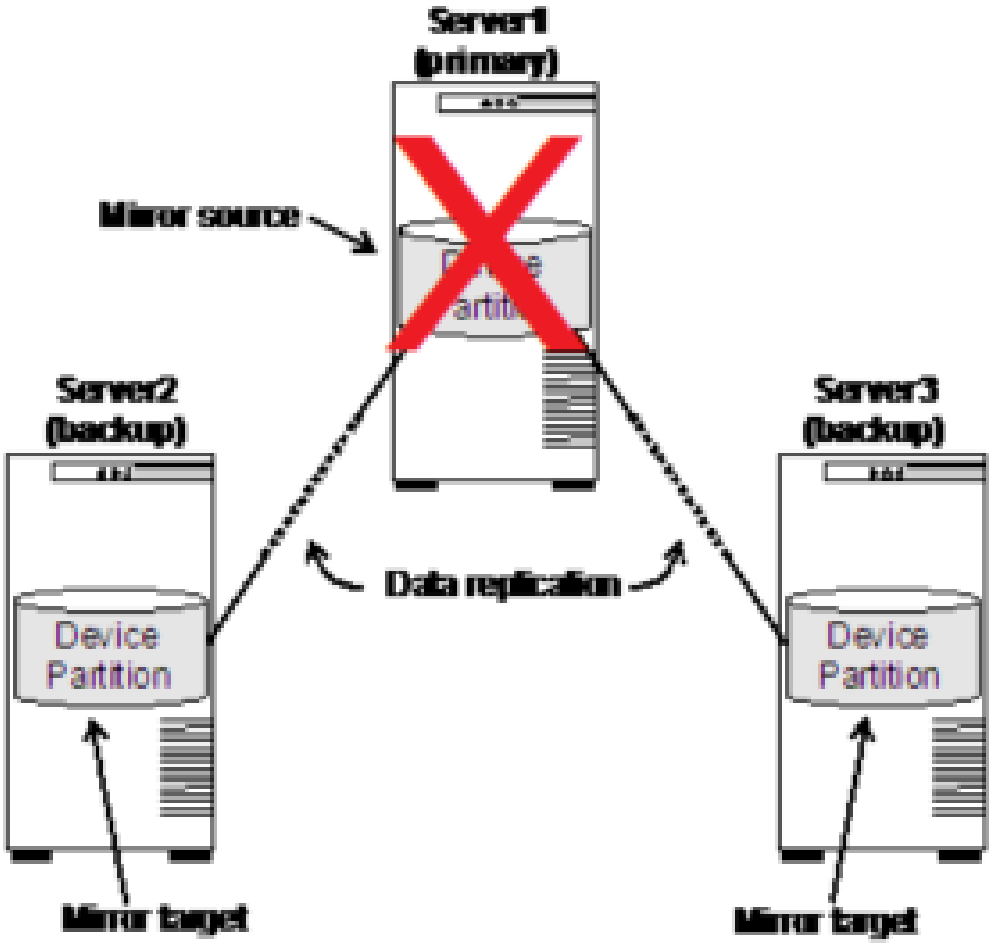

**結果:** LifeKeeper は、サーバ 2 の DataKeeper リソースを ISP にしません。サーバ 1 が動作可能になると、LifeKeeper はサーバ 1 の DataKeeper リソースを ISP にします。

## フェイルオーバのシナリオ – 3 ノード

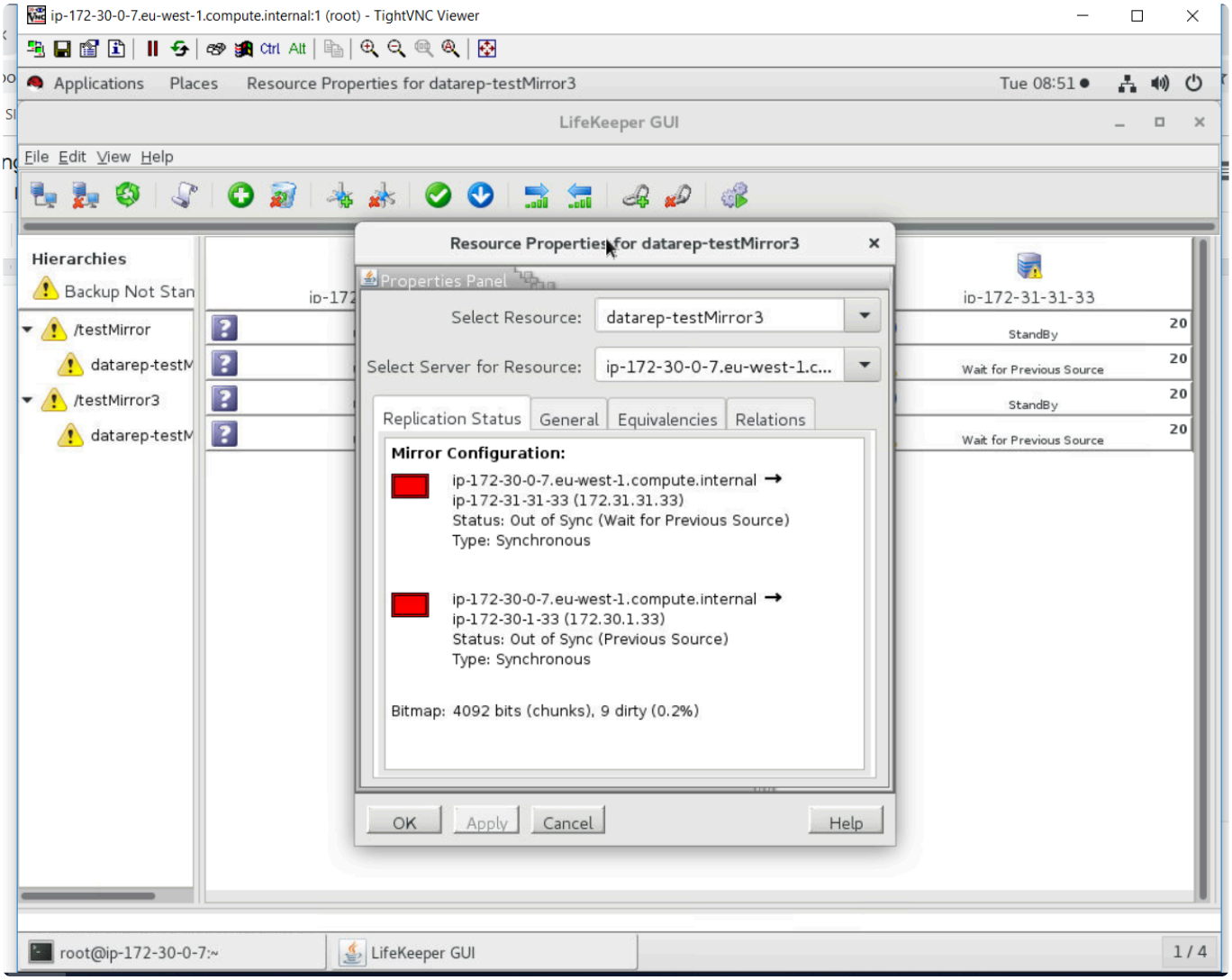
次の例は、SIOS DataKeeper を使用した場合にフェイルオーバ中に何が起こるかを示しています。この例では、LifeKeeper for Linuxクラスターは、サーバ1（プライマリサーバ）、サーバ2（バックアップサーバ）、およびサーバ3（バックアップサーバ）の3つのサーバで構成されています。

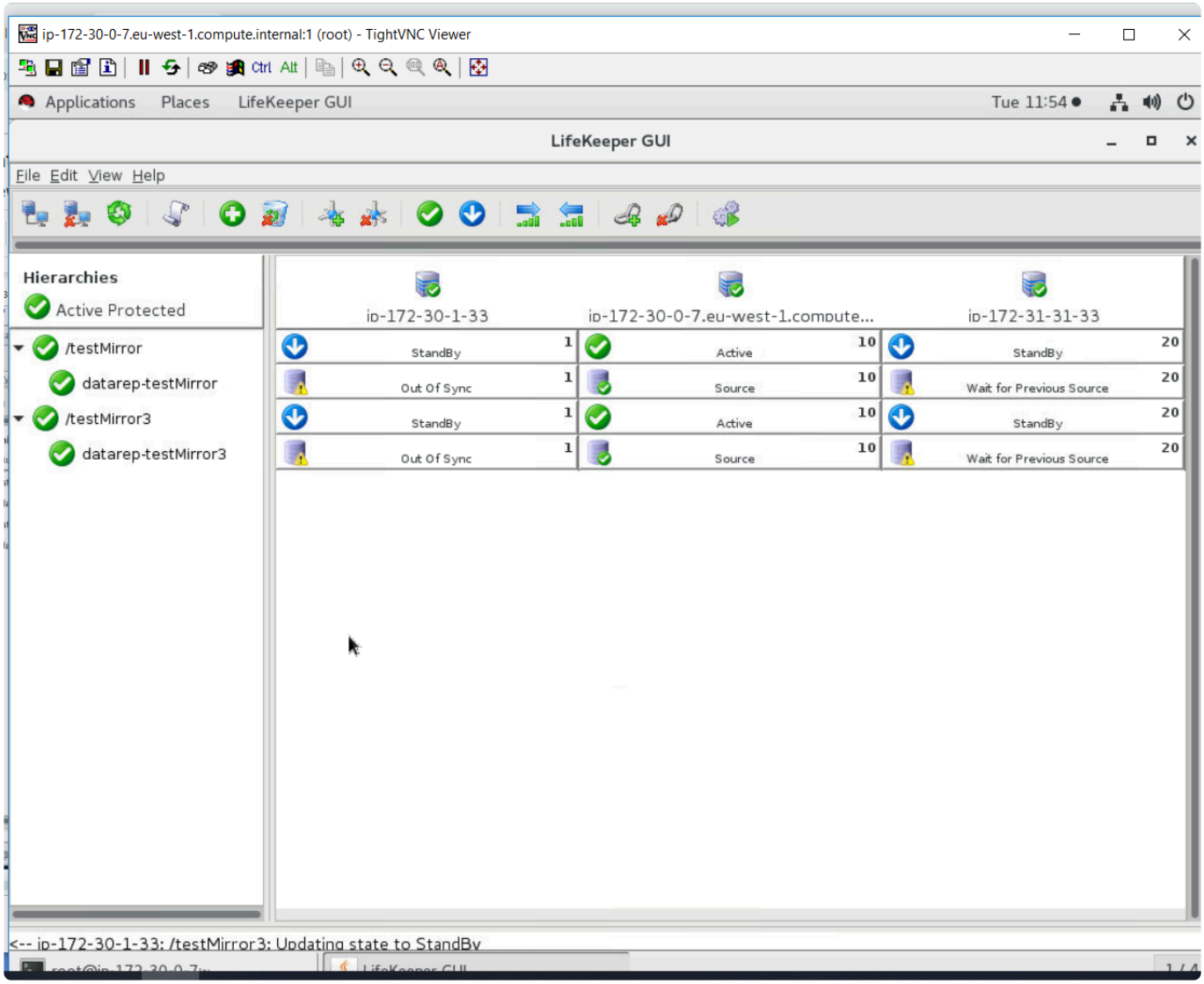



サーバ1（優先順位1）がサーバ2（優先順位10）およびサーバ3（優先順位20）への複製を正常に完了した後、サーバ1 は動作不能になります。

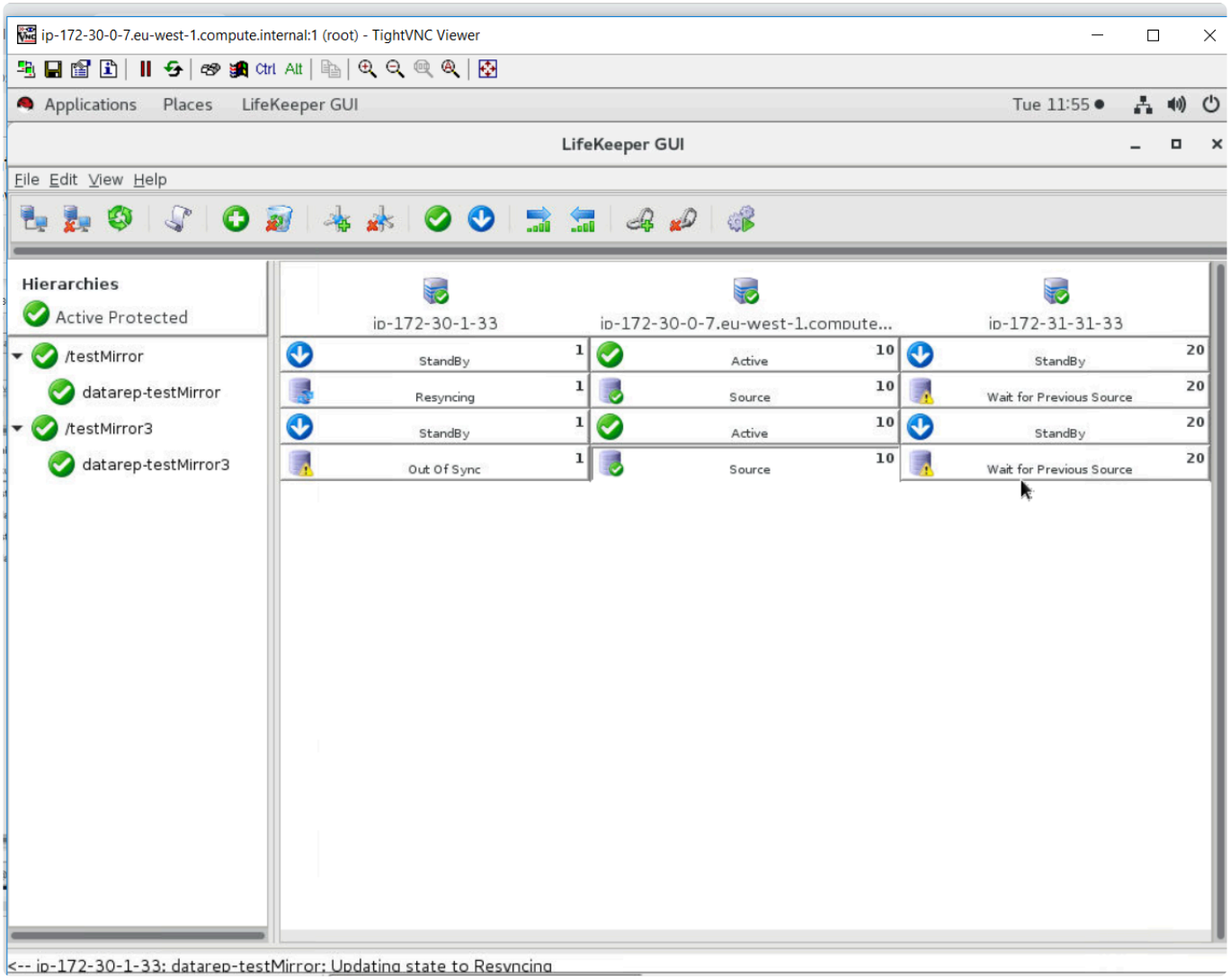



**結果：** フェイルオーバは、2番目に優先順位の高いサーバ2 で行われます。 サーバ2 は、プライマリサーバの役割を引き継ぎます。 v9.3.2 以前のバージョンでは、サーバ3 は完全再同期化とともにミラーに追加されます。 v9.3.2 では、サーバ2はサーバ1（前のサーバ）がクラスターに戻るのを待ってからサーバ3 への複製を再開します。これにより、サーバ1 のビットマップとサーバ2 のビットマップをマージして、サーバ1 とサーバ3 の両方に部分的に再同期することができます。 サーバ1 がクラスターに再接続するのを待っている間、LifeKeeper GUI はサーバ3 のステータスを「Out of Sync (Wait for Previous Source)」として表示します。 サーバが接続されていない間、サーバ1 のステータスは「Unknown」になり、最初に接続されたときに GUI のステータスに「Out of Sync」と表示されます。 ミラーのプロパティページは、それを「Out of Sync (Previous Source)」として識別します。


ip-172-30-0-7.eu-west-1.compute.internal:1 (root) - TightVNC Viewer
Applications Places Resource Properties for datarep-testMirror3 Tue 08:51
LifeKeeper GUI
File Edit View Help
Hierarchies
Backup Not Stan
/testMirror
datarep-testM
/testMirror3
datarep-testM
ip-172-31-31-33
StandBy 20
Wait for Previous Source 20
StandBy 20
Wait for Previous Source 20
Resource Properties for datarep-testMirror3
Properties Panel
Select Resource: datarep-testMirror3
Select Server for Resource: ip-172-30-0-7.eu-west-1.c...
Replication Status General Equivalencies Relations
Mirror Configuration:
ip-172-30-0-7.eu-west-1.compute.internal →
ip-172-31-31-33 (172.31.31.33)
Status: Out of Sync (Wait for Previous Source)
Type: Synchronous
ip-172-30-0-7.eu-west-1.compute.internal →
ip-172-30-1-33 (172.30.1.33)
Status: Out of Sync (Previous Source)
Type: Synchronous
Bitmap: 4092 bits (chunks), 9 dirty (0.2%)
OK Apply Cancel Help
root@ip-172-30-0-7:~
LifeKeeper GUI
1 / 4




サーバ1 が再接続すると、そのビットマップがマージされて再同期が開始され、その時点でステータスは「Resyncing」と表示されます。




再同期が完了するとステータスは「Target」に更新され、サーバ3 はそのステータスを「Resyncing」に変更して再同期を開始します。

注記： SIOS DataKeeper はミラーソースを追跡するために以下のフラグを設定します。

*$LKROOT/subsys/scsi/resources/netraid/$TAG_last_owner*

*$LKROOT/subsys/scsi/resources/netraid/$TAG_source*

*$TAG_last_owner* フラグは、現在ミラーソースとして機能しているシステム上にあり、 *$TAG_source* フラグには、ローカルノードがミラーの一部であった最後の時点でソースだったシステムの名前が含まれています。

サーバ1 がサーバ2 にフェイルオーバすると、サーバ2 に *$TAG_last_owner* フラグが設定されます。サーバ2 の *$TAG_source* フラグは、サーバ1 を前のソースとして識別します（これには、サーバ1 とサーバ3 への部分的な再同期を実行するために必要なビットマップが含まれています）。サーバ1 が復旧すると、SIOS DataKeeper はサーバ1から *$TAG_last_owner* フラグを削除します。次にサーバ2 はサーバ1 からのビットマップをマージし、サーバ2 からサーバ1 へのデータの再同期を開始します。サーバ1 が復旧すると、SIOS DataKeeper はサーバ1 から *$TAG_last_owner* フラグを削除します。次にサーバ2 はサーバ1 のビットマップをマージし、サーバ2 からサーバ1 へのデータの再同期を開始します。サーバ1 への再同期が完了すると、サーバ1 の *$TAG_source* フラグがサーバ2 の名前で更新されます。サーバ1 が同期された

後、サーバ2 はサーバ3 へ同じ再同期を実行します。サーバ3 の *$TAG_source* フラグは、サーバ2 の名前で更新されます。

# 5.5.3. インストールと設定

## SIOS DataKeeper for Linux のインストールと設定

ハードウェア / ソフトウェア要件

## DataKeeper リソースを設定する前に

以下のトピックには、DataKeeper リソースの作成と管理を行う前に考慮が必要な情報があります。また、3 種類の DataKeeper リソースについても説明しています。LifeKeeper Core のリソース階層を設定する手順については、LifeKeeper の設定 セクションを参照してください。

---

ハードウェアとソフトウェアの要件

全般的な設定

ネットワーク設定

データレプリケーションパスの変更

ネットワーク帯域幅の要件の特定

Linux システム（物理または仮想）での変化率の測定

WAN の設定

SIOS DataKeeper for Linux のリソースタイプ

DataKeeper 構成での I/O フェンシング

リソースの設定作業

DataKeeper リソース階層の作成

リソース階層の拡張

リソース階層の拡張解除

リソース階層の削除

DataKeeper リソースを Out of Service にする

DataKeeper リソースを In Service にする

リソース階層のテスト

# 5.5.3.1. ハードウェアとソフトウェアの要件

SIOS DataKeeper を インストールするには 、LifeKeeper の構成が次の要件を満たしている必要があります。

## ハードウェアの要件

- サーバ - LifeKeeper for Linux をサポートする 2 台以上のサーバ。
- IP ネットワークインターフェースカード - 各サーバにネットワークインターフェースカードが 1 つ以上必要です。ただし、LifeKeeper クラスタには 2 つのコミュニケーションパスが必要です。独立した 2 つのサブネットを使用する 2 つの分離した LAN ベースのコミュニケーションパスが推奨され、これらの 1 つ以上をプライベートネットワークとして構成する必要があります。ただし、TCP と TTY を組み合わせて使用することもできます。

> 注記: ソフトウェアミラーリングの特性により、サーバ間のネットワークトラフィックが多くなる可能性があります。このため、SIOS DataKeeper のデバイス用に個別のプライベートネットワークを実装することが推奨されます。この実装には、各サーバに追加のネットワークインターフェースカードが必要になることがあります。

- ディスクまたはパーティション - ソースとターゲットのディスクまたはパーティションとして動作する、プライマリサーバとバックアップサーバのディスクまたはパーティション。ターゲットのディスクまたはパーティションは、ソースのディスクまたはパーティション以上のサイズである必要があります。

> 注記: SIOS Data Replication 7.1.1 のリリースから、パーティションが作成されていないディスク全体（ */dev/sdd* ）の複製が可能になりました。旧バージョンの SIOS Data Replication では、ディスクを複製するには、パーティションを作成する必要がありました（ */dev/sdd1* のような 1 つの大きいパーティションの場合でも）。SIOS Data Replication 7.1.1 からこの制限が取り除かれました。

> 注記: SPS for Linux v9.5.0 のリリースに伴い、すべてのディスクは一意に識別できる必要があります。DataKeeper は、デバイス名（ */dev/sdd* ）を使用してデバイスを識別することを許可していましたが、状況によっては、デバイス名が変更され、データが破損する可能性があります。GPT パーティションテーブルを使用すると、一意の識別子を提供できます。

## ソフトウェアの要件

- オペレーティングシステム – SIOS DataKeeper は、Linux カーネル 2.6 をベースにする主要な Linux のディストリビューションと共に使用できます。サポートするディストリビューションのリストについては、SPS for Linuxリリースノート を参照してください。非同期ミラーリングとインテントログは、2.6.16 以降の Linux カーネルを使用するディストリビューションでのみサポートされます。複数のターゲットのサポート（複数のミラーターゲットのサポート）には、2.6.7 以降の Linux

カーネルが必要です。

- **LifeKeeper Installation** スクリプト - 多くの場合、以下のパッケージをインストールする必要があります (特定の SIOS DataKeeper の要件については、SPS for Linux リリースノート の「製品要件」セクションを参照してください)。

  **HADR-generic-2.6**

  SIOS DataKeeper を インストールする前 に、LifeKeeper クラスタの各サーバにこのパッケージをインストールする必要があります。HADR パッケージはSPS のインストールイメージファイル内にあり、Installation の **setup** スクリプトにより自動的に適切なパッケージがインストールされます。

- **LifeKeeper** ソフトウェア - 各サーバに同じバージョンの LifeKeeper Core をインストールする必要があります。また、使用を計画している同じバージョンの Recovery Kit も各サーバにインストールする必要があります。特定の SPS の要件については、SPS for Linuxリリースノート を参照してください。

- **SIOS DataKeeper** ソフトウェア – SPS クラスタの各サーバには SIOS DataKeeper ソフトウェアが必要です。SIOS DataKeeper のインストールとアンインストールの手順については、SPS for Linux インストールガイド を参照してください。

# 5.5.3.2. 全般的な設定

- ターゲットのディスクまたはパーティションのサイズ（バックアップサーバ上）は、ソースのディスクまたはパーティションのサイズ（プライマリサーバ上）以上である必要があります。

- DataKeeper リソースを作成して拡張すると、同期プロセスにより ターゲットのディスクまたはパーティション に存在するデータが削除され、ソースのパーティションにあるデータに置き換えられます。

# 5.5.3.3. DataKeeper for Linux のネットワーク設定

- 各ペアのサーバー間でデータのレプリケーション用に選択するパスは、あらかじめそれらのサーバー間の LifeKeeper コミュニケーションパスとしても設定されている必要があります。ネットワークパスを変更する方法については、データレプリケーションパスの変更を参照してください。

- IP リカバリーキットがネットワークインターフェースを設定して仮想 IP アドレスを追加している間、通信回線が一時的に切断される場合があります。 これは、ネットワークインターフェイスを使用したミラーの再同期に影響を及ぼし、ミラーリングが予期せず停止する可能性があります。 したがって、IP リカバリーキットによって保護されている仮想 IP (VIP) アドレスを、レプリケーションに使用されているのと同じネットワークインターフェイス (NIC) に配置しないことをお勧めします。

- このリリースの SIOS DataKeeper は、DataKeeper リソースの Automatic スイッチバックをサポートしていません。さらに、 Automatic スイッチバックの制限は、DataKeeper リソースの上に存在する他の LifeKeeper リソースにも適用されます。

- Fusion-io を使用する場合のネットワーク設定情報については、Fusion-io を使用するクラスター化の「ネットワーク」セクションを参照してください。

# 5.5.3.4. データレプリケーションパスの変更

LK 7.1 から、lk_chg_value を使用して、ミラーのレプリケーションパスのIPアドレスを変更できるようになりました。例えば、ミラーのレプリケーションパスを IP アドレスの 192.168.0.1 から 192.168.1.1 に変更するには、以下の操作を行ってください。

```
# lkstop (lk_chg_value は、LifeKeeper の動作中は実行できません)。

# lk_chg_value -o 192.168.0.1 -n 192.168.1.1

# lkstart
```

この IP アドレスを使用するミラーに含まれるすべてのサーバで、これらのコマンドを実行してください。

> **注記:** このコマンドは、該当アドレスを使用するコミュニケーションパスも変更します。

# 5.5.3.5. ネットワーク帯域幅の要件の特定

SIOS DataKeeper をインストールする前に、現在の構成の複製に仮想マシンを使用するか、物理的な Linux サーバを使用するかによりネットワーク帯域幅の要件を特定する必要があります。仮想マシン（VM）を使用する場合は、Linux システム（物理または仮想）の更新頻度の測定 方法を使用して、複製を計画している仮想マシンの変化率を測定してください。この値は、仮想マシンの複製に必要となるネットワーク帯域幅を表します。

ネットワーク帯域幅の要件を特定した後、ネットワークが最大のパフォーマンスを発揮するように構成してください。ネットワーク帯域幅の要件が、現在使用できるネットワーク能力を超えている場合は、以下のオプションを 1 つ以上検討しなければならない可能性があります。

- SIOS DataKeeper（または可能な場合はネットワークハードウェア）の圧縮を有効にする。
- ネットワーク能力を増強する。
- 複製するデータ量を低減する。
- 一時データおよびスワップファイル用に、複製しないローカルのストレージリポジトリを作成する。
- 毎日、ピーク時以外に複製を手動でスケジュールする。

# 5.5.3.5.1. Linux システム（物理または仮想）での変化率の測定

データは使用できるネットワーク内で複製できます。マルチサイト、すなわち広域ネットワーク（WAN）構成では、「ソースパーティションが 1 日中更新されるときに、パーティションを正常に複製してミラーをミラーリング状態に維持するために十分な帯域幅があるか」という質問に対して特別な検討が必要です。

ミラーがミラーリング状態でない場合にはパーティションのスイッチオーバは許可されないので、ミラーをミラーリング状態に維持することが重要です。

データを非同期キューに追加することにより、短期間に急増した書き込み動作を処理します。ただし、長期間にわたって複製されるすべてのボリュームのディスク書き込み動作の合計が、平均してネットワークが送信できる変化量を下回ることを確認してください。

ネットワーク能力が不十分なためにディスクの変化率に対処できず、非同期キューがいっぱいになった場合、ミラーは同期動作に戻ります。これによりソースサーバのパフォーマンスに悪影響を及ぼすことがあります。

## 基本変化率の測定

以下のコマンドを使用して、ミラーリングするファイルまたはパーティションを特定してください。例えば /dev/sda3 を使用して、1 日に書き込まれたデータ量を測定します。

```
MB_START=`awk '/sda3 / { print $10 / 2 / 1024 }' /proc/diskstats`
```

1 日後

```
MB_END=`awk '/sda3 / { print $10 / 2 / 1024 }' /proc/diskstats`
```

1 日の変化率 (単位: MB) は `MB_END - MB_START` で得られます。

SIOS DataKeeper が 1 日にミラーリングできるおよその量は以下のとおりです。

T1 (1.5Mbps) – 14,000 MB/日(14 GB)

T3（1.5 Mbps） – 410,000 MB/日 (410 GB)

ギガビット（1 Gbps） - 5,000,000 MB/日（5 TB）

## 詳細変化率の測定

変化率を収集する最良の方法は、一定期間（例: 1 日）ディスクの書き込み動作をログに記録して、ディスクの書き込みのピーク期間を特定することです。

ディスクの書き込み動作を追跡するには、システムのタイムスタンプをログに記録して */proc/diskstats* の

ダンプを行う cron ジョブを作成してください。例えば、2 分間隔でディスクの統計値を収集するには、*/etc/crontab* に以下のリンクを追加します。:

```
*/2 * * * * root ( date ; cat /proc/diskstats ) >> /path_to/
filename.txt
```

1 日、1 週間などの期間が経過した後、cron ジョブを無効にし、得られたデータファイルを安全な場所に保存します。

# 収集した詳細変化率データの解析

roc-calc-diskstats ユーティリティは、前述の手順で収集したデータを解析します。このユーティリティは、長期間ログに記録された出力を持つ */proc/diskstats* 出力ファイルから、データセットに含まれるディスクの変化率を計算します。

roc-calc-diskstats をダウンロードするには ここをクリック

**使用法:**

```
# ./roc-calc-diskstats <interval> <start_time> <diskstats-data-file> [dev-
list]
```

**使用例（概要のみ）:**

```
# ./roc-calc-diskstats 2m “Jul 22 16:04:01” /root/diskstats.txt
sdb1,sdb2,sdc1 > results.txt
```

この例は、概要（およびディスク別のピーク I/O 情報）を *results.txt* にダンプします。

**使用例（概要とグラフデータ）:**

```
# export OUTPUT_CSV=1

# ./roc-calc-diskstats 2m “Jul 22 16:04:01” /root/diskstats.txt
sdb1,sdb2,sdc1 2> results.csv > results.txt
```

この例は、グラフデータを *results.csv* に、概要（およびディスク別のピーク I/O 情報）を *results.txt* にダンプします。

**結果の例（results.txt）**

```
Sample start time: Tue Jul 12 23:44:01 2011

Sample end time: Wed Jul 13 23:58:01 2011

Sample interval: 120s #Samples: 727 Sample length: 87240s
```

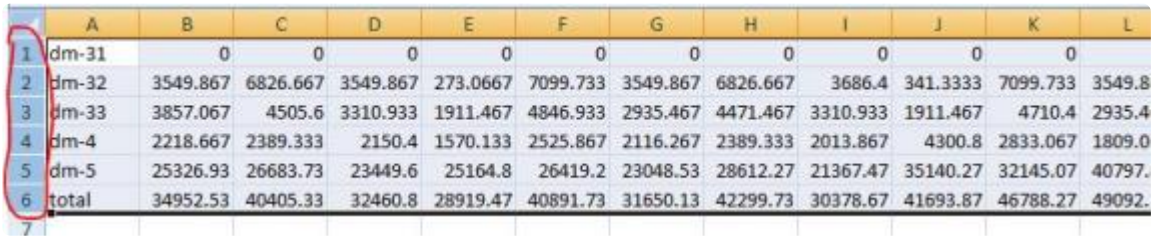
```
(Raw times from file: Tue Jul 12 23:44:01 EST 2011, Wed Jul 13 23:58:01
EST 2011)

Rate of change for devices dm-31, dm-32, dm-33, dm-4, dm-5, total

dm-31 peak:0.0 B/s (0.0 b/s) (@ Tue Jul 12 23:44:01 2011) average:0.0 B/s
(0.0 b/s)

dm-32 peak:398.7 KB/s (3.1 Mb/s) (@ Wed Jul 13 19:28:01 2011) average:19.5
KB/s (156.2 Kb/s)

dm-33 peak:814.9 KB/s (6.4 Mb/s) (@ Wed Jul 13 23:58:01 2011) average:11.6
KB/s (92.9 Kb/s)

dm-4 peak:185.6 KB/s (1.4 Mb/s) (@ Wed Jul 13 15:18:01 2011) average:25.7
KB/s (205.3 Kb/s)

dm-5 peak:2.7 MB/s (21.8 Mb/s) (@ Wed Jul 13 10:18:01 2011) average:293.0
KB/s (2.3 Mb/s)

total peak:2.8 MB/s (22.5 Mb/s) (@ Wed Jul 13 10:18:01 2011) average:349.8
KB/s (2.7 Mb/s)
```

# 詳細変化率データのグラフ作成

お客様に固有の経時的な帯域幅のニーズを分かりやすくするために、テンプレートスプレッドシート diskstats-template.xlsx が用意されています。このスプレッドシートにはサンプルデータがあり、`roc-calc-diskstats`で収集したデータで上書きできます。

diskstats-template.xslx をダウンロードするには[ここをクリック]

1. results.csv を開き、total 列を含めて **すべての行** を選択してください。



| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | dm-31 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 2 | dm-32 | 3549.867 | 6826.667 | 3549.867 | 273.0667 | 7099.733 | 3549.867 | 6826.667 | 3686.4 | 341.3333 | 7099.733 | 3549.8 |
| 3 | dm-33 | 3857.067 | 4505.6 | 3310.933 | 1911.467 | 4846.933 | 2935.467 | 4471.467 | 3310.933 | 1911.467 | 4710.4 | 2935.4 |
| 4 | dm-4 | 2218.667 | 2389.333 | 2150.4 | 1570.133 | 2525.867 | 2116.267 | 2389.333 | 2013.867 | 4300.8 | 2833.067 | 1809.0 |
| 5 | dm-5 | 25326.93 | 26683.73 | 23449.6 | 25164.8 | 26419.2 | 23048.53 | 28612.27 | 21367.47 | 35140.27 | 32145.07 | 40797. |
| 6 | total | 34952.53 | 40405.33 | 32460.8 | 28919.47 | 40891.73 | 31650.13 | 42299.73 | 30378.67 | 41693.87 | 46788.27 | 49092. |
| 7 | | | | | | | | | | | | |

2. **diskstats-template.xlsx** を開き、 **diskstats.csv** ワークシートを選択してください。

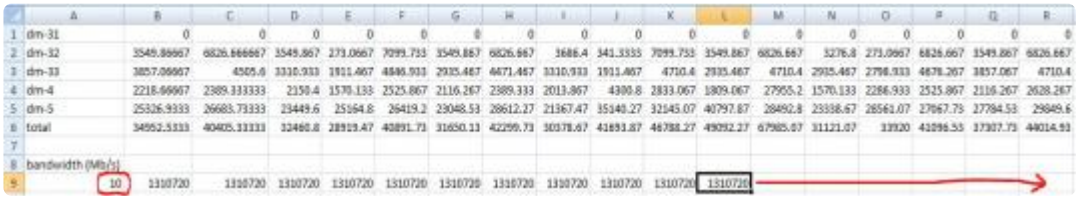
3. セル 1-A を右クリックし、 **[Insert Copied Cells]** を選択してください。

4. レプリケーション用に割り当てた帯域幅の量を反映するように、ワークシートの左下にあるセルの

**bandwidth** 値を調整してください。

単位: メガビット/秒（Mb/sec）

**注記:** その右側にあるセルの値は、収集した生データに合わせて自動的にバイト / 秒単位に変換されます。

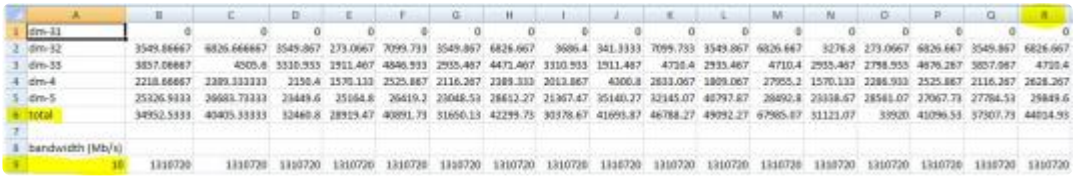


5. 以下の行/列番号を記録してください。

   a. Total(下のスクリーンショットでは行 6)

   b. Bandwidth(下のスクリーンショットでは行 9)

   c. 最終データポイント(下のスクリーンショットでは列 R)



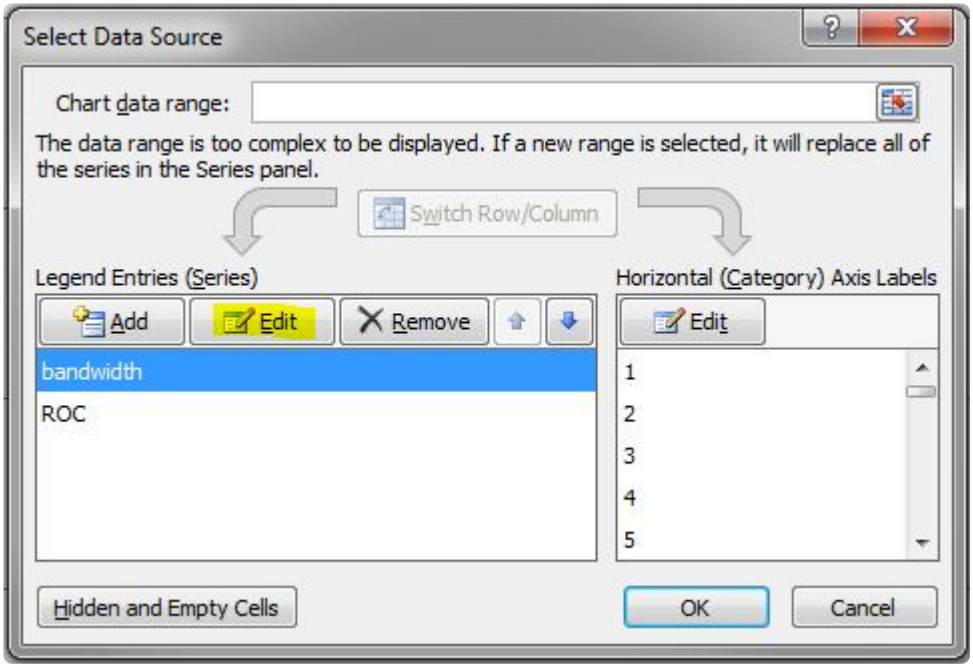
6. **bandwidth vs ROC** ワークシートを選択してください。

7. グラフを右クリックし、 **[Select Data...]** を選択してください。

   a. **Bandwidth 系列** を調整してください。

      i. 左の **[Series]** リストから **bandwidth** を選択してください。

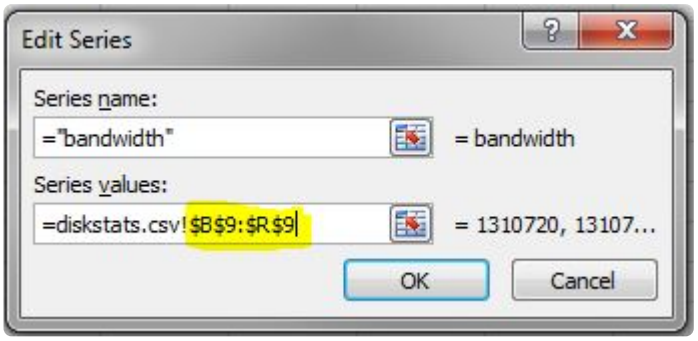
      ii. **[Edit]** をクリックしてください。




iii. 以下の構文を使用して、 **[Series Values]** フィールドを調整してください。

```
"=diskstats.csv!$B$<row>:$<final_column>$<row>"
```

例: "=diskstats.csv!$B$9:$R:$9"

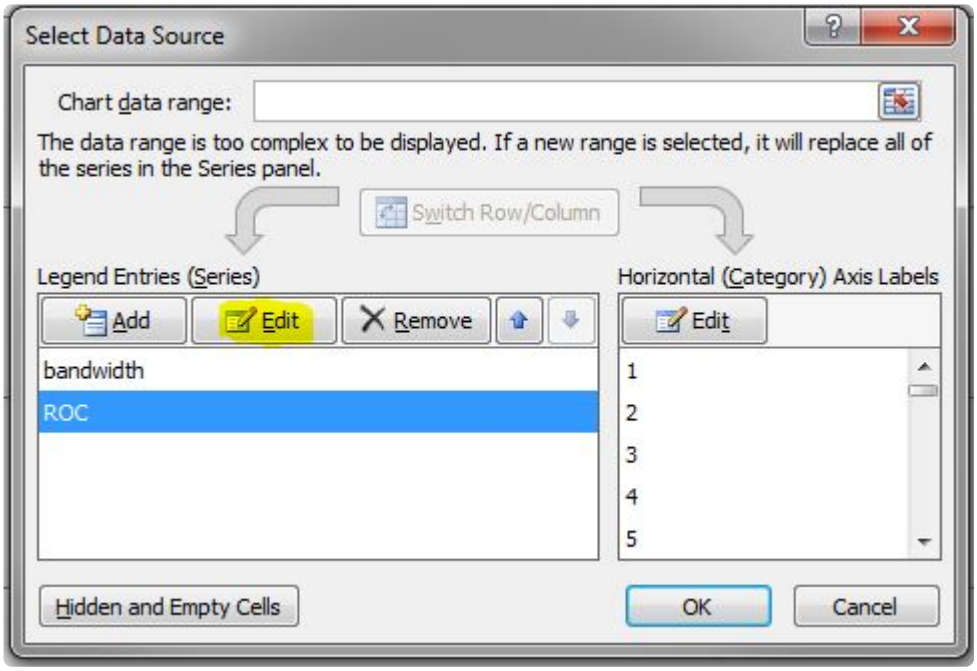



iv. **[OK]** をクリックしてください。

b. **ROC 系列** を調整してください。

i. 左の **[Series]** リストから **ROC** を選択してください。

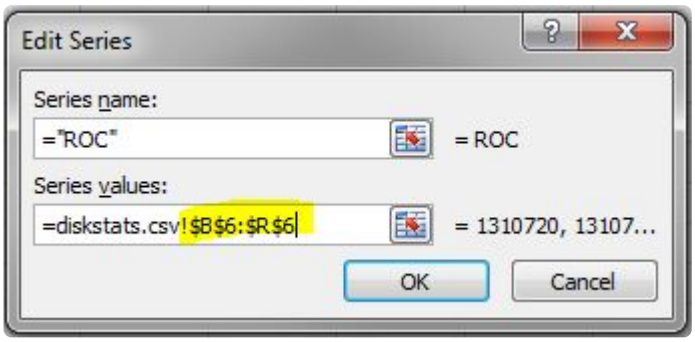
ii. **[Edit]** をクリックしてください。




iii. 以下の構文を使用して、 **[Series Values]** フィールドを調整してください。

"=diskstats.csv!$B$<row>:$<final_column>$<row>"

例:"=diskstats.csv!$B$6:$R:$6"




iv. **[OK]** をクリックしてください。

c. **[OK]** をクリックしてウィザードを終了してください。

8. Bandwidth vs ROC のグラフが更新されます。結果を解析して、データのレプリケーションをサポートするために十分な帯域幅があるかどうかを判断してください。

# roc-calc-diskstats

```
#!/usr/bin/perl
# Copyright (c) 2011, SIOS Technology, Corp.
# Author: Paul Clements
use strict;
sub msg {
        printf STDERR _;
}
sub dbg {
        return if (! $ENV{'ROC_DEBUG'});
        msg _;
}
$0 =~ s@^.*/@@; # basename
sub usage {
        msg "Usage: $0 <interval> <start-time> <iostat-data-file> [dev-lis
t]\n";
        msg "\n";
        msg "This utility takes a /proc/diskstats output file that contains\
n";
        msg "output, logged over time, and calculates the rate of change of\
n";
        msg "the disks in the dataset\n";
        msg "OUTPUT_CSV=1 set in env. dumps the full stats to a CSV file on ST
DERR\n";
        msg "\n";
        msg "Example: $0 1hour \"jun 23 12pm\" steeleye-iostat.txt sdg,sdh\n";
        msg "\n";
        msg "interval - interval between samples\n";
        msg "start time - the time when the sampling starts\n";
        msg "iostat-data-file - collect this with a cron job like:\n";
        msg "\t0 * * * * (date ; cat /proc/diskstats) >> /root/diskstats.txt\
n";
        msg "dev-list - list of disks you want ROC for (leave blank for al
l)\n";
        exit 1;
}
usage if (ARGV < 3);
my $interval = TimeHuman($ARGV[0]);
my $starttime = epoch($ARGV[1]);
my $file = $ARGV[2];
my $blksize = 512; # /proc/diskstats is in sectors
my %devs = map { $_ => 1 } split /,/, $ARGV[3];
my %stat;
my $firsttime;
my $lasttime;
```

```
# datestamp divides output
my %days = ( 'Sun' => 1, 'Mon' => 1, 'Tue' => 1, 'Wed' => 1,
                'Thu' => 1, 'Fri' => 1, 'Sat' => 1);
my %fields = ( 'major'  => 0,
                'minor' => 1,
                'dev' => 2,
                'reads' => 3,
                'reads_merged' => 4,
                'sectors_read' => 5,
                'ms_time_reading' => 6,
                'writes' => 7,
                'writes_merged' => 8,
                'sectors_written' => 9,
                'ms_time_writing' => 10,
                'ios_pending' => 11,
                'ms_time_total' => 12,
                'weighted_ms_time_total' => 13 );
my $devfield = $fields{'dev'};
my $calcfield = $ENV{'ROC_CALC_FIELD'} || $fields{'sectors_written'};
dbg "using field $calcfield\n";
open(FD, "$file") or die "Cannot open $file: $!\n";
foreach (<FD>) {
        chomp;
        _ = split;
        if (exists($days{$_[0]})) { # skip datestamp divider
                if ($firsttime eq '') {
                        $firsttime = join ' ', _[0..5];
                }
                $lasttime = join ' ', _[0..5];
                next;
        }
        next if ($_[0] !~ /[0-9]/); # ignore
        if (!%devs || exists $devs{$_[$devfield]}) {
                push {$stat{$_[$devfield]}}, $_[$calcfield];
        }
}
{$stat{'total'}} = totals(\%stat);
printf "Sample start time: %s\n", scalar(localtime($starttime));
printf "Sample end time: %s\n", scalar(localtime($starttime + (({$stat{'tota
l'}} - 1) * $interval)));
printf "Sample interval: %ss #Samples: %s Sample length: %ss\n", $interval,
({$stat{'total'}} - 1), ({$stat{'total'}} - 1) * $interval;
print "(Raw times from file: $firsttime, $lasttime)\n";
print "Rate of change for devices " . (join ', ', sort keys %stat) . "\n";
foreach (sort keys %stat) {
        my vals = {$stat{$_}};
        my ($max, $maxindex, $roc) = roc($_, $blksize, $interval, vals);
```

```
        printf "$_ peak:%sB/s (%sb/s) ( %s) average:%sB/s (%sb/s)\n", HumanSiz
e($max), HumanSize($max * 8), scalar localtime($starttime + ($maxindex * $inte
rval)), HumanSize($roc), HumanSize($roc * 8);
}
# functions
sub roc {
        my $dev = shift;
        my $blksize = shift;
        my $interval = shift;
        my ($max, $maxindex, $i, $first, $last, $total);
        my $prev = -1;
        my $first = $_[0];
        if ($ENV{'OUTPUT_CSV'}) { print STDERR "$dev," }
        foreach (_) {
                if ($prev != -1) {
                        if ($_ < $prev) {
                                dbg "wrap detected at $i ($_ < $prev)\n";
                                $prev = 0;
                        }
                        my $this = ($_ - $prev) * $blksize / $interval;
                        if ($this > $max) {
                                $max = $this;
                                $maxindex = $i;
                        }
                        if ($ENV{'OUTPUT_CSV'}) { print STDERR "$this," }
                }
                $prev = $_; # store current val for next time around
                $last = $_;
                $i++;
        }
        if ($ENV{'OUTPUT_CSV'}) { print STDERR "\n" }
        return ($max, $maxindex, ($last - $first) * $blksize / ($interval *
($i - 1)));
}
sub totals { # params: stat_hash
        my $stat = shift;
        my totalvals;
        foreach (keys %$stat) {
                next if (!defined($stat{$_}));
                my vals = {$stat{$_}};
                my $i;
                foreach (vals) {
                        $totalvals[$i++] += $_;
                }
        }
        return totalvals;
}
```

```
# converts to KB, MB, etc. and outputs size in readable form
sub HumanSize { # params: bytes/bits
        my $bytes = shift;
        my suffixes = ( '', 'K', 'M', 'G', 'T', 'P' );
        my $i = 0;
        while ($bytes / 1024.0 >= 1) {
                $bytes /= 1024.0;
                $i++;
        }
        return sprintf("%.1f %s", $bytes, $suffixes[$i]);
}
# convert human-readable time interval to number of seconds
sub TimeHuman { # params: human_time
        my $time = shift;
        my %suffixes = ('s' => 1, 'm' => 60, 'h' => 60 * 60, 'd' => 60 * 60 *
24);
        $time =~ /^([0-9]*)(.*?)$/;
        $time = $1;
        my $suffix = (split //, $2)[0]; # first letter from suffix
        if (exists $suffixes{$suffix}) {
                $time *= $suffixes{$suffix};
        }
        return $time;
}
sub epoch { # params: date
        my $date = shift;
        my $seconds = `date +'%s' --date "$date" 2>&1`;
        if ($? != 0) {
                die "Failed to recognize time stamp: $date\n";
        }
        return $seconds;
}
```

# 5.5.3.6. WAN の設定

WAN ネットワークの特性により、SIOS DataKeeper を WAN 環境で使用するには特殊な設定が必要です。以下のヒントに従うことが推奨されます。

- 意図しないフェイルオーバを防止するために manual failover confirmation を有効にする必要があります。多くの WAN は LAN よりも信頼性が低い上、WAN のミラー設定はコミュニケーションパスが 1 本だけになるため、通常得策といえます。このオプションを有効にすると、LifeKeeper のフェイルオーバはユーザが `lk_confirmso` コマンドを使用してフェイルオーバを確認した場合にのみ実行されます。詳細については、 `lk_confirmso` のマニュアルページを参照してください。

- WAN の待機時間とスループットに基づいて、 `LKDR_ASYNC_LIMIT` の適切な値を決定してください。 `LKDR_ASYNC_LIMIT` パラメータ ( */etc/default/LifeKeeper* に設定) が、SIOS DataKeeper が許可する未処理の書き込み動作 (ミラーあたり) の数を指定します。このパラメータのデフォルト値は 4096 ですが、大きい値を指定するとミラーの書き込みパフォーマンスを向上できます。この値を増加することの短所は、どの時点でもステータスとセカンダリとの間で非同期になっているデータ量が多くなることです。 `LKDR_ASYNC_LIMIT` パラメータについての詳細は **SIOS DataKeeper for Linux** によるミラーリング の非同期ミラーリング を参照してください。.

- 低速の WAN リンク経由で大量のデータをミラーリングする場合、最初のデータ全体の再同期を行わずに、ソースのディスクまたはパーティションのコピーをリモート（ディザスタリカバリ）サイトに輸送するのが望ましいことがあります。最初の再同期を回避するには、全体の再同期の回避 の手順に従ってください。

> **重要 :** LifeKeeper の GUI にある [New Replicated Filesystem] オプションを使用して階層を作成した場合、この手順は不要です。[New Replicated Filesystem] オプションは、最初の全体の再同期を回避するように最適化されています。

- WAN リンクに 15 秒以上のダウンタイム期間が定期的に発生する場合、LifeKeeper のハートビートパラメータを調整することが推奨されます。詳細については、LifeKeeper のハートビートの調整 を参照してください。

# 5.5.3.7. SIOS DataKeeper for Linux のリソースタイプ

DataKeeper リソース階層を作成するときに、リソースタイプを選択するように LifeKeeper から要求されます。DataKeeper リソースには、いくつかのタイプがあります。お使いの環境に最適なタイプを選択するときに、以下の情報が役立ちます。

## Replicate New File System

Replicate New File System を選択すると、NetRAID デバイスが作成/拡張され、NetRAID デバイスに指定のマウントポイントがマウントされます。また、LifeKeeper がサポートするファイルシステムと NetRAID デバイスの両方が、LifeKeeper で保護されます。ローカルのディスクまたはパーティションがフォーマットされます。

このリソースが2番目のノードに拡張されると、完全再同期は行われず、ファイルシステムのメタデータデータのみ再同期が行われます。ただし、3番目以上のノードに拡張されると、そのノードに対して完全再同期が行われます。完全再同期を回避するには、全同期の回避 に記載されている手順に従ってください。

注意：データがすべて削除されます。

## Replicate Existing File System

Replicate Existing File System を選択すると、現在マウントされているディスクまたはパーティションが使用され、ディスクまたはパーティションのデータが削除されることなく NetRAID デバイスが作成されます。SIOS DataKeeper はローカルのディスクまたはパーティションをアンマウントし、ローカルのディスクまたはパーティションを使用して NetRAID デバイスを作成します。そして、NetRAID デバイスにマウントポイントをマウントします。次に、NetRAID デバイスと LifeKeeper がサポートするファイルシステムの両方を LifeKeeper で保護します。

**重要:** SIOS Protection Suite for Linux のマルチサイトクラスタ階層を作成する場合、作成プロセス中にアプリケーションが停止します。階層の作成と拡張が完了した後、アプリケーションを再起動する必要があります。

## DataKeeper Resource

DataKeeper リソース を選択すると、NetRAID デバイスが作成/拡張され、ファイルシステムは含めずに LifeKeeper で保護されます。RAW I/O デバイスを使用できるデータベースを使用している場合は、この複製タイプを選択できます。

ユーザがデータアクセスを続行できるように、SIOS DataKeeper は、現在マウントされている NetRAID デバイスのアンマウントと削除は実行しません。ユーザは、手動スイッチオーバの前に NetRAID デバイスを手動でアンマウントし、手動スイッチオーバの後に他のサーバにマウントする必要があります。

**注記:** DataKeeper リソースの作成後に、手動マウントしたファイルシステムを LifeKeeper で保護する場合は、以下の操作を行います。

1. LifeKeeper がサポートするファイルシステムで、NetRAID デバイスをフォーマットしてください。

2. NetRAID デバイスをマウントしてください。

3. NetRAID デバイスを使用して、共有ストレージのディスクまたはパーティションにあるかのようにファイルシステムの階層を作成し、拡張してください。

これで、LifeKeeper のファイルシステムリカバリキットが、フェイルオーバ時のファイルシステムのマウント / アンマウントを実行します。

# 5.5.3.8. DataKeeper 構成での I/O フェンシング

DataKeeper 構成では、原理上ストレージのリザベーションを使用した I/O フェンシングを使用することが出来ないため、単体ではスプリットブレイン状態を回避することが出来ません。
そのため、以下のような方法で DataKeeper リソースがスプリットブレイン状態に陥ることを防止する必要があります。

## IP リソースを使用した排他制御

IP リソースには重複チェックを利用した排他制御機能があるため、複数ノードで同一の IP リソースが起動することはありません。この性質を利用して DataKeeper リソースのスプリットブレインを防止することが出来ます。
DataKeeper の子リソースに適当な仮想 IP アドレスを割り当てたIPリソースを配置することで、親になる DataKeeper リソースが複数ノードで同時に起動することを防止します。
なお、IP リソースの重複チェックはすべてのノードが同一サブネットに所属する必要があるため、この手法はローカルネットワークだけで構成された環境のみ適用できます。

## Quorum/Witness 機能による排他制御

LifeKeeper が持つ Quorum/Witness 機能を使用することで、複数ノードが同時にアクティブ状態になることを防止します。
Quorum/Witness 機能の詳細はテクニカルドキュメンテーションの Quorum/Witness を参照してください。

# 5.5.3.9. リソースの設定作業

SIOS DataKeeper の設定作業はすべて、LifeKeeper のグラフィカルユーザインターフェース（GUI）から実行できます。LifeKeeper の GUI では、SIOS DataKeeper のリソースの設定、管理、監視の作業をガイド付きで行うことができます。

## 概要

SIOS DataKeeper の設定に関して、以下の作業を行うことができます。

- **Create a Resource Hierarchy** - DataKeeper リソース階層を作成します。
- **Delete a Resource Hierarchy** - DataKeeper リソース階層を削除します。
- **Extend a Resource Hierarchy** - DataKeeper リソース階層をプライマリサーバからバックアップサーバに拡張します。
- **Unextend a Resource Hierarchy** - LifeKeeper クラスタ内にある 1 台のサーバの DataKeeper リソース階層を拡張解除（削除）します。
- **Create Dependency** - 既存のリソース階層と別のリソースインスタンスとの間に子の依存関係を作成し、クラスタ内のすべての対象サーバに依存関係の変更を伝播します。
- **Delete Dependency** - リソースの依存関係を削除して、クラスタ内にあるすべての対象サーバに依存関係の変更を伝播します。
- **In Service** - リソース階層をアクティブにします。
- **Out of Service** - リソース階層を非アクティブにします。
- **View/Edit Properties** - リソース階層のプロパティの表示または編集を行います。

# 5.5.3.9.1. DataKeeper リソース階層の作成

プライマリーサーバで以下の操作を行ってください。

1. **[Edit] > [Server] > [Create Resource Hierarchy]** を選択してください。

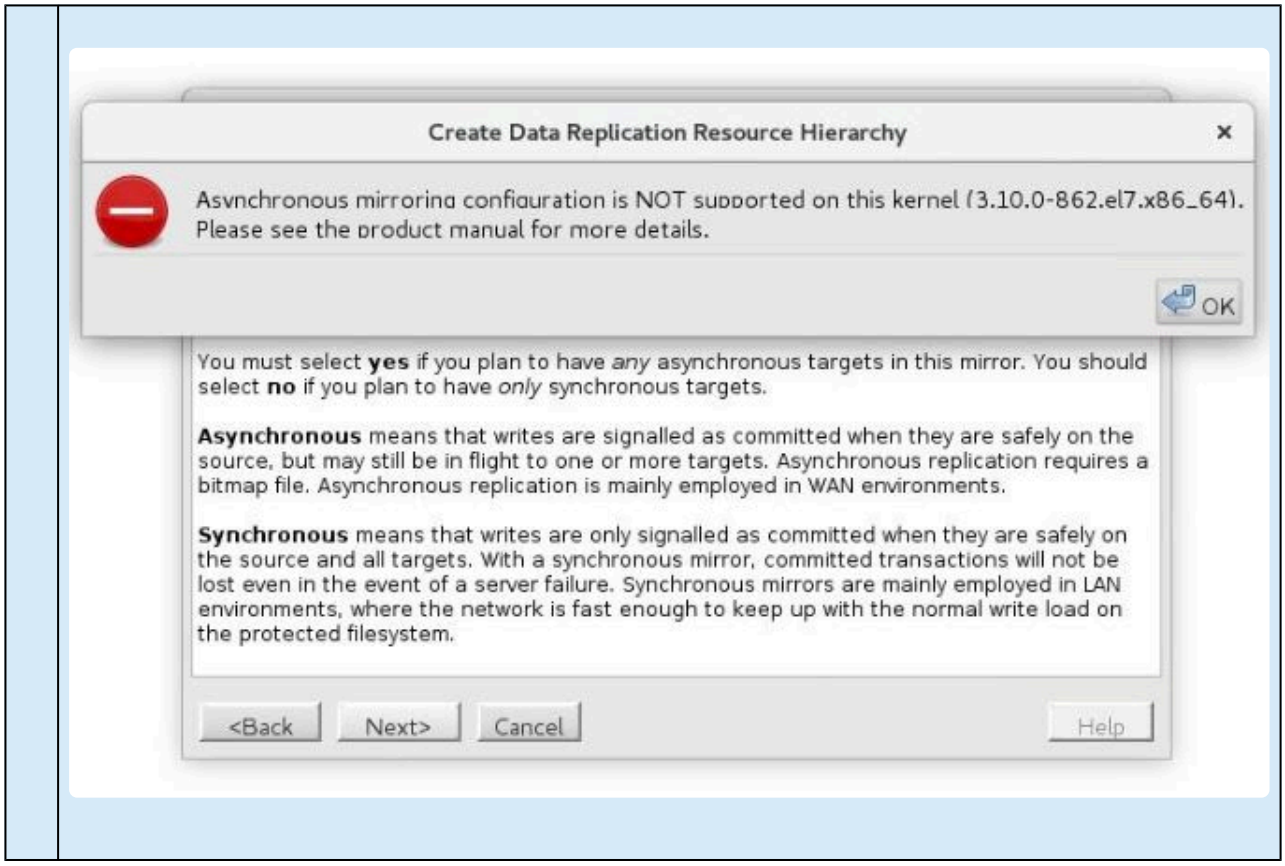
   ***[Create Resource Wizard]*** ダイアログボックスが表示されます。

2. ドロップダウンリストから **[Data Replication]** オプションを選択し、 **[Next]** をクリックして続行してください。

3. 以下の情報を入力するように要求されます。ダイアログボックスで **[Back]** ボタンが有効な場合は、前のダイアログボックスに戻ることができます。これは、エラーが発生して、前に入力した情報を修正する必要がある場合に便利な機能です。いつでも **[Cancel]** をクリックして、作成処理全体を取り消すことができます。

| フィールド | ヒント |
|---|---|
| Switchback Type | **[intelligent switchback]** を指定する必要があります。これは、バックアップサーバにフェイルオーバした後、管理者が手動で DataKeeper リソースをプライマリーサーバにスイッチバックする必要があることを意味します。<br><br>**注意:** このリリースの SIOS DataKeeper は、DataKeeper リソースの Automatic スイッチバックをサポートしていません。さらに、 Automatic スイッチバックの制限は、DataKeeper リソースを含む階層内の他の LifeKeeper リソースにも適用されます。 スイッチバックは、DataKeeper リソースを含む階層を移動しようとしません。 |
| Server | 作成する NetRAID デバイスが存在するサーバ（通常はプライマリーサーバー）の名前を選択してください。ドロップダウンリストボックスには、クラスタ内のすべてのサーバが表示されます。 |
| Hierarchy Type | 以下のいずれかを選択して、作成するデータレプリケーションのタイプを選択してください。<br>• Replicate New File System<br>• Replicate Existing File System<br>• DataKeeper Resource |
| Bi | インテントログの記録に使用するビットマップファイルの名前を入力してください。 **[none]** を |

| | |
|---|---|
| tmapFile | 入力すると、インテントログは使用されず、すべての再同期が部分的ではなく全体の再同期になります。<br><br>**重要:** ビットマップファイルは btrfs ファイルシステム（またはサポートされていないその他の LifeKeeper for Linux ファイルシステム）上に置いてはいけません。データレプリケーションのビットマップファイルが btrfs ファイルシステム（またはサポートされていないその他の LifeKeeper for Linux ファイルシステム）上に置かれると、LifeKeeper がミラーを構成しようとした時、"invalidargument" エラーの原因になります。ビットマップファイルのデフォルトの置き場所は、 /opt/LifeKeeper の下です。このデフォルトの置き場所は、/opt/LifeKeeper が btrfs ファイルシステム上にある場合変更されます。<br><br>**注記:** btrfs は、現在 LifeKeeper for Linux でサポートされていません。 |
| EnableAsynchronousReplication? | このレプリケーションリソースによるターゲットシステムへの非同期レプリケーションのサポートを許可するには、 **[Yes]** を選択してください。すべてのターゲットについて同期レプリケーションを使用する場合は、 **[No]** を選択してください。後で、レプリケーションリソースが各ターゲットサーバに拡張されるときに、実際のレプリケーションタイプ（同期または非同期）を選択するように要求されます。(両方のレプリケーションタイプの詳細については、SIOS DataKeeper によるミラーリング を参照してください)。これらのターゲットのいずれかへのレプリケーションを非同期で実行する場合は、他のターゲットへのレプリケーションを同期実行する場合でもここでは **[Yes]** を選択する必要があります。<br><br>**注記:** 非同期ミラーリングをサポートしない環境で非同期ミラーリングを選択した場合は、以下のメッセージが表示されます。<br>Asynchronous mirroring configuration is NOT supported on this kernel (<linuxカーネルのバージョン>). Please see the product manual for more details. |




以降の一連のダイアログボックスは、**[Hierarchy Type]** で選択した項目によって異なります。一部のダイアログボックスはすべての階層タイプで同じですが、表示される順序と必要な情報が少し異なることがあります。以下の 3 つのトピックで、階層作成の残りのプロセスについて説明しています。

- DataKeeper Resource
- New Replicated File System
- Replicate Existing File System

# 5.5.3.9.1.1. Replicate New File System

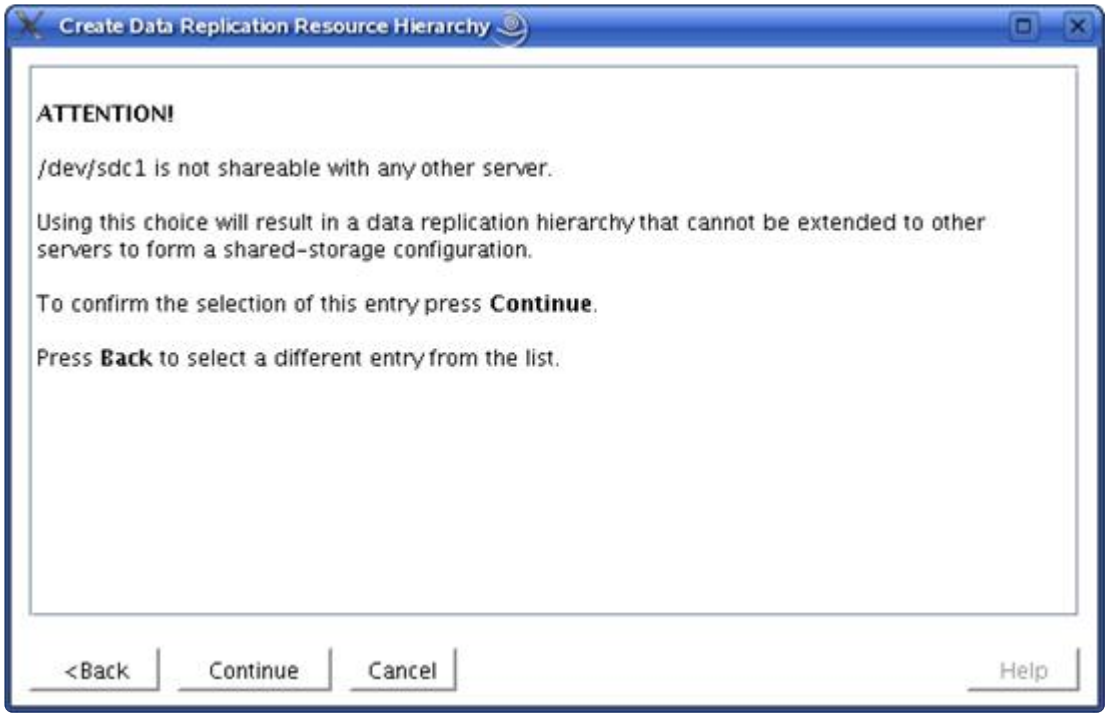
このオプションは、NetRAID デバイスを作成し、LifeKeeper がサポートするファイルシステムタイプでフォーマットします。ファイルシステムを NetRAID デバイスにマウントし、マウントしたファイルシステムと NetRAID デバイスの両方を LifeKeeper で保護します。NetRAID デバイスとローカルのディスクまたはパーティションがフォーマットされ、既存のデータが削除されます。新しいファイルシステムにミラーを作成し、LifeKeeper で保護する場合にこのオプションを選択してください。このリソースタイプには、1 つの空いているディスクまたはパーティションが必要です。

**注意:** このオプションを選択すると、ローカルのディスクまたはパーティションがフォーマットされ、既存のデータがすべて削除されます。

1. 要求されたら、以下の情報を入力してください。

| フィールド | ヒント |
|---|---|
| Source Disk or Partition | ドロップダウンリストには、 以下のものを除いて 、使用できるすべてのディスクが表示されます。<br>• 現在マウントされているもの<br>• スワップディスクまたはスワップパーティション<br>• LifeKeeper が保護するディスクまたはパーティション<<br><br>ドロップダウンリストには、root (*/*)、boot (*/boot*)、 */proc* 、floppy、cdrom などの特殊なディスクまたはパーティションも表示されません。<br><br>**注記:** ソースディスクまたはパーティションは一意に識別できる必要があります。v9.5.0 より、LifeKeeper ではデバイス名を使用してデバイスを識別できなくなります。 |

2. 非共有のソースのマウントポイントを選択した場合、以下の画面が表示されます。




3. 残りの情報を指定してリソースの構成を完了してください。

| フィールド | ヒント |
|---|---|
| New Mount Point | 新しいファイルシステムの **新しいマウントポイント** を入力してください。これは、複製したディスクまたはパーティションが配置されるマウントポイントです。 |
| New File System Type | **ファイルシステムタイプを** 選択します。LifeKeeper がサポートするファイルシステムタイプのみを選択できます。 |
| DataKeeper Resource Tag | DataKeeper リソースインスタンスの一意の DataKeeper リソースタグ名を選択するか、入力してください。 |
| File System Resource Tag | ファイルシステムリソースインスタンスのファイルシステムリソースタグを選択するか、入力してください。 |
| Bitmap File | ビットマップのファイル名を入力してください。デフォルトのファイル名は /opt/LifeKeeper/bitmap_$TAG です。<br><br>**重要:** ビットマップファイルは btrfs ファイルシステム（または LifeKeeper for Linux がサポートしていないその他のファイルシステム）上に置いてはいけません。これらのファイルシステム上に置くと、LifeKeeper がミラーを構成しようとした時 "invalid argument" エラーが発生します。 |

4. **[Next]** をクリックして、 **確認** 画面に進んでください。

5. 確認画面に、新しいファイルシステムの作成場所、およびローカルのディスクまたはパーティションについて保留中の再フォーマットに関する警告が表示されます。 **[Create]** をクリックして、 **リソ**

ースの作成 を開始します。

6. リソースを新しいファイルシステムに作成するために有効なデータを指定したかどうかが、LifeKeeper により検証されます。LifeKeeper が問題を検知した場合は、情報ボックスにエラーが表示されます。検証が正常に完了すると、リソースが作成されます。ディスクまたはパーティションのサイズにより、ファイルシステムの作成には数分かかることがあります。

   **[Next]** をクリックして次に進んでください。

7. 新しい複製ファイルシステムのリソース階層が正常に作成されたことを示す情報ボックスが表示されます。複製を開始してリソース階層を LifeKeeper で保護するには、クラスタ内の別のサーバにリソース階層を **拡張** する必要があります。

   リソースを拡張する場合は **[Next]** 、後でリソースを拡張する場合は **[Cancel]** をクリックしてください。

   **[Continue]** をクリックすると、 ***Pre-extend Wizard*** が起動します。リソース階層を別のサーバに拡張する方法の詳細については、 リソース階層の拡張 の手順 2 を参照してください。

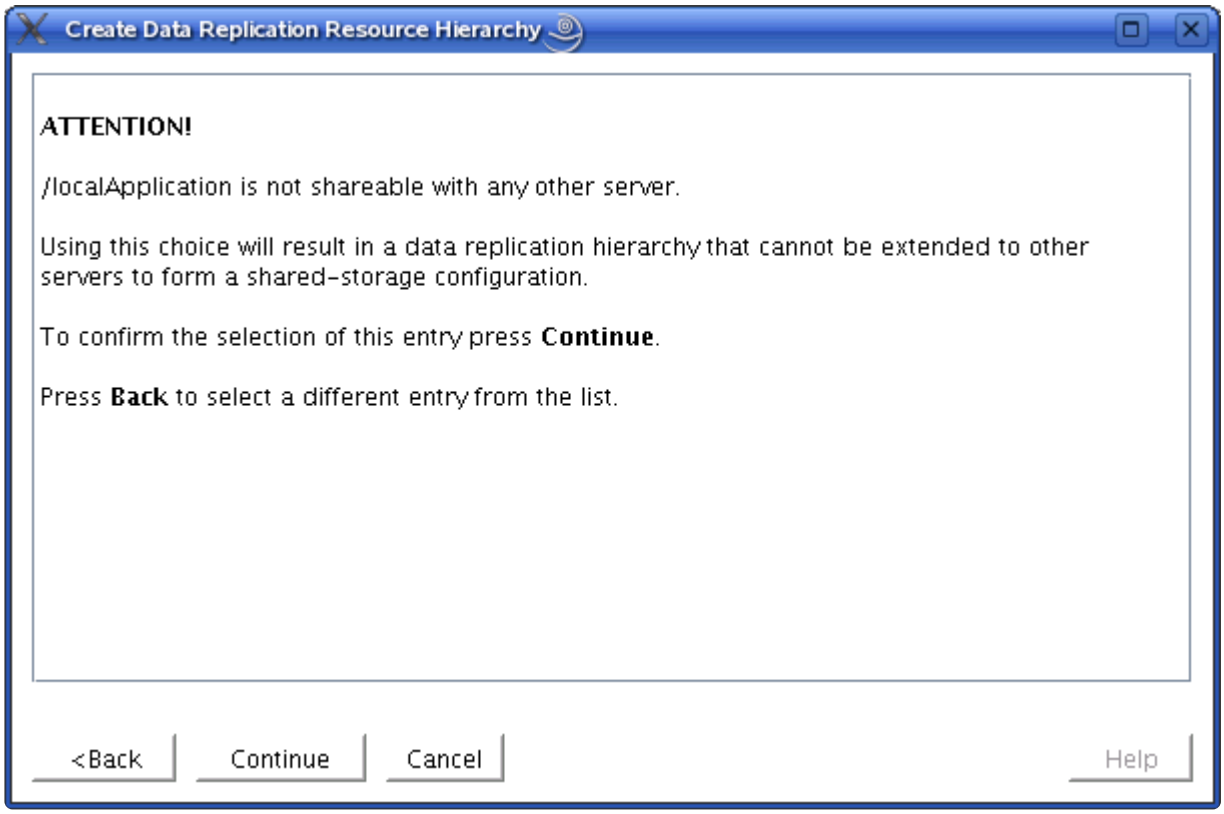
# 5.5.3.9.1.2. Replicate Existing File System

このオプションは、ローカルのディスクまたはパーティションに現在マウントされているファイルシステムをアンマウントし、NetRAID デバイスを作成して、ファイルシステムを NetRAID デバイスに再マウントします。NetRAID デバイスとマウントされたファイルシステムの両方が、LifeKeeper で保護されます。既存のファイルシステムにミラーを作成し、LifeKeeper で保護する場合にこのオプションを選択してください。

1. 要求されたら、以下の情報を入力してください。

| フィールド | ヒント |
|---|---|
| Existing Mount Point | これは、プライマリサーバの NetRAID デバイスにマウントするマウントポイントです。ローカルのディスクまたはパーティションがすでに、このマウントポイントにマウントされている必要があります。<br><br>**注記:** ソースディスクまたはパーティションは一意に識別できる必要があります。v9.5.0 より、LifeKeeper ではデバイス名を使用してデバイスを識別できなくなります。 |

2. 非共有のソースのマウントポイントを選択した場合、以下の画面が表示されます。




3. 残りの情報を指定してリソースの構成を完了してください。

| フィールド | ヒント |
| --- | --- |
| DataKeeper Resource Tag | DataKeeper リソースインスタンスの一意の **DataKeeper リソースタグ名** を選択するか、入力してください。 |
| File System Resource Tag | **ファイルシステムリソースタグ** の名前を選択するか、入力してください。 |
| Bitmap File | ビットマップのファイル名を入力してください。デフォルトのファイル名は /opt/LifeKeeper/bitmap_$TAG です。<br><br>**重要:** ビットマップファイルは btrfs ファイルシステム（または LifeKeeper for Linux がサポートしていないその他のファイルシステム）上に置いてはいけません。これらのファイルシステム上に置くと、LifeKeeper がミラーを構成しようとした時 "invalid argument" エラーが発生します。 |

4. **[Next]** をクリックして、DataKeeper リソースをプライマリサーバに作成してください。

5. DataKeeper リソースのを作成するために有効なデータを指定したかどうかが、LifeKeeper により検証されます。LifeKeeper が問題を検知した場合は、情報ボックスにエラーが表示されます。検証が正常に完了すると、リソースが作成されます。

   **[Next]** をクリックしてください。

6. 既存のレプリケーションファイルシステムのリソース階層が正常に作成されたことを示す情報ボックスが表示されます。レプリケーションを開始してリソース階層を LifeKeeper で保護するには、クラスタ内の別のサーバにリソース階層を **拡張** する必要があります。

   リソースを拡張する場合は **[Next]** 、後でリソースを拡張する場合は **[Cancel]** をクリックしてください。

   **[Continue]** をクリックすると、 *Pre-extend Wizard* が起動します。リソース階層を別のサーバに拡張する方法の詳細については、 リソース階層の拡張の手順 2 を参照してください。

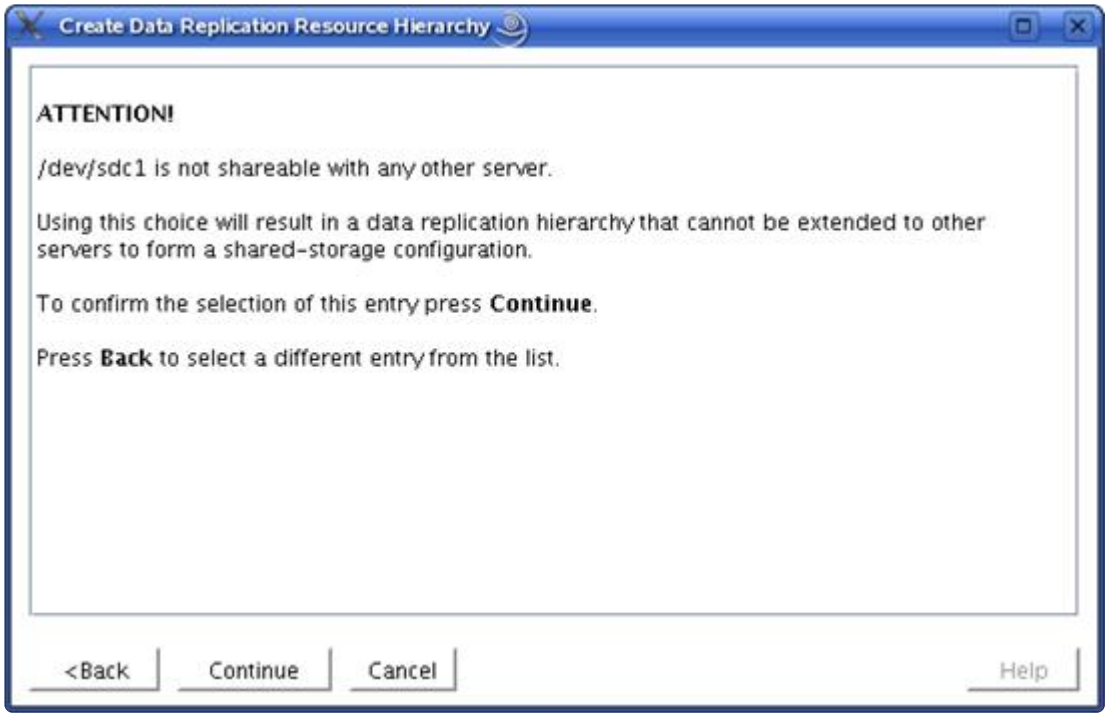
# 5.5.3.9.1.3. DataKeeper Resource

このオプションは、NetRAID デバイスのみを作成し（ファイルシステムは作成しない）、NetRAID デバイスを LifeKeeper で保護します。ディスクまたはパーティション上に DataKeeper デバイスのみを作成し、LifeKeeper で保護する場合にこのオプションを選択してください。読み取り可能なミラーを作成するには、このデバイス上にファイルシステムを作成し、マウントする操作を手動で行う必要があります。このリソースタイプには、1 つの空いているディスクまたはパーティションが必要です。

1. 要求されたら、以下の情報を入力してください。

| フィールド | ヒント |
|---|---|
| Source Disk or Partition | ドロップダウンボックスのソースディスクまたはパーティションのリストには、以下のものを除いて、使用できるすべてのディスクが表示されます。<br>• ○ 現在マウントされているもの<br>• ○ スワップディスクまたはスワップパーティション<br>• ○ LifeKeeper が保護するディスクまたはパーティション<br><br>ドロップダウンリストには、root (*/*)、boot (*/boot*)、*/proc* 、floppy、cdrom などの特殊なディスクまたはパーティションも表示されません。<br><br>**注記:** VMware を使用する場合は、「VMware の既知の問題」を参照してください。<br><br>**注記:** ソースディスクまたはパーティションは一意に識別できる必要があります。v9.5.0 より、LifeKeeper ではデバイス名を使用してデバイスを識別できなくなります。 |

2. 非共有のソースのマウントポイントを選択した場合、以下の画面が表示されます。




3. 残りの情報を指定してリソースの構成を完了してください。

| フィールド | ヒント |
|---|---|
| DataKeeper Resource Tag | DataKeeper リソースインスタンスの一意の **DataKeeper リソースタグ名** を選択するか、入力してください。 |
| Bitmap File | ビットマップのファイル名を入力してください。デフォルトのファイル名は /opt/LifeKeeper/bitmap_$TAG です。<br><br>**重要:** ビットマップファイルは btrfs ファイルシステム（または LifeKeeper for Linux がサポートしていないその他のファイルシステム）上に置いてはいけません。これらのファイルシステム上に置くと、LifeKeeper がミラーを構成しようとした時 "invalid argument" エラーが発生します。 |

4. **[Next]** をクリックします。

5. 使用する前に、ファイルシステムを手動で作成し、NetRAID デバイス（*/dev/mdX*）にマウントする必要があることを示す情報ウィンドウが表示されます。

   **[Create]** をクリックして、DataKeeper デバイスをローカルのディスクまたはパーティションに作成してください。

6. 情報ボックスが表示され、DataKeeper リソースのを作成するために有効なデータを指定したかどうかが、LifeKeeper により検証されます。LifeKeeper が問題を検知した場合は、情報ボックスにエラーが表示されます。検証が正常に完了すると、リソースが作成されます。

   **[Next]** をクリックして次に進んでください。

7. DataKeeper リソースデバイスが正常に作成されたことを示す情報ボックスが表示されます。データの複製を開始し、バックアップ/ターゲットサーバを LifeKeeper で保護するには、クラスタ内の別のサーバに階層を **拡張** する必要があります。

   リソースを拡張する場合は **[Continue]** 、後でリソースを拡張する場合は **[Cancel]** をクリックしてください。

   **[Continue]** をクリックすると、 ***Pre-extend Wizard*** が起動します。リソース階層を別のサーバに拡張する方法の詳細については、 リソース階層の拡張 の手順 2 を参照してください。

# 5.5.3.9.2. リソース階層の拡張

この操作は **[Edit]** メニューから開始できます。または **[Create Resource Hierarchy]** オプションの動作が完了すると自動的に開始されます。その場合は、手順 2 を参照してください。

1. **[Edit]** メニューの **[Resource]** から **[Extend Resource Hierarchy]** を選択します。 ***Pre-Extend Wizard*** が表示されます。拡張操作に慣れていない場合は、 **[Next]** をクリックしてください。LifeKeeper の **[Extend Resource Hierarchy]** のデフォルト値が分かっていて、入力と確認を省略する場合は **[Accept Defaults]** をクリックしてください。

2. ***Pre-Extend Wizard*** に以下の情報を入力します。

   **注記:** 最初の 2 つのフィールドは ***[Edit]*** メニューから拡張を開始した場合にのみ表示されます。

| フィールド | ヒント |
|---|---|
| *Template Server* | 現在 In Service のDataKeeper リソース階層 が存在するテンプレートサーバを選択してください。ここで選択する **テンプレートサーバ** と次のダイアログボックスで選択する **拡張するタグ** によって、 *In Service* （アクティブ）のリソース階層が表されることを理解しておくことが重要です。<br><br>選択したテンプレートサーバで In Service でないリソースタグを選択した場合、エラーメッセージが表示されます。このダイアログのドロップダウンボックスには、クラスタ内にある全サーバの名前が表示されます。 |
| *Tag to Extend* | これは、テンプレートサーバからターゲットサーバに拡張する *DataKeeper* インスタンスの名前です。ドロップダウンボックスには、テンプレートサーバ上に作成したすべてのリソースが表示されます。 |
| *Target Server* | 拡張先のサーバを入力するか、選択してください。 |
| *Switchback Type* | *[intelligent switchback]* を指定する必要があります。これは、バックアップサーバにフェイルオーバした後、管理者が手動で *DataKeeper* リソースをプライマリサーバにスイッチバックする必要があることを意味します。<br><br>**注意**: このリリースの *SIOS DataKeeper* は、 *DataKeeper* リソースの自動スイッチバックをサポートしていません。さらに、自動スイッチバックの制限は、 *SIOS DataKeeper* リソースの上に存在する他の *LifeKeeper* リソースにも適用されます。 |
| *Template Priority* | テンプレートの優先順位を選択するか、入力してください。これはサーバで現在 *In Service* の *DataKeeper* 階層の優先順位です。 *1* ～ *999* の範囲で、まだ優先順位として使用されていない値が有効で、小さい数字ほど優先順位が高くなります(数値 *1* が最高の優先順位)。拡張処理時に、別のシステムですでに使用中の優先順位をこの階層に対して指定することはできません。デフォルト値をが推奨されます。<br><br>**注記**: このフィールドは、階層をはじめて拡張するときにだけ表示されます。 |
| *Target* | ターゲットの優先順位を選択するか、入力してください。これは、他のサーバにある |

| | |
|---|---|
| *Priority* | 同等の階層に対する、新しく拡張する *DataKeeper* 階層の優先順位です。 *1* ～ *999* の範囲で、まだ優先順位として使用されていない値が有効で、リソースのカスケーディングフェイルオーバシーケンスにおけるサーバの優先順位を示します。数値が小さいほど優先順位は高くなります（数値 *1* が最高の優先順位）。 *LifeKeeper* のデフォルトでは、階層が作成されたサーバに「 *1* 」が割り当てられることに注意してください。優先順位は連続している必要はありませんが、特定のリソースについて 2 つのサーバに同じ優先順位を割り当てることはできません。 |

拡張前のチェックが正常に終了したというメッセージが表示されたら、 *[Next]* をクリックしてください。

拡張する階層に応じて、拡張されるリソースタグ（一部編集不可）を示す一連の情報ボックスが表示されます。

3. ***[Next]*** をクリックして、 ***[Extend Resource Hierarchy]*** の構成タスクを開始してください。

4. 次のセクションには、別のサーバに *DataKeeper* リソースを拡張するために必要な手順を示します。

## DataKeeper リソース階層の拡張

1. *pre-extend* スクリプトが正常に実行されたというメッセージが表示されたら、以下の情報を指定するように要求されます。

| フィールド | ヒント |
|---|---|
| *Mount Point* | ターゲットサーバ上にあるファイルシステムマウントポイントを入力してください（ *DataKeeper* リソースに関連する、 *LifeKeeper* が保護するファイルシステムがない場合は、このダイアログは表示されません）。 |
| *Root Tag* | ルートタグを選択するか、入力してください。これは、ターゲットサーバ上にあるファイルシステムリソースインスタンスの一意の名前です（ *DataKeeper* リソースに関連する、 *LifeKeeper* が保護するファイルシステムがない場合は、このダイアログは表示されません）。 |
| *Target Disk or Partition* | 複製するファイルシステムの配置先となる、ターゲットサーバ上のディスクまたはパーティションを選択してください。<br><br>ドロップダウンボックスのディスクまたはパーティションのリストには、以下のものを除いて、使用できるすべてのディスクが表示されます。<br>• すでにマウント済みのもの<br>• スワップディスクまたはスワップパーティション<br>• スワップディスクまたはスワップパーティション<br><br>ドロップダウンリストには、 *root (/)*、*boot (/boot)*、*/proc, floppy*、*cdrom* などの特殊なディスクまたはパーティションも表示されません。 |

| | |
|---|---|
| | **注記:** ターゲットのディスクまたはパーティションは、ソースのディスクまたはパーティション以上のサイズである必要があります。 |
| *DataKeeper Resource Tag* | DataKeeper リソースタグの名前を選択するか、入力してください。 |
| *Bitmap File* | インテントログの記録に使用するビットマップファイルの名前を選択するか入力してください。 *[None]* を選択すると、インテントログは使用されず、すべての再同期が部分的ではなく全体の再同期になります。 |
| *Replication Path* | ターゲットサーバとクラスタ内の他の指定サーバとの間で複製に使用する、ローカルとリモートの IP アドレスのペアを選択してください。有効なパスおよび対応する IP アドレスは、このサーバのペアに対して指定した *LifeKeeper* コミュニケーションパスのセットから得られます。 *DataKeeper* の特性により、プライベート（専用）ネットワークを使用することが強く推奨されます。<br><br>*DataKeeper* リソースをすでに 1 台以上のターゲットサーバに拡張している場合、追加のサーバに対する拡張を実行すると、新しいターゲットサーバと既存のサーバとの組み合わせのそれぞれについて、繰り返し複製パスを指定するように要求されます。 |
| *Replication Type* | 指定したサーバのペアについて使用する複製タイプとして、 *[synchronous]* または *[asynchronous]* を選択してください。<br><br>前述の *[Replication Path]* フィールドと同様に、 *DataKeeper* リソースをすでに 1 台以上のターゲットサーバに拡張している場合、追加のサーバに対する拡張を実行すると、新しいターゲットサーバと既存のサーバとの組み合わせのそれぞれについて、繰り返し複製タイプを指定するように要求されます。 |

2. ***[Extend]*** をクリックして次に進んでください。拡張が実行中であることを確認する情報ボックスが表示されます。

3. ***[Finish]*** をクリックして、 *DataKeeper* リソースインスタンスが正常に拡張されたことを確認してください。

4. ***[Done]*** をクリックして、 ***[Extend Resources Hierarchy]*** メニューを終了してください。

**注記:** 必ず すべての サーバでスイッチオーバを手動実行して、新しいインスタンスの機能をテストしてください。詳細については、リソース階層のテスト を参照してください。この時点で、 *SIOS DataKeeper* がソースからターゲットのディスクまたはパーティションにデータの再同期を開始しています。 *LifeKeeper* の *GUI* では、ターゲットサーバにあ

> る *DataKeeper* リソースのステータスは「 *Resyncing* 」になります。再同期が完了すると、ステータスは「 *Target* 」になります。これは通常のスタンバイ状態です。

再同期中、 *DataKeeper* リソース、およびそれに依存するリソースはフェイルオーバできません。これは、データの破損を防止するためです。

# 5.5.3.9.3. リソース階層の拡張解除

LifeKeeper クラスタ内にある 1 台のサーバからリソース階層を削除するには、次の手順を実行します。

1. **[Edit]** メニューの **[Resource]** から **[Unextend Resource Hierarchy]** を選択してください。

2. DataKeeper リソースを拡張解除する **ターゲットサーバ** を選択してください。DataKeeper リソースが現在 In Service （アクティブ）のサーバは選択できません。

   **注記:** 右側のペインから個々のリソースインスタンスを右クリックして **[Unextend]** 作業を選択した場合、このダイアログボックスは表示されません。

   **[Next]** をクリックしてください。

3. **拡張解除する DataKeeper 階層** を選択し、 **[Next]** をクリックしてください（このダイアログは、いずれかのペインでリソースインスタンスを右クリックして、 **[Unextend]** を選択した場合には表示されません）。

4. 選択したターゲットサーバと DataKeeper リソース階層の拡張解除を確認する情報ボックスが表示されます。 **[Unextend]** をクリックしてください。

5. DataKeeper リソースが正常に拡張解除されたことを確認する別の情報ボックスが表示されます。 **[Done]** をクリックして、 **[Unextend Resource Hierarchy]** メニューを終了してください。

   **注記:** これで、データがバックアップサーバにレプリケーションされなくなります。

# 5.5.3.9.4. リソース階層の削除

LifeKeeper 構成内の すべての サーバから DataKeeper リソースを削除するには、次の手順を実行してください。

> 注記: DataKeeper リソースは、削除する前に Out of Service にすることが推奨されます。 Out of Service にしない場合、 **md** と **NetRAID** のデバイスが削除されず、ファイルシステムを手動でアンマウントする必要があります。DataKeeper リソースを Out of Service にする を参照してください。

1. **[Edit]** メニューの **[Resource]** から **[Delete Resource Hierarchy]** を選択してください。
2. 削除する DataKeeper リソース階層が存在する **ターゲットサーバ** の名前を選択してください。
   **注記:** 左側ペインのグローバルリソースまたは右側ペインの個々のリソースインスタンスを右クリックして [Delete Resource] 作業を選択した場合、このダイアログボックスは表示されません。
3. **[Hierarchy to Delete]** を選択してください（左右のペインのリソースインスタンスを右クリックして **[Delete Resource]** 作業を選択した場合、このダイアログは表示されません） **[Next]** をクリックしてください。
4. 選択したターゲットサーバと、削除の対象として選択した階層を確認する情報ボックスが表示されます。 **[Delete]** をクリックしてください。
5. DataKeeper リソースが正常に削除されたことを確認する別の情報ボックスが表示されます。
   **[Done]** をクリックして終了してください。

> 注記: リソースを削除する前にマウントされた状態の NetRAID デバイスは、マウントされたまま残ります。それ以外の NetRAID デバイスは削除されます。

# 5.5.3.9.5. DataKeeper リソースを Out of Service にする

DataKeeper リソースを Out of Service にすると、LifeKeeper によるリソースの保護が解除されます。ミラーが解除され、ファイルシステムがアンマウントされます（該当する場合）。 **md** デバイスが停止し、**nbd** サーバとクライアントが強制終了されます。

**警告:** データのミラーリングを停止して LifeKeeper の保護を解除する場合以外は、DataKeeper リソースを Out of Service にしないでください。 **一時停止** の操作を使用して、ミラーリングを一時停止してください。

1. LifeKeeper の GUI の右側ペインにある、 In Service の **DataKeeper リソース** を右クリックしてください。
2. リソースのポップアップメニューの **[Out of Service]** をクリックしてください。
3. 選択したリソースが Out of Service になることを示すダイアログボックスが表示されます。この操作に関連するリソースの依存関係がダイアログに表示されます。 **[Next]** をクリックしてください。
4. 情報ボックスに、 Out of Service にするリソースの結果が表示されます。 **[Done]** をクリックしてください。

# 5.5.3.9.6. DataKeeper リソースを In Service にする

DataKeeper リソースを In Service にする操作は、リソースの作成と似ています。LifeKeeper は **nbd** サーバとクライアントを起動し、ソースとターゲットのデバイス間でデータを同期する **md** デバイスを起動して、ファイルシステムをマウントします（該当する場合）。

1. 右側のペインにある **DataKeeper リソースインスタンス** を右クリックしてください。

2. ポップアップメニューの **[In Service]** をクリックしてください。選択したサーバとリソースを In Service にすることを確認するダイアログボックスが表示されます。 **[In Service]** をクリックしてリソースを In Service にしてください。

3. 情報ボックスに、 In Service にするリソースの結果が表示されます。この操作に関連するリソースの依存関係が確認ダイアログに表示されます。 **[Done]** をクリックしてください。

# 5.5.3.9.7. リソース階層のテスト

手動スイッチオーバを開始することによって、DataKeeper リソース階層をテストできます。このテストは、プライマリサーバからバックアップサーバへのリソースインスタンスのフェイルオーバをシミュレートします。

## LifeKeeper の GUI からの手動スイッチオーバの実行

手動スイッチオーバを開始するには、LifeKeeper の GUI で **[Edit] > [Resource] > [In Service]** を選択します。例えば、バックアップサーバで In Service リクエストが実行されると、DataKeeper リソース階層がバックアップサーバ側で In Service になり、プライマリサーバ側では Out of Service になります。この時点で、元のバックアップサーバがプライマリサーバに、元のプライマリサーバがバックアップサーバになります。

新しいプライマリサーバーの DataKeeper リソースの状態は、LifeKeeper GUI で [Source] に設定されています。スイッチオーバー中は、各ターゲットの DataKeeper リソースの状態は最初は [Out of Sync] に設定され、データがまだそのターゲットに複製されていないことを示します。マルチターゲット構成の場合、前のソースには[Out of Sync]、その他のターゲットには [Out of Sync (Wait for Previous Source)] と表示されます。 再同期は、前のソースから各ターゲットの順に、自動的に開始されます（状態は [Resyncing] に移行します）。再同期が完了すると、状態は [Target] に変わります。これは通常の スタンバイ 状態です。これらの状態通常素早く変化するため、GUI には表示されない場合があります。

**注記:** 再同期中は、DataKeeper リソースの手動フェイルオーバはできません。

**[Out of Service]** 要求を実行した場合、リソース階層は他のサーバで In Service にならずに、Out of Service になります。リソースを同じサーバ上で In Service に戻すことができるのは、再同期中にリソースが Out of Service になった場合のみです。

# 5.5.3.10. DataKeeper イベントテーブル

次の表は、DataKeeper 固有のイベントと関連するトラップ番号のリストです。 LK_NOTIFY_ALIAS が設定されている場合、これらのイベントはメール通知を生成します。

| LifeKeeper イベント / 説明 | トラップ # | オブジェクト ID |
|---|---|---|
| **DataKeeper Target Disk Mounted**<br><br>DataKeeper ミラーのターゲットディスクが直接マウントされている場合、稼働中のノードから送信されます。 | 142 | .1.3.6.1.4.1.7359.1.0.142 |
| **DataKeeper Resynchronization Complete**<br><br>DataKeeper ミラーがターゲットと再同期したときに、稼働中のノードから送信されます。 | 143 | .1.3.6.1.4.1.7359.1.0.143 |

# 5.5.4. 管理

## SIOS DataKeeper for Linux　の管理

以下のトピックには、リソースを作成した後の SIOS DataKeeper for Linux の動作と問題を理解し、管理するのに役立つ情報があります。

---


- ミラーステータスの表示
- GUI からのミラーの管理
  - ミラーを強制的にオンラインにする
  - 一時停止と再開
  - 圧縮レベルの設定
- コマンドラインからのミラー管理
- コマンドラインからのミラーステータスの監視
- サーバの障害
- 再同期
- 全同期の回避

# 5.5.4.1. ミラーのステータスの表示

**[Replication Status]** ダイアログには、ミラーに関する以下の情報が表示されます。

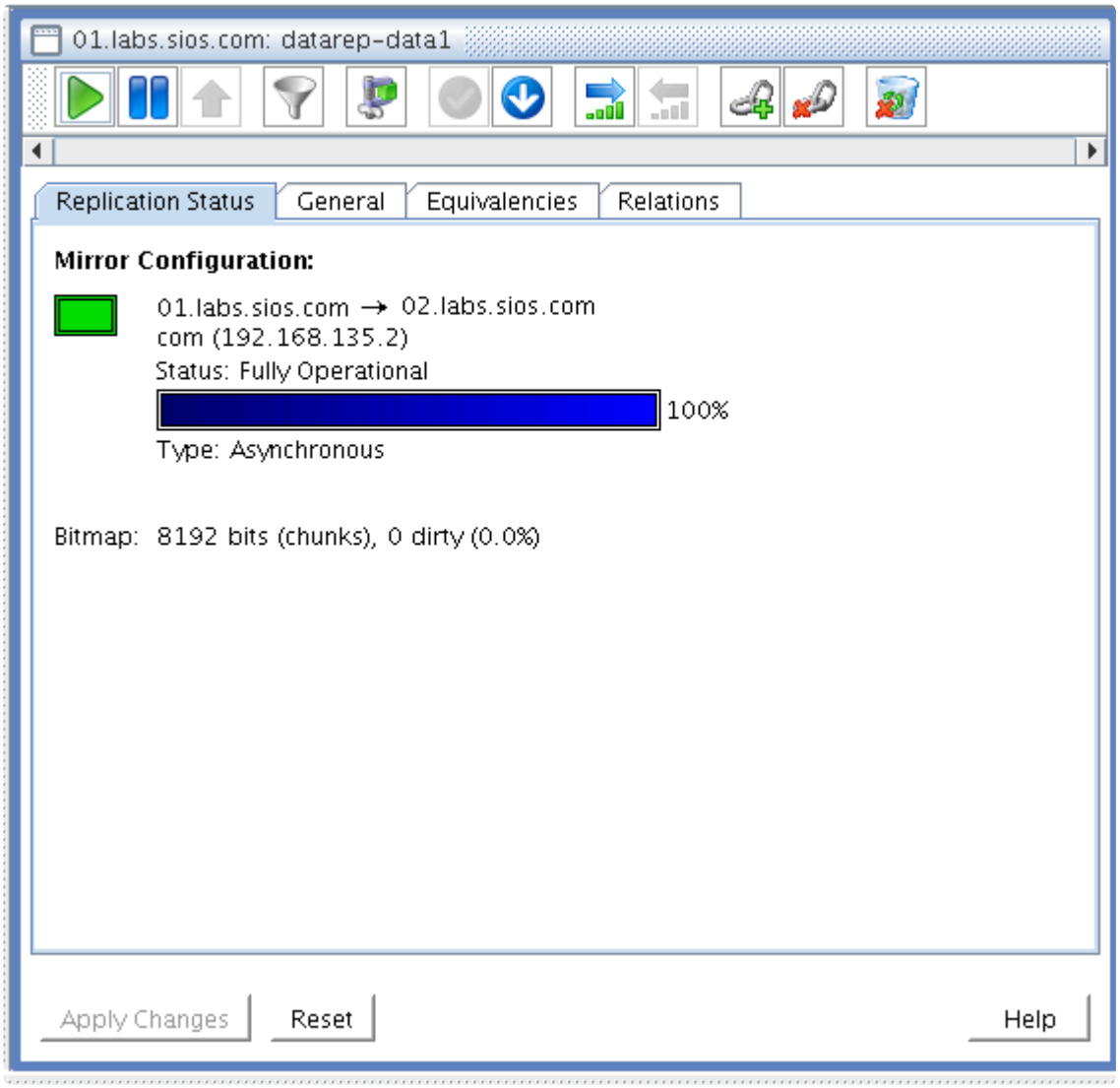
- **Mirror status**:Fully Operational（フルに動作可能）、Paused（一時停止）、Resyncing（再同期中）、またはOut Of Sync（同期停止）
- **Synchronization status**: 同期が完了した割合
- **Replication type**: synchronous（同期）または asynchronous（非同期）
- **Replication direction**: ソースサーバからターゲットサーバに
- **Bitmap**: ビットマップ / インテントログの状態
- **Network Compression Level**: 圧縮レベル（有効の場合）

**[Replication Status]** ダイアログを表示するには、次の手順に従います。

1. **[View]** メニューをクリックし、 **[Properties Panel]** を選択します。
2. **[LifeKeeper status]** 表示にある **DataKeeper** リソース をクリックします。

   または

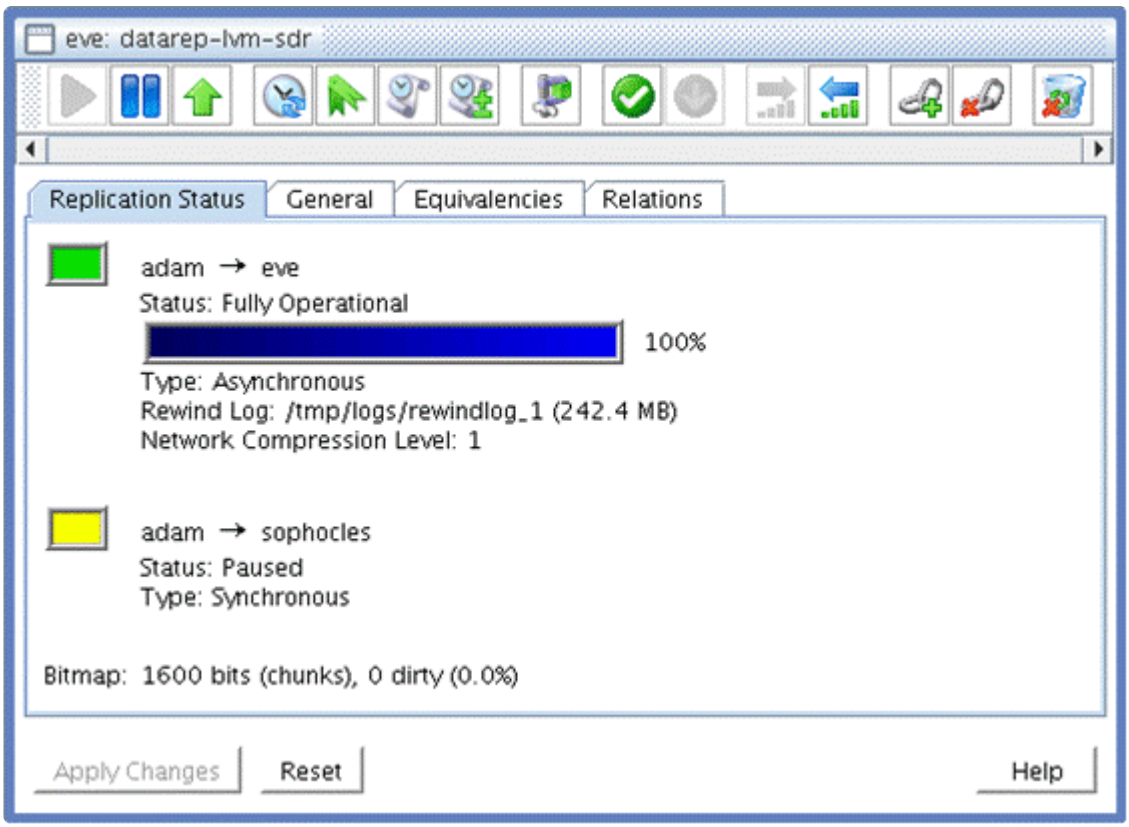
1. **[LifeKeeper status]** 表示にある **DataKeeper** リソース を右クリックします。
2. ポップアップメニューから **[Properties]** を選択します。


01.labs.sios.com: datarep-data1
Replication Status
General
Equivalencies
Relations
Mirror Configuration:
01.labs.sios.com → 02.labs.sios.com
com (192.168.135.2)
Status: Fully Operational
100%
Type: Asynchronous
Bitmap: 8192 bits (chunks), 0 dirty (0.0%)
Apply Changes
Reset
Help

# 5.5.4.2. GUI からのミラーの管理

SIOS DataKeeper のミラーは、LifeKeeper の GUI から以下の 2 とおりの方法で実行できます。

1. **[Properties Panel]** を有効にし、ツールバーのアイコン（スクリーンショットを参照）をクリックします。




説明を表示するには、それぞれのアイコンをクリックしてください。

または

2. **data replication** リソース を右クリックし、ポップアップメニューから動作を選択します。

# ミラーを強制的にオンラインにする

**[Force Mirror Online]** は、両方のサーバが動作不能になり、かつプライマリサーバの再起動後にリソースを In Service にできない場合にのみ使用してください。 **[Force Mirror Online]** を選択すると、*data_corrupt* フラグが削除され、DataKeeper リソースが In Service になります。詳細については、トラブルシューティング セクションの「プライマリサーバがリソースを ISP にできない」を参照してください。

**注記:** mirror_settings はターゲットシステム（ミラーのソースになるシステムには無関係に設定を有効にする場合はすべてのシステム）で実行する必要があります。設定の変更内容を有効にするには、ミラーを **一時停止** してから **再起動** する必要があります。

# 5.5.4.2.1. 一時停止と再開

## ミラーの一時停止

## ミラーの再開

ミラーを一時停止して、すべての書き込みをターゲットディスクに複製する動作を一時的に停止できます。例えば、ミラーを一時停止して、ターゲットディスクのスナップショットを収集することも、トラフィックのピーク時にソースシステムの I/O パフォーマンスを向上させることもできます。

ミラーを一時停止すると、ミラーは、ターゲットシステムの通常のファイルシステムのマウントポイントに読み取りアクセスするように（カーネル 2.6.19 以降は読み取り / 書き込みアクセス）マウントされます。ミラーの一時停止中にターゲットに書き込まれたデータはすべて、ミラーの再開時に上書きされます。

# 5.5.4.2.2. 圧縮レベルの設定

ネットワークの圧縮レベルは、0 ～ 9 の値に設定できます。値 0 は、圧縮を無効にします。レベル 1 は最も高速で圧縮率が最も低いレベルです。一方、レベル 9 は最も低速ですが圧縮率が最高です。ネットワーク圧縮は通常、WAN 環境で利用することによって効果を発揮します。

# 5.5.4.3. コマンドラインからのミラー管理

ミラーの管理は、LifeKeeper の GUI からの操作だけでなく、コマンドラインからも実行できます。DataKeeper のリソースの管理に使用できるコマンドがいくつかあります（*$LKROOT/bin* ディレクトリを参照）。

## ミラーの操作

```
mirror_action <tag> <action> [source] [target(s)]
```

**<tag>** DataKeeper リソースを表す LifeKeeper リソースタグ

**<action>** 以下のいずれか:pause, resume, force, fullresync

**[source]** ( オプション ) 現在のソースシステム (ソースが指定されない場合、コマンドが実行されたカレントシステムを使用)

**[target]** ( オプション ) 操作対象のターゲットシステム (またはシステムのリスト。ターゲットが指定されない場合、該当するすべてのターゲットを使用)

注記: force 操作を使用する場合、ソースノードを指定するために source 引数が必要ですが、target(s) 引数は必要ありません。pause, resume, fullresync のいずれかの操作を使用するときに target(s) 引数を指定する場合は、source 引数も必要です。

例:

*datarep-ext3* という名前のミラーを一時停止する。

```
mirror_action datarep-ext3 pause
```

*adam* から eve と *sophocles* の両方のシステムへの複製を再開する。

```
mirror_action datarep-ext3 resume adam eve sophocles
```

システム *eve* へのオンラインミラーリングを強制実行する。

```
mirror_action datarep-ext3 force eve
```

*adam* から *sophocles* への複製を再開し、これらのシステム間で全体の再同期を強制実行する。

```
mirror_action datarep-ext3 fullresync adam sophocles
```

## ミラーのサイズ変更

mirror_resize コマンドは、リソースの再作成をせずにミラーのサイズを変更することができま

す。ミラーのサイズを変更する前に、ソースシステムおよびターゲットシステムの両方でベースとなるデバイスのサイズを変更する必要があります。その後、ソースシステム上でこのコマンドを実行してください。ベースとなるデバイスのサイズが自動的に検出され、新しいミラーのサイズとして使用されます。また、オプションでサイズを指定することもできます。

LifeKeeper v9.5.0 以降では、リソースが In-Service でもミラーのサイズを変更できます。 ただし、ミラーサイズを縮小する場合は、リソースを Out-of-Service する必要があります。 一部のファイルシステムは、ミラーサイズの縮小をサポートしていないことに注意してください。

! ミラーがオンラインのときにサイズ変更を実行すると、 **full resync** が必要になります。ミラーが停止しているときにサイズを変更すると、拡張した領域のデータのみが同期されます。

✻ **注記:** エラー等でコマンドが途中で終了した場合は、”-f” オプションを付けて再度実行してください。正しく終了しないと、データ不整合が発生する可能性があります。

```
mirror_resize [-f] [-s <size>] <tag>
```

`<tag>` ミラーリソースのタグ。

`-f` ユーザに確認せずにサイズ変更を強制実行（推奨しません）。[-f] オプションを追加すると、 **full-resync** も強制的に実行 されます。

`-s <size>` ミラーサイズ (KB 単位) を指定。

## ミラーのサイズコマンドの利用条件

- ベースとなるデバイスは論理ボリューム (LV) であること
- 単一のターゲットであること（複数ターゲット構成は非サポート）

✻ **注記:** `mirror_resize` は、マルチターゲット構成ではサポートされていません。

## EXT ファイルシステムを利用したミラーのサイズ変更の推奨手順

1. 基礎となるデバイスのリサイズを実行します。ソースとターゲットの両方に対して実行します。(ターゲットサイズはソースサイズ以上である必要があることに注意してください)。 **注記:** これは、階層が In-Service かつ、アクティブであるときに、同期に影響を与えることなく実行できます。
2. 1 つのサーバーで `mirror_resize` を実行します。 データレプリケーションリソースが In-Service の場合、稼働中のサーバー (ソース) で `mirror_resize` を実行します。 **ミラーがオンラインのときにサイズ変更を実行すると、 full resync が必要になります** 。 これにより、ミラーの内部メタデータとビットマップが更新され、新しく拡張されたディスクサイズが反映されます。

**例:** `mirror_resize -s <size in KB> <tag>`

3. リソースが稼働していない場合は、ステップ 2 で `mirror_resize` を実行したサーバーで、ミラー (datarep) リソースのみを稼働させます。新しく拡張されたデバイスの再同期 発生します。 **注記:** “wait to resync” が有効になっている場合は、`/opt/LifeKeeper/bin/mirr_action resume` を実行して、部分的な再同期をただちに強制実行します。

4. ファイルシステムのサイズ変更は、ファイルシステムが In-Service (マウント) または Out-of-Service (アンマウント) の状態で実行できます。 ファイルシステムをマウントした状態でサイズ変更を行う場合は、ファイルシステムリソースを In-Service にします。 **注記:** ファイルシステムリソースが In-Service になっていない場合は、fsck が必要になる場合があります。

5. ミラーデバイス上でファイルシステムのリサイズを実行します。 (例: resize2fs /dev/mdX ここで X は /dev/md0 のようにリサイズされるミラーの md デバイス番号)

6. 階層が完全に In-Service でない場合は、 In-Service 化することができます。ターゲットへの切り替えを許可する前に、すべての再同期を完了する必要があります。

## XFS ファイルシステムを使用したミラーサイズ変更の推奨手順

1. 基礎となるデバイスのリサイズを実行します。ソースとターゲットの両方に対して実行します。(ターゲットサイズはソースサイズ以上である必要があることに注意してください)。 **注記:** これは、階層が In-Service かつ、アクティブであるときに、同期に影響を与えることなく実行できます。

2. 1 つのサーバーで `mirror_resize` を実行します。 データレプリケーションリソースが In-Service の場合、稼働中のサーバー (ソース) で `mirror_resize` を実行します。 **ミラーがオンラインのときにサイズ変更を実行すると、 full resync が必要になります** 。 これにより、ミラーの内部メタデータとビットマップが更新され、新しく拡張されたディスクサイズが反映されます。

   **例:** `mirror_resize -s <size in KB> <tag>`

3. リソースが稼働していない場合は、ステップ 2 で `mirror_resize` を実行したサーバーで、ミラー (datarep) リソースのみを稼働させます。新しく拡張されたデバイスの再同期 発生します。 **注記:** “wait to resync” が有効になっている場合は、`/opt/LifeKeeper/bin/mirr_action resume` を実行して、部分的な再同期をただちに強制実行します。

4. ファイルシステムリソースが In-Service でない場合は、すぐに In-Service にします。 XFS ファイルシステムのサイズ変更は、サービス中の (マウントされた) ファイルシステムで実行する必要があります。

5. ファイル システムのサイズ変更を実行します。 (例: xfs_growfs -d /path_to_file_system)

6. 階層が完全に In-Service でない場合は、 In-Service 化することができます。ターゲットへの切り替えを許可する前に、すべての再同期を完了する必要があります。

# ビットマップの管理

```
bitmap -a <num>|-c|-d|-x <size_kb>|-X <bitmap_file>
```

-a <num>ビットマップファイルに非同期書き込みのパラメータを追加します。同期ミラーのアップグレードにより非同期ターゲットを含むようになった場合、これは必須です。<num> のデフォルト値は 256 です。この制限に最適な値を計算するには、 **SIOS DataKeeper for Linux** によるミラーリング の非同期ミラーリング情報 を参照してください。

-c ビットマップファイルをクリーニングします (すべてのビットを *0* に設定)。ソースディスクの余分な複製がターゲットに存在する場合、これを使用することで全体の再同期を回避できます。、 **このオプションは、特に注意して使用してください** 。

-d ビットマップファイルをダーティに設定します (すべてのビットを *1* に設定)。例えばスプリットブレインの状況が発生した後などに、このオプションを使用して全体の再同期を強制実行できます。

-m ビットマップを読み取り、マージストリームを作成します。

-X <bitmap file> ビットマップファイルを調べて、ビットマップとミラーに関する有用な情報を表示します。

-x <size_kb> size_kb. のディスクが有効になるようにビットマップファイルを拡張します。
(**注記** : このオプションはミラーのサイズ変更のために内部的にのみ使用されます。)

さらに、mdadm コマンドを使用して、DataKeeper のリソースを管理できます。これは、DataKeeper のリソースが実際には md デバイスに存在するからです。詳細については、mdadm(8) のマニュアルページを参照してください。

**注記** : mdadm を使用するときには、OS 同梱の mdadm ではなく *$LKROOT/bin/mdadm* を必ず使用してください。

# 5.5.4.4. コマンドラインからのミラーステータスの監視

通常、ミラーステータスは、LifeKeeper の GUI から、**[Resource Properties]** ダイアログの **[Replication Status]** を使用して確認できます。ただし、以下の操作でもミラーのステータスを監視できます。

```
$LKROOT/bin/mirror_status <tag>
```

### 例:

```
# mirror_status datarep-ext3-sdr
[-]    eve -> adam
   Status: Paused
   Type: Asynchronous
[-]    eve -> sophocles
   Status: Resynchronizing
   [=>              ] 11%
   Resync Speed: 1573K/sec
   Type: Synchronous
Bitmap: 4895 bits (chunks), 4895 dirty (100.0%)
```

以下のコマンドも役に立つことがあります。

```
cat /proc/mdstat
```

サンプルの ***mdstat*** ファイルを示します。

```
eve:~ # cat /proc/mdstat
Personalities : [raid1]
md1 : active raid1 nbd10[1] nbd8[3](F) sdb1[0]
   313236 blocks super non-persistent [3/2] [UU_]
   bitmap: 3/3 pages [12KB], 64KB chunk, file: /opt/LifeKeeper/
```

```
        bitmap_ext3-sdr

      unused devices: <none/></tag>
```

# 5.5.4.5. サーバの障害

プライマリサーバとバックアップサーバの両方が動作不能になった場合、DataKeeper リソースは、 **両方**のサーバが再び動作可能になった場合にのみ In Service / アクティブになります。これは、間違った方向への再同期に起因するデータの破損を防ぐためです。動作可能なサーバが、リソースが「**In Service Protected**」( **ISP** ) である最後のサーバであることが確実に分かっている場合は、DataKeeper リソースを右クリックし、 **[Force Mirror Online]** を選択してそのリソースを強制的にオンラインにすることができます。

# 5.5.4.6. 再同期

DataKeeper リソースの再同期中、ターゲットサーバにあるこのリソースインスタンスのステータスは「**Resyncing**」（再同期中）になります。ただし、リソースインスタンスは、プライマリサーバの「**ソース**」（**ISP**）です。LifeKeeper の GUI は、ターゲットサーバにある DataKeeper のリソースのステータスを以下のアイコンで表示します。

プライマリサーバにある DataKeeper のリソースは、以下のアイコンで表示されます。

再同期が完了するとすぐに、ターゲットのリソースの状態が「**ターゲット**」になり、アイコンが以下のように変化します。

全同期は、ディスクパーティション全体を同期させます

再同期プロセスについて、以下の点に注意してください。

- SIOS DataKeeper のリソースとその親リソースは、プライマリでの障害発生時に再同期プロセス中のターゲットにはフェイルオーバできません。
- ターゲットサーバの同期中に DataKeeper リソースが out of service / 非アクティブになった場合、そのリソースは、同じシステム、またはすでに同期済みの別のターゲット（複数のターゲットが存在する場合）でのみ In Service / アクティブにすることができ、再同期が続行されます。
- 再同期プロセス中にプライマリサーバが動作不能になった場合、同期プロセス中のターゲットサーバはすべて、DataKeeper リソースを In Service にすることができません。プライマリサーバが再び動作可能になった後に、ミラーの再同期が続行されます。

# 5.5.4.7. 全同期の回避

大量のデータを WAN リンク経由で複製する場合、膨大なネットワーク帯域幅と時間を消費する可能性がある全同期は避けることが望ましいです。新しいカーネルと共に使用する場合、SIOS DataKeeper はビットマップテクノロジを使用して全同期をほぼ防ぐことができます。ただし、既存のデータを複製する場合、ミラーの初期設定時に発生する最初の全同期を回避することはできません（新規データの場合には SIOS DataKeeper は全同期を実行しないので以降の手順は不要です）。

> ✻ 全同期は、ディスクパーティション全体を同期させます

既存のデータを複製するときに、全同期を回避する方法がいくつかあります。ここでは推奨する2とおりの方法を説明します。

## 方法 1 – 2番目のノードへの複製

1 番目の方法では、RAW ディスクイメージを取得してターゲットサイトに輸送します。データがターゲットシステムに到着するまで、ソースシステムのミラーをアクティブにしておくことができるので、この方法ではダウンタイムが最小になります。

### 手順

1. ミラーを作成してください（[Replicate Existing Filesystem] を選択）。ただし、ターゲットシステムにミラーを拡張しないでください。

2. ミラーを out of service にしてください。

3. ソースディスクまたはパーティションのイメージを取得してください。この例では、選択したディスクまたはパーティションは /dev/sda1 です
   ```
   root@source# dd if=/dev/sda1 of=/tmp/sdr_disk.img bs=65536
   ```
   （ブロックサイズの引数65536は単に効率的にするためです。）

   ディスクまたはパーティションの RAW ディスクイメージを持つファイルが作成されます。
   ファイルの代わりに、ハードドライブやその他の記憶デバイスも使用できます。

4. オプション手順 - ソースディスクまたはパーティションのチェックサムを取得してください。
   ```
   root@source# md5sum /dev/sda1
   ```

5. オプション手順 - ディスクイメージファイルを圧縮してください。
   ```
   root@source# gzip /tmp/sdr_disk.img
   ```

6. ビットマップファイルをクリアしてください。
   （最後の引数はリソース作成時に指定したビットマップファイルのパスに置き換えてください。）
   ```
   root@source# /opt/LifeKeeper/bin/bitmap -c /opt/LifeKeeper/bitmap__dr
   ```

7. ミラーと依存ファイルシステム、およびアプリケーション（存在する場合）のサービスを開始して

ください。ビットマップファイルにより、データがターゲットシステムに転送される間に発生した変更内容が追跡されます。

8. 好みの転送方法を使用して、ターゲットシステムにディスクイメージを転送してください。
9. オプション手順 - ターゲットシステムでディスクイメージファイルを圧縮解除してください。
   ```
   root@target# gunzip /tmp/sdr_disk.img.gz
   ```
10. オプション手順 – イメージファイルのチェックサムが、手順 4 で取得した元のチェックサムと一致することを確認してください。
    ```
    root@target# md5sum /tmp/sdr_disk.img
    ```
11. イメージをターゲットシステム（例: /dev/sda2）に転送してください。
    ```
    root@target# dd if=/tmp/sdr_disk.img of=/dev/sda2 bs=65536
    ```
12. 両方のシステムで、/etc/default/LifeKeeper に LKDR_NO_FULL_SYNC=1 を設定してください。
    ```
    root@source# echo 'LKDR_NO_FULL_SYNC=1' >> /etc/default/LifeKeeper
    root@target# echo 'LKDR_NO_FULL_SYNC=1' >> /etc/default/LifeKeeper
    ```
13. ミラーをターゲットに拡張してください。部分的な再同期が実行されます。
14. /etc/default/LifeKeeper を編集してLKDR_NO_FULL_SYNC エントリを削除します。

## 全再同期を行わずに 3番目のノードまたは追加のノードに拡張する

### ソースからデータをコピーする方法

これらの手順では、ミラーがすでに作成され、2 番目のノード（ターゲット1）に拡張されていることを前提としています。

1. 各システムの /etc/default/LifeKeeper で、LKDR_NO_FULL_SYNC=1 を設定します。
   ```
   root@source# echo 'LKDR_NO_FULL_SYNC=1' >>/etc/default/LifeKeeper
   root@target1# echo 'LKDR_NO_FULL_SYNC=1' >>/etc/default/LifeKeeper
   root@target2# echo 'LKDR_NO_FULL_SYNC=1' >>/etc/default/LifeKeeper
   ```
2. 新しいターゲット（ターゲット2）にミラーを拡張します。部分的な再同期が実行されます。
3. 新しいターゲット（ターゲット2）へのミラーを一時停止します。
4. ミラーをサービス停止にします。
5. ターゲット2で `umount <filesystem>` を実行して、ターゲット2 の一時停止したミラーのファイルシステムをアンマウントします。
6. ターゲット2で `mdadm --stop /dev/md#` を実行して、ターゲット2で実行されている md デバイスを停止します。 # は、 */proc/mdstat* で報告された値です。
7. ソースノードでソースディスクまたはパーティションのコピーを作成します。これは、dd またはデ

ィスクベンダーやクラウドベンダーのツールを使用して実行できます。コピーはファイルレベルのコピーではなく、**ブロックごとに同一のコピーでなければなりません**。

8. オプション手順 - チェックサムデータを収集してディスクイメージを検証します（md5sum、sha256sum など）。

9. オプション手順 - ディスクイメージを圧縮します。

10. ミラーとそれに依存するファイルシステムおよびアプリケーション（ある場合）を In-service にします。ビットマップファイルは、データがターゲットシステムに転送されている間に加えられた変更を追跡します。

11. ターゲット2 へのミラーがまだ一時停止していることを確認します。一時停止していない場合は、手順 4 で再起動します。

12. ファイルシステムと md デバイスがターゲット2 で実行されていないことを確認します。実行されている場合、ファイルシステムをアンマウントして md デバイスを停止します。

13. ターゲット2 のターゲットディスクにディスクイメージを転送します。

14. ディスクイメージが正しいことを確認します。ディスクの内容を検証するには、md5sum または sha256sum を使用できます。

15. 一時停止したミラーをターゲット2 に対して再開します。ソース上のビットマップは、ターゲット2 が一時停止してから実施された変更を追跡していましたが、ミラーが再開されると、これらの変更はターゲット2 に送信されます。

16. /etc/default/LifeKeeper を編集してLKDR_NO_FULL_SYNC エントリを削除します。

### 一時停止したターゲットからデータをコピーする手順

これによりダウンタイムは発生しなくなりますが、ターゲットが一時停止している間はデータの冗長性がなくなります。

1. この手順では、ミラーがすでに作成され、2 番目のノード（ターゲット1）に拡張されていることを前提としています。

2. 各システムの /etc/default/LifeKeeper で、LKDR_NO_FULL_SYNC=1 を設定します。
```
root@source# echo 'LKDR_NO_FULL_SYNC=1' >>/etc/default/LifeKeeper
root@target1# echo 'LKDR_NO_FULL_SYNC=1' >>/etc/default/LifeKeeper
root@target2# echo 'LKDR_NO_FULL_SYNC=1' >>/etc/default/LifeKeeper
```

3. 新しいターゲット（ターゲット2）にミラーを拡張します。部分的な再同期が実行されます。

4. 新しいターゲット（ターゲット2）へのミラーを一時停止します。

5. ターゲット1 へのミラーを一時停止します。

6. ターゲット2 で `umount <filesystem>` を実行して、ターゲット2 の一時停止したミラーのファイルシステムをアンマウントします。

7. ターゲット2 で `mdadm --stop /dev/md#` を実行して、ターゲット2 で実行されている md デバイスを停止します。 # は、 */proc/mdstat* で報告された値です。

8. ターゲット1 で `umount <filesystem>` を実行して、ターゲット1 の一時停止したミラーのファイルシステムをアンマウントします。

9. ターゲット1 で `mdadm --stop /dev/md#` を実行して、ターゲット1 で実行されている md デバイスを停止します。 # は、 */proc/mdstat* で報告された値です。

10. ターゲット1 のターゲットディスクまたはパーティションのコピーを作成します。これは、dd またはディスクベンダーやクラウドベンダーのツールを使用して実行できます。コピーはファイルレベルのコピーではなく、 **ブロックごとに同一のコピーでなければなりません** 。

11. オプション手順 - チェックサムデータを収集してディスクイメージを検証します（ md5sum、sha256sum など）。

12. オプション手順 - ディスクイメージを圧縮します。

13. ターゲット1 への複製を再開します。ビットマップファイルは、データがターゲット2 に転送されている間に加えられた変更を追跡します。

14. ターゲット2 へのミラーがまだ一時停止していることを確認します。一時停止していない場合は、手順 4 で再起動します。

15. ファイルシステムと md デバイスがターゲット2で実行されていないことを確認します。実行されている場合、ファイルシステムをアンマウントして md デバイスを停止します。

16. ターゲット2 のターゲットディスクにディスクイメージを転送します。

17. オプション手順 - ディスクイメージの圧縮を解除します。

18. オプション手順 - ディスクイメージが正しいことを確認します（md5sum、sha256sum など）。

19. 一時停止したミラーをターゲット2 に対して再開します。ソース上のビットマップは、ターゲット2 が一時停止してから実施された変更を追跡していましたが、ミラーが再開されると、これらの変更はターゲット2 に送信されます。

20. /etc/default/LifeKeeper を編集してLKDR_NO_FULL_SYNC エントリを削除します。

## 方法 2

ターゲットシステムを簡単に輸送できる場合、またはシステムの設定時にターゲットシステムがソースと同じ場所にある場合に、この方法を使用できます。この方法では、最初の全同期を高速なローカルネットワークで実行できるように、最終的な WAN ミラーを作成するネットワークルートを LAN ミラーに一時的

に変更します。以下の例では、ソースサイトはサブネット 10.10.10.0/24 にあり、ターゲットサイトがサブネット10.10.20.0/24 にあると仮定しています。ソースとターゲットのシステムの間に一時的に静的ルートを設定することにより、ローカルのイーサネット接続またはループバックケーブルを使用して「WAN」トラフィックをあるサーバーから別のサーバーに直接送信できます。

## 手順

1. ソースサイトでシステムをインストールし、設定してください。

2. 静的ルートを追加してください。
   ```
   root@source# ip route add 10.10.20.0/24 dev eth0
   root@target# ip route add 10.10.10.0/24 dev eth0
   ```
   この時点で、両方のシステムが LAN 上で相互に通信できる必要があります。

3. LifeKeeper でコミュニケーションパスを設定してください。

4. ミラーを作成し、ターゲットに拡張してください。全同期が実行されます。

5. ミラーを Pause にしてください。ミラーが再開されるまで、変更内容はビットマップファイルで追跡されます。

6. 静的ルートを削除してください。
   ```
   root@source# ip route del 10.10.20.0/24
   root@target# ip route del 10.10.10.0/24
   ```

7. ターゲットシステムをシャットダウンし、恒久的に配置する場所に輸送してください。

8. ターゲットシステムを起動し、ソースとのネットワーク接続を確立してください。

9. Resume Replication を実行してください。部分的な再同期が実行されます。

# 5.5.4.8. 再同期前のデータ検証 (Wait to Resync)

破損データや整合性のとれないデータがターゲットへ複製されるのを防ぐため、LifeKeeper は、データ複製の前に、リソースが In Serviceになるまで待機します。v9.5.2 からはデフォルトでLifeKeeper は、データの複製前にアプリケーションタイプ 'filesys' の親リソースが In Service になるのを待ちます。 これにより、データの複製前に、ファイルシステムの正確性が保証されます。ファイルシステムのマウントに失敗した場合は、適切なリカバリー対応として、データが複製される前にカレントシステムのファイルシステムを修復するか、別のシステムのデータを確認してください。

［Wait to Resync］機能は、 /etc/default/LifeKeeper の “LKDR_WAIT_TO_RESYNC” で設定可能です。
［Wait to Resync］ 機能を設定するにあたり 3 つのオプションがあります。:

1. **False** ［Wait to Resync］ 機能を無効にします。再同期前に親リソースは確認されません。再同期は、他に障害となる問題がなければ DataKeeper リソースの初期のリストア中に開始されます。

2. **<resource type>** 特定のリソースタイプを指定してください。デフォルトでは 'filesys' が指定されています。インストールされている任意のリソースタイプが選択可能です。 '/opt/LifeKeeper/bin/typ_list -f:' コマンドを実行することで、インストールされたリソースタイプの一覧を確認することができます。このコマンドのアウトプットは “アプリケーション:タイプ”の形式で表示されます。以下例です。

```
# typ_list -f:
gen:app
gen:filesys
gen:qsp
gen:hanfs
gen:nfs
scsi:DEVNAME
scsi:device
```

最初の項目はアプリケーションで、 “:” に続く2番目の項目はタイプになります。例えば、上記の [gen:app] の “gen” の部分がアプリケーションで “app” の部分がタイプになります。 “:” の右側はどのタイプでも指定が可能です。LifeKeeper は指定したタイプのDataKeeper リソースの親が In Service になるまで再同期を開始しません。例えば、/etc/default/LifeKeeper でリソースタイプ “LKDR_WAIT_TO_RESYNC=app”が定義されていて、リソース階層が下記のようになっていると仮定します。

```
# lcdstatus -q
LOCAL         TAG             ID                          STATE   PRIO  PRIMARY
ip-10-0-2-128 app1            app1                        OSU       1  ip-10-0-2-128
ip-10-0-2-128  SPSL1           SPSL1                       OSU       1  ip-10-0-2-128
ip-10-0-2-128   /maxdb1        /maxdb1                     OSU       1  ip-10-0-2-128
ip-10-0-2-128    datarep-maxdb1 536681bc-9259-4254 OSU       1  ip-10-0-2-128
ip-10-0-2-128  SPSL2           SPSL2                       OSU       1  ip-10-0-2-128
ip-10-0-2-128   /maxdb2        /maxdb2                     OSU       1  ip-10-0-2-128
ip-10-0-2-128    datarep-maxdb2 952e4ebd-507d-4b43 OSU       1  ip-10-0-2-128
ip-10-0-2-128 app3            app3                        OSU       1  ip-10-0-2-128
ip-10-0-2-128  app2            app2                        OSU       1  ip-10-0-2-128
ip-10-0-2-128   /maxdb1        /maxdb1                     OSU       1  ip-10-0-2-128
ip-10-0-2-128    datarep-maxdb1 536681bc-9259-4254 OSU       1  ip-10-0-2-128
```

app1、app2、app3 がリソースタイプ “app” をもっています。この場合、netraid リソース “datarep-maxdb1” は、3つ全ての gen:app となる親リソースが In Service になるまで同期を実施しません。netraid リソース “datarep-maxdb2”は、 “app1” のみがgen:app となる親なので、 “app1” が In Service になるまでは同期しません。

3. **Hierarchy** 同期前にDataKeeper リソースの の全ての親リソースが In Service になる必要があります。

> **注記:** 特に XFS ファイルシステムにおけるデータの不整合により In Service の処理に失敗することが頻繁にあるため ‘filesys’ タイプがデフォルトになっています。しかしながら、ファイルシステムはデータの不整合や破損があってもマウント可能です。アプリケーションを In Service にするか、階層を In Service にすることを要求すれば、データの高い正確性を保証するでしょう。

ユーザーはこの機能の情報を3つの方法で確認することができます。:

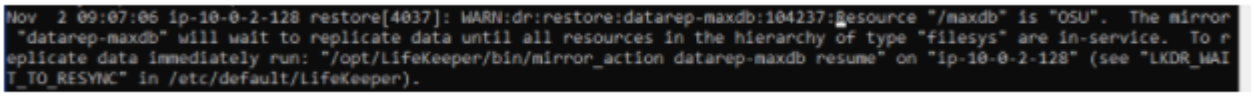
1. 要求された親リソースが Out of Service の場合は、GUI がターゲットを ［Wait to Resync］ と表示します。:

2. 要求された親リソースが Out of Service の場合は、In Service 処理の間に警告メッセージがログに記録されます。このメッセージは、確認中のタイプを指定し、“LKDR_WAIT_TO_RESYNC=hierarchy” となっている場合に、確認中の全ての階層を指定します。

   a. LKDR_WAIT_TO_RESYNC=filesys となっている場合の In Service 処理中に表示されるログメッセージの例です。:

b. LKDR_WAIT_TO_RESYNC=hierarchy となっている場合のログファイルの中のログメッセージです。:

WARN:dr:recover:datarep-maxdb:104237:Mirror “datarep-maxdb” will wait to reconnect targets until parent file system “/maxdb” is in-service. To reconnect targets immediately run: “/opt/LifeKeeper/bin/mirror_action datarep-maxdb resume” on “ip-10-0-2-128” (see “LKDR_WAIT_FOR_FILE_SYSTEM_TO_MOUNT” in /etc/default/LifeKeeper).



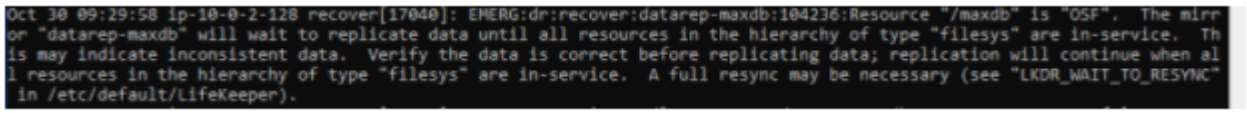

3. 要求された親リソースが 障害ステータス (OSF) の場合は、エマージェンシーメッセージが LifeKeeper ログファイルと全てのオープンしているターミナルに記録されます。

a. この場合、GUI は［Wait to Resync］ステータスを表示します。

b. 以下のメッセージが全てのターミナルと LifeKeeper ログファイルに記録されます。:

recover[26796]: EMERG:dr:recover:datarep-maxdb:104236:Resource “/maxdb” is “OSF”. The mirror “datarep-maxdb” will wait to reconnect targets until parent file system “/maxdb” is in-

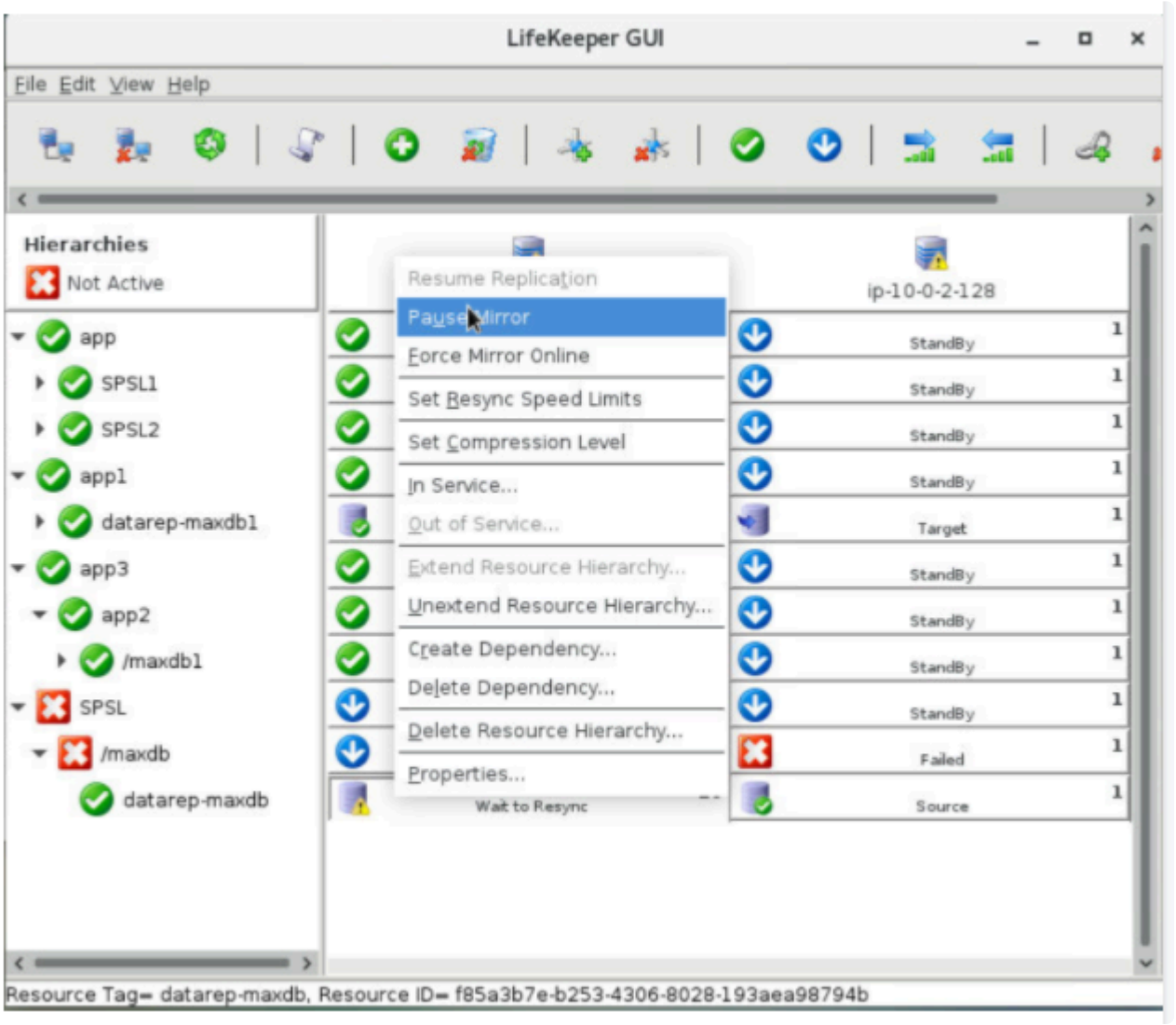
service. This may indicate inconsistent data. Do not bring the resource “/maxdb” in-service until the data has been verified; replication will continue when “/maxdb” is in-service. A full resync may be necessary (see “LKDR_WAIT_FOR_FILE_SYSTEM_TO_MOUNT” in /etc/default/LifeKeeper).



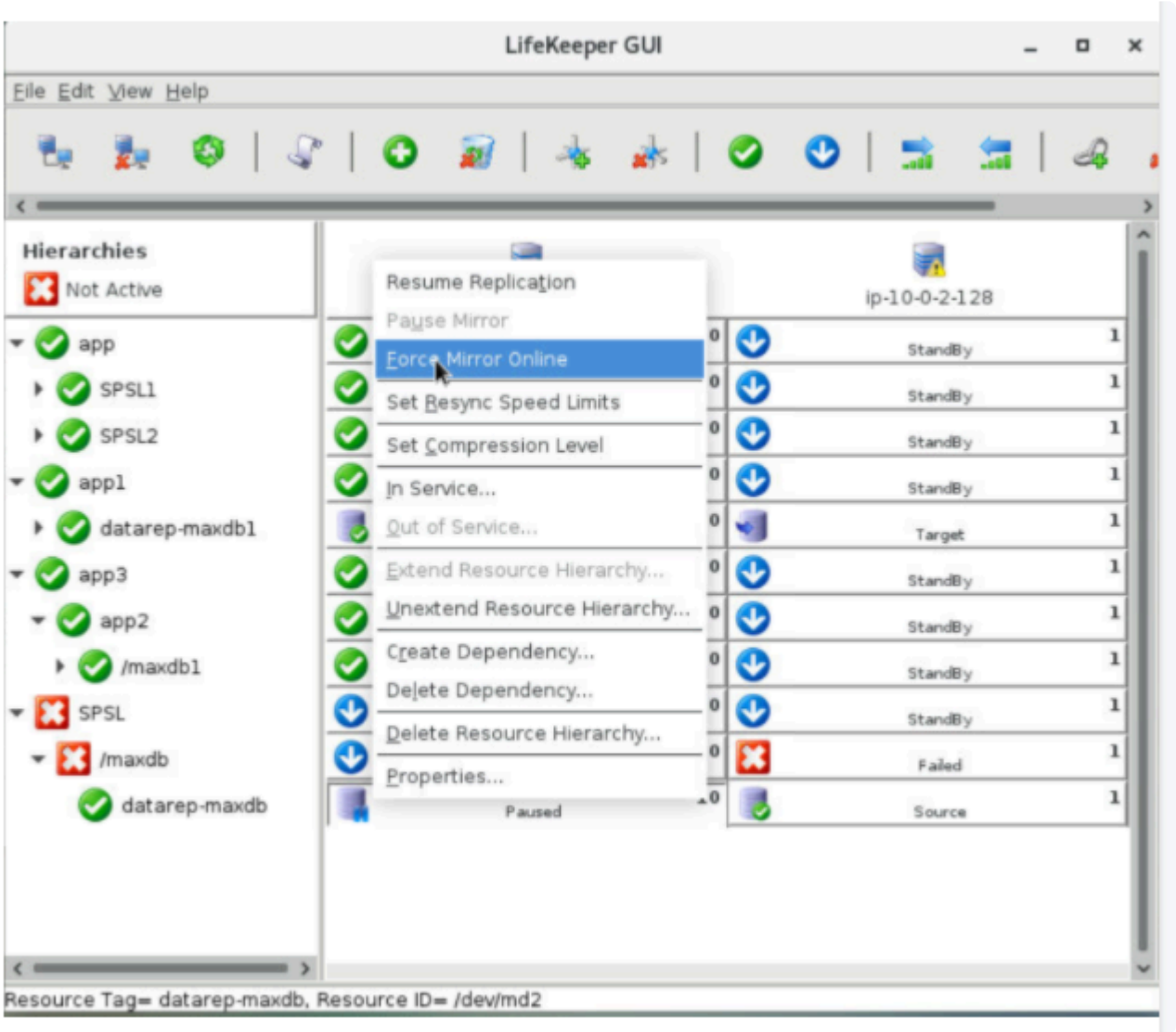

c. ファイルシステムのデータが検証されるまで、ファイルシステムリソースを In Service にしないでください。一旦ファイルシステムリソースが In Service になると、次の DataKeeper リソースに対する quickCheck のサイクル中に複製が再開されます。

d. 障害の発生したファイルシステムリソースは、マウント、ログの再生、fsck がファイルシステムを修復できないことを示します。これは、別のノードでファイルシステムを In Service にすることができた場合、ディスク上のデータ不整合を示している可能性があります。ファイルシステムを障害が発生したノードでリカバリーもしくは別のノードへの切り替えのどちらかの方法にて修復できたら、全てのノードのデータの一貫性を保つため、全同期が必要となります。

e. ファイルシステムがマウントできることおよびデータが検証されたことを確認するため、ミラーがターゲットにマウントされます。 これは [Pause Mirror] の機能を使用して実行可能です。

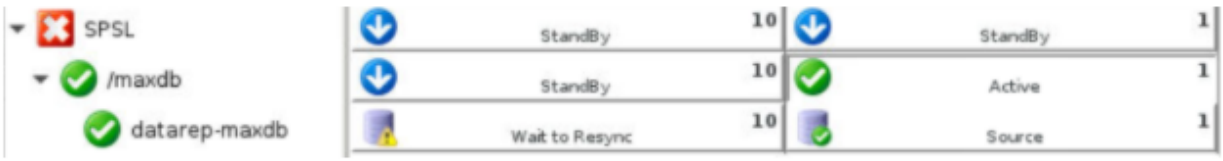



ミラーが一時停止されている際、LifeKeeper は自動的にファイルシステムをターゲットにマウントします。この処理に成功したら、データの整合性を確認してください。ターゲット側(一時停止されているサーバー)で [Force Mirror Online] を選択すると、ミラーを強制的にオンラインにすることができます。 **警告:** この処理は、親リソースが In Service になっていない場合であっても複製を再開します。データが検証済みの場合にのみミラーを強制的にオンラインにしてください。




データに全く問題ないことが確認できるまでは、ターゲット側の［Resume Replication］を選択 しないでください 。ミラーを一時停止した後に［Resume Replication］を選択すると、親リソースが In Service になっていない場合や、障害が発生した場合であっても、複製が開始されます。

f. データが検証され、親リソースの障害も修正され In Service になったら、DataKeeper リソースに対する次の quickCheck サイクルが、これ以上データの再同期を待つ必要がないことを検知します。親リソースが In Service であったとしても、［Wait to Resync］が表示される quickCheck サイクルまでに数分 (デフォルトで最大２分) 要します。




g. データの破損 (例：ファイルシステムを一つのノードにマウントできるが、別のノードにはマウントできないなど) が発見された場合や、破損が疑わしい場合は、全同期を実施することを提案します。全同期を強制的に実施するため、ミラーはまず最初に一時停止される必要があります。ミラーが一時停止されたら、ソースシステムで以下のコマンドを実行してください。:

```
/opt/LifeKeeper/bin/mirror_action <tag> fullresync
```

# 5.5.5. DataKeeper で LVM を使用する

LifeKeeper for Linux は現在 LVM 上での DataKeeper の使用と DataKeeper 上での LVM の使用の両方をサポートしています。標準的な DataKeeper の設定では LVM 上での DataKeeper の使用がサポートされているので、Logical Volume Manager Recovery Kit をインストールしないでください。必要な Application Recovery Kit は、DataKeeper だけです。ただし、DataKeeper 上で LVM を使用する場合は、Logical Volume Manager Recovery Kit が必要になります。

SIOS では LVM 上での DataKeeper 使用を推奨していますが、DataKeeper 上で LVM が使用される場合は、2 段階のリソース階層作成プロセスが必要になります。プライマリーサーバー上で LVM ボリュームグループと論理ボリュームを作成する前に、DataKeeper の Hierarchy Type「Data Replication Resource」でリソース階層を作成します。そして、生成された md デバイスに対してボリュームグループと論理ボリュームを作成します。最後に、保護するアプリケーションに関連した Application Recovery Kit の設定手順に従ってリソース階層を作成します。完成したリソース階層は以下に示す図 3 のようになります。

> **注意:** データの整合性を保つため、DataKeeper 上に LVM を構成する場合は、DataKeeper ミラーが 1 つだけ存在するか、複数の **同期** ミラーが存在する必要があります。

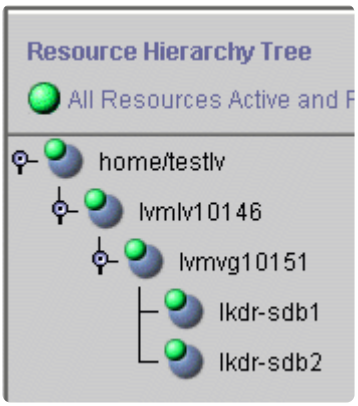



**図 3: DataKeeper 上の LVM を含む階層**

単一のターミナルリーフリソースを各 DataKeeper リソースの子として作成する必要があります。 これにより、階層のサービスが停止するときに単一のリソースが提供され、スイッチオーバー操作中に、別のノードでリソースを起動する前に、階層全体が確実にサービスが停止されます。 詳細については、create_terminal_leaf(1M) のマニュアルページを参照してください。

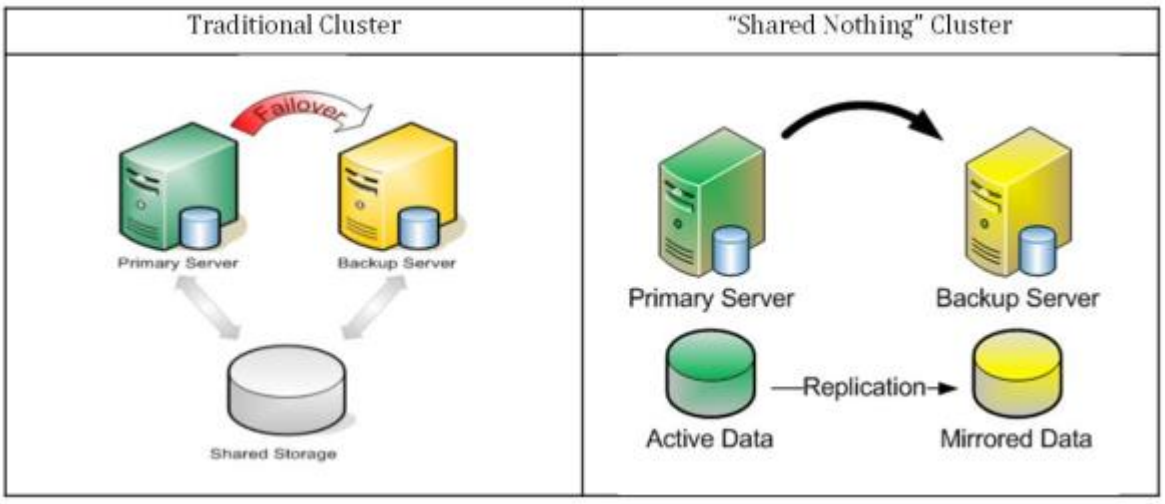
図 3 の階層のターミナルリーフを作成するには、次のコマンドを実行します。

```
$LKROOT/bin/create_terminal_leaf home/testlv
```

# 5.5.6. Fusion-io を使用するクラスター化

## DataKeeper のパフォーマンスを最大に発揮させるための Fusion-io のベストプラクティス

LifeKeeper for Linux はブロックレベルのデータレプリケーション機能を備えており、共有ストレージを使用しないHAクラスタを非常に簡単に設定できます。Fusion-io を使用すると、LifeKeeper for Linux ではフェイルオーバ保護用として「shared nothing」クラスタを構成できます。




クラスター設定の一部としてデータレプリケーションを活用する場合、ディスクへの書き込みと同等な速度でデータがネットワーク上で複製可能なだけの十分な帯域幅があることが重要です。以下のベストプラクティスを使用すると、高速ストレージを使用する場合に「shared nothing」LifeKeeper クラスタ設定を最大に活用できます。

### ネットワーク

- **10 Gbps の NIC を使用する。** Fusion-io のフラッシュベースのストレージデバイス (または OCZ、LSI、などの同様な製品) は、750 MB/ 秒を超えるデータ速度でデータを書き込むことができます。1 Gbps の NIC は理論的な最大値である約 125 MB/ 秒のみを送出できるので、ioDrive の性能を活用すると、1 Gbps のネットワーク接続での複製よりもはるかに高速で簡単にデータを書き込むことができます。リアルタイムのデータレプリケーションを実行するのに十分な帯域幅をサーバー間に確保するには、レプリケーショントラフィックの転送に必ず 10 Gbps の NIC を使用してください。

- **ジャンボフレームを有効にする。** 使用しているネットワークカードとスイッチでサポートされている場合、ジャンボフレームを有効にするとネットワークのスループットが増加するだけでなく、CPU サイクルが減少します。ジャンボフレームを有効にするには、以下の設定を実行してください (RedHat/CentOS/OEL Linux ディストリビューションの例)。

    - 次のコマンドを実行してください。
      ```
      ip link set <interface_name> mtu 9000
      ```

    - 再起動後も変更内容を保持するには、以下のファイルに「MTU=9000」を追加してください。

*/etc/sysconfig/network-scripts/ifcfg-<interface_name>*

- エンドツーエンドのフレーム動作を検証するには、次のコマンドを実行してください。
  `ping -s 8900 -M do <IP-of-other-server>`

- **NIC の転送キュー長を変更する。**
  - 次のコマンドを実行してください。
    `ip link set <interface_name> txqueuelen 10000`
  - 再起動後も変更内容を保持するには、 */etc/rc.local* に追加してください。

- **NIC の netdev_max_backlog を変更する。**
  - 次の設定を */etc/sysctl.conf* に追加してください。

```
net.core.netdev_max_backlog = 100000
```

### TCP/IP の調整

レプリケーションのパフォーマンスを向上することが確認された **TCP/IP の調整** を以下に示します。

- */etc/sysctl.conf* を編集して、以下のパラメータを追加してください ( **注記**: これらは例であり、使用している環境で異なる場合がある)。

```
net.core.rmem_default = 16777216
net.core.wmem_default = 16777216
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
net.ipv4.tcp_rmem = 4096 87380 16777216
net.ipv4.tcp_wmem = 4096 65536 16777216
net.ipv4.tcp_timestamps = 0
net.ipv4.tcp_sack = 0
net.core.optmem_max = 16777216
net.ipv4.tcp_congestion_control=htcp
```

## 設定上の推奨項目

- Fusion-io ドライブ上に、ビットマップファイルを入れる小型の (約 100 MB) のディスクパーティションを割り当ててください。このパーティションにファイルシステムを作成し、マウントしてください (例: */bitmap* )。

```
# mount | grep /bitmap
/dev/fioa1 on /bitmap type ext3 (rw)
```

- ミラーを作成する前に、 */etc/default/LifeKeeper* 内の以下のパラメータを調整してください。

  ```
  LKDR_CHUNK_SIZE=4096
  ```

  デフォルト値は 256 です。

- 通常と同様に、ミラーを作成してクラスタを設定してください。

- ビットマップファイルは、上で作成したパーティションに作成するように設定してください。

- 再同期を高速化するための設定をおこないます。DataKeeper リソースの右メニューから、Set Resync Speed Limits を選択して、ウイザードに以下の値を設定してください。

  - `Minimum Resync Speed Limit: 200000`

- 同時に他の I/O が実行されているときに許可する再同期の速度を指定します。再同期の実行時に通常の I/O 動作が妨げられないようにするには、経験則として、この値をドライブの最大書き込みスループットの半分以下に設定する必要があります。

  - `Maximum Resync Speed Limit: 1500000`

- 再同期に使用する最大帯域幅を指定します。可能な最大速度で再同期が実行されるように、この値を十分高い値に設定する必要があります。

# 5.5.7. DataKeeper for Linux が保護している領域に対する、スナップショット機能の利用についての注意事項

DataKeeperで保護している領域に対して、ディスク領域のスナップショット機能を適用した場合、リストアしたデータを正とした完全同期が必要になります。

本稿で言及するスナップショット機能は、以下のようなものを含みます。

1. AWSなどのクラウド環境が提供しているもの
2. vSphereなどの仮想化ソフトウエアが提供しているもの
3. 物理環境において、ストレージが提供しているもの

これらのスナップショットのリストアは、OSを経由せずに行われます。その場合、DataKeeperはその書き込みをターゲット側に反映することができません。またbitmapにも書き込まれないため、差分同期で整合性を保つことも出来ません。

スナップショットを利用する場合の手順は以下のようになります。

1. Pause mirrorでミラーを静止状態とします
2. snapshotのリストアを行います
3. リストアした正のデータが格納されたディスクをソースとして、ミラーの完全同期を行います

ミラーの完全同期を実施するまでは、ソースとターゲットのデータは整合性のない状態となります。そのためスナップショットの適用後は完全同期の実行が必須となります。これはミラーのサイズによっては非常に長い時間を要する上に、完了するまではリソースを切り替えることができません。実施の際にはご注意ください。

なお、ソース、ターゲットの両方に同じスナップショットを戻し完全同期を回避するといった手法は、データの整合性が担保されないためサポートされません。

# 5.5.8. DataKeeper for Linuxトラブルシューティング

このセクションでは、DataKeeper for Linux の使用時に遭遇する可能性がある問題に関する情報を示します。必要に応じて、エラーの原因およびエラー状態を解消するために必要な処置についても説明しています。

DataKeeper for Linux に固有のメッセージについては、DataKeeper メッセージカタログ を参照してください。他の LifeKeeper コンポーネントからメッセージが送出されることもあります。その場合は、総合メッセージカタログ を参照してください。これらのメッセージカタログには、LifeKeeper for Linux の使用中に遭遇される可能性のあるすべてのエラーコード (操作、管理、および GUI に関するものを含む) の一覧があります。また、エラーコードの原因に関する追加の説明や、問題解決のために必要な処置についても、必要に応じて記載されています。この一覧から、受信したエラーコードを検索できます。また、該当する LifeKeeper コンポーネントの個別のメッセージカタログに直接アクセスすることもできます。

以下の表に、予測される問題と推奨される処置を示します。

| 症状 | 推奨される処置 |
|---|---|
| GUI に［Wait to Resync］が表示される。 | 再同期処理は親リソースが In Service になるまで待機します。LifeKeeper のログは、再同期処理の開始を妨げているリソースと、どのリソースが起動される必要があるかを示したメッセージを含みます。リソースが 単純に OSU になっているのであればリソースを In Service にするか ログメッセージに記載されている `mirror_action` コマンドを実行してください。リソースが OSF の場合は、リソースを In Service にする前に問題を解決することで、データの再同期が行われます。詳細については 再同期前のデータ検証 を参照してください。 |
| LKCLI のエクスポート/インポート機能を用いて、DataKeeper リソースのミラーリング方式がノード毎に異なるようなリソースを作成することはできません | GUI を用いて DataKeeper リソースを作成した場合、プライマリーからセカンダリー間を非同期転送、セカンダリーからプライマリー間を同期転送する設定が出来ますが、その構成をLKCLIでエクスポートした場合に別のクラスター構成にインポートすることはできません。<br><br>**注記:** この構成は、lkcliのエクスポート/インポート機能ではサポートされていません。lkcliのエクスポートはエラーを返しませんが、lkcliのインポートは失敗します。 |
| netraid ミラーに一意の識別子がないという警告メッセージ | 一意の識別子を使用するには、設定をできるだけ早く変更する必要があります。<br><br>修復するための推奨手順は次の通りです。<br>1. すべてのノードで LifeKeeper を起動します。<br><br>2. 安全でない netraid リソースと、各ノードの各リソースの元になるディスクを特定します。各ノードでマッピングが異なる場合があるため、これは各ノードで実施して |

| | |
|---|---|
| | ください。<br><br># ins_list -r netraid -f: \| grep DEVNAME \| grep -v mapper \| cut -f4,5 -d:<br>datarep-test1:/dev/xvdb<br>datarep-test2:/dev/xvdc<br>datarep-filesys3:/dev/xvdd<br><br>**注記:** これは netraid「tag:ID」のリストです。以下の手順では、上記のデバイス名マッピングの一致を想定しています。<br><br>3. デバイスが GPT で設定されているかどうかを確認します。各デバイスについて、GPT getId を実行します。<br><br>`#/opt/LifeKeeper/lkadm/subsys/scsi/gpt/bin/getId -i /dev/xvdb`<br><br>4757cd62-e065-4013-8514-1031b446aa24<br><br>4. getId が一意の ID を返した場合は、インスタンスを更新します。<br><br>#ins_setid -t datarep-test1 -i “4757cd62-e065-4013-8514-1031b446aa24”<br><br><GPT ではないデバイスがある場合は、ステップ5 に進んでください。><br><br>5. 安全でない netraid リソースに依存するリソースを特定します。<br><br># ins_list -r netraid -f: \| grep DEVNAME \| grep -v mapper \| cut -f4 -d: \| while read entry; do dep_list -p $entry -f: 2>/dev/null \| cut -f1 -d:; done<br>/test1<br>/test2<br>filesys3<br><br>**注記:** これは、netraid デバイスに関連付けられたファイルシステムの **タグのリスト** です。<br><br>6. ファイルシステムのマウントポイントを特定します。通常、ファイルシステムリソースのタグはマウントポイントと同じです。同じでない場合は次のコマンドを使用して、ファイルシステムタグをファイルシステムのマウントポイントと一致させることができます。 |

```
# ins_list -t filesys3 -f: | cut -d: -f5
/test3
```

7. すべてのアクティビティーを停止して、ファイルシステムリソースのみを netraid デバイスで In Service にします。

a. すべてのリソースを Out of Service にします。
b. 安全でない netraid デバイス上のファイルシステムのみを In Service にします。

安全でないデバイスについてのみ、以下の手順に従ってください。

8. リソースが In Service で影響を受けるファイルシステム上のすべてのデータのバックアップを取得します。

9. すべてのクラスターノードですべてのリソースを Out of Service にします。

10. LifeKeeper 構成のバックアップを取得します（`lkbackup -c --cluster`）。

11. 安全でない各 netraid リソースに対して階層を削除します。

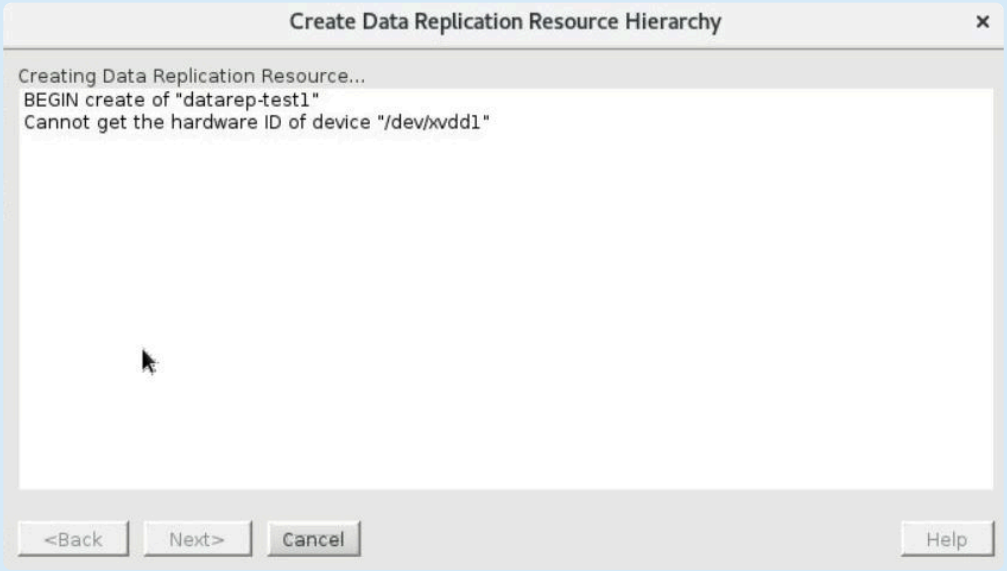
a. 階層に注意してください。影響を受けるアプリケーションは削除されます。
b. 複雑な構成では、階層が複数のブランチで構成されている複数の階層を削除する必要がある場合があります。

12. この時点で残っているのは、安全でない netraid リソースとの依存関係がないリソースのみです。ほとんどの場合、それは IP リソース、EC2 リソースなどです。
13. すべてのノードで `lkstop` を実行します。
14. 影響を受けるすべてのファイルシステムがアンマウントされていることを確認します。
15. 各ノードのデバイスを GPT パーティションテーブル（gdisk や parted などを使用）で再構成するか、LVM を使用します。
16. すべてのノードで LifeKeeper を起動します。
17. ファイルシステムごとにレプリケートされた新しいファイルシステムを作成し、各リソースをすべてのノードに拡張します。

- /dev/xvdb1 -> /test1
- /dev/xvdc1 -> /test2
- /dev/xvdd1 -> /test3

| | |
|---|---|
| | 18. バックアップから各マウントポイントにデータをリストアします。データは自動的にターゲットに再同期されます。<br>19. ステップ 11 で削除したアプリケーション階層を再作成します。<br><br>DataKeeper ストレージ構成オプションの詳細については、SIOS製品ドキュメント を参照してください。 |
| DataKeeper リソースを削除した後に NetRAID デバイスが削除されない。 | NetRAID デバイスがマウントされている場合、DataKeeper リソースを削除しても NetRAID デバイスは削除されません。手動でデバイスをアンマウントした後、以下のコマンドを使用してNetRAIDデバイスを削除することができます。<br><br>*mdadm -S <md_device> (<md_device> を調べるには cat /proc/mdstat)* |
| フェイルオーバー中のエラー | デバイスのステータスを確認してください。再同期が進行中の場合、フェイルオーバーは実行できません。 |
| プライマリーサーバーに障害が発生すると、セカンダリーサーバーの DataKeeper リソースが ISP になります。ただし、プライマリーサーバーが再起動すると、両方のサーバーで DataKeeper リソースが OSF になります。 | DataKeeper リソース階層の作成時に選択した「スイッチバックタイプ」を確認してください。このリリースでは、DataKeeper リソースの自動スイッチバックはサポートされていません。リソースプロパティーのウィンドウで、スイッチバックタイプを [Intelligent] に変更できます。 |
| 両方のサーバーが動作不能になってからプライマリーサーバーが再起動したときに、リソースを ISP にすることができない。 | セカンダリーサーバーよりも前にプライマリーサーバーが動作可能になった場合、DataKeeper リソースを強制的にオンラインにすることができます。このためには、リソースプロパティーのダイアログを開き、 **[Replication Status]** タブ、 **[Actions]** ボタンを順にクリックし、次に **[Force Mirror Online]** を選択してください。 **[Continue]** をクリックして確認してから、 **[Finish]** をクリックしてください。 |
| 現在マウントしている NFS ファイルシステムに DataKeeper 階層を作成するとき | 現在 NFS がエクスポートしたファイルシステムに、DataKeeper 階層を作成しようとしています。エクスポートする前に、このファイルシステムをレプリケーションする必要があります。 |

| | |
|---|---|
| のエラー | |
| DataKeeper リソース作成時の GUI のウイザードに、新しく作成したパーティションがリストされない。 | Linux OS は、システムを次回再起動するまで、新しく作成したパーティションを認識しないことがあります。新しく作成したパーティションのエントリーを調べるには、/proc/partitions ファイルを表示してください。新しく作成したパーティションがこのファイルに表示されない場合、システムを再起動する必要があります。 |
| プライマリーとバックアップの両方のサーバーで、リソースが緑（ISP）で表示される。 | これは、「スプリットブレイン」のシナリオで、一時的な通信障害により発生することがあります。通信の再開後、両方のシステムがそれぞれ、それ自体をプライマリーと見なします。<br><br>いずれのシステムが最終のプライマリーシステムであったかが不明なので、DataKeeper はデータを再同期しません。手動操作が必要です。<br><br>**ビットマップを使用しない場合**:<br><br>最終のバックアップであったサーバーを特定し、そのサーバーのリソースをサービス休止にする必要があります。その後、DataKeeper が全体の再同期を実行します。<br><br>**ビットマップを使用する場合**:<br><br>最終のバックアップであったサーバーを特定し、そのサーバーのリソースをサービス休止にする必要があります。その後、DataKeeper が部分的な再同期を実行します。 |
| Core – 言語環境の影響 | LifeKeeper の一部のスクリプトは Linux のシステムユーティリティーの出力を解析し、情報を抽出するために特定のパターンに依存します。英語圏以外のロケールで一部のコマンドが実行されている場合、予測されたパターンは変更され、LifeKeeper スクリプトは必要な情報の取得に失敗します。このため、 */etc/default/LifeKeeper* では、言語環境変数 LC_MESSAGES が POSIX「C」のロケール (LC_MESSAGES=C) に設定されています。言語を英語にして Linux をインストールする必要はありません (インストールメディアで使用できる任意の言語を選択可能)。 */etc/default/LifeKeeper* の LC_MESSAGES の設定は LifeKeeper にのみ影響します。 */etc/default/LifeKeeper* の LC_MESSAGES の値を変更する場合は、LifeKeeper の動作に悪影響を及ぼす可能性があることに注意してください。悪影響は、メッセージカタログがさまざまな言語とユーティリティーに対応してインストールされているかどうか、および LifeKeeper が予期していないテキスト出力をそれらが生成するかどうかに左右されます。 |
| GUI – GUI の終了後に Web ブラウザから再接続したときに、GUI のログインプロンプトが再表示されないことがある。 | GUI アプレットを終了するか切断してから、同じ Web ブラウザのセッションから再接続しようとすると、ログインプロンプトが表示されないことがあります。<br><br>**回避策:** Web ブラウザを閉じ、Web ブラウザを開き直してからサーバーに接続します。Firefox ブラウザを使用している場合は、Firefox のウィンドウをすべて閉じてから、開き直します。 |

| | |
|---|---|
| DataKeeper リソースの作成 (および拡張) に失敗する | 特定の環境（IDE ディスクエミュレーションを使用した仮想化環境、HP CCISS ストレージを使用したサーバー、ソリッドステートデバイス(SSD)、Amazon EBS ストレージなど）で DataKeeper を使用している場合、ミラー作成時にエラーが発生することがあります。<br>  <br>これは、LifeKeeper が問題のディスクを認識せず、デバイスに関連付ける一意の ID を取得できないためです。<br><br>**回避策:** GUID パーティションを作成し、一意の ID をパーティションに割り当てるか、LVM を使用してください。 |
| アップグレード後、ターゲットの状態が「Out of Sync」になる | LifeKeeper 9.2.2 以降では NU デバイスの使用を推奨していないため、NU デバイスで構成された DataKeeper リソースが存在する環境では、このような症状が発生します。NU デバイスを使用した環境で LifeKeeper 9.2.2 以降にアップグレードする場合には、/etc/default/LifeKeeper に以下の設定を追加してください。<br><br>`LKDR_ALLOW_NU=TRUE`<br><br>**NUデバイスの確認方法:**<br><br>lcdstatus コマンドを実行し、ID に NU- で始まる文字列が含まれる場合は NU デバイスを使用した環境です。 |
| ターゲットに拡張しても、非同期または同期を設定するための「Replication Type」の入力画面が表示されない。 | ミラーが作成されると、[Enable Asynchronous Replication（非同期レプリケーションを有効にする）] で [no] が選択されました。ミラーを削除し、[Enable Asynchronous Replication（非同期レプリケーションを有効にする）] プロンプトが表示されたら [yes] を選択してください。 |
| レプリケーション | ミラーの拡張を解除して再度拡張し、Replication Type を要求されたら「synchronous」 |

| | |
|---|---|
| タイプが同期ではなく非同期になっている。2つのシステム間のレプリケーションは、当初非同期レプリケーション用に構成されていたが、代わりに同期レプリケーションを要求される場合。 | を選択してください。 |
| ターゲットが前のソースを待機中で、同期していない。 | 前のソースをクラスターに接続します。前のソースがタイムリーにクラスターに再参加できない場合は、現在のミラーソースで `$LKROOT/bin/mirror_action <tag> fullresync <source> <target>` コマンドを実行し、ターゲットを完全再同期で再接続してください。 |

# 5.6. コマンドラインインターフェース

## 導入

LifeKeeper のコマンドラインインターフェースは、グラフィカルユーザインターフェースの代わりに使用できます。shell スクリプトに CLI の呼び出しを組み込んで、頻繁に使用する管理タスクを自動化することができます。

## ドキュメントの内容

このガイドでは次の事項について説明します。

- コマンド. CLI コマンドについて説明します。
- shell スクリプトの例. スクリプトで使用する CLI の例をいくつか示します。

## LifeKeeper Documentation

以下に、SIOS Technology Corp が発行している SPS 関連情報の一覧を示します。

- LifeKeeper for Linux リリースノート
- LifeKeeper for Linux テクニカルドキュメンテーション
- LifeKeeper for Linux インストレーションガイド
- オプションの Recovery Kit ドキュメンテーション

このドキュメンテーション、および他の LifeKeeper Recovery Kit に関連するドキュメンテーションは、以下の Web サイトで入手できます。

http://jpdocs.us.sios.com

# shell スクリプトの例

## 階層を作成して拡張する shell スクリプトの例 (抜粋)

2 つの階層間の依存関係を作成する方法の例も示します。

### # システムパラメータが必要

```
LKROOT=/opt/LifeKeeper

OBJ_DIR=/opt/LifeKeeper/lkadm

LKBIN=/opt/LifeKeeper/bin

ExtendPath=/opt/LifeKeeper/lkadm/bin

PATH=$PATH:$LKBIN
```

### # 汎用の引数

```
LocalServer=unix121.ha.uk.sbphrd.com

TargetServer=unix122.ha.uk.sbphrd.com

Node2Priority=10
```

### # 上記の変数は、後続のコマンドで使用される

# 5.6.1. コマンド

## LifeKeeper の bin コマンドのグループ化と基本的な説明

コマンドは、以下のディレクトリにあります。 */opt/LifeKeeper/bin*

これを独自のパスに入れるには、以下のコマンドを実行します。

```
. /etc/default/LifeKeeper
```

### LifeKeeper、GUI などの開始 / 停止

> **!** LifeKeeperが起動し、クラスター内の他のサーバーとの通信を確立する際、クラスター内のすべてのサーバーとの通信が確立されるまで、LifeKeeperはデータレプリケーション（DataKeeper）リソースがIn-Service（手動または自動）になるのを許可しません。すべてのサーバーとの通信が確立される前にデータレプリケーションリソースをIn-Serviceにしようとすると、データレプリケーションリソースはOSFとマークされます（詳細については、システムのログファイルを参照してください）。

`lkstart` : LifeKeeper Core を起動します。

オプション:

なし

`lkstop` : LifeKeeper Coreを停止します。オプションにより停止時の動作が変わります。

> **!** 「lkstop -f」を使用してLifeKeeperを停止すると、データレプリケーションリソースを含め、In-Serviceのリソースは構成された状態で実行されたままとなります。「lkstop -f」の後、階層/リソースが正しいサーバーでIn-Serviceに戻れることが重要です。すべてのサーバーとの通信が確立されると、LifeKeeperはすべてのリソースを自動的にIn-Serviceにします。LifeKeeperがすべてのサーバーの再起動とクラスターへの再参加を待機している間、データレプリケーションリソースは一時的にOSFとマークされる場合があります（詳細については、システムのログファイルを参照してください）。

> **✻** 注意: リソースの停止が失敗した場合でも `lkstop` の **プロセスは続行** され、lkstopが失敗することは **ありません** 。

> **✻** **注意:** `lkstop` の実行および/または複数のクラスターノードのシャットダウンを平行して行わないでください。

> **注意:** クラスター内の処理を完了させるため各 `lkstop` コマンドを発行する際には間隔を最低2分空けることを推奨します。

オプション:

なし – LifeKeeper Core及び保護対象サービスを停止します。シャットダウンストラテジーの設定に関わらず待機系へのスイッチオーバーは行われません。

-r – システム起動時の自動起動を有効のままにします。

-n – LifeKeeper Core及び保護対象サービスを停止します。 シャットダウンストラテジーで “Switchover Resources” に設定されている場合は待機系にスイッチオーバーされます。また、/etc/default/LifeKeeper で LKSTOP_FAILOVER_N=1 が設定されている場合は、シャットダウンストラテジーの設定にかかわらずフェイルオーバーが発生します。

-f – LifeKeeper Coreのみ停止します。保護対象のサービスは停止しません。

-i – LifeKeeper Coreのみ停止します。保護対象のリソースは停止しません。続行するため (yes/no) の確認を行ってください。

`lkGUIserver` : LifeKeeper GUI デーモンプロセスを開始 / 停止します。

オプション:

`start`

`stop`

`restart`

`lkGUIapp` : LifeKeeper Java アプリケーションを開始します。

オプション:

なし

## LK とその他の LifeKeeper のコマンドの監視

`lcdstatus` : LifeKeeper のリソース、コミュニケーションパスなどの状態を表示します。

オプション:

`-d - <node to run command on>`

-q : 短いレポート

lcdsync：メモリにある LifeKeeper の設定情報をディスクに書き込みます。

オプション:

-d - <other node to run it on>

lcdrcp：コミュニケーションパス経由で、ある LifeKeeper ノードからファイルを別の LifeKeeper ノードに転送します。

オプション:

lcdrcp <file names> {dest:ofile | dest:odir}

lcdremexec：指定の LifeKeeper ノードで、指定のコマンドを実行します。

オプション:

-d <node to run command on> <command>

lcdrecover：リソース階層インスタンス状態の確認と設定を行います。

オプション:

man -M /opt/LifeKeeper/man lcdrecover を参照してください。

## 階層を in service または out of service にする

perform_action: 指定リソースで指定動作を実行します。

指定の階層から別のノードへの切り替えに使用できます。

オプション:

-a <action name>

-t <tag name>

例:

perform_action -a restore -t $LKTag：階層を in service にします。

perform_action -a remove -t $LKTag: 階層を out of service にします。

## LifeKeeperのコンフィグレーションをチェックする

lkchkconf : /etc/default/LifeKeeperに設定された値について、以下の内容をチェックします。

- LifeKeeperの起動時に読み込む設定が実際に反映されているか。 反映されていない場合には、その旨を通知するエラーメッセージを出力します。

- LifeKeeperの監視実行の間隔と、リソースの監視処理のタイムアウト値の設定値に矛盾がないか。 リソースの監視処理のタイムアウト値に、LifeKeeperの監視実行間隔（LKCHECKINTERVALの設定値）より大きい値が設定されている場合はその旨を通知するエラーメッセージが出力されます。

  オプション:

  なし

# 5.6.1.1. lklogmsg

このコマンドは、LifeKeeper ロギングインターフェースへのコマンドラインインターフェースを提供します。Recovery Kit の実装の一部であるスクリプトで使用できます。

## 名称

lklogmsg — LifeKeeper ロギングインターフェースへのコマンドラインインターフェース

## 書式

```
LKROOT/bin/lklogmsg [-o] [-e] [-h] [-p progname] -l level -s source -a action
-t tag -i messageid [-c command - cmd args ]|[-- message]
```

## 構文

```
lklogmsg [-o] [-e] [-h] [-p progname] -l level -s source -a action -t tag -i
messageid [-c command - cmd args ]|[-- message]
```

ログに記録されるメッセージは、特別な引数「--」を前に付けて最後の引数として渡されます。または、メッセージパラメーターが存在しない場合は、このコマンドは stdin から入力を読み取り、各行を個別のログメッセージに変換します。

また、 -c オプションでコマンドと引数が指定される場合は、追加の操作モードを使用できます。このモードでは、コマンドの stdout と stderr はパイプを介してリダイレクトされ、 lklogmsg によってキャプチャされます。コマンドの stdinは、 */dev/null* から取得されます。

オプション -o を指定すると、メッセージは stdout にエコーされます。

オプション -e を指定すると、メッセージは stderr にエコーされます。

オプション -h を指定すると、 lklogmsg は nohup と同じアクションを実行します。

-p オプションでは、 lklogmsg を呼び出しているスクリプトの名前を指定します。これは、ログに記録されるアプリケーション名として使用されます。

-l オプションでは、 LK_NOTIFY 、LK_FATAL 、LK_ERROR 、LK_WARN 、LK_TRACE 、LK_INFO または LK_DEBUG のいずれかを指定します。

-s オプションでは、 lklogmsg を呼び出しているApplication Recovery Kit 名 (mysql、lvm、oracleなど) を指定します。

-a オプションでは、 lklogmsg を呼び出している Recovery Kit のアクション (例：restore、remove、recover、create、extend) を指定します。

-i オプションでは、Application Recovery Kit に割り当てられた範囲の（6桁の）メッセージ ID を指定し

ます。

## 例

`/opt/LifeKeeper/bin/lklogmsg -e -l LK_ERROR -s gopher -a recover -t sample_tagname -i` 222016 Unable to restart gopher server sends “Unable to restart gopher server” to both the LifeKeeper log, and to `stderr`.

`/opt/LifeKeeper/bin/lklogmsg -l LK_INFO -s webserver -a create -t sample_sdr_resource -i` 100315

# 5.6.1.2. SYS – LifeKeeper クラスタ内のシステムに関する LifeKeeper のコマンド

sys_list：特定の LifeKeeper ノードが既知のシステムをリストします。

オプション:

```
-d <other node to run it on>
```

sys_create：LifeKeeper ノード上にある他のシステムの情報を作成します。

オプション:

```
-s <remote system name>

-d <node to run command on>
```

sys_remove: LifeKeeper ノード上にある他のシステムの情報を削除します。

オプション:

```
-d <dest>

-s <remote system name to be removed from list>
```

sys_getstate：指定の LifeKeeper ノードに、指定の LifeKeeper ノードの状態をリストします。

オプション:

```
-d <node to run command on>

-s <system concerning the state being checked>
```

sys_setstate：指定の LifeKeeper ノードに、指定の LifeKeeper ノードの状態を設定します。

オプション:

```
-d <node to run command on>

-s <system concerning the state being set>

-S <actual state> {DEAD|ALIVE|UNKNOWN}

-R <reason for state setting>
```

sys_getdescr：システムが現在の状態に変化した理由に関する一部の情報を出力します。

オプション:

```
-d <node to run command on>

-s <system to get data on>
```

# 5.6.1.3. NET – コマンドに関するコミュニケーションパス

net_create ：2 つの SPS ノード間にコミュニケーションパスを作成します。

オプション:

```
-d <node to run command on>

-s <other system>

-D <device path>

-n <TTY or TCP>

-b <baud rate>

-r <remote IP address>

-l <local IP address>

-p <priority>
```

net_remove ：2 つの SPS ノード間にあるコミュニケーションパスを削除します。

オプション:

```
-d <node to run command on>

-s <remote server name to be removed from>

-D <device path>

-r <remote IP address>
```

net_list ：指定の SPS ノードに関するコミュニケーションパスの情報をリストします。

オプション:

```
-d <node to run command on>

-f: <フィールド区切り文字「:」>

-s <system name>
```

net_change：指定のコミュニケーションパスに関する特定の情報を変更します。

オプション:

```
-d <node to run command on>

-s <server name for data to be modified>

-D <device>
```

Crelcm：コミュニケーションパスを作成します。

```
/opt/LifeKeeper/lkadm/bin/crelcm <node 1> <node 2> <net type> <baud rate> <IP address 1> <IP address 2> <prio>
```

portio：2 つの SPS ノード間のシリアル通信をテストします。

# 5.6.1.4. FLAG – LifeKeeper の内部フラグに関するコマンド

`flg_create` ：指定ノードに指定の LifeKeeper フラグを設定します。

オプション:

```
-d <node to run command on>

-f <flag name>
```

`flg_remove` ：指定ノードから指定の LifeKeeper フラグを削除します。

オプション:

```
-d <node to run command on>

-f <flag name>
```

`flg_list` ：指定ノードに設定された LifeKeeper のフラグをすべてリストします。

オプション:

```
-d <node to run command on>
```

# 5.6.1.5. TYP – リソース階層タイプに関する LifeKeeper のコマンド

typ_create ：指定ノードに指定のリソースタイプを作成します。

オプション:

-d <node to run command on>

-a <app type> (アプリケーションが最初に必要)

-r <resource type>

typ_remove ：指定ノードのリソースツリーから指定のリソースタイプを削除します。

オプション:

-d <node to run command on>

-a <application type>

-r <resource type>

typ_list ：指定ノード上にあるリソースタイプをすべてリストします。

オプション:

-d <node to run command on>

-f: <フィールド区切り文字「:」>

-a <app type>

# 5.6.1.6. APP – リソースアプリケーションに関する LifeKeeper のコマンド (関連タイプのグループ)

app_create ：指定ノードに指定のリソースアプリケーションを作成します。

オプション:

```
-d <node to run on>

-a <application name>
```

app_remove ：指定ノードのリソースツリーから指定のリソースアプリケーションを削除します。

オプション:

```
-d <dest>

-a <application type>
```

app_list ：指定ノード上にあるリソースアプリケーションをすべてリストします。

オプション:

```
-d <node to run on>
```

# 5.6.1.7. DEP – リソースアプリケーションを相互に関連付ける方法に関する LifeKeeper のコマンド

## コマンドの実行

- /opt/LifeKeeper/bin へ移動
- [コマンド] [オプション] の形式で実行 (例： (**./dep_list^-P**)

# コマンド

注意：コマンドの詳細については、LifeKeeperと共にインストールされているマニュアルページを参照してください。.

## dep_create – 2つのリソースインスタンスの間に依存関係を作成する

オプション/説明

**オプション:**

-p <parent tag>
-c <child tag>
-d <dest> （任意）

**説明:** 1つのシステム上で、 **<parent tag>** と **<child tag>** で指定したタグを持つリソースインスタンスの間に依存関係を作成します。

**追加情報:**

- -p および -c で指定したリソースはいずれも、引数 -d で指定したシステム上にある必要があります。これは、親リソースが適切な動作のために子リソースを必要とすることを意味します。
- -d で宛先が指定されていない場合、コマンドを実行したシステムが想定されます。
- **-p および -c で指定したリソースはいずれも、事前に作成しておく必要があります。**

注意: 依存関係は、コマンドが実行されるシステムまたは 引数 **-d** 指定されたシステム上にのみ作成されます。依存関係の設定を確実に行うためには、このコマンドをすべてのクラスターノードに対して実行する必要があります。 これを行わないと、予期せぬ障害が発生する可能性があります。

## dep_remove – 2 つのリソースインスタンス間にある依存関係を削除する

オプション/説明

オプション：

-p <parent tag> (任意)
-c <child tag> (任意)
-d <dest> (任意)

**説明:** 1つのシステム上で、 <parent tag> と <child tag> で指定したタグを持つリソースインスタンス間の依存関係を削除します。

**追加情報:**

- -d で宛先が指定されていない場合、コマンドを実行したシステムが想定されます。
- -c で子リソースが指定されていない場合、-p で指定された親リソースとのすべての依存関係が削除されます。
- -p で親リソースが指定されていない場合、-c で指定された子リソースを持つすべての依存関係が削除されます。
- -p、-cのいずれか、または両方を指定する必要があります。 **どちらも指定されていない** 場合、コマンドはエラーとなります。

注意: 依存関係は、コマンドを実行したシステムまたは **引数 -d** で指定されたシステムでのみ削除されます。したがって、依存関係を完全に削除するには、すべてのクラスターノードに対してコマンドを実行する必要があります。 これを行わないと、予期せぬ障害が発生する可能性があります。

## dep_list – 2 つのインスタンス間にある依存関係をリストする

オプション/説明

オプション:

-p <ofchildtag> (任意)
-c <ofparenttag> (任意)
-C <allchild> (任意)
-P <allparent> (任意)
-r <typ> (任意)
-a <app> (任意)
-d <dest> (任意)
-f <field separator of ':'> (任意)

**説明:** リソースインスタンス間の依存関係に関する情報を、標準出力に出力します。 **注意:** -d で宛先が指定されていない場合、コマンドを実行した現在のシステムが想定されます。

このコマンドを実行すると、各文字列が次の形式で表示されます。
**Parent =** _____ **Child =** _____

```
varfs:part10
part10:ha5
usrfs:part20
part20:ha5
```

各文字列には、 **区切り文字** で区切られた2つのフィールドがあります。dep_list コマンドの後に -f <任意の文字> を使用して、文字列内の2つのリソースインスタンスを区切る区切り文字を設定します。上記の例では、区切り文字としてコロン (:) を使用しています。区切り文字より左側フィールドは依存関係のあるリソースのうち親リソースのタグ名、右側のフィールドは子リソースのタグ名を示しています。

例:

```
[ bin]# ./dep_list -f:
SAP-PDX_ERS10:ers-ip-12.1.2.0
ers-ip-12.1.2.0:ec2-12.1.2.0
SAP-PDX_ASCS00:ascs-ip-12.1.1.0
ascs-ip-12.1.1.0:ec2-12.1.1.0
```

**注意**: -f オプションを使用しない場合でも、出力には control->A の区切り文字が含まれます。 (*これは印刷不可能な文字で、文字列には表示されません。*)

**追加情報:**

- -p オプションが指定されている場合、<ofchildtag>で指定されたリソースの直接の親の依存関係のみを出力します。
- -c オプションが指定されている場合、<ofparenttag> で指定されたリソースの直接の子の依存関係のみを出力します。
- -C オプションが指定されている場合、 <allchild> で指定されたリソースのすべての直接および間接の子の依存関係を出力します。
- -P オプションが指定されている場合、<allparent> で指定されたリソースのすべての直接および間接の親の依存関係を出力します。
- -p、-c、-P、または -Cオプションがいずれも指定されていない場合、すべての依存関係が出力されます。
- -r オプションを指定すると、子の typ のすべての依存関係がリストされます。
- -a オプションを指定すると、アプリケーションの app のすべての依存関係がリストされます。

## eqv_create – 2 つのノード間に指定リソースのイクイバレンシーを作成する

オプション/説明

**オプション:**

```
-t <tag>
-o <tag on equivalent system>
-S <equivalent system>
-e <{COMMON|SHARED|COMPOSITE}>
-p <Priority> (任意)
```

```
-d <dest> (任意)
-r <Priority on equivalent system> (任意)
```

**説明:** ローカルシステム（-d で宛先が指定されていない場合）の <tag> で指定されたリソースと <equivalent system> 上の <tag on equivalent system> で指定した リソースとの間のイクイバレンシーを、構成データベース内に作成します。

**追加情報:**

- -d で宛先が指定されている場合、コマンドは <dest> システムで実行されます。指定されていない場合、コマンドはローカルシステムで実行されます。-d の使用がどのように影響するかについては、以下の例を参照してください。
- -p の引数 <Priority> および -r の引数 <Priority on equivalent system> は、イクイバレンシー情報を持つシステムのリソースの優先順位を表します。
- <Priority> および <Priority on equivalent system> オプションは、指定されていない場合、それぞれデフォルトで 1、2 になります。
- -S <equivalent system> で指定されたリモートシステムにイクイバレンシーを自動的に追加します。
- -e で指定するイクイバレンシータイプは、現在 **SHARED**のみ がサポートされます。

この例では、次のように仮定しています。:
NodeAのリソース： **TagOnNodeA**
NodeBのリソース： **TagonNodeB**

**例:**
**TagOnNodeA** と **TagOnNodeB** の間にイクイバレンシーを作成するには、以下を実行します（両方ともNodeAで実行します）。

- `eqv_create -t TagOnNodeA -p 1 -S NodeB -o TagOnNodeB -r 10 -e SHARED`

または

- `eqv_create -d NodeB -t TagOnNodeB -p 10 -S NodeA -o TagOnNodeA -r 1 -e SHARED`

## eqv_remove – 2 つのノード間にある指定リソースのイクイバレンシーを削除する

オプション/説明

**オプション:**

```
-S <equivalent system>
-t <tag>
-e <{COMMON|SHARED|COMPOSITE}>
-d <dest> (任意)
-o <tag on equivalent system> (任意)
```

**説明:** ローカルシステム（-d で宛先が指定されていない場合）の <tag> で指定されたリソースと <equivalent system> 上の <tag on equivalent system> で指定したリソースとの間のイクイバレンシー

を、構成データベースから削除します。

**追加情報:**

- -d で宛先が指定されている場合、コマンドは <dest> システムで実行されます。指定されていない場合、コマンドはローカルシステムで実行されます。-d の使用がどのように影響するかについては、以下の例を参照してください。
- -d で宛先が指定されていない場合、 <tag> はローカルノードのリソースタグを表します。指定されている場合、 <tag> は <dest> のリソースタグです。
- -e で指定するイクイバレンシータイプは、現在 **SHARED のみ** がサポートされます。
- -S <equivalent system> で指定されたリモートシステムのイクイバレンシーを自動的に削除します。

✻ この例では、次のように仮定しています。:
NodeAのリソース： **TagOnNodeA**
NodeBのリソース： **TagonNodeB**

**例:**
**TagOnNodeA** と **TagOnNodeB** の間のイクイバレンシーを削除するには、以下を実行します（両方ともNodeAで実行します）。

- `eqv_remove -t TagOnNodeA -S NodeB -e SHARED`

または

- `eqv_remove -d NodeB -t TagOnNodeB -S NodeA -e SHARED`

## eqv_list - リソースインスタンス間にあるイクイバレンシー関係をリストする

オプション/説明

オプション:

`-d <dest>` (任意)
`-S <system>` (任意)
`-t <tag name>` (任意)
`-f: <field separator of ':'>` (任意)

**説明:**
リソースインスタンス間のイクイバレンシー関係に関する情報を、標準出力 に出力します。

**追加情報:**

- -dで宛先が指定されている場合、イクイバレンシーのリストはそのシステムからのリストになり、指定されていない場合は、ローカルシステムからのリストになります。
- -S で システムが指定されている場合、イクイバレンシーのリストは、ローカルシステム間に存在する任意のイクイバレンシーに対するものになります。

  **例:**

- iwstp:varfs:remote:varfs_backup:SHARED:1:2
- iwstp:usrfs:remote:usrfs_backup:SHARED:1:2

上記の例の各行には、 **区切り文字** で区切られた各文字列のフィールドが含まれています。 eqv_list コマンドの後に-f <任意の文字> を使用して、出力される文字列のフィールドを区切る区切り文字を設定します。上記の例では、区切り文字としてコロン (:) を使用しています。フィールドは以下のとおりです（フィールド1、2、6はローカルシステムの情報、フィールド3、4、7はリモートシステムに関する情報、フィールド5はイクイバレンシーのタイプです）。:

1. イクイバレンシーのリソースタグ1が存在するローカルシステム名: ***iwstp***
2. リソースタグ1のタグ名: ***varfs***
3. イクイバレンシーのリソースタグ2が存在するリモートシステム名: ***remote***
4. リソースタグ2のタグ名: ***varfs backup***
5. イクイバレンシーのタイプ: ***SHARED.***
6. ローカルシステム/リソースの優先順位: ***1***
7. リモートシステム/リソースの優先順位: ***2***

この関数の残りの引数は、以下のように、出力される情報を制限します。

8. -e SHARED プションは、すべての SHARED イクイバレンシー情報を出力します。
9. -t オプションは、引数で指定されたタグに関連するイクイバレンシーのみを含むように出力を制限します。

## hry_setpri - 指定のノードまたはそのノード上にある階層の優先順位を設定する

オプション/説明

**オプション:**

```
-t <tag(s)>
-p <priority>
-d <dest> (任意)
-q <Details in Description> (任意)
-l: <Details in Description> (任意)
```

**説明:** この関数は、指定された <tag(s)> リソース階層内のリソースインスタンスを <priority> に設定します。 -d で宛先が指定されていない場合、コマンドを実行した現在のシステムが想定されます。

**追加情報:**

- 関連するすべてのイクイバレンシーが更新されます。
- <priority> は、既存のイクイバレンシーによって使用中であってはなりません。
- 階層は、-t オプションで指定されたルートリソース <tag(s)> によって識別されます。共通のリソースを持つ複数のルート階層が存在する場合、すべてのルートは、「-t H1,H2」など、 カンマ(スペースは使用できません）で区切られたタグのリストで指定する必要があります。
- 新しい優先順位は、-p オプションを使用して指定します。

- -q オプションを指定した場合、標準出力は すべて抑制されます。
- -l オプションを指定すると、この階層が存在するシステムのリストと、それに関連する優先順位が表示されます。例：

```
teak 10
ash 20
plum 30
```

**注意:** 1つのシステム上にしか存在しないリソース階層の優先順位を設定することはできません。 hry_setpri コマンドは、プライマリーシステムを超えて拡張されていないリソースが見つかるたびに、警告メッセージを発行します。

# 5.6.1.8. INS – 個々の LifeKeeper 階層インスタンスに関するコマンド

ins_list ：指定のリソース階層インスタンスについて現在の情報をリストします。

オプション:

-d <dest>

-f:<フィールド区切り文字「:」>

-a / -r / -t / -i は、オプションのアプリケーション、タイプ、タグ、および ID の情報を指定します。

ins_setas ：指定階層の自動スイッチバックストラテジーを設定します。

オプション:

-d <dest>

-t <tag name>

-s <switchback typ> {INTELLIGENT|AUTOMATIC}

ins_setinit ：LifeKeeper の起動時に指定リソースを初期化する方法を定義します。

オプション:

-d <dest>

-t <tag name>

-I <init state> {AUTORES_ISP|INIT_ISP| INIT_OSU}

ins_setinfo：指定のリソース階層について、情報文字列を定義します。

オプション:

-d <dest>

-t <tag name>

-v <string of information>

ins_setstate ：指定ノード上にある指定リソース階層の状態を設定します。

オプション:

```
-d <dest>

-t <tag name>

-S <state to set instance> {ISP|ISU|OSU}

-R <reason for state setting>

-A <recursively set all resources that depend on this one>
```

ins_gettag：関連する ID のタグ名をリストします。

オプション:

```
-i <id>
```

# 5.6.1.8.1. 階層の拡張解除

```
/opt/LifeKeeper/lkadm/bin/unextmgr <Node_Name> <Tag_Name>
```

# 5.6.1.9. マニュアルページへのアクセス

LifeKeeper for Linux のインストールには、マニュアルページが含まれています。これは、 **lkMAN package** とともに提供されます（完全なパッケージ名については、以下を参照してください）。

```
steeleye-lkMAN-9.5.1-7154.noarch
```

## LifeKeeper コマンドのマニュアルページを表示するには、以下の手順に従ってください。

1. OS に LifeKeeper for Linux がインストールされていることを **確認** します。(LifeKeeper for Linux のインストールについては、 こちら をクリックしてください。)
2. コマンドプロンプトから、次のコマンドを **実行** します。

```
rpm -qa | grep steel
```

**注意:** これにより、インストールされているすべての LK rpm パッケージが表示されます（以下を参照）。

3. **lkMAN** パッケージがインストールされていることを確認したら 、使用している OS に manpage rpms がインストールされていることを確認します。

4. 以下のコマンドを **実行** して、MANPATH を設定します。

```
[ec2-user@use1pdxcs01 ~]$ MANPATH=/opt/Lifekeeper/man
[ec2-user@use1pdxcs01 ~]$ export MANPATH
[ec2-user@use1pdxcs01 ~]$ echo $MANPATH
/opt/Lifekeeper/man
```

> **注意:** MANPATH は、再起動した後毎回設定する必要がある場合があります。

これで、マニュアルページにアクセスできます。

## マニュアルのあるコマンド

特定のコマンドのマニュアルは、以下に示すように /opt/LifeKeeper/bin にあります。

```
[root@use1pdxcs01 bin]# ls
a2p                     lcdmachfail             openssl
app_create              lcdrcp                  pcregrep
app_list                lcdrecover              pcretest
app_remove              lcdremexec              perform_action
backupadm               lcdreserve              perl
c2ph                    lcdrmipc                perl5.8.8
can_talk                lcdstatus               perlbug
certtool                lcdsync                 perlcc
checkrklic              lcduname                perldoc
cpan                    lcdwait                 perlivp
create_terminal_leaf    lcm                     piconv
credstore               libnetcfg               pl2pm
c_rehash                lkbackup                pod2html
cre_mbox                lkcheck                 pod2latex
curl                    lk_chg_value            pod2man
delallsys               lkchkconf               pod2text
dep_create              lkcli                   pod2usage
dep_list                lk_configsnmp           podchecker
dep_remove              lk_configsnmp           podselect
dprofpp                 lk_confirmso            portio
empty_mbox              lkcore_parameter        prove
enc2xs                  lkdefault               psed
eqv_create              lkdiskmon               psktool
eqv_list                lkdisktest              pstruct
eqv_remove              lkexterrlog             qwk_storage_exit
eventslcm               lkGUIapp                qwk_storage_init
find2perl               lkGUIserver             rcv
flg_create              lkID                    rcv_msg
```

## マニュアルページにアクセスするためのコマンド

上記のコマンドのマニュアルページを表示するには、次のコマンドを実行します。

- man <command>

例:

```
# man dep_list
```

## トラブルシューティング

> ! 手順１～３を実施済で以下のエラーとなる場合、MANPATHの設定が不足している可能性があります。（手順４以降を実施してください）

```
[ec2-user@use1pdxcs01 ~]$ man dep_list
No manual entry for dep_list
```

# 5.6.2. LKCLI（LifeKeeper コマンドラインインターフェース）

LKCLI は、LifeKeeper の GUI で操作可能な機能をコマンドラインインターフェースで実現しています。また、GUI には無い機能として、コミュニケーションパスおよびリソース情報の export/import 機能があります。export/import 機能は、作成したシステムの量産や、テスト環境から本番環境のデプロイを容易に行える手段となります。

## サポート対象

次に示す環境での使用をサポート対象としています。

- ノード数： 2ノード構成のクラスタ環境のみサポートされます。
- OS：LifeKeeper 本体でサポートしている OS はすべてサポートされます。
- コミュニケーションパス：ノード間が TCP/IP で接続されている環境のみサポートされます。
- アプリケーションリカバリーキット： ARK 一覧 に記載されたアプリケーションリカバリーキットのみで構成された環境のみサポートされます。ノード内にサポート対象外の ARK リソースが作成されている場合、その環境はサポート対象外となります。

サポート対象外の環境でのコマンドライン操作をご検討されているお客様は、「 コマンドラインインターフェース 」の使用をご検討ください。

## root 以外のユーザーで実行する方法

LKCLI を実行するには、ユーザーは LifeKeeper の次の 3 つグループのいずれかのメンバーである必要があります：lkadmin、lkoper または lkguest。これらのグループについては こちら を参照ください。
root 以外のユーザーで LKCLI を実行する場合は、sudo コマンドを使用して実行してください。なお、lkcli コマンドは絶対パスで指定してください。コマンド実行時に sudo コマンドを実行しているユーザーの認証が求められます。以下、LifeKeeper 起動の例です。

```
$ sudo /opt/LifeKeeper/bin/lkcli start
```

LifeKeeper の sudo 設定は /etc/sudoers.d/lifekeeper で行っています。もし /etc/sudoers で LifeKeeper の設定がインクルードされていない場合は、 "#includedir /etc/sudoers.d" もしくは "#include /etc/sudoers.d/lifekeeper" を /etc/sudoers に追記してください。その他にも /etc/sudoers で secure_path オプションを設定することでパスを指定したり、認証の設定を変更することができます。詳細は man sudoers(5) を参照してください。

## 制限事項

- ホスト名を正しく設定できていない場合、 GUI の起動やログインができません。このとき一部の

lkcli コマンドは使用できますが、多くの lkcli コマンドの処理が停止します。ホスト名が正しく設定されていることを確認してください。

## コマンド一覧

| | |
|---|---|
| lkcli license | ライセンスキーのインストール |
| lkcli start | LifeKeeper の起動 |
| lkcli stop | LifeKeeper の停止 |
| lkcli import | コミュニケーションパス、リソース作成 |
| lkcli export | コミュニケーションパス、リソース情報の出力 |
| lkcli clean | コミュニケーションパスとリソースの削除 |
| lkcli commpath | コミュニケーションパスの操作 |
| lkcli dependency | 依存関係の操作 |
| lkcli resource | リソースの操作 |
| lkcli status | リソース状態の一覧表示 |
| lkcli log | LifeKeeper のログ表示 |
| lkcli server | サーバーの設定･操作 |
| lkcli mirror | DataKeeper のミラーリング操作 |
| lkcli esxi | VMDK RK で利用する ESXi ホストの設定 |

---

### すべてのコマンドで利用できる共通オプション

| オプション | デフォルト | 説明 |
|---|---|---|
| [--remote <str>] | | コマンドを実行したいマシンのホスト名。<br>このオプションを指定しない場合は、ローカルマシン上で実行されます。<br>リモートでコマンドを実行するには、事前に実行したいマシンとコミュニケーションパスが双方向に接続されている必要があります。<br>なお、start、stop、commpath および clean コマンドはサポートされません。 |

注記: オプション表の [ ] は、任意のオプションであることを表しています。

---

## lkcli license

LifeKeeper のライセンスを登録します。

| 権限 | lkadmin |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --file <str> | | ライセンスファイルのパス |

---

## lkcli start

LifeKeeper を起動します。

| 権限 | lkadmin |
|---|---|

---

## lkcli stop

LifeKeeper を停止します。

| 権限 | lkadmin |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| [-f] | | LifeKeeper daemon のみを停止し、保護しているサービスは停止しません。 |
| [-r] | | LifeKeeper daemon の自動起動の設定を変更せずに LifeKeeper を停止します。 |
| [-n] | | LifeKeeperの停止と共に、フェイルオーバーが実行されます。<br>このオプションは -f, -i, -r とは併用できません。 |
| [-i] | | LifeKeeper daemon のみを停止し、保護対象のサービスは停止しません。停止処理を続行するために「yes/no」の確認を行う必要があります。 |

---

## lkcli import

LifeKeeper の設定をファイルから読み込み、コミュニケーションパス、リソースを作成します。

### lkcli import commpath

LifeKeeper の設定をファイルから読み込み、コミュニケーションパスを作成します。
双方向にコミュニケーションパスを接続するには、ローカルマシン・リモートマシンの双方でこのコマンドを実行し、コミュニケーションパスを作成する必要があります。

| 権限 | lkadmin |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --file <str> | | LifeKeeperの設定ファイルのパス。<br>`lkcli export` で出力を保存したファイル (YAML 形式) からコミュニケーションパスを作成します。<br>作成できるコミュニケーションパスのプロトコルは、TCP/IP (ソケット) です。 |

### lkcli import resource

LifeKeeper の設定をファイルから読み込み、リソースを作成します。

| 権限 | lkadmin |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --file <str> | | LifeKeeper の設定ファイルのパス。<br>`lkcli export` で出力を保存したファイル (YAML 形式) からリソースを作成します。<br>作成できるリソースは、 ARK 一覧 を参照してください。 |

> **注記:**
> ・ホスト名や IP アドレス、アプリケーションなどの環境が揃っていないと失敗します。
> ・失敗してもロールバックはされません。一部のみ作成される場合があります。
> ・既にリソースが存在する状態での実行は、サポート対象外です。

---

## lkcli export

LifeKeeper の設定を出力。
現在の LifeKeeper の設定が出力され、コマンドを実行したノードとコミュニケーションパスが接続されているすべてのノードが対象となります。
出力結果をファイルに YAML 形式で保存してください。

```
# lkcli export > lk_export.yml
```

> **注記:**
> ・出力できるコミュニケーションパスのプロトコルは、TCP/IP (ソケット) です。
> ・出力できるリソースは、 ARK 一覧 を参照してください。
> ・リソースの状態 (In Service、Out of Service など) は出力されません。
> ・サーバーのプロパティー値は出力されません。
> ・出力されるのは LifeKeeper 上の設定値のみで、保護するアプリケーション自体の設定は出力されません。
> ・出力した設定ファイルについては、ホスト名と IP アドレスのみ手動で編集してください。その他項目の手動での変更は、サポート対象外です。

| 権限 | lkadmin |
|---|---|

---

## lkcli clean

LifeKeeper の設定を削除。
コマンドを実行したノードのコミュニケーションパスとリソース設定を削除します。
全てのノードの LifeKeeper の設定を削除したい場合は、すべてのノードでこのコマンドを実行してください。その際、コミュニケーションパスが削除されると、その後リモートでコマンドを実行できないのでご注意ください。

| 権限 | lkadmin |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --mode <str> | | **all** または **resource** を指定。<br>• **all** : 全てのコミュニケーションパスと全てのリソースを削除。<br>• **resource** : 全てのリソースを削除。 |

---

## lkcli commpath

コミュニケーションパスの操作

### lkcli commpath create

コマンドを実行したノードのコミュニケーションパスを作成。
作成されるコミュニケーションパスは、ローカルマシンからリモートマシンへのパスとなります。そのため、双方向にコミュニケーションパスを接続するには、ローカルマシン・リモートマシンの双方でこのコマンドを実行し、コミュニケーションパスを作成する必要があります。作成できるコミュニケーションパスのプロトコルは、TCP/IP (ソケット) です。

| 権限 | lkadmin |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --laddr <str> | | コミュニケーションパスに設定するローカルマシン上の IP アドレス。 |
| --raddr <str> | | コミュニケーションパスに設定するリモートマシン上の IP アドレス。 |
| --dest <str> | | コミュニケーションパスに設定するリモートマシンのホスト名。 |
| [--priority <str>] | 既存のパスの最大値 +1 | コミュニケーションパスのプライオリティー。<br><br>**注記:** ローカルマシン・リモートマシンの双方で同一の値を指定してください。 |

### lkcli commpath delete

コマンドを実行したノードのコミュニケーションパスを削除。
削除されるコミュニケーションパスは、ローカルマシンからリモートマシンへのパスとなります。そのため、双方向にコミュニケーションパスを削除するには、ローカルマシン・リモートマシンの双方でこのコマンドを実行し、コミュニケーションパスを削除する必要があります。

| 権限 | lkadmin |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --laddr <str> | | コミュニケーションパスに設定するローカルマシン上の IP アドレス。 |
| --raddr <str> | | コミュニケーションパスに設定するリモートマシン上の IP アドレス。 |
| --dest <str> | | コミュニケーションパスに設定するリモートマシンのホスト名。 |

---

## lkcli dependency

LifeKeeper のリソースの依存関係を作成・削除します。

### lkcli dependency create

2つのリソース間に新しく依存関係を作成します。

| 権限 | lkadmin |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --parent <str> | | 親となるリソースのタグ名。 |

| | | |
|---|---|---|
| --child <str> | | 子となるリソースのタグ名。 |

### lkcli dependency delete

2つのリソース間にある依存関係を削除します。

| 権限 | lkadmin |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --parent <str> | | 削除したい依存関係の親にあたるリソースのタグ名。 |
| --child <str> | | 削除したい依存関係の子にあたるリソースのタグ名。 |

---

## lkcli resource

LifeKeeper のリソースを操作します。

### lkcli resource create

リソースを作成します。

| 権限 | lkadmin |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --tag <str> | | 作成するリソースのタグ名。<br>存在するタグ名は、作成できません。その他については、「リソースタグ名の制限」を参照してください。 |
| [--switchback <str>] | INTELLIGENT | リソースのスイッチバック設定。<br>「INTELLIGENT」か「AUTOMATIC」を指定してください。 |
| ARK 毎のオプション | | オプションは ARK によって異なります。<br>ARK 毎のオプションに関しては、「ARK 毎のサブコマンド」を参照してください。 |

### lkcli resource extend

リソースを拡張します。

> **注記:** 依存関係が設定されていても、対象のリソースのみ拡張されます。拡張はリソース毎に実施する必要があります。

| 権限 | lkadmin |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --tag <str> | | 拡張するリソースのタグ名。<br>拡張先に存在するタグ名は、拡張できません。その他については、「リソースタグ名の制限」を参照してください。 |
| --dest <str> | | リソース階層を拡張するターゲットサーバーのホスト名。 |
| [--switchback <str>] | INTELLIGENT | リソースのスイッチバック設定。<br>「INTELLIGENT」か「AUTOMATIC」を指定してください。 |
| [--template_priority <num> | 1 | 拡張元のリソース階層の優先順位。<br>優先順位は 1 ～ 999 の範囲で未使用の値が有効で、小さい数字ほど優先順位が高くなります。 |
| [--target_priority <num> | 10 | 拡張先のリソース階層の優先順位。<br>優先順位は 1 ～ 999 の範囲で未使用の値が有効で、小さい数字ほど優先順位が高くなります。 |
| ARK 毎のオプション | | オプションは ARK によって異なります。<br>ARK 毎のオプションに関しては、「ARK 毎のサブコマンド」を参照してください。 |

### lkcli resource config

リソースの設定を変更します。

| 権限 | lkadmin |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --tag <str> | | 変更するリソースのタグ名。 |
| ARK 毎のオプション | | オプションは ARK によって異なります。<br>ARK 毎のオプションに関しては、「ARK 毎のサブコマンド」を参照してください。 |

### lkcli resource unextend

リソースの拡張を解除します。

| 権限 | lkadmin |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --tag <str> | | 拡張を解除するリソースのタグ名。 |

| | | |
|---|---|---|
| --dest <str> | | リソース階層の拡張を解除するターゲットサーバーのホスト名。 |

### lkcli resource delete

リソースを削除します。

| 権限 | lkadmin |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --tag <str> | | 削除するリソースのタグ名。 |

### lkcli resource restore

実行ノード上のリソース階層を In Service にします。

| 権限 | lkadmin, lkoper |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --tag <str> | | In Service にするリソースのタグ名。 |

### lkcli resource remove

実行ノード上のリソース階層を Out of Service にします。

| 権限 | lkadmin, lkoper |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --tag <str> | | Out of Service にするリソースのタグ名。 |

### lkcli resource info

リソースのプロパティー情報を出力します。出力内容は ARK 毎に異なります。

| 権限 | lkadmin, lkoper, lkguest |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --tag <str> | | プロパティー情報を出力するリソースのタグ名。 |

### lkcli resource eqv

リソースのイクイバレンシ情報を出力します。

| 権限 | lkadmin, lkoper, lkguest |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --tag <str> | | イクイバレンシ情報を出力するリソースのタグ名。 |

### lkcli resource reorder-priority

実行ノード上のリソースの優先順位を変更します。

| 権限 | lkadmin |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --tag <str> | | 優先順位を変更するリソースのタグ名。 |
| --priority <num> | | 変更後の優先順位。<br>優先順位は 1 ～ 999 の範囲で未使用の値が有効で、小さい数字ほど優先順位が高くなります。 |

### lkcli resource switchback

実行ノード上のリソースのスイッチバック設定を変更します。

| 権限 | lkadmin |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --tag <str> | | 変更するリソースのタグ名。 |
| --switchback <str> | INTELLIGENT | リソースのスイッチバック設定。<br>「INTELLIGENT」か「AUTOMATIC」を指定してください。 |

---

## lkcli status

LifeKeeper のステータスを表示します。
出力されるステータス情報は「 ステータスの詳細表示 」をご参照ください。

| 権限 | lkadmin, lkoper, lkguest |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| [-q] | | ローカルシステムに存在する各リソースの情報を短いレポートで出力します。 |

| | | |
|---|---|---|
| [-e] | | ローカルシステムに存在する各リソースの情報を短いレポートで出力し、（次に優先度の高い）バックアップシステムをリストします。 |
| [-u] | | -q または -e オプションの出力の重複リソースエントリーを抑制します。 |
| [-r <str>] | | リソースルートタグの指定。<br>レポートを特定のリソースルートタグに制限します。 |

---

## lkcli log

LifeKeeper のログを表示します。

| 権限 | lkadmin, lkoper, lkguest |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --lines <num> | 10 | 表示するログの行数。 |

---

## lkcli server

LifeKeeper のサーバーに関する操作を行います。

### lkcli server info

サーバーのシャットダウン方法とフェイルオーバー機能を確認します。

| 権限 | lkadmin, lkoper, lkguest |
|---|---|

### lkcli server shutdown-strategy

サーバーのシャットダウン方法を設定します。

| 権限 | lkadmin |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --mode <str> | | **switchover** または **do_not_switchover** の値。<br>• **switchover** : LifeKeeper は、正常なシャットダウンでバックアップサーバーのリソースを起動します。<br>• **do_not_switchover** : LifeKeeper は、正常なシャットダウンではバックアップサーバーのリソースを起動しません。 |

### lkcli server confirmso

LifeKeeper クラスタのノード障害によるフェイルオーバーが発生した時、バックアップノードへの切り替えの実行に対してユーザー確認をするか設定します。
詳細については、「 [Confirm Failover] と [Block Resource Failover] の設定 」の [Confirm Failover] の項目を参照してください。

| 権限 | lkadmin |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --server <str> | | フェイルオーバー先のサーバー名。 |
| --value <str> | | **enable** または **disable** の値。<br>• **enable** : フェイルオーバー時のユーザー確認を有効にします。<br>• **disable** : フェイルオーバー時にユーザー確認をしません。 |

### lkcli server block-failover

指定したシステムでのリソース障害に起因するフェイルオーバーをブロックする設定を行います。
詳細については、「 [Confirm Failover] と [Block Resource Failover] の設定 」の [Block Resource Failover] の項目を参照してください。

| 権限 | lkadmin |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --server <str> | | フェイルオーバー先のサーバー名。 |
| --value <str> | | **enable** または **disable** の値。<br>• **enable** : 指定サーバーへのフェイルオーバーをブロックします。<br>• **disable** : 指定サーバーへのフェイルオーバーをブロックしません。 |

### lkcli server block-all-failovers

全サーバーでノード障害とリソース障害のいずれかの障害に起因するフェイルオーバーもブロックする設定を行います。ブロックの設定を解除する場合は、次の設定を見直してください。

- ノード障害に起因するフェイルオーバーのブロック設定
  - lkcli server confirmso
  - /etc/default/LifeKeeper の CONFIRMSODEF パラメーター
  - /etc/default/LifeKeeper の CONFIRMSOTO パラメーター
- リソース障害に起因するフェイルオーバーのブロック設定
  - lkcli server block-failover

---

## lkcli mirror

DataKeeper によるミラーリング操作を行います。

ミラーリングの詳細については、「 SIOS DataKeeper for Linux によるミラーリング 」を参照してください。

> **注記:** `lkcli mirror` コマンドは DataKeeper Recovery Kit のインストールが必要になります。

| オプション | デフォルト | 説明 |
|---|---|---|
| --tag <str> | | DataKeeper リソースのタグ名。 |

### lkcli mirror status

ミラーのステータスを表示します。

| 権限 | lkadmin, lkoper, lkguest |
|---|---|

### lkcli mirror resume

ミラーを再開します。

| 権限 | lkadmin |
|---|---|

### lkcli mirror pause

ミラーを一時停止します。

| 権限 | lkadmin |
|---|---|

### lkcli mirror fullresync

全体を再同期します。

| 権限 | lkadmin |
|---|---|

### lkcli mirror force

ミラーリングを強制実行します。

> **注記:** ソース – ターゲット間の同期が不完全な状態でも強制的にミラーが開始されるため、データの一貫性を保証できなくなります。不用意に実行した場合、リプリケーション領域のデータが破損する可能性があります。

| 権限 | lkadmin |
|---|---|

---

## lkcli esxi

VMDK RK を利用するときの ESXi ホストの設定を行います。
VMDK RK の詳細については「 VMDK as Shared Storage Recovery Kit 管理ガイド 」を、ESXi ホストの設定については「 ESXi ホストの登録 」や「 メンテナンス 」を参照してください。

### lkcli esxi add

ESXi ホストの情報を追加します。オプションでユーザー名とパスワードを指定しない場合は対話的にユーザー名･パスワードを入力することになります。

| 権限 | lkadmin |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --host <host> | | 追加する ESXi ホスト名。 |
| [--user <user>] | | ESXi ホストにログインするときのユーザー名。 |
| [--password <password>] | | ESXi ホストにログインするときのパスワード。 |

### lkcli esxi list

登録されている ESXi ホストの一覧を表示します。

| 権限 | lkadmin, lkoper, lkguest |
|---|---|

### lkcli esxi update

登録されている ESXi ホストの情報を更新します。オプションでユーザー名とパスワードを指定しない場合は対話的にユーザー名･パスワードを入力することになります。

| 権限 | lkadmin |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --host <host> | | 更新する ESXi ホスト名。 |
| [--user <user>] | | ESXi ホストにログインするときのユーザー名。 |
| [--password <password>] | | ESXi ホストにログインするときのパスワード。 |

### lkcli esxi delete

登録されている ESXi ホストの情報を削除します。

| 権限 | lkadmin |
|---|---|

| オプション | デフォルト | 説明 |
|---|---|---|
| --host <host> | | 削除する ESXi ホスト名。 |

# 5.6.2.1. ARK 毎のサブコマンド

## ARK 一覧


- Apache
- DataKeeper
- DB2
- EC2
- FileSystem
- Multipath Recovery Kits – DMMP / HDLM / PPATH
- Generic Application
- IP
- LB Health Check
- MySQL
- NFS
- Recovery Kit for Oracle Cloud Infrastructure
- Oracle
- OracleListener
- OraclePDB
- PostgreSQL
- Postfix
- QSP
- Raw
- Route53
- Samba
- SAP
- SAP HANA
- SAP MaxDB
- Sybase
- WebSphere MQ


---

## Apache

リソースの種類は `app: webserver, typ: apache` です。
このコマンドを利用する上での前提条件や詳細な動作については「Apache Recovery Kit 管理ガイド」の「LifeKeeper の設定作業」を参照してください。

### create apache

| オプション | デフォルト | 説明 |
|---|---|---|
| --root <str> | | Apache Web Server のルートディレクトリのフルパス<br>相対パスやシンボリックリンクを使用することはできません。 |
| --path <str> | | Apache Web Server デーモンのフルパス名 (ファイル名を含む) |

### extend apache

オプションなし

### config apache

コンフィグレーションする項目はありません

---

## DataKeeper

リソースの種類は app: scsi, typ: netraid です。
このコマンドを利用する上での前提条件や詳細な動作については「SIOS DataKeeper for Linux」の「リソースの設定作業」を参照してください。

### create dk

| オプション | デフォルト | 説明 |
|---|---|---|
| --mode <str> | | レプリケーションタイプ<br>「synchronous (同期)」または「asynchronous (非同期)」を指定します。<br>「synchronous」を指定すると、すべてのターゲットに対するレプリケーション設定が「synchronous」に限定されます。 |
| [--bitmap <str>] | /opt/LifeKeeper/bitmap_<Tag 名> | インテントログに使用するビットマップファイルのパス。ビットマップファイルは非同期レプリケーションでは必須であり、同期レプリケーションでは使用が推奨されます。詳細については、 DataKeeper リソース階層の作成 を参照してください。 |
| --hierarchy <str> | | 作成するデータレプリケーションのタイプ<br>タイプにより、オプションが異なります。<br>< new \| existing \| dronly > |

#### --hierarchy new

| オプション | デフォルト | 説明 |
|---|---|---|
| --device <str> | | ソースディスクまたはパーティション |
| --fstype <str> | | ファイルシステムタイプ<br>LifeKeeper がサポートするファイルシステムタイプのみ指定できます。 |
| --mount_point <str> | | 新しいファイルシステムの 新しいマウントポイント |
| --fstag <str> | | ファイルシステムリソースのタグ名 |

### --hierarchy existing

| オプション | デフォルト | 説明 |
|---|---|---|
| --mount_point <str> | | プライマリーサーバーの NetRAID デバイスにマウントするマウントポイント |
| --fstag <str> | | ファイルシステムリソースのタグ名 |

### --hierarchy dronly

| オプション | デフォルト | 説明 |
|---|---|---|
| --device <str> | | ソースディスクまたはパーティション |

## extend dk

| オプション | デフォルト | 説明 |
|---|---|---|
| --mode <str> | | レプリケーションタイプ<br>「synchronous」または「asynchronous」を指定します。 |
| --laddr | | ローカル IP アドレス |
| --raddr | | リモート IP アドレス |
| [--bitmap <str>] | /opt/LifeKeeper/bitmap_< Tag 名> | インテントログの記録に使用するビットマップファイルのパス |
| [--device <str>] | 拡張元と同一 | ソースディスクまたはパーティション |
| [--fstag <str>] | | ファイルシステムリソースのタグ名 |

## config dk

| オプション | デフォルト | 説明 |
|---|---|---|
| --resync_speed_min <num> | | 再同期速度制限 (KB/s) の最小値を設定します |
| --resync_speed_max <num> | | 再同期速度制限 (KB/s) の最大値を設定します |
| --compression_level <num> | | ネットワークの圧縮レベル (0～9) を設定します |

---

# DB2

このコマンドを利用する上での前提条件や詳細な動作については「DB2 Recovery Kit 管理ガイド」の「LifeKeeper の設定作業」を参照してください。

### create db2

| オプション | デフォルト | 説明 |
|---|---|---|
| --instance <str> | | DB2 インスタンス<br>保護する DB2 インスタンスを指定します。 |

### extend db2

オプションなし

### config db2

コンフィグレーションする項目はありません

---

## EC2

リソースの種類は app: comm, typ: ec2 です。
このコマンドを利用する上での前提条件や詳細な動作については「Recovery Kit for EC2 管理ガイド」の「設定」を参照してください。

### create ec2

| オプション | デフォルト | 説明 |
|---|---|---|
| --type <str> | | 作成する EC2 リソースの種類の指定します。<br>ルートテーブルシナリオを選択する場合は “RouteTable”、 Elastic IP シナリオを選択する場合は ”ElasticIP ”を指定します。 |

#### --type RouteTable

| オプション | デフォルト | 説明 |
|---|---|---|
| --ip_resource <str> | | 事前に作成した IP リソースのタグ名を指定します。 |

#### --type ElasticIP

| オプション | デフォルト | 説明 |
|---|---|---|
| --eip <str> | | 保護したい Elastic IP の IP アドレス。 |
| --dev <str> | | EIP がアタッチされたネットワークインターフェース名。 |

### extend ec2

オプションなし

### config ec2

コンフィグレーションする項目はありません

---

## FileSystem

このコマンドを利用する上での前提条件や詳細な動作については「ファイルシステムリソース階層の作成」や「ファイルシステムリソース階層の拡張」を参照してください。

### create fs

| オプション | デフォルト | 説明 |
|---|---|---|
| --mountpoint <str> | | ファイルシステムのマウントポイントを指定。 |

### extend fs

| オプション | デフォルト | 説明 |
|---|---|---|
| [--mountpoint <str>] | 拡張元と同一 | ファイルシステムのマウントポイントを指定。 |

### config fs

| オプション | デフォルト | 説明 |
|---|---|---|
| --mountopt <str> | | ファイルシステムのマウントオプションを指定。 |

> ✻ 注意 :config コマンドは、ファイルシステムが稼働中またはマウントされているときに使用できますが、変更されたマウントオプションは、ファイルシステムが再マウントされるまで有効になりません。

## Multipath Recovery Kits – DMMP / HDLM / PPATH

> ✻ 注意 :これらの Recovery kit には lkcli コマンドが直接含まれていませんが、ファイルシステムリカバリーキット または raw リカバリーキット を介して作成、拡張、および削除されます。

---

## Generic Application

リソースの種類は `app: gen, typ: app` です。

このコマンドを利用する上での前提条件や詳細な動作については「Generic Application リソース階層の作成」や「Generic Application リソース階層の拡張」を参照してください。

### create gen

| オプション | デフォルト | 説明 |
|---|---|---|
| --restore <str> | | restore スクリプトのパスを指定。 |
| --remove <str> | | remove スクリプトのパスを指定。 |
| [--quickCheck <str>] | | quickCheck スクリプトのパスを指定。 |
| [--recover <str>] | | recover スクリプトのパスを指定。 |
| [--appinfo <str>] | | アプリケーションに関するオプション情報を指定。 |

### extend gen

| オプション | デフォルト | 説明 |
|---|---|---|
| [--appinfo <str>] | 拡張元と同一 | アプリケーションに関するオプション情報を指定。 |

### config gen

| オプション | デフォルト | 説明 |
|---|---|---|
| [--restore <str>] | | 更新する restore スクリプトのパスを指定。 |
| [--remove <str>] | | 更新する remove スクリプトのパスを指定。 |
| [--quickCheck <str>] | | 更新する quickCheck スクリプトのパスを指定。 |
| [--recover <str>] | | 更新する recover スクリプトのパスを指定。 |
| [--all <str>] | No | **Yes** または **No** を指定。<br>• **Yes**：全てのクラスターノードのスクリプトを更新。<br>• **No**：コマンド実行ノードのスクリプトを更新。 |

---

## IP

リソースの種類は `app: comm, typ: ip` です。
このコマンドを利用する上での前提条件や詳細な動作については「IP Recovery Kit 管理ガイド」の「設定」を参照してください。

### create ip

| オプション | デフォルト | 説明 |
|---|---|---|

| | | |
|---|---|---|
| --ipaddr <str> | | 仮想 IP アドレス<br><br>実 IP アドレスを保護する場合は '0.0.0.0' を指定します |
| [--netmask <str>] | ipaddr から適当と思われるものを補完 | 仮想 IP のネットマスク<br><br>ipaddr で '0.0.0.0' を指定した場合は device で指定されたネットワークインターフェースのネットマスクが使用されます |
| [--device <str>] | ipaddr と netmask から適当と思われるものを補完<br><br>注意：ipaddr で '0.0.0.0' を指定した場合は、device の指定は必須となります | 仮想 IP 、もしくは実 IP と紐づくネットワークインターフェース名 |

## extend ip

| オプション | デフォルト | 説明 |
|---|---|---|
| [--ipaddr <str>] | 拡張元の情報から補完 | 拡張先となるノード上の仮想 IP アドレス<br><br>実 IP アドレスを保護する場合は '0.0.0.0' を指定します |
| [--netmask <str>] | ipaddr から適当と思われるものを補完 | 拡張先となるノード上の仮想 IP のネットマスク<br><br>ipaddr で '0.0.0.0' を指定した場合は device で指定されたネットワークインターフェースのネットマスクが使用されます |
| [--device <str>] | ipaddr と netmask から適当と思われるものを補完<br><br>注意：ipaddr で '0.0.0.0' を指定した場合は、device の指定は必須となります | 拡張先となるノード上の仮想 IP 、もしくは実 IP と紐づくネットワークインターフェース名 |

## config ip

| オプション | デフォルト | 説明 |
|---|---|---|
| [--pinglist <str>] | | オプション用 Ping 先リスト (カンマ区切りで複数指定も可能) |
| [--srcaddr <str>] | | **0** または **1** を指定。<br>仮想 IP アドレスを同一サブネットへの外部通信 IP トラフィックのソースアドレスに使用するかを指定。<br>• **0** : 使用しない。 |

| | | |
|---|---|---|
| | | • **1**：使用する。 |
| [--restoremode <str>] | | **Enabled** または **Disabled** を指定。<br>IP リソースに対するリストアおよびリカバリーを有効･無効に設定。<br>• **Enabled**：リストアおよびリカバリーを有効<br>• **Disabled**：リストアおよびリカバリーを無効 |

---

# LB Health Check

リソースの種類は `app: comm, typ: lbhc` です。
このコマンドを利用する上での前提条件や詳細な動作については「LB Health Check Kit 管理ガイド」を参照してください。

## create lbhc

| オプション | デフォルト | 説明 |
|---|---|---|
| --port <str> | | ロードバランサーからの接続を待ち受けるポート番号を指定します。 |
| [--msg <str>] | なし | デーモンからの応答文字列を指定します。この設定は任意です。 |

## extend lbhc

オプションなし

## config lbhc

| オプション | デフォルト | 説明 |
|---|---|---|
| --port <str> | | ロードバランサーからの接続を待ち受けるポート番号を指定します。 |
| [--msg <str>] | なし | デーモンからの応答文字列を指定します。この設定は任意です。 |

---

# MySQL

リソースの種類は `app: database, typ: mysql` です。
このコマンドを利用する上での前提条件や詳細な動作については「MySQL Recovery Kit 管理ガイド」の「インストール」を参照してください。

## create mysql

| オプション | デフォルト | 説明 |
|---|---|---|

| | | |
|---|---|---|
| --cnf <str> | | MySQL のコンフィグファイルの絶対パス |
| --bin <str> | | MySQL の実行バイナリーのあるディレクトリの絶対パス |
| [--instance <str>] | なし | 保護したい MySQL のインスタンス番号。<br>もし MySQL をシングルインスタンスで使用している場合は、この番号を指定しないでください。 |

### extend mysql

| オプション | デフォルト | 説明 |
|---|---|---|
| [--bin <str>] | | 拡張先となるノード上にある MySQL の実行バイナリーのあるディレクトリの絶対パス |

### config mysql

コンフィグレーションする項目はありません

---

## NFS

リソースの種類は `app: gen, typ: nfs` です。
このコマンドを利用する上での前提条件や詳細な動作については「NFS Recovery Kit 管理ガイド」の「設定作業」を参照してください。

### create nfs

| オプション | デフォルト | 説明 |
|---|---|---|
| --export <str> | | NFS ファイルシステムのエクスポートポイント。 |
| --ip <str> | | NFS ファイルシステムにアクセスするためにクライアントが使用する仮想 IP アドレスに対応する IP リソースのタグ名。 |

### extend nfs

オプションなし

### config nfs

コンフィグレーションする項目はありません

---

# Recovery Kit for Oracle Cloud Infrastructure

リソースの種類は app: comm, typ: ocivip です。
このコマンドを利用する上での前提条件や詳細な動作については「Recovery Kit for Oracle Cloud Infrastructure 管理ガイド」を参照してください。

## create ocivip

| オプション | デフォルト | 説明 |
|---|---|---|
| --ipaddr <str> | なし | セカンダリー・プライベート IP アドレスを指定します。ここで指定した IP アドレスが VNIC に割り当てられます。 |
| --device <str> | なし | IP アドレスを割り当てるネットワークインターフェース名を指定します。 |

## extend ocivip

| | | |
|---|---|---|
| --device <str> | 拡張元ノードで指定しているネットワークインターフェース名 | 拡張先ノードで IP アドレスを割り当てるネットワークインターフェース名を指定します。 |

## config ocivip

ARK 固有のオプションはありません

---

# Oracle

リソースの種類は app: database, typ: oracle です。
このコマンドを利用する上での前提条件や詳細な動作については「Oracle Recovery Kit 管理ガイド」の「LifeKeeper の設定作業」を参照してください。

## create oracle

| オプション | デフォルト | 説明 |
|---|---|---|
| --sid <str> | | データベースの ORACLE_SID。 |
| [--listener <str>] | なし | Oracle リソースに依存して含まれる Oracle Listener リソースのタグ名。 |
| [--user <str>] | なし | Oracle データベースのユーザー名。 |
| [--password <str>] | なし | Oracle データベースのユーザーのパスワード。 |

### extend oracle

オプションなし

### config oracle

| オプション | デフォルト | 説明 |
|---|---|---|
| --user <str> | | Oracle データベースのユーザー名。 |
| --password <str> | | Oracle データベースのユーザーのパスワード。 |
| --role <str> | | ユーザーのロール。<br>sysdba または sysoper を指定してください。 |

---

## OracleListener

リソースの種類は `app: database, typ: listener` です。
このコマンドを利用する上での前提条件や詳細な動作については「Oracle Recovery Kit 管理ガイド」の「複数のリソースに対する共有 Oracle Listener の作成」を参照してください。

### create listener

| オプション | デフォルト | 説明 |
|---|---|---|
| --exe <str> | | Listener の実行パス。 |
| --config <str> | | Listener の実行設定ファイルのパス。 |
| --protection <str> | | Listenerの保護レベル。<br>• **Full** : 開始、停止、監視および回復<br>• **Intermediate** : 開始、監視および回復<br>• **Minimal** : 開始および監視のみ |
| --recovery <str> | | Listener のリカバリーレベル。<br>• **Standard** : 標準の LifeKeeper リカバリーを有効にします。すべての listener 障害がローカルで発生し、必要であれば、有効なバックアップサーバーへのフェイルオーバーを行います。<br>• **Optional** : オプションの LifeKeeper リカバリーを有効にします。すべての listener 障害がローカルで発生しますが、有効なバックアップサーバーへのフェイルオーバーは行いません。 |
| [--user <str>] | なし | システムユーザー名。<br>Listener を開始、停止、監視、および回復する権限を持つシステムユーザーを指定してください。 |

| | | |
|---|---|---|
| [--listener <str>] | LISTENER | 保護する Oracle Listener の名前。 |
| [--iptag <str>] | なし | このリソース階層の依存関係として保護される IP リソースのタグ名。 |

### extend listener

| オプション | デフォルト | 説明 |
|---|---|---|
| [--exe <str>] | 拡張元と同一 | Listener の実行パス。 |
| [--config <str>] | 拡張元と同一 | Listener の実行設定ファイルのパス。 |

### config listener

| オプション | デフォルト | 説明 |
|---|---|---|
| --type <str> | | 変更する項目名。項目によりオプションが異なります。<br>• **ProtectionLevel** : Listener の保護レベル。<br>• **RecoveryLevel** : Listener のリカバリーレベル。<br>• **Listener** : 保護する Oracle Listener の名前。 |

#### --type ProtectionLevel

| オプション | デフォルト | 説明 |
|---|---|---|
| --value <str> | | create の protection オプションと同様。 |

#### --type RecoveryLevel

| オプション | デフォルト | 説明 |
|---|---|---|
| --value <str> | | create の recovery オプションと同様。 |

#### --type Listener

| オプション | デフォルト | 説明 |
|---|---|---|
| --value <str> | | 保護する Oracle Listener の名前。 |
| [--iptag <str>] | なし | 依存関係として保護される IP リソースのタグ名。 |

---

## OraclePDB

このコマンドを利用する上での前提条件や詳細な動作については「Oracle Recovery Kit 管理ガイド」の「Oracle Multitenant 構成による Pluggable Database の設定」を参照してください。

### create pdb

| オプション | デフォルト | 説明 |
|---|---|---|
| --sid <str> | | データベースの ORACLE_SID。 |
| --pdb <str> | | 保護するPDBリスト（カンマ[,]区切りで複数指定できます。） |

### extend pdb

オプションなし

### config pdb

| オプション | デフォルト | 説明 |
|---|---|---|
| --pdb <str> | | 保護するPDBリスト（カンマ[,]区切りで複数指定できます。） |

---

## PostgreSQL

リソースの種類は `app: database, typ: pgsql` です。
このコマンドを利用する上での前提条件や詳細な動作については「PostgreSQL Recovery Kit 管理ガイド」の「インストール」を参照してください。

### create pgsql

| オプション | デフォルト | 説明 |
|---|---|---|
| --datadir <str> | | データベースのデータがあるディレクトリの絶対パス。 |
| --port <num> | | PostgreSQL が使用しているポート番号。 |
| --socket <str> | | PostgreSQL が使用しているソケットのパス。 |
| --dbuser <str> | | PostgreSQL が使用しているユーザー名。 |
| --logfile <str> | | ログを出力先となる絶対パス。 |
| [--exepath <str>] | /usr/bin | 実行ファイルがあるディレクトリの絶対パス。 |
| [--clientexe <str>] | <exepath>/psql | 実行ファイル “psql” の絶対パス。 |
| [--adminexe <str>] | <exepath>/pg_ctl | 実行ファイル “pg_ctl” の絶対パス。 |

### extend pgsql

| オプション | デフォルト | 説明 |
|---|---|---|

| | | |
|---|---|---|
| [--exepath <str>] | 拡張元と同一 | 拡張先となるノード上で実行ファイルがあるディレクトリの絶対パス。指定しなかった場合は、拡張元の設定を引き継ぎます。 |

### config pgsql

| オプション | デフォルト | 説明 |
|---|---|---|
| [--dbuser <str>] | なし | PostgreSQL が使用しているユーザー名。 |

---

## Postfix

このコマンドを利用する上での前提条件や詳細な動作については「Postfix リソース階層の作成」や「Postfix リソース階層の拡張」を参照してください。

### create postfix

| オプション | デフォルト | 説明 |
|---|---|---|
| [--binary <binary location>] | /usr/sbin | postfix コマンドが置かれているディレクトリ。 |
| [--config <config file location>] | /etc/postfix | postfix の設定ファイル (main.cf) が置かれているディレクトリ。 |

### extend postfix

オプションなし

### config postfix

コンフィグレーションする項目はありません

---

## QSP

リソースの種類は `app: gen, typ: qsp` です。
このコマンドを利用する上での前提条件や詳細な動作については「Quick Service Protection (QSP) Recovery Kit」を参照してください。

### create qsp

| オプション | デフォルト | 説明 |
|---|---|---|

| | | |
|---|---|---|
| --service <str> | | 保護したいサービス名。 |
| [--quickCheck <str>] | enable | 監視機能の有効･無効を設定します。<br>有効にする場合は “enable”、無効にする場合は “disable” を指定します。 |

### extend qsp

オプションなし

### config qsp

| オプション | デフォルト | 説明 |
|---|---|---|
| [--quickCheck <str>] | なし | 監視機能の有効･無効。<br>有効にする場合は “enable”、無効にする場合は “disable” を指定します。 |
| [--timeout_restore <num>] | なし | restore のタイムアウトの秒数。0 を指定した場合はタイムアウトしません。 |
| [--timeout_remove <num>] | なし | remove のタイムアウトの秒数。0 を指定した場合はタイムアウトしません。 |
| [--timeout_quickcheck <num>] | なし | quickCheck のタイムアウトの秒数。0 を指定した場合はタイムアウトしません。 |
| [--timeout_recover <num>] | なし | recover のタイムアウトの秒数。0 を指定した場合はタイムアウトしません。 |

---

## Raw

このコマンドを利用する上での前提条件や詳細な動作については「Rawデバイスリソース階層の作成」や「Rawデバイスリソース階層の拡張」を参照してください。

### create raw

| オプション | デフォルト | 説明 |
|---|---|---|
| --partition <str> | | Raw デバイスのパーティション。 |

### extend raw

オプションなし

### config raw

コンフィグレーションする項目はありません

---

## Route53

リソースの種類は app: comm, typ: route53 です。
このコマンドを利用する上での前提条件や詳細な動作については「Route53 Recovery Kit 管理ガイド」の「設定」を参照してください。

### create route53

| オプション | デフォルト | 説明 |
|---|---|---|
| --domain <str> | | 保護したい Route53 に存在するドメイン名。 |
| --hostname <str> | | 保護したいホスト名。 |
| --ip_resource <str> | | 事前に作成した IP リソースのタグ名。 |

### extend route53

オプションなし

### config route53

コンフィグレーションする項目はありません

---

## Samba

このコマンドを利用する上での前提条件や詳細な動作については「Samba リソース階層の作成」や「Samba リソース階層の拡張」を参照してください。

### create samba

| オプション | デフォルト | 説明 |
|---|---|---|
| --config <config file name> | /etc/samba/smb.conf | Samba 設定ファイルの絶対パス名。 |

### extend samba

オプションなし

### config samba

コンフィグレーションする項目はありません

---

## SAP

詳細については SAP Recovery Kit 管理ガイド の設定に関連する項目を参照してください。

### create sap

| オプション | デフォルト | 説明 |
|---|---|---|
| --sid <str> | | SAP インストール時の SID |
| --instance <str> | | SAP インスタンス(例：ASCS10) |
| [--protection_level <FULL\|STANDARD\|BASIC\|MINIMUM>] | FULL | 保護対象の SAP リソースの保護レベル |
| [--recovery_level <FULL\|REMOTE\|LOCAL\|OFF>] | FULL | 保護対象の SAP リソースのリカバリーレベル |
| [--ip <str>] | | 指定した SAP リソースの仮想 IP を保護する LifeKeeper IP リソースタグ。この IP リソースは、SAP HANA リソースが作成されているサーバーで In Service になっている必要があります。 このパラメーターには カンマ区切りされた IP リソースタグが許可されます。この引数により指定された IP リソースは、SAP リソースの子リソースになります。 |
| [--dependent_fs <str>] | | 指定の SAP インスタンスに関連するファイルシステムを保護するファイルシステムリソース (gen:filesys) もしくは NFS (gen:nfs) リソースに対する カンマ区切りされた LifeKeeper リソースタグ (例：/sapmnt,/usr/sap/SID/ASCS10)。全ての指定したリソースは、SAP HANA リソースが作成されているサーバーで In Service になっている必要があります。 この引数により指定されたファイルシステムおよび/もしくは NFS リソースは SAP リソースの子リソースになります。注意: このパラメーターは、 --automate_fs_creation とは併用できません。 |
| [--dependent_cs <str>] | | 作成されている SAP リソースの子リソースとして追加されるセントラルサービスリソース (例：ASCS10) に対する LifeKeeper SAP リソースタグ。指定したセントラルサービスリソースは、SAP HANA リソースが作成されているサーバーで In Service になっている必要があります。 注 |

| | | |
|---|---|---|
| | | 意: このパラメーターは、PAS インスタンスを保護するリソースを作成する時に適用可能である場合があります。詳細は SAP リソース階層の作成 の「プライマリーアプリケーションサーバーリソースの作成」の項目を参照してください。 |
| [--automate_fs_creation] | | SAP リソースの作成中に、LifeKeeper は既存の SAP ディレクトリ構造をスキャンし、検出した共有ファイルシステムマウントを保護するために LifeKeeper リソースの自動作成を試みます。 このプロセス中に作成されたリソースは SAP リソースの子リソースになります。プロセス完了後、階層の状態を確認してください。注意: 複製されたファイルシステムの階層はこのプロセスで自動的に作成されません。このパラメーターは --dependent_fs parameterと併用して使用することはできません。 |

## extend sap

オプションなし

## config sap

| オプション | デフォルト | 説明 |
|---|---|---|
| [--protection_level <FULL\|STANDARD\|BASIC\|MINIMUM>] | | 保護対象の SAP リソースの 保護レベル を更新します。このコマンドは SAP リソースが現在 In Service になっているサーバー上で実行する必要があります。 |
| [--recovery_level <FULL\|REMOTE\|LOCAL\|OFF>] | | 保護対象の SAP リソースの リカバリーレベル を更新します。このコマンドは SAP リソースが現在 In Service になっているサーバー上で実行する必要があります。 |
| [--sshcc_action <start\|stop\|migrate[:<node>]\|maintenance:<enable\|disable\|check>>] | | SAP SIOS HA Cluster Connector (SSHCC) アクションを実行します。このコマンドは SAP リソースが現在 In Service になっているサーバー上で実行する必要があります。このパラメーターの設定可能な値は下記の通りです。: |

| | | |
|---|---|---|
| | | start – HA cluster connector 経由で SAP リソースを In Service にします。<br><br>stop – HA cluster connector 経由で SAP リソースを Out of Service にします。<br><br>migrate[:<node>] – SAP リソースを別のサーバーへ移行します。 オプションとして、対象のマイグレーションサーバーは “:” をマイグレートパラメーターに追加することで指定することが可能です。 (例： lkcli resource config sap —tag SAP-SID_ASCS10 —sshcc_action migrate:node2)。対象サーバーが明示的に提供されない場合は、HA cluster connector が、現在のクラスターの状況をベースに最適な対象サーバーを決定します。<br><br>maintenance:<enable\|disable\|check> – 指定の SAP リソースを含む階層に対して SAP メンテナンスモード の有効化、無効化もしくはステータスの確認を行います。 |

# SAP HANA

詳細については SAP HANA Recovery Kit管理ガイド を参照してください。

## create hana

| オプション | デフォルト | 説明 |
|---|---|---|
| --sid <str> | | SAP HANA インストール時の SID |
| --instance <str> | | SAP HANA インスタンス (例 : HDB00) |
| [--ip <str>] | | SAP HANA データーベースの仮想 IP を保護する LifeKeeper IP リソースタグ。この IP リソースは、SAP HANA リソースが作成されているサーバーで In Service になっている必要があります。 |
| [--set_local_recovery_policy <enable\|disable>] | enable | SAP HANA リソースが作成されているサーバーのローカルリカバリーを有効もしくは無効にするパラメーターで |

| | | |
|---|---|---|
| | | す。 |

### extend hana

| オプション | デフォルト | 説明 |
|---|---|---|
| [--set_local_recovery_policy <enable\|disable>] | enable | SAP HANA リソースが拡張されているサーバーのローカルリカバリーを有効もしくは無効にするパラメーターです。 |

### config hana

| オプション | デフォルト | 説明 |
|---|---|---|
| [--stop_all_dbs] | | 全てのクラスターノードで SAP HANA データベースを停止します。 (例：メンテナンスのため)。このコマンドは HANA リソースが現在 In Service になっているクラスターノード上で実行する必要があります。 |
| [--takeover_with_handshake <target server>] | Local server | 指定されたターゲットサーバーで SAP HANA データベースの “Takeover with handshake” を実行します。SAP HANA 2.0 SPS04 以降の機能で、新しいデータベースホスト上でSAP HANA システムレプリケーションの引き継ぎを行う前にプライマリーデータベースを (完全に停止するのではなく)一時停止することで、スイッチオーバー時のプライマリーデータベースの停止時間を抑えます。 |
| [--set_local_recovery_policy <enable\|disable>] | | SAP HANA リソースが拡張されているサーバーのローカルリカバリーを有効もしくは無効にするパラメーターです。 |

## SAP MaxDB

詳細については SAP MaxDB Recovery Kit 管理ガイド の「リソース設定作業」を参照してください。

### create sapdb

| オプション | デフォルト | 説明 |
|---|---|---|
| [--prog | /etc/opt/sdb にあるパス | SAP MaxDB のプログラムパスです。別のディレクトリパスを入力することもできます。パス名に使用できる有効な文字は、文字、数字および特殊文字 – _ . / です。 |

| | | |
|---|---|---|
| <str>] | | |
| [--sid <str>] | まだ構成されていない1番目のインスタンス | SAP MaxDB のインスタンス ID です。 |
| [--user <str>] | /etc/opt/sdb にあるユーザー | SAP MaxDB コマンドを所有しているか、SAP MaxDB コマンドを実行する権限のあるシステムユーザーです。 このユーザーは、テンプレートサーバー上に存在している必要があります。 |
| [--xuser <str>] | 構成された1番目の XUSER key | XUSER key は、SAP MaxDB ツールで使用するデータベースユーザーのデータを保存するために使用されます。XUSER key は、システムユーザーとデータベースインスタンスの組み合わせに対して一致するサーバー上に構成される必要があります。 |

### extend sapdb

| オプション | デフォルト | 説明 |
|---|---|---|
| [--prog <str>] | ターゲット上の/etc/opt/sdb にあるパス | SAP MaxDB のプログラムパスです。別のディレクトリパスを入力することもできます。パス名に使用できる有効な文字は、文字、数字および特殊文字 – _ . / です。 |
| [--user <str>] | テンプレートサーバー上のこのインスタンスに対して定義されたユーザー | ターゲットサーバー上で SAP MaxDB コマンドを所有しているか、SAP MaxDB コマンドを実行する権限のあるシステムユーザーです。 このユーザーは、ターゲットサーバー上に存在している必要があります。 |
| [--xuser <str>] | テンプレートサーバー上のこのインスタンスに対する XUSER key | XUSER key は、SAP MaxDB ツールで使用するデータベースユーザーのデータを保存するために使用されます。XUSER key は、システムユーザーとデータベースインスタンスの組み合わせに対するターゲットサーバー上に構成される必要があります。 |

### config sapdb

オプションなし

---

## Sybase

このコマンドを利用する上での前提条件や詳細な動作については「Sybase Recovery Kit 管理ガイド」の「インストール」を参照してください。

### create sybase

| オプション | デフォル | 説明 |
|---|---|---|

| | ト | |
|---|---|---|
| --basedir <str> | | Sybaseのインストールディレクトリ |
| --datadir <str> | | Sybaseのインスタンスデータディレクトリ |
| --instance <str> | | データベースインスタンス |
| --user <str> | | データベースのユーザー<br>※指定しない場合、saユーザーとしてデータベースへのアクセス、操作を試みます。 |
| --password <str> | | データベースのユーザーのパスワード |
| --backup <str> | none | Backupサーバー |

### extend sybase

オプションなし

### config sybase

| オプション | デフォルト | 説明 |
|---|---|---|
| --user <str> | | データベースのユーザー |
| --password <str> | | データベースのユーザーのパスワード |
| --backup <str> | 現在の値 | Backupサーバー |
| --monitor <str> | 現在の値 | Monitorサーバー |

## WebSphere MQ

このコマンドを利用する上での前提条件や詳細な動作については WebSphere MQ Recovery Kit 管理ガイド の「設定」セクションを参照してください。

### create mq

| オプション | デフォルト | 説明 |
|---|---|---|
| --qmgr <str> | | キューマネージャーの名前 |
| [--protect_listener <YES\|NO>] | YES | キューマネージャーに関連付けられたMQリスナーを保護するかどうかを指定します。 |
| [--channel <str>] | SYSTEM.DEF.SVRCONN | キューマネージャーの接続チャネル。 |
| [--ip <str>] | | リスナーのIPアドレスを保護するLifeKeeper IPリソースタグ。IPリソースは、MQリソースが作成されているサーバーでIn Serviceになって |

| | | |
|---|---|---|
| | | いる必要があります。この引数で指定されたIPリソースは、MQリソースの子リソースになります。 |

## extend mq

オプションなし

## config mq

詳細については、 リソースプロパティ設定の編集 を参照してください。

| オプション | デフォルト | 説明 |
|---|---|---|
| [--protect_listener <YES\|NO>] | | キューマネージャーに関連付けられたMQリスナーを保護するかどうかを指定します。このパラメーターは、MQリソースがIn Service（ISP）になっているサーバーでのみ変更できます。 |
| [--test_queue <str>] | | PUT/GETヘルスチェックを実行する際に使用するLifeKeeper MQ Recovery Kitのテストキューの名前。値を空の文字列 ( `--test_queue ""` ) に設定して、PUT/GETテストを無効にすることができます。 |
| [--log_level <ERROR\|INFORMATIONAL\|DEBUG\|FINE>] | | MQリソースのログレベル。 |
| [--immediate_stop_timeout <str>] | | キューマネージャーの即時停止を試行する際に使用されるタイムアウト値（単位：秒）。 |
| [--preemptive_stop_timeout <str>] | | キューマネージャーの先制終了を試行するときに使用されるタイムアウト値（単位：秒）。 |
| [--channel <str>] | | キューマネージャーの接続チャネル。このパラメーターは、MQリソースがIn Service（ISP）になっているサーバーでのみ変更できます。 |
| [--cmd_server_protection <Full\|Minimal>] | | MQコマンド・サーバーの保護レベル。 |

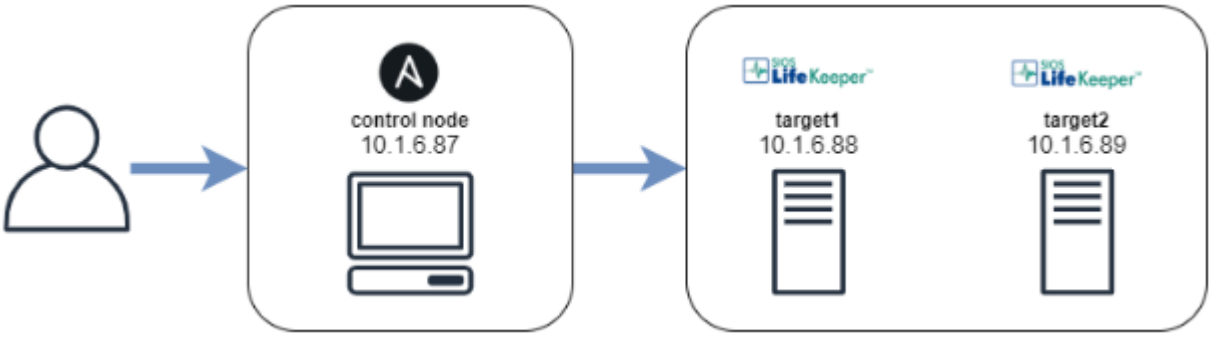
# 5.6.3. Ansible による LifeKeeper のセットアップ

## 概要

本ガイドは、設定ファイルを元に Ansible で、複数ノードに LifeKeeper をインストールする手順を記載しています。
主に、既に Ansible をご利用しているユーザーが、ご自身の環境に LifeKeeper のインストールを追加する際に参照して頂くことを想定しています。
そのため、本ガイドでは、実際の Ansible のタスクのサンプルを記載しています。
お客様自身でプレイブックをお持ちであれば、設定値などは環境に応じて変更してご活用ください。

## 環境

ここでは2ノード構成を作成します。




OS: CentOS 8.0
Python: 3.6.8
Ansible: 2.9.9
LifeKeeper: 9.5.1

## LifeKeeper をインストールするターゲットのセットアップ

### ホスト名と名前解決サービスの設定

LifeKeeper は、クラスター内のターゲットを名前解決できる必要があります。
その為、DNS サービスまたは、/etc/hosts を設定してクラスター間で名前解決できるように設定してください。
本ガイドでは、各 /etc/hosts を設定しています。以下で、設定を変更できます。

**hosts.yml**

```
---
# /etc/hostsにansibleを実行対象のホストをすべて登録する
- name: update /etc/hosts
  lineinfile:
    dest: /etc/hosts
    state: present
    insertafter: EOF
    regexp: "^{{ item.address }}"
    line: "{{ item.address }}\t{{ item.hostname }}"
  with_items:
    - address: "10.1.6.88"
      hostname: "target1"
    - address: "10.1.6.89"
      hostname: "target2"
    - address: "10.1.6.87"
      hostname: "control-node"
```

## ファイアーウォールの設定

ファイアーウォールが有効の場合、各ターゲットでファイアーウォールの設定が必要です。
LifeKeeper では、ターゲット間で TCP の 7365 ポートを利用して通信します。
本ガイドでは、そのポートを、default の public ゾーンに設定しています。

**firewall.yml**

```
---
  - name: enable lifekeeper port
    firewalld:
      zone: public
      port: 7365/tcp
      permanent: yes
      state: enabled
    register: firewalld

  - name: reload
    command: firewall-cmd --reload
    when: firewalld.changed
```

## SELinux の設定

LifeKeeper を使用するには、SELinux を disabled にする必要があります。

以下で、設定を変更できます。

**selinux.yml**

```
---
- name: disable SELinux
  become: yes
  selinux: state=disabled
  register: selinux

- name: reboot a node
  become: yes
  reboot:
  when: selinux.reboot_required
```

---

## LifeKeeper のインストールイメージとライセンスの準備

1. LifeKeeper のインストールイメージとライセンスを取得してください。

   製品購入済みのお客様用
   LifeKeeper 製品のソフトウェアをダウンロード
   https://download.sios.com/

   評価目的のお客様用
   評価版ダウンロード
   https://sios.jp/products/lkdk/product/evaluation_lk.html

## LifeKeeper のインストール用のレスポンスファイルの準備

LifeKeeper をインストールするためには、インストール用のレスポンスファイルが必要です。既にレスポンスファイルをお持ちの方は、この項目をスキップしてください。本ガイドでは、control-nodeマシンで、レスポンスファイルを作成します。

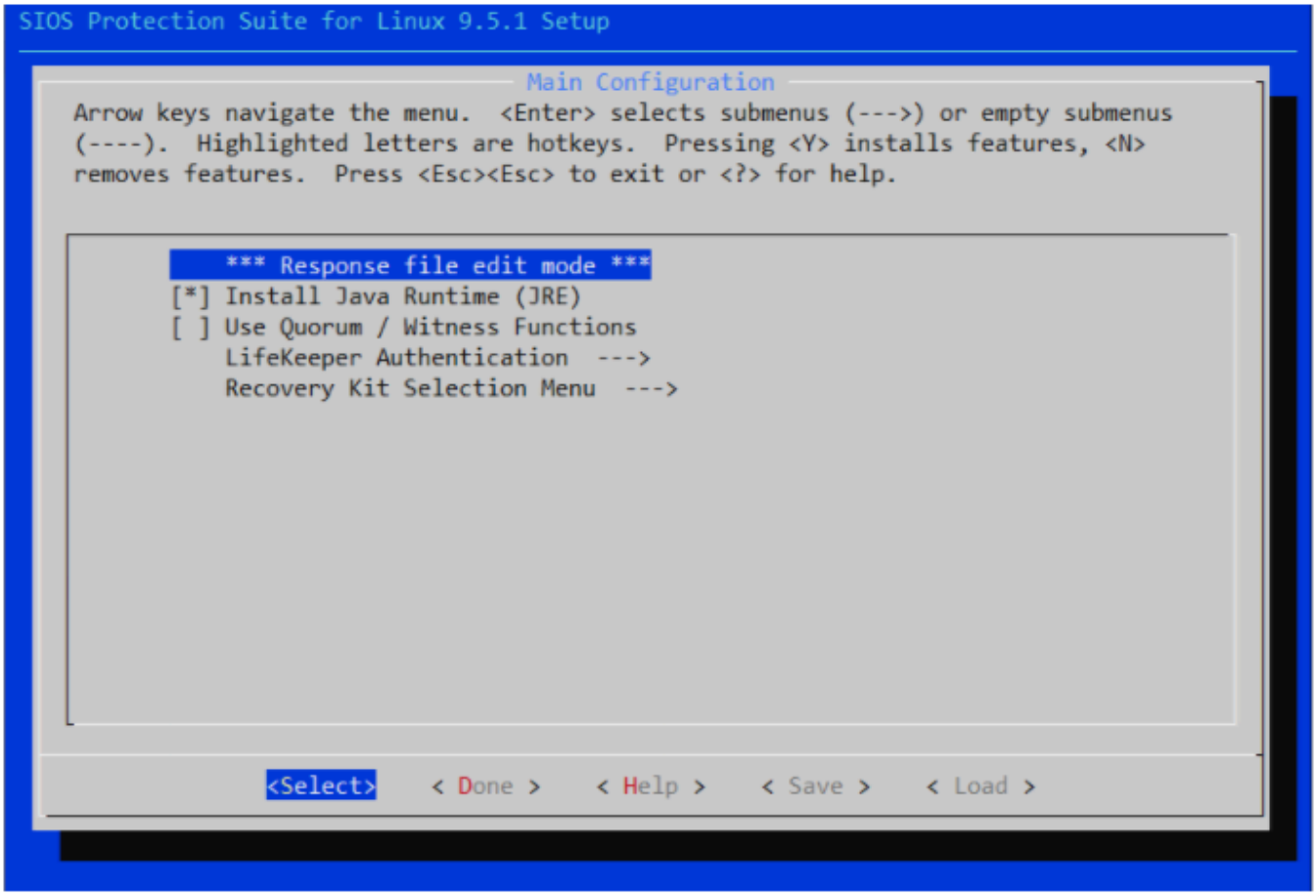
1. 以下のダウンロード先から create_response_file スクリプトを入手してください。

   自動インストール用設定 作成ツール

2. レスポンスファイルのファイル名を指定して、create_response_file スクリプトを実行します。

```
# ./create_response_file /root/lifekeeper/LKCONF
```

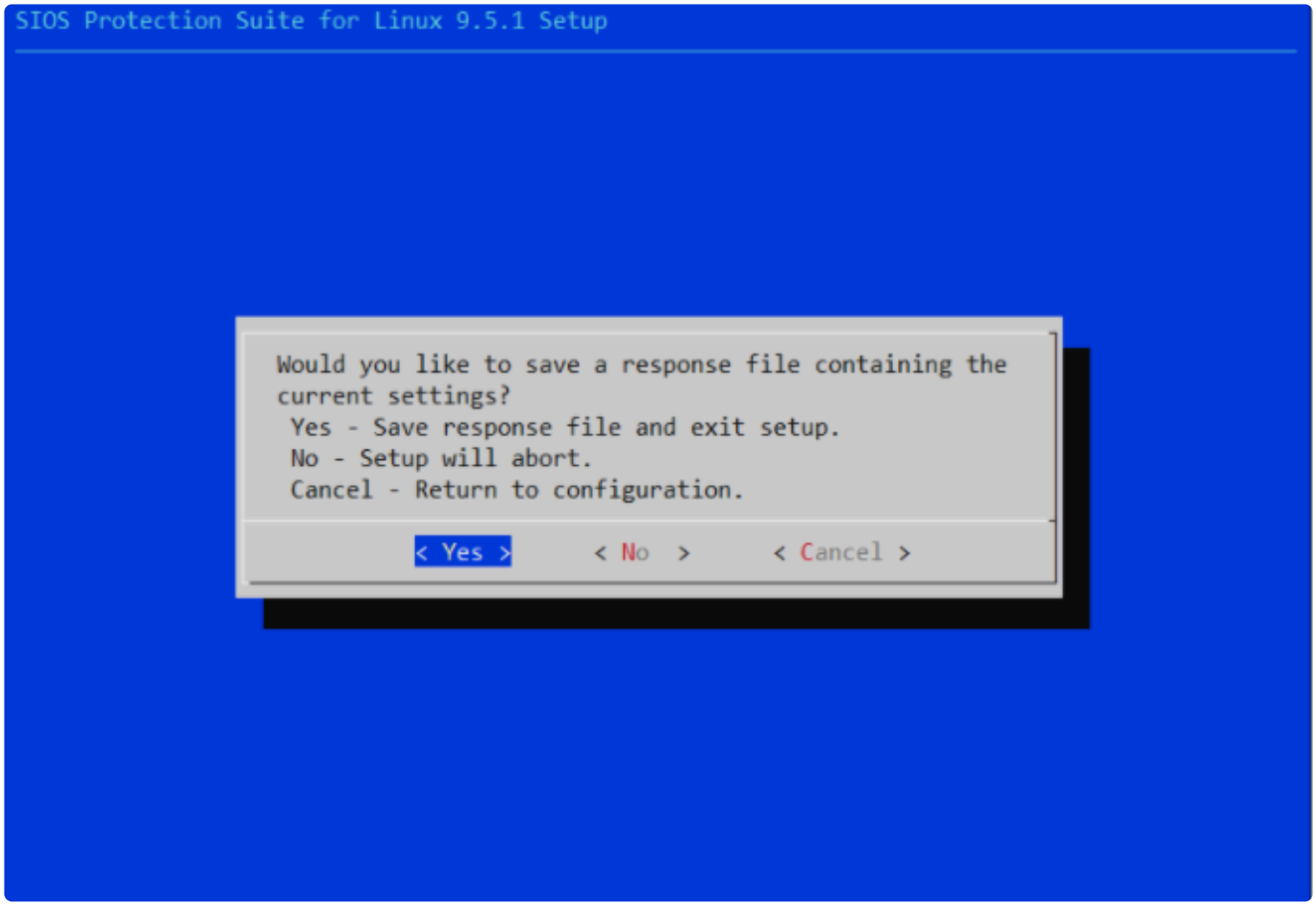
3. create_response_file スクリプトを起動すると、以下のメニュー画面が表示されます。




4. 必要な設定を選択してください。

- Install Java Runtime (JRE) の選択
- Recovery Kit Selection Menu でインストールしたい Recovery Kit の選択

5. 設定後、< Done > を選択してください。その後、< Yes > を選択し終了します。




6. 生成された LKCONF ファイルを確認し、設定が正しいことを確認してください。以下の例は、Oracle、PostgreSQL、MySQL、DataKeeper for Linux、Apache Recovery Kits を選択した場合のレスポンスファイルの例です。

```
# LifeKeeper setup response file
# DO NOT EDIT MANUALLY
LKCONF_INSTALL_JRE="y"
LKCONF_SELONLY="y"
LKCONF_AUTH="y"
LKCONF_LKUSER_lkadmin="root"
LKCONF_steeleye_lkORA="y"
LKCONF_steeleye_lkPGSQL="y"
LKCONF_steeleye_lkSQL="y"
LKCONF_steeleye_lkDR="y"
LKCONF_steeleye_lkAPA="y"
```

## 取得した各ファイルの配布とディレクトリーの指定

1. 取得したインストールイメージとすべてのクラスターノードのライセンスと事前に作成したレスポンスファイルを control-node マシンへ配布してください。以下は、取得した各ファイルのディレクトリ構成例です。

```
/root/lifekeeper/
    ├sps.img  ： LifeKeeper のインストールイメージ
    ├LKCONF   ： LifeKeeper のサイレントインストール用のレスポンスファイル
    └ licenses/
        ├target1/ ： target1 のライセンスファイル群
        └target2/ ： target2 のライセンスファイル群
```

**inventory**

```
[targets]
target1
target2
```

2. 使用するディレクトリーを指定してください。

**defaults/main.yml**

```
---
# control-node側でインストールに必要なファイルを保存しておくディレクトリ
control_node_dir: /root/lifekeeper
# ターゲット側でインストールに一時的に必要なファイルを保存しておくディレクトリ
target_dir: /tmp/lifekeeper
# ターゲット側でインストールイメージのマウントディレクトリ
target_mnt: /mnt
```

以下の処理は各ターゲットで実行されるようにしてください。

## LifeKeeper のインストールイメージとライセンスとレスポンスファイルの配布

**deploy.yml**

```
---
  - name: deploy to lifekeeper install files
    copy:
      src: "{{ control_node_dir }}/"
      dest: "{{ target_dir }}"
```

## LifeKeeperのインストールイメージのマウント

**mount.yml**

```
---
```

```
  - name: mount setup image
    mount:
      path: "{{ target_mnt }}"
      src: "{{ target_dir }}/sps.img"
      fstype: iso9660
      state: mounted
```

## LifeKeeper のインストール

### install.yml

```
---
  - name: install lifekeeper
    shell:
      # setupを直接実行すると失敗します。
      cmd: script -q -e -c "{{ target_mnt | quote }}/setup -f {{ target_dir |
quote }}/LKCONF -q y" {{ target_dir | quote }}/lifekeeper_install.log
```

## LifeKeeper の環境変更の追加

### bash_profile.yml

```
---
  - name: update lifekeeper PATH in bash_profile
    blockinfile:
      path: /root/.bash_profile
      marker: "# {mark} ANSIBLE MANAGED BLOCK: LifeKeeper"
      block: |
        PATH=$PATH:/opt/LifeKeeper/bin
        MANPATH=$MANPATH:/opt/LifeKeeper/man
        export PATH MANPATH
```

## LifeKeeper のライセンスのインストール

### license.yml

```
---
- name: install lifekeeper license
  shell:
```

```
    cmd: |
      for file in `ls`; do
        /opt/LifeKeeper/bin/lkcli license --file $file
      done
    chdir: "{{ target_dir }}/licenses/{{ inventory_hostname }}"
```

## LifeKeeper の起動

### lkstart.yml

```
---
- name: start lifekeeper
  command: /opt/LifeKeeper/bin/lkcli start

- name: lifekeeper status
  command: /opt/LifeKeeper/bin/lkcli status
```

## LifeKeeper のセットアップ後、不要なファイルの削除

### clean.yml

```
---
  - name: unmount a mounted lifekeeper image
    mount:
      path: "{{ target_mnt }}"
      state: absent

  - name: delete lifekeeper files
    file:
      path: "{{ target_dir }}"
      state: absent
```

## Ansible Playbook の例

前のセクションに記載された全てのタスクファイルを作成し、それらのタスクファイルを組み合わせることで以下の例に示すような Playbook を作成できます。

### deploy_lifekeeper.yml

```
---
```

```
- name: deploy and install lifekeeper on all targets
  hosts: targets
  tasks:
    - include_vars: main.yml
    - include_tasks: hosts.yml
    - include_tasks: firewall.yml
    - include_tasks: selinux.yml
    - include_tasks: deploy.yml
    - include_tasks: mount.yml
    - include_tasks: install.yml
    - include_tasks: bash_profile.yml
    - include_tasks: license.yml
    - include_tasks: lkstart.ym l
    - include_tasks: clean.yml
```

この Playbook の例は、下記のコマンドを root として実行するか、既存の Ansible Playbook に追加することができます。

```
# ansible-playbook deploy_lifekeeper.yml -i inventory
```

本ガイドに記載されている全ての Ansible タスクが実行されると、Ansible はインベントリファイルで指定された全てのクラスターノードに LifeKeeer をセットアップすることができます。

ここまでの手順で、LifeKeeper がインストールされたマシンが作成されます。

# 5.6.4. LKCLI ガイド

## 概要

このガイドは、LKCLI (LifeKeeper Command Line Interface) を使用してコマンドラインから LifeKeeper 環境を設定することを支援するために作成されています。LKCLI を使って環境を作成する場合、以下の順序に従ってください。

1. コミュニケーションパスの作成･削除
2. リソースの作成
3. ステータスの確認
4. スイッチオーバーの動作確認
5. メンテナンス
6. クラスターの設定を引き継いで複製する

各LKCLIコマンドの詳細については、 LKCLI (LifeKeeper Command Line Interface) を参照してください。

# 5.6.4.1. コミュニケーションパスの作成･削除

## 概要

### コミュニケーションパスとは？

LifeKeeper では、複数台のノードをお互いに接続することで、クラスターを構成していきます。
もしあるノードで障害が発生しても、クラスター内の別のノード上のリソースに切り替えることで可用性を維持することができます。

そのため、各ノード間で通信を行うための経路を事前に設定する必要があります。
この通信経路のことを LifeKeeper では「コミュニケーションパス」と呼びます。

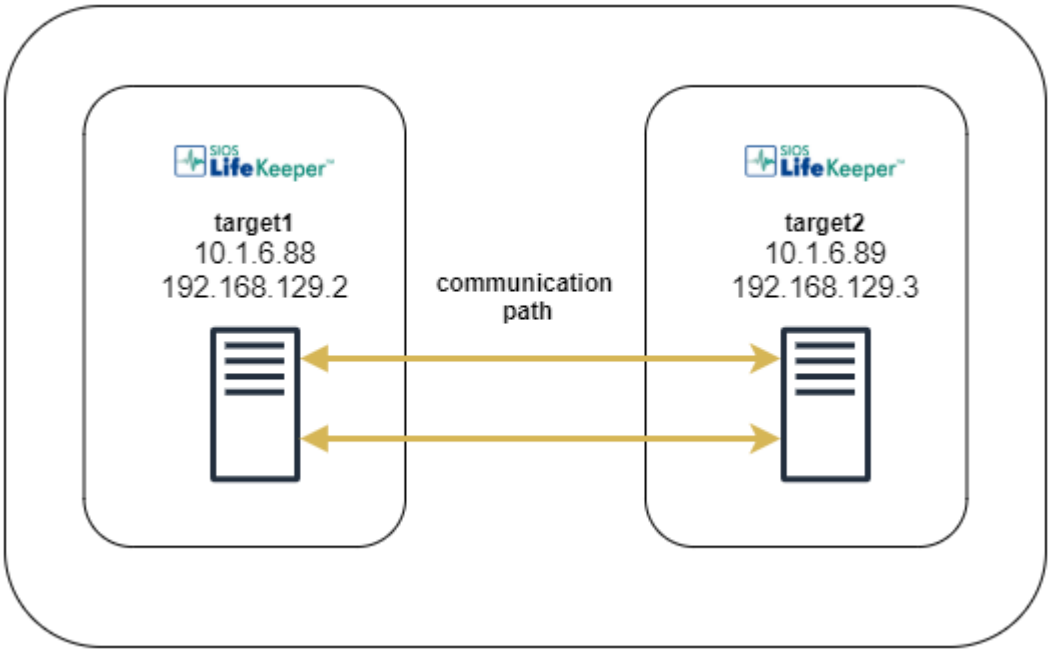
コミュニケーションパスを設定すると、各ノードはお互いに「ハートビート」と呼ばれる、LifeKeeper が正常に動作しているかを示す信号を送信します。
このハートビートが受信できなくなると、ノードに障害があったと判断されます。
もしネットワークに障害が発生した場合、お互い正常であるがハートビートを受け取れなくなるため、LifeKeeper ではすべての2つのノード間を、2本以上のコミュニケーションパスで接続することを推奨しています。

またコミュニケーションパスは、ハートビートのほかに「 LifeKeeper の内部通信」にも利用されます。そのため、コミュニケーションパスが接続されているノード間であれば、リモート先のノード上の LifeKeeper をネットワーク越しに操作することが可能です。

本ガイドでは、2つのノード間で1本のコミュニケーションパスを作成する手順と、そのコミュニケーションパスを削除する手順を説明します。

## 構成

本ガイドでは、LifeKeeper v9.5.0 の使用を想定しています。また、本ガイドに記載されているコマンドなどは以下の図を想定しています。




まずはじめに、コマンドを実行する環境で以下の項目をご確認ください。

- コミュニケーションパスで使用するポート( 7365/tcp ポート)が使用できる
  - ファイアーウォールで通信が許可されている、またはファイアーウォールが無効になっている
- LifeKeeper を起動しているマシンが2台存在する

次に、CLI コマンドの実行に必要な各マシンの情報を、必要であれば記録してください。

1. 各マシンで利用できる IP アドレス

```
# ip address | grep 'inet'
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
    inet 10.1.6.88/16 brd 10.1.255.255 scope global noprefixroute ens192
    inet6 fe80::a633:2758:4976:b42/64 scope link noprefixroute
    inet 192.168.129.2/24 brd 192.168.129.255 scope global noprefixroute ens22
4
    inet6 fe80::34a4:417f:b58b:dc15/64 scope link noprefixroute
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
```

2. 各マシンのホスト名

```
# hostname
target1
```

※ OS によってコマンドが異なる場合があります

# LifeKeeper のコミュニケーションパスの作成

コミュニケーションパスを作成するコマンドは、1度につき「コマンド実行元から指定先への片方向のみ」作成します。そのため、コミュニケーションパスを双方向に通信できるようにするには、2台のマシンそれぞれでコマンドを実行する必要があります。

## target1 で実行

1. コミュニケーションパスが接続されていないことを確認します。

```
[target1]# lkcli status -q
LOCAL    TAG    ID     STATE      PRIO   PRIMARY
```

2. target1 から target2 へのコミュニケーションパスを作成します。

```
[target1]# lkcli commpath create --laddr 10.1.6.88 --raddr 10.1.6.89 --dest ta
rget2
Performing commpath 'target2:10.1.6.88/10.1.6.89' create...
Commpath 'target2:10.1.6.88/10.1.6.89' created successful.
```

### コマンド引数

| 項目 | 入力値 |
|---|---|
| --laddr | 接続元となるローカルノード上の IP アドレス |
| --raddr | 接続先となるリモートノード上の IP アドレス |
| --dest | 接続先となるリモートノードのホスト名 |

3. target2 へのコミュニケーションパスが作成されていることを確認します。
   ※ この段階では片方向の接続のみであるため、STATE は DEAD となります

```
[target1]# lkcli status -q
LOCAL    TAG    ID     STATE      PRIO   PRIMARY

MACHINE  NETWORK ADDRESSES/DEVICE      STATE      PRIO
target2  TCP     10.1.6.88/10.1.6.89   DEAD          1
```

## target2 で実行

4. コミュニケーションパスが接続されていないことを確認します。

```
[target2]# lkcli status -q
LOCAL    TAG    ID     STATE      PRIO   PRIMARY
```

5. target2 から target1 へのコミュニケーションパスを作成します。

```
[target2]# lkcli commpath create --laddr 10.1.6.89 --raddr 10.1.6.88 --dest ta
rget1
Performing commpath 'target1:10.1.6.89/10.1.6.88' create...
Commpath 'target1:10.1.6.89/10.1.6.88' created successful.
```

### コマンド引数

| 項目 | 入力値 |
|---|---|
| --laddr | 接続元となるローカルノード上の IP アドレス |
| --raddr | 接続先となるリモートノード上の IP アドレス |
| --dest | 接続先となるリモートノードのホスト名 |

6. コミュニケーションパスが確立していることを確認します。
   STATE が ALIVE になることを確認します。
   ※ コミュニケーションパス作成後、ALIVE になるまで数秒かかります

```
[target2]# lkcli status -q
LOCAL    TAG    ID     STATE      PRIO   PRIMARY

MACHINE  NETWORK ADDRESSES/DEVICE      STATE      PRIO
target1  TCP     10.1.6.89/10.1.6.88   ALIVE         1
```

# LifeKeeperのコミュニケーションパスの削除

コミュニケーションパスを削除するコマンドは、1度につき「コマンド実行元から指定先への片方向のみ」削除します。そのため、コミュニケーションパスを完全に削除するには、2台のマシンそれぞれでコマンドを実行する必要があります。

## target2 で実行

1. 双方向に接続されたコミュニケーションパスが存在することを確認します。

```
[target2]# lkcli status -q
LOCAL    TAG    ID     STATE      PRIO   PRIMARY
```

```
MACHINE   NETWORK ADDRESSES/DEVICE      STATE     PRIO
target1   TCP     10.1.6.89/10.1.6.88   ALIVE        1
```

2. target2 から target1 へのコミュニケーションパスを削除します。

```
[target2]# lkcli commpath delete --laddr 10.1.6.89 --raddr 10.1.6.88 --dest ta
rget1
```

コマンド引数

| 項目 | 入力値 |
|---|---|
| --laddr | 接続元となるローカルノード上の IP アドレス |
| --raddr | 接続先となるリモートノード上の IP アドレス |
| --dest | 接続先となるリモートノードのホスト名 |

3. target1 へのコミュニケーションパスが削除されたことを確認します。

```
[target2]# lkcli status -q
LOCAL     TAG    ID     STATE      PRIO   PRIMARY
```

## target1 で実行

4. target2へのコミュニケーションパスが残っていることを確認します。

```
[target1]# lkcli status -q
LOCAL     TAG    ID     STATE      PRIO   PRIMARY

MACHINE   NETWORK ADDRESSES/DEVICE      STATE     PRIO
target2   TCP     10.1.6.88/10.1.6.89   DEAD         1
```

5. target2 へのコミュニケーションパスを削除します。

```
[target1]# lkcli commpath delete --laddr 10.1.6.88 --raddr 10.1.6.89 --dest ta
rget2
```

6. target2 へのコミュニケーションパスが削除されたことを確認します。

```
[target1]# lkcli status -q
LOCAL     TAG    ID     STATE      PRIO   PRIMARY
```

以上で、コミュニケーションパスの作成･削除の手順は完了となります。

# 5.6.4.2. リソースの作成

## リソースの作成

保護するサービスやアプリケーションに対応するリソースを作成します。

> **注記:** コミュニケーションパスは「LKCLI のコミュニケーションパスの作成・削除」を参照して、事前に作成してください。

- ファイルシステムリソースの作成
- IP リソースの作成
- PostgreSQL リソースの作成
- DataKeeper リソースの作成

> LKCLI コマンドの一覧は LKCLI（LifeKeeper コマンドラインインターフェース）、LKCLI サブコマンドの一覧については ARK 毎のサブコマンド を参照してください。

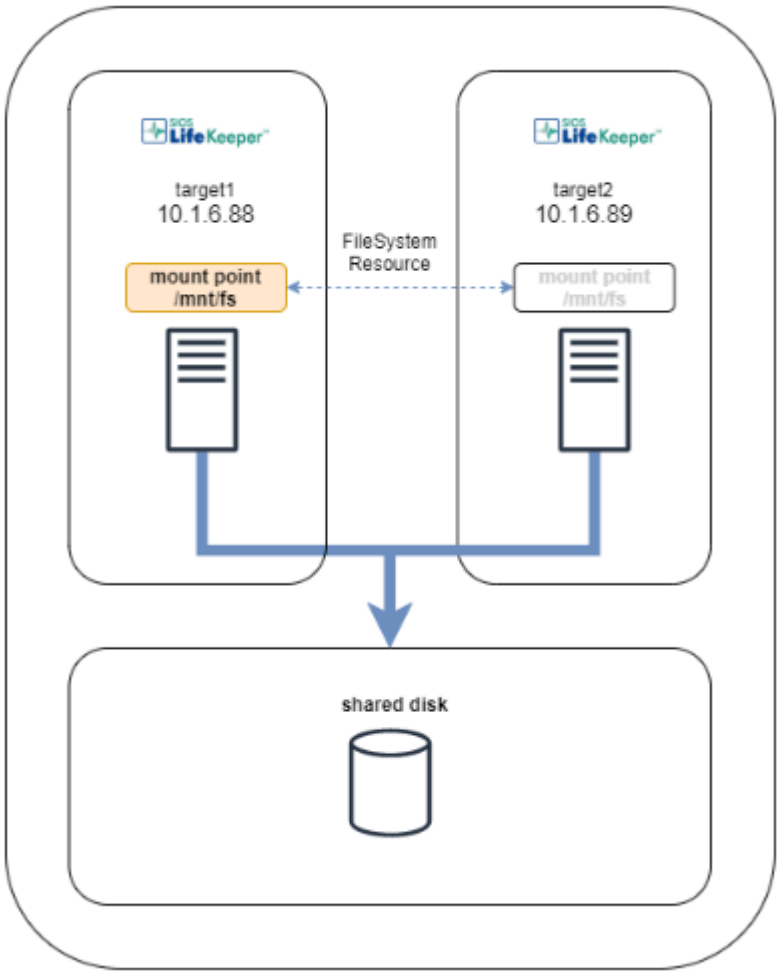
> 各リカバリーキットの管理ガイドについては アプリケーションリカバリーキット を参照してください。

# 5.6.4.2.1. ファイルシステムリソースの作成

ファイルシステムリソースの作成手順を例示します。ファイルシステムリソースは共有ストレージ上のファイルシステムをクラスターノード間で切り替えることを可能する機能を提供します。DataKeeperを使用してレプリケーションファイルシステム階層を作成するには、DataKeeperリソース階層の作成 の情報を参照してください。

## 構成図

本ガイドで作成する環境は、以下のような2ノード構成です。




LifeKeeperの制限事項にあたらないことを確認してください。

ファイルシステムリソースを作成するには、以下の条件を満たしている必要があります。

- 共有ストレージが物理的に接続され、各ノードから同じ状態で接続できること
- partedなどのユーティリティーを使い、 GUIDパーティションテーブル (GPT) で共有ディスクのパーティションが切られていること
- mkfsなどのユーティリティーを使い、ファイルシステムが作成されていること
- 各ノードでファイルシステムをマウント、アンマウントができること

### target1 で実行

1. ファイルシステムのマウント
   ファイルシステムリソース作成対象のファイルシステムを任意のディレクトリーにマウントします。
   例では /dev/sdb1 を /mnt/fs にマウントしています。/mnt/fs がマウントポイントになります。

```
[target1]# df
Filesystem              1K-blocks    Used Available Use% Mounted on
/dev/mapper/centos-root  14034944 6904924   7130020  50% /
devtmpfs                   929204       0    929204   0% /dev
tmpfs                      941312       0    941312   0% /dev/shm
tmpfs                      941312   25948    915364   3% /run
tmpfs                      941312       0    941312   0% /sys/fs/cgroup
/dev/sda1                 1038336  148528    889808  15% /boot
tmpfs                      188264       0    188264   0% /run/user/0
/dev/sdb1                 1044132   32992   1011140   4% /mnt/fs
```

2. リソースの作成
   以下のコマンドを実行します。

```
[target1]# lkcli resource create fs --tag fs-tag --mountpoint /mnt/fs
```

#### リソースの設定値

| 項目 | 入力値 |
|---|---|
| --tag | タグ名 |
| --mountpoint | マウントポイント |

3. リソースの拡張
   さらに以下のコマンドを実行します。

```
[target1]# lkcli resource extend fs --tag fs-tag --dest target2
```

#### リソースの設定値

| 項目 | 入力値 |
|---|---|
| --tag | 作成したリソースのタグ名 |
| --dest | バックアップノード名 |

4. リソースの確認
   リソースの作成および拡張が完了したら、次のコマンドを実行してください。
   リソースの情報が表示されます。

```
[target1]# lkcli status -q

LOCAL   TAG           ID                                      STATE      PRIO  PRI
MARY
target1 fs-tag        /mnt/fs                                 ISP           1  tar
get1
target1  device28856  36000c292eb0c693b2efb44ed56556636-1  ISP           1  tar
get1
target1   disk28786   36000c292eb0c693b2efb44ed56556636    ISP           1  tar
get1
```

ファイルシステムリソースを作成すると上記のように自動的に3つのリソースが依存関係を持った状態で作成されます。これらは、上位からマウントポイント、デバイス、ディスクを管理しており、それぞれ制御する対象が異なっています。これらは 3 つで 1 つの役割を担うため、依存関係の操作などで順序の変更などを行わないようにしてください。

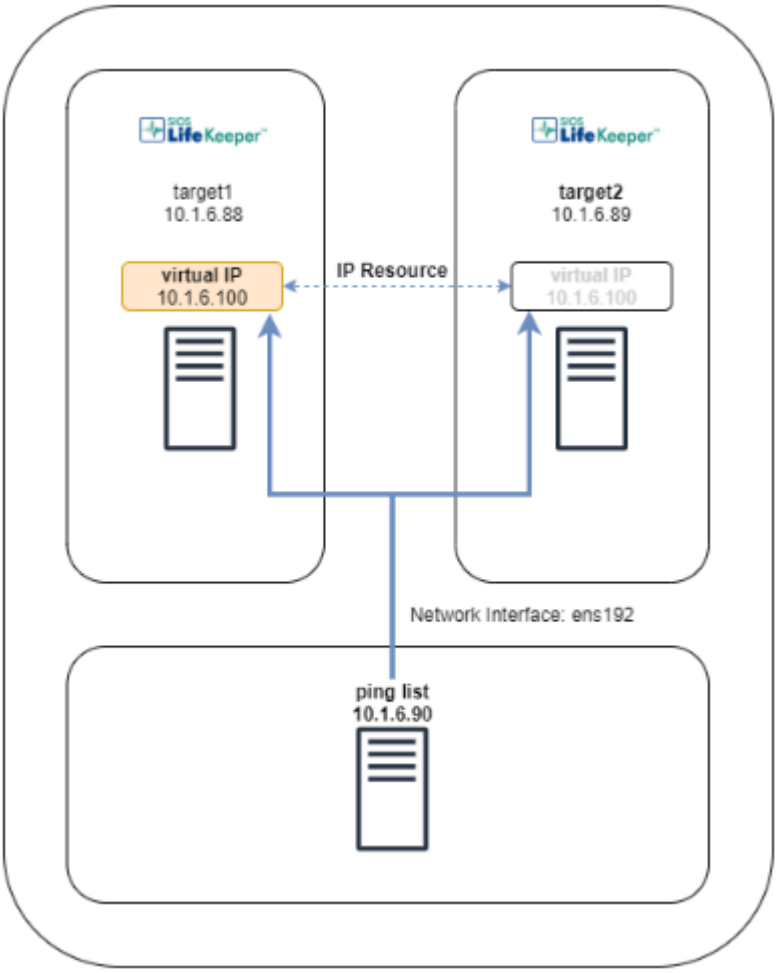
---

# 5.6.4.2.2. IP リソースの作成

IP リソースの作成手順を例示します。
IP リソースはクラスター間で切り替えることができる仮想 IP アドレスを作成し、それを保護します。

### 構成図

本ガイドで作成する環境は、以下のような2ノード構成です。




Ping で応答できる仮想 IP アドレス(上記では 10.1.6.100)を用意してください。
Ping 応答できるマシン(上記では 10.1.6.90)も用意してください。

Ping 応答は以下のようなコマンドで確認できます。

```
# ip -4 addr add 10.1.6.100/24 dev ens192

# ping 10.1.6.90 -b -c 1 -w 1 -I 10.1.6.100 10.1.6.90
```

結果に応答があること。

```
# ip -4 addr delete 10.1.6.100/24 dev ens192
```

### 制限事項

- 作成しようとしている仮想IPアドレスに重複は無いか確認してください。
- 仮想 IP アドレスと同じネットワークに、クラスターを構成するノード以外で Ping 応答できるマシンが存在するか確認してください。
  ※ IP リソースはブロードキャスト Ping、あるいはユニキャスト Ping を使用して応答確認を行います。そのため、クラスターを構成しているノード以外で Ping に応答できるマシンが必要です。

### target1 で実行

1. リソースの作成
   以下のコマンドを実行します。

```
[target1]# lkcli resource create ip --tag ip-tag --ipaddr 10.1.6.100
```

#### リソースの設定値

| 項目 | 入力値 |
|---|---|
| --tag | タグ名 |
| --ipaddr | 仮想 IP アドレス |

2. Ping リストの設定
   以下のコマンドを実行して、Ping リストを設定します。

```
[target1]# lkcli resource config ip --tag ip-tag --pinglist 10.1.6.90
```

その後、リソースを In Service の状態にします。

```
[target1]# lkcli resource restore --tag ip-tag
```

3. リソースの拡張
   さらに以下のコマンドを実行します。

```
[target1]# lkcli resource extend ip --tag ip-tag --dest target2
```

#### リソースの設定値

| 項目 | 入力値 |
|---|---|
| --tag | 作成したリソースのタグ名 |
| --dest | バックアップノード名 |

4. 拡張したリソースへ Ping リストの設定
   以下のコマンドを実行して、拡張したリソースに対しても、Ping リストを設定します。

```
[target1]# lkcli resource config ip --tag ip-tag --pinglist 10.1.6.90 --remot
e target2
```

5. リソースの確認
   リソースの作成および拡張が完了したら、次のコマンドを実行してください。
   リソースの情報が表示されます。

```
[target1]# lkcli status -q
LOCAL    TAG     ID             STATE     PRIO  PRIMARY
target1  ip-tag  IP-10.1.6.100  ISP          1  target1
```

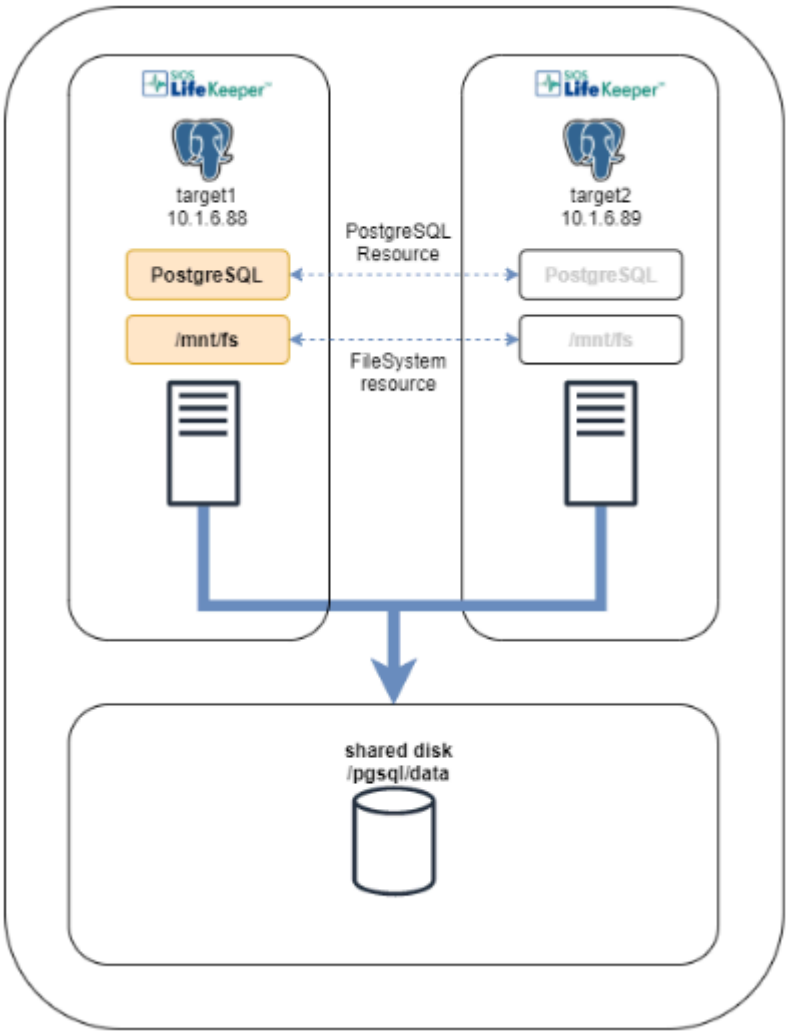
---

# 5.6.4.2.3. PostgreSQL リソースの作成

PostgreSQL リソースの作成手順を例示します。
PostgreSQL リソースは、LifeKeeper ノード上の PostgreSQL インスタンスを保護する機能を提供します。

## 構成図

本ガイドで作成する環境は、以下のような2ノード構成です。
事前に PostgreSQL ARK をインストールしてください。




本ガイドでの環境 (PostgreSQL)

- 本ガイドでは、PostgreSQL 9.2.24 を使用します。
- PostgreSQL のデータディレクトリーは、共有ストレージ上のファイルシステムに作成します。
- PostgreSQL データベース管理者ユーザーは、データベースの初期化時に作成される `postgres` を使用します。
- アクティブ / スタンバイ設定で、PostgreSQL リソースインスタンスを保護します。

※ ファイルシステムの制限事項にあたらないことを確認してください。詳細は こちら をご参照ください。

### target1、target2 で実行

1. PostgreSQL のインストール
   インストール後、PostgreSQL サービスの自動起動は、次のようにして無効にしてください。

```
# systemctl disable postgresql.service
```

### target1 で実行

2. ファイルシステムのマウント
   「ファイルシステムリソースの作成」の1. を参照して、データディレクトリーの作成先となるファイルシステムをマウントしてください。

3. データディレクトリーの作成
   PostgreSQL のデータディレクトリーを共有ディスク上に作成してください。

```
[target1]# mkdir -p /mnt/fs/pgsql/data
[target1]# chown -R postgres:postgres /mnt/fs/pgsql
```

   /usr/lib/systemd/system/postgresql.service ファイルに記載されている次のパスを、共有ディスク上のデータディレクトリーに変更してください。
   Environment=PGDATA=/mnt/fs/pgsql/data

4. データベースの初期化
   次のコマンドを実行し、データベースを初期化してください。
   データディレクトリーの下にデータベースが作成されます。

```
[target1]# postgresql-setup initdb
```

5. PostgreSQL サービスの起動
   以下のコマンドを実行し、PostgreSQL サービスを起動します。

```
[target1]# systemctl start postgresql.service
```

6. リソースの作成
   以下のコマンドを実行します。

```
[target1]# lkcli resource create pgsql --tag pgsql-tag --datadir /mnt/fs/pgsql/data --port 5432 --socket /tmp/.s.PGSQL.5432 --dbuser postgres --logfile /tmp/pgsql-5432.lk.log
```

### リソースの設定値

| 項目 | 入力値 |
|---|---|
| --tag | タグ名 |

| | |
|---|---|
| --datadir | PostgreSQL データベースのデータがあるディレクトリーの絶対パス。 |
| --port | PostgreSQL が使用しているポート番号 |
| --socket | PostgreSQL が使用しているソケットのパス |
| --dbuser | PostgreSQL データベース管理者ユーザー名 |
| --logfile | PostgreSQL の起動や停止に使用する pg_ctl のログファイルの絶対パス。 |

7. リソースの拡張
   さらに以下のコマンドを実行します。

```
[target1]# lkcli resource extend pgsql --tag pgsql-tag --dest target2
```

### リソースの設定値

| 項目 | 入力値 |
|---|---|
| --tag | 作成したリソースのタグ名 |
| --dest | バックアップノード名 |

8. リソースの確認
   リソースの作成および拡張が完了したら、次のコマンドを実行してください。
   リソースの情報が表示されます。

```
[target1]# lkcli status -q
LOCAL    TAG            ID                                  STATE     PRIO  P
RIMARY
target1  pgsql-tag      target1.pgsql-5432                  ISP          1  t
arget1
target1   /mnt/fs       /mnt/fs                             ISP          1  t
arget1
target1    device17885  36000c292eb0c693b2efb44ed56556636-1  ISP          1  t
arget1
target1     disk17816   36000c292eb0c693b2efb44ed56556636    ISP          1  t
arget1
```

PostgreSQL リソースを作成すると、上記のように自動的に下位にファイルシステムリソースが依存関係を持った状態で作成されます。

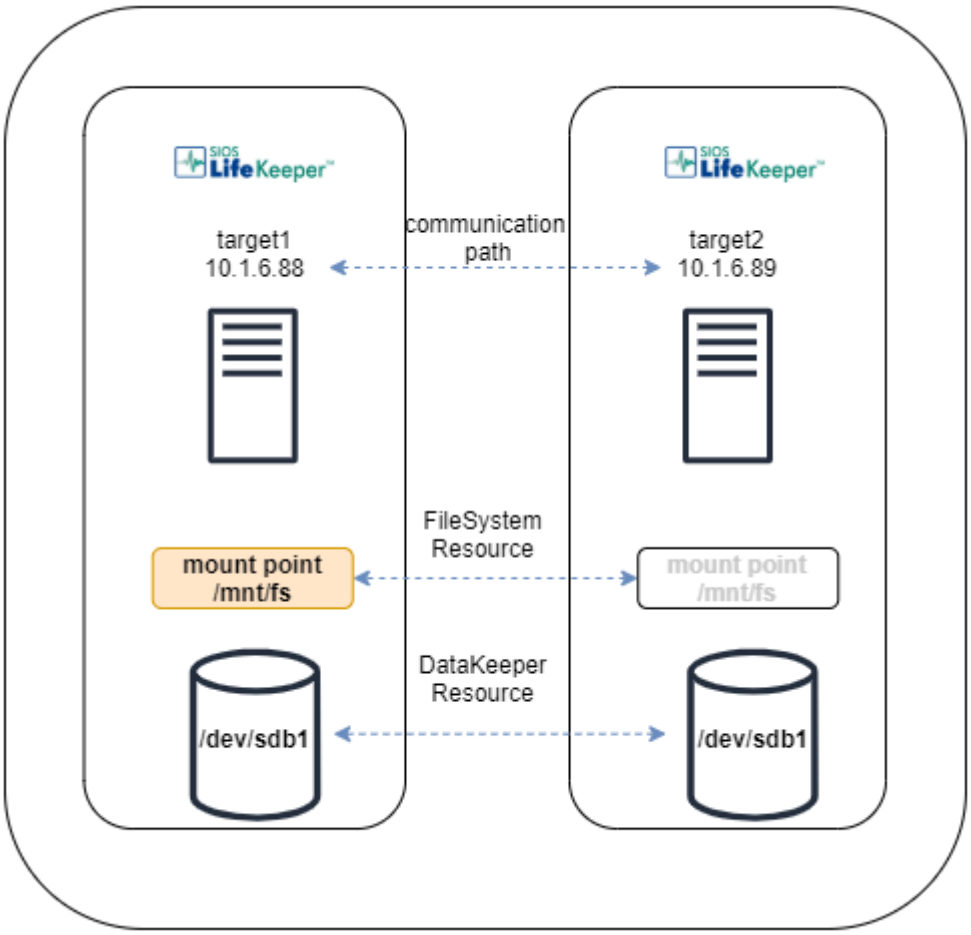
---

# 5.6.4.2.4. DataKeeper リソースの作成

DataKeeper リソースの作成手順を例示します。
DataKeeper リソースは、共有ストレージを使用せずに可用性の高いクラスタを構築する機能を提供します。

### 構成図

本ガイドで作成する環境は、以下のような 2ノード構成です。
新しいファイルシステムにミラーを作成し、LifeKeeper で保護します。




構成が以下の DataKeeper の要件を満たしていることを確認してください。
ハードウェアとソフトウェアの要件

### target1、target2 で実行

1. デバイスの確認

```
# parted /dev/sdb print
Model: VMware Virtual disk (scsi)
Disk /dev/sdb: 1074MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
```

```
Disk Flags:

Number  Start  End  Size  File system  Name  Flags
```

2. パーティションの作成
   デバイスの /dev/sdb にパーティションを作成します。

```
# parted -s /dev/sdb mklabel gpt
# parted /dev/sdb mkpart primary 0% 100%
# parted /dev/sdb print
Model: VMware Virtual disk (scsi)
Disk /dev/sdb: 1074MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name     Flags
 1      1049kB  1073MB  1072MB  ext4         primary
```

### target1 で実行

1. リソースの作成
   以下のコマンドを実行します。

```
[target1]# lkcli resource create dk --tag dk-tag --mode synchronous --hierarch
y new --device /dev/sdb1 --fstype xfs --mount_point /mnt/dk --fstag fs-tag
```

### リソースの設定値

| 項目 | 入力値 |
|---|---|
| --tag | DataKeeper リソースのタグ名 |
| --mode | レプリケーションタイプ |
| --hierarchy | 作成するデータレプリケーションのタイプ |
| --device | ソースディスクまたはパーティション |
| --fstype | ファイルシステムタイプ |
| --mount_point | マウントポイント |
| --fstag | ファイルシステムリソースのタグ名 |

2. DataKeeper リソースの拡張
   さらに以下のコマンドを実行します。

```
[target1]# lkcli resource extend dk --tag dk-tag --dest target2 --mode synchro
nous --laddr 10.1.6.88 --raddr 10.1.6.89
```

### リソースの設定値

| 項目 | 入力値 |
|---|---|
| --tag | 作成した DataKeeper リソースのタグ名 |
| --dest | リソース階層を拡張するターゲットサーバのホスト名 |
| --mode | レプリケーションタイプ |
| --laddr | コミュニケーションパスに設定するローカルマシン上の IP アドレス |
| --raddr | コミュニケーションパスに設定するリモートマシン上の IP アドレス |

3. ファイルシステムリソースの拡張
   さらに以下のコマンドを実行します。

```
[target1]# lkcli resource extend fs --tag fs-tag --dest target2
```

### リソースの設定値

| 項目 | 入力値 |
|---|---|
| --tag | 作成したファイルシステムリソースのタグ名 |
| --dest | リソース階層を拡張するターゲットサーバのホスト名 |

4. リソースの確認
   リソースの作成および拡張が完了したら、次のコマンドを実行してください。
   リソースの情報が表示されます。

```
[target1]# lkcli status -q

LOCAL   TAG           ID                                  STATE     PRIO  PRI
MARY
target1 fs-tag        /mnt/dk                             ISP          1  tar
get1
target1  dk-tag       36000c292eb0c693b2efb44ed56556636-1  ISP          1  tar
get1
```

   上記のように自動的に上位にファイルシステムリソースが依存関係を持った状態でDataKeeperリソースが作成されます。

5. ミラーリングの確認
   DataKeeper リソースを作成すると、全同期が実行されますので、全同期が完了するまで待ってください。以下のコマンドで状況が確認できます。Status が Fully Operational になっていれば全同期は

完了です。

```
[target1]# lkcli mirror status --tag dk-tag
```

### リソースの設定値

| 項目 | 入力値 |
|---|---|
| --tag | 作成した DataKeeper リソースのタグ名 |

### DataKeeper リソースの export/import

クラスターの設定を引き継いで複製する を参考にして DataKeeper とファイルシステムリソースを複製してください。
なお、import については、各ノードでファイルシステムをアンマウントした状態で実行してください。

# 5.6.4.3. ステータスを確認するには？

## ステータスを確認するには？

現在の LifeKeeper の状態を確認するには、 `lkcli status` を使用します。
表示されるメッセージには、実行したノード内のリソースやコミュニケーションパスのステータスが含まれています。

多くのケースでは、一部メッセージが省略された `lkcli status -q` でも、十分な情報を得られます。`q` オプションを指定した場合、上のブロックに現在のリソースの情報、下のブロックに現在のコミュニケーションパスの情報が表示されます。

```
# lkcli status -q
LOCAL    TAG             ID              STATE     PRIO  PRIMARY
target1  ip-10.1.6.100  ip-10.1.6.100  ISP          1  target1

MACHINE  NETWORK ADDRESSES/DEVICE     STATE     PRIO
target2  TCP     10.1.6.88/10.1.6.89  ALIVE        1
```

--remote オプションを利用することで、各ノードにログインすることなく、リソースのステータスを確認できます。その他、オプションや表示されるメッセージなどの詳細につきましては LifeKeeperCLI のマニュアル をご参照ください。

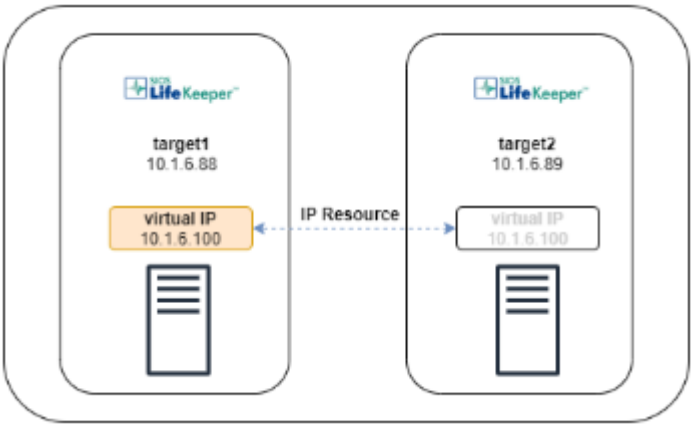
# 5.6.4.4. スイッチオーバーの動作確認

## スイッチオーバーの動作確認

ここでは target1 から target2 へスイッチオーバーを実施する手順を記載します。

### 構成

本ガイドでは LifeKeeper v9.5.0 を使用し、構成は以下の2ノードを想定しています。




コマンドは **target2** で実行してください。

1. target1 上でスイッチオーバーしたいリソースのステータスが ISP であることを確認します。

```
[target2]# lkcli status -q --remote target1
LOCAL    TAG             ID              STATE     PRIO  PRIMARY
target1  ip-10.1.6.100  ip-10.1.6.100  ISP          1  target1

MACHINE  NETWORK ADDRESSES/DEVICE     STATE     PRIO
target2  TCP     10.1.6.88/10.1.6.89  ALIVE        1
```

2. target2 上でスイッチオーバーしたいリソースのステータスが OSU であることを確認します。その際、そのリソースのタグ名をメモしておきます。

```
[target2]# lkcli status -q
LOCAL    TAG             ID              STATE     PRIO  PRIMARY
target2  ip-10.1.6.100  ip-10.1.6.100  OSU         10  target1

MACHINE  NETWORK ADDRESSES/DEVICE     STATE     PRIO
target1  TCP     10.1.6.89/10.1.6.88  ALIVE        1
```

3. target2 にリソースを切り替えます。

```
[target2]# lkcli resource restore --tag ip-10.1.6.100
BEGIN restore of "ip-10.1.6.100"
END successful restore of "ip-10.1.6.100"
```

4. target2 上でリソースが ISP であることを確認します。

```
[target2]# lkcli status -q
LOCAL    TAG            ID             STATE     PRIO  PRIMARY
target2  ip-10.1.6.100  ip-10.1.6.100  ISP         10  target1

MACHINE  NETWORK ADDRESSES/DEVICE     STATE     PRIO
target1  TCP     10.1.6.89/10.1.6.88  ALIVE        1
```

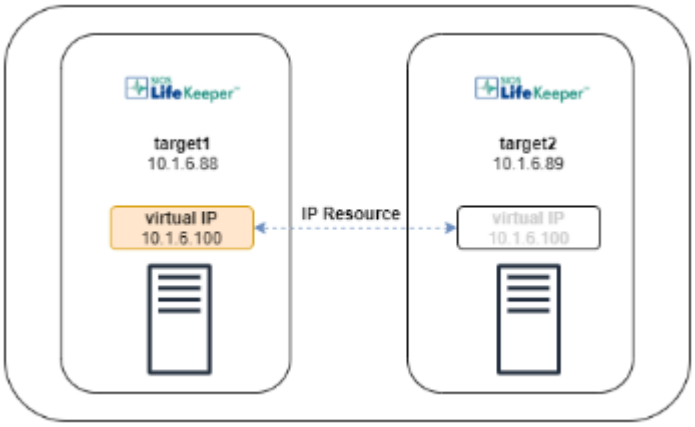
# 5.6.4.5. メンテナンス

## メンテナンス

ここでは、LifeKeeper で保護しているマシンやリソースでメンテナンスを実施する場合の手順を説明します。

### 構成

本ガイドでは LifeKeeper v9.5.0 を使用し、構成は以下の2ノードを想定しています。




## LifeKeeper で保護されているマシンのメンテナンス

ここでは、LifeKeeper が動作しているマシンのシャットダウンなど、LifeKeeper 本体・マシン内のすべてのリソースに影響を与えるメンテナンス作業を、target1 で実施する手順を記載します。

コマンドは **target1** で実行してください。

1. target1 でアクティブになっているリソースを、target2 に切り替えます。

    i. target1 ですべてのリソースのステータスを確認します。もし STATE が ISP になっているリソースがある場合は、そのリソースのタグ名をメモします。

```
[target1]# lkcli status -q
LOCAL    TAG             ID              STATE      PRIO  PRIMARY
target1  ip-10.1.6.100  ip-10.1.6.100  ISP           1  target1

MACHINE  NETWORK ADDRESSES/DEVICE     STATE      PRIO
target2  TCP     10.1.6.88/10.1.6.89  ALIVE         1
```

    ii. target1 で ISP になっているリソースをひとつずつ target2 に切り替えます。なお、—remote オプションを使うことで、target1 上からコマンドを実行できます。

```
[target1]# lkcli resource restore --tag ip-10.1.6.100 --remote target2
BEGIN restore of "ip-10.1.6.100"
END successful restore of "ip-10.1.6.100"
```

iii. target1で、すべてのリソースのステータスを再確認します。
すべてのリソースで STATE が OSU になっていることを確認してください。

```
[target1]# lkcli status -q
LOCAL    TAG            ID             STATE     PRIO  PRIMARY
target1  ip-10.1.6.100  ip-10.1.6.100  OSU          1  target1

MACHINE  NETWORK ADDRESSES/DEVICE     STATE     PRIO
target2  TCP     10.1.6.88/10.1.6.89  ALIVE        1
```

2. target1 の LifeKeeper を停止します。-fオプションをつけることで、LifeKeeper 停止時に行われる、リソースの停止処理をスキップします。

```
[target1]# lkcli stop -f
Removed /etc/systemd/system/lifekeeper-graphical.target.requires/lifekeeper.service.
Removed    /etc/systemd/system/lifekeeper-multi-user.target.requires/lifekeeper.service.
```

3. target1 でメンテナンスを実施してください。

4. target1 の LifeKeeper を起動します。

```
[target1]# lkcli start
Created symlink /etc/systemd/system/lifekeeper-graphical.target.requires/lifekeeper.service → /usr/lib/systemd/system/lifekeeper.service.
Created symlink /etc/systemd/system/lifekeeper-multi-user.target.requires/lifekeeper.service → /usr/lib/systemd/system/lifekeeper.service.
```

5. 必要に応じて target1 でリソースを ISP にします。

```
[target1]# lkcli resource restore --tag ip-10.1.6.100
BEGIN restore of "ip-10.1.6.100"
END successful restore of "ip-10.1.6.100"
```

参考資料： LifeKeeper が保護するシステムのメンテナンス

## LifeKeeper で保護されているリソースのメンテナンス

ここでは、マシン内の特定のリソースのみのメンテナンス作業を実施する場合の手順を記載します。

コマンドは **target1** で実行してください。

1. target1 でアクティブになっているリソースを、target2 に切り替えます。

   i. target1 ですべてのリソースのステータスを確認します。もし STATE が ISP になっているリソースがある場合は、ステップ ii 以降の手順でリソースをスイッチオーバーしてください。

```
[target1]# lkcli status -q
LOCAL    TAG            ID             STATE     PRIO  PRIMARY
target1  ip-10.1.6.100  ip-10.1.6.100  ISP          1  target1

MACHINE  NETWORK ADDRESSES/DEVICE     STATE     PRIO
target2  TCP     10.1.6.88/10.1.6.89  ALIVE        1
```

   ii. target1 で ISP になっているリソースについて、target2 でのタグ名を確認します。

```
[target1]# lkcli status -q --remote target2
LOCAL    TAG            ID             STATE     PRIO  PRIMARY
target2  ip-10.1.6.100  ip-10.1.6.100  OSU         10  target1

MACHINE  NETWORK ADDRESSES/DEVICE     STATE     PRIO
target1  TCP     10.1.6.89/10.1.6.88  ALIVE        1
```

   iii. target1 で ISP になっているリソースをひとつずつ target2 に切り替えます。

```
[target1]# lkcli resource restore --tag ip-10.1.6.100 --remote target2
BEGIN restore of "ip-10.1.6.100"
END successful restore of "ip-10.1.6.100"
```

   iv. target1 で、すべてのリソースのステータスを再確認します。
   すべてのリソースで STATE が OSU になっていることを確認してください。

```
[target1]# lkcli status -q
LOCAL    TAG            ID             STATE     PRIO  PRIMARY
target1  ip-10.1.6.100  ip-10.1.6.100  OSU          1  target1

MACHINE  NETWORK ADDRESSES/DEVICE     STATE     PRIO
target2  TCP     10.1.6.88/10.1.6.89  ALIVE        1
```

2. OSU にしたリソースに対してメンテナンスを実施してください。

3. 必要に応じて target1 でリソースを ISP にします。

```
[target1]# lkcli resource restore --tag ip-10.1.6.100
BEGIN restore of "ip-10.1.6.100"
END successful restore of "ip-10.1.6.100"
```

参考資料： リソース階層のメンテナンス

## リソースの設定変更

ここでは、例としてIP リソースの設定を変更する手順を記載します。
IP リソース以外については「ARK 毎のサブコマンド 」を参考に、手順を適宜読み替えてください。

本ガイドでは LifeKeeper v9.5.0 を使用します。

1. 設定を変更したいリソースのタグ名を確認します。

```
# lkcli status -q
LOCAL    TAG             ID              STATE     PRIO  PRIMARY
target1  ip-10.1.6.100  ip-10.1.6.100  ISP          1  target1

MACHINE  NETWORK ADDRESSES/DEVICE     STATE     PRIO
target2  TCP     10.1.6.88/10.1.6.89  ALIVE        1
```

2. タグ名をもとに、リソースの種類と現在の設定値を確認します。
   なお、リソースの種類は「ARK 毎のサブコマンド 」をご参照ください。

```
# lkcli resource info --tag ip-10.1.6.100
---
app: comm
priority: 1
properties:
  device: ens192
  ipaddr: 10.1.6.100
  netmask: 255.255.255.0
  pinglist: ''
  realip: 0
  restoremode: Enabled
  srcaddr: 0
switchback: INTELLIGENT
tag: ip-10.1.6.100
typ: ip
```

3. リソースの設定を変更します。ここでは `restoremode` を Disabled に変更します。

```
# lkcli resource config ip --tag ip-10.1.6.100 --restoremode Disabled
Performing restoremode change ...

restoremode change successful.
```

4. リソースの設定値が変化していることを確認します。

```
# lkcli resource info --tag ip-10.1.6.100
---
app: comm
priority: 1
properties:
  device: ens192
  ipaddr: 10.1.6.100
  netmask: 255.255.255.0
  pinglist: ''
  realip: 0
  restoremode: Disabled
  srcaddr: 0
switchback: INTELLIGENT
tag: ip-10.1.6.100
typ: ip
```

# 5.6.4.6. クラスターの設定を引き継いで複製する

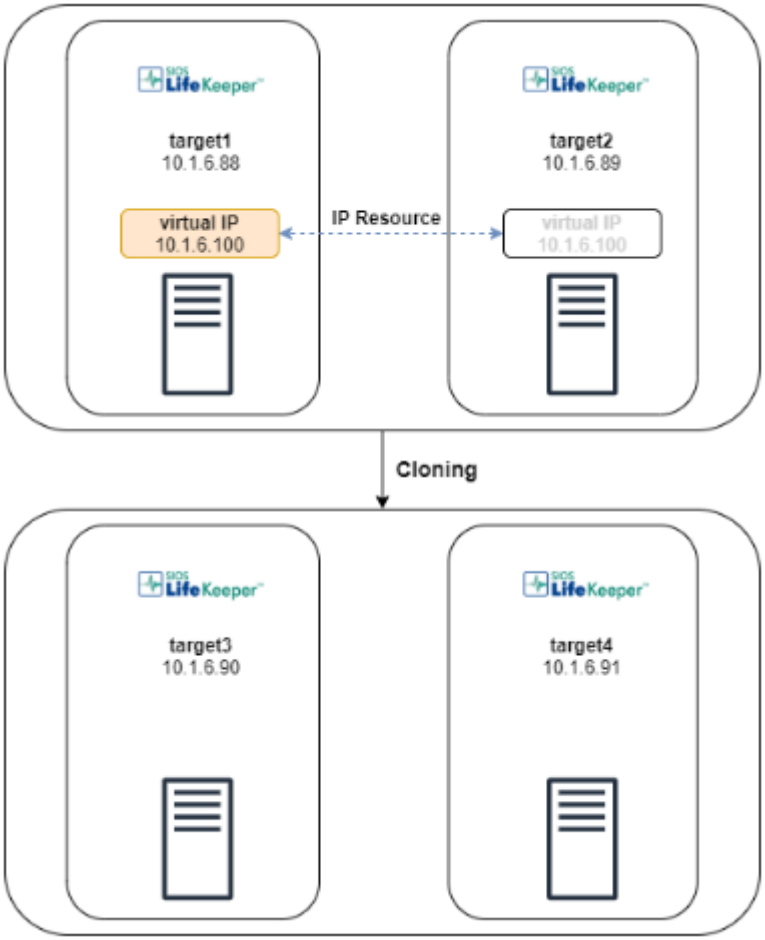
## クラスターの設定を引き継いで複製する

ここでは、コミュニケーションパスとリソースが設定されたクラスターをもとに、同じ設定を持つクラスターを複製する手順を記載します。

### 構成

本ガイドでは LifeKeeper v9.5.0 を使用します。

構成は以下の2ノードのクラスターを想定し、すでに LifeKeeper の設定が完了しているクラスターと、その複製先となる LifeKeeper がインストールされたクラスターを用意してください。なお、複製先の LifeKeeper には、コミュニケーションパス・リソースが設定されていない状態にしてください。
もし複数のクラスターを用意できない場合でも、インポート直前に LifeKeeper の設定を削除することで、動作を確認することができます。



## 注意点

export コマンドでは、出力できるリソースやコミュニケーションパスに一部制限があります。手順を実行する前に、サポートされた環境であるか以下の項目を確認してください。
制限事項の詳細につきましては、 LKCLI のマニュアル をご参照ください。

- エクスポートするクラスターのコミュニケーションパスのプロトコルが、すべてTCP/IPであることを確認してください。

```
# lkcli status -q
LOCAL    TAG             ID              STATE     PRIO  PRIMARY
target1  ip-10.1.6.100  ip-10.1.6.100  ISP          1  target1

MACHINE  NETWORK ADDRESSES/DEVICE     STATE     PRIO
target2    TCP   10.1.6.88/10.1.6.89  ALIVE        1
```

- エクスポートするクラスターのすべてのリソースが、LKCLI でサポートされている ARK であることを確認してください。なお、リソースの種類やサポートしている ARK であるかは ARK一覧 をご参照ください。

```
# lkcli resource info --tag ip-10.1.6.100
---
app: comm
priority: 1
properties:
  device: ens192
  ipaddr: 10.1.6.100
  netmask: 255.255.255.0
  pinglist: ''
  realip: 0
  restoremode: Disabled
  srcaddr: 0
switchback: INTELLIGENT
tag: ip-10.1.6.100
typ: ip
```

## 手順

**target1** でコマンドを実行してください。

1. LifeKeeper の設定が完了しているクラスターで、現在の設定をファイルに保存します。

```
[target1]# lkcli export > src_settings.yml
```

2. エクスポートされたファイルを複製先のクラスターにコピーします。

**target3** でコマンドを実行してください。

3. エクスポートされたファイルの IP アドレスとホスト名を複製先の環境にあわせて書き換えてください。

```
[target3]# sed -e "s/10\.1\.6\.88/10\.1\.6\.90/" -e "s/target1/target3/" -e
"s/10\.1\.6\.89/10\.1\.6\.91/" -e "s/target2/target4/" src_settings.yml > des
t_settings.yml
```

4. LifeKeeper の複製先のクラスターでコミュニケーションパス・リソースがまだつくられていないことを確認します。なおリソースの確認は、クラスター内のすべてのノードで確認してください。
もしコミュニケーションパス・リソースが存在する場合は `lkcli clean --mode all` ですべて削除することができます。なお、clean コマンドを実行すると、設定を元に戻すことはできませんのでご注意ください。

```
[target3]# lkcli status -q
LOCAL    TAG   ID    STATE     PRIO  PRIMARY
```

**target3・target4** でコマンドを実行してください。

5. LifeKeeper の複製先でコミュニケーションパスをインポートして、コミュニケーションパスが作成されることを確認します。なおこのコマンドは、クラスター内のすべてのノードで実行してください。

```
# lkcli import commpath --file dest_settings.yml
Performing commpath 'target3:10.1.6.90/10.1.6.91' create...
Commpath 'target3:10.1.6.90/10.1.6.91' created successful.
```

```
# lkcli status -q
LOCAL    TAG   ID    STATE     PRIO  PRIMARY

MACHINE  NETWORK ADDRESSES/DEVICE     STATE     PRIO
target4    TCP   10.1.6.90/10.1.6.91  ALIVE        1
```

※ コミュニケーションパスのステータスは、2つのノード間で双方向に作成されたとき ALIVE となります。

6. LifeKeeper の複製先でリソースをインポートして、リソースが作成されることを確認します。なおこのコマンドは、クラスター内で1度だけ実行してください。

```
# lkcli import resource --file dest_settings.yml
BEGIN create of "ip-10.1.6.100"
```

```
LifeKeeper application=comm on target1.
LifeKeeper communications resource type= ip on target1.
Creating resource instance with id ip-10.1.6.100 on machine target1
Resource successfully created on target1
BEGIN restore of "ip-10.1.6.100"
END successful restore of "ip-10.1.6.100"
END successful create of "ip-10.1.6.100".
Removing ping list for subnet 172.31.0.0...
Performing restoremode change ...

restoremode change successful.
Building independent resource list
Checking existence of extend and canextend scripts
Checking extendability for ip-10.1.6.100
Pre Extend checks were successful
Extending resource instances for ip-10.1.6.100
Creating dependencies
Setting switchback type for hierarchy
Creating equivalencies
LifeKeeper Admin Lock (ip-10.1.6.100) Released
Hierarchy successfully extended
Removing ping list for subnet 172.31.0.0...
```

7. コミュニケーションパスとリソースが作成されていることを確認します。

```
# lkcli status -q
LOCAL    TAG              ID           STATE     PRIO  PRIMARY
target3  ip-10.1.6.100  ip-10.1.6.100  ISP          1  target3

MACHINE  NETWORK ADDRESSES/DEVICE     STATE     PRIO
target4  TCP     10.1.6.90/10.1.6.91  ALIVE        1
```

# 6. アプリケーションリカバリーキット

LifeKeeper for Linux Application Recovery Kits (ARKs) は特定のアプリケーションを LifeKeeper 上で管理、制御するためのツールとユーティリティを含んでいます。今バージョンの LifeKeeper で利用できる ARK は以下の通りです。

> **重要** ：SIOS は次の Recovery Kit のサポートを終了します。これらは LifeKeeper for Linux の次のメジャーリリースでサポートされなくなります。
> (なお、新規受注は2022年12月31日をもって終了しています。)
> • NEC iStorage StoragePathSavior
> • Samba
> • Postfix
> • HULFT HUB Recovery Kit

Apache Recovery Kit 管理ガイド

DB2 Recovery Kit 管理ガイド

LB Health Check Kit 管理ガイド

Recovery Kit for EC2 管理ガイド

Logical Volume Manager Recovery Kit 管理ガイド

IP Recovery Kit 管理ガイド

MySQL Recovery Kit 管理ガイド

WebSphere MQ Recovery Kit 管理ガイド

NAS Recovery Kit 管理ガイド

NFS Recovery Kit 管理ガイド

Recovery Kit for Oracle® Cloud Infrastructure 管理ガイド

Oracle® Recovery Kit 管理ガイド

PostgreSQL Recovery Kit 管理ガイド

Postfix Recovery Kit 管理ガイド

Recovery Kit for Route 53 管理ガイド

Samba Recovery Kit 管理ガイド

SAP Recovery Kit 管理ガイド

SAP HANA Recovery Kit管理ガイド

SAP MaxDB Recovery Kit 管理ガイド

Sybase Recovery Kit 管理ガイド

VMDK as Shared Storage Recovery Kit 管理ガイド

# 6.1. Apache Recovery Kit 管理ガイド

LifeKeeper for Linux Apache Web Server Recovery Kit は、LifeKeeper 環境で Apache Web Server ソフトウェアの障害回復能力を向上します。

このガイドでは次の事項について説明します。

- LifeKeeper for Linux ドキュメンテーション：LifeKeeper for Linux のドキュメンテーションの全リスト、および情報が入手可能な場所。
- 要件：リカバリーソフトウェアのインストールおよびセットアップを行うには、サーバーが一定のハードウェア / ソフトウェア要件を満たしている必要があります。LifeKeeper for Linux Apache Recovery Kit のインストール / アンインストール方法については、LifeKeeper for Linux インストールガイドを参照してください。
- Recovery Kit の設定：必要とする保護と柔軟性が得られるように LifeKeeper を設定するには、設定ルールを把握している必要があります。設定を適切にプランニングするには、ネットワーク設定、インターフェースの選択、ユーザシステムのセットアップ、階層オプション、Apache 設定作業について理解することが必要です。このセクションでは、設定のプランニングだけでなく、設定の例を示し、Recovery Kit を設定するために必要な特定の作業について説明します。
- トラブルシューティング：このセクションには、情報メッセージとエラーメッセージ、および推奨される対処方法の一覧を示します。

# 6.1.1. LifeKeeper のドキュメンテーションと Apache リファレンス

以下に、SIOS Technology Corp が発行している LifeKeeper 関連情報の一覧を示します。

- LifeKeeper for Linux リリースノート
- LifeKeeper for Linux テクニカルドキュメンテーション (LifeKeeper GUI 内の [Help] メニューから参照できます)。
- LifeKeeper for Linux インストールガイド

このドキュメンテーション、および他の LifeKeeper Recovery Kit に関連するドキュメンテーションは、以下のサイトからオンラインで使用できます。

http://docs.us.sios.com

## 参考ドキュメント

以下に、Apache Web Server アプリケーションおよび LifeKeeper Apache Recovery Kit に関連する参考ドキュメントの一覧を示します。

- Apache のオンラインドキュメンテーション
- Apache: The Definitive Guide, 2nd Edition, Ben Laurie and Peter Laurie, O'Reilly & Associates, Inc. 1999

# 6.1.2. Apache Recovery Kit の要件

Apache Recovery Kit のインストールやアンインストールを試みる前に、このパッケージのハードウェア / ソフトウェア要件と、インストールとアンインストールの手順を理解しておく必要があります。

## キットのハードウェア / ソフトウェア要件

Apache Recovery Kit をインストールして設定する前に、使用している設定が以下の要件を満たしていることを確認してください。

- サーバー：Recovery Kit を使用するには、LifeKeeper for Linux テクニカルドキュメンテーションと LifeKeeper リリースノートに記載されている LifeKeeperの要件に従って設定した、サポートされるコンピュータが 2 台以上必要です。これらのドキュメントは、SIOS のテクニカルドキュメンテーションサイト (docs.us.sios.com) にあります。

- LifeKeeper ソフトウェア：各サーバに同じバージョンの LifeKeeper ソフトウェアとパッチをインストールする必要があります。特定の LifeKeeper の要件については、LifeKeeper リリースノートおよび LifeKeeper for Linux テクニカルドキュメンテーションを参照してください。

- LifeKeeper IP Recovery Kit：同じバージョンの Recovery Kit を各サーバーにインストールする必要があります。

- IP ネットワークインターフェース：各サーバーは、イーサーネット TCP/IP をサポートするネットワークインターフェースを 1 つ以上必要とします。IP スイッチオーバが正しく動作するには、ローカルネットワークに接続されているユーザシステムが標準の TCP/IP 仕様に準拠している必要があります。
  注記: 各サーバーが必要とするネットワークインターフェースが 1 つだけであっても、異種メディア要件、スループット要件、単一障害点の排除、ネットワークのセグメンテーションといった多くの理由のために、複数のインターフェースを使用すべきです。

- TCP/IP ソフトウェア：各サーバーは TCP/IP ソフトウェアも必要とします。

- Apache ソフトウェア：Apache Recovery Kit を設定する前に、各サーバに Apache HTTP Server、および使用する DSO (動的共有モジュール) をインストールし、設定しておく必要があります。各サーバには同じバージョンをインストールする必要があります。サポートされるバージョンは LifeKeeper for Linux サポートマトリックス を参照してください。

Apache Recovery Kit のインストール / アンインストール方法については、LifeKeeper インストールガイドを参照してください。

# 6.1.3. LifeKeeper における Apache Web Server の設定

本セクションでは、LifeKeeper における典型的な Apache Web Server の設定の定義および例と、Apache Web Server の設定を開始する前に考慮すべき情報について説明します。

LifeKeeper Core のリソース階層を設定する手順については、LifeKeeper for Linux テクニカルドキュメンテーション を参照してください。

# 6.1.3.1. 設定の定義と例

Apache Web Server は、httpd デーモンの複数インスタンスの同時実行をサポートしています。LifeKeeper の各 Apache Web Server 階層は、固有の「サーバルート」ディレクトリを持つ個別の Apache インスタンスに対応します。各インスタンスは、「仮想ホスト」を使用するように設定されているかどうかにより、1 つ以上の Web サイトをサポートできます。

主に、サーバルートディレクトリは 1 つの Apache Web Server インスタンスを定義します。これは、Web インスタンスの設定方法を指定する conf/httpd.conf 設定ファイルがこのディレクトリに入れられるからです。このファイル内にある Apache 設定ディレクティブにより、このインスタンスのログファイル、Web ドキュメント、その他の設定ファイルなどの場所、および使用する IP/ ドメイン名のアドレスが指定されます。

LifeKeeper ファイルシステム (共有ストレージを使用) を使用するかどうかに基づいて、Apache Web Server の設定を分類すると便利です。1 つの共有ファイルシステムを、サーバルートディレクトリ (設定ファイル conf/httpd.conf とともに使用) やドキュメントルートディレクトリ (さらにオプションで httpd 実行可能ファイル自体) に使用できます。特定の Apache インスタンスについて、ローカル設定と共有設定のいずれを使用するかは、2 つの主な要因によって決まります。その 1 つは、設定ファイルや Web サイトのドキュメントの複数の同一コピーを個別に管理することの困難さであり、もう 1 つは、2 台以上のサーバで共有 (またはミラー化) できるストレージの可用性とアクセス容易性です。ただし、1 台のサーバ上に、ローカルの Apache インスタンスと共有の Apache インスタンスの両方を設定できることに注意してください。

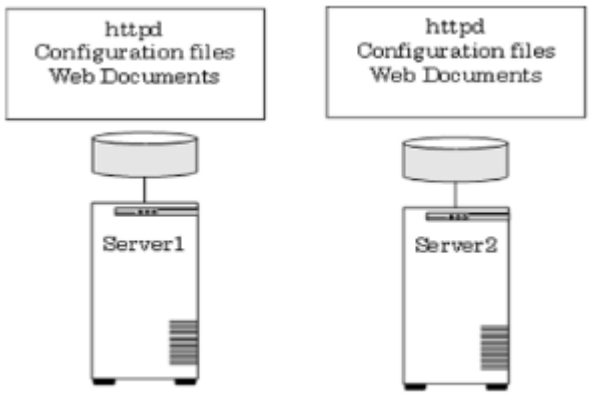
以降のセクションでは、LifeKeeper 環境でのローカル と 共有 の Apache Web Server 設定の例を示し、それぞれの主な特徴の概要を説明します。

## ローカル設定

一般的なローカル設定では、複数のサーバ間で共有されるものはありません。Apache Web Server の設定ファイル、Web ドキュメント、DSO モジュール (およびその設定ファイル)、および httpd 実行可能ファイルのそれぞれについて、同一コピーが各サーバの正確に同じ位置に存在します。各サーバに Apache コンポーネントの同一コピーを維持するのは、Apache 管理者の責任です。

設定ファイルにより、各 Web サイトに IP アドレス、または特定の IP アドレスをマップするドメインアドレスが割り当てられます。それぞれのアドレスについて LifeKeeper の IP アドレスが作成され、Apache リソース階層に追加されます。あるサーバから別のサーバに Apache 階層がスイッチオーバすると、元のサーバではこの特定の httpd インスタンスが停止し、IP アドレスが非アクティブになります。次に、スイッチオーバ先のサーバで、IP アドレスがアクティブになり、インスタンスが開始されます。その後、TCP/IP 経由でクライアントがスイッチオーバ先のサーバ上にある同一の Web サイトに自動的に接続されます。

図 1: ローカル設定




設定に関する注記:

- 図 1 は、共有ファイルシステムに何も配置しないローカル設定の例です。
- 各サーバの同じ位置 (通常は /usr/bin/httpd) に、同一バージョンの Apache Web Server 実行可能ファイルがあります。
- 各サーバは同一のサーバルートディレクトリを持ち、そこには各インスタンスの設定ファイルの同一コピーが配置されます。
- 各サーバは同一のドキュメントルートディレクトリを持ち、そこには各インスタンスの Web ドキュメントの同一コピーが配置されます。
- DSO モジュールを使用する場合は、各サーバの同じ場所にその同一コピーがあります。

Server 1 での Apache Web Server リソース階層の作成:

| | |
|---|---|
| サーバ: | Server1 |
| Web Server バイナリの位置: | /usr/sbin/httpd |
| Web Server のルートディレクトリ: | /home/www/examples/instance1/ |
| ルートタグ | apache-www.examples.instance1 |

Apache Web Server リソース階層の Server 2 への拡張:

| | |
|---|---|
| テンプレートサーバ: | Server1 |
| 拡張するタグ: | apache-www.examples.instance1 |
| ターゲットサーバ | Server2 |

| ターゲットの優先順位: | 10 |
|---|---|

Apache リソース階層を 1 台以上の追加サーバに拡張するときには、ローカル設定と共有設定のいずれの場合でも、すべてのサーバで同一の Web Server バイナリの位置、および同一の Web Server のルートディレクトリを使用する必要があることに注意してください。詳細については、前述の説明、および[Apache Web Server の設定に関する考慮事項] のセクションを参照してください。また、階層の拡張時に、Apache リソース階層の一部を構成する依存リソースがすべて、LifeKeeper により拡張されます。

## 共有設定

一般的な共有設定では、サーバルートディレクトリとドキュメントルートディレクトリはすべて、同一の共有ファイルシステムに存在します。同一の設定ファイルと Web ドキュメントが複数のサーバで共有されるので、各サーバに同一コピーを維持する必要はありません。DSO モジュールを使用する場合は、必要となる設定ファイルやリソースとともに、同じ共有ファイルシステムに配置できます。

Web ドキュメントのみを共有ファイルシステムに配置できることに注意してください。この方法では、サーバルートディレクトリがローカルなので一般的なローカル設定に似ていますが、階層には共有ファイルも含まれます。

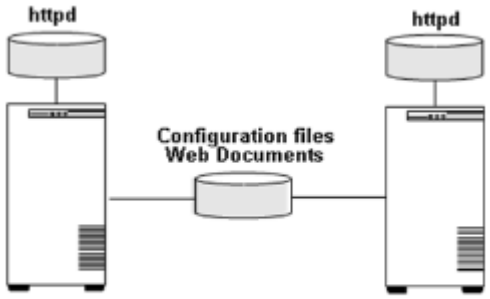
この Apache リソース階層で Apache 実行可能ファイルの特定バージョンまたは個別のコピーを使用する場合は、この実行可能ファイルも共有ファイルシステムに配置できます。この実行可能ファイルは、このインスタンスにのみ使用可能になります。この場合は、Web Server バイナリの位置の入力を要求されたときに、共有ファイルシステム上にある httpd 実行可能ファイルのフルパスを入力します。

使用できる共有ファイルシステムは 1 つのみであることに注意してください。これにより、確実に、共有ストレージにあるすべての必須コンポーネントが同時に利用可能になります。共有ファイルシステム上の Web Server バイナリの位置を使用する場合は、同じ共有ファイルシステム上にある Web Server のルートディレクトリも選択する必要があります。また、このサーバルート用に設定した DocumentRoot ディレクトリがすべて、同じ共有ファイルシステム上に存在する必要があります。同様に、共有ファイルシステム上の Web Server のルートディレクトリを選択するときには、DocumentRoot ディレクトリがすべて、同じ共有ファイルシステム上に存在する必要があります。バイナリとサーバルートが共有ファイルシステムに存在しないが、いずれかの DocumentRoot ディレクトリが共有される場合、すべての DocumentRoot ディレクトリを同じファイルシステム上で共有する必要があります。

これらの規則は、以下のように要約できます。

- Apache 実行可能ファイルを共有する場合は、サーバルートディレクトリを共有する必要があります。
- サーバルートディレクトリを共有する場合は、すべての DocumentRoot ディレクトリを共有する必要があります。
- DocumentRoot ディレクトリを 1 つでも共有する場合は、すべての DocumentRoot ディレクトリを共有する必要があります。
- 個々の Apache リソース階層について、共有ファイルシステムを 1 つのみ使用できます。

図 2:共有設定




設定に関する注記:

- 図 2 は、設定ファイルと Web ドキュメントを共有する共有設定の例です。
- Web ドキュメントのみを共有ファイルシステムに配置できます。この方法は一般的なローカル設定に似ていますが、階層には共有ファイルシステムも含まれる点が異なります。
- DSO モジュールを使用する場合は、必要な設定ファイルやリソースとともに、共有ファイルシステムに配置できます。

Server 1 での Apache Web Server リソース階層の作成:

| | |
|---|---|
| サーバ: | Server1 |
| Web Server バイナリの位置: | /usr/sbin/httpd<br>または<br>/shared/example/instance2/bin/httpd |
| Web Server のルートディレクトリ: | /shared/example/instance2 |
| ルートタグ | apache-shared.example.instance2 |

Apache Web Server リソース階層の Server 2 への拡張:

| | |
|---|---|
| テンプレートサーバ: | Server1 |
| 拡張するタグ: | apache-shared.example.instance2 |

| | |
|---|---|
| ターゲットサーバ | Server2 |
| ターゲットの優先順位: | 10 |

# 6.1.3.1.1. アクティブ / スタンバイ設定とアクティブ / アクティブ設定

Apache Web Server は、LifeKeeper ではアクティブ / アクティブアプリケーションと呼ばれます。これは、いつでも複数の Apache インスタンスをサーバ上で実行できるということです。例えば、2 台のサーバが Apache インスタンスを 1 つずつ実行していて、1 台のサーバに障害が発生した場合、このサーバの Apache インスタンスがもう 1 台のサーバにフェイルオーバでき、さらにもともと実行していたインスタンスの実行も続行できます。一部のアプリケーションではこの機能をサポートしていないため、アプリケーションのインスタンスごとにサーバを使用可能にしておく必要があります。そのようなアプリケーションは、アクティブ / スタンバイアプリケーションと呼ばれます。いずれの設定も使用できるアプリケーションもあります。

Apache を「アクティブ / スタンバイ」モードで実行したい状況が発生することがあります。特に、例えばサーバの 1 台を Apache の実行専用にする場合です。この特殊な場合では、バックアップサーバで何も実行しないように、標準的な Apache のデフォルトインストールの自動起動を無効にする必要があります。

1 台以上の特定サーバで Apache インスタンスを手動で in service にすることで、任意に負荷を分散できます。さらに、各インスタンスのサーバ優先順位を調整することで、障害が発生したときに、Apache インスタンスが最終手段として特定サーバにフェイルオーバするように設定することも、複数のサーバにフェイルオーバして負荷を分散するように設定することもできます。

クラスタ内のすべてのサーバで Apache の自動起動を無効にした場合、このインスタンスで LifeKeeper の IP アドレスを使用するように設定することで、デフォルトサーバのルートディレクトリ「/etc/http」を LifeKeeper の Apache リソース履歴に使用できます。また、ドキュメントのルートディレクトリ用に共有ファイルシステムを使用できます。ただし、これはアクティブ / スタンバイ設定 (前述) なので、通常の方法ではデフォルトのインスタンスを起動できなくなります。さらに、サーバのルートは一意である必要があるため、デフォルトサーバのルートディレクトリを複数の履歴に使用することはできません。

# 6.1.3.2. Apache Web Server の設定に関する考慮事項

Apache リソース階層を作成する前に、Apache Web Server アプリケーションについて以下の設定を完了したことを確認する必要があります。

1. ディストリビューション付属の Apache パッケージをインストールした場合、通常はシステムの起動時に自動起動するよう設定されるため、LifeKeeper の保護と競合します。 競合を回避するためには、各ディストリビューションのマニュアルを参照して、自動起動を無効化してください。

   **注記: SuSE で Apache を使用する場合**: デフォルト設定を使用して Apache を SuSE にインストールする場合、httpd.conf 設定ファイルは ServerRoot のサブディレクトリ conf に配置されません。この場合は、設定ファイルを /etc/httpd/conf ディレクトリに再配置する必要があります。

2. LifeKeeper の個々の Apache Web Server 階層について、個別のルートディレクトリを作成する必要があります。この「サーバルート」ディレクトリは、Apache の「ServerRoot」設定とコマンドラインパラメータに対応します。LifeKeeper の各 Apache Web Server 階層は、一意の Apache インスタンスおよび関連付けられたサーバルートディレクトリに対応します。サーバルートディレクトリは、すべてのサーバ上で同一であり、特定の Apache 階層用に設定する必要があることに注意してください。Web サイトのすべての設定ファイルの情報は、LifeKeeper ソフトウェアから検出とアクセスができるように、サーバルートからの標準の場所 (conf/httpd.conf) に配置する必要があります。

3. BindAddress または Listen のディレクティブを使用して特定の LifeKeeper の IP アドレスをリスンするように、すべての Web サイト (仮想ホスト) を設定する必要があります。これらの LifeKeeper が保護する IP アドレスは、Apache 階層を作成する場所にあらかじめ作成し、in service 可能にしておく必要があります。LifeKeeper が保護する IP アドレスは、Apache リソース階層に自動的に追加されます。

   LifeKeeper の共有ファイルシステムを使用する場合は、Apache 階層を作成する前に、ファイルシステムの作成に必要な準備をすべて行う必要があります。特に、サーバ上のApache 階層を作成する場所に、ファイルシステムをマウントする必要があります。LifeKeeper のファイルシステム階層をあらかじめ作成していない場合、このファイルシステム階層は Apache 階層とともに自動作成され、その後 Apache リソース階層に結合されます。

   仮想ホストの設定の詳細については、Apache Web Server のドキュメンテーションを参照してください。前述したように、すべての Apache インスタンスが、LifeKeeper が保護する特定のアドレスをリスンするように設定する必要があります。例えば、IP ベースの仮想ホストとネームベースの仮想ホストを組み合わせたインスタンスの設定ファイルには、以下のようなディレクティブがあります。

```
User webuser
Group webgroup
ServerName localhost

Listen 172.17.100.55:8000
NameVirtualHost 172.17.100.55:8000
```

```
Listen 172.17.100.56:80

<Virtualhost site.name_one:8000>
ServerName site.name_one
DocumentRoot /shared/site/name_one
</VirtualHost>

<VirtualHost site.name_two:8000>
ServerName site.name_two
DocumentRoot /shared/site/name_two
</VirtualHost>

<VirtualHost 172.17.100.56:80>
ServerName site.ip
DocumentRoot /shared/site/ip
</VirtualHost>
```

4. Apache インスタンスに対する SSL サポートを有効にしている場合、LifeKeeper が保護する適切な IP アドレスを使用するように SSL Listen ディレクティブ (多くの場合、個別の ssl.conf ファイル内にある) を設定する必要があります。この設定を行わない場合、IP アドレス 0.0.0.0 が LifeKeeper で保護されていないことを示すエラーが発生して、Apache 階層の作成に失敗します。一部の Linux 配布の Apache 設定ファイルでは、デフォルトで SSL サポートが有効になっていることに注意してください。

   例えば、/etc/httpd/conf.d/ssl.conf にあるデフォルトの SSL 設定ファイルで、エントリ

   ```
   Listen 0.0.0.0:443
   ```

   を以下のように変更します。

   ```
   Listen 172.17.100.55:443
   ```

5. ローカル設定の場合、プライマリサーバとすべてのバックアップサーバの同じ場所に Apache をインストールして設定し、同一の (または同等の) 設定ファイルをすべてのサーバ上の同じサーバルートディレクトリに設定する必要があります。また、すべてのサーバ上にすべてのドキュメントルートディレクトリが存在し、同一のファイルを持つ必要があります (「設定と定義の例」のローカル設定参照)。

6. 共有設定では通常、LifeKeeper の共有ファイルシステム上にサーバルートディレクトリを設定します。使用できる共有ファイルシステムは 1 つのみであることに注意してください。これにより、確実に、共有ストレージにあるすべての必須コンポーネントが同時に利用可能になります。このため、すべてのドキュメントルートディレクトリが、同じ共有ファイルシステムのサブディレクトリである必要があります。ただし、サーバルートディレクトリ自体のサブディレクトリにする必要はありません。Apache 実行可能ファイルも同じ共有ファイルシステムに配置できますが、この実行可能ファイルは、この特定の Apache リソース階層のみが使用できます。

   **注記**: 共有ストレージを使用する目的で、サーバルートディレクトリを共有ファイルシステムに配置

する必要はありません。設定ファイルなどを入れるローカルのサーバルートディレクトリを選択でき、ドキュメントルートディレクトリのみを共有ファイルシステムに配置できます。ただし、すべてのサーバ上に (前述のローカル設定について)、同一のサーバルートディレクトリ、および同一の (または同等の) 設定ファイルを設定する必要があります。また、すべてのドキュメントルートディレクトリが同じ共有ファイルシステム上に存在する必要があります (「設定と定義の例」の共有設定参照)。

7. 一部の Web サイトの実装では、特定の機能に対する Apache のサポートを拡張するために、DSO (Dynamic Shared Object: 動的共有オブジェクト) モジュールを活用しています。例えば、PHP や Perl の機能を実装するモジュールがあります。実行時に Apache コアから、これらのモジュールのロードとアクセスができます。モジュールを使用する場合は、クラスタ内のすべてのサーバでモジュールを同一設定にする必要があります。Linux プラットフォームでモジュールを使用するように Apache を設定する方法については、モジュールパッケージのドキュメンテーション、およびベンダ提供のドキュメンテーションを参照してください。使用するモジュールおよびそのリソースによっては、フェイルオーバを適切に実行するために、一部のオブジェクトを共有ストレージに配置する必要があります。場合によっては、Generic Application Recovery Kit、またはカスタムリカバリキットを使用して、個別モジュールの保護が必要になることがあります。

8. SSL (Secure Sockets Layer: セキュアソケットレイヤ) モジュールを Apache とともに使用する場合は、サーバをパスワードで保護しないことが重要です。Web サーバがパスワードで保護されている場合、デーモンを起動するたびに管理者がパスワードをプロンプトに入力しなければなりません。この手動操作は、リカバリ時間が重要な HAクラスタ環境とは矛盾するため、LifeKeeper はパスワードで保護されたインスタンスをサポートしません。以下のコマンドを使用してパスワードを削除してください。

   ```
   openssl rsa -in server.key -out unprotected_server.key
   ```

   プロンプトが表示されたら、サーバキーパスワードを入力してください。サイトのセキュリティを維持するために、必ずファイルはルートでのみ読み取り可能にしてください。

   ```
   chmod 400 unprotected_server.key
   ```

   Apache インスタンスの階層作成時に、Recovery Kit により、リソースがパスワードで保護されていないことがチェックされます。リソースがパスワードで保護されている場合、階層の作成に失敗し、エラーメッセージが表示されます。ただし、インスタンスが別のサーバに拡張されている場合、Recovery Kit はバックアップサーバでパスワードの保護をチェックしません。拡張する階層がパスワードで保護されていないことを確認する必要があります。

   サーバキーファイル (Apache の設定ファイルの SSLCertificateKeyFile ディレクティブで指定) は、同じ名前を持ち、クラスタ内のすべてのサーバ上で同じ場所にある必要があります。

   **注記:** LifeKeeper が使用する httpd プロセスの PID ファイル名は、以下の形式を取ります。

   ```
   「/var/run/httpd.<TAG name>.pid」
   ```

   この PID ファイル名は、OS が使用するデフォルトの PID ファイル名とは異なります。この PID ファイルを参照する必要がある場合 (ログのローテーションなど)、LifeKeeper の PID ファイル名と形式に注意してください。

9. SuSE環境で、"/etc/sysconfig/apache2"に定義されている"APACHE_SERVER_FLAGS"を設定する場合は、フラグ名の前に"-D"を記述してください。

   **例:** `APACHE_SERVER_FLAGS=”-D SSL”`

   "-D"がない場合、リソースの作成や起動に失敗することがあります。

# 6.1.4. Apache のLifeKeeper の設定作業

以下の設定作業は、LifeKeeper GUI を使用して実行できます。以下の 4 つの作業は、Apache リソースインスタンスに特有のものであり、Recovery Kit ごとに異なるため、本書で説明しています。

- リソース階層の作成：アプリケーションリソース階層を LifeKeeper クラスタに作成します。
- リソース階層の拡張：リソース階層をプライマリサーバからバックアップサーバに拡張します。
- リソース階層の拡張解除：リソース階層を LifeKeeper クラスタ内の 1 つのサーバから拡張解除 (削除) します。
- リソース階層の削除：リソース階層を LifeKeeper クラスタ内のすべてのサーバから削除します。

以下の作業は、すべての Recovery Kit で共通の作業であり、手順が同一なので、LifeKeeper テクニカルドキュメンテーションの「GUIの管理」セクションで説明しています。

- リソース依存関係の作成: 既存のリソース階層と別のリソースインスタンスとの間に親子の依存関係を作成し、クラスタ内のすべての対象サーバに依存関係の変更を反映します。
- リソース依存関係の削除: リソースの依存関係を削除して、クラスタ内のすべての対象サーバに依存関係の変更を反映します。
- In Service: リソース階層を特定のサーバで In Service にします。
- Out of Service: リソース階層を特定のサーバで Out of Service にします。
- プロパティの表示 / 編集: 特定のサーバでリソース階層のプロパティを表示または編集します。

注記：本セクションの残りの部分では、LifeKeeper GUI の [Edit] メニューから特定の作業を選択して Recovery Kit を設定する方法について説明します。設定作業はツールバーから選択することもできます。状況表示ウィンドウのリソース階層ツリー (左側のペイン) のグローバルリソースを右クリックして、[Edit] メニューと同じドロップダウンメニューの選択項目を表示することもできます。言うまでもなく、これは階層がすでに存在している場合にだけ可能な方法です。

サーバおよび特定リソースの状況に応じて、状況表示ウィンドウのリソース階層表 (右側のペイン) のリソースインスタンスを右クリックして、リソース階層の作成を除くすべての設定作業を実行することもできます。

# 6.1.4.1. Apache Web Server リソース階層の作成

**重要:**

Web Server リソース階層を作成する前に、Apache の設定ファイルに LifeKeeper が保護する既存の IP リソースが含まれていることを確認する必要があります。

Web ドキュメントまたは設定ファイルが共有ディスク上にある共有環境では、共有ファイルシステムがマウントされていることを確認する必要があります。また、重要事項として、リソースをバックアップサーバに拡張するには、稼働しているコミュニケーションパス (すなわちハートビート) が必須であることを念頭においてください。

DataKeeperを使用してレプリケーションファイルシステム階層を作成するには、DataKeeperリソース階層の作成 の情報を参照してください。

プライマリサーバーからリソースインスタンスを作成するには、以下の手順を実行します。

1. LifeKeeper GUI メニューから [Edit] を選択し、次に [Server] を選択してください。メニューから、[Create Resource Hierarchy] を選択してください。

   リソースを作成するときは、Apache Web Server が停止している必要があります。ただし、システム設定ファイルにlisten変数を設定している場合は、デフォルトのデーモンの動作を許可できます。

   [Create Resource Wizard] ダイアログボックスが表示され、クラスター内にインストールされて認識済みの Recovery Kit がすべて、ドロップダウンリストボックスに表示されます。

2. [Apache Web Server] を選択し、 **[Next]** をクリックしてください。

3. 以下の情報を入力するようにプロンプトが表示されます。ダイアログボックスで [Back] ボタンが有効な場合は、前のダイアログボックスに戻ることができます。これは、エラーが発生して、前に入力した情報を修正する必要がある場合に便利な機能です。いつでも [Cancel] をクリックして、作成処理全体を取り消すことができます。

| フィールド | ヒント |
|---|---|
| Switchback Type | [intelligent] または [automatic] を選択してください。この設定は、バックアップサーバーへのフェイルオーバー後にサーバが復帰したときに、このサーバーに Apache インスタンスをスイッチバックする方法を指定します。スイッチバックタイプは、必要な場合、[Resource Properties] ダイアログボックスの [General] タブで後から変更できます。 |
| Server | Apache Web Server を配置するサーバー (通常これをプライマリサーバーまたはテンプレートサーバーと呼ぶ) を選択してください。クラスタ内のすべてのサーバがドロップダウンリストボックスに表示されます。 |
| Web Server | httpd Apache Web Server デーモンのフルパス名 (ファイル名を含む) を選択するか、入力してください。デフォルトは、/usr/sbin/httpd です。 |

| | |
|---|---|
| Binary Location | |
| Web Server Root Directory | Web Server のルートディレクトリのフルパスを指定する必要があります。相対パスやシンボリックリンクを使用することはできません。Apache Web Server の設定ファイルは、サーバルートの conf/httpd.conf にあります。<br><br>注記: この時点で、保護される IP リソースが利用可能かどうかが LifeKeeper によってチェックされます。また、Apache Web Server リソース階層の作成に使用する入力データが妥当かどうかもチェックされます。この 2 つのチェックのどちらかで問題が検出された場合、[ERROR] ボックスが表示されます。Web Server のルートディレクトリーのパスは妥当であるが、Apache の設定自体にエラーがある場合は、そのエラーを修正してから、階層の作成を続けることができます。ここで必要な LifeKeeper IP リソースを作成することもできます。 |
| Root Tag | Web Server 階層に付けるタグ名を選択するか、入力してください。デフォルトの apache<root directory>を選択するか、独自のタグ名を入力することができます。 |

4. [Create] をクリックしてください。[Create Resource Wizard] によって、Apacheリソース階層が作成されます。LifeKeeperによって入力データが検査されます。問題が検出された場合、情報ボックスにエラーメッセージが表示されます。

5. 情報ボックスが表示され、Apacheリソース階層が正常に作成されたこと、およびフェイルオーバによる保護を実現するにはその階層をクラスタ内の別のサーバーに拡張する必要があることが示されます。[Next] をクリックしてください。

   注記: 新しい Apacheインスタンスの起動に失敗したというエラーメッセージが表示されることがあります。新しいApache階層を別のシステムに拡張するには、起動する(In Service にする) 必要があることに注意してください。この時点で、表示されたエラーメッセージに基づいて問題を修正し、新しい階層を In Service にしてから、階層の拡張を続行できます。

6. [Continue] をクリックしてください。[ExtendWizard] が起動されます。リソース階層を別のサーバに拡張する方法の詳細については、「Apache リソース階層の拡張」(後述) の手順 2 を参照してください。

   [Cancel] をクリックすると、別のダイアログボックスが表示され、Apacheリソース階層をLifeKeeperの保護下に置くには、いずれかの時点でここに戻って Apacheリソース階層を別のサーバーに拡張する必要があることが警告されます。

# 6.1.4.2. Apache Web Server リソース階層の拡張

この操作は [Edit] メニューから開始できます。または [Create Resource Hierarchy] オプションの動作が完了すると自動的に開始されます。その場合は、手順 2 を参照してください。

1. [Edit] メニューの [Resource] から [Extend Resource Hierarchy] を選択してください。[Pre-Extend Wizard] が表示されます。拡張操作に慣れていない場合は、[Next] をクリックしてください。LifeKeeper のリソース階層の拡張作業のデフォルト値が分かっていて、入力と確認を省略する場合は [Accept Defaults] をクリックしてください。

2. [Pre-Extend Wizard] に以下の情報を入力してください。

   注記: 最初の 2 つのフィールドは、[Edit] メニューの [Extend] から操作を開始した場合にだけ表示されます。

| フィールド | ヒント |
|---|---|
| Template Server | Apache リソースが現在 in service のサーバを入力してください。ここで選択するテンプレートサーバと次のダイアログボックスで選択する [Tag to Extend] によって、in service のリソース階層が表されることを認識しておくことが重要です。選択したテンプレートサーバで out of service のリソースタグを選択すると、エラーメッセージが表示されます。このダイアログのドロップダウンボックスに、クラスタ内の全サーバの名前が表示されます。 |
| Tag to Extend | テンプレートサーバからターゲットサーバに拡張する Web Server インスタンスの名前を選択してください。ウィザードのドロップダウンリストボックスに、テンプレートサーバ上に作成した全リソースのうち、前のダイアログボックスで選択したリソースの一覧が表示されます。 |
| Target Server | Web Server リソース階層を拡張するターゲットサーバを選択してください。ドロップダウンボックスには、選択した階層に含まれないクラスタ内のサーバ名の一覧が表示されます。 |
| Switchback Type | [intelligent] または [automatic] を選択してください。ここでは、バックアップサーバへのフェイルオーバの後、Web Server インスタンスが in service に戻ったときに、このサーバにどのようにスイッチバックされるのかを設定します。[intelligent] の場合、インスタンスをプライマリ / 元のサーバにスイッチバックするときに管理者の操作が必要になります。[automatic] の場合、プライマリサーバがオンラインに戻り、LifeKeeper コミュニケーションパスを再確立した直後に自動的にスイッチバックが行われます。 |
| Template Priority | テンプレート階層の優先順位を選択するか、入力してください。優先順位は、1 ~ 999 の範囲で未使用の値が有効で、小さい数字ほど優先順位が高くなります (数字 1 が最高の優先順位に相当)。拡張処理時に、別のシステムですでに使用中の優先順位をこの階層に対して指定することはできません。デフォルト値を使用することを推奨します。注記: このフィールドは、階層をはじめて拡張するときにだけ表示されます。 |
| Target Priority | 拡張する Web Server 階層のターゲットの優先順位を選択するか、入力してください。この優先順位は1 ~ 999 の数値で表され、リソースのカスケーディングフェイルオーバシーケンスにおけるサーバの優先順位を示します。階層の優先順位は数字でソートされ、小さい数字ほど優先順位が高くなります (数字 1 が最高の優先順位に相当)。LifeKeeper では、階層が作成され |

| | |
|---|---|
| | たサーバに「1」が自動的に割り当てられることに注意してください。優先順位は連続している必要はありませんが、特定のリソースについて 2 つのサーバに同じ優先順位を割り当てることはできません。 |
| | 拡張前のチェックが正常に終了したというメッセージが表示された後、[Next] をクリックしてください。<br><br>拡張する階層に応じて、拡張されるリソースタグを示す一連の情報ボックス (編集不可) が表示されます。[Extend] をクリックしてください。 |
| Network Interface | ネットワークインターフェースを選択するか、入力してください。これは、IP リソースがターゲットサーバで使用するネットワークインターフェース (つまりイーサーネットカード) の名前です。 |
| Backup Interface | IP リソースを拡張するサーバで IP ローカルリカバリ機能を使用する場合は、バックアップインターフェースを選択してください。デフォルト値は [none] です。ただし、このサーバ上で他のネットワークインターフェースカードを設定している場合は、そのネットワークインターフェースカードがドロップダウンリストに表示されます。 |
| IP Resource Tag | IP リソースタグを選択するか、入力してください。これは、ターゲットサーバに拡張される IP リソースが使用するリソースタグ名です。 |
| Root Tag | ルートタグを選択するか、入力してください。これは、Web Server 階層に付けたタグ名です。デフォルトでは、テンプレートサーバとターゲットサーバの両方でルートタグ名は同一です。 |
| Mount Point | Web Server のルートディレクトリが共有ファイルシステム上に置かれている場合にのみ表示されます。<br><br>Web Server のルートディレクトリが置かれている共有ファイルシステムのマウントポイントを選択するか、入力してください。テンプレートサーバとターゲットサーバで、共有 Web Server のルートディレクトリのマウントポイントが同一である必要があります。多くの場合、ダイアログボックスに表示されたデフォルトのマウントポイントを選択する必要があります。 |
| Root Tag | Web Server のルートディレクトリが共有ファイルシステム上に置かれている場合にのみ表示されます。<br><br>ルートタグを選択するか、入力してください。これは、共有ファイルシステムのタグ名です。 |

3. 拡張が実行中であることを示す情報ボックスが表示されます。同じ Apache リソースインスタンスをクラスタ内の別のサーバに拡張する場合は、[Next Server] をクリックしてください。その場合は、リソース階層を拡張する操作を繰り返します。

   [Finish] をクリックした場合、Web Server リソースの拡張処理が正常に完了したかどうかが検証されます。

4. 最後のダイアログボックスで [Done] をクリックして、[Extend Resources Hierarchy] メニューの選

択を終了してください。

**注記**: 必ず両方のサーバで新しいインスタンスの機能をテストしてください。

# 6.1.4.3. Apache Web Server リソース階層の拡張解除

1. [Edit] メニューの [Resource] から [Unextend Resource Hierarchy] を選択してください。

2. 拡張解除する Web Server リソースが存在するターゲットサーバを選択してください。Web Server が現在 in service になっているサーバは選択できません (このダイアログボックスは、右側のペインでリソースインスタンスを右クリックして [Unextend] 作業を選択した場合には表示されません)。

   [Next] をクリックしてください。

3. 拡張解除する Web Server 階層を選択し、[Next] をクリックしてください (このダイアログボックスは、いずれかのペインでリソースインスタンスを右クリックして [Unextend] 作業を選択した場合には表示されません)。

4. 拡張解除の対象として選択したターゲットサーバと Web Server リソース階層を確認する情報ボックスが表示されます。[Unextend] をクリックしてください。

5. Web Server リソースが正常に拡張解除されたことを示す別の情報ボックスが表示されます。

6. [Done] をクリックして終了してください。

# 6.1.4.4. Apache Web Server リソース階層の削除

階層を out of service にする前に階層を削除した場合、リソース階層は LifeKeeper の保護から除外されますが、Apache インスタンスは現在アクティブなサーバで動作を続行します (ただし、手動で停止した場合やシステムを再起動した場合を除く)。別の IP アドレスを持つ同じ Apache 階層を再作成したり、以前使用した IP アドレスを使用して新しい Apache 階層を作成 (ただし別のサーバルートを使用) しようとすると、同じアドレスで動作中の Apache インスタンスとの競合が発生します。

LifeKeeper 環境内のすべてのサーバからリソース階層を削除するには、以下の手順を実行してください。

1. LifeKeeper GUI メニューから [Edit] を選択し、次に [Resouce] を選択してください。ドロップダウンメニューから、[Delete Resource Hierarchy] を選択してください。

2. 削除する Web Server リソース階層が存在するターゲットサーバの名前を選択してください。[Next] をクリックして次のダイアログボックスに進んでください。

   注記: 右側のペインで個々のリソースインスタンスを右クリックするか、またはリソースが 1 台のサーバ上のみにあるときに左側のペインでグローバルリソースを右クリックして [Delete Resource] 作業を選択した場合、ターゲットサーバのダイアログボックスは表示されません。

3. [Hierarchy to Delete] を選択してください (このダイアログボックスは、左側または右側のペインでリソースインスタンスを右クリックして、[Delete Resource] 作業を選択した場合には、表示されません。)このリストボックスには、ターゲットサーバ上にある、in service と out of service の両方の階層がすべて表示されます。Apache インスタンスを停止してリソース階層を LifeKeeper の保護から削除する場合は、削除前に、選択した階層を必ず out of service にする必要があります。

   [Next] をクリックしてください。

4. 選択したターゲットサーバと、削除の対象として選択した階層を確認する情報ボックスが表示されます。[Next] をしてください。

5. Web Server リソースが正常に削除されたことを確認する別の情報ボックスが表示されます。

6. [Done] をクリックして終了してください。

# 6.1.4.5. Apache Web Server リソース階層のテスト

手動スイッチオーバを開始することによって、Apache リソース階層をテストできます。このテストでは、プライマリサーバからバックアップサーバへのリソースインスタンスのフェイルオーバをシミュレートします。

## GUI からの手動スイッチオーバの実行

スイッチオーバを手動で開始するには、LifeKeeper GUI から [Edit] の [Resource] を選択し、ドロップダウンメニューから [In Service] を選択します。例えば、バックアップサーバで in service 要求を実行すると、アプリケーション階層がバックアップサーバで in service になり、プライマリサーバでは out of service になります。この時点で、元のバックアップサーバがプライマリサーバに、元のプライマリサーバがバックアップサーバに変わります。

Out of Service 要求を実行すると、アプリケーションは他のサーバで in service にならずに、out of service になります。

## リカバリ動作

プライマリサーバに障害が発生すると、Apache Recovery Kit ソフトウェアは以下の作業を実行します。

- プライマリサーバの 1 つ以上の物理ネットワークインターフェースで IP アドレスを in service にすることにより、Apache をバックアップサーバで in service にします。
- ファイルシステムを使用する場合は、そのサーバの共有ディスク上にファイルシステムをマウントします。
- Apache に関連するデーモンプロセスを開始します。

リカバリ後に、ブラウザの [Reload] / [Refresh] ボタンを使用して、再接続できます。

# 6.1.5. Apache Web Server トラブルシューティング

本セクションでは、LifeKeeper Apache Web Server リソース階層の作成 / 拡張中、およびリソースの削除 / リストア中に表示される可能性のあるメッセージの一覧を示します。また、必要に応じて、エラーの原因や、エラー状態を解決するために必要な処置について補足説明します。他の LifeKeeper コンポーネントからメッセージが送出されることもあります。その場合は、メッセージカタログ(弊社のテクニカルドキュメンテーションサイトの「エラーコードの検索」内) を参照してください。メッセージカタログには、操作、管理、GUI など、LifeKeeper for Linux を使用しているときに表示される可能性がある、すべてのエラーコードの一覧を示します。また、エラーコードの原因に関する補足説明や、問題解決のために必要な処置についても、必要に応じて記載します。この一覧から、受信したエラーコードを検索できます。また、該当する LifeKeeper コンポーネントの個別のメッセージカタログに直接アクセスすることもできます。

本セクションのメッセージは、以下のトピックに分類されています。

- Apache 階層作成時のエラー
- Apache 階層拡張時のエラー
- Apache 階層リソースの リストア、リムーブ、リカバリー時のメッセージとエラー

# 6.1.5.1. Apache 階層作成時のエラー

Apache 階層の作成時に表示されるエラーメッセージを、説明とともに以下に示します。LifeKeeper Core および他の Recovery Kit が表示するエラーメッセージはこのガイドには記載していません。ここに記載されている問題を解決するために、階層の作成を中断できます。そして中断した時点から、Apache の設定に必要な新しい LifeKeeper リソースの作成も含めて、階層の作成を続行できます。

## Web Server バイナリの位置検証時

"Error: valid_httpd_path: Must specify absolute path to httpd executable."

有効な Apache httpd 実行可能ファイルの絶対パスを省略せずに入力してください。

"Error: valid_httpd_path: File does not exist at path specified."

指定した場所に、有効な Apache httpd 実行可能ファイルが存在しません。

"Error: valid_httpd_path: Httpd failed to display Server version."

指定した場所にある httpd 実行可能ファイルが、Apache 標準の「サーバのバージョン」を表示しません。

"Error: valid_httpd_path: Incorrect version $MAJOR.$MINOR.$POINT of Apache at $HTTPD_PATH.

指定した場所にある Apache httpd 実行可能ファイルが表示する「サーバのバージョン」が正しくありません。

## Web Server のルートディレクトリの検証時

"Error: valid_http_root: Cannot find Apache configuration file at $CONF_FILE."

指定したサーバルートディレクトリの conf/httpd.conf に、有効な Apache 設定ファイルが必要です。デフォルト設定を使用して Apache を SuSE にインストールする場合、httpd.conf 設定ファイルは ServerRoot のサブディレクトリ conf に配置されません。この場合は、設定

ファイルを /etc/httpd/conf ディレクトリに配置し直す必要があります。

"Error: valid_http_root: Must specify absolute path to Apache server root directory."

サーバルートディレクトリの絶対パスを省略せずに入力してください。

"Error: valid_http_root: Apache instance at $HTTP_ROOT is already under LifeKeeper protection."

各インスタンスには、一意のサーバルートディレクトリが必要であり、設定ファイルが conf/httpd.conf に必要です。指定されたサーバルートディレクトリは、すでに他の Apache インスタンスが使用しています。

"Syntax error on line <line number> of <configuration file path>, etc…"

Apache 設定ファイルに構文エラーがあります。これらのエラーメッセージは、httpd -T コマンドを使用して $CONF_FILE の構文をチェックするときに表示されます。詳細については、表示されたエラーメッセージを確認してください。

"Error: valid_http_root: Since $HTTPD_PATH is shareable on $HTTPD_PATH_SHARED, $HTTP_ROOT must be also."

httpd 実行可能ファイルが共有 / 共有可能ストレージに存在する場合は、サーバルートとすべての DocumentRoot ディレクトリも共有 / 共有可能ストレージに存在する必要があります。

"Error: valid_http_root: Since $HTTP_ROOT is shareable on $HTTP_ROOT_SHARED, all document root directories must shareable on this same filesystem."

サーバのルートディレクトリが共有 / 共有可能ストレージに存在する場合は、すべての DocumentRoot ディレクトリも共有 / 共有可能ストレージに存在する必要があります。

"Error: http_docs_shared: Since one/more Apache document root directories are shareable on $docs_shared, $curr_root must be also.

いずれかの DocumentRoot ディレクトリが共有 / 共有可能ファイルシステムに存在する場合、すべての DocumentRoot ディレクトリが同じファイルシステムに存在する必要があります。

"Error: valid_http_root: Must include BindAddress or Listen directives for each Apache instance. Check the Apache configuration file at $CONF_FILE."

複数の Apache インスタンスを実行するには、各設定ファイルに BindAddress または Listen ディレクティブが必要です。詳細については、前のセクションにあるApache Web Server の設定に関する考慮事項 を参照してください。

"Error: valid_http_root: Default IP address * not allowed for LifeKeeper protection. Check the Apache configuration file at $CONF_FILE."

個々の Apache インスタンスについて、LifeKeeper が保護する IP アドレスを 1 つ以上指定する必要があります。

"Error: valid_http_root: A Listen directive is being used which specifies an IP address but no port. Check the Apache configuration file at $CONF_FILE."

Listen ディレクティブの正しい構文は、「Listen [IPaddress:] port number」です。これは、Apache では構文エラーとは見なされませんが、間違って解釈されます (IP アドレスの最初の数字がポート番号の指定と見なされる)。

"Error: valid_http_root: IP address $ip is not LifeKeeper protected."

Apache の設定ファイルが、LifeKeeper の保護下に設定されていない IP アドレスまたはドメイン名を参照しています。これらの LifeKeeper IP アドレスのリソースをあらかじめ作成しておく必要があります。

## Apache リソース階層の作成時

"Error: Could not find IP resource for $IP_ADD on machine $MACH."

このリソースを作成してから、Apache リソースの作成を続行する必要があります。

"Error: Create Apache file system hierarchy failure for filesystem $FSNAME used by server root $HTTP_ROOT."

"Error: Failure bringing Apache Resource $TAG into service on machine $MACH."

メッセージについては、Apache のエラーログを確認してください (デフォルトの位置は /var/log/httpd/error_log ですが、他のログがリストされることがあります)。

この問題の原因の多くは、Apache の設定ファイルのエラーです。問題を修正した後、このリソースを手動で in service にすることができます。

"LifeKeeper: RESTORE: *ERROR* Apache: The instance is Password Protected."

LifeKeeper Apache Web Server Recovery Kit は、SSL 対応 Web サーバに使用される、パスワードで保護された秘密キーファイルをサポートすることができません。これは、Apache を起動するたびに手動操作が必要になり、再起動とフェイルオーバの自動実行を妨げるからです。本書の「Apache Web Server の設定に関する考慮事項」セクションに、秘密キーファイルのパスワード保護を解除する方法 (SSLCertificateKeyFile ディレクティブで指定) が記載されています。このメッセージは、Apache とともに SSL モジュールを使用する環境にのみ適用されます。

# 6.1.5.2. Apache 階層拡張時のエラー

Apache 階層の拡張時に表示されるエラーメッセージを、説明とともに以下に示します。これらのエラーメッセージは、GUI が「Executing the pre-extend script....」というメッセージを表示して、新しいシステムに階層を拡張する前の階層の検証時に表示されることに注意してください。

各メッセージの前には、

*“Error – canextend(template_server, tag, app_type/resource_type, target_server) -”.*

のようなエラーメッセージが表示されます。

各メッセージの後には、

*“Error – extmgr(template_server, tag, target_tag, target_server) -”.*

のようなエラーメッセージが表示されます。

## ターゲットシステムの Apache 設定ファイルの検証時

*“Cannot find Apache configuration file at $CONF_FILE on $TARGET_SYS.”*

指定したサーバのルートディレクトリの conf/httpd.conf に、有効な Apache の設定ファイルが必要です。

*“DocumentRoot directory “$doc” in $CONF_FILE on $TARGET_SYS was not found in the configuration file on $TEMPLATE_SYS.”*

または

*“DocumentRoot directory “$doc” in $CONF_FILE on $TEMPLATE_SYS was not found in the configuration file on $TARGET_SYS.”*

ターゲットサーバとテンプレートサーバの設定ファイルを比較するときに、これらのいずれとも一致しない DocumentRoot ディレクトリが 1 つ以上見つかりました。表示されたエラーメッセージの詳細を確認して、2 つの設定ファイルの違いを調べてください。DocumentRoot ディレクトリのパスを正しく入力しなかった場合、それぞれの設定ファイルには他方の設定ファイルに存在しないエントリがあるように見えるので、通常はこれらのエラーメッセージの両方が表示されます。

*“IP:port combination “$ipp” in $CONF_FILE on $TARGET_SYS was not found in the configuration file on $TEMPLATE_SYS.”*

または

*“IP:port combination “$ipp” in $CONF_FILE on $TEMPLATE_SYS was not found in the configuration file on $TARGET_SYS.”*

ターゲットサーバーとテンプレートサーバーの設定ファイルを比較するときに、一方のサーバーで使用され

るように設定されているが、他方のサーバーでは設定されていない IP/ ポートの組み合わせが 1 つ以上見つかりました。使用される IP/ ポートの組み合わせは、Apache のさまざまな設定ディレクティブを使用して、IP アドレス、ポート、およびドメイン名で指定されます。比較されるのは、実際に使用される IP/ ポートの組み合わせであり、IP/ ポートの指定に使用されるディレクティブではありません。表示されたエラーメッセージの詳細を確認して、2 つの設定ファイルの違いを調べてください。

*“SSLCertificateKeyFile “$file” in $CONF_FILE on $SYS1 was not found in the configuration file on $SYS2.”*

ターゲットシステムの Apache の設定ファイル内で SSLCertificateKeyFile として指定されたファイル名が、テンプレートシステムで指定されているファイル名と一致しません。これらの設定は同一である必要があります。このメッセージは、Apache とともに SSL モジュールを使用する環境にのみ適用されます。

*“Apache SSLCertificateKeyFile exists on $SYS1 but not on $SYS2.”*

Apache の設定ファイル内に指定された SSLCertificateKeyFile が、一方のシステムには存在しますが、他方のシステムには存在しません。このファイルは、両方のノードに存在する必要があります。このメッセージは、Apache とともに SSL モジュールを使用する環境にのみ適用されます。

*“WARNING: PHP configuration file $PHP_CONFIG appears to be different on $SYS1 and $SYS2.”*

ターゲットシステムの PHP モジュールの設定ファイルが、テンプレートシステムの PHP モジュールの設定ファイルと異なります。両方のサーバの設定を確認して、同一にする必要があります。このメッセージは、Apache とともに PHP モジュールを使用する環境にのみ適用されます。

# 6.1.5.3. Apache 階層リソースの リストア、リムーブ、リカバリー時のメッセージとエラー

以下のトピックに含まれるエラーメッセージは、LifeKeeper のエラーログに出力されます。

このエラーログは、「lk_log log」と入力すると表示できます。

## Apache リソースを In Service にするとき (リストア)

*“LifeKeeper: RESTORE: APACHE: RESTORING $TAG TO SERVICE START AT: <date>”*

情報メッセージです。リストアの開始時刻を記録します。各リストアの開始時に記録されます。

*“LifeKeeper: RESTORE APACHE RESOURCE $TAG END err=$err AT: <date>”*

情報メッセージです。リストアの完了時刻を記録します。各リストアの完了時に記録されます。リストア時にエラーが発生した場合は、追加のメッセージがこれら 2 つのメッセージの間に記録され、err=$err に表示される値がゼロ以外の値になります。

*“Apache: No instance information found for Tag=$TAG.”*

エラー: restore スクリプトに渡されたタグ値を持つインスタンスが定義されていないことを示します。LifeKeeper が既知のタグのみが In Service と Out of Service の動作の選択肢として使用できるため、GUI で発生する可能性は低いです。

*“LifeKeeper: RESTORE: Apache: Tag=$TAG already running.”*

情報メッセージです。表示されたインスタンスがすでに起動して動作中であることを示します。

*“LifeKeeper: RESTORE: Apache: Existing processes terminated for ID=$ID.”*

情報メッセージです。このインスタンス ID について、既存の httpd プロセスが実行中であることが検出されましたが、PidFile が見つからないか、無効です。このため、実行中のプロセスが強制終了されました。

*“LifeKeeper: RESTORE: Apache: Invalid PidFile=$PIDFILE has been deleted.”*

情報メッセージです。このインスタンスについて既存の PidFile が見つかりましたが、その内容が無効です。このため、PidFile が削除されました。

*“LifeKeeper: RESTORE: Apache: Tag=$TAG is being restarted.”*

情報メッセージです。表示されたインスタンスが部分的に動作中であるが、再起動が必要であることを示します。PidFile (親 httpd プロセスのプロセス ID を持つ) がまだ存在している場合、インスタンスは HUP 信号を使用して再起動されます。PidFile が見つからない場合、インスタンスは完全に停止されてから再起動されます。

*“LifeKeeper: RESTORE: *ERROR* Apache: Error in web server configuration file $CONF_FILE for*

*instance $ID."*

*"LifeKeeper: RESTORE: *ERROR* Apache: Execute the following command to check the syntax of this file:"*

*"LifeKeeper: RESTORE: *ERROR* Apache: $HTTPD_PATH -t -d $SERVER_ROOT -f $CONF_FILE."*

インスタンスの起動前に、httpd -t オプションを使用して、設定ファイルの構文がチェックされます。-d オプションは、ServerRoot ディレクトリをチェックします。モジュールを使用するように Apache を設定した場合、モジュールに関するその他のオプションも表示されることがあります。階層作成時に検出された構文エラーは LifeKeeper GUI に表示されますが、その後に持ち込まれた構文エラーは、GUI や LifeKeeper のログには表示されません。以下のコマンドを手動で実行して、設定の不具合を特定する必要があります (該当する場合は、モジュールのオプションを追加する)。

*$HTTPD_PATH -t -d $SERVER_ROOT -f $CONF_FILE*

*"LifeKeeper: RESTORE: *ERROR* Apache: Error starting web server instance $INSTANCE."*

*"LifeKeeper: RESTORE: *ERROR* Apache: Restore of tag $TAG failed."*

*"LifeKeeper: RESTORE: *ERROR* Apache: Examine the Apache error log at $ERROR_LOG"*

*"LifeKeeper: RESTORE: *ERROR* Apache: to determine the cause of the problem."*

パラメータを指定した httpd デーモンの実行時にエラーが発生しました。使用している httpd 実行可能ファイル、設定ファイル、および一般設定に問題がないかどうかをチェックしてください。

*"LifeKeeper: RESTORE: *ERROR* Apache: Web server instance $ID did not start correctly."*

*"LifeKeeper: RESTORE: *ERROR* Apache: Restore of tag $TAG failed."*

*"LifeKeeper: RESTORE: *ERROR* Apache: Examine the Apache error log at $ERROR_LOG"*

*"LifeKeeper: RESTORE: *ERROR* Apache: to determine the cause of the problem."*

多くの場合、Web サイトが予測どおりに応答しなくとも、httpd デーモンが起動したように見えることがあることに注意してください。restore スクリプトは、使用する IP/ ポートの組み合わせをすべてチェックし、設定されたすべてのサイトがフルに機能することを確認します。設定されたすべてのサイトがフルに機能しない場合、このメッセージが出力され、リストアは失敗します。

サイトが Out of Service 状態になっていても、1 つ以上の httpd プロセスが動作し続けている場合があることに注意してください (1 つ以上の Web サイトが動作している可能性があり、それらの Web サイトを強制終了したくないため、意図的にプロセスの動作を継続させています)。できるだけ早く問題を解決し、インスタンスを In Service にする必要があります。問題が解決されない場合、LifeKeeper が最終的にインスタンスのリカバリおよびリストアを自動的に試行します。LifeKeeper によるリカバリとリストアができない場合は、階層が別のサーバにフェイルオーバします。

## Apache リソースを Out of Service にするとき (リムーブ)

*“LifeKeeper: REMOVE: APACHE: REMOVE $TAG FROM SERVICE START AT: <date>”*

情報メッセージです。削除の開始時刻を記録します。各削除の開始時に記録されます。

*“LifeKeeper: REMOVE APACHE RESOURCE $TAG END err=$err AT: <date>”*

情報メッセージです。削除の完了時刻を記録します。各削除の完了時に記録されます。

削除時にエラーが発生した場合は、追加のメッセージがこれら 2 つのメッセージの間に記録され、err=$err に表示される値がゼロ以外の値になります。

"LifeKeeper: REMOVE: *WARNING* APACHE: Error attempting to kill parent process for INSTANCE=$INSTANCE."

親 httpd プロセスを強制終了しようとするエラーがありました (親プロセスのプロセス ID は Pidfileに格納される)。

*“LifeKeeper: REMOVE: *ERROR* APACHE: Error attempting to kill all processes for INSTANCE=$INSTANCE.”*

親 httpd プロセスが正常に強制終了されたように見えますが、このインスタンスの 1 つ以上のプロセスがまだ動作中です。通常、削除動作では、このインスタンスのすべてのプロセスを強制終了できます。強制終了できない場合、このメッセージが出力され、削除は失敗します。

## Apache リソースを In Service に戻すとき (リカバリ)

LifeKeeper Core は、Apache の quickCheck スクリプトを実行して、ローカルサーバで In Service 状態にあるすべての Apache インスタンスの健全性を定期的にチェックします。Apache の quickCheck スクリプトは、リストア時と削除時にインスタンスの状態をチェックするものと同じスクリプトを使用して、Web サイトをチェックします。インスタンスが完全に動作していない場合、recover スクリプトが実行され、インスタンスの再起動が試行されます。これは、以下に示す最初のメッセージを記録し、restore を起動します。restore スクリプトの結果に応じて、最後のエラーメッセージまたは成功メッセージを出力し、restore と同じ結果を返します。restore/recover が失敗した場合、このインスタンスは別のサーバにフェイルオーバされます。

*“LifeKeeper: RECOVER: APACHE: Invoking restore for Apache instance “$ID” at: <date>”*

*“LifeKeeper: RECOVER: APACHE: Restore for Apache instance $ID returned error $RET at: <date>”*

*“LifeKeeper: RECOVER: APACHE: Restore for Apache instance $ID successful at: <date>”*

# 6.1.6. Apache Recovery Kit 処理概要

## 起動処理 (restore)

以下の順序で Apache リソースの起動を試みます。

1. httpd コマンドの -t オプションを使用し、設定ファイルの構文チェックを行います。構文エラーがある場合はエラーで終了します。

2. 設定ファイルから *ssl.conf* が読み込まれるかの情報を元に、SSL モジュールを使用するかどうかを判定します。

3. 以下コマンドで httpd を起動します。起動に失敗した場合はエラーで終了します。
   - SSL モジュールを使用する場合

     ```
     $HTTPD_PATH $DOPTS -d "$SERVER_ROOT" -f "$CONF_FILE" -c "SSLPassPh
     raseDialog exec:/usr/bin/true" -c "PidFile $PIDFILE"
     ```

   - SSL モジュールを使用しない場合

     ```
     $HTTPD_PATH $DOPTS -c "PidFile $PIDFILE" -d "$SERVER_ROOT" -f "$CO
     NF_FILE"
     ```

     各変数には以下の値が格納されます。
     `$HTTPD_PATH`：リソース作成時に指定した httpd バイナリのパス
     `$DOPTS`：*/etc/sysconfig/* に存在する httpd の起動オプション
     `$SERVER_ROOT`：リソース作成時に指定した ServerRoot のパス
     `$CONF_FILE`：*$SERVER_ROOT/conf/httpd.conf* (httpd の設定ファイル)
     `$PID_FILE`：PID ファイルのパス

4. httpd プロセスが起動済みであり、http サーバーに TCP 接続できることを確認します。確認に失敗した場合はエラーで終了します。

## 停止処理 (remove)

以下の順序で Apache リソースの停止を試みます。

1. PID ファイル */var/run/httpd.$INSTANCE.pid* を削除します。
   $INSTANCE はリソース ID から先頭の “apache-” を除いた文字列です。

2. TERM シグナルでプロセスを終了させます。

   ```
   kill -TERM $pid
   ```

3. プロセスを終了できなかった場合、 KILL シグナルでプロセスを終了させます。

```
kill -KILL $pid
```

# 監視処理 (quickCheck)

以下の順序で Apache リソースの監視を試みます。

1. PID ファイル */var/run/httpd.$INSTANCE.pid* が存在し、中身が実際の httpd の PID と一致することを確認します。
   $INSTANCE はリソース ID から先頭の “apache-” を除いた文字列です。

2. http サーバーに TCP 接続できるかどうかを確認します。

いずれかの確認に失敗した時点でリソース障害と判定され、エラーで終了します。
監視処理は 20 秒でタイムアウトします。タイムアウトすると処理を中止しますが、リソース障害とは判定されません。

# 回復処理 (recover)

以下の順序で Apache リソースの回復を試みます。

1. 起動処理と同じ処理を行います。起動に失敗した場合はエラーで終了します。

# 6.2. DB2 Recovery Kit 管理ガイド

## DB2 Recovery Kit テクニカルドキュメンテーション

LifeKeeper for Linux DB2 Recovery Kit は、DB2 データベース・インスタンスの障害回復機能を提供します。LifeKeeper は、DB2 Universal Database 製品ファミリとともに使用して、長時間のシステム停止や人手の介入なしにデータベース・サーバの障害を効果的に回復することによって、DB2 運用環境の可用性を高めます。

## ドキュメントの内容

本書は以下のトピックについて説明しています。

- ドキュメンテーションとリファレンス 。LifeKeeper for Linux のドキュメントと所在の一覧です。
- 要件 。DB2 Recovery Kit を適切に設定、導入、運用するうえで必要なハードウェアとソフトウェアについて説明しています。LifeKeeper for Linux ソフトウェアの具体的なインストール / アンインストール方法については、SIOS Protection Suite インストールガイド を参照してください。
- 概要 。DB2 Recovery Kit の特徴と機能について説明しています。
- LifeKeeper for Linux DB2 Recovery Kit の設定 。DB2 Recovery Kit を適切に設定するうえで必要な手順について説明しています。
- LifeKeeper 設定作業 。LifeKeeper GUI を使用して DB2 リソース階層を作成し管理する作業について説明しています。
- トラブルシューティング 。LifeKeeper for Linux のエラー・メッセージの一覧とその説明です。
- 付録 。DB2 で RAW I/O を使用する設定手順です。

# 6.2.1. ドキュメンテーションとリファレンス

以下に、SIOS Technology Corp が発行している SPS 関連情報の一覧を示します。

- SPS for Linux リリースノート
- SPS for Linux テクニカルドキュメンテーション
- SIOS Protection Suite インストールガイド
- Optional Recovery Kit ドキュメンテーション

このドキュメンテーション、および他の SPS Recovery Kit に関連するドキュメンテーションは、以下の Web サイトで入手できます。

http://jpdocs.us.sios.com

# 6.2.2. ハードウェア要件とソフトウェア要件

SPS for Linux DB2 Recovery Kit をインストールする前に、SPS の構成が次の要件を満たしている必要があります。SPS for Linux ハードウェアおよびソフトウェアの設定に関する具体的な手順については、SIOS Protection Suite インストールガイド を参照してください。

## ハードウェア要件

- サーバ 。製品媒体に付属する SPS for Linux リリースノート および SPS for Linux インストールガイド (製品メディアに収録) に記載の要件に従って設定されたLifeKeeper for Linux 対応サーバ。

- **IP** ネットワーク・インターフェース・カード 。各サーバは、イーサネット TCP/IP をサポートするネットワーク・インタフェースを 1 つ以上必要とします。ただし、LifeKeeper クラスタには 2 つの通信パスが必要で、独立した 2 つのサブネットを使用する 2 つの分離した LAN ベース通信パスが推奨され、これらの 1 つ以上をプライベート・ネットワーク として構成する必要があることを忘れないでください。TCP と TTY のハートビートの組み合わせもサポートされています。

## ソフトウェア要件

- **TCP/IP** ソフトウェア 。LifeKeeper 環境の各サーバには TCP/IP ソフトウェアが必須です。

- **IBM** ソフトウェア 。特定の Linux ディストリビューションとハードウェア・アーキテクチャにおける具体的な DB2 バージョンの要件については、SPS for Linux リリースノート を参照してください。

- LifeKeeper ソフトウェアクラスタの各サーバにインストールした LifeKeeper ソフトウェアのバージョン、および、適用した LifeKeeper ソフトウェア・パッチのバージョンが同一であることが絶対条件です。

- **LifeKeeper for Linux DB2 Recovery Kit** - DB2 Recovery Kit は CD-ROM 媒体で提供されます。Red Hat Package Manager（rpm）でパッケージされており、rpm を使用してインストール / アンインストールを行います。LifeKeeper for Linux DB2 Recovery Kit CD には、次のrpmファイルが格納されています。

  **steeleye-lkDB2**

  LifeKeeper for Linux ソフトウェアの具体的なインストール / アンインストール方法については、SPS for Linux インストールガイド を参照してください。

- **LifeKeeper for Linux NFS Recovery Kit** - DB2 EEE デプロイメントおよびマルチ・パーティション ESE デプロイメントを使用する場合に必要です。この recovery kit は CD の steeleye-lkNFS パッケージに格納されています。

  **重要:**「DB2 EEE またはマルチ・パーティション ESE と NFS に関する問題 」を参照してください。重要な設定情報が記載されています

# 6.2.3. DB2 Recovery Kit の概要

## LifeKeeper for Linux DB2 Recovery Kit

Version 8 以降では、DB2 UDB Enterprise Edition（EE）と Enterprise-Extended Edition（EEE）を組み合わせ、DB2 UDB Enterprise Server Edition（ESE）という 1 つの製品を構成しています。旧バージョンは、標準的なリレーショナル・データベース管理システム としての Enterprise Edition（EE）と、マルチ・パーティション・データベースに対応する EE データベース・サーバの拡張版としての Enterprise-Extended Edition（EEE）の、2 種類の別個のエンタープライズ・レベル・データベース・サーバで構成されています。

LifeKeeper for Linux DB2 Recovery Kit は、EE、WE、WSE 環境のデータベース・マネージャ、および EEE 環境のデータベース・パーティション・サーバを保護します。組み合 わせの ESE 環境では、Recovery Kit はデータベース・マネージャとデータベース・パーティション・サーバの両方を保護します。

ユーザは、LifeKeeper クラスタ内の各マシンに DB2 管理サーバを定義することもできます。DB2 管理サーバを定義すると、LifeKeeper は、DB2 階層作成操作と DB2 階層回復操作の機能として、DB2 管理サーバの起動を試みます。

# 6.2.4. LifeKeeper for Linux DB2 Recovery Kit の設定

本セクションでは、LifeKeeper for Linux DB2 Recovery Kit の設定の詳細について説明します。DB2 Recovery Kit の設定と管理を開始する前に考慮すべき事項についても記載しています。LifeKeeper Core リソース階層の設定方法については、LifeKeeper for Linux テクニカルドキュメンテーション を参照してください。

# 6.2.4.1. DB2 で RAW I/O を使用する場合

DB2 で RAW I/O デバイスを使用する場合は、LifeKeeper Core CD から LifeKeeper Raw I/O Recovery Kit をインストールする必要があります。また、使用前に RAW I/O デバイスを 適切に設定する必要もあります。手順については、「付録 」を参照してください。

# 6.2.4.2. DB2 の使用

## DB2 プロセスの起動回数が多くなりすぎる場合の回避方法

場合によっては、LifeKeeper の保護下で DB2 8.x を使用する場合に、DB2 プロセスの起動回数が多くなりすぎる場合があります。カーネルのネットワーク・パラメータに次の変更を加えることで、この状況を緩和できます。上記のように、DB2 8.x を使用する予定の各 LifeKeeper クラスタ・システムで、*/etc/sysctl.conf* ファイルに次の行を追加します。

```
net.ipv4.tcp_syn_retries=1
```

次に **sysctl -p** を実行すると、この変更が有効になります。

## DB2 インスタンスが頻繁にクラッシュ (Panic) する場合の回避方法

systemdを使用している環境で DB2 を使用する場合、インスタンスがクラッシュすることがあります。systemd のパラメータに次の変更を加えることで、この状況を回避できます。各 LifeKeeper クラスタ・システムで、/etc/systemd/logind.conf ファイルに以下の設定を行ってください。

```
RemoveIPC=no
```

次に **systemctl restart systemd-logind** を実行すると、この変更が有効になります。 詳細はこちら をご覧ください。

# 6.2.4.3. DB2 シングルパーティションの設定の検討事項

LifeKeeper for Linux DB2 Recovery Kit をシングルパーティション環境またはワークグループ環境で運用する前に、以下の事項を考慮する必要があります。

1. LifeKeeperではDB2 インスタンスのホームディレクトリーおよび関連するデータベース、テーブルスペース、リソースの場所が共有ファイルシステムに保存される必要があります。共有ファイルシステムにレプリケーション（SIOS DataKeeper）ファイルシステムリソースを使用する場合は、DB2リソースを作成する前に作成する必要があります。共有ファイルシステムは、階層の作成時に自動的に保護されます。DB2 リソース 階層 の作成時に、DB2 データベースマネージャが親リソースとして作成され、インスタ ンスのホームディレクトリおよび実際のデータベースを含む共有ファイルシステムが従属リソースとして作成されます。したがって、DB2 階層作成後にLifeKeeperで保護されていない共有ファイルシステム上にデータベースを作成する場合は、そのファイルシステム用のリソース階層を作成して、DB2 リソース階層の従属リソースにする必要があります。
2. プライマリーシステムのデータベースマネージャが動作不能になると、サービスは定義済みのバックアップシステムにフェイルオーバーされます。バックアップシステムのデータベースサービスは、従属リソースがフェイルオーバーされ、データベースマネージャがサービスを開始すると直ちに利用可能になります。以前に接続されていた DB2 クライアントは接続を解除されるため、バックアップサーバーに再接続する必要があります。コミットされていない SQL 文はすべてロールバックされるため、再入力する必要があります。

# 6.2.4.4. DB2 マルチ・パーティションの設定の検討事項

**DB2 マルチ・パーティションの制約：** あるマシン上に LifeKeeper DB2 リソース階層を作成すると、そのマシン上の DB2 マルチ・データベース・パーティション・サーバはすべて保護されます。保護するノードは、次のファイルを調べて判断されます。

**<インスタンス・ホーム>/sqllib/db2nodes.cfg**

この Recovery Kit では、将来、N-way フェイルオーバ機能が追加される予定です。

# 6.2.4.4.1. DB2 EEE またはマルチ・パーティション ESE と NFS に関する問題

DB2 インスタンスの実行中に DB2 インスタンスのホーム・ディレクトリの NFS エクス ポート・ポイントが使用できなくなると、システムはエクスポート・ポイントが再び使用可能になるまで待機し、応答しない状態になります。多くのシステム操作は、システムの再起動を含めて、正しく動作しません。DB2 マルチ・パーティション・クラスタ用 の NFS サーバも LifeKeeper で保護する必要があり、NFS リソースをシャットダウンする前に DB2 クラスタの全パーティションも手動でサービス休止にする場合以外は、NFS サーバを手動でサービス休止にしてはいけないことに注意してください。また、DB2 パー ティションは、NFS リソースがサービス中でないとサービスを開始できません。

不注意で NFS サーバを停止してクラスタをハングさせないよう、次のようにすることを推奨します。

**NFS に関する推奨事項**

**追加サーバの使用：** DB2 インスタンスのホーム・ディレクトリをマウントする NFS エク スポート・ポイント用に別のクラスタを用意することを強く推奨します。このクラスタの NFS エクスポート・ポイントは、LifeKeeper NFS Server Recovery Kit で保護する必要があります。

少なくとも 2 台の追加サーバを用意できない場合は、DB2 クラスタに追加サーバを 1 台追加して、前述の問題が発生する可能性を減らすことができます。この追加サー バは NFS 階層をエクスポートします。クラスタ内の別のノードの 1 つがバックアッ プの役目を果たします。この構成では、NFS 階層がバックアップ・ノードにフェイ ルオーバされた場合に、前述の問題が発生します。このクラスタの NFS エクスポート・ポイントは、LifeKeeper NFS Server Recovery Kit で保護する必要があります。

**追加サーバを使用できない場合：** これは、望ましくない選択肢です。ただし、DB2 マルチ・パーティションと同じクラスタで NFS サーバを実行する場合、NFS エクス ポート・ポイントは LifeKeeper NFS Server Recovery Kit で保護する必要があります。 現在、 LifeKeeper では、DB2 パーティションとそれを管理する NFS サーバとの関係が認識されないことに注意してください。したがって、クラスタ内のどのノードでも、LifeKeeper の停止または開始を行う前に、手動で次の手順を踏む必要があります。

1. 1 台のサーバで LifeKeeper を停止する場合は、クラスタ内の別のサーバで NFS サーバが実行中であることを確認する必要があります。そうでない場合、LifeKeeper の シャットダウン処理で、DB2 パーティションのサービスを停止しようとしてハングする可能性があります。一般に、LifeKeeper の再起動時に問題が発生しないようにするには、LifeKeeper を停止する前に、すべての DB2 パーティションを別のサーバに切り替えるか、手動でサービスを停止する必要があります。

2. クラスタ全体をシャットダウンするには、手動ですべての DB2 パーティション・ リソースのサービスを休止する必要があります。次に、すべての DB2 NFS サーバ・ リソースのサービスを休止して、最後に LifeKeeper をシャットダウンします。

3. LifeKeeper をシャットダウンする前に、忘れずに DB2 リソースのサービスを休止していれば、LifeKeeper を正常に再起動できるはずです。次に NFS サーバ・リソースのサービスを開始して、引き続き、再起動するすべての DB2 パーティションのサービスを開始します。

4. LifeKeeper をシャットダウンする前に DB2 パーティションのサービスを休止することを忘れた場合

は、LifeKeeper を再起動する前に、そのパーティションの NFS サーバ・リソースがクラスタ内のどこかで実行中であることを確認する必要があります。

# 6.2.4.4.2. 設定条件

マルチ・パーティション環境で DB2 Recovery Kit が正しく動作するには、以下の要件を 満たしている必要があります。

1. NFS 階層のために追加のクラスタを使用できない場合、LifeKeeper for Linux DB2 Recovery Kit によって、NFS が保護しているファイル・システム上のアクティブ inode の数が制限されることを忘れないでください。したがって、この状態を避けるため、トップレベルのディレクトリーを保護し、完全修飾ディレクトリー名を使用して インスタンスのホーム・ディレクトリーをエクスポートすることをお勧めします。トップレベル・ディレクトリーは、ユーザが直接そこにディレクトリーを変更（すなわち *cd <トップレベル・ディレクトリー>* の実行）できないように、保護されます。

2. 本書の「ソフトウェア要件」セクションで説明したように、IBM の最新 Fix Pack （EEE デプロイメントの場合）をインストールしていることを確認します。

3. *db2nodes.cfg* ファイルのホスト名の値を、 **hostname** コマンドが返す値と同じにします。

   例：

   db2nodes.cfg ファイル：

   **0 server1.sc.steeleye.com 0**

   さらに、サーバーの/etc/hosts ファイルのホスト名の値も db2nodes.cfg ファイルのホスト名の値と同じにする必要があります。

   また、サーバーの/etc/hosts ファイルに、そのファイルに含まれる各サーバー・エントリ のローカル・ホスト名と完全修飾ホスト名が両方記述されていることを確認する必要もあります。

   例：

   */etc/hosts* ファイル

   ```
   127.0.0.1 localhost localhost.localdomain

   9.21.55.53 server1.sc.steeleye.com server1
   ```

4. db2setup スクリプトを実行するときに、この時点で、DB2 Warehouse Control Database（DWCNTRL）または DB2 Sample Database の作成を 選択しないでください。 DB2 リソース階層を正しく作成するには、これらのデータベースを共有ファイル・システム上に作成する必要があります。db2setup スクリプト実行中に、どちらかのデータベースの作成を選択すると、データベースが共有ファイル・システムではなくホーム・ディレクトリーに作成されます。これらのデータベースを作成する場合は、 db2setup スクリプトでは作成せず、別の方法で共有ファイル・システムを指定して作成する必要があります。

   Version 8.1 以降では、セットアップ・スクリプトの実行中に DB2 ツール・カタログが作成されないようにします。必要に応じて、このデータベースは共有ファイル・ システム上に置き、セットアッ

プの完了後かつ階層の作成前に作成しなければなりません。

5. アクティブ／アクティブ環境、つまりマルチ・パーティション・サーバー環境では、 設定内の各サーバーは、フェイルオーバー・シナリオですべてのデータベース・インスタンスを実行できる必要があります。特定のシステム・リソース下で可能な DB2 インスタンスまたはパーティション・サーバーの最大数の決定方法については、『IBM Getting Started Guide』を参照してください。

6. 共有ファイル・システムを選択または作成して、そのファイル・システムをエクスポートします（つまり、/export/db2home）。このファイル・システムは DB2 インスタンスのホームとして使用されます。（注記：本要件のひとつの例外として、常に同一のサーバー上で動作しているパーティションがあります。そのケースにおいては、NFS のエクスポートは不要で、インスタンスのホームディレクトリーは、共有ストレージ上に配置することができます。）

7. LifeKeeper NFS リソース階層を作成して、エクスポートしたファイル・システムを保護します。このファイル・システムは、NFS 階層に従属リソースとして含める必要があります。

8. エクスポートされるサーバーも含めて、クラスタの各サーバーに共有ファイル・システムを NFS マウントします。マウント・オプションについては『DB2 Quickstart Guide』を参照してください。DB2 インスタンスの作成時には、インスタンスのホーム・ディレクトリーは NFS でマウントしたファイル・システム上に配置する必要があります。 ファイル・システムのマウント時には、NFS 階層作成時に使用し、 LifeKeeper によって保護されている切り替え可能な IP アドレスを使用します。さらに、ホーム・ディレクトリーのマウント・ポイントを LifeKeeper クラスタの全サーバー の/etc/fstab ファイルで指定しておく必要があります。 構成内の各サーバーは、同一のマウント・ポイント（すなわち/db2/home）にファイル・システムをマウントする必要があります。

   注記： DB2 リソース階層を作成する前に、NFS 階層を作成してテストすることをお勧めします。NFS 階層の作成とテストの詳細な手順については、LifeKeeper for Linux NFS Server Recovery Kit 管理ガイド を参照してください。

9. 構成内の全サーバーについて、次の DB2 環境変数をインスタンスのパーティション総数と同じ値に設定します。 この環境変数を設定するには、インスタンス・オーナーとしてログオンして、db2set コマンドを発行します。この環境変数を調整することで、 考えられるすべてのフェイルオーバー・シナリオに対応できます。

   **db2set DB2_NUM_FAILOVER_NODES=<**インスタンスのパーティション数**>**

10. 以下の DB2 ユーティリティーを使用して、既存の DB2 インスタンスと DB2 管理サーバーをアップデートします。

    db2iupdt および dasiupdt

11. インスタンスのデータを管理するデータベース・パーティション・サーバーを持つクラスタの各サーバーに LifeKeeper DB2 階層を作成する必要があります。 データベースとテーブルスペースは、共有ファイル・システム上に存在する必要があります。データベース・パーティション・サーバーそれぞれと、 NFS のエクスポートされたホーム・ディレクトリーには別個の LUN が必要です。従属リソースには、実際のデータベ ースとテーブルスペースが配置されるファイル・システムが 格納されます。

12. DB2 階層作成後に、保護されていない LifeKeeper ファイル・システム上にデータベ ースを作成する

場合は、 そのファイル・システム用のリソース階層を作成して、 DB2 リソース階層の従属リソースにする必要があります。 この階層は、db2node.cfg ファイルによってサーバー上で実行されることが示されているすべてのパーティション・サーバーを保護します。

13. フェイルオーバーが正しく実行されるようにするには、各データベース・パーティション・サーバーのファイル・システムに、 必ず固有の番号がつけられている必要があります。

    例：

    データベース・パーティション・サーバー node0 のマウント・ポイントは次のようにします。

    **/<FSROOT>/<db2 インスタンス名>/NODE0000**

    データベース・パーティション・サーバー node1 の マウント・ポイントは次のようにします。

    **/<FSROOT>/<db2 インスタンス名>/NODE0001**

    **注記：** この例では、2 台のパーティション・サーバーがあり、各サーバーのファイル・システムは別個の LUN にマウントされています。

14. あるマシンで DB2 階層を正しく作成するには、そのマシンのすべてのデータベース・パーティション・サーバーが動作している必要があります。

15. プライマリ・システムのデータベース・パーティション・サーバーが動作不能になると、サービスは定義済みのバックアップ・システムにフェイルオーバーされます 。バックアップ・システムのデータベース・サービスは、従属リソースがフェイルオーバーされ、データベース・パーティション・サーバーがサービスを開始すると直ちに利用可能に なります。以前に接続されていた DB2 クライアントは接続を解除されるため、バックアップ・サーバーに再接続する必要があります。 コミットされていない SQL 文はすべてロールバックされるため、再入力する必要があります。

# 6.2.4.5. すべての DB2 構成に共通する設定の検討事項

1. DB2 インスタンス名に使用する文字は英数字だけにする必要があります。

2. DB2 クライアントは、LifeKeeper が保護している IP アドレス経由でデータベースに接続するよう設定する必要があります。 ユーザは **$instancehome/sqllib/profile.env** に**“DB2SYSTEM=<浮動 IP>”** と定義して、クライアントの浮動 IP アドレスを カタログすることができます。

3. DB2 リソース階層を保護している構成内の各サーバの/etc/services ファイルは、 保護されているインスタンスに関して、同一のサービス・エントリを持つ必要があります。また、保護されている DB2 インスタンスのユーザ ID、グループ ID、 インスタ ンスのホーム・ディレクトリは、リソースが保護されるすべてのサーバで同じでな くてはいけません。

   DB2では、以下のエントリをデフォルトとして */etc/services* に追加します。

   DB2_db2inst1 60000/tcp

   *DB2_db2inst1_1 60001/tcp*

   *DB2_db2inst1_2 60002/tcp*

   *DB2_db2inst1_END 60003/tcp*

   *db2c_db2inst1 50001/tcp*

4. 回復は、システム・クラッシュ時など、DB2 が突然終了したあとに実行されます。次に、障害発生時に DB2 の回復にかかる時間を大幅に 短縮するためのヒントを挙げます。

   - º DB2 が処理するログ・レコードを制限します。これは、設定パラメータの

**SOFTMAX** と **LOGFILSIZ** を適切に設定することによって 実現できます。ログ・ファイルのサイズを 4MB（1000 個の 4KB ページ）にして、次のようにアク ティブなログ・スペースの量をログ・ファイルの 25%（1MB）に設定して ください。

**db2 UPDATE DB CFG FOR <db 名> USING SOFTMAX 25**

**db2 UPDATE DB CFG FOR <db 名> USING LOGFILSIZ 1000**

º 次のように、負荷を処理するのに十分な数のページ・クリーナが存在するようにします。

**db2 UPDATE DB CFG FOR <db 名> USING NUM_IOCLEANERS <数値>**

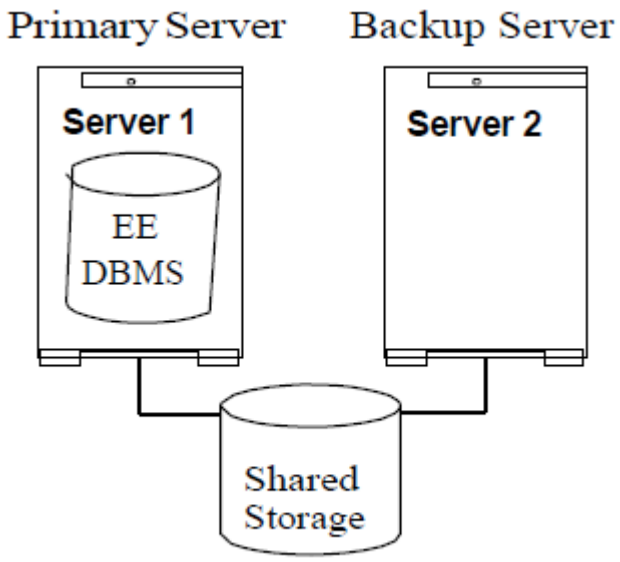
5. DB2 障害モニター(DB2 Fault Monitor)が有効になっている場合は、無効にしてください。

6. 全てのサーバに DB2 をインストールしてください。

# 6.2.4.6. 構成例

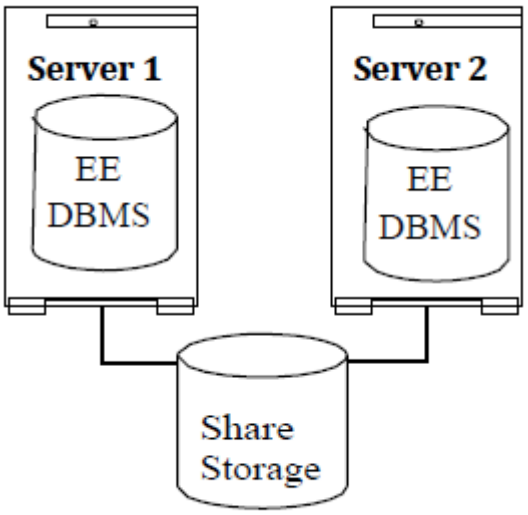
フェイルオーバが発生したときに、LifeKeeper for Linux DB2 Recovery Kit を使用しているとどの ような処理が行われるか、その例をいくつか示します。以下の図では、EE と EEE を使用してデ ータベース設定を示しています。必要に応じて、ESE に置き換えてください。

## 構成 1：DB2シングル・パーティションのアクティブ／スタンバイ構成



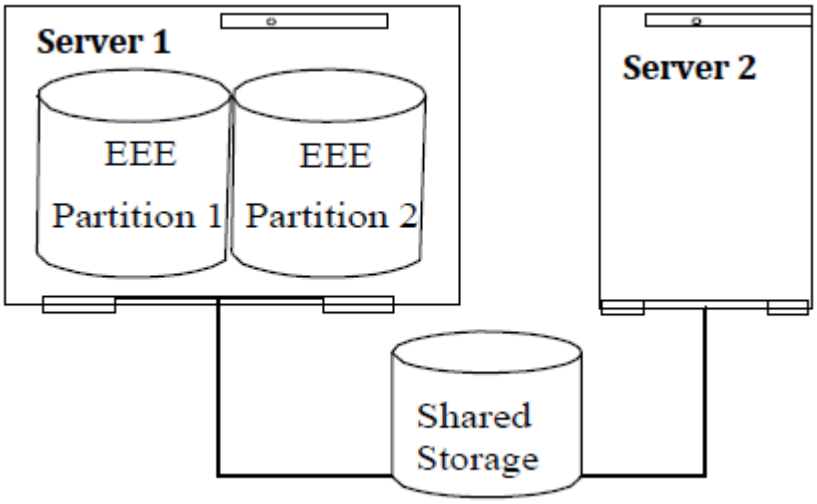

ある DB2 インスタンスがサーバ 1 で保護されていて、別の DB2 インスタンスがサーバ 2 で保護されています。障害発生時には、各サーバが別のサーバのリソースを引き継ぎます。

## 構成 2：DB2シングル･パーティションのアクティブ／アクティブ構成




ある DB2 インスタンスがサーバ 1 で保護されていて、別の DB2 インスタンスがサーバ 2 で保護されています。障害発生時には、各サーバが別のサーバのリソースを引き継ぎます。

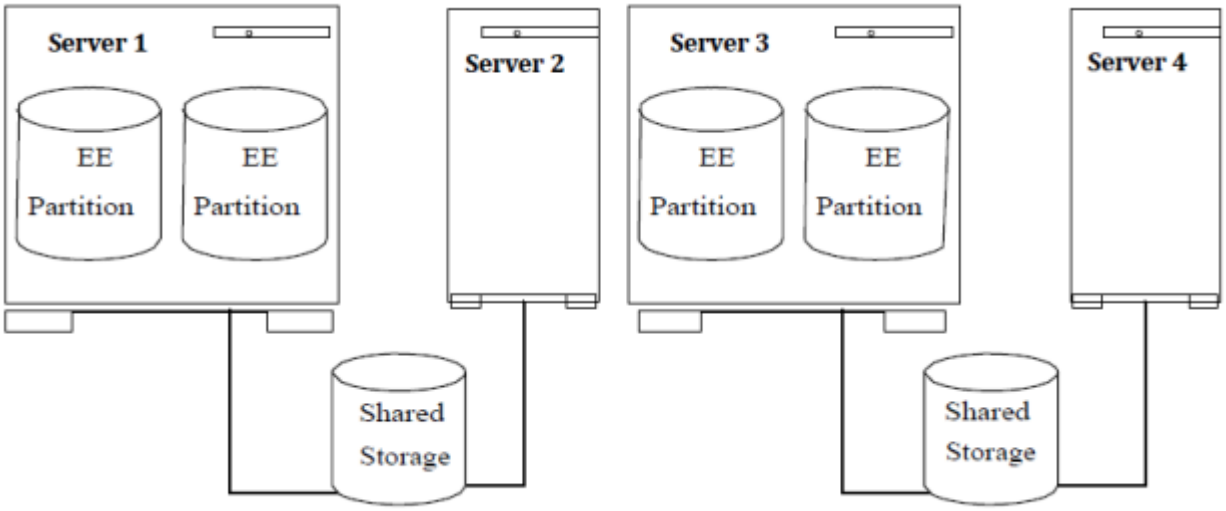
## 構成 3：DB2 マルチ･パーティションのアクティブ／スタンバイ構成（1 クラスタ）




2 つのデータベース･パーティション･サーバを持つ 1 つの DB2 インスタンスが、1 つ の LifeKeeper DB2 リソース階層を持つサーバ 1 で保護されています。障害発生時には、 サーバ 2 が DB2 リソース階層の所有権を引き継ぎます。

注記：以下に挙げるクラスタ群の構成では、どの場合も複数物理ノード上でマルチ･パー ティションを持つ 1 つの DB2 インスタンスだけを保護していることに注意してください。

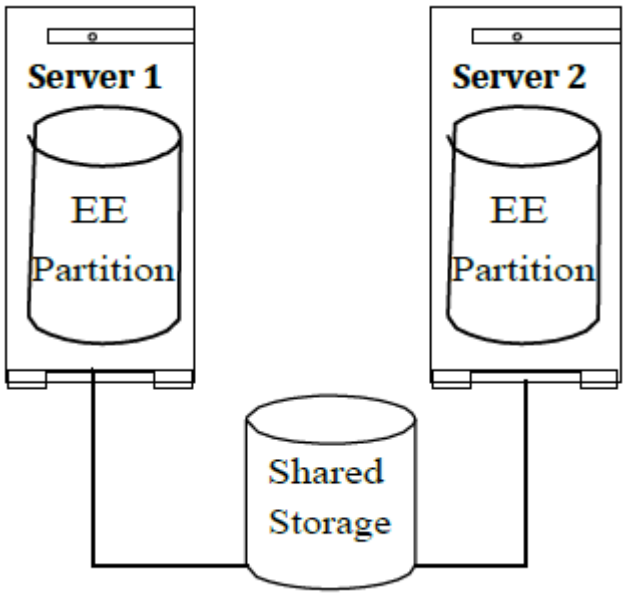
# 構成 4：DB2 マルチ・パーティションのアクティブ／スタンバイ構成（クラスタ群）




2 つのデータベース・パーティション・サーバを持つ 1 つの DB2 インスタンスがサーバ 1 で、2 つのデータベース・パーティション・サーバがサーバ 3 で保護されています。サーバ 1 上に LifeKeeper DB2 リソース階層が 1 つあり、それがサーバ 2 に拡張されていて、サーバ 3 にある別の DB2 リソース階層がサーバ 4 に拡張されています。サーバ 1 で障害が発生すると、サーバ 2 がそのリソースを引き継ぎます。サーバ 3 で障害が発生すると、サーバ 4 がそのリソースを引き継ぎます。

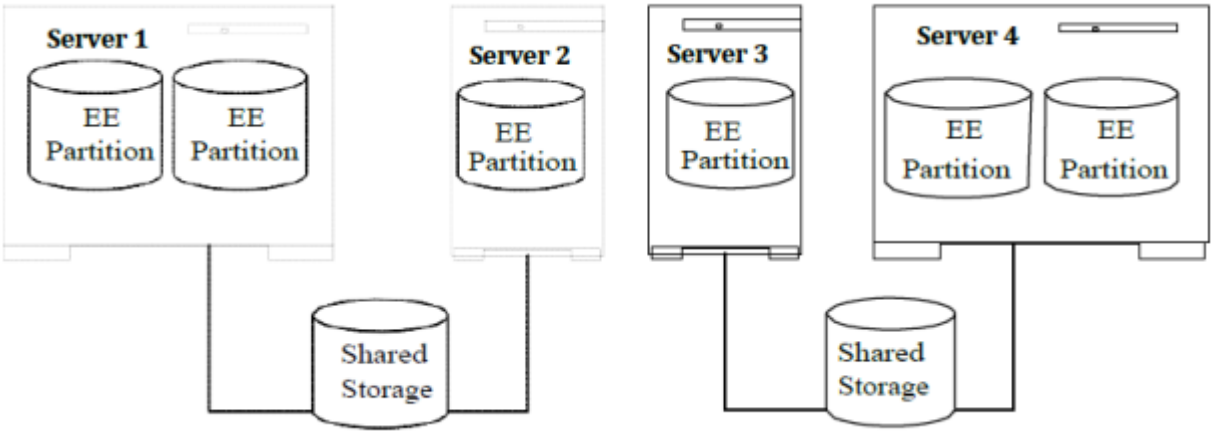
DB2 インスタンスのホーム・ディレクトリをエクスポートしているサーバとそのバック アップ・サーバが同時に動作不能になった場合、DB2 データベースにはアクセスできません。また、エクスポートされた DB2 インスタンス・ディレクトリ（プライマリおよび すべてのバックアップ）の NFS 階層が同時に動作不能になった場合、DB2 データベース は、NFS 階層が回復されるまで、アクセスできません。

## 構成 5：DB2 マルチ・パーティションのアクティブ／アクティブ構成（1 クラスタ）




1 つのデータベース・パーティション・サーバを持つ 1 つの DB2 インスタンスがサーバ 1 で、1 つのデータベース・パーティション・サーバがサーバ 2 で保護されています。 サーバ 1 に LifeKeeper DB2 リソース階層が 1 つあり、サーバ 2 に別の DB2 リソース階層があります。障害発生時には、各サーバが他のサーバのリソースを引き継ぎます。

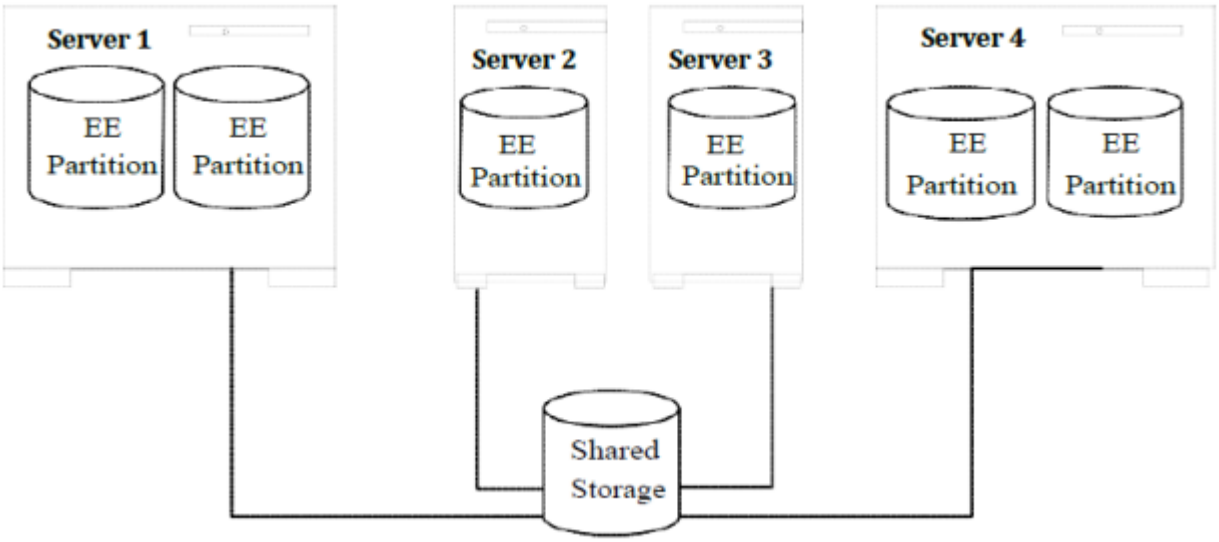
## 構成 6：DB2 マルチ・パーティションのアクティブ／アクティブ構成（クラスタ群）




2 つのデータベース・パーティション・サーバを持つ 1 つの DB2 インスタンスがサーバ 1 で、1 つのデータベース・パーティション・サーバがサーバ 2 で、1 つのデータベース・パーティション・サーバがサーバ 3 で、 2 つのデータベース・パーティション・ サーバがサーバ 4 で保護されています。クラスタの各サーバに LifeKeeper DB2 リソース 階層が 1 つあります。障害発生時には、サーバ 1 とサーバ 2 が互いのリソースを引き継ぎ、サーバ 3 とサーバ 4 が互いのリソースを引き継ぎます。

DB2 インスタンスのホーム・ディレクトリをエクスポートしているサーバとそのバック アップ・サーバが同時に動作不能になった場合、DB2 データベースにはアクセスできません。また、エクスポートされた DB2 インスタンス・ディレクトリ（プライマリおよびすべてのバックアップ）の NFS 階層が同時に動作不能になった場合、 DB2 データベース は、NFS 階層が回復されるまで、アクセスできません。

## 構成 7：DB2 マルチ・パーティション（4 ノード Fibre Channel クラスタ）




2 つのデータベース・パーティション・サーバを持つ 1 つの DB2 インスタンスがサーバ 1 で、1 つのデータベース・パーティション・サーバがサーバ 2 で、1 つのデータベー ス・パーティション・サーバがサーバ 3 で、2 つのデータベース・パーティション・ サーバがサーバ 4 で保護されています。クラスタの各サーバに LifeKeeper DB2 リソース 階層が 1 つあります。 障害発生時には、クラスタの各サーバが互いのバックアップになります。

DB2 インスタンスのホーム・ディレクトリをエクスポートしているサーバとそのバック アップ・サーバが同時に動作不能になった場合、DB2 データベースにはアクセスできません。また、エクスポートされた DB2 インスタンス・ディレクトリ（プライマリおよび すべてのバックアップ）の NFS 階層が同時に動作不能になった場合、 DB2 データベース は、NFS 階層が回復されるまで、アクセスできません。

# 6.2.5. LifeKeeper for Linux DB2 Recovery Kit の設定タスク

LifeKeeper for Linux DB2 Recovery Kit の管理タスクはすべて LifeKeeper GUI から実行できます。LifeKeeper GUI は DB2 リソースの設定、管理、監視を行うためのガイド付きインターフェースを提供します。

LifeKeeper for Linux DB2 リカバリーキットの設定には以下のタスクがあります。

- DB2 リソース階層の作成 DB2 リソース階層を作成します。ファイルシステムにDataKeeperによるレプリケーションファイルシステムを利用する場合にはDB2リソース階層を作成する前にDataKeeperファイルシステムソースを作成する必要があります。
- DB2 リソース階層の削除 DB2 リソース階層を削除します。
- DB2 リソース階層の拡張 DB2 リソース階層をプライマリーサーバーからバックアップサーバーに拡張します。
- DB2 リソース階層の拡張解除 LifeKeeper クラスターの 1 つのサーバーの DB2 リソース階層を拡張解除（削除）します。
- 従属関係の作成 既存のリソース階層と別のリソースインスタンスとの間に子従属関係を作成し、クラスタ内のすべての対象サーバーに従属関係の変化を通知します。
- 従属関係の削除 リソースの従属関係を削除して、クラスター内のすべての対象サーバーに従属関係の変化を通知します。
- In Service リソース階層をアクティブにします。
- Out of Service リソース階層を非アクティブにします。
- 表示 / 編集 リソース階層のプロパティーを表示または編集します。

# 6.2.5.1. DB2 リソース階層の作成

プライマリーサーバーで次の作業を行います。

1. **[Edit] > [Server] > [Create Resource Hierarchy]** を選択します。

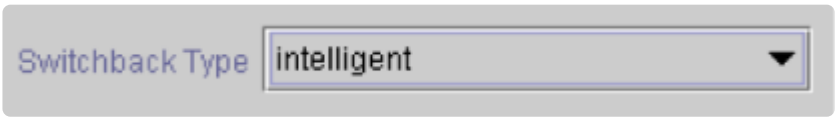
2. **リカバリーキットの選択** ダイアログが表示されます。ドロップダウンリストから **[DB2 Database]** を選択します。

   **[Next]** をクリックして次に進みます。

   注記：階層作成手順のどの段階でも、**[Cancel]** ボタンをクリックすると、作成プロセス全体がキャンセルされます。

3. **Switchback type** ダイアログが表示されます。スイッチバックタイプによって、フェイルオーバー後、バックアップサーバーで DB2 リソースが In Service (アクティブ)になったとき、DB2 リソースをプライマリーサーバーにスイッチバックする方法が決まります。スイッチバックタイプは intelligent または automatic です。インテリジェントスイッチバックの場合、リソースをプライマリーサーバーに戻すために管理者の操作が必要ですが、自動スイッチバックの場合は、プライマリーサーバーがオンラインに戻り、LifeKeeper との通信パスが再確立されるとすぐにスイッチバックが実行されます。

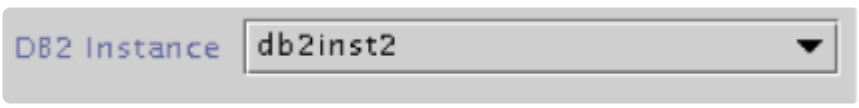
   **[次へ]** をクリックして次に進みます。

4. **[Server]** ダイアログが表示されます。DB2 リソースを作成するサーバーの名前を選択します（通常は、これがプライマリーサーバーです）。クラスター内のすべてのサーバーがドロップダウンリストボックスに表示されます。

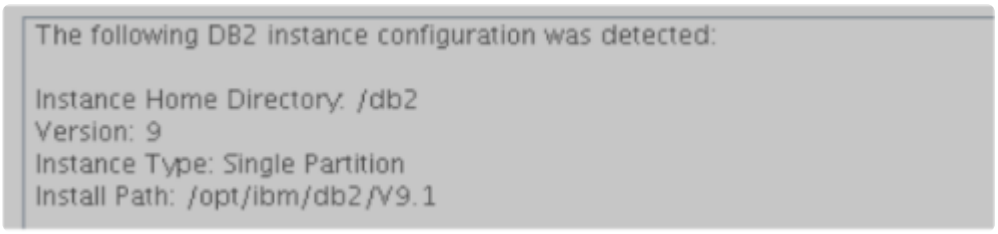
   **[Next]** をクリックして次に進みます。

5. **DB2 Instance** ダイアログが表示されます。保護する DB2 インスタンスの名前を選択するか入力します。

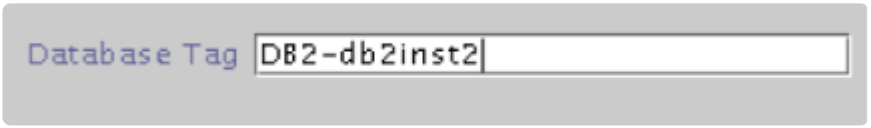
**[Next]** をクリックして次に進みます。

6. 検出されたインスタンスに関する情報を示す情報ボックスが表示されます。



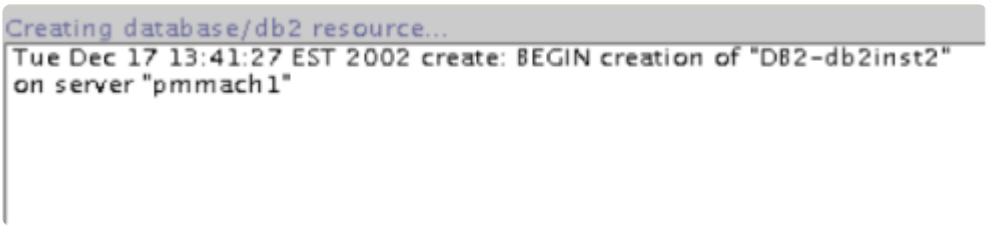

**[Continue]** をクリックして次に進みます。

7. **Database Tag** ダイアログが表示されます。このダイアログでは、新しい DB2 データベースリソースインスタンス用の他と重複しないタグ名が自動的に入力されています。



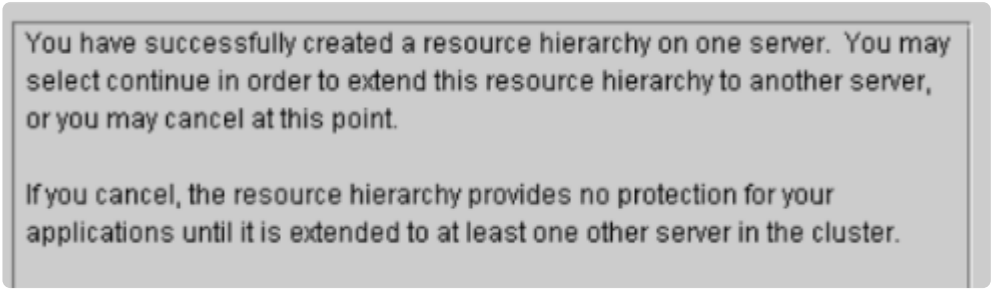

**[Create]** をクリックして次に進みます。

8. 階層の作成が開始されたことを示す情報ボックスが表示されます。

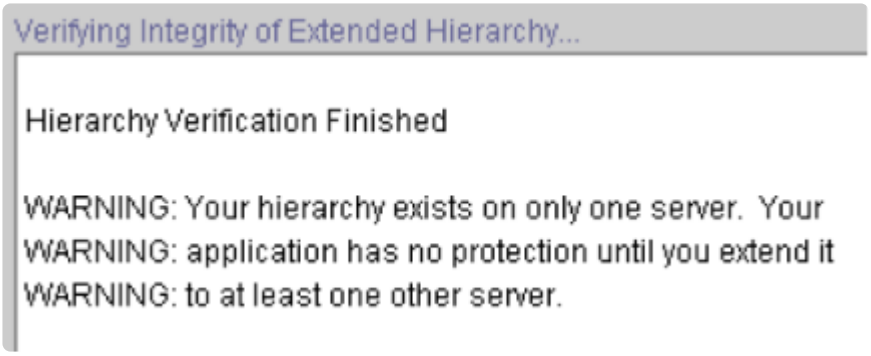



**[次へ]** をクリックして次に進みます。

9. DB2 Create Resource Hierarchyに成功したことを示す情報ボックスが表示されます。リソース階層を LifeKeeper で保護するには、クラスター内の別のサーバーにリソース階層を拡張する必要があります。




リソースを拡張する場合は、 **[Continue]** をクリックします。 後でリソースを拡張する場合は、 **[Cancel]** をクリックします。




10. **[Done]** をクリックして、 **[Create Resource Hierarchy]** メニューを終了します。

# 6.2.5.2. DB2 リソース階層の削除

プライマリーサーバーで次の作業を行います。LifeKeeper 構成内のすべてのサーバーから DB2 リソースを削除するには、次の手順を実行します。

1. **[LifeKeeper GUI]** メニューから **[Edit]** を選択し、次に **[Resource]** を選択します。ドロップダウンメニューから、 **[Delete Resource Hierarchy]** を選択します。

2. DB2 リソース階層を削除するターゲットサーバーの名前を選択します。

   **注記：** グローバルリソースの左ペインまたは個々のリソースインスタンスの右ペインで右クリックして **[Delete Resource Hierarchy]** を選択した場合は、このダイアログは表示されません。

   **[次へ]** をクリックして次に進みます。

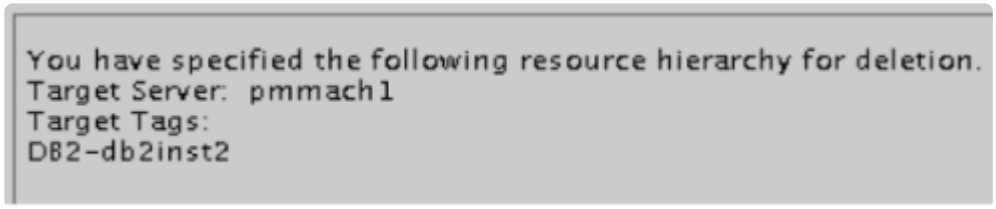
3. **削除する階層** を選択します。削除するリソース階層を特定して、強調表示にします。

   **注記：** グローバルリソースの左ペインまたは個々のリソースインスタンスの右ペインで右クリックして **[Delete Resource Hierarchy]** を選択した場合は、このダイアログは表示されません。

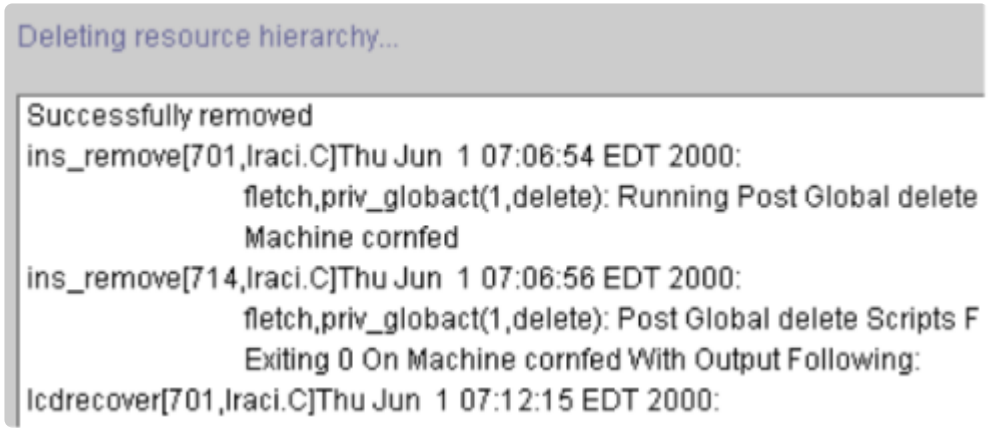
   **[Next]** をクリックして次に進みます。

4. 選択したターゲットサーバーと、削除の対象として選択した階層を確認する情報ボ ックスが表示されます。




**[Delete]** をクリックして次に進みます。

5. DB2 リソースインスタンスの削除に成功したことを示す情報ボックスが表示されます。




6. **[Done]** をクリックして、 **[Delete Resource Hierarchy]** メニューを終了します。

# 6.2.5.3. DB2 リソース階層の拡張

階層は、作成後に、クラスタ内の別のサーバに拡張する必要があります。リソースインスタンスを拡張するには、次の 3 つの方法が可能です。

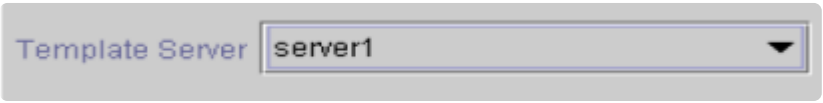
1. DB2 リソース階層の作成に成功したときに、 **[Continue]** を選択して、バックアップサーバーにリソース階層を拡張する処理に進むことができます。

2. LifeKeeper GUI の左ペインまたは右ペインで拡張されていない階層を右クリックします。

3. LifeKeeper GUI で、ドロップダウンメニューから **[Edit] > [Resource] > [Extend Resource Hierarchy]** を選択します。選択すると、 **[Extend Resource Hierarchy]** ウィザードが起動されます。 **[Extend Resource Hierarchy]** オプションの **[Accept Defaults]** ボタンは、 LifeKeeperの **[Extend Resource Hierarchy]** の既定値をよく知っていて、値の入力や確認をしないで素早く LifeKeeper リソース階層を拡張したいユーザ向けです。GUI ダイアログを使用して対話的に段階を追って LifeKeeper リソース階層を拡張する場合は、 **[Next]** ボタンを選択します。

   a. 最初に表示されるダイアログボックスでは、DB2 リソース階層が現在 In Service であるテンプレートサーバー を選択します。 テンプレートサーバー で選択するテンプレートサーバーと、次のダイアログボックスの **[拡張するタグ]** で選択する拡張するタグが、In Service (アクティブ)のリソース階層を表すことを忘れないでください。選択したテンプレートサーバーでサービス中になっていない リソースタグを選択すると、エラーメッセージが表示されます。このダイアログのドロップダウンボックスには、クラスター内の全サーバの名前が示されます。

   注記： DB2 リソース階層作成後、引き続きリソース階層の拡張作業に入った場合は、作成段階でテンプレートサーバが特定されているため、このダイアログボックスは表示されません。これは、GUI ウィンドウの左ペインの DB2 リソースアイコンまたは右ペインの DB2 リソースボックスを右クリックして、 **[Extend Resource Hierarchy]** を選択した場合も同様です。

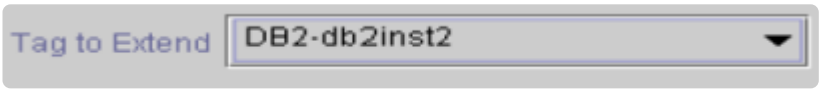
   注記： 階層拡張の途中で **[Cancel]** ボタンをクリックすると、拡張処理が取り消されます。ただし、すでにリソースを別のサーバーに拡張している場合は、明示的に拡張解除するまで、そのインスタンスの拡張は有効です。

   **[Next]** をクリックして次に進みます。

   b. **[拡張するタグ]** を選択します。これは、テンプレートサーバーからターゲットサーバーに拡張する DB2インスタンスの名前です。ウィザードのドロップダウンリストボックスには、テンプレートサーバー上に作成されているすべてのリソースが表示されます。

   注記： ここでも、DB2 リソース階層の作成後すぐにリソース階層の拡張作業に進んだ場合は、作成段階ですでにリソースのタグ名が特定されているため、このダイアログボックスは

表示されません。これは、GUI ウィンドウの左ペインの DB2 リソースアイコンまたは右ペインの DB2 リソースボックスを右クリックし て、[Extend Resource Hierarchy]を選択した場合も同様です。




**[Next]** をクリックして次に進みます。

c. DB2 リソース階層を拡張するターゲットサーバを選択します。

**[Next]** をクリックして次に進みます。

d. **[Switchback type]** ダイアログが表示されます。スイッチバックタイプによって、フェイルオーバー後、バックアップサーバーで DB2 リソースがIn Service (アクティブ)になったとき、DB2 リソースをプライマリーサーバーにスイッ チバックする方法が決まります。スイッチバックタイプは intelligent または automatic です。インテリジェントスイッチバックの場合、リソースをプライマリサーバーに戻すために管理者の操作が必要ですが、自動スイッチバックの場合 は、プライマリサーバーがオンラインに戻り、LifeKeeper との通信パスが再確立されるとすぐにスイッチバックが実行されます。

**[Next]** をクリックして次に進みます。

e. テンプレートの優先順位を選択または入力します。これはサーバーで現在 In Service の DB2 階層の優先順位です。優先順位は、1～999 の範囲で未使用の値が有効で、小さい数字ほど優先順位が高くなります（数字 1 が最高の優先順位に相当し ます）。拡張操作時には、別のシステムで既に使用中の優先順位をこの階層に対 して指定することはできません。デフォルト値を推奨します。

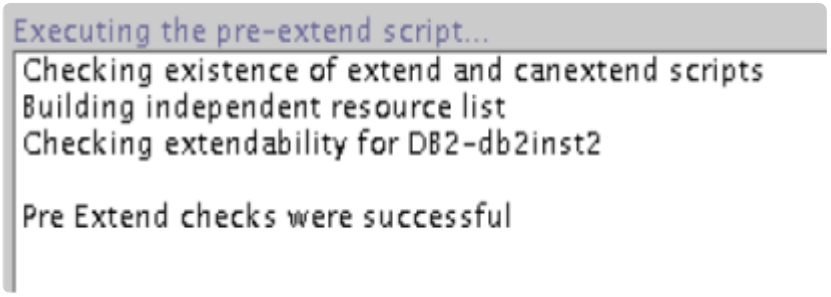
**注記：** このフィール ドは階層を最初に拡張するときだけ表示されます。

**[Next]** をクリックして次に進みます。

f. ターゲットの優先順位を選択または入力します。これは、別サーバーにある同等の階層に対する新しく拡張する DB2 階層の優先順位です。1～999 の範囲の未使用の値が有効で、リソースの連鎖フェイルオーバシーケンスにおけるサーバーの優先順位を示します。数値が小さいほど優先順位は高くなります（1 は最高の優先 順位を表わします）。LifeKeeper のデフォルトでは、階層が作成されたサーバーに「1」が割り当てられることに注意してください。優先順位は連続している必要はありませんが、特定のリソースについて 2 つのサーバーに同じ優先順位を割り当てることはできません。

**[Next]** をクリックして次に進みます。

g. 環境がチェックされ、このリソースを拡張するのに必要な要件がすべて満たされていることを示す情報ボックスが表示されます。満たされていない要件がある場合、 **[Next]** ボタンが使用できなくなり、 **[Back]** ボタンが有効になります。




リソースの拡張の設定を変更する場合は、 **[Back]** ボタンをクリックします。 後でリソースを拡張する場合は、 **[Cancel]** をクリックします。リソース階層の拡張を実行する場合は、 **[Next]** をクリックします。

**[終了]** をクリックして、DB2 リソースインスタンスの拡張に成功したことを確認します。

4. **[完了]** をクリックして、 **[Extend Resource Hierarchy]** メニューを終了します。

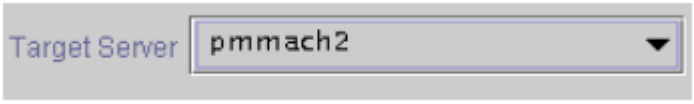
**注記：** 必ず両方のサーバーで新しいインスタンスの機能をテストしてください。

# 6.2.5.4. DB2 階層の拡張解除

1. LifeKeeper GUI メニューで、 **[Edit] > [Resource] > [Unextend Resource Hierarchy]** を選択します。

2. リソースが現在サービス中（アクティブ）のサーバは選択できません。

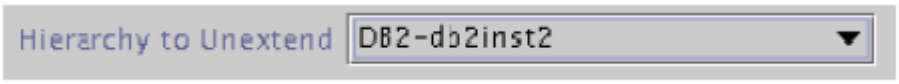
   **注記：** グローバルリソースの左ペインまたは個々のリソースインスタンスの右ペ インで右クリックして **[Unextend]** を選択した場合は、このダイアログは表示されません。

   **[次へ]** をクリックして次に進みます。

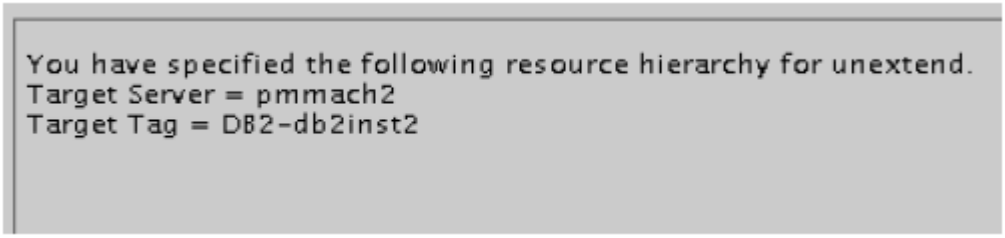
3. **[拡張解除する階層]** を選択します。

   **注記：** グローバルリソースの左ペインまたは個々のリソースインスタンスの右ペインで右クリックして **[Unextend]** を選択した場合は、このダイアログは表示されません。

   **[Next]** をクリックして次に進みます。

4. 拡張解除するために選択したターゲットサーバーと DB2 リソース階層を確認する情報ボックスが表示されます。

   **[Unextend]** をクリックします。

5. DB2 リソースの拡張解除に成功したことを示す別の情報ボックスが表示されます。

6. **[完了]** をクリックして、 **[Unextend Resource Hierarchy]** メニューを終了します。

# 6.2.5.5. DB2 リソース階層のテスト

DB2 リソース階層のテストは、手動でサーバーを切り替えて、プライマリサーバからバックアップサーバーへのリソースインスタンスのフェイルオーバーをシミュレートすることによって行えます。

## LifeKeeper GUI からの手動切り替えの実行

LifeKeeper GUI で **[Edit] > [Resource] > [In Service]** を選択すると、手動切り替えを開始できます。たとえば、バックアップサーバーで In Service リクエストが実行されると、DB2 リソース階層がバックアップサーバー側で In Service になり、プライマリサーバー側では Out of Service になります。この時点で、元のバックアップサーバーがプライマリサーバーになり、元のプライマリサーバーがバックアップサーバーになります。

**Out of Service** を実行すると、リソース階層は他のサーバーで In Service にならずに停止されます。再同期中にリソースが Out of Service になった場合、サービス中に戻すことができるのは休止になったのと同じサーバ上のみです。

**重要：** バックアップサーバーでリソース階層を In Service にしたあとは、データベースに接続してみる必要があります。特に、RAW デバイスをテーブルスペースコンテナとして使用している場合は接続を試みる必要があります。この作業は、バックアップサーバーですべてのディスクパーティションが認識されており、RAW バインディングが正しく確立されていることを確認するために必要です。

これは、すべてのディスクパーティションがバックアップサーバー側から見えるようにして、RAW バインディングを正しく確立するために必要です。バックアップサーバーで RAW バインディングが確立されていない場合、その原因として最も可能性が高いのは、新しいパーティションをプライマリサーバーに作成して構成に追加したのに、バックアップサーバー側のパーティションテーブルが更新されていないためです。

解決策は、パーティションテーブルが正しくアップデートされるように、バックアップサーバーを再起動することです。

# 6.2.6. DB2 トラブルシューティング

| 症状 | 考えられる要因 |
|---|---|
| 1 つ以上の DB2 EEE パーティション サーバーの開始に失敗する | **db2nodes.cfg** ファイルのポート番号が */etc/services* ファイルで設定されている範囲を超えている可能性があります。 **db2nodes.cfg** ファイルで設定されているポート番号を調べて、 */etc/services* のポート番号の範囲におさまっているか確認します。 |
| LifeKeeperの *"In-Service"* または *"Out-of- Service"* 操作でハングする | DB2 の環境変数 **DB2_NUM_FAILOVER_NODES** が適切に設定されていない可能性があります。設定内のすべてのサーバについて、この環境変数を、インスタンス内のパーティション総数と同じ値に設定します。<br><br>例： **db2set DB2_NUM_FAILOVER_NODES =**<インスタンス内のパーティション数> |
| LifeKeeperの *"In-Service"* 操作がハングする | DB2 管理サーバーで **dasupdt** コマンドが実行されていない可能性があります。 DB2 管理サーバーで **dasupdt** コマンドが正常に実行されたことを確認します。 |
| LifeKeeper の最初のスイッチオーバー操作が失敗する | DB2 Fenced ユーザーがバックアップ サーバー上に作成されていない可能性があります。 指定されたインスタンスの DB2 Fenced User が、プライマリーと同じユーザーおよびグループ ID で存在することを確認してください。 保護されたインスタンスが管理サーバーグループのメンバーでもあることを確認します。 |
| 既存の DB2 リソース階層に新しいノードを追加する必要がある | 既存の LifeKeeper DB2 リソース階層に新しいノードを追加する方法の詳細については、ノードユーティリティのマニュアルページを参照してください。 |
| 管理サーバーが起動しない | 別の管理サーバーが、指定されたポートで既に実行されていないか確認します。 |
| DB2 リソース階層の作成に時間がかかる | 大規模な DB を持つ DB2 インスタンスを保護するには、リソースの作成に時間がかかる場合があります。 リソースを作成する前にアクティブ化してください。 **データベースを起動するコマンド**: `db2 activate db <db_name>`. 以下の考慮事項を参照してください。 |
| スイッチオーバー後に DB2 データベースがアクティブにならない | DB2 Kit は、フェイルオーバー後にデータベースの暗黙的なアクティブ化を使用します。 これはパフォーマンス上の理由から、フェイルオーバーのスイッチオーバー中にキットが各データベースを明示的にアクティブ化しないことを意味します。 代わりに、データベースへの最初の接続でそのデータベースがアクティブ化されます。 |

## 考慮事項

**注記:** このキットはもともと **implicit activation** 用に設計されました。 これは、データベースへの最初の接続がそのデータベースをアクティブにすることを意味します。 スイッチオーバーまたはフェールオーバーの一部として **explicit activation** が必要な場合は、必要に応じてデータベースをアクティブ化するための Generic Application リソースを作成できます。 リソースを作成する前に明示的なアクティブ化を使用すると、DB2 リソースおよび関連するファイル システムの依存関係を作成するのに必要な時間が短縮され

ることが示されています。

データベースを暗黙的(implicitly)にアクティブ化するコマンドは次のとおりです。

- `db2 "connect to <dbname>".`

Generic Application リソース内でデータベースを検索/アクティブ化するには、次のコマンドを使用してください。

- `su - db2user -c 'db2 activate db ${dbname}'`

この情報は、IBM サポート ページから取得したものです。 db2 の使用法について詳しくは、IBM サポートを参照してください。

## エラーメッセージ

DB2 Kit の使用中に発生する可能性のあるすべてのメッセージのリストについては、DB2 Recovery Kit メッセージカタログ を参照してください。

# 6.2.7. 付録： DB2 で RAW I/O を使用する設定

DB2 インスタンスが LifeKeeper で保護されるように、DB2 で RAW I/O デバイスを設定するにはいくつか要件があります。

## 要件

- Linux OS が RAW I/O デバイスをサポートしている必要があります。ほとんどのディストリビューションでは、カーネル 2.4 でサポートされましたが、カーネル 2.2 からサポートしているディストリビューションもあります。

- すべての RAW I/O デバイスを共有ディスク･パーティションにバインドする必要があります。共有 SCSI ディスク･パーティションの数が必要になります。正確な数は、 RAW I/O デバイスに置くテーブルスペースの数によって決まります（RAW デバイスにテーブルスペースを書き込む場合のガイドラインについては、DB2 のドキュメ ントを参照してください）。

- DB2 Version 7.1 Fix Pack 3 以降または DB2 Version 8 もしくはそれ以降が必要です。

## RAW I/O の設定手順

以下の手順 1～4 は、『IBM Db2 Universal Database リリース･ノート バージョン 7.2/バー ジョン 7.1 フィックスパック 3』の第 7.3.1.1 節（「Linux でのロー I/O の使用法」）から抜き出したものです。 この例では、使用する RAW パーティションは/dev/sda5 です。ここ には重要なデータを置かないでください。

手順 4 または手順 5 は、複数論理ノードを使用しているかどうかで異なることに注意してください。

1. このパーティションの 4096 バイト･ページの数を計算し、必要に応じて端数を切り捨てます。例：

   例:

   ```
   # fdisk /dev/sda

   Command (m for help):p

   Disk /dev/sda:255 heads, 63 sectors, 1106 cylinders

   Units = cylinders of 16065 * 512 bytes
   ```

| Device Boot | Start | End | Blocks | ID | System |
|---|---|---|---|---|---|
| dev/sda1 | 1 | 23 | 4200997 | 83 | Linux |
| /dev/sda2 | 524 | 1106 | 4682947+ | 5 | Extended |
| /dev/sda5 | 524 | 1106 | 4682947 | 83 | Linux |

   ```
   Command (m for help):q
   ```

```
#
```

/dev/sda5 のページ数：

```
num_pages = floor( ((1106-524+1)*16065*512)/4096 )

num_pages = 11170736
```

2. このパーティションに、未使用の RAW デバイスをバインドします。これはマシンを再起動するたびに実行する必要があり、root アクセスを必要とするため、システム初期化ファイル（rc.local または boot.local）に RAW バインディングを追加することもできます。 このバインディングは、階層を LifeKeeper の保護下においたあと、削除する必要があります。LifeKeeper は、LifeKeeper が保護している RAW I/O デバイスの RAW バインディングを再確立します。

   次のように、raw -qa を使用して、どのロー・デバイス・ノードが使用中か確認します。

   ```
   raw /dev/raw/raw1 /dev/sda5

   /dev/raw/raw1:bound to major 8, minor 5
   ```

3. RAW デバイス・コントローラとディスク・パーティションにグローバル読み取りパーミッションを設定します。RAW デバイスにグローバル読み書きパーミッションを設定します。

   ```
   # chmod a+r /dev/rawctl

   # chmod a+r /dev/sdb1

   # chmod a+rw /dev/raw/raw1
   ```

4. **重要：** この手順は、DB2 EE を使用している場合、または、DB2 EEE 構成でフェイルオーバ後も複数論理ノード（MLN）を決して実行しない場合にだけ、あてはまります。 ある時点で MLN を実行する可能性がある場合は、手順 5 に進みます。

   ディスク・パーティションではなくロー・デバイスを指定して、DB2 にテーブルスペースを作成します。

   例：

   ```
   CREATE TABLESPACE dms1

   MANAGED BY DATABASE

   USING (DEVICE '/dev/raw/raw1' 11170736)
   ```

   ロー・デバイスのテーブルスペースは、DB2 がサポートしている他のページ・サイズでもサポートされます。

5. **重要：** この手順は、MLN を実行している場合、またはフェイルオーバ後のある時点 で MLN を実行

する場合に行う必要があります。

ディスク・パーティションではなくロー・デバイスを指定して、DB2 にテーブルスペースを作成し、各 DB2 インスタンス・パーティションに異なる RAW I/O デバイス を指定します。

例：

```
CREATE TABLESPACE dms1

MANAGED BY DATABASE

USING (DEVICE '/dev/raw/raw1' 11170736) on NODE (NODENUM)

USING (DEVICE '/dev/raw/<異なる RAW デバイス・ノード>' ####### ) on NODE
(NODENUM)
```

**注記：** 各 DB2 ノード（データベース・パーティション・サーバー）が異なる RAW I/O デバイスを使用する必要があるため、ON NODE を使用する必要があります。 フェイルオーバが正しく動作するように、異なるマシンでノードが実行されている場合でもON NODE を指定する必要があります。

# 6.3. Recovery Kit for EC2™ 管理ガイド

Recovery Kit for EC2 は、障害の発生したプライマリーサーバから Elastic IP をバックアップサーバーに復旧する仕組みを提供します。また、複数の Availability Zoneで IP Recovery Kit を動作可能にする仕組みも提供します。

Recovery Kit for EC2 の定義、シナリオ、および運用の比較と詳細については、Recovery Kit for EC2™ 運用の原則 を参照してください。

## LifeKeeper ドキュメンテーション

以下は、SIOS Technology Corp が発行している LifeKeeper for Linux 関連ドキュメントの一覧です。

- LifeKeeper for Linux テクニカルドキュメンテーション
- LifeKeeper for Linux リリースノート
- SIOS Technology Corp. ドキュメンテーション

詳細については、Amazon Elastic Compute Cloud (EC2) ドキュメント を参照してください。

「Amazon Web Services」、「Powered by Amazon Web Services」のロゴ、「AWS」、「Amazon EC2」、「EC2」、「Amazon Elastic Compute Cloud」、「Amazon Route 53」、「Amazon Virtual Private Cloud」および「Amazon VPC」は、米国その他の国における Amazon.com, Inc. またはその関連会社の商標です。なお、本文および図表中では、「™」、「®」は明記しておりません。(一部記載がございます。)

# 6.3.1. Recovery Kit for EC2™ 運用の原則

Recovery Kit for EC2™ には機能が 2 つあります。

1. ルートテーブルシナリオ (バックエンドクラスタ) は、LifeKeeper が保護する IP リソースに Amazon VPC™ 内のクライアントから接続できるように、ルートテーブルを管理します。

2. Elastic IP シナリオ (フロントエンドクラスタ) は、インターネットから使用できる Elastic IP を管理します。

## ルートテーブルシナリオ (バックエンドクラスタ):

ルートテーブルの管理と運用について理解しやすいように、図 1 に示すシナリオについて考えてみましょう。

この設定例は、1 つの Amazon VPC™ と 2 つの Availability ゾーン (AZ) で構成されています。

各 AZ にサブネットが 2 つあります。

- 1 つ目のサブネット (以下「パブリックサブネット」) は、ルートテーブル別のインターネットゲートウェイを経由してインターネットに接続します (10.0.1.0/24 および 10.0.3.0/24 のルートテーブルを参照)。

- 2 つ目のサブネット (以下「プライベートサブネット」) は、ルートテーブル別のNATインスタンスを経由してインターネットに接続します (10.0.2.0/24 および 10.0.4.0/24 のルートテーブルを参照)。

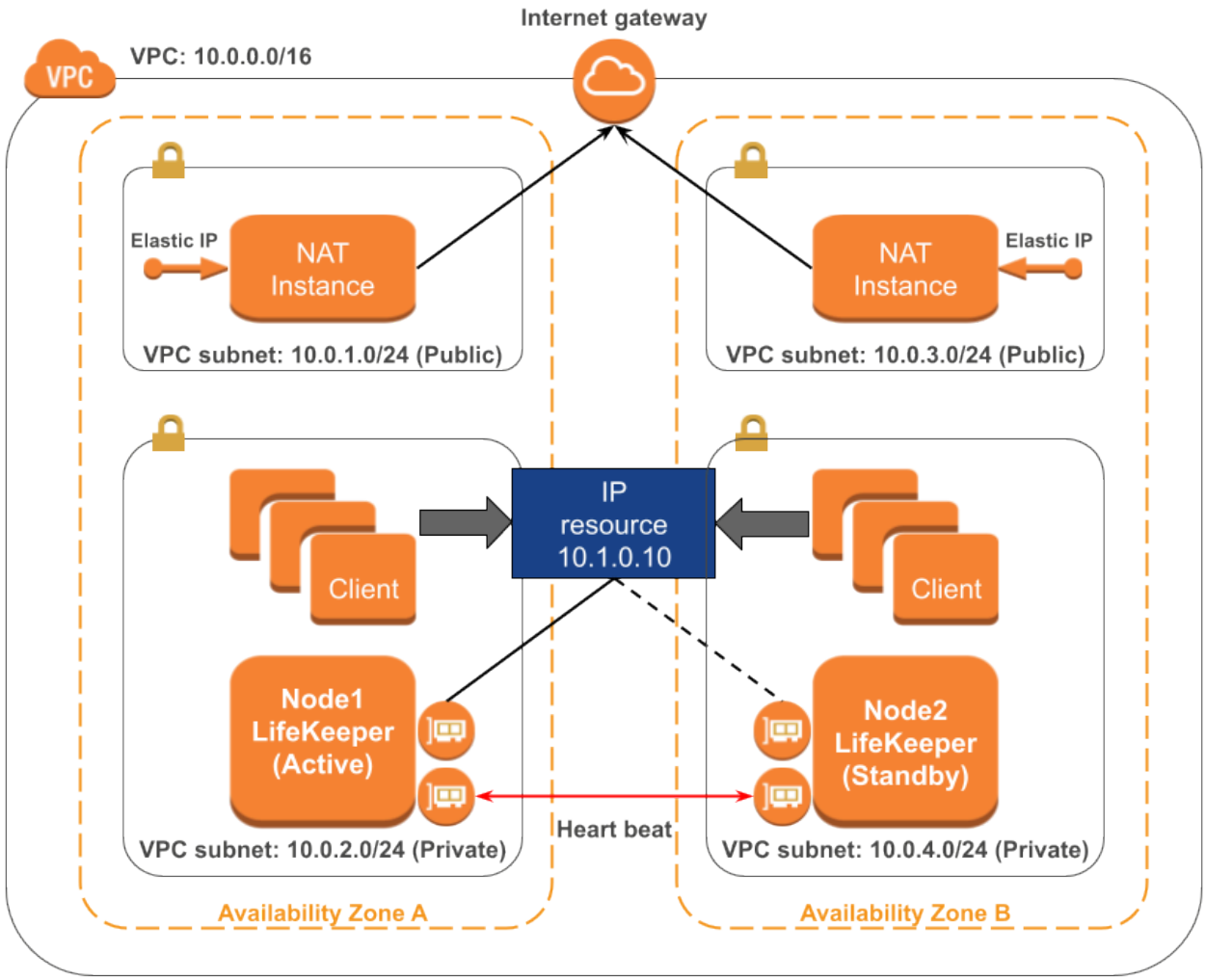
各パブリックサブネットには、NAT の Elastic IP を割り当てる EC2 インスタンス (以下「NAT インスタンス」) が 1 つあります。

各プライベートサブネットには、LifeKeeper のアクティブ / スタンバイ (以下それぞれ「Node1」および「Node2」) の EC2 インスタンスが 1 つあり、Node1/Node2 により保護されたアプリケーションを使用するクライアントが複数あります。

Node1/Node2 はそれぞれ、Elastic Network Interface (ENI) を 2 つ持ちます。

各インスタンスと各ノードとの間で通信が可能なように、ネットワーク ACL とセキュリティグループを設定します。

図 1: ルートテーブルシナリオ




## 10.0.1.0/24 および 10.0.3.0/24 のルートテーブル

| 接続先 | ターゲット | 注記 |
|---|---|---|
| 10.0.0.0/16 | ローカル | デフォルト |
| 0.0.0.0/0 | インターネットゲートウェイ | インターネットに接続するには、Elastic IP を割り当てる必要があります。 |

## 10.0.2.0/24 のルートテーブル

| 接続先 | ターゲット | 注記 |
|---|---|---|
| 10.0.0.0/16 | ローカル | デフォルト |
| 10.1.0.10/32 (IP リソース) | LifeKeeper のアクティブなノード上の Elastic Network Interface (ENI) | このターゲットは、スイッチオーバ中に Recovery Kit for EC2 により更新されます。 |
| 0.0.0.0/0 | NAT インスタンス (10.0.1.0) | NAT 経由でインターネットに接続 |

### 10.0.4.0/24 のルートテーブル

| 接続先 | ターゲット | 注記 |
|---|---|---|
| 10.0.0.0/16 | ローカル | デフォルト |
| 10.1.0.10/32 (IP リソース) | LifeKeeper のアクティブなノード上の Elastic Network Interface (ENI) | このターゲットは、スイッチオーバ中に Recovery Kit for EC2 により更新されます。 |
| 0.0.0.0/0 | NAT インスタンス (10.0.3.0) | NAT 経由でインターネットに接続 |

リソースのスイッチオーバが実行されると、LifeKeeper は Node1 の IP リソースを Out of Service にします。各プライベートサブネット内の 10.1.0.10/32 のターゲットエントリは、Node2 の ENI を反映するように更新されます。Node2 の IP リソースが In-Service になります。このため、IP アドレス 10.1.0.10 へのトラフィックは、プライベートサブネット内でのルートテーブル設定の変更により効果的に Node2 に転送されます。

パブリックサブネットを含む他のサブネットから IP アドレス 10.1.0.10 にアクセスする必要がある場合、それぞれのサブネットのルートテーブルのエントリに接続先 10.1.0.10/32 のルートを追加してください。LifeKeeperは、VPC内のルートテーブルで接続先が "10.1.0.10/32" になっている全てのエントリを制御します。

## Elastic IP シナリオ (フロントエンドクラスタ):

Elastic IP の管理と運用について理解しやすいように、図 2 に示すシナリオについて考えてみましょう。

この設定例は、1 つの Amazon VPC™ と 2 つの Availability ゾーン (AZ) で構成されています。

各 AZ にサブネットが 1 つあります。

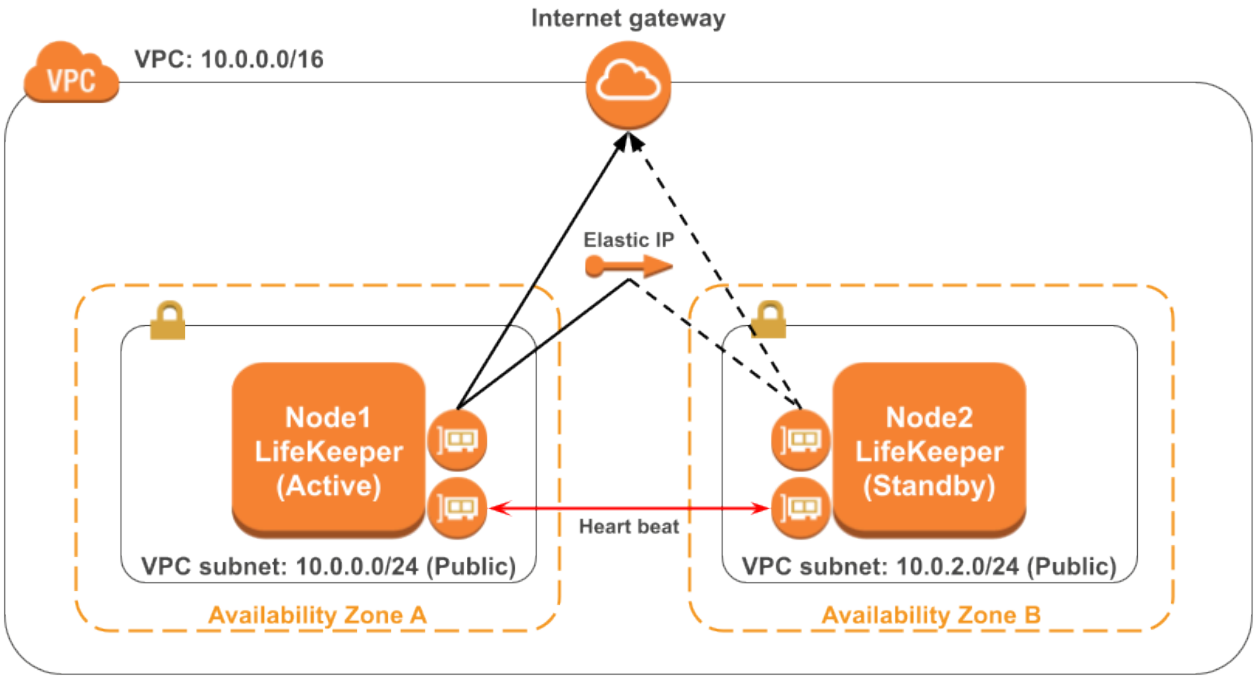
各サブネットは、ルートテーブル別のインターネットゲートウェイを経由して、インターネットに接続します。

サブネットには、LifeKeeper のアクティブ / スタンバイ (以下それぞれ「Node1」および「Node2」) の EC2 インスタンスが 1 つあります。

Node1/Node2 はそれぞれ、Elastic Network Interface (ENI) を 2 つ持ちます。

各ノード間で通信が可能なように、ネットワーク ACL とセキュリティグループを設定します。

図 2: Elastic IP シナリオ




システム管理者が、フロントエンドクラスタの Elastic IP アドレスを ENI に割り当てます。

Node1 をリソースのプライマリサーバと仮定すると、システム管理者は[リソース階層の作成](#) セクションで説明しているウィザードを使用して、Node1 上に EC2 リソース階層を作成します。

リソースのスイッチオーバが実行されると、Recovery Kit for EC2 は Node1 の ENI と Elastic IP との関連付けを解除します。Recovery Kit for EC2 は、Elastic IP が Node2 の ENI と関連付けられているかどうかを調べ、関連付けられていない場合は、Elastic IP を ENI に関連付けます。このため、インターネット上のクライアントは、スイッチオーバ後に Elastic IP 経由で Node2 に接続できます。

注記：EC2インスタンスを制御するために、スタンバイノードもエンドポイントにアクセスできる必要があります。即ち、VPC 外部への接続が必要ですので、ご注意ください。詳しくは「[Recovery Kit for EC2™ の要件](#) 」を参照してください。
なお、PrivateLink を利用すればエンドポイントへのアクセスにパブリック IP アドレスを必要としません。詳しくは「[VPC エンドポイント](#) 」を参照してください。

# 6.3.2. Recovery Kit for EC2™ 処理概要

各動作は Amazon Web Services より提供されている AWS CLI を元に実施します。

Recovery Kit for EC2 はリソース作成時に指定したリソースタイプにより処理が異なります。ご利用のリソースタイプに合わせてご参照ください。

- ルートテーブルシナリオ（バックエンドクラスタ）
- Elastic IP シナリオ（フロントエンドクラスタ）

## 起動処理 (restore)

EC2 リソースへ起動処理が実施された際、各シナリオで以下の動作を実施します。

### ルートテーブルシナリオ（バックエンドクラスタ）

1. RouteTable の情報を取得します。
2. RouteTable の更新を行います。

### Elastic IP シナリオ（フロントエンドクラスタ）

1. Elastic Network Interface (ENI) を取得します。
2. ENI に Elastic IP を関連付けします。
3. LinkUp しているかを確認します。
   LinkUp していない場合は、LinkUp させます。
4. Elastic IP と ENI との関連付けを確認します。

AWS CLI から応答がない場合は EC2_RESTORE_TIMEOUT の値（デフォルト値：300秒）でタイムアウトします。

## 停止処理 (remove)

EC2 リソースへ停止処理が実施された際、各シナリオで以下の動作を実施します。

### ルートテーブルシナリオ（バックエンドクラスタ）

本シナリオでは、EC2リソースに関する処理はありません。

### Elastic IP シナリオ（フロントエンドクラスタ）

ENI ID と Elastic IP の関連付けを解除します。
関連付けを解除後に、解除されたことを確認します。

AWS CLI から応答がない場合は EC2_REMOVE_TIMEOUT の値（デフォルト：300秒）でタイムアウトします。

# 監視処理 (quickCheck)

EC2 リソースへ監視処理が実施された際、各シナリオで以下の動作を実施します。

## ルートテーブルシナリオ（バックエンドクラスタ）

アクティブなノード上の ENI が正しいかどうかを監視します。

## Elastic IP シナリオ（フロントエンドクラスタ）

アクティブなノード上の ENIと Elastic IP の関連付けや LinkUp の有無を監視します。

AWS CLI からの応答がない場合は EC2_QUICKCHECK_TIMEOUT の値（デフォルト：100秒）でタイムアウトします。ただし、タイムアウトではリソース障害とは判断しません。
AWS CLI を利用した取得した結果と、ノード上のメタデータの情報を元にして正常ではないと判断した場合は、リソース障害となります。

# 回復処理 (recover)

EC2 リソースへ回復処理が実施された際、各シナリオで以下の動作を実施します。

## ルートテーブルシナリオ（バックエンドクラスタ）

起動処理と同じ処理を実行します。

## Elastic IP シナリオ（フロントエンドクラスタ）

起動処理と同じ処理を実行します。

AWS CLI から応答がない場合は EC2_RECOVER_TIMEOUT の値（デフォルト：300秒）でタイムアウトします。

# Recovery Kit for EC2 のパラメーターについて

処理概要の中で使用されている Recovery Kit for EC2 のパラメーターについては、EC2パラメーター一覧 をご参照ください。

# 6.3.3. Recovery Kit for EC2™ の要件

Recovery Kit for EC2 のインストール / アンインストールを試みる前に、Amazon Web Service のソフトウェア要件と Recovery Kit for EC2 のインストール / アンインストールの手順を理解しておく必要があります。

## Amazon Web Service およびソフトウェアの要件

Recovery Kit for EC2 をインストールして設定する前に、使用している設定が以下の要件を満たしていることを確認してください。

### Amazon Virtual Private Cloud (VPC):

- この Recovery Kit を使用するには、VPC を AWS 内に設定する必要があります。
- 異なる Availability Zone (AZ) に作成された 2 つ以上のサブネット
- 各サブネットに、関連するルートテーブルがあります。
- パブリック (フロントエンド) クラスターを設定する場合、Elastic IP を 1 つ以上割り当てる必要があります。

### Amazon Elastic Compute Cloud (EC2):

- この Recovery Kit を使用するには、EC2 インスタンスが 2 つ以上必要です。
- インスタンスは、各サブネット上で関連付けられます。
- インスタンスは、Elastic Network Interface (ENI) に接続されます。
- ルートテーブル (バックエンドクラスター) リソースを作成する場合、各インスタンスで使用するネットワークインターフェースは、ネットワークの送信元/送信先チェックが無効になっている必要があります。
- AWS Command Line Interface (AWS CLI) を全ての EC2 インスタンスにインストールする必要があります。インストール方法は、「AWS Command Line Interface のインストール」を参照してください。
- 全ての EC2 インスタンスで、Amazon EC2 サービスのエンドポイント（AWS のリージョンとエンドポイント）にプロトコル HTTP および HTTPS を使用してアクセスできる必要があります。EC2 および OS の設定を適切に行ってください。
- Amazon EC2 インスタンスのメタデータを取得するため、IPアドレス 169.254.169.254 にプロトコル HTTP でアクセスできる必要があります。
- AWS CLI を用いているため、TCP ポート 443 でのアウトバウンド接続を有効にする必要があります。
- Auto Recovery 機能は LifeKeeper の回復機能と競合する可能性があるので、併用は推奨しません。

  **注記:** 設定ファイル `/etc/default/LifeKeeper` のパラメーター PATH に AWS CLI 実行ファイルのパスが設定されていない場合、PATH に AWS CLI 実行ファイルのパスを追加してください。

### AWS Identity and Access Management (IAM):

LifeKeeper が AWS を操作するために、以下のアクセス権限を持った IAM ユーザーもしくは IAM ロールが必要です。EC2 インスタンスに適切な権限を持つ IAM ロールをアタッチするか、root ユーザーの AWS CLI プロファイルに権限のある IAM ユーザーを登録してください。AWS の IAM ロー

ルと AWS CLI の詳細については AWS のユーザーガイド、Amazon EC2 の IAM ロール および AWS CLI の開始方法 を参照してください。

**ルートテーブル (バックエンド) 構成:**

- ec2:DescribeRouteTables
- ec2:ReplaceRoute
- ec2:DescribeNetworkInterfaceAttribute
- ec2:ModifyNetworkInterfaceAttribute

**Elastic IP (フロントエンド) 構成:**

- ec2:DescribeAddresses
- ec2:AssociateAddress
- ec2:DisassociateAddress

## LifeKeeper ソフトウェア:

各サーバに同じバージョンの LifeKeeper ソフトウェアとパッチをインストールする必要があります。具体的な LifeKeeper の要件については、LifeKeeper for Linux テクニカルドキュメンテーション および LifeKeeper for Linux リリースノート を参照してください。

### LifeKeeper Recovery Kit for EC2:

各サーバに同じバージョンの Recovery Kit for EC2 ソフトウェアとパッチをインストールする必要があります。

### LifeKeeper IP Recovery Kit:

ルートテーブル (バックエンドクラスタ) を保護する目的で Recovery Kit for EC2 を使用する場合、各サーバに同じバージョンの LifeKeeper for Linux IP Recovery Kit ソフトウェアとパッチをインストールする必要があります。

**注記:** 最新リリースの互換性と発注情報については、 LifeKeeper for Linux リリースノート を参照するか、営業担当者に問い合わせてください。LifeKeeper Recovery Kit for EC2 のインストール / アンインストール方法の具体的な手順については、LifeKeeper for Linuxインストールガイド を参照してください。

Recovery Kit for EC2を使う場合には、Quorum/Witnessの使用を推奨します。
詳細については、こちら を参照してください。

# 6.3.4. Recovery Kit for EC2™ の設定

必要とする保護と柔軟性が得られるように LifeKeeper を設定するには、設定要件を把握している必要があります。設定を適切にプランニングするには、Amazon、Amazon Virtual Private Cloud (VPC)、Amazon Elastic Compute Cloud (EC2)、およびユーザシステムの階層設定オプションを理解する必要があります。このセクションでは、設定のプランニングに加え、Recovery Kit を設定するために必要な特定の作業についても説明します。

## Amazon EC2 特有の設定上の考慮事項

Recovery Kit for EC2 を正しく設定するために、以下のトピックを見直して、設定作業を完了するために必要な情報を用意してあることを確認してください。

- ユーザーシステムのセットアップ

更なる設定上の考慮事項については、以下のトピックを参照してください。

- EC2™ リソース監視と設定に関する考慮事項
- EC2™ ローカルリカバリと設定に関する考慮事項

## Amazon EC2 特有の設定上の考慮事項

このセクションでは、EC2 リソースの以下の設定タスクについて説明します。設定タスクは EC2 リソースインスタンスに特有のものであり、Recovery Kit ごとに異なります。

- EC2™ リソース階層の作成：アプリケーションリソース階層を LifeKeeper クラスターに作成します。
- EC2™ リソース階層の削除：リソース階層を LifeKeeper クラスター内のすべてのサーバーから削除します。
- EC2™ リソース階層の拡張：リソース階層をプライマリーノードからセカンダリーノードへ拡張します。
- EC2™ リソース階層の拡張解除：リソース階層を LifeKeeper クラスター内の 1 つのサーバーから拡張解除 (削除) します。
- EC2 設定のプロパティの確認および編集: EC2 リソースの設定を表示し、その一部を修正することが可能です。
- Recovery Kit for EC2™ のチューニング Recovery Kit for EC2 の動作を調整します。

以下の作業は、LifeKeeper for Linux テクニカルドキュメンテーション の管理 セクションに記載されています。これらは、すべての Recovery Kit で同じ手順を使用する共通の作業です。

- リソース依存関係の作成：既存のリソース階層と別のリソースインスタンスとの間に親子の依存関係を作成し、クラスター内のすべてのサーバーに依存関係の変更を反映します。
- リソース依存関係の削除：リソースの依存関係を削除して、クラスター内のすべてのサーバーに依存関係の変更を反映します。
- In Service：リソース階層を特定のサーバーで In Service にします。
- Out of Service：リソース階層を特定のサーバーで Out of Service にします。
- プロパティの表示 / プロパティの編集：特定のサーバーでリソース階層のプロパティを表示または編集します。

このセクションの残りの部分では、LifeKeeper GUI の [Edit] メニューから作業を選択することによって、Recovery Kit を設定する方法を説明します。設定作業はツールバーから選択することもできます。

- 状況表示ウィンドウのリソース階層ツリー (左側のペイン) のグローバルリソースを右クリックすると、[Edit] メニューと同じドロップダウンメニューの選択項目が表示されます。これは階層がすでに存在している場合にだけ可能な方法です。
- 状況表示ウィンドウのリソース階層表 (右側のペイン) のリソースインスタンスを右クリックすると、サーバーおよび特定リソースの状況に応じて、リソース階層の作成を除くすべての設定作業を実行できます。

# 6.3.4.1. EC2™ イベントテーブル

次の表には、EC2™ 固有のイベントの Recovery Kit と関連するトラップ番号が含まれています。
LK_NOTIFY_ALIAS が設定されている場合、イベントはメール通知を生成します。

| LifeKeeper イベント/説明 | Trap # | Object ID |
|---|---|---|
| **AWS API failure** | 180 | .1.3.6.1.4.1.7359.1.0.180 |

# 6.3.4.2. Recovery Kit for EC2™ のチューニング

Recovery Kit for EC2™ で設定可能なパラメータはパラメータ一覧 を参照してください。

# 6.3.4.3. EC2™ リソース階層の作成

プライマリーサーバーからリソースインスタンスを作成するには、以下の手順を完了する必要があります。

1. LifeKeeper GUI メニューから **[Edit]** を選択し、次に **[Server]** を選択してください。ドロップダウンメニューから、 **[Create Resource Hierarchy]** を選択してください。

2. ダイアログボックスが表示され、クラスタ内にインストール済みの認識されている Recovery Kit がすべて、ドロップダウンリストボックスに表示されます。ドロップダウンリストから **[Amazon EC2]** を選択し、 **[Next]** をクリックしてください。

3. 以下の情報を入力するようにプロンプトが表示されます。(ダイアログボックスで [Back] ボタンが有効な場合は、前のダイアログボックスに戻ることができます。これは、以前に入力した情報を訂正する必要がある場合に特に役立ちます。)

   **注記:** 階層作成の途中で [Cancel] ボタンをクリックすると、作成処理全体が取り消されます。

| フィールド | ヒント |
|---|---|
| Switchback Type | フェイルオーバ後にサーバが復帰したときに、このサーバに EC2 インスタンスを戻す方法を指定します。[intelligent] または [automatic] を選択できます。<br><br>• [intelligent] の場合、インスタンスをプライマリ / オリジナルサーバにスイッチバックするときに管理者の介入が必要になります。<br><br>• [automatic] の場合、プライマリサーバがオンラインに戻り、LifeKeeper コミュニケーションパスを再確立した直後に自動的にスイッチバックが行われます。<br><br>**注記:** スイッチバックタイプは、必要な場合、[Resource Properties] ダイアログボックスの [General] タブで後から変更できます。 |
| Server | [Server] で EC2 リソースの作成先のサーバ (通常、これをプライマリサーバまたはテンプレートサーバと呼ぶ) を選択します。クラスタ内のすべてのサーバがドロップダウンリストに表示されます。 |
| EC2 Resource type | EC2 Recovery Kit は、ルートテーブルと Elastic IP という 2 通りの AWS リカバリシナリオに対する保護を提供します。<br><br>ルートテーブルシナリオはローカル仮想 IP アドレスと組み合わせて使用し、通常はバックエンドクラスタに使用します。<br><br>Elastic IP シナリオは Elastic IP の保護に使用し、通常はフロントエンドクラスタに使用します。<br><br>使用する EC2 タイプを選択してください。 |

| | |
|---|---|
| IP resource | このフィールドは、ルートテーブルシナリオでのみ表示され設定されます。IP リソースを選択してください。これは LifeKeeper が保護する仮想 IP アドレスであり、VPC のルートテーブルアドレスで設定されます。 **注記**：リストには ISP および IPv4 ベースの IP リソースのみが表示されます。 |
| Disable SourceDestChecks | このフィールドは、ルートテーブルシナリオで、送信元/送信先チェックが有効のときのみ表示されます。送信元/送信先チェックを無効にする場合は Yes を選択してください。 |
| Network Interface | このフィールドは、Elastic IP シナリオでのみ表示され設定されます。Elastic IP に関連付けるネットワークインターフェースを選択してください。 |
| Elastic IP | このフィールドは、Elastic IP シナリオでのみ表示され設定されます。ネットワークインターフェースに関連付ける Elastic IP を選択してください。 |
| EC2 Resource Tag | 作成している EC2 リソースインスタンスに固有の EC2 リソースタグ名を選択または入力します。このフィールドには、デフォルトのタグ名である ec2-<resource> が自動的に表示されます。<resource> はリソース名です。このタグは、後から変更できます。 |

4. [Create] をクリックしてください。[Create Resource Wizard] によって、EC2 リソースが作成されます。

5. この時点で情報ボックスが表示され、LifeKeeper は、EC2 リソース階層を作成するために有効なデータが指定されたかどうかを検証します。LifeKeeper が問題を検知した場合は、情報ボックスにエラーが表示されます。検証が正常に完了すると、リソースが作成されます。 **[Next]** をクリックしてください。

EC2 リソース階層が正常に作成されたことを示す情報ボックスが表示されます。リソース階層を LifeKeeper で保護するには、クラスタ内の別のサーバにそのリソース階層を拡張する必要があります。

**[Continue]** をクリックすると、[Pre-Extend configuration task] が起動されます。リソース階層を別のサーバに拡張する方法の詳細については、リソース階層の拡張 を参照してください。

ここで **[Cancel]** をクリックすると、別のダイアログボックスが表示され、後で EC2 リソース階層を別のサーバに手動で拡張して LifeKeeper の保護下に置く必要があることが警告されます。

# 6.3.4.4. EC2™ リソース階層の削除

LifeKeeper 環境のすべてのサーバーからリソース階層を削除するには、次の手順を実行してください。

1. LifeKeeper GUI メニューから **[Edit]** を選択し、次に **[Resouce]** を選択してください。ドロップダウンメニューから **[Delete Resource Hierarchy]** を選択してください。

2. EC2 リソース階層から削除するターゲットサーバの名前を選択し、 **[Next]** をクリックしてください。

   **注記:** このダイアログボックスは、いずれかのペインでリソースインスタンスを右クリックして [Delete Resource] を選択した場合には、表示されません。

3. [Hierarchy to Delete] を選択してください。削除するリソース階層を指定して強調表示にしてから、**[Next]** をクリックしてください。

   **注記:** このダイアログボックスは、左右どちらかの インでリソースインスタンスを右クリックして [Delete Resource] を選択した場合には、表示されません。

4. 選択したターゲットサーバーと、削除の対象として選択した階層を確認する情報ボックスが表示されます。 **[Delete]** をクリックして次に進んでください。

5. EC2 リソースが正常に削除されたことを確認する情報ボックスが表示されます。

6. **[Done]** をクリックして終了してください。

# 6.3.4.5. EC2™ リソース階層の拡張

階層の作成後、クラスター内の別のサーバーに拡張する必要があります。3 通りのシナリオで、テンプレートサーバーからターゲットサーバーにリソースインスタンスを拡張できます。

- リソースの作成後、[Continue] をクリックして、別のサーバにリソースを拡張します。
- 次に説明するように [Edit] メニューから [Extend Resource Hierarchy] を選択します。
- 左側または右側のペインから拡張されていない階層を右クリックします。

どのシナリオでも同じダイアログボックスが表示されます (いくつかの例外については、以下に詳細を明記)。

1. LifeKeeperGUI メニューから [Extend] ウィザードを開始する場合は、 **[Edit]** を選択し、次に **[Resource]** を選択します。ドロップダウンメニューから **[Extend Resource Hierarchy]** を選択します。これで [Extend Resource Hierarchy] ウィザードが起動されます。拡張操作に慣れていない場合は、 **[Next]** をクリックします。LifeKeeper の [Extend Resource Hierarchy] のデフォルト値が分かっていて、入力と確認を省略する場合は **[Accept Defaults]** をクリックします。

2. [Pre-Extend Wizard] で以下の情報を入力します。

**注記:** 最初の 2 つのフィールドは [Edit] メニューの [Extend] から操作を開始した場合にだけ表示されます。階層を拡張する手順の間に [Cancel] をクリックすると、どの時点であってもそのサーバへの拡張処理がキャンセルされるので注意してください。ただし、すでにリソースを別のサーバに拡張している場合は、明示的に拡張解除するまで、そのインスタンスの拡張は有効です。

| フィールド | ヒント |
|---|---|
| スイッチバックタイプ | スイッチバックタイプを選択します。ここでは、バックアップサーバへのフェイルオーバの後、EC2 インスタンスが In Service に戻ったときに、このサーバに EC2 インスタンスをスイッチバックする方法を指定します。[intelligent] または [automatic] を選択できます。<br>• intelligent の場合、インスタンスをプライマリ / オリジナルサーバにスイッチバックするときに管理者の介入が必要になります。<br>• automatic の場合、プライマリサーバがオンラインに戻り、LifeKeeper コミュニケーションパスを再確立した直後に自動的にスイッチバックが行われます。<br>スイッチバックタイプは、必要な場合 [Resource Properties] ダイアログボックスの [General] タブで後から変更できます。 |
| Template Priority | テンプレートの優先順位を選択するか、入力してください。これはサーバで現在 In Service の EC2 階層の優先順位です。1 ～ 999 の範囲で、まだ優先順位として使用されていない値が有効で、小さい数値ほど優先順位が高くなります (数値 1 が最高の優先順 |

| | |
|---|---|
| | 位)。拡張処理時に、別のシステムですでに使用中の優先順位をこの階層に対して指定することはできません。デフォルト値を使用することを推奨します。<br><br>**注記:** このフィールドは、階層を最初に拡張するときにだけ表示されます。 |
| Target Priority | ターゲットの優先順位を選択するか、入力してください。これは、他のサーバにある同等の階層に対する、新しく拡張する EC2 階層の優先順位です。1 ～ 999 の範囲で、まだ優先順位として使用されていない値が有効で、リソースのカスケーディングフェイルオーバシーケンスにおけるサーバの優先順位を示します。数値が小さいほど優先順位が高くなります (数値 1 が最高の優先順位)。<br><br>**注記:** LifeKeeper のデフォルトでは、階層が作成されたサーバに「1」が割り当てられます。優先順位は連続している必要はありませんが、特定のリソースについて 2 つのサーバに同じ優先順位を割り当てることはできません。 |

3. 情報ボックスが表示され、LifeKeeper が環境のチェックを正常に完了し、この EC2 リソースを拡張するためのすべての要件が満たされていることが示されます。満たされていない要件がある場合は、 **[Next]** ボタンは選択できなくなり、 **[Back]** ボタンが有効になります。 **[Back]** をクリックした場合、情報ボックスに表示されるエラーメッセージの内容に従って、リソースの拡張を変更できます。ここで **[Cancel]** をクリックした場合は、EC2 リソース階層を他のサーバに手動で拡張して、LifeKeeper の保護下に置く必要があります。 **[Next]** をクリックすると、[Extend Resource Hierarchy] 設定作業に入ります。

4. リソース階層構造の拡張を行うために、以下の情報の入力が要求されます。

| フィールド | ヒント |
|---|---|
| EC2 Resource Tag | EC2 リソースタグを選択するか、入力してください。これは、ターゲットサーバに拡張される EC2 リソースが使用するリソースタグ名です。<br>**注記:** このフィールドは編集できません。 |

5. 拡張が実行中であることを示す情報ボックスが表示されます。同じ EC2 リソースインスタンスをクラスタ内の別のサーバに拡張する場合は、 **[Next Server]** をクリックしてください。その場合は、リソース階層を拡張する操作を繰り返します。 **[Finish]** をクリックすると、LifeKeeper は EC2 リソースの拡張が正常に完了したことを確認します。

6. **[Done]** をクリックして、 **[Extend Resources Hierarchy]** メニューの選択を終了します。

**注記:** 必ずすべてのサーバで新しいインスタンスの機能をテストしてください。

# 6.3.4.6. EC2™ ローカルリカバリと設定に関する考慮事項

## ローカルリカバリシナリオ (バックエンドクラスター):

保護されているルートテーブルの障害が Recovery Kit for EC2 によって検出されると、結果として生じる障害によって EC2 ローカルリカバリスクリプトが起動されます。ローカルリカバリでは、すべてのルートテーブル内から指定された IP リソースエントリを収集し、そのエントリのターゲットをアクティブなサーバ上にある ENI に変更します。また、ネットワークインターフェースの送信元/送信先チェックを無効にします。ローカルでリカバリの試みが失敗すると、LifeKeeper は EC2 リソース、および依存関係を持つすべてのリソースをスタンバイサーバにフェイルオーバします。このシナリオの設定については、運用の原則 セクションを参照してください。

**注記:** 対応する EC2 リソースが作成されると、Recovery Kit がルートテーブルの設定を保護するので、ルートテーブルを手動で変更しないでください。

以下の例で、典型的なローカルリカバリのシナリオを示します。Recovery Kit が、ルートテーブル内にある IP ルートのターゲット設定が正しくないことを検出すると、ローカルリカバリにより、アクティブなサーバ上の ENI にターゲットが置換されます。このプロセスでは、ルートテーブル B の 10.1.0.20/32 のエントリは何も変更されません。

| | |
|---|---|
| IP リソース | 10.1.0.10 |
| アクティブなノード上の ENI | eni-01234567 |

### ルートテーブル A – 前の状態

| 接続先 | ターゲット |
|---|---|
| 10.1.0.10/32 | eni-89abcdef |
| 10.0.0.0/16 | ローカル |

### ルートテーブル A – 後の状態

| 接続先 | ターゲット |
|---|---|
| 10.1.0.10/32 | eni-01234567 |
| 10.0.0.0/16 | ローカル |

### ルートテーブル B – 前の状態

| 接続先 | ターゲット |
|---|---|
| 10.1.0.10/32 | eni-89abcdef |
| 10.1.0.20/32 | eni-89abcdef |
| 10.0.0.0/16 | ローカル |

### ルートテーブル B – 後の状態

| 接続先 | ターゲット |
|---|---|
| 10.1.0.10/32 | eni-01234567 |
| 10.1.0.20/32 | eni-89abcdef |
| 10.0.0.0/16 | ローカル |

## Elastic IP シナリオ (フロントエンドクラスタ):

保護されている Elastic IP の障害が Recovery Kit for EC2 によって検出されると、結果として生じる障害によって EC2 ローカルリカバリスクリプトが起動されます。ローカルリカバリにより、Elastic IP がアクティブなノード上の ENI に割り当てられます。ローカルでリカバリの試みが失敗すると、LifeKeeper は EC2 リソース、および依存関係を持つすべてのリソースをスタンバイサーバにフェイルオーバします。このシナリオの設定については、運用の原則 セクションを参照してください。

# 6.3.4.7. EC2™ リソース監視と設定に関する考慮事項

## ルートテーブルシナリオ (バックエンドクラスタ)

Recovery Kit は AWS CLI を使用して、保護されている IP リソースに VPC 内のクライアントから接続できるようにルートテーブルの設定を監視します。Recovery Kit は、VPC 内のすべてのルートテーブルに対して、IP リソース宛のターゲットがアクティブサーバの ENI に正しく設定されていることを確認します。また、ネットワークインターフェースの送信元/送信先チェックが無効になっているかも確認します。正しいことが確認されない場合は、Recovery Kit が EC2 のローカルリカバリプロセスを実行します。

## Elastic IP シナリオ (フロントエンドクラスタ):

Recovery Kit は AWS CLI を使用して、Elastic IP と アクティブなサーバ上にある ENI との関連付けを監視します。Recovery Kit は、アクティブサーバに接続されている ENI に Elastic IP が正しく関連付けられていることを確認します。正しいことが確認されない場合は、Recovery Kit が EC2 のローカルリカバリプロセスを実行します。

**注記:** どちらのシナリオにおいても、AWS CLI でタイムアウトが発生した場合、フェイルオーバは実行されず、リソースは ISP 状態のままになります。タイムアウト関連のメッセージのみが LifeKeeper ログに記録されます。Recovery Kit はチェック間隔の経過後に監視を再度実行します。タイムアウト値を設定する方法の詳細については、EC2パラメータ一覧 を参照してください。

# 6.3.4.8. EC2™ リソース階層の拡張解除

階層全体を拡張解除するには、以下の手順を実行してください。

1. **LifeKeeper GUI メニュー** から **[Edit]** を選択し、次に **[Resouce]** を選択してください。ドロップダウンメニューから **[Unextend Resource Hierarchy]** を選択してください。

2. EC2 リソースから、拡張解除の対象となるターゲットサーバを選択してください。EC2 リソースが現在 In Service になっているサーバは選択できません。 **[Next]** をクリックしてください。

   **注記:** 右側のペインから個々のリソースインスタンスを右クリックして **[Unextend]** 作業を選択した場合、このダイアログボックスは表示されません。

3. 拡張解除する EC2 階層を選択してください。 **[Next]** をクリックしてください。

   **注記:** 左側のペインにあるグローバルリソースを右クリックするか、右側のペインにある個々のリソースインスタンスを右クリックして **[Unextend]** 作業を選択した場合、このダイアログボックスは表示されません。

4. 拡張解除の対象として選択したターゲットサーバと EC2 リソース階層を確認する情報ボックスが表示されます。 **[Unextend]** をクリックしてください。

5. EC2 リソースが正常に拡張解除されたことを示す情報ボックスが表示されます。

6. **[Done]** をクリックして終了してください。

# 6.3.4.9. EC2™ ユーザーシステムのセットアップ

## ルートテーブルシナリオ (バックエンドクラスタ)

Recovery Kit for EC2 のルートテーブル保護オプションを使用すると、VPC 内のルートを自動更新することができます。フェイルオーバ中、Recovery Kit はターゲットサーバの仮想 IP アドレスが表す新しい Elastic Network Interface (ENI) の場所を反映するように、ルートテーブルを更新します。LifeKeeper が VPC 内のルートテーブルの保護、監視、および更新を実行できるようにするには、以下の設定手順を実行する必要があります。アクティブ / アクティブ設定の場合も同様です。

- LifeKeeper for Linux IP Recovery Kit で保護する仮想 IP アドレスは、VPC 内の割り当て済み CIDR の範囲外にする必要があります。
- 仮想 IP アドレスを LifeKeeper で保護してから、Recovery Kit for EC2 リソースを作成する必要があります。
- ENI のソース / ターゲットのチェックを無効にする必要があります。これは、インスタンスが仮想 IP アドレスのネットワークパケットを受信するために必要です。
- LifeKeeper IP リソースのブロードキャスト PING チェックを無効にする必要があります。LifeKeeper は、ローカルサブネット上の IP アドレスに対してブロードキャスト PING テストを実行することで、IP リソースを監視します。複数のアベイラビリティーゾーンが存在する環境では、複数のアベイラビリティーゾーンに異なるサブネットが存在するので、この機能は使用できません。この機能を無効にするには、 */etc/default/LifeKeeper* の設定ファイルの NOBCASTPING エントリを以下のように設定する必要があります。

  NOBCASTPING=1

- ルートテーブルには、仮想 IP アドレスおよびアクティブなサーバの ENI のルートエントリが必要です。>

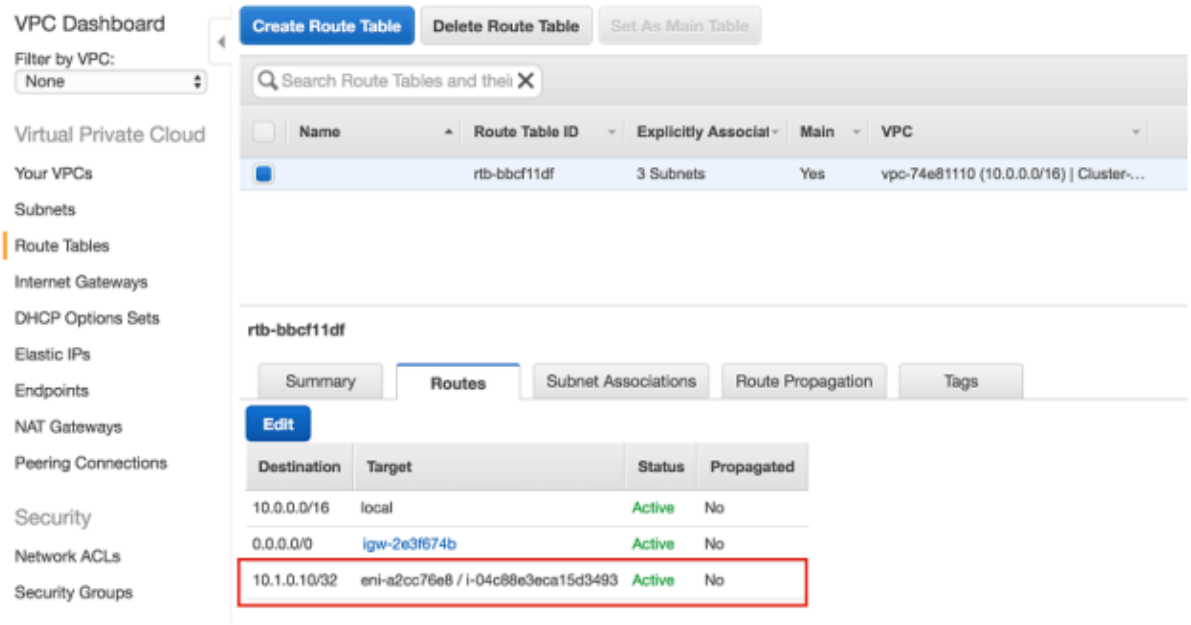
  **注記:**

  対応する EC2 リソースが作成されると、Recovery Kit がルートテーブルの設定を保護するので、ルートテーブルを手動で変更しないでください。

  **例**

  送信先: VIP 10.1.0.10/32

  ターゲット: eni-a2cc76e8




## Elastic IP シナリオ (フロントエンドクラスタ)

Recovery Kit for EC2 の Elastic IP (EIP) 保護オプションを使用すると、特定の ENI (アクティブサーバまたはバックアップサーバ上の EC2 が使用する ENI) と EIP の関連付けを再実行できます。

アクティブサーバまたはバックアップサーバ上の ENI と EIP との関連付けについて、LifeKeeper が保護、監視、および更新を実行できるようにするには、以下の設定手順を実行する必要があります。

- 1 つの ENI は、1 つの Elastic IP とのみ関連付けることができます。その他の EIP (EC2 リソースにより使用される EIP を除く、すべての EIP) は、その ENI と関連付けることができません。他の EIP を関連付けると、Recovery Kit はその ENI にすでに関連付けていた他の EIP の関連付けを解除します。

注記:

- AWS の Elastic Block Store (EBS) は 1 つの EC2 インスタンスにのみ接続可能なため、EBS を使用する HA クラスタ構成を作成する際は、DataKeeper for Linux を使用することを推奨します。
- /etc/default/LifeKeeper の RESRVRECTIMEOUT の値を、デフォルト値の 150 から 300 に増加させることを推奨します。RESRVRECTIMEOUT は、別のプロセスがすでにリソースをリザーブしているときに、リソースを「リカバリ」用にリザーブするまで、LifeKeeper プロセスが待機する間スリープ状態になっている秒数です。

# 6.3.5. Recovery Kit for EC2™ トラブルシューティング

メッセージカタログ は、LifeKeeper for Linux の使用時に検出されるすべてのメッセージのリストを示すとともに、適宜、エラーの原因とエラー状態を解決するために必要な処置の説明を提供します。受け取ったエラーコードをこのリストから探すことができます。また、Recovery Kit for EC2 の使用時に検出されるすべてのメッセージがリストされた Recovery Kit for EC2 メッセージカタログ に直接アクセスすることもできます。

# 6.4. LB Health Check Kit 管理ガイド

## 概要

LB Health Check Kit は、Microsoft Azure（以下Azure）やGoogle Cloud™ 環境などのクラウド環境でロードバランサーの負荷分散対象インスタンスへのヘルスチェックプローブを待ち受けて応答する仕組みを提供します。

本書では LB Health Check Kit の動作詳細を解説します。LifeKeeper の一般的な用語、使用方法等については、 SIOS Protection Suite for Linux / LifeKeeper for Linux テクニカルドキュメンテーション を参照してください。

また、本書の内容はあらゆる環境において正常に動作することを保証するものでありません。

## 前提条件

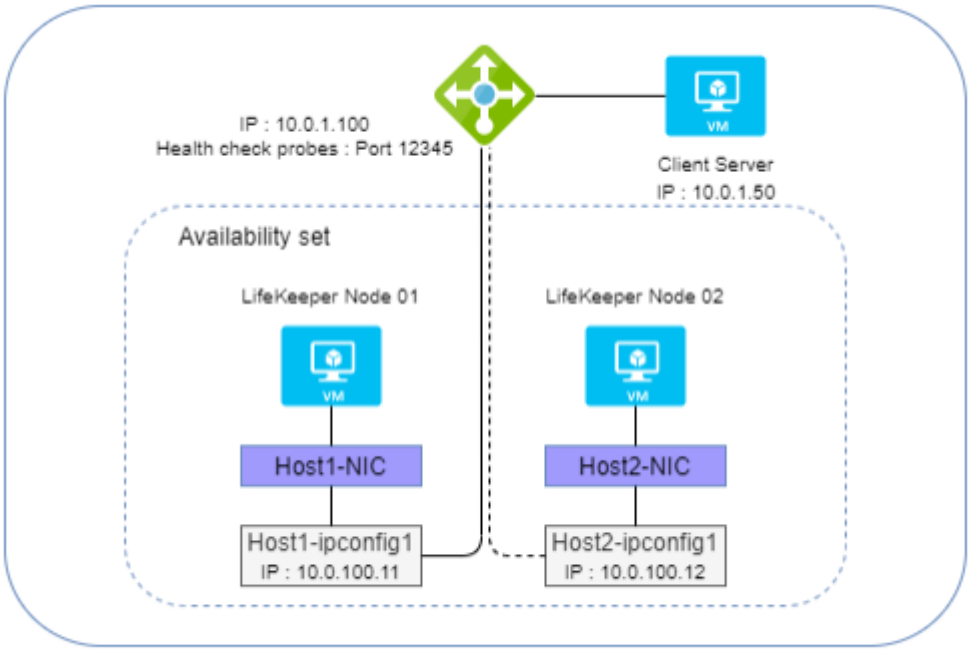
- LB Health Check Kit は Single Server Protectionでは利用できません。
- プローブのプロトコルはTCPのみをサポートします。
- サポートされるOSや構成につきましては、オンラインドキュメントの リリースノート および SIOS Protection Suite for Linux / LifeKeeper for Linux サポートマトリックス をご確認ください。

Google Cloud、BigQuery および Google Compute Engine は Google LLC の商標です。

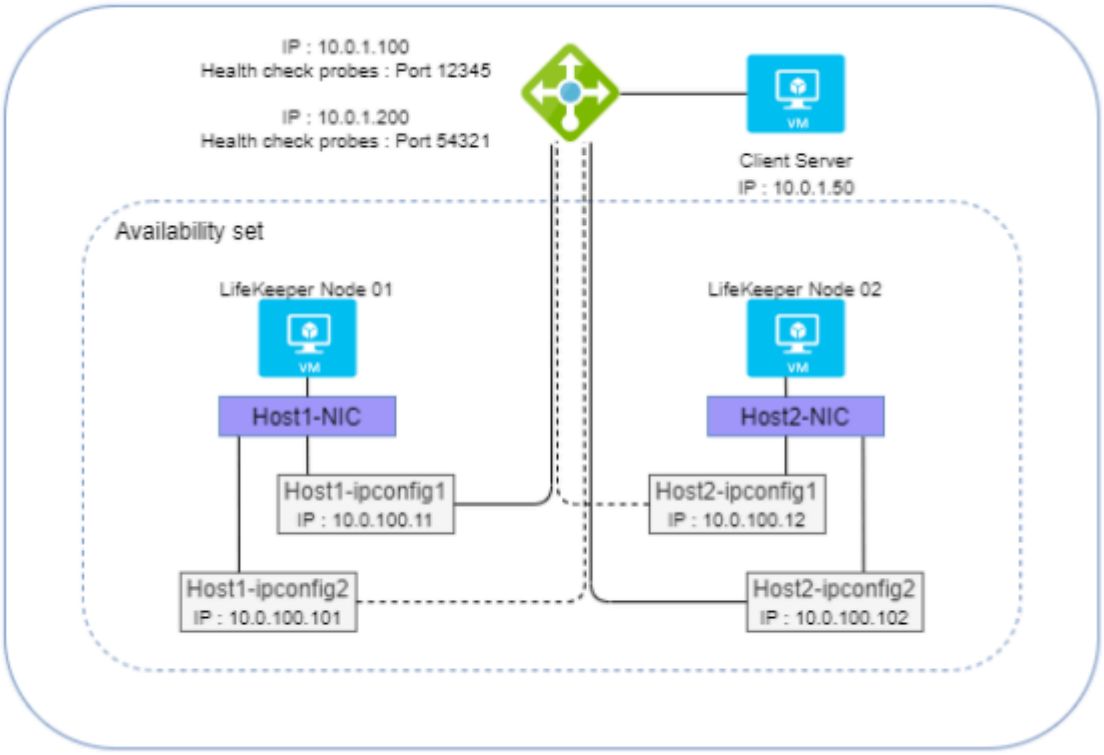
# 6.4.1. 環境構成例

## Azure を使用する場合の構成例

### Active / Standby 構成

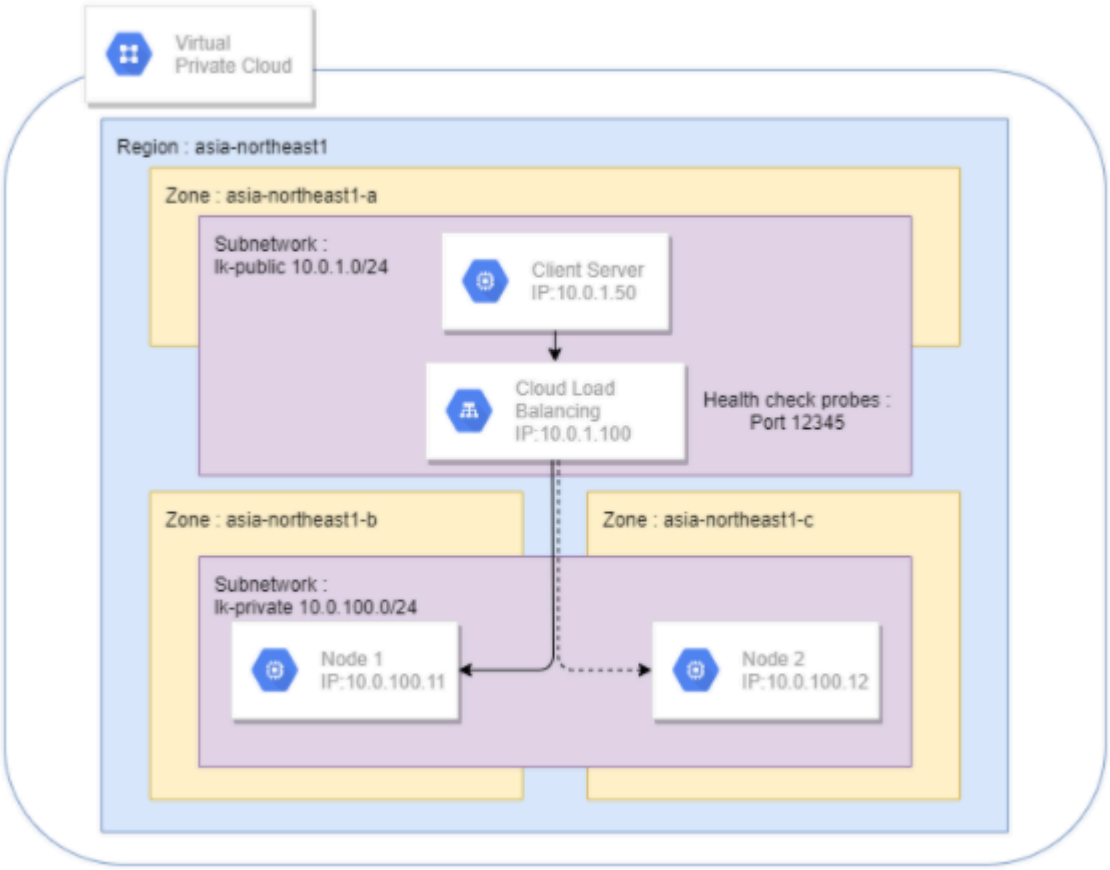



### Active / Active 構成

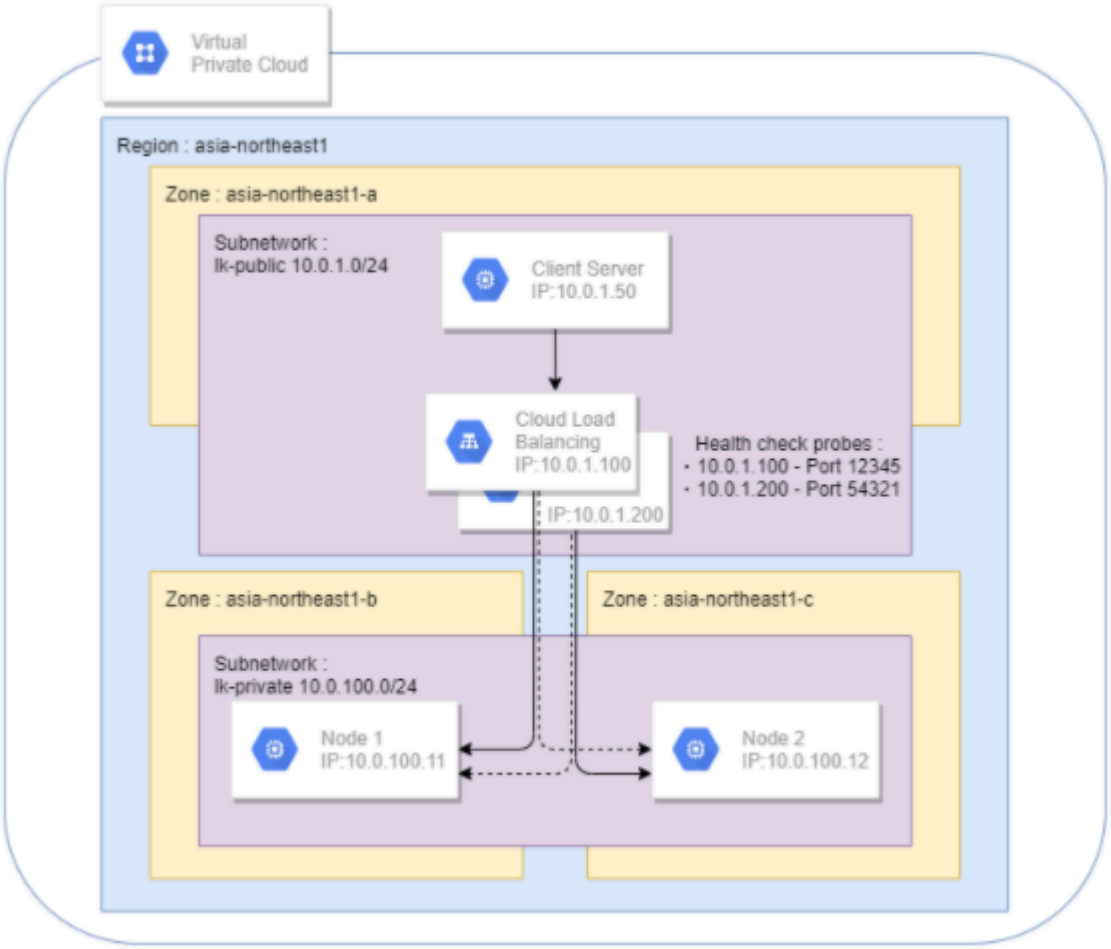


# Google Cloud を使用する場合の構成例

## Active / Standby 構成



## Active / Active 構成



# 6.4.2. Generic ARK for Load Balancer probe reply から LB Health Check Kit へのアップグレード

✻ SIOS Protection Suite for Linux v9.6.2 より、SIOSは新しい LB Health Check Kitをリリースしました。v9.6.2 以降をご利用の場合は、新しい（ビルトインの）LB Health Check Kitを使用する必要があります。既存の Generic ARK for Load Balancer probe reply は v9.6.2 ではサポートされていません。以下のアップグレード手順に従ってください。

Generic ARK for Load Balancer probe reply からのアップグレード手順を以下のサンプル構成例で説明します。

| | |
|---|---|
| 既存の **Gen LB** リソース | リソース名: GenLB<br>port : 12345<br>message : 未使用<br>親リソース名: gen1<br>子リソース名: /disk1, ip-192.168.181.203 |
| 移行後の **LB Health Check** リソース | リソース名: lbhc-12345<br>port : 12345<br>message : 未使用<br>親リソース名: gen1<br>子リソース名: /disk1, ip-192.168.181.203 |
| 拡張サーバー | node2 |

## SIOS Protection Suite/LifeKeeper のアップグレード

1. クラスター内のすべてのノードで、SIOS Protection Suiteを最新バージョンにアップグレードします ( SPS のアップグレード を参照してください).

2. リソース階層を以前の設定に復元する必要がある場合に備えて、LifeKeeperの設定をバックアップします。PATH に /opt/LifeKeeper/bin をいれておきます。

```
lkbackup -c --cluster
```

## 依存関係の削除

Gen LB 削除前に依存関係を削除します。

```
lkcli dependency delete --parent gen1 --child GenLB
```

```
lkcli dependency delete --parent GenLB --child /disk1
```

```
lkcli dependency delete --parent GenLB --child ip-192.168.181.203
```

## Gen LB の削除

Gen LB の Active ノードで GenLB を Out of Service してから、削除します。

```
lkcli resource remove --tag GenLB
```

```
lkcli resource delete --tag GenLB
```

## LB Health Checkリソースを作成

LB Health Check リソースを作成します。

```
lkcli resource create lbhc --tag lbhc-12345 --port 12345
```

```
lkcli resource extend lbhc --tag lbhc-12345 --dest node2
```

## 依存関係の再作成

依存関係を再作成します。

```
lkcli dependency create --parent gen1 --child lbhc-12345
```

```
lkcli dependency create --parent lbhc-12345 --child /disk1
```

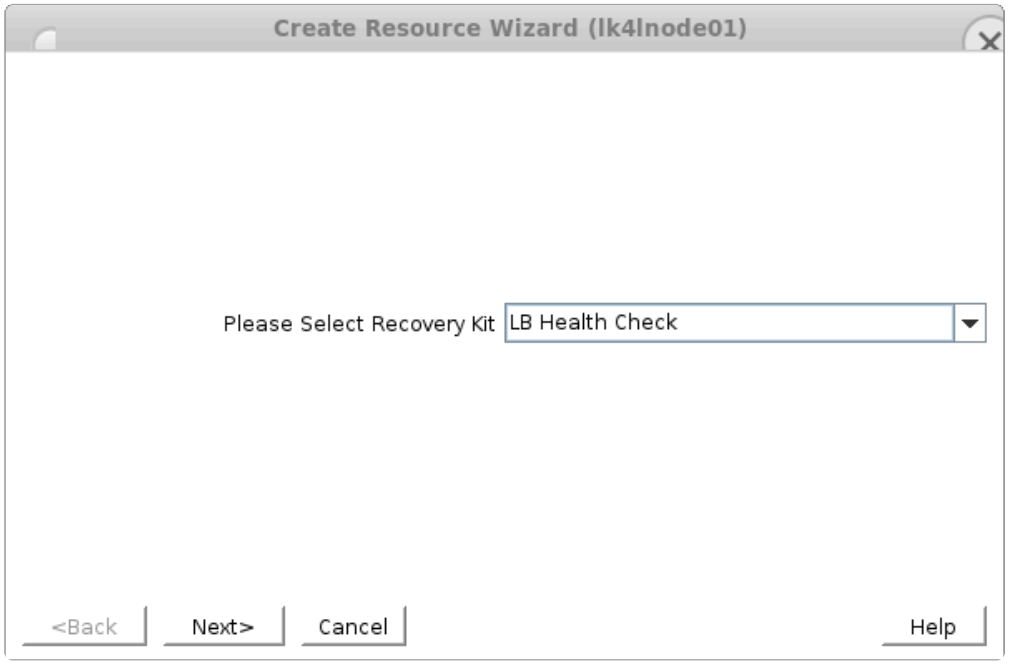
```
lkcli dependency create --parent lbhc-12345 --child ip-192.168.181.203
```

# 6.4.3. リソースの作成/拡張/変更

## リソース階層の作成

LB Health Check リソースの作成は、以下の手順で行います。

1. リソース階層の作成 を参考に **[Create Resource Wizard]** を起動します。
2. Recovery Kit として **[LB Health Check]** を選択します。

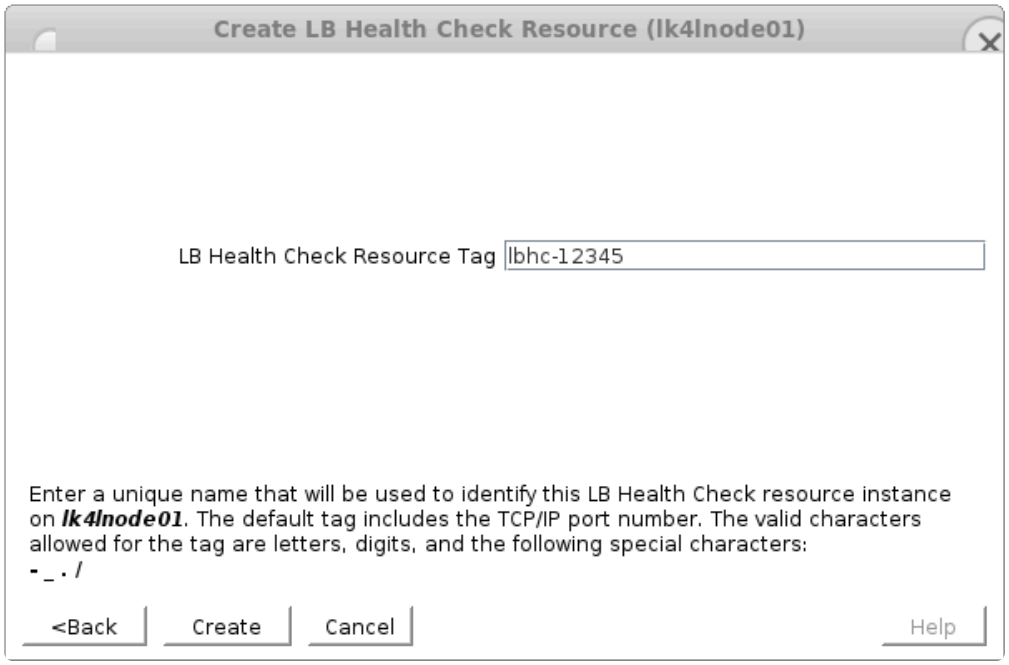



3. 以下のパラメーターを入力します。

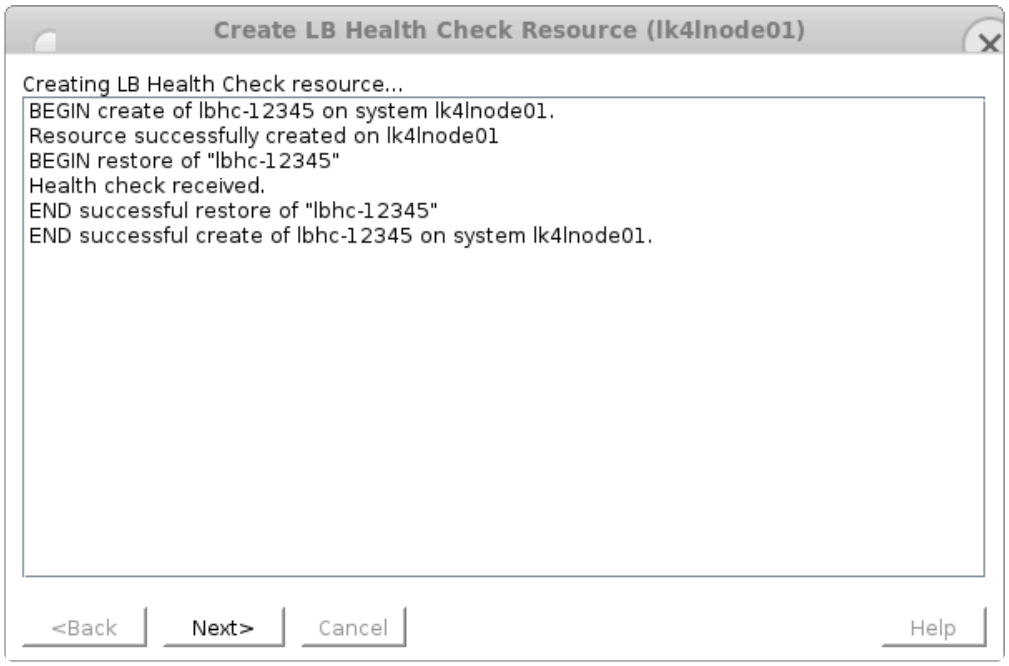
階層作成の途中で **[Cancel]** ボタンをクリックすると、作成処理全体が取り消されます。

| フィールド | ヒント |
|---|---|
| Switchback Type | フェイルオーバー後にこのノードが復帰したときに、このノードに LB Health Check リソースを戻す方法を指定します。[intelligent] または [automatic] を選択できます。 |
| Server | リソースを作成するノードを選択します。クラスター内のすべてのノードがドロップダウンリストに表示されます。 |
| Reply daemon Port | ロードバランサーからの接続を待ち受けるポート番号を指定します。 |
| Reply daemon message | デーモンからの応答文字列を指定します。この設定は任意です。 |
| LB Health Check | 作成している LB Health Check リソースに固有の タグ名を入力します。このフィールドには、デフォルトのタグ名である lbhc-<port> が自動的に表示されます。<port> は指定した |

| | |
|---|---|
| Resource Tag | Reply daemon port です。 |




4. 全てのパラメーターを入力すると、LB Health Check リソースをISP(サービス稼動状態)にするために、指定されたPort番号でヘルスチェックプローブを待ち受けるデーモンを起動します。LifeKeeperが問題を検知した場合は、情報ボックス、あるいはログファイルにエラーが表示されます。




リソースの作成に成功すると、 **[Pre-Extend configuration task]** が起動します。

リソース階層を別のノードに拡張する方法の詳細については、リソース階層の拡張 を参照してください。

# リソース階層の削除

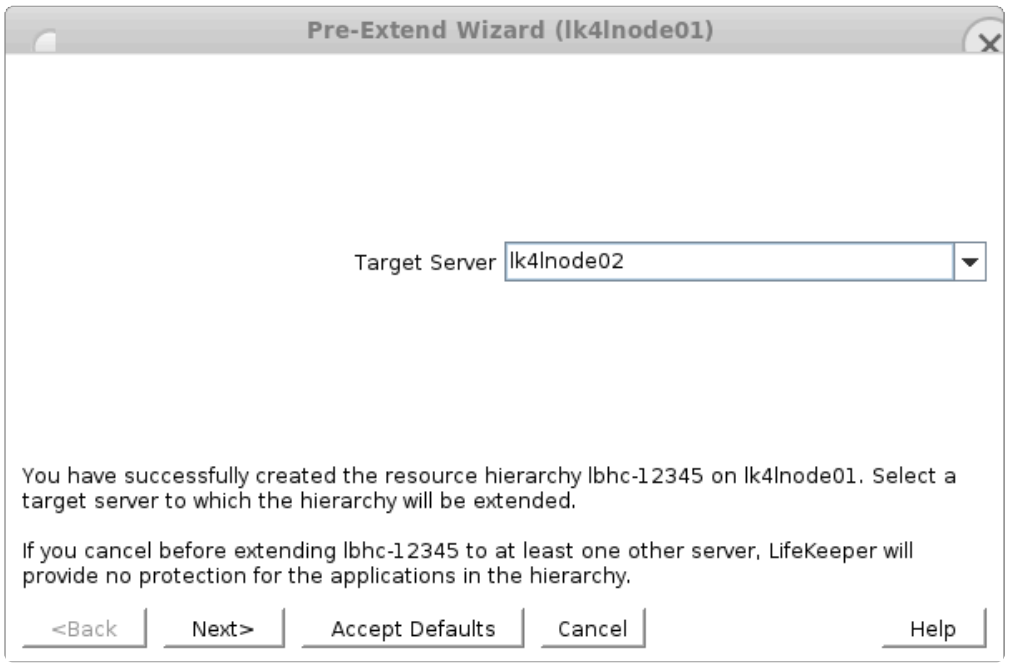
LB Health Check リソースの削除は、すべてのサーバからの階層の削除 を参照してください。

> ! LB Health Check リソースを削除する際、リソースの Out of Service、すなわち lbhcd デーモンの停止が行われます。

# リソース階層の拡張

LB Health Check リソース階層の拡張は、以下の手順で行います。

1. リソース階層の拡張 を参考に **[Extend Resource Hierarchy]** ウィザードを起動します。




2. **[Pre-Extend Wizard]** で以下の情報を入力します。

| フィールド | ヒント |
|---|---|
| Switchback Type | フェイルオーバー後に拡張先ノードが復帰したときに、そのノードに LB Health Check リソースを戻す方法を指定します。[intelligent] または [automatic] を選択できます。 |
| Template Priority | 拡張元ノードのリソース階層の優先順位を入力します。 |
| Target Priority | 拡張先ノードのリソース階層の優先順位を入力します。 |

3. 対象の LB Health Check リソースの拡張要件を満たしていることを確認するために CanExtend が実行されます。
4. 以下のパラメーターを入力し、リソースの拡張を行います。

| フィールド | ヒント |
| --- | --- |
| LB Health Check Resource Tag | LB Health Check リソースタグを入力します。これは、スタンバイノードに拡張される LB Health Check リソースが使用するリソースタグ名です。デフォルトでは、拡張元ノードでのリソースタグ名が表示されます。 |

5. 同じ LB Health Check リソースをクラスター内の別のノードに拡張する場合は、 **[Next Server]** をクリックし、同様の操作を繰り返します。 **[Finish]** をクリックすると、LifeKeeper は LB Health Check リソースの拡張が正常に完了したことを確認します。

## リソース階層の拡張解除

LB Health Check リソースの拡張解除は、階層の拡張解除 を参照してください。

## リソースプロパティの変更

LB Health Check リソースのポート番号、応答文字列は後から変更できます。変更前に LB Health Check リソースを Out of Service にしてから、 Properties Panel の [Resource Configuration] タブですべてのサーバーに対して変更し、In Service にしてください。lkcli config でも同様な方法で変更できます。

# 6.4.4. 処理概要

## 起動処理(restore)

以下の順序で LB Health Check リソースの起動を試みます。

1. restoreスクリプトは、指定されたリソースIDを使用して、関連するデーモンプロセスがすでに開始されているかどうかを判断します。すでに開始されている場合、restore スクリプトはexit code 0で終了します。

2. restoreスクリプトはSocketオブジェクトを生成し、ロードバランサーからの接続を待ちます。$HC_TIMEOUT で定義された秒数以内に接続が確認できなかった場合、restoreスクリプトはexit code 1で終了します。

3. restore スクリプトは lbhcd デーモンスクリプトを起動します。デーモンスクリプトは$HC_TIMEOUTで定義された秒数以内に接続が確認できなかった場合, $id-hc-ngフラグを作成します。 次の接続試行が検出されるとこのフラグは削除され、タイムアウトはリセットされます。 備考: $id-hc-ng フラグは、quickCheck による障害検知に使用します。

4. restore スクリプトは exit 0で終了します。

## 停止処理(remove)

以下の順序で LB Health Check リソースの停止を試みます。

1. removeスクリプトは、指定されたリソースIDに対応する実行中のLB Health Checkデーモンプロセスの PIDを取得します。

2. removeスクリプトは、ステップ1で見つかったPIDを持つデーモンプロセスがまだ実行中であるかどうかを確認します。すでに停止している場合は、remove スクリプトはexit 0で終了します。

3. デーモンプロセスがまだ動作している場合は、removeスクリプトがそれを終了させます。

4. removeスクリプトはexit 0で終了します。

## 監視処理(quickCheck)

以下の順序で LB Health Check リソースの監視を試みます。

1. quickCheckスクリプトは、ロードバランサーヘルスチェックリソースのリソースステータスを取得します。ステータスがISPでない場合（つまり、リソースがすでに停止している場合）、quickCheckスクリプトはexit code 0を返して終了します。

2. quickCheckスクリプトは、lbhcd デーモンが起動しているかチェックします。起動していない場合は recover イベントを発行して、exit code 1 で終了します。

3. quickCheckスクリプトは、$id-hc-ngフラグの存在をチェックします（$id-hc-ngフラグは、$HC_TIMEOUT秒以内に接続試行を受信しなかった場合に、restoreスクリプトのステップ3で生成されるフラグです）。 このフラグが存在する場合、probefail イベントを発行して exit code 1 で終了します。

4. quickCheckスクリプトは、exit code 0 で終了します。

# 回復処理(recover)

以下の順序で LB Health Check リソースの回復を試みます。

1. recovery スクリプトは、指定されたリソースIDを使用して、関連するデーモンプロセスがすでに開始されているかどうかを判断します。すでに開始されている場合、recovery スクリプトはデーモンプロセスを終了させます。

2. restore スクリプトを呼び出し、restore スクリプトの終了コードを返します。

# 各処理の実行時間

## restore, recover

restore 実行時にロードバランサーから$HC_TIMEOUT秒間接続がない場合、restoreの処理はタイムアウトします。指定したポートのロードバランサーからの接続要求が長時間ない場合、一般に設定エラーまたはネットワークの問題でロードバランサーがサーバーと通信できない可能性があることを示します。

## remove, quickCheck

これらの処理に要する時間に制限はありません。

# 6.5. Logical Volume Manager Recovery Kit 管理ガイド

## Logical Volume Manager Recovery Kit テクニカルドキュメンテーション

LifeKeeper for Linux Logical Volume Manager Recovery Kit は、他の Application Recovery Kit に、論理ボリュームのサポートを提供します。LifeKeeper で保護されたアプリケーションは、ストレージ管理の簡素化、要件変更に応じた動的なボリュームのサイズ変更など、Logical Volume Manager (LVM) が提供するメリットを活用できます。

Logical Volume Manager Recovery Kit は、単体で使用されることがなく、他のリソース階層に依存して使われる点で、他のほとんどの Application Recovery Kit と異なります。そのため、Application Recovery Kit に通常付随する多くの作業 (リソース階層の作成など) は、Logical Volume Manager Recovery Kit に直接該当しません。

Linux OS 内の LVM にはミラー機能があります。 この機能は、Logical Volume Manager Recovery Kit ではサポートされていません。

## ドキュメントの内容

このガイドでは次の事項について説明します。

- ドキュメンテーションとリファレンス
  関連する LifeKeeper for Linux のドキュメントと、その入手場所、LVM 製品に関する多数の役立つドキュメントの参照情報の一覧を記載します。

- 要件
  Logical Volume Manager Recovery Kit を適切に設定、インストール、運用する上で必要なハードウェアとソフトウェアを説明します 。LifeKeeper for Linux ソフトウェアをインストールまたはアンインストールする方法の詳細については、LifeKeeper for Linux インストレーションガイド を参照してください。

- 概要
  Logical Volume Manager Recovery Kit と対応するリソースタイプの概要を説明します。

- リソース階層の作成と管理
  Logical Volume Manager Recovery Kit の管理作業の詳細を説明します。

- トラブルシューティング
  情報メッセージとエラーメッセージ、および推奨される対処方法の一覧を示します。

- 処理概要
  Logical Volume Manager Recovery Kit が提供する起動処理、停止処理、監視処理、および回復処理の処理概要について説明します。

# 6.5.1. ドキュメンテーションとリファレンス

以下に、SIOS Technology Corp が発行している LifeKeeper 関連情報の一覧を示します。

- LifeKeeper for Linux リリースノート
- LifeKeeper for Linux テクニカルドキュメンテーション
- LifeKeeper for Linux インストレーションガイド
- アプリケーションリカバリーキット

LVM に関する情報は以下を参照してください。

- 論理ボリュームマネージャーの管理 Red Hat Enterprise Linux 7
- 論理ボリュームの設定および管理 Red Hat Enterprise Linux 8
- 論理ボリュームの設定および管理 Red Hat Enterprise Linux 9
- 論理ボリューム(LVM) | ストレージ管理ガイド | SUSE Linux Enterprise Server 12 SP5
- 論理ボリューム(LVM) | ストレージ管理ガイド | SUSE Linux Enterprise Server 15 SP4
- LVM HowTo (最新情報ではありません)

# 6.5.2. Logical Volume Manager Recovery Kit の要件

LifeKeeper for Linux Logical Volume Manager Recovery Kit をインストールする前に、LifeKeeper の構成が次の要件を満たしている必要があります。LifeKeeper for Linux ハードウェアおよびソフトウェアの設定に関する具体的な手順については、LifeKeeper for Linux インストレーションガイド を参照してください。

# 6.5.2.1. Logical Volume Manager ハードウェア要件とソフトウェア要件

## ハードウェア要件

- サーバー ：Logical Volume Manager Recovery Kit を使用するには、LifeKeeper for Linux リリースノート および LifeKeeper for Linux インストレーションガイド に記載の要件に従って構成されたコンピューターが 2 台以上必要です。

- データストレージ ：Logical Volume Manager Recovery Kit は、共有ストレージ構成で使用する、または、DataKeeper for Linux によって提供される複製されたストレージ構成で使用することができます。Network Attached Storage (NAS) と合わせて使用することはできません。それ以外に、論理ボリューム上のアプリケーションを保護するリカバリキットの要件を超えるような、ストレージ構成に関する特別な要件はありません。

## ソフトウェア要件

- オペレーティングシステム ：サポートされているディストリビューションとバージョンについては、LifeKeeper for Linux サポートマトリックス を参照してください。

- **Logical Volume Manager (LVM)** ：LVM は、すべての主要 Linux ディストリビューションに含まれています。Logical Volume Manager Recovery Kit をインストールするには、lvm2 パッケージがインストールされている必要があります。サポートされる LVM のバージョンは、Linux ディストリビューションで提供されるバージョンです。

- **LifeKeeper for Linux** ソフトウェア ：各サーバーに同一のバージョンの LifeKeeper Core ソフトウェア、Logical Volume Manager Recovery Kit を含む任意の Application Recovery Kit、およびパッチをインストールする必要があります。LifeKeeper Core ソフトウェアおよび Logical Volume Manager Recovery Kit をインストールする手順については、LifeKeeper for Linux インストレーションガイド を参照してください。

Logical Volume Manager Recovery Kit は、LVM を使用した LifeKeeperで保護するディスクリソースを管理するクラスター内の各サーバー上にインストールする必要があります。

Logical Volume Manager Recovery Kit は、LVM ボリューム上に配置されるアプリケーションのリソース階層を作成および拡張する前にインストールする必要があります。

# 6.5.3. 概要

## LVM の操作

LVM は現在すべての主要 Linux ディストリビューションに含まれる標準的なボリューム管理製品です。LVM は、複数の物理ディスクまたはディスクパーティションをボリュームグループとして知られるエンティティにひとまとめにグループ化することができます。ボリュームグループはさらに分割されまたは仕切られて複数の論理ボリュームになります。論理ボリュームは、標準のブロックデバイスとしてアクセスできます。そのため、ブロックデバイスを直接扱うことのできるファイルシステムや任意のアプリケーションによって使用できます。

LVM は、主にストレージ管理の簡素化のために使用されます。論理ボリュームをストレージ要件の変化に応じて、動的にサイズ変更することができ、物理ディスクやパーティションが sda や sdc1 などと命名されるのとは違って、ボリュームグループと論理ボリュームは管理者が選んだIDによってわかりやすく命名されます。

次の図は、LVM エンティティ間の関係を示しています。ファイルシステムまたはアプリケーションは、論理ボリュームを使用します。論理ボリュームは、ボリュームグループをパーティションで仕切ることによって作成されます。ボリュームグループは、1 つ以上の物理ディスクパーティションまたはディスクの集合です。

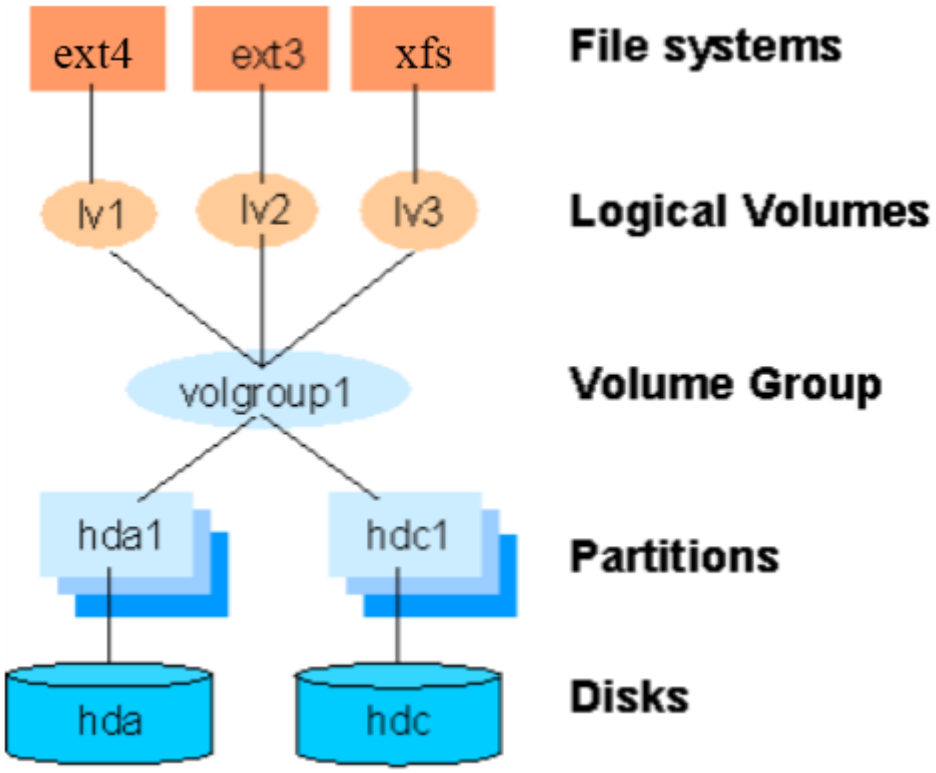



**図 1: LVM のエンティティ間の関係**

# LifeKeeper for Linux Logical Volume Manager Recovery Kit

Logical Volume Manager Recovery Kit は、その他の Application Recovery Kit が論理ボリューム上で正常に連携するために必要なサポートを提供します。このサポートを実現するために、Logical Volume Manager Recovery Kit は 2 つの新しいリソースタイプをインストールします。lvmlv と lvmvgで、それぞれ論理ボリュームとボリュームグループに連携します。 lvmlv および lvmvg リソースは、他の LifeKeeper リソースを動作できるようにするという、内部的な用途のためのみに存在しています。

図 1 に示すように、各ボリュームグループには1つ以上の論理ボリュームからの依存関係があります。逆に言えば、各論理ボリュームは依存関係のあるボリュームグループを1つは持っていなければいけません。lvmlv と lvmvg のリソースタイプを含む一般的なリソース階層は、図 1 に示すような関係になっています。実際のリソース階層の例を見るには、リソース階層の作成と管理 セクションにある図 2 を参照してください。

Logical Volume Manager Recovery Kit では、lvm2 パッケージで提供されるコマンドを使用して、リソース階層のボリュームグループ (lvmvg) と論理ボリューム (lvmlv) リソースを管理します。ボリュームグループと論理ボリュームは、フェイルオーバーやスイッチオーバーの操作中にリソース階層が in service になると設定あるいは起動し、リソース階層が out of service になると設定解除となります。

# 6.5.3.1. Logical Volume Manager Recovery Kit 注意事項と制約事項

以下の注意事項と制約事項は、本バージョンの Logical Volume Manager Recovery Kit に適用されます。

## Raw I/O および全ディスクのサポート

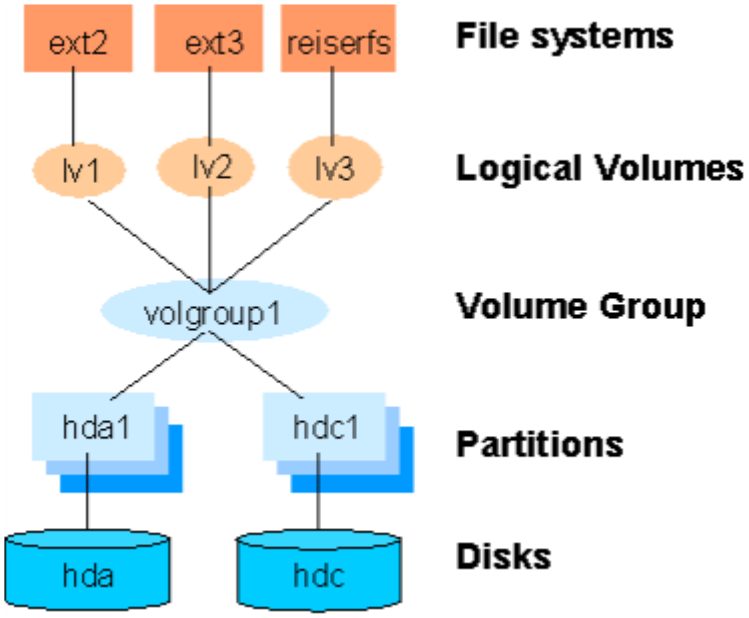
図 1 にはディスクパーティション上の様々なファイルシステムの下にある論理ボリュームとボリュームグループが示されていますが、Logical Volume Manager Recovery Kit を Raw I/O Recovery Kit と組み合わせて使用した場合、論理ボリュームへの raw アクセスをサポート可能であることは重要です。このとき、複数のディスクパーティション (例: `/dev/sdc1`) ではなく、1 つ以上のディスク全体 (例: `/dev/sdc`) で構成されるボリュームグループを管理できることに注目してください。

LVM の使用上のオプションについて詳細を知るには、LVMをDataKeeperで使用する セクションも参照してください。

## LVM ミラーリング機能はサポートされていません

Linux OS 内の LVM にはミラー機能があります。 この機能は、Logical Volume Manager Recovery Kit ではサポートされていません。

# LVM Figure 1


ext2
ext3
reiserfs
lv1
lv2
lv3
volgroup1
hda1
hdc1
hda
hdc
File systems
Logical Volumes
Volume Group
Partitions
Disks

# 6.5.4. LifeKeeper Logical Volume Manager リソース階層の作成と管理

Logical Volume Manager リソース階層は、論理ボリュームリソースの高位に配置されるリソースの作成処理中に自動的に作成されます。拡張、削除と拡張解除についても同様です。

以下の図は、Logical Volume Manager リソースを含むリソース階層全体を示す LifeKeeper GUI のスクリーンショットです。リソースのタグ名はデフォルトの名前ではなく、LifeKeeper ID を明確にするために表示されていることに注意してください。

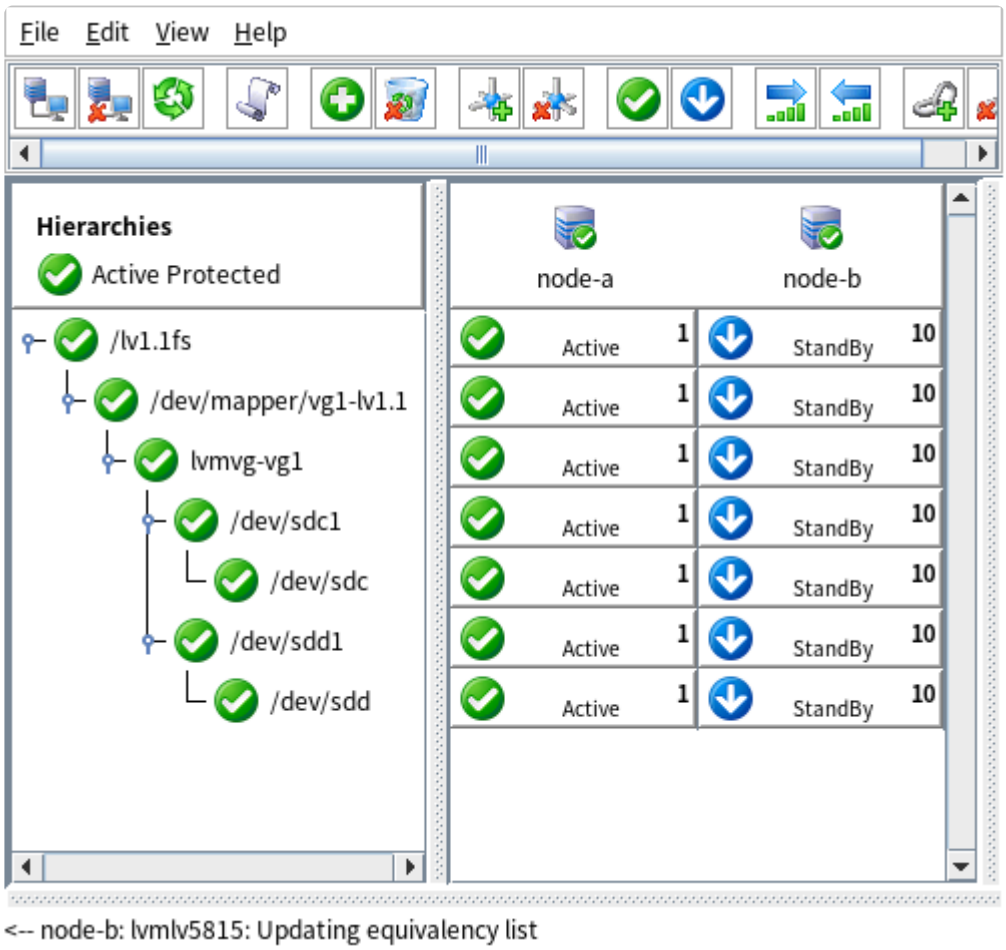



**図 2: Logical Volume Manager リソースを含むリソース階層**

図 2 に示されるリソース階層はファイルシステムリソース階層です。これは、[Edit] > **[Server] > [Create Resource Hierarchy]** メニューから [File System] リカバリキットを選択して作成されます。これは、LVM 論理ボリューム */dev/mapper/vg1-lv1.1* (*/dev/vg1/lv1.1*) にマウントされたファイルシステムリソース `/lv1.1fs`で構成されています。その論理ボリュームは、LifeKeeper ID `lvmvg-vg1` で表される vg1 ボリュームグループの一部です。ボリュームグループ `vg1` は、*/dev/sdc1* および */dev/sdd1* という2つの物理ディスクパーティションで構成されています。リソース階層内の、各ディスクパーティション配下には、さらにディスクデバイス、*/dev/sdc* および */dev/sdd* があります。

# 6.5.4.1. Logical Volume Manager リソース階層作成の手順

ファイルシステムまたは高レベルアプリケーションが LVM 論理ボリュームを使用するリソース階層を作成するには、次の高レベル手順に従う必要があります。

1. LVM ボリュームグループと論理ボリュームの望ましい構成を決定してください。このとき、以下の点を考慮してください。
   - そのボリュームグループに関連するすべてのディスクリソースを、LifeKeeper クラスター内のサーバー間で一緒に移動させる必要があります。
   - そのボリュームグループに関連するすべての論理ボリューム (およびファイルシステムまたはそれを使うアプリケーション) は、LifeKeeper クラスター内のサーバー間で一緒に移動しなければなりません。
   - ( **SLES** と **RHEL7** のみ ) Logical Volume Manager Recovery Kit のインストールの時に lvm2-lvmetad は無効になります。ただし、Logical Volume Manager Recovery Kit のインストール後に lvm2 をインストールした場合は、自分で無効にする設定が必要です。設定方法は OS のマニュアルを参照してください。
2. アプリケーションのプライマリーサーバーになるシステム上で、必要なボリュームグループと論理ボリュームを作成しアクティブにします。作成には lvm2 パッケージに含まれるツールを使いますが、その説明はドキュメンテーションとリファレンス トピックの参照先にある LVM *HowTo* ドキュメントで説明しています。
   共有ストレージを使用してる場合は、ボリュームグループに割り当てられたすべての物理ボリュームが、保護されたアプリケーションを実行しようとしている LifeKeeper クラスター内のマシン間で適切に共有されていることを確認してください。DataKeeper で LVM を使用したい場合は、DataKeeper で LVM を使用する のトピックを参照してください。
3. 各論理ボリューム上でファイルシステムを作成します。そうせずに、Raw I/O を使用したい場合は、Raw デバイスをそれぞれの論理ボリュームデバイスにバインドします。
4. 使用するアプリケーションに関連する Application Recovery Kit 管理ガイドに記載されている設定手順に従い、保護されたアプリケーションをファイルシステム上で設定します。
   適切な Application Recovery Kit 管理ガイドに記載の手順に従って、アプリケーションリソース階層を作成および拡張します。

# 6.5.4.2. DataKeeper で LVM を使用する

LifeKeeper for Linux は現在 LVM 上での DataKeeper の使用と DataKeeper 上での LVM の使用の両方をサポートしています。標準的な DataKeeper の設定では LVM 上での DataKeeper の使用がサポートされているので、Logical Volume Manager Recovery Kit をインストールしないでください。必要な Application Recovery Kit は、DataKeeper だけです。ただし、DataKeeper 上で LVM を使用する場合は、Logical Volume Manager Recovery Kit が必要になります。

SIOS では LVM 上での DataKeeper 使用を推奨していますが、DataKeeper 上で LVM が使用される場合は、2 段階のリソース階層作成プロセスが必要になります。プライマリーサーバー上で LVM ボリュームグループと論理ボリュームを作成する前に、DataKeeper の Hierarchy Type「Data Replication Resource」でリソース階層を作成します。そして、生成された md デバイスに対してボリュームグループと論理ボリュームを作成します。最後に、保護するアプリケーションに関連した Application Recovery Kit の設定手順に従ってリソース階層を作成します。完成したリソース階層は以下に示す図 3 のようになります。

> **注意:** データの整合性を保つため、DataKeeper 上に LVM を構成する場合は、DataKeeper ミラーが 1 つだけ存在するか、複数の **同期** ミラーが存在する必要があります。

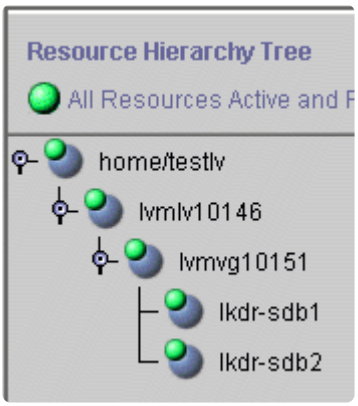



**図 3: DataKeeper 上の LVM を含む階層**

単一のターミナルリーフリソースを各 DataKeeper リソースの子として作成する必要があります。 これにより、階層のサービスが停止するときに単一のリソースが提供され、スイッチオーバー操作中に、別のノードでリソースを起動する前に、階層全体が確実にサービスが停止されます。 詳細については、create_terminal_leaf(1M) のマニュアルページを参照してください。

図 3 の階層のターミナルリーフを作成するには、次のコマンドを実行します。

```
$LKROOT/bin/create_terminal_leaf home/testlv
```

# 6.5.4.3. ボリュームグループの再設定

> **重要** ：DataKeeper上のLVMを保護している場合、当ページで案内している手法によるリソース階層の自動修正はできません。
> 追加したデバイスに対応するData Replicationリソースの作成/削除、mdデバイスのLVMへの追加/削除やLifeKeeperリソースの依存関係の作成/削除はすべて手動で行う必要があります。
> なお、手動での設定が正しく実施できているかどうかは図 5 の画面で追加デバイスが “No added devices.”、削除デバイスが “No deleted devices.” になっていることで確認できます。

LVM を利用する最大のメリットの 1 つは、ストレージ要件の変化に応じて、論理ボリュームを動的にサイズ変更できることです。これは、LVM ボリュームグループ定義への物理パーティションやディスクの追加または削除を伴う場合があるため、Logical Volume Manager Recovery Kit には、そのような変更内容を反映して、既存のリソース階層を修正する仕組みが含まれています。

変更内容を反映してリソース階層を修正する前に、すべてのボリュームグループ、論理ボリュームおよびファイルシステムの再設定を LifeKeeper の外部で行う必要があります。その方法については、ドキュメンテーションとリファレンス セクションで紹介する LVM *HowTo* ドキュメントを参照してください。いずれかの手順において、LifeKeeper で保護されているリソースを **マウント解除** または **設定解除** する必要が生じた場合、必ず リソースを **Out of Service** にしてください。

> **重要**：新しいデバイスは、LifeKeeper が再設定を実行する前に、両方のシステム (共有) から見えていなければいけません。

これらの変更に従ってリソース階層を更新するには、まず、変更した ボリュームグループの **[Resource Properties]** ダイアログを開きます。このダイアログは、アクティブなボリュームグループリソースを右クリックするか、 **[Properties]** を選択するか、 **[Edit] > [Resource] > [Properties]** メニューを選択し、 **[Select Resource]** フィールドで適切なボリュームグループリソースを選択することで開けます。 **[Resource Properties]** ダイアログは以下の図 4 に示すように表示され、下部に **[Resource Configuration]** ボタンがあります。

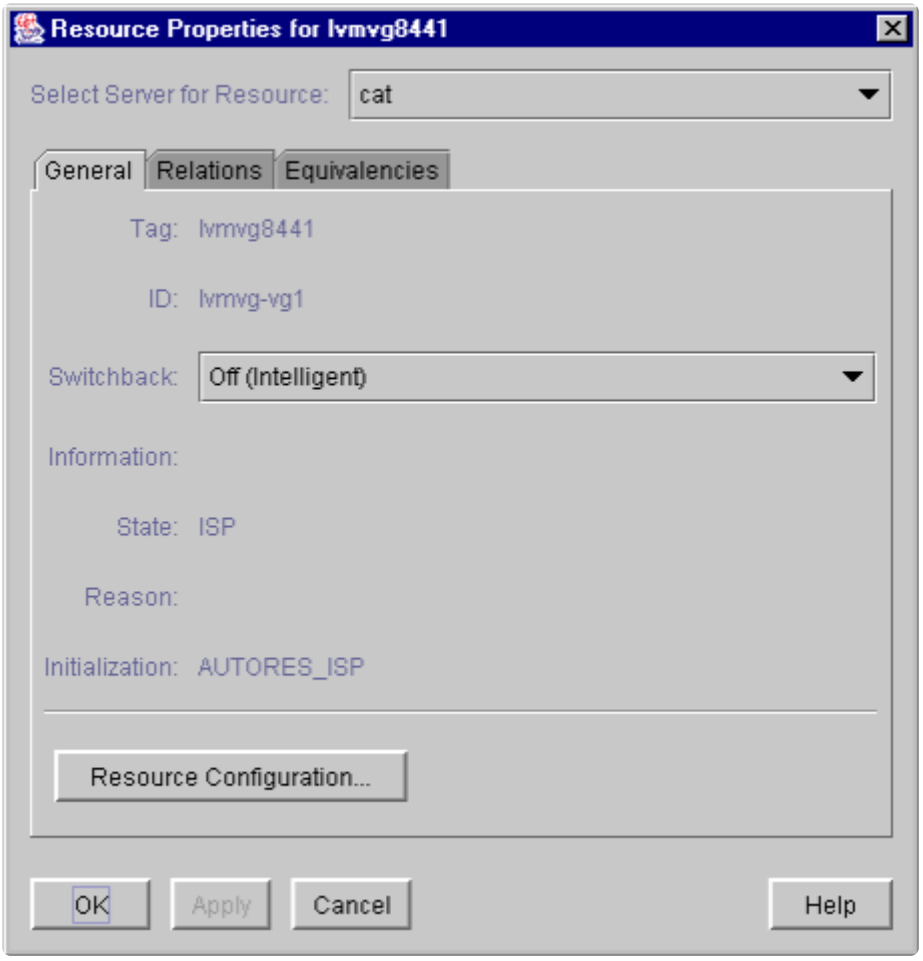



**図 4: LVM ボリュームグループ [Resource Properties] ダイアログ**

**[Resource Configuration]** ボタンをクリックすると、リソース階層を再設定して、ボリュームグループリソースに加えられた変更を反映するための仕組みが開始されます。少し待つと、情報ボックスに LifeKeeper が検出したボリュームグループの変更内容が表示されます。以下の図 5 は、ボリュームグループに1つのディスクパーティションが追加された例を示しています。

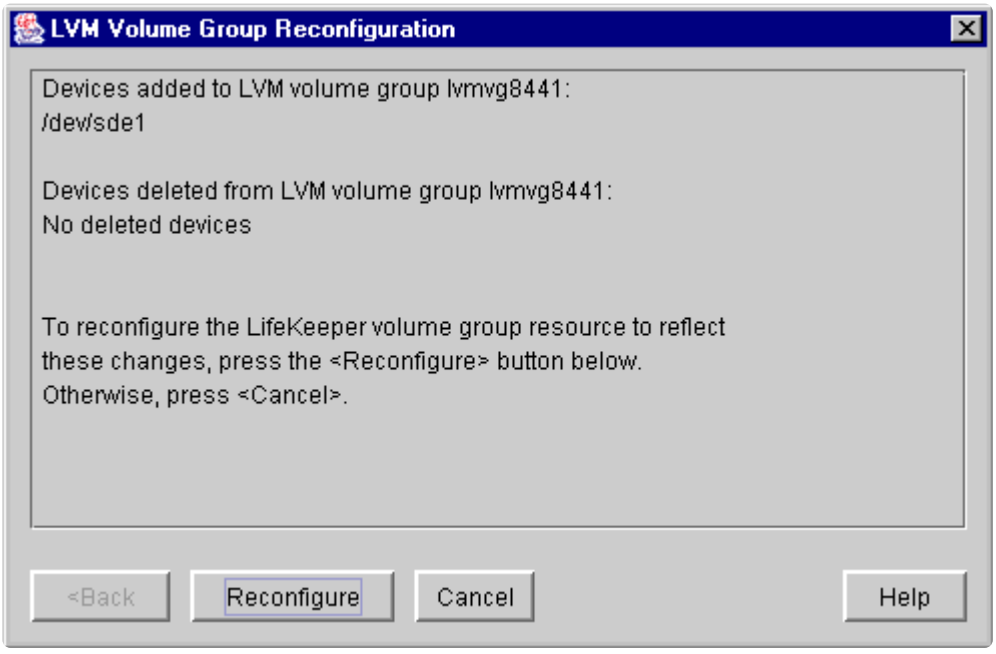



**図 5: デバイスが追加された LVM ボリュームグループの再設定**

情報ボックスの説明にあるように、検出された変更内容を反映してボリュームグループのリソース階層を再設定するには、単に **[Reconfigure]** ボタンをクリックしてください。リソース階層の変更をそれ以上進めたくない場合は、 **[Cancel]** ボタンをクリックしてください。

**[Reconfigure]** ボタンをクリックした後、再設定手順の進捗状況を示す情報ボックスが表示されます (以下の **図6** ) 処理が正常に完了すると、 **[Done]** ボタンが有効になります。 **[Done]** ボタンをクリックすると、情報ボックスが閉じ、 **[Resource Properties]** ダイアログに戻ります。

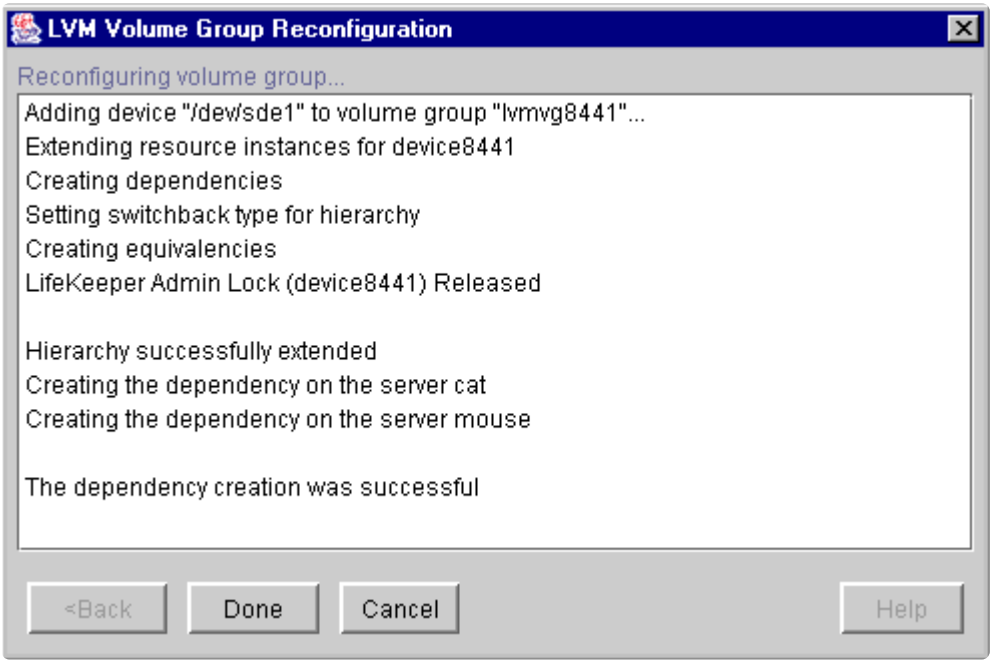



**図 6:** デバイスが追加された**LVM** ボリュームグループの再設定

以下の 2 つの図では、ボリュームグループからデバイスパーティションが削除された際に、再設定の進行中に表示される情報ボックスの例を示します。

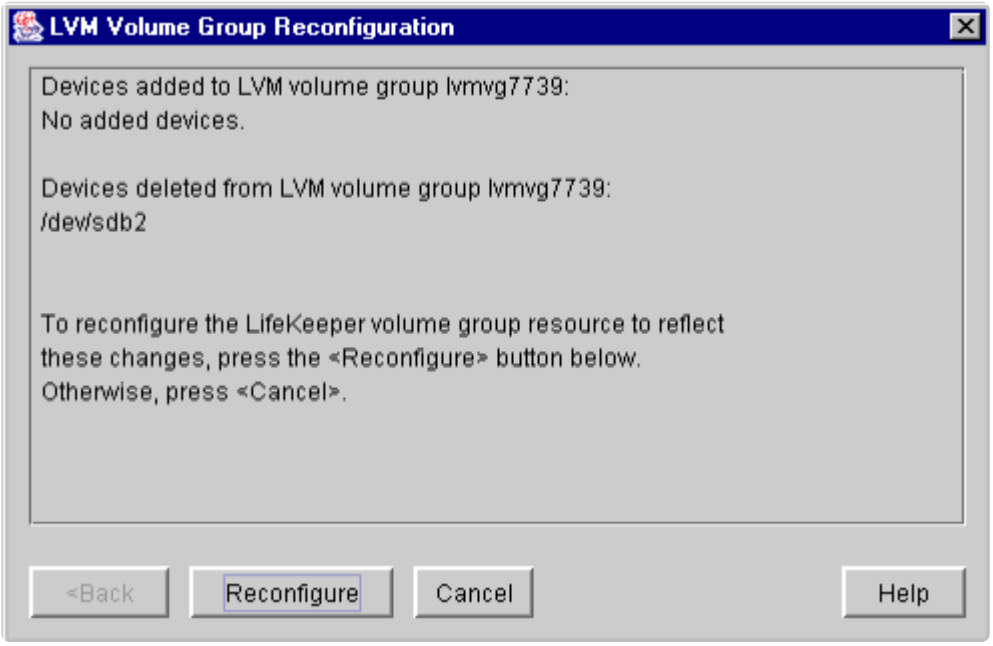



**図 7:** デバイスが削除された**LVM** ボリュームグループの再設定

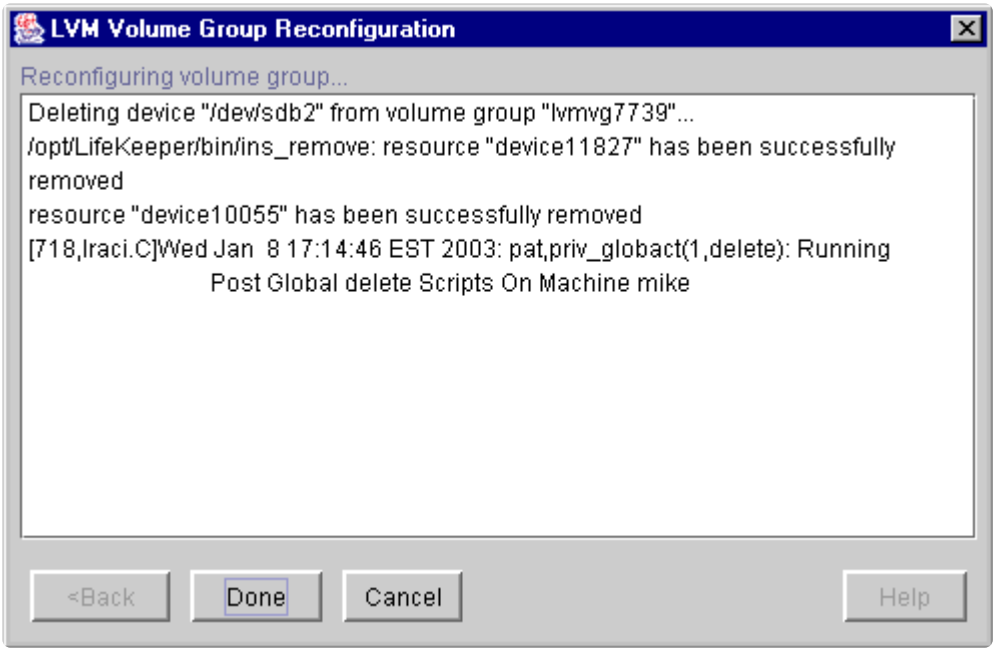



**図 8:** デバイスが削除された**LVM** ボリュームグループの再設定

# 6.5.5. LVM トラブルシューティング

## エラーメッセージ

このセクションでは、Logical Volume Manager Recovery Kit の使用中に表示される可能性のあるメッセージの一覧を示します。必要に応じて、エラーの原因およびエラー状態を解消するために必要な処置についても説明しています。

Logical Volume Manager Recovery Kit は、リソース階層の作成および拡張時に他の LifeKeeper コンポーネントを利用するため、それらの他のコンポーネントによるメッセージが表示されることもあります。そのような場合は、メッセージカタログ を参照してください。メッセージカタログには、操作、管理、GUI など、LifeKeeper for Linux を使用しているときに遭遇する可能性がある、すべてのエラーコードが列挙されています。また、エラーコードの原因に関する追加の説明や、問題解決のために必要な処置についても、必要に応じて記載されています。この一覧から、受信したエラーコードを検索できます。また、該当する LifeKeeper コンポーネントの個別のメッセージカタログに直接アクセスすることもできます。

### Logical Volume Manager Recovery Kit のエラーメッセージ

| エラー番号 | エラーメッセージ |
|---|---|
| 110000 | <LVM resource type> resource type is not installed on <SPS server name>.<br><br>処置: LVM Recovery Kit を特定されたシステムにインストールしてください。 |
| 110001 | This script must be executed on <SPS server name>. |
| 110002 | Failed to create <device name> hierarchy. |
| 110003 | Failed to create dependency <resource tag>-<resource tag> on machine <SPS server name>. |
| 110004 | LifeKeeper internal ID <resource ID> already in use. |
| 110005 | <LVM resource type> constructor requires a valid argument. |
| 110006 | Usage: adddelpv <VG tag> [addlist\|dellist] |
| 110007 | WARNING: Failure in updating list of LifeKeeper-controlled volume groups ( */etc/lkvgs*). |
| 110008 | WARNING: The device hierarchy for <device name>, with tag <device resource tag>, cannot be extended automatically.<br><br>処置: デバイス階層がまだ拡張されない場合、LifeKeeper GUI を使って拡張してください。そしてボリュームグループリソースからデバイス階層まで依存関係を作成してください。 |
| 110009 | Failed to create a dependency between volume group resource <volume group tag> and device resource <device tag>.<br><br>処置: LifeKeeper GUI を使って、依存関係を作ってください。 |
| 110010 | Failed to make the LVM logical volume <Logical Volume Path> active with error code <error |

| | |
|---|---|
| | code>. |
| 110011 | Failed to `vgscan` the LVM volume group <Volume Group Name> with error code <error code>. |
| 110012 | Failed to `vgimport` the LVM volume group <Volume Group Name> with error code <error code>. |
| 110013 | Failed to make the LVM volume group <Volume Group Name> active with error code <error code>. |
| 110014 | <Output from lvmdevices command> |
| 110015 | Failed to make the LVM volume group active even when pvscan was tried. |

# 6.5.6. LVM Recovery Kit 処理概要

LVM を含むファイルシステムリソースの作成を行うと、以下の様なリソース階層が作成されます。

```
[filesys]
└[lvmlv]
  └[lvmvg]
    └[device]
      └[disk]
```

## 起動処理 (restore)

### [lvmlv] リソース

以下のコマンドを実行し、論理ボリュームをアクティブ化します。この結果をもとに起動処理は成功またはエラーで終了します

```
# lvchange -ay <論理ボリュームのパス>
```

### [lvmvg] リソース

まず、lvmdevices コマンドがインストールされた環境に限り、以下のコマンドを実行し、ボリュームグループに紐づく物理ボリュームとなる保護対象のディスクを追加します。コマンドがインストールされていない環境ではこのコマンド実行はスキップします。

```
# lvmdevices --adddev <物理ボリュームのパス>
```

次に以下の処理を最大 2回（固定）実施し、ボリュームグループをアクティブ化します。それぞれのコマンド実行後には vgdisplay コマンドによる状態確認が行われ、最終的な状態としてボリュームグループがアクティブな状態であると判断できた場合は、起動処理は成功で終了します。

1. 以下のコマンドを実行し、ボリュームグループ一覧を取得します。

   ```
   # vgscan
   ```

2. 以下のコマンドを実行し、ボリュームグループのインポートを実施します。

   ```
   # vgimport -f <ボリュームグループ名>
   ```

3. 以下のコマンドを実行し、ボリュームグループのアクティブ化を実施します。

   ```
   # vgchange -ay <ボリュームグループ名>
   ```

項目1. ~ 項目3. の処理においてボリュームグループがアクティブとならない場合は、最初の 1回に限り、以下のコマンドを実行し、ボリュームグループに紐づく物理ボリュームとなる保護対象のディスクを指定し、物理ボリューム一覧を更新します。

```
# pvscan --cache <物理ボリュームのパス>
```

# 停止処理 (remove)

## [lvmlv] リソース

以下のコマンドを最大 10回（固定）実行し、論理ボリュームを非アクティブ化します。この結果をもとに停止処理は成功またはエラーで終了します。

```
# lvchange -an <論理ボリュームのパス>
```

## [lvmvg] リソース

以下の処理を実施し、ボリュームグループを非アクティブ化します。それぞれのコマンド実行後には vgdisplay コマンドによる状態確認が行われ、最終的な状態としてボリュームグループがエクスポートされた状態であると判断できた場合は、停止処理は成功で終了します。

1. 以下のコマンドを最大 10回（固定）実行し、ボリュームグループの非アクティブ化を実施します。

   ```
   # vgchange -an <ボリュームグループ名>
   ```

2. 以下のコマンドを実行し、ボリュームグループのエクスポートを実施します。

   ```
   # vgexport <ボリュームグループ名>
   ```

最後に、lvmdevices コマンドがインストールされた環境に限り、以下のコマンドを実行し、ボリュームグループに紐づく物理ボリュームとなる保護対象のディスクを削除します。コマンドがインストールされていない環境ではこのコマンド実行はスキップします。

```
# lvmdevices --deldev <物理ボリュームのパス>
```

# 監視処理 (quickCheck)

## [lvmlv] リソース

以下のコマンドを実行し、論理ボリュームがアクティブであることを確認します。この結果をもとに監視処理は成功またはエラーで終了します。

```
# dmsetup info <論理ボリュームのパス>
```

### [lvmvg] リソース

このリソースに監視処理はありません。

## 回復処理 (inactive)

### [lvmlv] リソース

1. 監視処理と同一の内容を実施します。
   論理ボリュームがアクティブであれば回復処理は成功で終了します。
   論理ボリュームが非アクティブであれば次の項目2. の処理へ移行します。

2. 停止処理と同一の内容を実施し、次の項目3. の処理へ移行します。
   本項目で実施される処理は、結果判定されません。

3. 起動処理と同一の内容を実施します。
   論理ボリュームのアクティブ化の成功または失敗により、回復処理は成功またはエラーで終了します。

### [lvmvg] リソース

このリソースに回復処理はありません。

# 6.6. IP Recovery Kit 管理ガイド

## IP Recovery Kit テクニカルドキュメンテーション

SIOS Protection Suite for Linux Internet Protocol (IP) Recovery Kit は、LifeKeeper 環境で、障害が発生したプライマリーサーバからバックアップサーバへ IP アドレスを復旧するメカニズムを提供します。IP Recovery Kit は、LifeKeeper によって保護されたアプリケーションへの接続に使用できる IP アドレスを定義できます。他の LifeKeeper リソースと同様に、IP リソースのスイッチオーバは、障害が発生した結果として自動的に開始するか、管理者の操作によって手動で開始することができます。

IP Recovery Kit は、論理インターフェース機能を使用した TCP/IP プロトコルスイートの実装をサポートし、追加のスタンバイネットワークインターフェースカードやダミーの IP アドレスを必要とせずに、IP アドレスのスイッチオーバとフェイルオーバを提供できるようにします。リリース 7.4 からは、IP Recovery Kit は IPv4 と IPv6 の両方のアドレスをサポートします。

- SIOS Protection Suite for Linux リリースノート
- SIOS Protection Suite for Linux テクニカルドキュメンテーション
- アプリケーションリカバリーキット

# 6.6.1. IP Recovery Kit 運用の原則

## 仮想 IP リソース

LifeKeeper は、仮想 IP アドレスをプライマリーサーバーの物理ネットワークインターフェースの 1 つに作成することによって、IP リソースを In Service が可能な状態にしています。ユーザーはこの仮想 IP アドレスを使用してノードに接続します。

IP Recovery Kit ソフトウェアは、選択されたアドレス、ネットワークマスク、インターフェースが確実に正しく機能できるかをチェックします。ソフトウェアがチェックする要素は以下のとおりです。

- 未使用のリソース: 新しい IP アドレスが、LifeKeeper クラスター内に存在する他のどの IP リソースにも、割り当てられていないことを確認します。
- 一意のアドレス: このアドレスは、ネットワーク上で「現在は」アクティブにできません。リソース作成中のチェックに加えて、リソースを In Service にする直前にも一意であることの確認が行われます。ネットワーク上で重複するアドレスが検出された場合、そのリソースを In Service にしません。

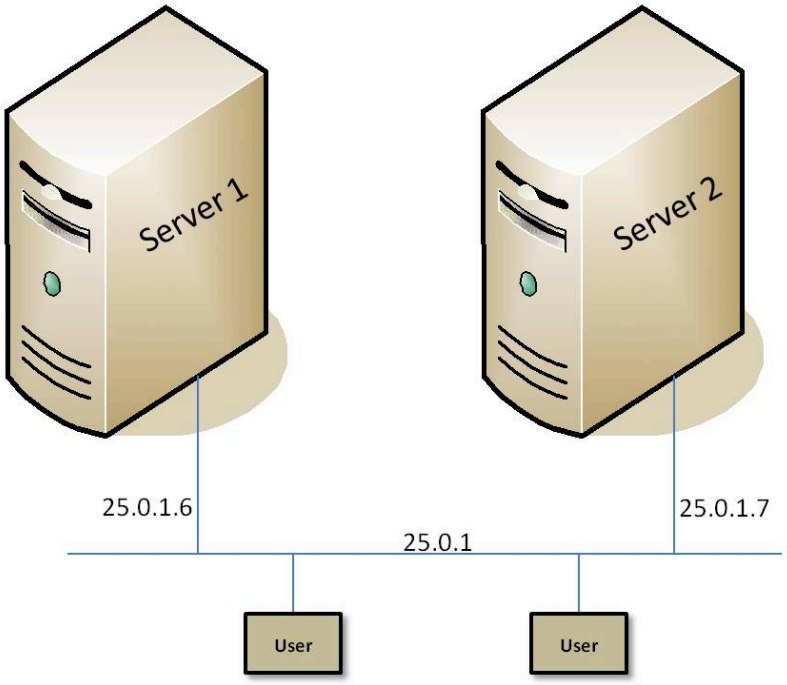
プライマリーサーバーに障害が発生すると、IP Recovery Kit は仮想 IP を、バックアップサーバーの物理ネットワークインターフェースの 1 つに設定することによって、その IP リソースをバックアップサーバーで In Service 状態にします。

リカバリー後はセッションコンテキストが失われているため、仮想 IP を利用するユーザーは最初に接続するのに使用した手順とまったく同じ手順で再接続する必要があります。

手動のスイッチオーバーでは、IP Recovery Kit はそのエイリアスアドレスをアクティブサーバーのサービスから削除した後、バックアップサーバーに追加します。

IP Recovery Kit の管理と運用について理解しやすいように、図 1 に示したシナリオについて考えてみましょう。この設定例には、Server1 と Server2 という 2 台のサーバーで構成されています。各サーバーは 1 つの LAN インターフェース (eth0) を備え、サブネット 25.0.1 に接続されています。ユーザーシステムもこのサブネット上にあります。Server1 および Server2 上の LAN インターフェースのアドレスは、それぞれ 25.0.1.6 と 25.0.1.7 です。

**図 1.管理および運用のシナリオ**




システム管理者は、 *ipname* という名前の IP リソースのエイリアスアドレスとして 25.0.1.10 を使用することにしました。システム管理者は、 */etc/hosts* ファイル (および使用されている場合は DNS) に以下のようなエントリーを作成します。

| | |
|---|---|
| 25.0.1.6 | server1 |
| 25.0.1.7 | server2 |
| 25.0.1.10 | ipname |

Server1 がこのリソースのプライマリーサーバーであれば、システム管理者はIP リソース階層の作成 セクションで説明しているウィザードを使用して、Server1 上に *ipname* 用の IP リソース階層を作成します。IP Recovery Kit は、 *ipname* (25.0.1.10) に関連付けられているアドレスを */etc/hosts* で見つけて使用可能であることを確認し、セカンダリーアドレスを Server1 上の eth0 で設定することによって、アドレスを In Service にします。これで Server1 上の eth0 は、 *server1* と *ipname* の両方に応答するようになります。

LifeKeeper7.3 以前のバージョンでは、新しいエイリアスアドレスは `ifconfig` もしくは `ip addr show` コマンドで確認することができました。LifeKeeper 7.4 からは、 `ip addr show` コマンドを使用してください (詳細については IPv6 既知の問題 を参照してください)。

ユーザーはその後、たとえば `telnet ipname` というように入力することで Server1 に接続できます。Server1 がクラッシュした場合、LifeKeeper は *ipname* のアドレスを Server2 の eth0 に自動的にスイッチオーバーします。Server1 上のユーザーセッションは終了します。ユーザーが telnet ipname を再実行した場合は、Server2 に接続されます。

*ipname* がどこでアクティブになって In Service であるかにかかわらず、アドレスとしての *server1* と *server2* はアクティブで、使用することができます。ただし、これらは LifeKeeper のリカバリーによって保護されていません。これらのアドレスは、切り替えられたアプリケーションに対してではなく、名前を使って特定のサーバーに対して接続する必要がある場合は常に使用できます。例えば、リモートシステム

管理や LifeKeeper コミュニケーションパスなどがあります(この場合、LifeKeeper コミュニケーションパスとして、例えば 25.0.1.6 と 25.0.1.7 が使用されます)。

## 実 IP リソース

LifeKeeper は、仮想 IP アドレスだけでなく、実 IP アドレス（ネットワークインターフェース 用に構成されたプライマリー IP アドレス）を保護することが可能です。これにより、Amazon Web Services (AWS) 環境において Recovery Kit for Route 53 と併用することで、仮想 IP アドレスを使わずに構成することができます。詳しくは Recovery Kit for Route 53 管理ガイド を参照してください。

## IP リソースの監視

LifeKeeper は以下の方法を順番に実行して、管理下にある IP リソースの状態を定期的に監視しています。

1. 該当するネットワークインターフェースにおいて、IP リソースがエイリアスとして設定されているかを確認します。

2. IP リソースが設定されているネットワークインターフェースのリンク状態を確認し、物理ネットワークに正しく接続されているかを検査します。

3. 保護された IP アドレスをソースアドレスとしてブロードキャスト ping テストを行い、IP リソースがネットワーク上で正しくデータの送受信を行えるかを確認します。

   ブロードキャスト ping テストはデフォルトのテストメカニズムです。保護された IP アドレスをソースアドレスとして、ブロードキャスト ping パケットを IP リソースに関連付けられたサブネットのブロードキャストアドレス宛に送信し、テストを行っています。ローカルシステム以外の任意のアドレスから応答を受信するとテストは成功したと判断します。

   ブロードキャスト ping に応答を返せるシステムがネットワーク上にない環境 (多くのシステムのデフォルト設定がこのような環境です) では、LifeKeeper はアドレスリストを定義し、ブロードキャスト ping の代わりに ping を送信するように設定可能です。このリストが定義されている場合、ブロードキャスト ping テストは省略され、リスト内のすべての IP アドレスに対し、同時に ping が実行されます。リスト内のいずれかの IP アドレスからの応答が受信されれば、テストは成功したと判断されます。この技術は、大規模なネットワークにおけるブロードキャストストームの抑止に有効です。

IP リソースの定期的なチェックの中で、これらのテストが失敗すると、LifeKeeper に障害が通知されます。LifeKeeper はまずローカルリカバリーを試み、IP リソースをローカルノード上でリストアしようとします。ローカルリカバリー手順の詳細については、IP ローカルリカバリーと設定に関する考慮事項 を確認してください。ローカルリカバリーが現在のノードでの IP リソースの起動に失敗した場合、LifeKeeper は IP リソースを含めたアプリケーション階層を、クラスター内の別の LifeKeeper システムに移動させます。

> 注意：実 IP リソースの場合、ローカルリカバリーは実施されず、強制的に失敗と判断

し、バックアップサーバーへのフェイルオーバーを実行します。

LifeKeeper は同じヘルスチェック機能を使い、 In Service の状態になった直後に IP リソースが正しく動作するかを判定します。チェックの結果、障害が確認された場合、起動は失敗します。

IP ヘルスチェックメカニズムは設定の変更とチューニングを実行することが可能です。詳細については、IP 構成の確認および編集 およびIP Recovery Kit のチューニング を確認してください。

# 6.6.2. IP Recovery Kit の要件

IP Recovery Kit のインストールやアンインストールを試みる前に、このパッケージのハードウェア / ソフトウェア要件と、インストールとアンインストールの手順を理解しておく必要があります。

## ハードウェア / ソフトウェア要件

LifeKeeper IP Recovery Kit をインストールして設定する前に、使用している設定が以下の要件を満たしていることを確認してください。

- **サーバ**：この Recovery Kit を使用するには、LifeKeeper for Linux テクニカルドキュメンテーション、およびLifeKeeper for Linux リリースノート で説明されている LifeKeeper の要件に従って設定したコンピュータが 2 台以上必要です。

- **LifeKeeper ソフトウェア** ：各サーバに同じバージョンの LifeKeeper ソフトウェアとパッチをインストールする必要があります。具体的な LifeKeeper の要件については、LifeKeeper for Linux テクニカルドキュメンテーション および LifeKeeper for Linux リリースノート を参照してください。

- **LifeKeeper IP Recovery Kit** ：同じバージョンの Recovery Kit を各サーバにインストールする必要があります。

- **IP ネットワークインターフェース** ：各サーバに、イーサーネット TCP/IP をサポートするネットワークインターフェースが 1 つ以上必要です。IP スイッチオーバーが正しく動作するには、ローカルネットワークに接続されているユーザシステムが標準の TCP/IP 仕様に準拠している必要があります。このインターフェースは設定する必要があります。 *ifcfg** ファイルが存在しない場合、インターフェースがダウンしたときに IP スイッチオーバーが失敗する可能性があります。

  **注記**：各サーバが必要とするネットワークインターフェースが 1 つだけであっても、異種メディア要件、スループット要件、単一障害点の排除、ネットワークのセグメンテーション、ローカルリカバリーのサポートといった多くの理由で、複数のインターフェースを使用すべきです。

  また、IP リソースの設定に使用する NIC には、あらかじめ OS の操作を使用して実 IP アドレスを設定しておいてください。その実 IP アドレスを使用して、IP ネットワーク上の基本的な疎通確認等を実施するようにしてください。

  動作確認を実施しているチャンネルボンディング、ネットワークチーミングの構成については、こちら を参照してください。

- **TCP/IP ソフトウェア:** 各サーバに TCP/IP ソフトウェアも必要です。

最新のリリースの互換性および購入方法については、LifeKeeper for Linux リリースノート を参照するか、営業担当者にお問い合わせください。

LifeKeeper IP Recovery Kit ソフトウェアのインストール / アンインストール方法の具体的な手順については、LifeKeeper for Linux インストールガイド を参照してください。

# 6.6.3. IP Recovery Kit の設定

必要とする保護と柔軟性が得られるように LifeKeeper を設定するには、設定ルールを把握している必要があります。設定を適切にプランニングするには、ネットワーク設定、インターフェースの選択、ユーザーシステムのセットアップ、階層オプション、IP 設定作業について理解することが必要です。このセクションでは、設定のプランニングに加え、Recovery Kit を設定するために必要な特定の作業についても、設定例を示して説明します。

## LifeKeeper での TCP/IP の設定

このセクションでは、TCP/IP の設定を開始する前に考慮する必要のある情報と、典型的な LifeKeeper IP 設定の例を紹介します。

LifeKeeper Core のリソース階層の設定に関する手順については、LifeKeeper for Linux テクニカルドキュメンテーション を参照してください。

## TCP/IP 固有の設定上の考慮事項

IP Recovery Kit を正しく設定するために、以下のトピックを見直して、設定作業を完了するために必要な情報を用意してあることを確認してください。

- インターフェースの選択
- ユーザーシステムのセットアップ
- 一般的な IP プランニングの考慮事項

更なる設定上の考慮事項とサンプルについては、以下のトピックを参照してください。

- IP のリソース監視と設定に関する考慮事項
- IP のローカルリカバリーと設定に関する考慮事項
- 設定例

## LifeKeeper の設定作業

次の IP リソースの設定作業は、IP リソースインスタンス特有のもので、Recovery Kit ごとに異なるため、本セクションで説明しています。

- リソース階層の作成：アプリケーションリソース階層を LifeKeeper クラスターに作成します。
- リソース階層の削除：リソース階層を LifeKeeper クラスター内のすべてのサーバーから削除します。
- リソース階層の拡張：リソース階層をプライマリーサーバーからバックアップサーバーへ拡張しま

す。

- リソース階層の拡張解除：リソース階層を LifeKeeper クラスター内の 1 つのサーバーから拡張解除(削除) します。
- リソース階層のテスト：IP リソース階層が正しく構成されているか、動作するかをテストします。
- IP 設定の確認および編集：IP リソースの設定を表示し、それらを修正することが可能です。
- IP Recovery Kit のチューニング：IP Recovery Kit の動作を調整します。

次の作業は、すべての Recovery Kit に共通する手順のため、LifeKeeper for Linux テクニカルドキュメンテーション の管理 セクションに記載されています。

- リソース依存関係の作成：既存のリソース階層と別のリソースインスタンスとの間に親子の依存関係を作成し、クラスター内のすべての対象サーバーに依存関係の変更を反映します。
- リソース依存関係の削除：リソースの依存関係を削除して、クラスター内のすべての対象サーバーに依存関係の変更を反映します。
- In Service：リソース階層を特定のサーバーで In Service にします。
- Out of Service：リソース階層を特定のサーバーで Out of Service にします。
- プロパティの表示 / 編集：特定のサーバーでリソース階層のプロパティを表示または編集します。

**注意:** このセクションの残りの部分では、LifeKeeper GUI の **[Edit]** メニューから作業を選択することによって、Recovery Kit を設定する方法を説明します。設定作業はツールバーから選択することもできます。状況表示ウィンドウのリソース階層ツリー(左側のペイン)のグローバルリソースを右クリックして、**[Edit]** メニューと同じドロップダウンメニューの選択項目を表示することもできます。言うまでもなく、これは階層がすでに存在している場合にだけ可能な方法です。

状況表示ウィンドウのリソース階層表(右側のペイン)のリソースインスタンスを右クリックして、サーバーおよび特定リソースの状況に応じて、リソース階層の作成を除くすべての設定作業を実行することもできます。

# 6.6.3.1. IP インターフェースの選択

IP リソースを作成するときには、IP リソースのアドレス、そのアドレスで使用するネットマスク、ネットワークインターフェースを選択します。すべての組み合わせが許されるわけではありません。指定するアドレスとネットマスクのペアと、現在 In Service になっているアドレスとネットマスクのすべてのペアによって選択肢が決まります。また、Recovery Kit の IP ローカルリカバリ機能を使用する予定がある場合、IP ローカルリカバリ セクションで追加の設定上の考慮事項を参照してください。

選択したアドレスとネットマスクのペアにより、リソースのサブネットが決まります。同じサブネット上で、別のアドレス (物理インターフェースアドレスまたは論理インターフェースアドレス) がいずれかのインターフェース上で現在 In Service である場合、そのインターフェースで IP リソースを設定する必要があります。ソフトウェアはテストを実行し、現在のネットワーク設定に基づいて、可能な選択肢を特定します。提示される選択肢からは、任意のものを選択できます。

IP リカバリキットソフトウェアが物理メディアのタイプを区別しないため、リソースが使用する物理ネットワークを決定し、適切にアドレスを選択してください。例えば、サブネット xx.yy.12 のイーサーネットバックボーンと、サブネット xx.yy.20 および xx.yy.30 のイーサーネット LAN に接続されたサーバがあるとします。最初のイーサーネットサブネット上にリソースを作成する場合は、xx.yy.20.120 のように、目的のサブネット上のアドレスを選択します。

一般的に、IP リカバリキットソフトウェアがあらゆる値のネットマスクの選択を許可したとしても、複数のネットマスクがパケットのミスルーティングを引き起こす原因となりうるため、同じ物理的なインターフェースに対して複数のネットマスクを選択することは避けなければなりません。

もう 1 つの考慮点として、すべての LifeKeeper サーバで、インターフェースの選択に一貫性を持たせる必要があります。サーバ A で、1 つのインターフェースに IP リソースをいくつか設定した場合は、サーバ B でも、1 つのインターフェースにそれらの IP リソースを設定する必要があります。

IP リソース階層を作成する時は、起動時に最初に立ち上がっていてプライマリおよびバックアップ側の両ノードで設定が一致しているネットワークインターフェースを利用するといいでしょう。例えば eth1 を利用する場合は、IP アドレスが割り当てられていないような構成、またはサーバ起動直後のいくつかのネットワークインターフェースが起動していないような時でも、eth1 は起動している必要があり、 *ifcfg-eth1* ファイルは両ノードで正しく設定されていなければなりません (`ifup/ifdown ifcfg-eth1` コマンドで確認する)。

# 6.6.3.2. IP ユーザーシステムのセットアップ

1 つの IP リソースが、2 台のサーバ間で IP Recovery Kit ソフトウェアによって切り替えられる場合、インターフェースが変更されるため、切り替えられる IP アドレスに対応する MAC アドレスが変化します。LAN 上のルータやユーザシステムは、この変更をそれぞれの ARP テーブルに反映しないと、新しい場所の IP アドレスにアクセスできません。オペレーティングシステムによっては、ネットワークインターフェースに新しい IP アドレスが追加されると、ARP パケットが自動的に送信され、サブネット上の全クライアントの ARP テーブルがアップデートされます。Linux にこの機能はありません。そのため LifeKeeper は、切り替え可能な IP アドレスをインターフェースに追加したら ARP パケットを送信し、クライアントの ARP キャッシュを強制的にアップデートする必要があります。

TCP/IP の実装は、このARP パケットに応じて必要な ARP のアップデートを実行する機能の点でさまざまです。以下に、重要な例の一部の説明を示します。

- **Linux TCP/IP の全機能の実装:** Linux や他のオペレーティングシステムのほとんどが備える TCP/IP の実装では、システムが ARP 要求パケットを受信したときに ARP キャッシュをアップデートする機能がサポートされています。LifeKeeper は前述のようにこの機能を利用して、それらのシステムの ARP キャッシュを強制的にアップデートします。

- **ARP キャッシュ:** 上記のような ARP の実装をサポートしていなくても ARP キャッシュはサポートしているユーザシステムは、通常、キャッシュ関連のタイマを備え、内容の新しさをあるレベルに維持しています。一部の実装では、タイマの値を減らすことによって、変更されたアドレスのマッピングがそのユーザシステムに反映されるのに必要な時間を最小限にすることができます。LAN 上のユーザー数が少ない場合は、この方法を採用しても構いません。タイマの値を減らす必要がないシステムもあります。たとえば、Windows NT の TCP 実装が使用するタイマ値は 10 秒なので、値を変更する必要はないでしょう。

- 静的アドレスマッピング: 動的 ARP キャッシュを持たないシステムや、キャッシュのタイミングを調整できないシステムでは、ルータを使用してマッピングの変更を処理できます。そのようなユーザシステムは、ルータ (ゲートウェイ) 経由で IP リソースのサブネットにアクセスします。この設定では、キャッシュのアップデートが必要なのはリソースサブネットに直接接続されているルータだけで、ユーザシステム自体で必要な変更はありません。

# 6.6.3.3. 一般的な IP プランニングの考慮事項

IP リソース階層で使用する予定のアドレス、ネットマスク、関連するホスト名 / ドメイン名を選択したら、各サーバの */etc/hosts* ファイルに適切なエントリを追加します。DNS を使用している場合は、DNS にもエントリを追加します。

**注記:** DNS を使用している場合でも、すべての LifeKeeper サーバ上のローカル */etc/hosts* ファイルに IP リソースのエントリを記述することを強くお勧めします。それによって、リカバリ時間が短縮されます。ただし、IP インスタンスを作成するときに入力するリソース名が IP アドレス自体の場合、ホストファイルのエントリは不要です。

保護された IP アドレスは、恒久的な論理インターフェースを作成する場合のように、システムのブート時にアクティブになるようシステムに設定しないでください。LifeKeeper ソフトウェアはシステムに代わってそれらの管理を行います。

いずれかのリソースアドレスが新しい(論理)サブネット上にある場合は、それらのサブネットへのルーティングを処理するようにルータをアップデートしてください。

# 6.6.3.4. IP のリソース監視と設定に関する考慮事項

デフォルトでは、LifeKeeper IP Recovery Kit は、IP アドレスの論理サブネットでブロードキャスト ping を実行し、応答を確認することによって IP リソースを監視します。このテストが正しく動作するには、IP リソースと同じ論理サブネットに属する IP アドレスを持ち、ブロードキャスト ping に応答できる別のシステムが、物理ネットワーク上に少なくとも 1 つ存在する必要があります。この要件を満たすには、ネットワーク上で常に 1 つ以上のシステムが利用可能なようにする必要があります。同じ論理サブネット上にルータを配置すれば、通常は十分にこの要件が満たされます。多くのデバイスにおいて、デフォルトの設定ではブロードキャスト ping に応答しないことに注意してください。そのため、少なくとも 1 つのデバイスの設定を変更しておく必要があるでしょう。

この要件を満たせない場合は、ブロードキャスト ping を完全に無効にするか、ブロードキャスト ping に代わる手段として確認される IP アドレスの静的リストを設定してください。これらの設定方法に関する追加情報については、IP Recovery Kit のチューニング および IP 構成の確認および編集 を参照してください。

# 6.6.3.5. IP ローカルリカバリーと設定に関する考慮事項

HA の構成におけるネットワークインターフェースの冗長性を確保するために、Linux 標準の NIC のボンディングの利用をお勧めします。LifeKeeper IP Recovery Kit は、ボンディングインターフェース上に作成された仮想 IP アドレスを完全にサポートします。

## ローカルリカバリーのシナリオ

IP Recovery Kit によって IP アドレスの障害が検出されると、結果として生じる障害によって IP ローカルリカバリースクリプトが起動されます。LifeKeeper は最初に、その IP アドレスを現在のネットワークインターフェース上で In Service の状態に戻そうとします。ローカルでリカバリーの試みが失敗すると、LifeKeeper は IP リソースと依存関係を持つすべてのリソースのバックアップサーバーへのフェイルオーバーを実行します。

✻ 注意：実 IP リソースの場合、ローカルリカバリーは実施されず、強制的に失敗と判断し、バックアップサーバーへのフェイルオーバーを実行します。

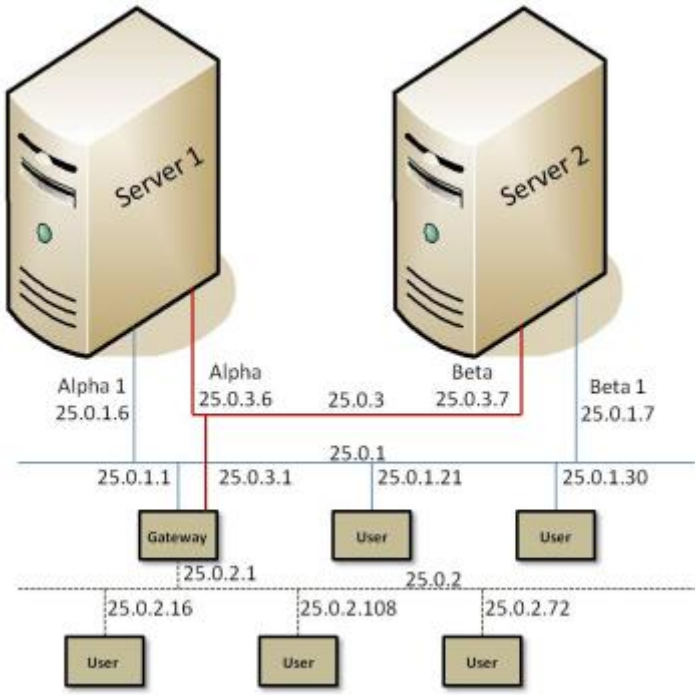
# 6.6.3.6. 設定例

このトピックでは、ネットワークの設定例を示し、2 種類の IP 設定例について説明します。最初の例は、単一の IP リソースに依存して既存のサブネットに設定されている、データベースアプリケーションの典型的な例を示しています。2 番目の例は、複数の IP リソースが設定されたアクティブ / アクティブのシナリオを示しています。

## ネットワーク設定

最初の 2 つの設定例は、以下の図に示すネットワーク設定を前提にしています。

ネットワーク設定




このネットワーク設定には以下のコンポーネントが存在します。

- サーバー。 Server 1 と Server 2 の 2 台のサーバーがあり、それぞれに適切な LifeKeeper およびアプリケーションソフトウェアがインストールされています。
- インターフェース。 各サーバーは、次のように設定された 2 つのイーサーネットインターフェース、eth0 と eth1 を備えています。

| インターフェース | Server 1 | Server 2 |
|---|---|---|
| eth0 | Server1<br><br>25.0.3.6 | Server2<br><br>25.0.3.7 |

| | | |
|---|---|---|
| eth1 | Server11<br><br>25.0.1.6 | Server21<br><br>25.0.1.7 |

- **ネットワーク。** ネットワークは、以下の 3 つのサブネットワークで構成されています。
  - 主にサーバー用の、トラフィックが少ないバックボーン(25.0.3)
  - サーバーとクライアント両方のための、トラフィックが多いバックボーン(25.0.1)
  - トラフィックが多いクライアントネットワーク(25.0.2)

    1 つのゲートウェイで、すべての LAN 間の相互接続ルーティングを行います。アドレス解決には DNS(図示されていない)が使用されます。

- **ハートビート。** TCP ハートビートコミュニケーションパスは、片方または両方のサーバーサブネットワークを使用して設定されます。

## 典型的な設定例

Server 1 と Server 2 は、共有ディスクに常駐する mydatabase と呼ばれるアプリケーションにアクセスします。mydatabase アプリケーションと、このアプリケーションへのアクセスに使用される IP リソースが一緒に切り替えられるようにするため、システム管理者は mydatabase アプリケーションリソースを作成し、依存関係で IP リソースをアプリケーション階層に追加します。

以下のような設定上の問題があります。

- **アプリケーション階層。** アプリケーション階層が存在しなければ、管理者はそれを IP リソースの親として指定できません。この例では、Server 1 がプライマリーサーバーです。アプリケーションリソースのタグは、mydatabase-on-server1 と mydatabase-on-server2 です。

- **IP リソース名。** システム管理者は、IP リソースの名前とアドレスを Server 1 と Server 2 の */etc/hosts* ファイル、および DNS データベースに追加します。この例では、IP リソース名は databaseip、そのネットワークアドレスは 25.0.1.2 です。名前と IP アドレスの関連付けが必要でない場合、この作業は不要です。

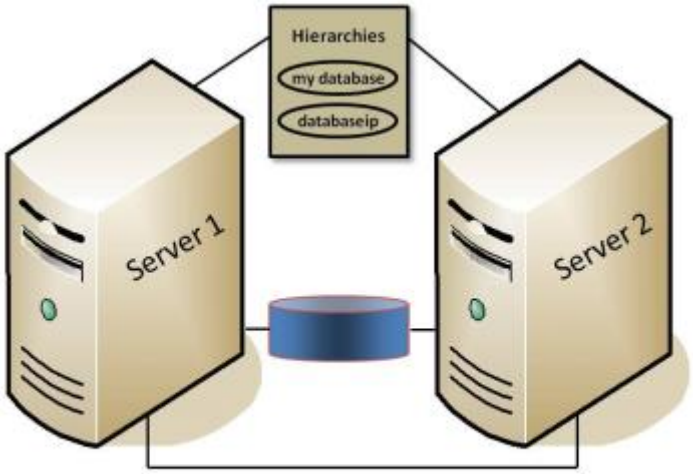
- **ルータ、ゲートウェイ、ユーザー。** databaseip は既存のサブネット上のアドレスであるため、追加の設定は不要です。IP リソースは、25.0.1 サブネット上にあります。すべてのユーザーが、現在 25.0.1 サブネットに到達するために使用しているルートを経由して databaseip に接続します。たとえば、25.0.2 のユーザーはゲートウェイ経由で接続し、25.0.1 のユーザーは直接接続します。

- **IP インスタンスの定義。** システム管理者が [Resource Hierarchy Create] 画面で IP リソースとして databaseip を入力すると、ソフトウェアによっていくつかのテストが実行されます。Server 1 が、databaseip に付随するアドレスを特定可能かどうかが確認されます(アドレスは hosts ファイル内に

あるか、DNS から検索できます)。検索されたアドレス 25.0.1.2 がすでに使用されていないかも検査されます。IP リソースは 25.0.1 サブネット上にあるため、IP リカバリーソフトウェアは、そのリソースが eth1 インターフェースに設定されていることを確認します。IP リソースに問題がなければ、ウィザードの残りのダイアログボックスに、図 3 の下の表に示したようなデフォルト値が設定されます。すべてのデフォルト値を選択すると、独立した IP リソース階層が ip-databaseip という名前で作成されます。

**注意:** それぞれの設定図に付随する表は、プライマリーサーバー(Server 1)の場合は [Create Resource Hierarchy] ウィザードに、バックアップサーバー(Server 2)の場合は [Extend Resource Hierarchy] ウィザードに入力される適切な情報の例を示しています。ウィザードに入力する情報の詳細については、このセクションで後述するLifeKeeper 設定作業 セクションを参照してください。これらの表は、Recovery Kit を設定するときに役立つ参照情報となります。

**図 3.IP リソース作成時の典型的な設定例**




設定に関する注意:

1. アプリケーションリソースは mydatabase-on-server1 です。
2. IP リソースは databaseip、タグ名は ip-databaseip です。
3. mydatabase-on-server1 が障害を起こすと、LifeKeeper はそれを Server 2 に切り替えます (ipdatabaseip は、依存関係がある場合にのみ切り替えられます)。
4. Server 1 が障害を起こすと、両方のリソースが Server 2 に切り替えられます。
5. スイッチオーバーの間に、databaseip のユーザーは切断されます。ユーザーは再度ログインすると、Server 2 のすべてのアプリケーションにアクセスできます。
6. 手動によるスイッチオーバーの間、databaseip 以外の接続を使って Server 1 に接続しているユーザーについては、Server 1 との接続が維持されます。

### Server 1 での IP リソース階層の作成

| | |
|---|---|
| Server: | Server1 |
| IP Resource: | databaseip |
| Netmask: | 255.255.252.0 |
| Network Interface: | eth1 |
| IP Resource Tag: | ip-databaseip |

### IP リソース階層の Server 2 への拡張

| | |
|---|---|
| Template Server: | Server1 |
| Tag to Extend: | databaseip |
| Target Server: | Server2 |
| Target Priority: | 10 |
| **IP Resource: | 25.0.1.2 |
| Netmask: | 255.255.252.0 |
| Network Interface: | eth1 |
| IP Resource Tag: | ip-databaseip |

** [Extend Wizard] では、その DNS 名に関連付けられている実際の IP アドレスが IP リソースとして表示されます。

## IP リソースのテスト

IP リソースが正常に作成されたことを確認するために、システム管理者は以下の作業を行う必要があります。

1. LifeKeeper の GUI メニューから、ip-databaseip が Server 1 で In Service (ISP)かどうかを確かめます。
2. リモートサーバーから、ping または telnet を使用してアドレス databaseip に接続します。
3. Server 2 で in_service オプションを選択し、さらに ip-databaseip を選択して、手動のスイッチオーバーをテストします。IP アドレスが Server 2 に移行されることを確認してください。

# アクティブ / アクティブ設定の例

同じネットワーク設定を使用した 2 番目の例では、各サーバーで 1 つずつアクティブになった 2 つの IP リソースを示します。

## リソースアドレス

この例では、IP リソースは server1ip (アドレス 25.0.6.20)と server2ip (アドレス 25.0.6.21)です。これらのリソースのエントリーが、各サーバーの */etc/hosts* ファイルと、DNS データベースになければなりません。

## ルータの設定

選択されたアドレスは新しい(論理)サブネット上にあるため、eth0 または eth1 に設定できます。ただし、どちらも同じインターフェースを使用している必要があります。

この例では、eth0 を選択すると、すべてのユーザーがゲートウェイを経由しなければならなくなります。eth1 を選択すると、サブネット 25.0.1 のユーザーはリソースに直接アクセスできます(新しいサブネットが内部のルーティングテーブルに追加されていることが前提です)。サブネット 25.0.2 上のユーザーには、やはりゲートウェイが必要です。この例の場合、選択されたインターフェースは eth1 です。

新しいサブネットをサポートするために、どの物理ネットワークが選択されたかにかかわりなく、ネットワーク管理者はルーティング情報をゲートウェイシステムに追加してから IP リソースを作成する必要があります

## 最初の IP リソース定義

管理者は、最初の IP リソースを Server 1 に作成します。eth0 は、各サーバーで利用可能な最初のインターフェースであり、デフォルトとして提示されます。eth1 をインターフェースとして定義するときは、利用可能なインターフェースの一覧から選択します。

### Server 1 での IP リソース階層の作成

| | |
|---|---|
| Server: | Server1 |
| IP Resource: | server1ip |
| Netmask: | 255.255.252.0 |
| Network Interface: | eth1 |
| IP Resource Tag: | ip-server1ip |

### IP リソース階層の Server 2 への拡張

| | |
|---|---|
| Template Server: | Server1 |
| Tag to Extend: | server1ip |
| Target Server: | Server2 |
| Target Priority: | 10 |

| | |
|---|---|
| **IP Resouce: | 25.0.6.20 |
| Netmask: | 255.255.252.0 |
| Network Interface: | eth1 |
| IP Resource Tag: | ip-server1ip |

** [Extend Wizard] では、その DNS 名に関連付けられている実際の IP アドレスが IP リソースとして表示されます。

## 2 番目の IP リソース定義

管理者は、2 番目の IP リソースを Server 2 に作成します。eth0 は、各サーバーで利用可能な最初のインターフェースであり、デフォルトとして提示されます。eth1 をインターフェースとして定義するときは、利用可能なインターフェースの一覧から選択します。

### Server 2 での IP リソース階層の作成

| | |
|---|---|
| Server: | Server2 |
| IP Resource: | server2ip |
| Netmask: | 255.255.252.0 |
| Network Interface: | eth1 |
| IP Resource Tag: | ip-server2ip |

### IP リソース階層の Server 1 への拡張

| | |
|---|---|
| Template Server: | Server2 |
| Tag to Extend: | server2ip |
| Target Server: | Server1 |
| Target Priority: | 10 |
| **IP Resouce: | 25.0.6.21 |
| Netmask: | 255.255.252.0 |

| | |
|---|---|
| Network Interface: | eth1 |
| IP Resource Tag: | ip-server2ip |

** [Extend Wizard] では、その DNS 名に関連付けられている実際の IP アドレスが IP リソースとして表示されます。

**注意:** サブネット 25.0.6 は Server 2 ではアクティブでないため、eth0 と eth1 の両方をプライマリーネットワークインターフェースのために選択できます。Server 1(バックアップサーバー)では、最初の IP リソース 25.0.6.20 が In Service であるため、eth1 だけが選択可能です。管理者が定義を保存すると、LifeKeeper はアドレス 25.0.6.21 を Server2 の eth1 で In Service にします。

## IP リソースのテスト

管理者は以下のテストを行って、新しいリソースが両方のサーバーで機能していることを確認する必要があります。

1. プライマリーサーバーの各リソースについて、ping または telnet を使用してそれぞれにアクセス可能であるか確認します。管理者は、すべてのユーザーサイトからの接続をテストすることもできます。

2. 手動で ip-server1ip を Server 2 で In Service にして、スイッチオーバーをテストします。両方のリソースが Server 2 で機能していることを確認してください。

3. 両方のリソースを Server 1 で In Service にします。両方のリソースが Server 1 で機能していることを確認してください。

4. ip-server2ip を、プライマリーサーバーの Server 2 で In Service に戻します。

# 6.6.3.7. IP リソース階層の作成

プライマリーサーバーからリソースインスタンスを作成するには、以下の手順を完了する必要があります。

1. LifeKeeper GUI メニューから **[Edit]** を選択し、次に **[Server]** を選択します。ドロップダウンメニューから、 **[Create Resource Hierarchy]** を選択してください。

2. ダイアログボックスが表示され、クラスター内にインストールされ認識されているすべての Recovery Kit が、ドロップダウンリストボックスに一覧で示されます。ドロップダウンリストから IP を選択し **[Next]** をクリックします。

3. 以下の情報を入力するようにプロンプトが表示されます。ダイアログボックスで **[Back]** ボタンが有効な場合は、前のダイアログボックスに戻ることができます。これは、エラーが発生して、以前に入力した情報を訂正する必要がある場合に特に役立ちます。

   階層作成の途中で **[Cancel]** ボタンをクリックすると、作成処理全体が取り消されます。

| フィールド | ヒント |
|---|---|
| **Switchback Type** | バックアップサーバーへのフェイルオーバー後にサーバーが復帰したときに、このサーバーに IP インスタンスを戻す方法を指定します。 *intelligent* または *automatic* を選択できます。 *intelligent* の場合、インスタンスをプライマリー / オリジナルサーバーにスイッチバックするときに管理者の介入が必要になります。 *automatic* の場合、プライマリーサーバーがオンラインに戻り、LifeKeeper コミュニケーションパスを再確立した直後に自動的にスイッチバックが行われます。スイッチバックタイプは、必要な場合、[Resource Properties] ダイアログボックスの [General] タブで後から変更できます。 |
| **Server** | IP アドレスを配置する サーバー (通常これをプライマリーサーバーまたはテンプレートサーバーと呼ぶ) を選択します。クラスター内のすべてのサーバーがドロップダウンリストに表示されます。 |
| **IP Resource** | 実際に保護する仮想 IP アドレスを入力します。これは、LifeKeeper がこのリソースのために使用する IP アドレスまたはシンボル名です。<br><br>また、仮想 IP アドレスだけでなく、物理ネットワークインターフェースに割り当てられた実 IP アドレスの保護を可能とします。実 IP アドレスを保護する場合は'0.0.0.0' を入力します。実 IP アドレスを保護する場合、後ほど選択するネットワークインターフェースに割り当てられた IP アドレスを保護します。<br><br>クライアントは、後ほど設定するネットワークインターフェース経由で、この IP アドレスに接続します。シンボル名を使用する場合、そのシンボル名は、ローカルの /etc/hosts ファイル内に存在するか、DNS によってアクセスできる必要があります。エイリアス名およびドメイン名は、上に示した条件を満たしていれば使用できます。この情報フィールドにデフォルト値はありません。<br><br>**注意:** シンボル名を使用する場合、このリソースを拡張するときに、実際の IP アドレ |

| | |
|---|---|
| | スが IP リソースの指定としてダイアログボックスに表示されるため注意が必要です。<br><br>**注意:** 実 IP リソースを利用する場合、物理ネットワークインターフェースに実 IP が割り当てられている必要があります。実 IP リソースは、 IPv4 のみサポートされます。 |
| **Netmask** | ネットワークマスク **(ネットマスク)** を選択または入力します。IP リソースは、このネットマスクをターゲットサーバーで使用します。有効な値は、特定の IP リソースアドレスのクラスを表す標準のネットマスクです。<br><br>IP Resource の項目で実 IP アドレスの保護（'0.0.0.0'）を指定した場合、ネットマスクの項目は表示されません。<br><br>**注意:** ここで選択したネットマスクと IP アドレスの組み合わせによって、IP リソースが使用する、ネットワーク設定と一貫性があるサブネットが決まります。 |
| **Network Interface** | LifeKeeper に保護されて IP リソースが配置される **ネットワークインターフェース** を選択または入力します。これは、IP アドレスがインターフェースで接続する物理イーサーネットカードです。選択が有効かどうかは、既存のネットワーク設定と、IP リソースアドレスおよびネットマスクのために選択された値によって決まります。デフォルト値は、一連の有効な選択項目のうち、手順 4 と 5 で選択したアドレスとネットマスクの値に最もよく一致するインターフェースです。 |
| **IP Resource Tag** | 作成している IP リソースインスタンスに固有の IP リソースタグ 名を選択または入力します。このフィールドには、デフォルトで以下の値が自動的に入力されます。<br><br>仮想 IP リソース：ip-<resource> (<resource> はリソース名または IP アドレスです。)<br>実 IP リソース　：realip-<network interface> (ネットワークインターフェースの項目で選択した値です。)<br><br>このタグは必要に応じて変更できます。 |

4. **[Create]** をクリックします。 **[Create Resource Wizard]** によって、IP リソースが作成されます。

5. この時点で情報ボックスが表示され、LifeKeeper は、IP リソース階層を作成するために有効なデータが指定されたかどうかを検証します。LifeKeeper が問題を検知した場合は、情報ボックスにエラーが表示されます。検証に成功すると、リソースが作成されます。 **[Next]** をクリックします。

6. 別の情報ボックスが表示され、IP リソース階層が正常に作成されたことが示されます。IP リソース階層を LifeKeeper の保護下に置くには、その階層をクラスター内の別のサーバーに拡張する必要があります。

**[Continue]** をクリックすると、 **[Pre-Extend configuration task]** が起動されます。リソース階層を別のサーバーに拡張する方法の詳細については、階層の拡張 を参照してください。

ここで **[Cancel]** をクリックすると、別のダイアログボックスが表示され、IP リソース階層を LifeKeeper の保護下に置くには、いずれかの時点でこの手順に戻り、階層を別のサーバーに拡張する必要があるという警告が表示されます。

# 6.6.3.8. リソース階層の削除

LifeKeeper 環境内のすべてのサーバからリソース階層を削除するには、以下の手順を実行します。

1. LifeKeeper GUI メニューから **[Edit]** を選択し、次に **[Resouce]** を選択します。ドロップダウンメニューから、 **[Delete Resource Hierarchy]** を選択します。

2. **[Target Server]** で、IP リソース階層を削除するターゲットサーバの名前を選択し、 **[Next]** をクリックします。(このダイアログボックスは、いずれかのペインでリソースインスタンスを右クリックして、 **[Delete Resource]** を選択した場合には表示されません。)

3. **[Hierarchy to Delete]** を選択します。削除するリソース階層を特定して、強調表示にします。 **[Next]** をクリックします。(このダイアログは、右ペインまたは左ペインのリソースインスタンス上で右クリックして **[Delete Resource]** を選択した場合には表示されません。)

4. 選択したターゲットサーバと、削除の対象として選択した階層を確認する情報ボックスが表示されます。 **[Delete]** をクリックして、次のダイアログボックスに進みます。

5. IP リソースが正常に削除されたことを確認する別の情報ボックスが表示されます。

6. **[Done]** をクリックして、 **[Delete Resource Hierarchy]** メニューの選択を終了します。

# 6.6.3.9. リソース階層の拡張

階層を作成後、クラスター内の別のサーバーに拡張する必要があります。3 通りのシナリオで、テンプレートサーバーからターゲットサーバーにリソースインスタンスを拡張できます。最初のシナリオは、リソース作成後、[Continue] をクリックして、別のサーバーにリソースを拡張する場合です。2 番目のシナリオは、以下に示すように [Edit] メニューから [Extend Resource Hierarchy] を選択する場合です。3 番目のシナリオは、左側または右側のペインから拡張されていない階層を右クリックする場合です。どのシナリオでも同じダイアログボックスが表示されます (いくつかの例外については、以下に詳細を明記する)。

1. LifeKeeperGUI メニューから ***[Extend wizard]*** を開始する場合は、 **[Edit]** を選択し、次に **[Resource]** を選択します。ドロップダウンメニューから **[Extend Resource Hierarchy]** を選択します。これで ***[Extend Resource Hierarchy wizard]*** が起動されます。拡張操作に慣れていない場合は、 **[Next]** をクリックしてください。LifeKeeper の **[Extend Resource Hierarchy]** のデフォルト値が分かっていて、入力と確認を省略する場合は **[Accept Defaults]** をクリックしてください。

2. ***Pre-Extend Wizard*** に以下の情報を入力します。 **注意**: 最初の 2 つのフィールドは **[Edit]** メニューの **[Extend]** から操作を開始した場合にだけ表示されます。階層を拡張する手順の間に **[Cancel]** をクリックすると、どの時点であってもそのサーバーへの拡張処理がキャンセルされますので注意してください。ただし、すでにリソースを別のサーバーに拡張している場合は、明示的に拡張解除するまで、そのインスタンスの拡張は有効です。

| フィールド | ヒント |
|---|---|
| **Template Server** | IP リソースが現在 In Service のサーバーを入力する。 |
| **Tag to Extend** | 拡張する IP リソースを選択する。これは、テンプレートサーバーからターゲットサーバーに拡張する IP インスタンスの名前です。ウィザードのドロップダウンリストボックスに、テンプレートサーバー上に作成した全リソースのうち、前のダイアログボックスで選択したリソースの一覧が表示されます。 |
| **Target Server** | IP リソース階層を拡張する **ターゲットサーバー** を選択してください。ドロップダウンボックスには、選択した階層に含まれないクラスター内のサーバー名の一覧が表示されます。 |
| **Switchback Type** | **スイッチバックタイプ** を選択してください。これは、バックアップサーバーへのフェイルオーバー後に IP インスタンスが In Service に戻る際に、このサーバーに IP インスタンスをスイッチバックする方法を指定します。 *[intelligent]* または *[automatic]* を選択できます。intelligent の場合、インスタンスをプライマリー / オリジナルサーバーにスイッチバックするときに管理者の介入が必要になります。automatic の場合、プライマリーサーバーがオンラインに戻り、LifeKeeper コミュニケーションパスを再確立した直後に自動的にスイッチバックが行われます。スイッチバックタイプは、必要に応じて [Resource Properties] ダイアログボックスの [General] タブで後から変更できます。 |
| **Template Priority** | **テンプレートの優先順位** を選択するか、入力してください。これはサーバーで現在サービス中の IP 階層の優先順位です。優先順位は、1 ～ 999 の範囲で未使用の値が有効で、小さい数字ほど優先順位が高くなります (1 が最高の優先順位)。拡張処理時 |

| | |
|---|---|
| | に、別のシステムですでに使用中の優先順位をこの階層に対して指定することはできません。デフォルト値を使用することを推奨します。 **注意:** このフィールドは、階層をはじめて拡張するときにだけ表示されます。 |
| **Target Priority** | **ターゲットの優先順位** を選択するか、入力してください。これは、新しく拡張する IP 階層が、他のサーバーに存在する同等の階層に対して持つ相対的な優先順位です。1 ～ 999 の範囲で、まだ優先順位として使用されていない値が有効で、リソースのカスケーディングフェイルオーバーシーケンスにおけるサーバーの優先順位を示します。数値が小さいほど優先順位が高くなります (数値 1 が最高の優先順位)。LifeKeeper のデフォルトでは、階層が作成されたサーバーに「1」が割り当てられることに注意してください。優先順位は連続している必要はありませんが、特定のリソースについて 2 つのサーバーに同じ優先順位を割り当てることはできません。 |

3. 情報ボックスが表示され、LifeKeeper による環境のチェックが正常に完了したこと、およびこの IP リソースを拡張するための要件がすべて満たされたことが示されます。満たされていない要件がある場合は、[Next] ボタンは選択できなくなり、[Back] ボタンが有効になります。[Back] をクリックした場合、情報ボックスに表示されるエラーメッセージの内容に従って、リソースの拡張を変更できます。ここで [Cancel] をクリックする場合は、いずれかの時点でこの手順に戻り、IP リソース階層を他のサーバーに拡張して LifeKeeper の保護下に置く必要があります。[Next] をクリックすると、[Extend Resource Hierarchy] 設定作業に入ります。
4. リソース階層構造の拡張を行うために、以下の情報の入力が要求されます。

| フィールド | ヒント |
|---|---|
| **IP Resource** | [Create Resource Wizard] で作成した **IP リソース** またはアドレスと同じです。実 IP アドレスを保護する場合は'0.0.0.0'を入力します。 |
| **Netmask** | これは、このテンプレートサーバーの IP リソースが作成されたときに選択された **ネットマスク** と同じで、ここではターゲットサーバーの IP リソースによって使用されます。<br><br>IP Resource の項目で実 IP アドレスの保護（'0.0.0.0'）を指定した場合、ネットマスクの項目は表示されません。 |
| **Network Interface** | **ネットワークインターフェース** を選択するか、入力してください。これは、IP リソースがターゲットサーバーで使用するネットワークインターフェース (つまりイーサーネットカード) の名前です。 |
| **IP Resource Tag** | **IP リソースタグ** を選択するか、入力してください。これは、ターゲットサーバーに拡張される IP リソースが使用するリソースタグ名です。 |

5. 拡張が実行されていることを示す情報ボックスが表示されます。

   同じ IP リソースインスタンスをクラスター内の別のサーバーに拡張する場合は、 **[Next Server]** をクリックします。その場合は、リソース階層の拡張の操作が繰り返されます。

   **[Finish]** をクリックすると、IP リソースの拡張が正常に完了したことが LifeKeeper によって確認

されます。

6. **[Done]** をクリックして、 **[Extend Resources Hierarchy]** メニューの選択を終了します。

   **注意:** 必ずすべてのサーバーで新しいインスタンスの機能をテストしてください。

# 6.6.3.10. リソース階層の拡張解除

1. LifeKeeper GUI メニューから **[Edit]** を選択し、次に **[Resouce]** を選択します。ドロップダウンメニューから、 **[Unextend Resource Hierarchy]** を選択します。

2. **[Target Server]** で、IP リソースを拡張解除するターゲットサーバを選択します。IP アドレスが現在 In Service のサーバは選択できません。

   **注記:** 右側のペインから個々のリソースインスタンスを右クリックして拡張解除作業を選択した場合、このダイアログボックスは表示されません。

   **[Next]** をクリックして次のダイアログボックスに進んでください。

3. **[IP Hierarchy to Unextend]** を選択してください。

   **注記:** 左側のペインからグローバルリソースを右クリックするか、右側のペインから個々のリソースインスタンスを右クリックして拡張解除作業を選択した場合、このダイアログボックスは表示されません。

   **[Next]** をクリックして次のダイアログボックスに進んでください。

4. 拡張解除のために選択したターゲットサーバと IP リソース階層を確認する情報ボックスが表示されます。

   **[Unextend]** をクリックします。

5. IP リソースが正常に拡張解除されたことを確認する別の情報ボックスが表示されます。

6. **[Done]** をクリックして、 **[Unextend Resource Hierarchy]** メニューを終了します。

# 6.6.3.11. リソース階層のテスト

手動スイッチオーバを開始することによって、IP リソース階層をテストできます。これにより、プライマリサーバからバックアップサーバへのリソースインスタンスのフェイルオーバをシミュレートします。

## GUI からの手動スイッチオーバの実行

手動スイッチオーバを開始するには、LifeKeeper の GUI メニューから **[Edit]** 、 **[Resource]** の順に選択し、最後にドロップダウンメニューから **[In Service]** を選択します。例えば、バックアップサーバで [In Service] 要求を実行すると、アプリケーション階層はバックアップサーバで in service になり、プライマリサーバでは out of service になります。この時点で、元のバックアップサーバがプライマリサーバに、元のプライマリサーバがバックアップサーバに変わります。

**[Out of Service]** 要求を実行すると、アプリケーションは他のサーバで in service にならずに、 out of service になります。

手動スイッチオーバでは、IP Recovery Kit はそのアドレスをアクティブサーバのサービスから削除した後、バックアップサーバに追加します。

スイッチオーバ後、その IP リソースのハードウェア (MAC) アドレスは別のアドレスになります。これは、IP リソースが別の LAN インターフェースに関係付けられるためです。ユーザシステムが再接続できるのは、ユーザシステムの TCP/IP ソフトウェアが、この新しいアドレスマッピングを特定してからです。IP Recovery Kit は、ARP (アドレス解決プロトコル) テーブルをアップデートして新しいマッピングを反映する必要があることを、接続されたすべてのサーバに自動的に通知します。

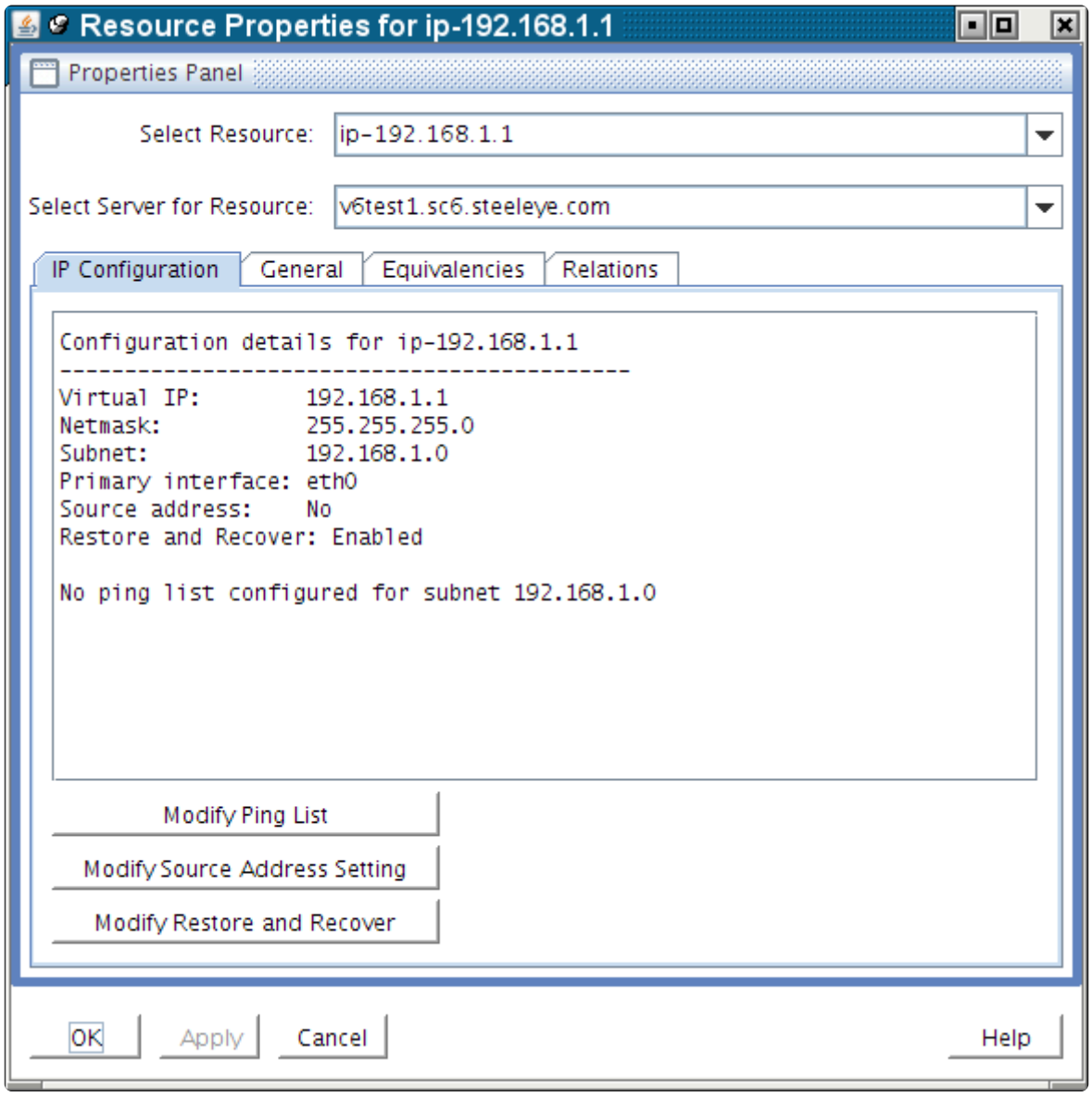
フル機能の TCP/IP 実装を実行しているユーザシステムは、すぐにアップデートされます。それより低機能の実装を備えたユーザシステムでは、アップデートが遅れたり、アドレス指定の仲介役としてルータが必要とされる場合があります。

# 6.6.3.12. IP 構成の確認および編集

**[IP Configuration Properties]** ページでは、特定の IP リソースに関する詳細な設定を確認し、選択された設定項目について変更することができます。

**[IP Configuration Properties]** ページにアクセスするには、LifeKeeper GUI メニューから **[Edit]**、**[Resource]** の順に選択します。ドロップダウンメニューから **[Properties]** を選択します。次にプロパティを表示するリソースとサーバを **[Resource list]** と **[Server list]** から選択します。同様に特定の IP リソースインスタンスを右クリックし、コンテキストメニューからプロパティ画面にアクセスすることも可能です。

以下は IP リソースのプロパティ画面の例です。

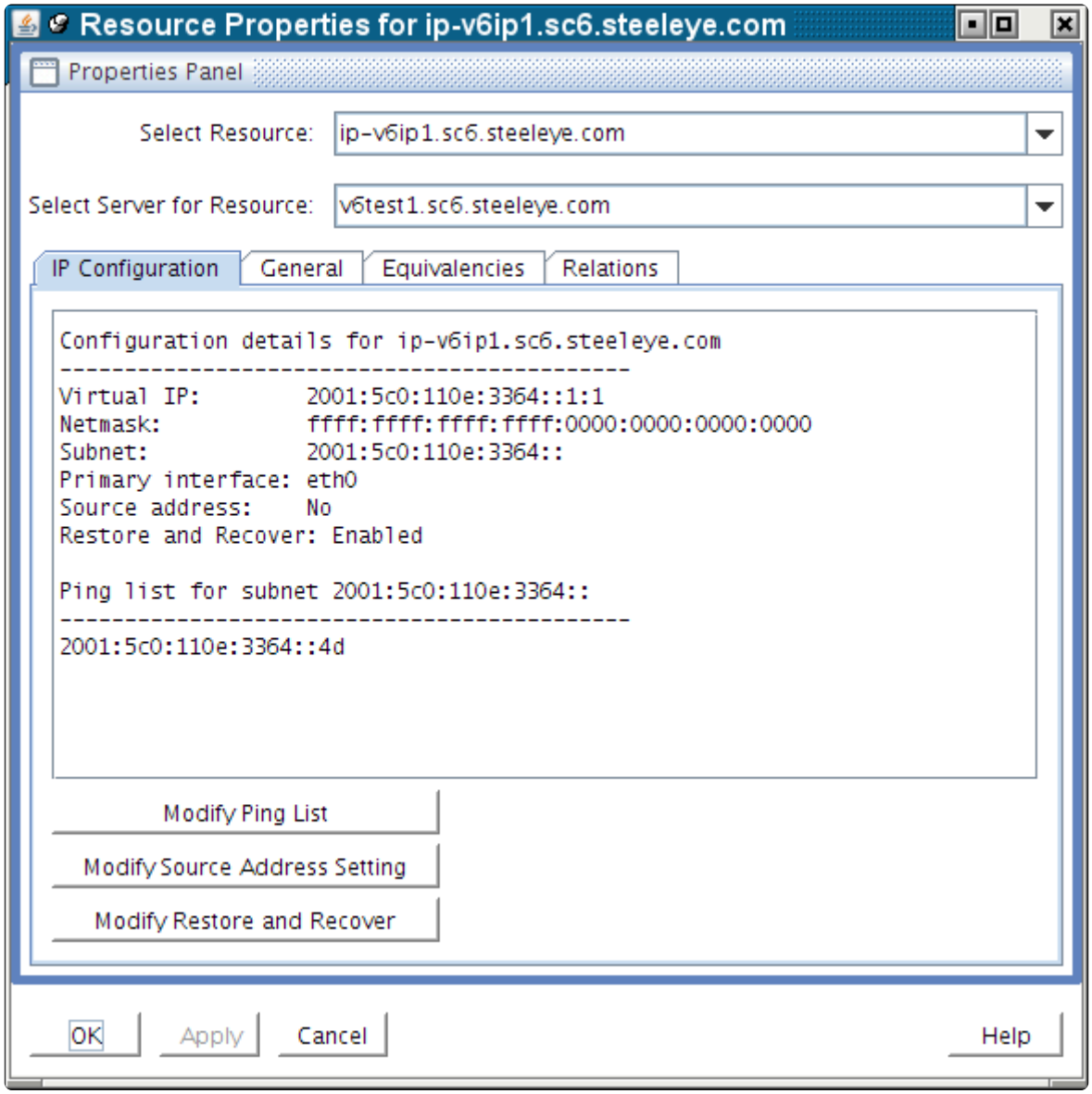



表示されたプロパティ画面にはタブが 4 つあります。1 番目のタブは **[IP Configuration]** で, 特定の IP リソースの設定情報が設定されています。残り 3 つのタブは、LifeKeeper のすべてのリソースタイプで共通の項目として使用できます。

[IP Configuration] タブには特定 IP リソースの以下の情報が表示されています。

| | |
|---|---|
| Virtual IP | IP リソースの仮想 IP アドレス |
| Netmask | 仮想 IP アドレスのネットマスク。この値により、サブネット分割が決定される |
| Subnet | 仮想 IP アドレスの論理サブネット、サブネット部分のビット数を含む |
| Primary interface | アクティブな場合、仮想 IP アドレスを設定する必要のあるネットワークインターフェース |
| Source address setting | 仮想 IP アドレスを同一サブネットへの外部通信 IP トラフィックのソースアドレスに使用するかどうかの設定 |
| Ping List | オプション用 Ping 先リスト。 IP ヘルスチェック時に、通常のブロードキャスト ping メカニズムの代替として ping を通して、この IP リソース (および同一サブネットの他の IP リソース) の動作をチェックする |

上記の図の例では、IP リソース上に Ping List は設定されていません。Ping List が設定されている場合、プロパティ画面は以下の図の例のようになります。

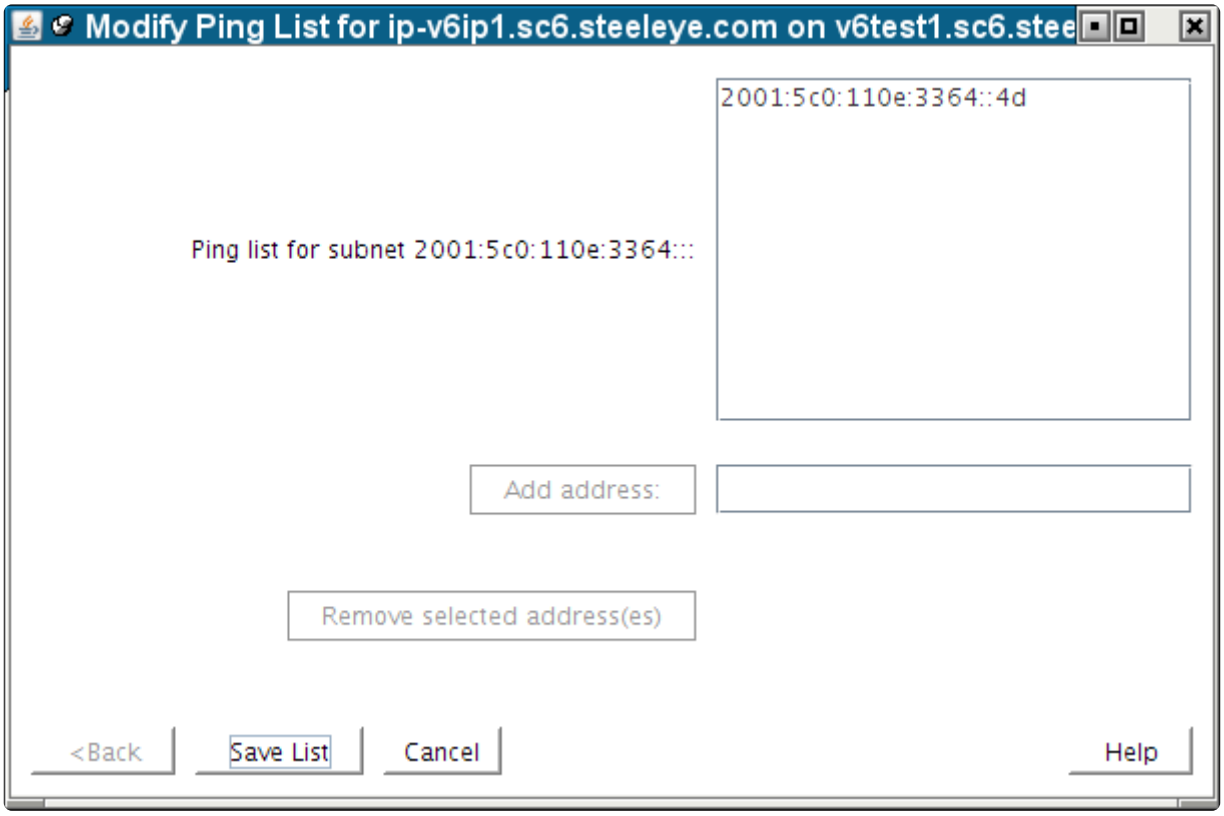



**[Modify Ping List]** および **[Modify Source Address Setting]** ボタンを使用して、この設定項目を変更することが可能です。以下のセクションで説明します。

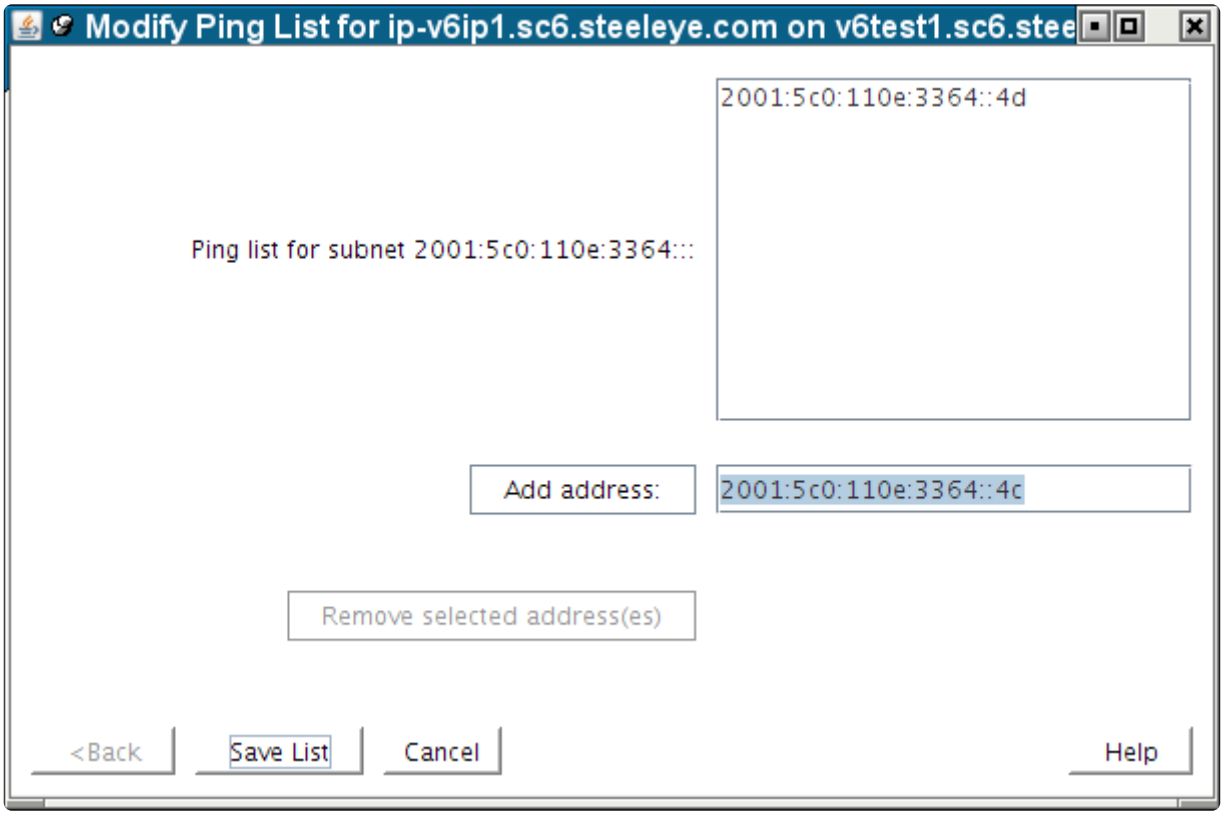
# Ping List の変更

IP リソースに関する Ping List 機能の使い方の説明については、IP リソースの監視 をご参照ください。

IP リソースの Ping List を作成または変更するには、 **[IP Configuration]** プロパティ画面の **[Modify Ping List]** ボタンをクリックします。以下の例のようにダイアログウィンドウが表示されます。

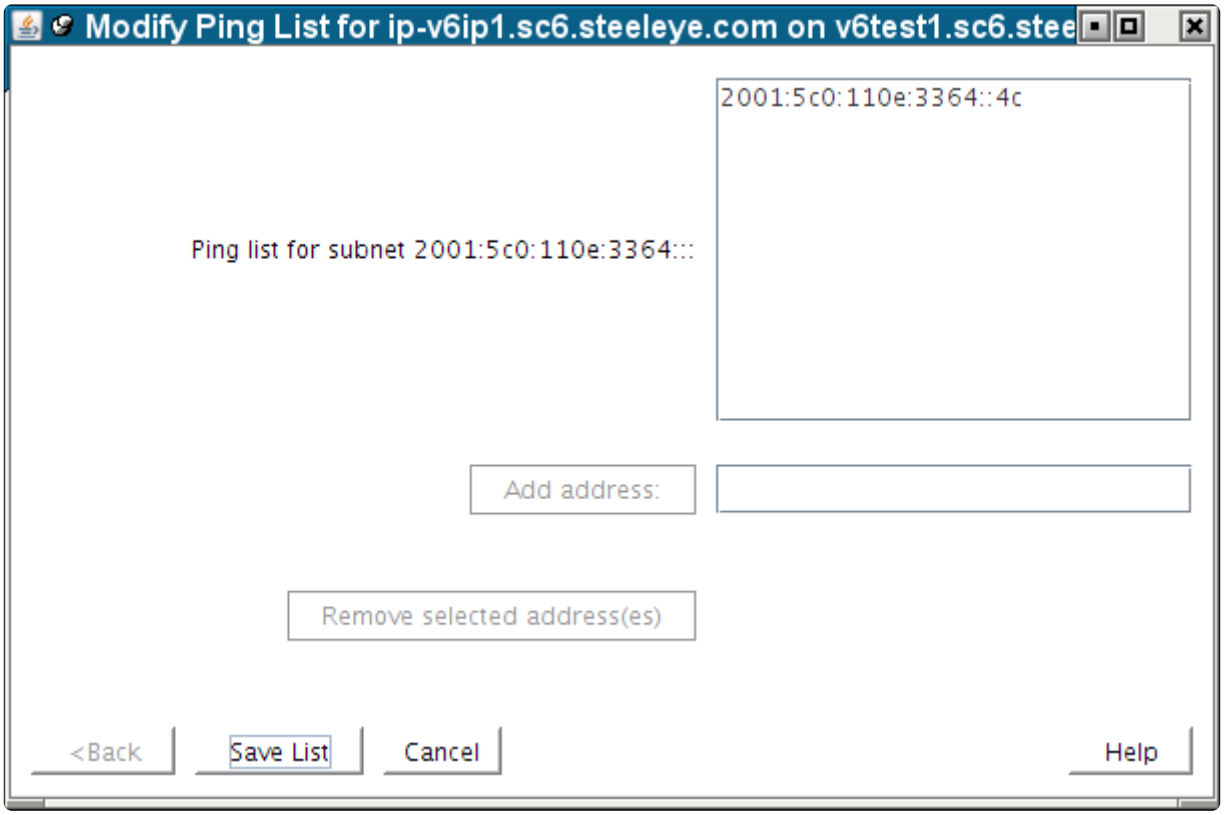



アドレスを Ping List に追加する場合、以下の 2 つの図に示すように [Add address] ボタンの横のフィールドにアドレスを入力し、 **[Add address]** をクリックします。 **[Add address]** ボタンは入力欄にアドレスを入力しない限りグレーアウトされています。



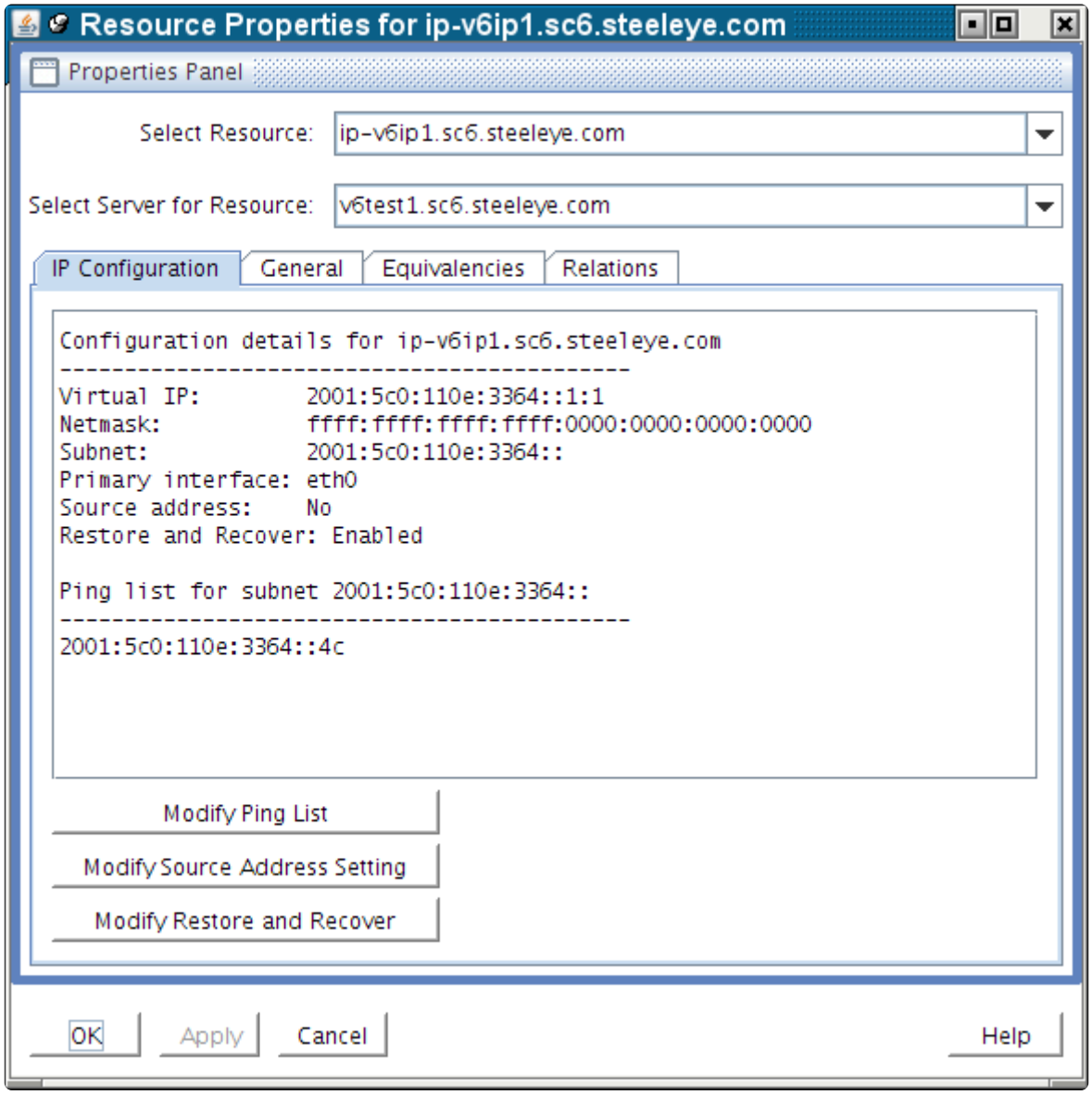

Ping List からアドレスを削除する場合は、削除したいアドレスを選択し、 **[Remove selected address(es)]** ボタンをクリックします。 **[Remove selected address(es)]** ボタンも同様に、リスト内で

アドレスを 1 つ以上選択しない限りグレーアウトされています。




変更済みリストを保存するため、**[Save List]** をクリックします。以下の確認ウィンドウが表示されます。

**[Done]** をクリックしてウィンドウを終了させ、 **[IP Configuration]** プロパティ画面に戻ると、Ping List が変更されているのが確認できます。




## Ping List の使用に関する重要事項

IP リソースの *Ping List* 構成はシステムごとに個別の設定となります。IP リソースが、Ping List 構成後に他のシステムに拡張された場合、同じ設定の Pinglist が拡張先 LifeKeeper システムに一緒にコピーされます。しかし、すでに IP リソースが拡張済みの場合は、Ping List の構成は IP リソースが定義されたそれぞれのシステムに個別に設定する必要があります。Ping List の変更は、IP リソースの状態にかかわらず実行できるため、それぞれのシステムの Ping List を変更するためにIP リソースのスイッチオーバを実行する必要はありません。

同一論理サブネットに複数の IP リソースが作られている場合、これらのリソースは同じ Ping List を共有する必要があります。これは [IP Configuration] プロパティ画面に反映され、Ping List 変更ダイアログボックスは IP リソースのサブネットごとに設定されます。

IP リソースに Ping List が定義されると、リソースのヘルスチェックはデフォルトのブロードキャスト ping メカニズムの代わりに、Ping List メカニズムを使用します。ブロードキャスト ping メカニズムを再度利用するためには、Ping List に定義した全 IP アドレスを削除する必要があります。

LifeKeeper の動作では、入力された Ping List の IP アドレスについて、フォーマットが正しいかどうか以

外の検査は実行しません。そのため、入力した IP アドレスが実際にネットワーク上に存在しており、常に正常に動作しているかについて、LifeKeeper がインストールされているシステムから ping を実行するなどの方法で確認する必要があります。また LifeKeeper がインストールされているシステムの IP アドレスは常に動作していることが予想されますが、これらの IP アドレスを Ping List に選択しないでください。これらのアドレスへの ping 実行は、NIC に関連付けられている実際の IP リソースの状態にかかわりなく、常に成功してしまう可能性があり、正しいチェックとなりません。

上記にて指摘している通り、IP リソースの Ping List を定義すると、LifeKeeper は自動的に Ping List メカニズムをブロードキャスト ping メカニズムの代わりに、対象の IP リソースおよび同一サブネットのアドレス設定をされた他の IP リソースに対して、使用します。この時、ブロードキャスト ping メカニズムを使用不可にするために、IP Recovery Kit のチューニング で解説されている NOBCASTPING 値の設定を行う必要はありません。しかしながら、ネットワーク上にブロードキャスト ping に対して回答可能なシステムが存在しない環境で設定する場合であれば、NOBCASTPING=1 と設定する必要があります。この設定を行わないと、IP リソース作成時にブロードキャスト ping によるチェックが実行されてしまうためです。この段階ではまだ Ping List を作成することができません。IP リソース作成が済み、Ping List を設定後であれば、デフォルトの NOBCASTPING=0 に戻して構いません。

リソース削除時に、Ping List の内容は削除されません。そのため、リソース削除後に以前と同一のサブネットアドレスを持つ IP リソースを作成した場合、以前設定した Ping List の内容が残ってしまうことに注意してください。

## ソースアドレス設定の変更

IP リソースのソースアドレス設定は、IP リソース起動時に、その IP リソースに関連付けられたサブネットへの送信トラフィックの送信元アドレスとして仮想 IP アドレスを使用するかどうか決定します。この値のデフォルトは [No] と設定されており、サブネットへの送信トラフィックの送信元アドレスは NIC に設定されたプライマリ IP アドレスが使用されます。仮想 IP アドレスは通常、クライアントの着信接続ポイント、すなわち、仮想 IP アドレスが使用される接続は、着信トラフィックとして開始されますので、この動作は大半のケースにおいて影響ありません（応答パケットでは、この設定に関係なく、着信トラフィックの送信先アドレスが応答パケットの送信元アドレスとして使用されます）。

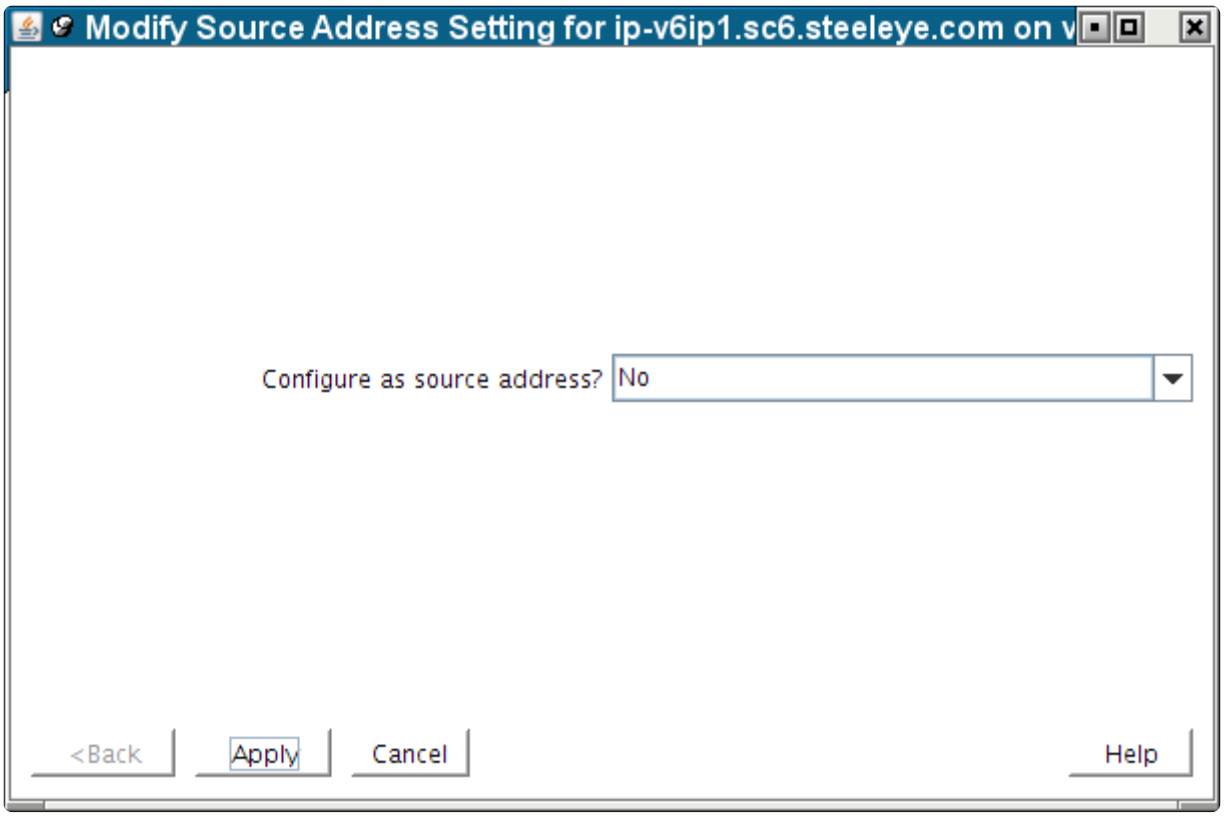
しかし特定の状況または設定のために、Lifekeeper システムからの接続に仮想 IP アドレスをソースとする必要がある場合もあり得ます。このために、ソースアドレスを変更するには、IP リソースが起動している状態で [Source Address Setting] を [Yes] にします。システムの TCP/IP ルートはこの設定を行うことで変更されます。

仮想 IP アドレスがシステムの実 IP アドレスと異なる論理サブネット上に存在する、例えば実 IP アドレスが 10.0.1.10/24 で仮想 IP アドレスが 10.1.0.10/24 に存在する場合、ソースアドレス設定に影響されず 10.0.1.10/24 への全ての送信トラフィックは実 IP アドレスから、10.1.0.10/24 への全ての送信トラフィックは、仮想 IP アドレスから行われることになります。この設定は AWS には適用されません。

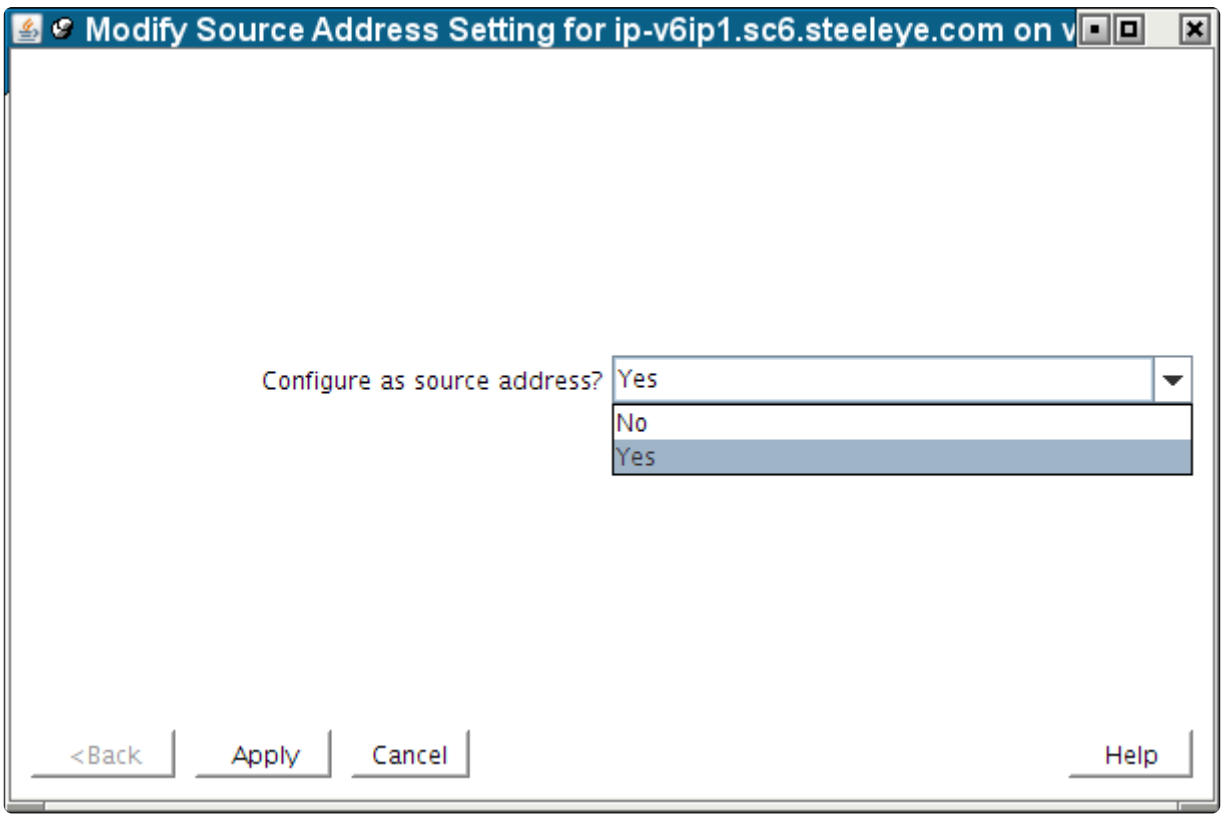
同様に、CIDR/サブネットの範囲外の仮想IPで構成される EC2 ルートテーブル構成では、そのIPアドレスがサブネット内で構成されている唯一のIPアドレスであるため（/32 のホストルート）、送信トラフィックのソースアドレスは、ソースアドレス設定に影響されず全て実 IP アドレスから行われることになります。

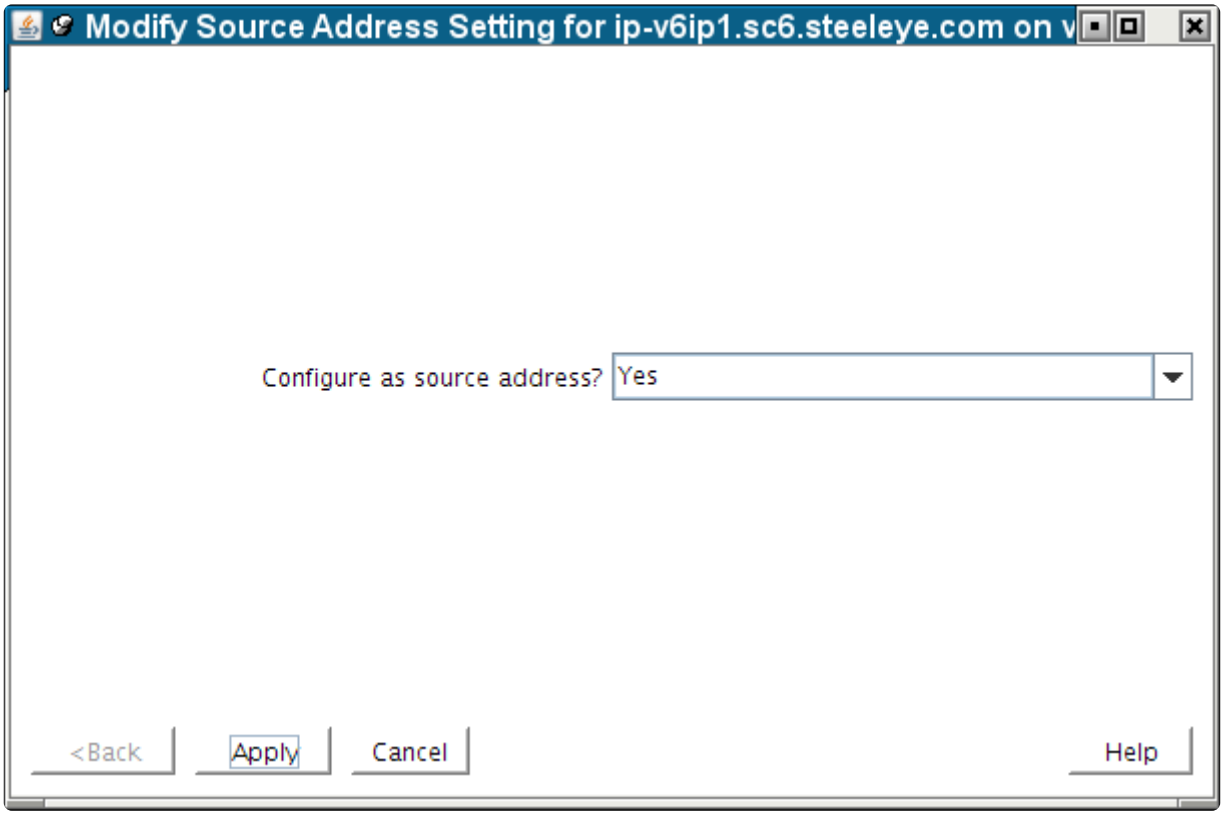
IP リソースのソースアドレス設定を変更するには、[IP Configuration] プロパティ画面の [Modify Source

Address Setting] ボタンをクリックします。以下の例のようにダイアログウィンドウが表示されます。

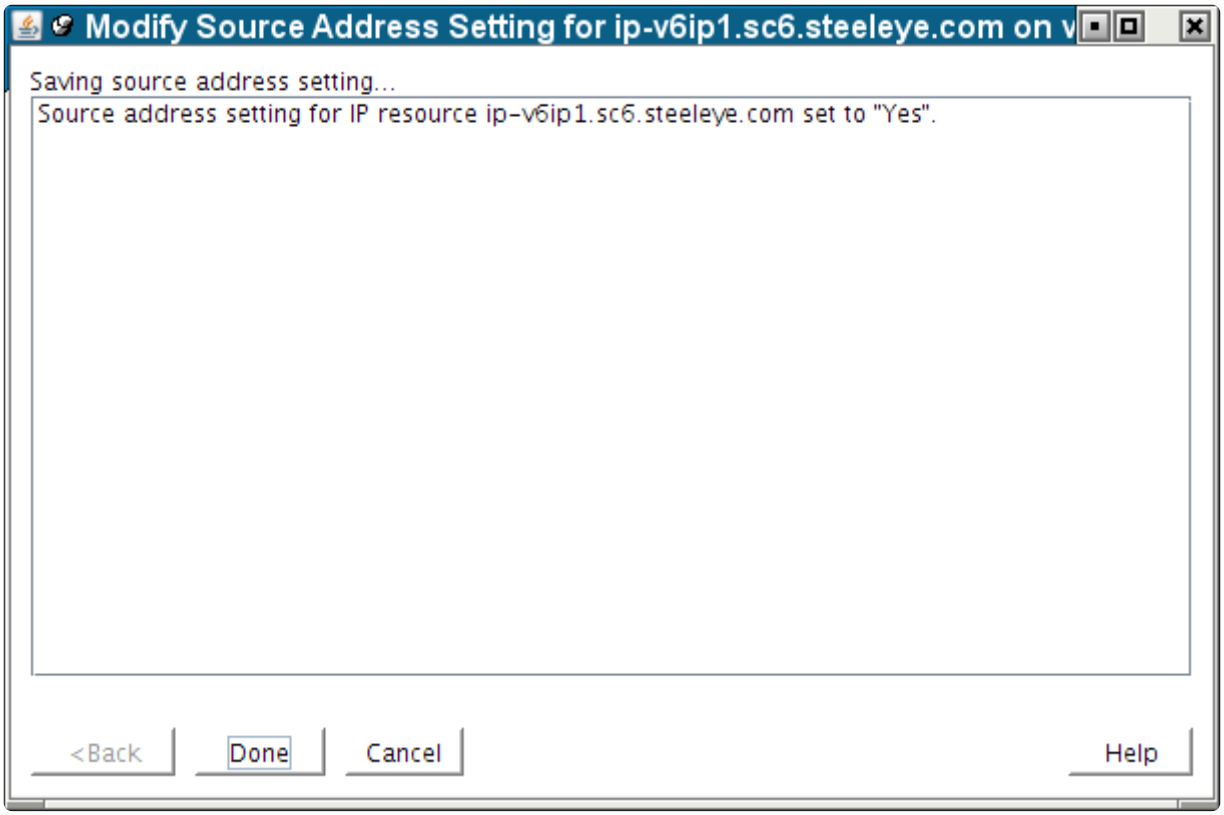



設定を変更するにはドロップダウンリストを選択し、 **[Yes]** または **[No]** を選択してください。

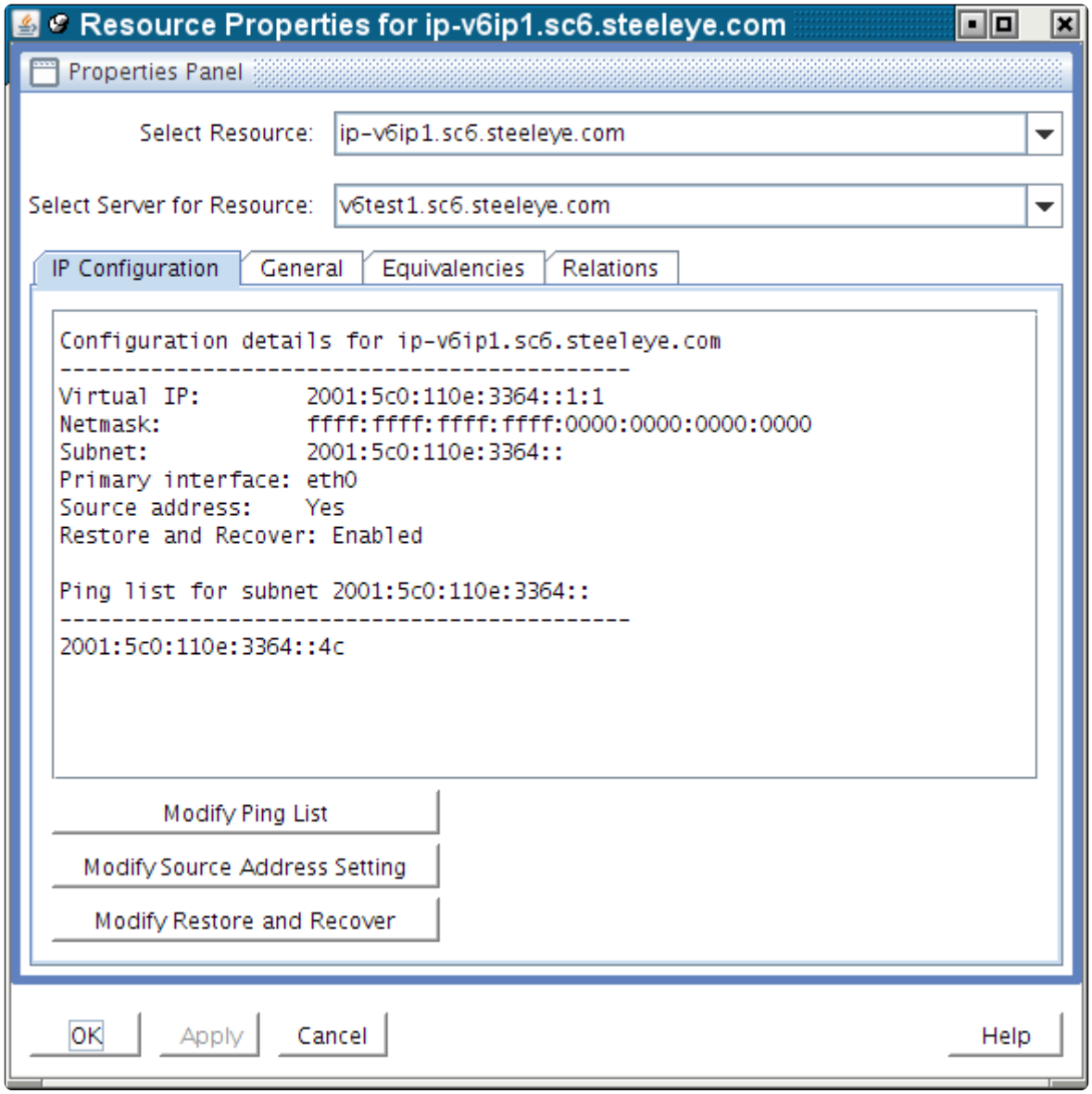





**[Apply]** をクリックし、変更を保存します。以下の確認ウィンドウが表示されます。




**[Done]** をクリックするとウィンドウが閉じ、 **[IP Configuration]** プロパティ画面に戻ります。ここで、変更した設定を確認できます。




## ソースアドレス設定に関する重要事項

IP リソースの **[Source Address]** 設定は、システムごとに個別の設定となります。 **[Source Address]** 設定の変更後に IP リソースを他のシステムに拡張する場合、その設定が拡張先の LifeKeeper システムに一緒にコピーされます。しかし、すでに IP リソースが拡張済みの場合は、IP リソースが定義されている各システム上で個別に **[Source Address]** 設定を変更する必要があります。

特定のサブネット内で、[Source Address] 設定を [Yes] にした場合、単一の IP リソースのみ設定が有効になります。外向けネットワークトラフィックのソースアドレスとして動作する IP アドレスは本来 1 つだけに限定されるためです。同じサブネットに属する複数の IP リソースが存在し、[Source Address] 設定が [Yes] になっている場合、最後に設定したリソースの IP アドレスが有効な送信トラフィックのソースアドレスとなり、その他のソースアドレス設定は無効になります。

**[Source Address]** 設定は、IP リソースの起動時のローカル TCP/IP 設定のみに影響します。したがってすでにリソースが起動中の場合は、ソースアドレス設定を変更後、リソースを一旦停止し、再起動させて、変更を TCP/IP 設定に反映させる必要があります。

**[Source Address]** 設定は、IP v4 アドレスにのみ影響があります。IP v6 アドレスには影響ありません。

# リストアおよびリカバリの変更

IP リソースに対する **[Restore and Recover]** の設定は、異なるサブネット間にディザスタリカバリシステムが存在するマルチサイトクラスタ環境では [Disable] にする必要があります。

この設定では既存の IP リソースに対するデフォルトのリストアおよびリカバリの動作を **[Enable]** もしくは **[Disable]** で選択することができます。 **[Enable]** オプションに設定すると、IP アドレスはアクティブになり、通常の監視およびリカバリプロセスが発生します。IP アドレスのリストアの現在のデフォルトの動作は **[Enable]** になっています。

**[Restore and Recover]** オプションを **[Disable]** に設定すると、リソースはネットワークもしくはネットワークアダプタ上で in service の状態になりますが、IP アドレスはアクティブになりません。この設定によりマルチサイトクラスタ (もしくは WAN) 環境での階層をディザスタリカバリシステム上で in service にすることが可能になります。

この設定はリソースを作成し拡張した後に選択することが可能です。

**アクティブな IP アドレス (ISP) に対する重要事項:** ISP およびアクティブな IP アドレスにおいて動作を **[Disable]** に設定しても、アクティブな IP アドレスは out of service になりません。

# 6.6.3.13. IP Recovery Kit のチューニング

IP Recovery Kit の動作のチューニング値をこちら で説明しています。これらの値は */etc/default/LifeKeeper* 設定ファイルを編集することにより設定可能です。IP Recovery Kit のコンポーネントはメモリに常駐していないため、 */etc/default/LifeKeeper* の値を変更すると、即時に反映されます。LifeKeeper の再起動を行う必要はありません。

# 6.6.4. IP Recovery Kit (IPv4) 処理概要

この処理概要は IPv4 アドレスを保護する場合を例に記載しています。

## 起動処理 (restore)

1. 保護対象の IP アドレスの確認
   保護している IP アドレスが物理 IP アドレスかどうかを確認します。
   物理 IP アドレスを保護している場合は、この時点で起動処理は成功で終了します。

   なお、[Restore and Recover] の設定（デフォルト設定 : [Enable]）が [Disable] の場合は、以降の処理は実施せず、起動処理は成功で終了します。

2. リンク状態の確認
   IP_NOLINKCHECK=1 が設定されている場合は、本項目の処理を実施せず、次の項目3. の処理へ移行します。

   以下のコマンドを実行し、出力の中に “NO-CARRIER” ステータスがあるかどうかにより、リンクダウン状態またはリンクアップ状態を判断します。

   ```
   # ip -o link show <デバイス>
   ```

   上記コマンドの結果においてデバイスがリンクアップ状態の場合は、次の項目3. の処理へ移行します。
   なお、デバイスがリンクダウン状態の場合は、一度だけデバイスの停止･起動を実施します。最終的にリンクアップ状態にならない場合は起動処理はエラーで終了します。

   RHEL8 系以降の環境の場合

   ```
   # nmcli connection down <デバイス>
   5秒待機（可変、IP_WAIT_LINKDOWN にて秒単位で変更可能）
   # nmcli connection up <デバイス>
   1秒待機（固定、変更不可）
   # ip -o link show <デバイス>
   ```

   SLES および RHEL7 系の環境の場合

   ```
   # ifdown <デバイス>
   5秒待機（可変、IP_WAIT_LINKDOWN にて秒単位で変更可能）
   # nmcli connection up <デバイス>
   1秒待機（固定、変更不可）
   # ip -o link show <デバイス>
   ```

また、NetworkManager が有効な環境では、ip コマンドによるリンク状態の確認に加え、以下のコマンドを実行し、NetoworkManager の起動状態を確認します。確認に失敗した場合は起動処理はエラーで終了します。

```
#nmcli connection show --active | grep -qE <デバイス>
```

3. 割り当て状態の事前確認
   以下のコマンドを実行し、デバイスに割り当てられている IP アドレスを取得し、出力の中に仮想 IP アドレスが存在しないことを確認します。

```
# ip -4 -o addr show dev <デバイス>
```

   仮想 IP アドレスが存在しない場合は、次の項目4. の処理へ移行します。
   仮想 IP アドレスが存在する場合は、監視処理と同一の項目を確認し、正常性を判断します。正常な場合は起動処理を成功で、そうでない場合は起動処理をエラーで終了します。

4. 重複確認
   NOIPUNIQUE=1 が設定されている場合は、本項目は実施せずに次の項目5. の処理へ移行します。

   以下のコマンドを実行し、ローカルノード上の実 IP アドレスから割り当て予定の仮想 IP アドレスに対して疎通確認をします。応答がある場合は、仮想 IP アドレスが重複していると判断し、起動処理はエラーで終了します。

```
# /opt/LifeKeeper/bin/lkping -b -c 1 -w IP_PINGTIMEOUT <仮想 IP>
```

5. 仮想 IP アドレスの割り当ておよび ARP テーブルのフラッシュ
   まず、以下のコマンドを実行し、仮想 IP アドレスの割り当てを実施します。コマンドの結果により割り当てに失敗したと判断された場合は、起動処理はエラーで終了します。

```
# ip -4 addr add <仮想 IP/プレフィックス> dev <デバイス>
```

   また、以下のコマンドを実行し、自ノード上の全てのネットワークデバイスに対して ARP キャッシュをフラッシュします。

```
# ip neigh flush dev <デバイス>
```

   つぎに、以下のコマンドを実行し、仮想 IP アドレスを割り当てたネットワーク上の ARP テーブルをフラッシュします。

```
# arping -Uq -c1 -s <仮想 IP> -I <デバイス> <仮想 IP>
```

```
# arping -Aq -c1 -s <仮想 IP> -I <デバイス> <仮想 IP>
# arping -Uq -c1 -s <仮想 IP> -I <デバイス> <ブロードキャストアドレス>
# arping -Aq -c1 -s <仮想 IP> -I <デバイス> <ブロードキャストアドレス>
```

なお、arping コマンドがインストールされていない場合に限り、仮想 IP アドレスを割り当てたネットワークデバイスに対してのみ、以下のコマンドを実行します。

```
# ip neigh flush dev <デバイス>
```

6. ソースアドレス設定の有効化
   IP リソースのプロパティ画面から Source Address Setting を有効にしている場合は、以下のコマンドを実行し、自ノードから仮想 IP アドレスが属するネットワークに対する通信の送信元アドレスを仮想 IP アドレスに設定します。

```
# ip -4 route change <ネットワークアドレス/プレフィックス> src <仮想 IP>
```

7. 疎通確認
   疎通確認処理は監視処理時と共通の処理を実施しています。別途記載している「疎通確認処理」の項目をご参照ください。この疎通確認の成功または失敗により、起動処理は成功またはエラーで終了します。

## 停止処理 (remove)

1. 保護対象の IP アドレスの確認
   保護している IP アドレスが物理 IP アドレスかどうかを確認します。
   物理 IP アドレスを保護している場合は、この時点で停止処理は成功で終了します。

2. 割り当て状態の確認
   以下のコマンドを実行し、デバイスに割り当てられている IP アドレスを取得し、出力の中に仮想 IP アドレスが存在することを確認します。

```
# ip -4 -o addr show dev <デバイス>
```

   仮想 IP アドレスが存在しない場合は、停止処理は成功で終了します。
   仮想 IP アドレスが存在する場合は、次の項目3. の処理へ移行します。

3. ソースアドレス設定の無効化
   IP リソースのプロパティ画面から Source Address Setting を有効にしている場合は、以下のコマンドを実行し、自ノードから仮想 IP アドレスが属するネットワークに対する通信の送信元アドレスを実 IP アドレスに設定します。

```
# ip -4 route change <ネットワークアドレス/プレフィックス>
```

4. 仮想 IP アドレスの取り外し
   以下のコマンドを実行し、仮想 IP アドレスの取り外しを実施します。コマンドの結果により停止処理は成功またはエラーで終了しますが、必ず次の項目5. 以降の処理を実施した上で終了します。

```
# ip -4 addr delete <仮想 IP/プレフィックス> dev <デバイス>
```

5. ホストルートの削除
   以下のコマンドを実行し、ホストルートを削除します。

```
ip -4 route delete <仮想 IP/プレフィックス> dev <デバイス>
```

6. ルーティングキャッシュのクリア
   以下のコマンドを実行し、ルーティングキャッシュをクリアします。

```
# echo 0 > /proc/sys/net/ipv4/route/flush 2> /dev/null
```

## 監視処理 (quickCheck)

[Restore and Recover] の設定（デフォルト設定 : [Enable]）が [Disable] の場合は、以降の処理は実施せず、監視処理は成功で終了します。

1. 割り当て状態の確認
   以下のコマンドを実行し、デバイスに割り当てられている IP アドレスを取得し、出力の中に保護している IP アドレスが存在することを確認します。

```
# ip -4 -o addr show dev <デバイス>
```

   保護している IP アドレスが存在しない場合は、この時点で監視処理はエラーで終了します。
   保護している IP アドレスが存在する場合は、次の項目2. の処理へ移行します。

2. リンク状態の確認
   本項目において IP_NOLINKCHECK=1 が設定されている場合は本項目の処理を実施せず、次の項目3. の処理へ移行します。

   以下のコマンドを実行し、出力の中に “NO-CARRIER”ステータスがあるかどうかにより、リンクダウン状態またはリンクアップ状態を判断します。

```
# ip -4 -o addr show dev <デバイス>
```

   上記コマンドの結果においてデバイスがリンクダウン状態の場合は、この時点で監視処理はエラーで終了します。

また、NetworkManagerが有効な環境では、ip コマンドによるリンク状態の確認に加え、以下のコマンドを実行し、NetworkManager の起動状態を確認します。確認に失敗した場合はこの時点で監視処理はエラーで終了します。

```
# nmcli connection --active grep -qE <デバイス>
```

3. 疎通確認
   疎通確認処理は起動処理と共通の処理を実施しています。別途記載している「疎通確認処理」の項目をご参照ください。この疎通確認の成功または失敗により、監視処理は成功またはエラーで終了します。

監視処理が IP_QUICKCHECK_TIMEOUT の値（デフォルト値：12秒）で終わらない場合はタイムアウトが発生し、リソース障害として判定されます。

# 回復処理 (recover)

1. 保護対象の IP アドレスの確認
   保護している IP アドレスが物理 IP アドレスかどうかを確認します。
   物理 IP アドレスを保護している場合は、この時点で回復処理はエラーで終了します。

2. 回復処理の実施
   上述の停止処理と起動処理を順に実行し、仮想 IP アドレスの回復を試みます。

# 疎通確認処理

この疎通確認処理は起動処理と監視処理で共通の処理となっており、以下の条件により大きく 3種いずれかの挙動になります。

- PingList の有無
- NOBCASTPING の設定値 (0 または 1)

本項における処理の中で「成功」または「エラー」で終了したら、起動処理または監視処理の元の項目へ戻ってください。

## ユニキャスト ping およびブロードキャスト ping による疎通確認

実施条件：PingList 有りかつ、NOBCASTPING=0 または 1

1. ユニキャスト ping による疎通確認
   以下のコマンドを実行し、PingList に記載されている各 IP に対してユニキャスト ping を実施します。

```
# /opt/LifeKeeper/bin/lkping -b -c 1 -w IP_PINGTIME -I <仮想 IP> <PingLi
st IP>
```

上記コマンドにおいて 1つでも応答がある場合は、本処理は成功で終了します。
上記コマンドにおいて 1つも応答がない場合は、次の項目2. の処理へ移行します。

2. ブロードキャスト ping による疎通確認
本項目において NOBCASTPING=1 が設定されている場合は、前の項目1. の処理に失敗していることを理由として、この時点において本処理はエラーで終了します。

本項目において NOBCASTPING=0 が設定されている場合は、以下のコマンドを実行し、仮想 IP アドレスが属するネットワークに対してブロードキャスト ping を実施します。

```
# /opt/LifeKeeper/bin/lkping <ブロードキャストアドレス> -b -c 1 -w IP_PINGTIME -I <仮想 IP> -z <自ノード IP>
```

上記コマンドにおいて 1つでも応答がある場合は、本処理は成功で終了します。
上記コマンドにおいて 1つも応答がない場合は、本処理はエラーで終了します。

### 疎通確認を実施せずに成功終了

実施条件 : PingList 無しかつ、NOBCASTPING=1
本条件においては疎通確認は実施せず、本処理は成功で終了します。

### ブロードキャスト ping のみによる疎通確認

実施条件 : PingList 無しかつ、NOBCASTPING=0
以下のコマンドを実行し、仮想 IP アドレスが属するネットワークに対してブロードキャスト ping を実施します。

```
# /opt/LifeKeeper/bin/lkping -b -c 1 -w IP_PINGTIME -I <仮想 IP> -z <自ノード IP>
```

上記コマンドにおいて 1つでも応答がある場合は、本処理は成功で終了します。
上記コマンドにおいて 1つも応答がない場合は、本処理はエラーで終了します。

## lkping コマンドについて

### コマンドの概要

lkping コマンドは、Lifekeeper for Linux 用にカスタマイズされた疎通確認コマンドであり、LifeKeeper の一部のリソースに対する処理に利用されます。

### ping コマンドとの差異

一般的に疎通確認に利用される ping コマンドと lkping コマンドでは、一部のオプションが異なりますが、疎通確認の内部動作は同じです。
lkping コマンドの結果がエラーとなるような状況では、ping コマンドでも同様にエラーとなることが想定

されます。

### lkping コマンド特有のオプション

以下のオプションについては lkping 特有です。その他のオプションは通常の ping コマンドと同じです。

-w：コマンドがタイムアウトするまでの秒数を指定。（ミリ秒ではない）
-z：ping 応答の受信を除外するアドレスを “,(カンマ)” 区切りで指定。

※-z オプションはブロードキャスト ping に対する自ノード上の IPアドレスからの応答を除外するために使用します。

## IP Recovery Kit のパラメーターについて

処理概要の中で使用されている IP Recovery Kit のパラメーターについては、IPパラメータ一覧 をご参照ください。

# 6.7. MySQL Recovery Kit 管理ガイド

## MySQL Recovery Kit のドキュメンテーション

SIOS Protection Suite for Linux MySQL Recovery Kit を使用すると、MySQL のリソースとデータベースのための LifeKeeper の障害回復保護機能を簡単に追加できます。これにより、プライマリデータベースサーバで障害が発生しても、わずかな時間で人手を介さずに指定されたバックアップサーバにて復旧することができます。

### SIOS Protection Suite/LifeKeeper のドキュメンテーション

以下の SPS 製品ドキュメンテーションを SIOS Technology Corp から取得可能です。

- ◦ SPS for Linux リリースノート
- ◦ SPS for Linux テクニカルドキュメンテーション (LifeKeeper GUI の **[Help]** メニューからもアクセス可能)

本ドキュメンテーションおよびオプションのリカバリキットに関連するドキュメンテーションは SIOS テクニカルドキュメンテーション の Web サイトで入手可能です。

# 6.7.1. 要件

## キットのハードウェア / ソフトウェア要件

リカバリソフトウェアをインストールおよびセットアップするには、サーバが一定のハードウェア / ソフトウェア要件を満たしている必要があります。LifeKeeper MySQL Recovery Kit をインストールまたは削除する具体的な方法については、SPS for Linux インストールガイド を参照してください。

設定が以下の要件を満たしていることを確認してください。

- **サーバ** 。この Recovery Kit には、SPS for Linux テクニカルドキュメンテーション および SPS for Linux リリースノート で説明されている要件に従って設定した LifeKeeper 対応の 2 台以上のコンピュータが必要です。
- **LifeKeeper ソフトウェア** 。各サーバに同じバージョンの LifeKeeper ソフトウェアとパッチをインストールする必要があります。具体的な LifeKeeper の要件については、SPS for Linux テクニカルドキュメンテーション および SPS for Linux リリースノート を参照してください。
- **LifeKeeper IP Recovery Kit** 。リモートクライアントが MySQL データベースにアクセスする場合に、このキットが必要になります。同じバージョンの Recovery Kit を各サーバにインストールする必要があります。
- **IP ネットワークインターフェース** 。各サーバは、イーサーネット TCP/IP をサポートするネットワークインターフェースを 1 つ以上必要とします。IP スイッチオーバが正しく動作するには、ローカルネットワークに接続されているユーザシステムが標準の TCP/IP 仕様に準拠している必要があります。

> 注記: 各サーバが必要とするネットワークインターフェースが 1 つだけであっても、heterogeneous な要件、スループット要件、単一障害点の排除、ネットワークのセグメンテーションといった多くの理由のために、複数のインターフェースを使用すべきです。

- **TCP/IP ソフトウェア** 。各サーバは TCP/IP ソフトウェアも必要とします。
- **MySQL ソフトウェア** 。各サーバは、LifeKeeper と LifeKeeper MySQL Recovery Kit の設定に先だって MySQL ソフトウェアをインストールして設定しておく必要があります。各サーバには同じバージョンをインストールする必要があります。最新のリリースの互換性および購入方法については、SPS for Linux リリースノート を参照するか、営業担当者にお問い合わせください。

# 6.7.2. MySQL Recovery Kit の構成

本セクションでは、典型的な LifeKeeper MySQL 設定の定義および例と、MySQL の設定を開始する前に検討すべき情報について説明します。

LifeKeeper Core リソース階層の設定方法については、LifeKeeper for Linux テクニカルドキュメンテーション のリソース階層 セクションを参照してください。

---

MySQL の設定上の考慮事項

クライアント設定の考慮事項

設定の要件

設定例

アクティブ / スタンバイ設定

アクティブ / アクティブ設定

複数データベースサーバ環境に関する考慮事項

LifeKeeper で mysqld グループを使用する

Network Attached Storage の使用

systemd 環境で使用する場合の考慮事項

# 6.7.2.1. MySQL の設定上の考慮事項

LifeKeeper MySQL 環境に関して、次のような特有の考慮事項について検討する必要があります。

プライマリサーバーとバックアップサーバーで MySQL データベースサービスを運用するには、ファイルシステムとディスクパーティションが各サーバーからアクセスできなければなりません。MySQL Recovery Kit の設定を開始する前に、必ず次の準備手順を終了し、各サーバー上のデータベースをテストして稼働させる必要があります。以下の手順では、ユーザー「mysql」は MySQL サーバーを開始するオペレーティングシステムユーザーを表します。

1. すべてのサーバーに MySQL サーバーとクライアントコンポーネントをインストールします。必ず、すべてのサーバーで同じバーションの MySQL クライアントコンポーネントとサーバーコンポーネントが稼働するようにします。MySQL 実行可能ファイルは、ローカルまたは共有ドライブ上に置くことができます。

2. LifeKeeper が保護する MySQL データベースサーバーを稼働させたいソケットおよび / またはポート上で mysqld が稼働しているサーバーがあれば、`mysqladmin` コマンドを使用して各 MySQL サーバーを停止します。

3. MySQL データディレクトリの内容を共有の場所に移動します。デフォルトで、MySQL データディレクトリはローカルドライブにインストールされます。この場所は、ディストリビューションのメカニズムによって異なります。バイナリ RPM は、データディレクトリを `/var/lib/mysql` にインストールします(必ず内容のみを移動して、ディレクトリはそのまま残してください。これにより、MySQL データベースサーバーが必要に応じてこのディレクトリ内にログを書き込むことができます。ステップ 4 で説明されている「mysql」ユーザーが、この場所にログを書き込む権限を持っていることを確認します)。

4. インストール処理で Linux ユーザー「mysql」が作成されなかった場合は、このユーザーを作成します。セキュリティ上の理由から、MySQL サーバーを「root」として実行しないでください(セキュリティ問題の詳細については、『 *MySQL Administration Guide* 』を参照してください)。必ず、「mysql」ユーザーのみがデータベースディレクトリへの読み取り/書き込み権限を持つようにします。「mysql」ユーザーおよびグループはすべてのサーバー上に作成する必要があります。ユーザー ID とグループ ID は、すべてのサーバーで同じでなければなりません。

5. **重要:** OS 自動起動によって開始されるサーバーは、LifeKeeper の保護下には入れません。さらに、このサーバーは LifeKeeper の保護下にあるサーバーと同じポート番号またはソケットを使用できません。

6. ソケットを共有ディスク上のデータディレクトリに書き込むことを推奨します。ソケットをローカルディスクに書き込む場合、階層が存在するすべての LifeKeeper サーバー上の同じパスに存在することを確認します。「mysql」ユーザーが、この場所にソケットを書き込む権限を持っていることを確認します。

7. 設定に適した mysql デーモンスタートアップコマンドを使用して MySQL サーバーを開始します。*my.cnf* ファイルで単一インスタンスを定義している設定の場合は、以下のコマンドを使用してください。

```
<start command> --user=mysql --socket=<socket> --port=<port number> --dat
```

```
adir=<path to the data directory> --log &
```

<start command> は、mysql バージョン 3.x では `safe_mysqld`、バージョン 4.x では `mysqld_safe` です。

*my.cnf* ファイルで mysqld グループを使用した設定の場合は、以下のコマンドを使用してください。
`mysqld_multi start <group number>`

<group number> は mysqld グループの *my.cnf* ファイルで定義したインスタンス番号を表しています。LifeKeeper で mysqld グループを使用する際の詳細については、LifeKeeper で mysqldGroup を使用するを参照してください。

Systemd を採用しているディストリビューションで MySQL（5.7.6 以降）が Systemd を使用するように設定されている場合、systemctl コマンドを利用する必要があります。詳しくは Systemd 環境で使用する場合の考慮事項 を参照してください。

8. 「mysql」という名前で MySQL データベースユーザーを作成します。このユーザーにパスワードを指定し、「shutdown」権限を付与します。これは、1 台のサーバー上でのみ行います(ユーザーの作成と権限付与の詳細については、『 *MySQL Administration Guide* 』を参照してください)。

9. サンプル設定ファイル *my.cnf* を必要な場所( */etc* もしくは */<datadir>* )にコピーします。このファイルには、データベースサーバー用とクライアントプログラム用のオプションが含まれています。

   ファイルは、 *MySQL* データ ディレクトリまたは /etc ディレクトリのどちらかに置くことが可能です。 */etc/my.cnf* ファイルにはグローバルオプションが含まれます。マシン上で常に 1 つのデータベースのみが稼働する場合(つまりアクティブ / スタンバイ設定)、もしくは mysqld グループ機能を使用している場合は */etc* に *my.cnf* ファイルを置きます。(LifeKeeper で mysqld グループを使用するを参照してください。)ファイルが */etc* にある場合、そのファイルを各 LifeKeeper バックアップサーバーにコピーする必要があります。データディレクトリ内の *my.cnf* ファイルには、サーバー固有のオプションが含まれている必要があります。複数サーバーおよびアクティブ / アクティブ設定の場合、mysqld グループ機能を使用しない限りはこのファイルをリソースインスタンスごとにデータディレクトリに格納する必要があります。(LifeKeeper で mysqld グループを使用する を参照してください。)

   **注記:** サーバー固有のオプションを含む *my.cnf* ファイルは /etc ディレクトリと */<datadir>* ディレクトリの両方に存在してはいけません。 もし、サーバー固有のオプションを含む *my.cnf* ファイルが /etc ディレクトリに存在し、同時に、保護された *my.cnf* ファイルが */<datadir>* にインストールされている場合、コンフリクトの原因となる場合があります。詳細は MySQL のドキュメントのグローバルオプションとサーバー固有のオプションの設定を参照してください。

   以下のエントリを追加または編集します。

   a. ファイルの「client」セクションに、接続に使用するユーザーとパスワードを指定します。

      ```
      [client]
      user =clientuser
      ```

```
password =password
.
.
.
```

b. ファイルの「mysqld」セクションには、接続に使用するソケットとポートを、mysqld プロセスに対する pid ファイルの場所とともに指定します。ユーザー変数では、mysqld プロセスを開始するオペレーティングシステムユーザーを指定します。

```
[mysqld]
socket =/home1/test/mysql/mysql.sock
port =3307
pid-file =/home1/test/mysql/mysqld.pid
user =osuser
```

**注記:** このファイルが正しく保護され、ユーザー「mysql」によって所有されていることを確認します。

**注記:** MySQL 階層を作成した後、`my.cnf` ファイル内の情報を変更する必要がある場合は、変更を加える前に階層を out-of-service (つまり OSU 状態)にして mysql サーバーインスタンスを停止します。

**注記:** 上記の `my.cnf` 設定の例は、単一のデータベースインスタンス `mysqld` について説明しています。mysqld グループを使用した設定例については LifeKeeper で mysqld グループを使用する を参照してください。

**注記:** include文はサポートしていません。単一の`my.cnf`に全ての設定を記述してください。

# 6.7.2.2. MySQL クライアント設定の考慮事項

MySQL データベースクライアントを設定する上で、次のような考慮事項があります。

- クライアントがリモートホストから接続する場合、クライアントが接続に使用する LifeKeeper 保護下の IP アドレスを 1 つ作成します。
- クライアントは、LifeKeeper が保護する IP アドレス経由でデータベースサーバに接続するよう設定する必要があります。
- クライアントがドメイン名を使用して接続する場合は、各クライアントの hosts ファイル内に保護する IP アドレスのエントリを作成するか、DNS 内に名前を設定します。クラスタ内のすべてのクライアントおよびすべての LifeKeeper サーバから ping を実行して、保護する IP アドレスをテストします。
- 各ユーザは、それぞれのマシンのホームディレクトリに *my.cnf* ファイルを持つことができますが、L-ifeKeeper は */etc* ディレクトリまたはデータディレクトリにある *my.cnf* ファイルのみを使用します。 *my.cnf* ファイルにはクライアント接続情報(つまり、ポート、ソケット ID、ユーザ、パスワード)が格納されています。

# 6.7.2.3. MySQL 設定の要件

それぞれの例では、1 つまたは 2 つのデータベースインスタンス (databaseA、databaseB) が使用されています。データベースタグ名は、これらのデータベースインスタンスを LifeKeeper に記述するための任意の名前です。分かりやすい名前を指定することをお勧めします。mysqld グループを使用した設定の LifeKeeper が提示するデフォルトのタグ名は mysql または mysql<group number> です (LifeKeeper で mysqld グループを使用する を参照)。設定例を理解するには、次のような設定要件を留意してください。

- **LifeKeeper 階層。** LifeKeeper の管理を実行するときに、主要な階層は、管理するサーバ上に作成されている階層を参照します。設定の表については、最初の管理画面で入力される情報は Server 1 の観点からのものです。2 番目の画面が表示されると、2 台目のサーバを管理するときに作成される階層を参照します。設定例では、2 台目のサーバは Server 2 です。

- **1 台のサーバによってロックされる共有ディスク。** LifeKeeper を使用するとき、LifeKeeper の保護下にある共有ストレージリソースは、1 台のサーバが使用するためにリザーブしています。これは SCSI リザベーションを使用して行われます。共有デバイスがディスクアレイの場合、LUN 全体がリザーブされます。共有デバイスがディスクの場合は、ディスク全体がリザーブされます。こうすることで、クラスタ内の他のサーバが不用意にデータを破壊してしまうことを避けられます。サーバに障害が発生すると、優先順位が最も高いバックアップサーバがそれまでのリザベーションを解除して新たにリザベーションを行い、他のサーバをすべてロックします。

- **共有ディスク上のデータディレクトリ。** LifeKeeper MySQL Recovery Kit が正しく機能するためには、データベースインスタンスのデータディレクトリ (datadir) が常に共有ディスク上にある必要があります。データディレクトリは、ファイルシステム上になければなりません。このファイルシステムは、プライマリサーバおよびバックアップサーバの両方からマウントできなければなりません。データディレクトリ (datadir) はレプリケーションされたディスクもしくは Network Attached Storage (NAS) 上に存在することが可能です。

# 6.7.2.4. MySQL の設定例

本セクションの例は、MySQL データベースインスタンスの設定方法を示しています。各図は、設定の種類と MySQL パラメータの関係を示しています。また、各設定は、本書で説明している設定の規則と要件に従って、MySQL 設定と LifeKeeper ソフトウェアとの互換性を確保しています。

本セクションでは、設定の要件を説明し、次にそれぞれの設定例を示します。

- アクティブ / スタンバイ
- アクティブ / アクティブ

本セクションに挙げたものは可能な設定の例にすぎませんが、これらの設定を理解し、設定規則に従うことにより、使用するコンピューティング環境で実現可能なソリューションを明確にし設定する際に役立ちます。

設定の要件

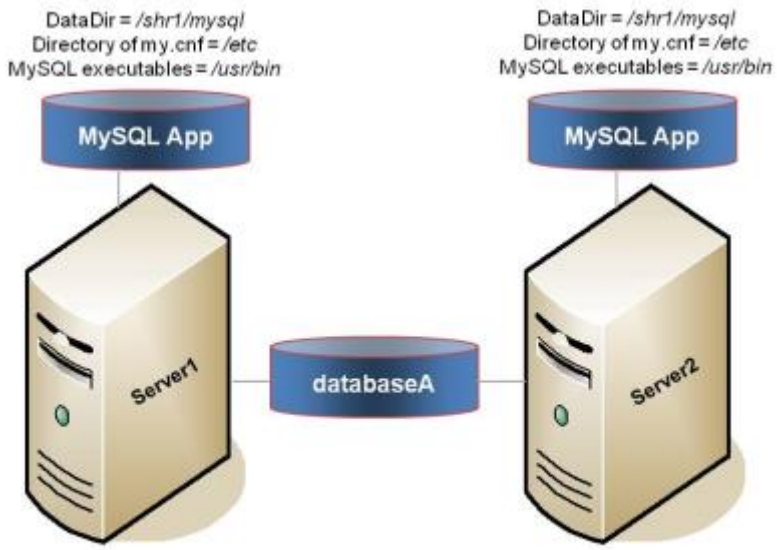
例 1 - アクティブ / スタンバイ設定

例 2 - アクティブ / アクティブ設定

# 6.7.2.5. アクティブ / スタンバイ設定

本セクションでは、アクティブ / スタンバイ設定の例を示します。この設定では、Server 1 がデータベースに排他的にアクセスしているため、Server 1 がアクティブであると見なされます。Server 2 は別の処理を実行します。Server 1 に障害が発生すると、Server 2 がデータベースにアクセス可能になり、LifeKeeper はデータベース操作を再確立します。

**図1.アクティブ / スタンバイ設定、例 1**




**設定に関する注記:**

- どちらのサーバも共有ディスク上の MySQL データディレクトリ(データベース databaseA が格納されている)を使用します。
- MySQL データディレクトリへのパスはどちらのサーバでも同じです。
- *my.cnf* 設定ファイルはローカルディスクの */etc* にあります。
- MySQL 実行ファイルは各サーバのローカルドライブの */usr/bin* にあります。
- Server 2 は、Server 1 がアクティブである間は、共有ディスク上のファイルとディレクトリにアクセスできません。

**Server 1 でのリソース階層の作成**

| | |
|---|---|
| サーバ: | Server1 |
| *my.cnf* ファイルがあるディレクトリ: | */etc* |
| MySQL 実行可能ファイルがあるディレクトリ: | */usr/bin* |

| | |
|---|---|
| データベースタグ: | mysql-on-server1 |

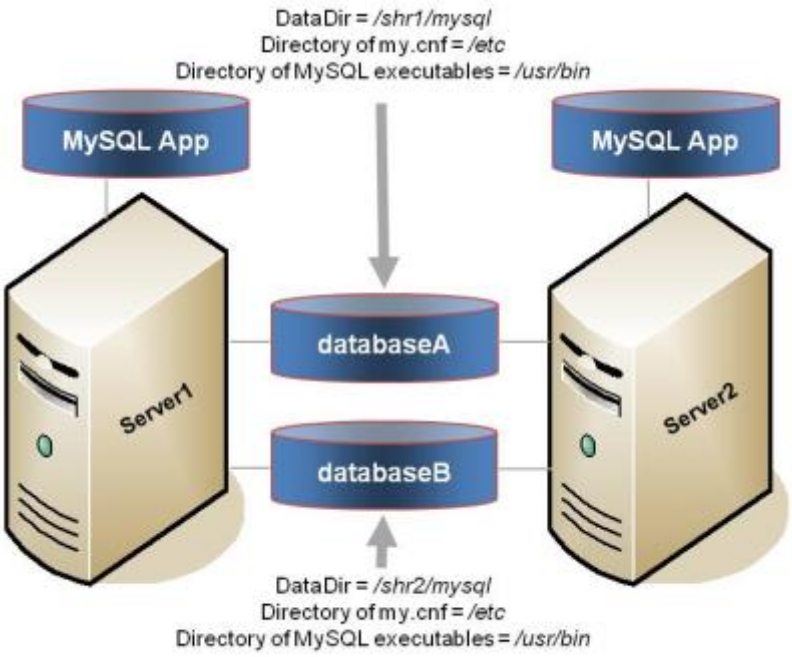
リソース階層の **Server 2** への拡張

| | |
|---|---|
| テンプレートサーバ: | Server1 |
| 拡張するタグ: | mysql-on-server1 |
| ターゲットサーバ: | Server2 |
| ターゲットの優先順位: | 10 |
| *my.cnf* ファイルがあるディレクトリ: | */etc* |
| MySQL 実行可能ファイルがあるディレクトリ: | */usr/bin* |
| データベースタグ: | mysql-on-server2 |

# 6.7.2.6. アクティブ / アクティブ設定

アクティブ / アクティブ設定は、異なるデータベースインスタンスを実行し、互いにバックアップの役割を果たす 2 台以上のサーバで構成されます。各データベースインスタンスは、別々の共有物理ディスク上に置かれていなければなりません。複数の MySQL データーベースインスタンス (同じバージョンもしくは異なるバージョン) をサポートする LifeKeeper の設定のために、SIOS では mysqld グループ 機能をサポートするバージョンの MySQL を使用することを推奨しています。これらの設定では *my.cnf* 設定ファイルは */etc* に存在します。mysqld グループ機能をサポートしないバージョンの MySQL では、 *my.cnf* 設定ファイルは各データベースインスタンスの共有ファイルシステムの MySQL データディレクトリに存在している必要があります(例: 下記図 2 では */shr1/mysql* および */shr2/mysql*)。

**図 2.アクティブ / アクティブ設定、例 2**




設定に関する注記:

- 各サーバは異なる共有ディスク上にある異なる MySQL データディレクトリ(データベースインスタンス database A と database B を含む)を使用します。
- MySQL データディレクトリへのパスは、サーバ上に定義されているインスタンスごとに異なります。
- *my.cnf* 設定ファイルは */etc* に配置され、各データベースインスタンスの mysqld グループセクションを含んでいます。各セクションはデータベースインスタンスに対する一意の MySQL データディレクトリ、ポート、ソケットを定義します。 *my.cnf* 設定ファイルはクラスタの全ノードで常に同一の内容であることに留意してください。mysqld グループをサポートしないバージョンの MySQL が起動しているシステムでは、各データベースインスタンスに対する *my.cnf* 設定ファイルは共有ドライブ上のデータディレクトリに配置されています。各設定ファイルはデータベースインスタンスに対する一意の MySQL データディレクトリ、ポート、ソケット定義を定義します。

- MySQL 実行ファイルは各サーバのローカルドライブの */usr/bin* にあります。
- 最初は、Server 1 で databaseA が、Server 2 で databaseB が稼動します。スイッチオーバが発生した場合は、1 台のサーバで両方のデータベースが稼動できます。

**Server 1** での **1** つ目のリソース階層の作成

| | |
|---|---|
| サーバ: | Server1 |
| *my.cnf* ファイルがあるディレクトリ: | */etc* |
| MySQL 実行可能ファイルがあるディレクトリ: | */usr/bin* |
| データベースタグ: | mysql-shared.example.instance1 |

**1** つ目のリソース階層の **Server 2** への拡張

| | |
|---|---|
| テンプレートサーバ: | Server1 |
| 拡張するタグ: | mysql-shared.example.instance1 |
| ターゲットサーバ: | Server2 |
| ターゲットの優先順位: | 10 |
| *my.cnf* ファイルがあるディレクトリ: | */etc* |
| MySQL 実行可能ファイルがあるディレクトリ: | */usr/bin* |
| データベースタグ: | mysql-shared.example.instance1 |

**Server 2** での **2** つ目のリソース階層の作成**:**

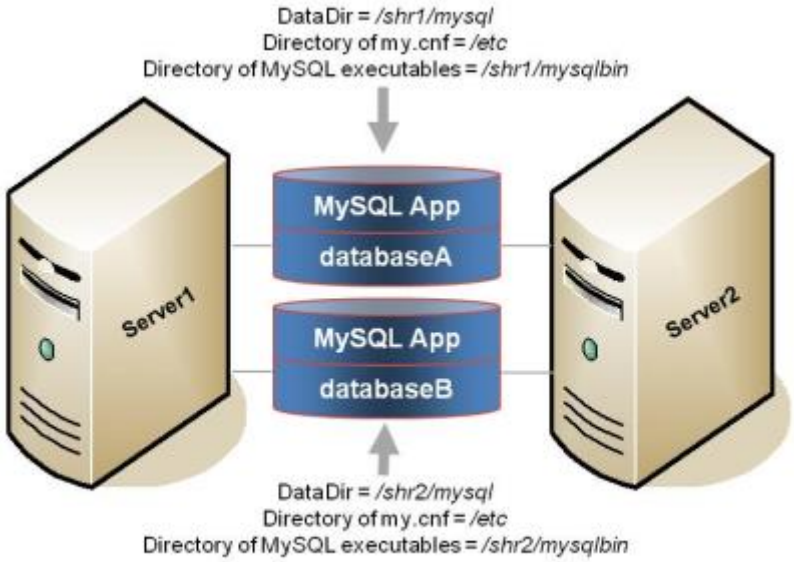
| | |
|---|---|
| サーバ: | Server2 |
| *my.cnf* ファイルがあるディレクトリ: | */etc* |
| MySQL 実行可能ファイルがあるディレクトリ: | */usr/bin* |
| データベースタグ: | mysql-shared.example.instance2 |

**2** つ目のリソース階層の **Server 1** への拡張**:**

| | |
|---|---|
| テンプレートサーバ: | Server2 |

| | |
|---|---|
| 拡張するタグ: | mysql-shared.example.instance2 |
| ターゲットサーバ: | Server1 |
| ターゲットの優先順位: | 10 |
| *my.cnf* ファイルがあるディレクトリ: | */etc* |
| MySQL 実行可能ファイルがあるディレクトリ: | */usr/bin* |
| データベースタグ: | mysql-shared.example.instance2 |

**図 3.アクティブ / アクティブ設定、例 2**




設定に関する注意:

- 各サーバは異なる共有ディスク上にある異なる MySQL データディレクトリ(データベースインスタンス database A と database B を含む)を使用します。
- MySQL データディレクトリへのパスは、サーバ上に定義されているデータベースインスタンスごとに異なります。
- *my.cnf* 設定ファイルは */etc* に配置され、各データベースインスタンスの mysqld グループセクションを含んでいます。各セクションはデータベースインスタンスに対する一意の MySQL データディレクトリ、ポート、ソケットを定義します。 my.cnf 設定ファイルはクラスタの全ノードで常に同一の内容であることを留意してください。mysqld グループをサポートしないバージョンの MySQL が起動しているシステムでは、各データベースインスタンスに対する my.cnf 設定ファイルがデータベースのデータディレクトリの共有ドライブ上に配置されています。各設定ファイルはデータベースインスタンスに対する一意の MySQL データディレクトリ、ポート、ソケットを定義します。
- データディレクトリを含む共有ディスクのそれぞれに MySQL 実行可能ファイルのコピーがありま

す。

- 最初は、Server 1 で databaseA が、Server 2 で databaseB が稼動します。スイッチオーバが発生した場合は、1 台のサーバで両方のデータベースが稼動します。

**Server 1 での 1 つ目のリソース階層の作成**

| | |
|---|---|
| サーバ: | Server1 |
| *my.cnf* ファイルがあるディレクトリ: | */etc* |
| MySQL 実行可能ファイルがあるディレクトリ: | */shr1/mysqlbin* |
| データベースタグ: | mysql-shared.example.instance1 |

**1 つ目のリソース階層の Server 2 への拡張:**

| | |
|---|---|
| テンプレートサーバ: | Server1 |
| 拡張するタグ: | mysql-shared.example.instance1 |
| ターゲットサーバ: | Server2 |
| ターゲットの優先順位: | 10 |
| my.cnf ファイルがあるディレクトリ: | */etc* |
| MySQL 実行可能ファイルがあるディレクトリ: | */shr1/mysqlbin* |
| データベースタグ: | mysql-shared.example.instance1 |

**Server 2 での 2 つ目のリソース階層の作成:**

| | |
|---|---|
| サーバ: | Server2 |
| my.cnf ファイルがあるディレクトリ: | */etc* |
| MySQL 実行可能ファイルがあるディレクトリ: | */shr2/mysqlbin* |
| データベースタグ: | mysql-shared.example.instance2 |

**2 つ目のリソース階層の Server 1 への拡張**

| | |
|---|---|
| テンプレートサーバ: | Server2 |
| 拡張するタグ: | mysql-shared.example.instance2 |

| | |
|---|---|
| ターゲットサーバ: | Server1 |
| ターゲットの優先順位: | 10 |
| *my.cnf* ファイルがあるディレクトリ: | */etc* |
| MySQL 実行可能ファイルがあるディレクトリ: | */shr2/mysqlbin* |
| データベースタグ: | mysql-shared.example.instance2 |

# 6.7.2.7. 複数データベースサーバ環境に関する考慮事項

複数の MySQL データベースサーバおよびデータベースがある場合、次のような設定上の考慮事項があります。

- アクティブ / アクティブまたは複数の(同一もしくは異なるバージョンの) MySQL インスタンスを稼動させる場合、可能であれば mysqld グループの機能を使用することを考慮してください。SIOS は、複数の MySQL データベースサーバ構成に対して mysqld グループ (mysqld_multi) を使用することを推奨します。

- アクティブ / アクティブまたは複数の MySQL インスタンスを稼働させる場合、共有ファイルシステムを */var/lib/mysql* としてマウントしないでください。mysql 起動コマンド (`safe_mysqld` or `mysqld_safe`)により、MySQL サーバが予期せずシャットダウンされるためです。

- mysqld グループの機能を使用しない場合、 *my.cnf* ファイルは、各アクティブ/アクティブもしくは複数サーバのデータディレクトリに保存する必要があります。mysqld グループを使用する構成では、 *my.cnf* ファイルはデータディレクトリではなく */etc* に保存されます。LifeKeeper および mysqld グループ機能に関する情報は LifeKeeper で mysqld グループを使用する を参照してください。

- MySQL 用の追加のポート番号は、 */etc/services* ファイルに指定する必要があります。

- 各 MySQL データベースサーバは、個別のポート上で稼働し、個別のソケットファイルにアクセスするように設定する必要があります。これらの設定オプションは、データディレクトリにある *my.cnf* ファイル内に指定されます。

- 各サーバは、個別の共有の位置からデータにアクセスするように設定する必要があります(つまり、各サーバは個別のデータディレクトリを使用する必要があります)。

# 6.7.2.8. LifeKeeper で mysqld グループを使用する

MySQL Application Recovery Kit は、mysqld_multi を介して管理される mysqld グループ機能を使用した *my.cnf* ファイルをサポートします。この MySQL の機能を使用すると、単一の my.cnf ファイル (通常 */etc* に保存される) を介して、容易に複数の MySQL インスタンスを設定できるようになります。本キットは mysqld グループのフォーマットを利用している my.cnf ファイルを検出し、管理者に対して保護対象とする mysqld グループの番号を選択することを要求します。管理者が選択できる選択リストは、my.cnf ファイルで定義されているグループ番号から、すでに本キットにより保護されているグループ番号を差し引いたものから決定します。

一般的には、mysqld グループ機能を使用して複数の MySQL インスタンスを設定および制御するほうが容易になります。SIOS は、アクティブ / アクティブもしくは複数インスタンスの設定時には、このアプローチを使用することを推奨します。

## my.cnf ファイル

mysqld グループ機能を使用する場合、以下が必須となります。

a. 単一の my.cnf ファイルが、データベースインスタンスの mysqld グループを定義するために使用される必要があります。

b. my.cnf ファイルは共有ストレージ上に配置しないでください。

c. 各クラスタノードに my.cnf ファイルの完全なコピーが必要です (*/etc/my.cnf* が望ましい)。

d. my.cnf ファイルに対して行ったすべての変更は、LifeKeper のすべてのクラスタノードに反映される必要があります。

Recovery Kit は、my.cnf ファイルが mysqld グループを使用していることを検知すると mysqld_multi コマンドを使用します。これに基づいて、LifeKeeper の制御下に配置する前に mysqld_multi を使用して MySQL インスタンスをテストすることができます。

以下は、mysqld_multi によって制御されている 2 つのデータベースインスタンスについて mysqld グループを使用した比較的複雑な my.cnf ファイルの解説です。`mysqld_multi` コマンド (および MySQL LifeKeeper Recovery Kit) は、管理者にさまざまな設定を行うためのオプションを提供しています。下記の例、`[mysqld1]` では、さまざまな MySQL コマンドの大部分についてデフォルトのロケーションを使用する比較的単純な MySQL インスタンスを定義しています。その次の例、`[mysqld55]`では、より多くの変更を行っています。コメントは各セクションにおいて LifKeeper がどのように MySQL と相互作用しているかについて説明しているので参考にしてください。

```
#The following client section defines which username/password combination wil
l be used for
#LifeKeeper  connections. The username/password combination needs to be define
d in each MySQL
```

```
# Database instance that will be described in this my.cnf file.
[client]
user     =  steeleye
password =  password
```

```
# This next section describes the default version of mysqld and mysqldadmin th
at mysqld_multi
# will use when processing mysqld_multi commands. The username/password combo
defines the
# MySQL account that mysqld_multi will use when working with the database inst
ances. This
# username and password combo needs to be defined in each MySQL Database insta
nce that will be
# controlled by mysqld_multi. See how to set up the multi_admin account in th
e MySQL Reference
# Manual, by issuing "mysqld_multi --example".
[mysqld_multi]
mysqld     = /usr/bin/mysqld_safe
mysqladmin = /usr/bin/mysqladmin
user       = multi_admin
password   = password
```

```
# The next section defines the first of two MySQL Database instances in this m
y.cnf file. Note
# that each section starts  with a [mysqldNN] where NN is the mysqld group num
ber (or instance).
# Each group name must have a number. There are a  number of directives that t
he LifeKeeper MySQL
# Recovery Kit will be looking for in these sections.
[mysqld1]
datadir  = /s11/mysql-data5077       #Defines where the data files for the ins
tance will live. For
                                     # LifeKeeper, this directory must be on L
ifeKeeper protected
                                     #   (shared or replicated) storage.
mysqld   = /usr/bin/mysqld_safe       # Defines specifically which mysqld comm
and will be used for
                                      #   starting the instance. This one is u
sing the
                                      #   default  mysqld_safe that came with
the distribution.
socket=/s11/mysql-data5077/moe.socket # Defines the location of the socket fo
r this instance.
                                      #   If the socket is not on LifeKeeper p
rotected storage, it
```

```
                                        #   needs to be defined in exactly the s
ame place on each
                                        #   node in the cluster and be owned by
the "user" defined
                                        #   below.
port     =  3307                       # Each instance needs its own, unique TC
P/IP port.
pid-file = /var/run/mysqld/mysqld.pid # The pid-file can be on LifeKeeper prot
ected or
                                       #   non-LifeKeeper protected storage.
log-error= /var/log/mysqld.log         # Location of the MySQL error log for th
is instance. Can be
                                       #   on LifeKeeper protected or non-LifeKe
eper protected
                                       #   storage.
user     = mysql                       # The Linux user name that will run the
MySQL processes.
```

```
#The next section defines the more complicated of the two MySQL instances. Ins
tance "55" is not
#using the default MySQL that came with the Linux distribution as it is using
the 5.5.12 version
#of MySQL that was installed from source. The binaries for this version were i
nstalled onto shared
#storage, and the binary directory is LifeKeeper protected.
[mysqld55]
datadir = /s11/mysql-data5512           # Same as above; this instance uses a
different data
                                        #   directory, and  this directory is
on LifeKeeper
                                        #   protected storage.
mysqld  =/s11/mysql5512/bin/mysqld_safe # For this instance, a different versi
on of mysqld_safe
                                        #   is used; the  one that is include
d with 5.5.12.
socket=/s11/mysql-misc5512/larry.socket # This instance has the socket on Life
Keeper protected
                                        #   storage, but not  in the default l
ocation (datadir).
port     = 3308                         # This instance has a unique TCP/IP po
rt as well.
pid-file = /var/run/mysqld/mysqld55.pid # This instance's pid-file is not on L
ifeKeeper protected
                                        #   storage.
log-error = /var/log/mysqld55.log       # This instance's log-error (error lo
g) is not on
```

```
                                        #   LifeKeeper protected  storage.
log-bin    = /s11/mysql-log5512/larry   # The log-bin directive specifies wher
e the binary
                                        #   transaction logs are  located for
this instance.
                                        #   These logs must be on LifeKeeper
protected storage
                                        #   (the recovery kit will enforce thi
s). By default,
                                        #   these logs are in the datadir.
user       = mysql                      # The Linux user name that will run th
e MySQL processes.
```

マルチインスタンス用 [mysqld<N>] の設定とシングルインスタンス用 [mysqld] の設定を併記する場合は、シングルインスタンス用の設定を最後に記載してください。

例：

```
[mysqld1]
(mysqld1の設定)

[mysqld2]
(mysqld2の設定)

[mysqld55]
(mysqld55の設定)

[mysqld]
(シングルインスタンス用mysqldの設定)
```

## mysqld_multi コマンド

例として mysql コマンドを実行します。

```
# mysqld_multi start 1
```

my.cnf で[mysqld1]として定義した mysqld グループ 1 インスタンスを、依存するすべての LifeKeeper 保護リソースが LifeKeeper のいずれかのノードで In Service であると仮定します。

以下のコマンドを実行します。

```
# mysqld_multi report 1
```

このインスタンスの状態がレポートされます (例：running (起動) または not running (停止))。このインスタンスが起動すると、LifeKeeper でのリソース作成が容易になります。

mysqld_multi 形式の my.cnf ファイルの設定方法に関する情報については以下のコマンドを実行してください。

```
# mysqld_multi --example
```

# 6.7.2.9. Network Attached Storage の使用

クラスタストレージとして NFS ファイルサーバ (Network Attached Storage) を使用するために LifeKeeper を設定するにあたっていくつかの特別な考慮事項があります。

## NAS Recovery Kit の使用

オプションの Network Attached Storage (NAS) Recovery Kit は、LifeKeeper for Linux で NFS サーバを共有ストレージアレイとして使用する場合に必要になります。各クラスタノードに NAS Recovery Kit (およびライセンス) をインストールしてください。詳細については、NAS Recovery Kit のドキュメンテーションを参照してください。

## エラーメッセージ

MySQL と Network Attached Storage (NAS) を使用する際に、システムクラッシュにより My SQL インスタンスがフェイルオーバ後に再起動できないといった事象に遭遇する可能性があります。MySQL エラーログで、エラーの原因が分かります。

### MySQL 5.0

```
110523 22:10:58 mysqld started
InnoDB: Unable to lock ./ibdata1, error: 11
InnoDB: Check that you do not already have another mysqld process
InnoDB: using the same InnoDB data or log files.
110523 22:10:58 InnoDB: Retrying to lock the first data file
InnoDB: Unable to lock ./ibdata1, error: 11
InnoDB: Check that you do not already have another mysqld process
InnoDB: using the same InnoDB data or log files.
```

### MySQL 5.5

```
110524 10:52:20 InnoDB: The InnoDB memory heap is disabled
110524 10:52:20 InnoDB: Mutexes and rw_locks use GCC atomic builtins
110524 10:52:20 InnoDB: Compressed tables use zlib 1.2.3
110524 10:52:20 InnoDB: Initializing buffer pool, size = 128.0M
110524 10:52:20 InnoDB: Completed initialization of buffer pool
InnoDB: Unable to lock ./ibdata1, error: 11
InnoDB: Check that you do not already have another mysqld process
InnoDB: using the same InnoDB data or log files.
110524 10:52:20 InnoDB: Retrying to lock the first data file
InnoDB: Unable to lock ./ibdata1, error: 11
InnoDB: Check that you do not already have another mysqld process
InnoDB: using the same InnoDB data or log files.
```

これは LifeKeeper によって制御されている NFS ファイルシステム上で、MySQL mysqld プロセスが

*ibdata1* ファイルを NFS lock に設定しているということを示しています。システムクラッシュによりロックがクリアされなかったので、LifeKeeper は MySQL インスタンスを In Service の状態に戻すことができません。MySQL は何か別のプロセスが *ibdata1* ファイルを使用しているとみなします。

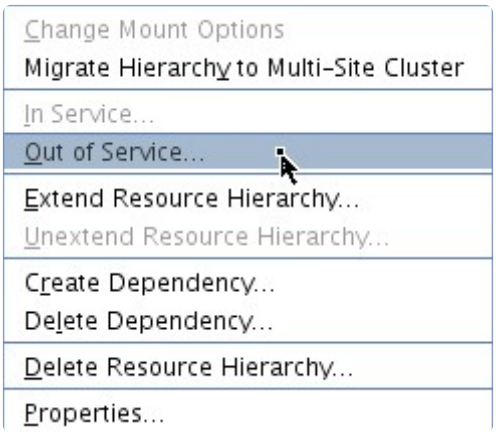
# 解決方法

この事象を解決するためには、ファイルシステムリソースを作成する前に「nolock」とよばれる NFS のオプションで *ibdata1* を格納する予定の NFS ファイルシステムをマウントしてください。デフォルトでは、NFS はファイルをロックさせるよう設定しています。「nolock」オプションがリソース作成前に使用されると、LifeKeeper はこのオプションを認識し、ファイルシステムリソースを In Service の状態にするたびにこのオプションを使用します。LifeKeeper は (クラスタノードから) ibdata1 を含んだファイルシステムへのアクセスを制御するので、通常、ロックは重要ではありません。テスト中に使用される NFS マウントオプションは「`rw,sync,tcp,nfsvers=3,nolock`」です。

MySQL のバイナリが格納されているファイルシステムといったような MySQL リソース階層によって使用されるその他のファイルシステム上で「nolock」を使用する必要はありません。

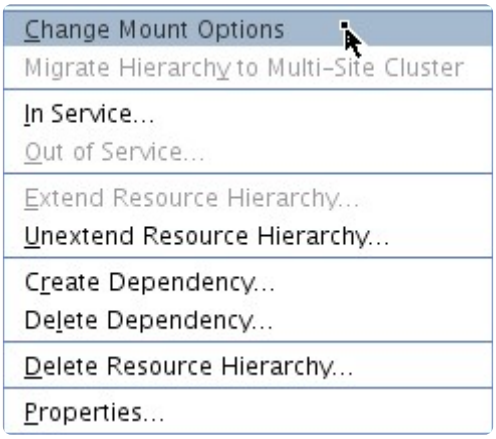
NAS ファイルシステムリソースがすでに「nolock」オプション設定なしに作成されている場合は以下の手順に従ってマウントオプションを変更してください。

1. LifeKeeper GUI を使用して変更する必要があるファイルシステムリソースに対して Out of Service 操作を行ってください。この操作は LifeKeeper GUI でファイルシステムリソース上にポインタを置き、右クリックしてドロップダウンメニューから **[Out of Service]** を選択することで可能です。この操作は親リソースも同じく Out of Service 状態にします。

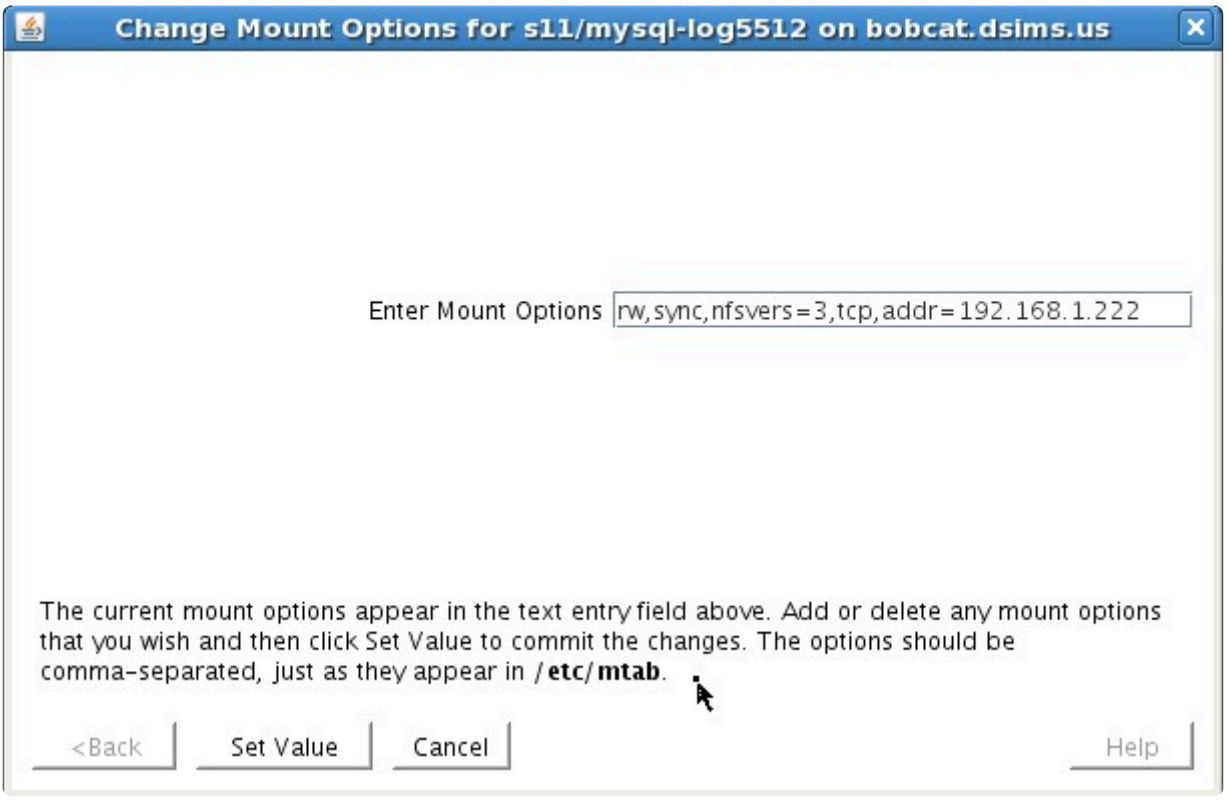



2. **[Out of Service]** 操作を実行し、プロセスを完了させてください。

3. 一旦ファイルシステムリソースが Out of Service 状態になったら、ポインタをリソース上に置き右クリックし、ドロップダウンメニューから **[Change Mount Options]** を選択してください。

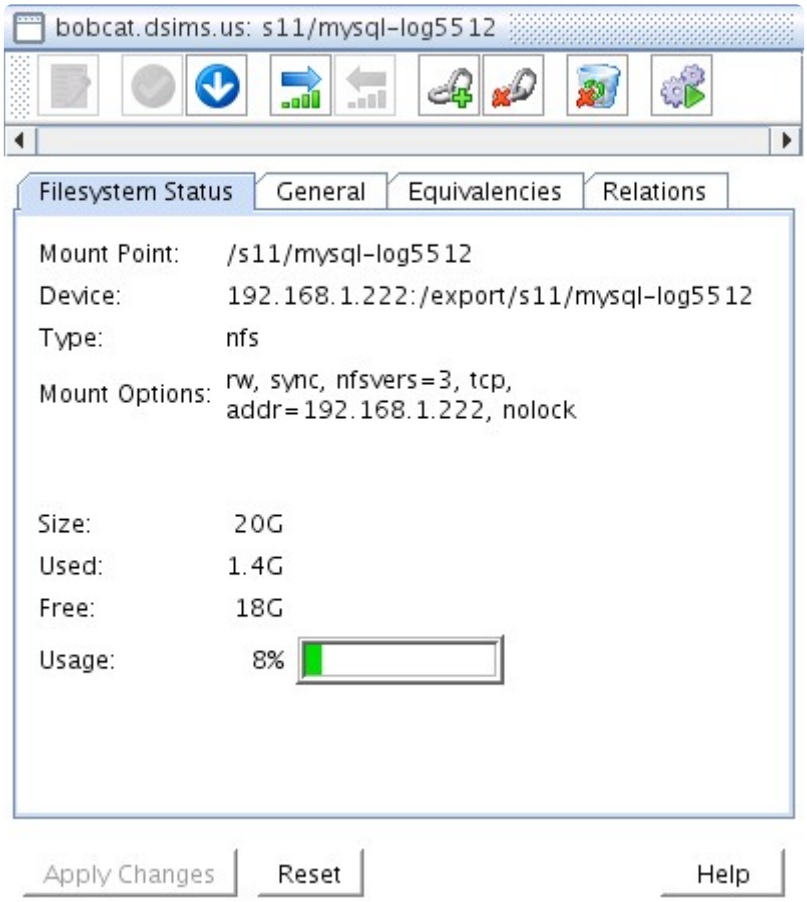



4. ポップアップウィンドウで、オプションの行に「**nolock**」を追加し、 **[Set Value]** をクリックしてください。クラスタの各ノードで手順 3 および 4 を繰り返す必要があります。




5. マウスを右クリックし、 **[In Service]** を選択し、NAS ファイルシステムリソースを In Service の状態に戻してください。

6. ファイルシステムリソースのプロパティパネルが、「nolock」が現在のマウントオプションであることを表示します。


bobcat.dsims.us: s11/mysql-log5512
Filesystem Status
General
Equivalencies
Relations
Mount Point: /s11/mysql-log5512
Device: 192.168.1.222:/export/s11/mysql-log5512
Type: nfs
Mount Options: rw, sync, nfsvers=3, tcp, addr=192.168.1.222, nolock
Size: 20G
Used: 1.4G
Free: 18G
Usage: 8%
Apply Changes
Reset
Help

# 6.7.2.10. systemd 環境で使用する場合の考慮事項

Systemd を採用しているディストリビューションで MySQL（5.7.6以降）が Systemd を使用するように設定されている場合、従来の mysqld_safe コマンドや mysqld_multi コマンドが使えません。これらのコマンドが使えない場合、LifeKeeper は systemctl コマンドを使って MySQL サービスの起動･停止を行います。従って、systemctl コマンドで起動･停止できる必要があります。

Managing MySQL Server with systemd を参考にして、MySQL の設定を行ってください。特に PID File の設定は、Systemd も LifeKeeper も必要とします。必ず Systemd の MySQL 設定と my.cnf の設定を同一にしてください。

リソース作成時は、以下の設定項目に注意してください。

| 設定項目 | 設定値 |
|---|---|
| *Location of my.cnf* | systemctl コマンドで MySQL サービスを起動する際に使用される my.cnf が格納されているディレクトリの絶対パス |
| *Location of MySQL executables* | mysqladmin コマンドがインストールされているディレクトリの絶対パス |

注記: "Location of my.cnf” の設定は、LifeKeeper 内部で my.cnf の設定を読み出すために使用されます。ここで設定した値が systemctl コマンドの起動で使われるわけではありません。デフォルトと異なるパスにある my.cnf を使う場合は、Systemd の MySQL 設定を正しく設定し、期待通り MySQL サービスが起動･停止できることを確認してください。
また、include 文はサポートしていません。単一の my.cnf に全ての設定を記述する必要があります。

# 6.7.3. MySQL インストール

## LifeKeeper での MySQL のインストール / 設定

---

LifeKeeper の設定作業

MySQL リソース階層の作成

MySQL リソース階層の削除

MySQL リソース階層の拡張

MySQL リソース階層の拡張解除

# 6.7.3.1. MySQL Recovery Kit の設定タスク

MySQL Recovery Kitの設定は LifeKeeper GUI を使用して実行できます。以下の 4 つの作業は、MySQL リソースインスタンス特有のものであり、Recovery Kit ごとに異なるため、本セクションで説明しています。

- リソース階層の作成 アプリケーションリソース階層を LifeKeeper クラスターに作成します。
- リソース階層の削除 リソース階層を LifeKeeper クラスター内のすべてのサーバーから削除します。
- リソース階層の拡張 リソース階層をプライマリサーバーからバックアップサーバーへ拡張します。
- リソース階層の拡張解除 リソース階層を LifeKeeper クラスター内の 1 つのサーバーから拡張解除 (削除) します。

以下の作業については、すべての Recovery Kit で同じ手順を使用する共通の作業であるため、 SPS for Linux テクニカルドキュメンテーション の管理 セクションを参照してください。

- リソース依存関係の作成 既存のリソース階層と別のリソースインスタンスとの間に親子の依存関係を作成し、クラスタ内のすべての対象サーバに依存関係の変更を反映します。
- リソース依存関係の削除 リソースの依存関係を削除して、クラスタ内のすべての対象サーバに依存関係の変更を反映します。
- In Service リソース階層を特定のサーバで In Service の状態にします。
- Out of Service リソース階層を特定のサーバで Out of Service の状態にします。
- プロパティの表示 / 編集 特定のサーバでリソース階層のプロパティーを表示または編集します。

**注記:** このセクションの残りの部分では、LifeKeeper GUI の **[Edit]** メニューから作業を選択することによって、Recovery Kit を設定する方法を説明します。設定作業はツールバーから選択することもできます。状況表示ウィンドウの **[Resource Hierarchy Tree]** (左側のペイン)でグローバルリソースを右クリックし、ドロップダウンメニューに **[Edit]** メニューと同じ選択項目を表示することもできます。

状況表示ウィンドウの **[Resource Hierarchy Table]** (右側のペイン)でリソースインスタンスを右クリックして、サーバーおよび特定リソースの状態に応じて、リソース階層の作成を除くすべての設定作業を実行することもできます。

# 6.7.3.2. MySQLリソース階層の作成

> **重要:**
>
> MySQL データーディレクトリファイル(datadir)が共有ディスク（またはDataKeeperによるレプリケーション）にある LifeKeeper クラスタ環境では、共有ファイルシステムがプライマリーサーバー（またはテンプレートサーバー）にマウントされていることを確認する必要があります。ファイルシステムリソースが最初に作成される場合、共有ファイルシステムは必ず各サーバーの同じマウントポイントにマウントされている必要があります。また重要事項として、リソースを作成するには稼働しているコミュニケーションパス(すなわちハートビート)が必須であることを念頭においてください。MySQL データディレクトリは、共有ディスク、レプリケーションされたディスク、もしくは Network Attached Storage 上に存在することが可能です。
>
> レプリケーション（SIOS DataKeeper）ファイルシステムリソースは、MySQLリソースを作成する前に作成する必要があります。他のストレージタイプ上のファイルシステムは、MySQLリソース階層作成時に自動的に作成されます。

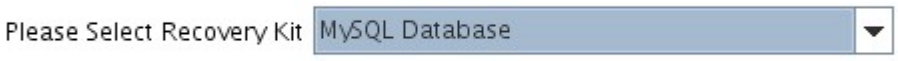
プライマリサーバーからリソースインスタンスを作成するには、以下の手順を実行します。

1. LifeKeeper GUI メニューから **[Edit]** を選択し、次に **[Server]** を選択します。ドロップダウンメニューから、 **[Create Resource Hierarchy]** を選択します。

   すでに行った選択を変更する場合、または MySQL リソース階層を作成するステップのいずれかでエラーメッセージが表示された場合、通常は 1 つ前に戻って、選択を変更するか訂正することができます。([Back] ボタンが有効な場合)。

   **重要:** リソースの作成時には、保護対象の MySQL データベースサーバーデーモン (mysqld) が稼働していなければなりません。

   ダイアログボックスが表示され、ドロップダウンメニューに、クラスターにインストールされ認識されているすべての Recovery Kit が表示されます。ドロップダウンメニューから **[MySQL Database]** を選択します。

   **[Next]** をクリックします。

   階層作成の途中で **[Cancel]** ボタンをクリックすると、作成処理全体が取り消されます。

2. **[Switchback Type]** を選択します。この選択によって、バックアップサーバーへのフェイルオーバーの後、MySQL インスタンスが In Service に戻るときに、プライマリーサーバーにどのようにスイッチバックされるのかが決まります。 ***[intelligent]*** または ***[automatic]*** を選択できます。 ***[intelligent]*** の場合、インスタンスをプライマリーサーバーにスイッチバックするときに管理者の介入が必要になります。 ***[automatic]*** の場合、プライマリーサーバがオンラインに戻り、LifeKeeper

コミュニケーションパスの接続が再確立した直後に自動的にスイッチバックが行われます。

スイッチバックタイプは、必要な場合 **[Resource Properties]** ダイアログボックスの **[General]** タブで後から変更できます。

**[Next]** をクリックします。

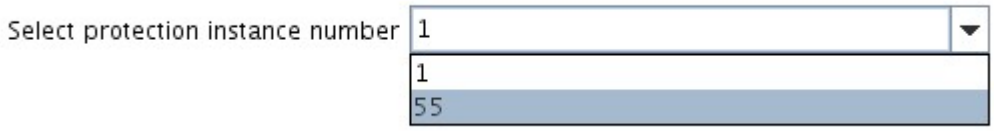
3. **[Server]** で、MySQL データベースインスタンスを配置するサーバー(通常これをプライマリサーバまたはテンプレートサーバと呼ぶ)を選択します。ドロップダウンメニューには、クラスタ内の全サーバーが表示されます。

**[Next]** をクリックして次のダイアログボックスに進んでください。

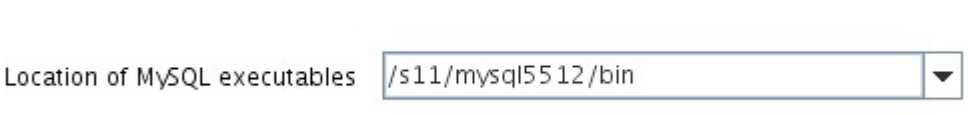
4. **[Location of my.cnf]** を選択または入力します。これは MySQL 設定ファイル (my.cnf) が存在する場所のフルパス名 (ファイル名は除く) です。

**[Next]** をクリックして次のダイアログボックスに進んでください。

5. mysqld_multi 形式の my.cnf ファイルを保有している場合は **[Protection Instance Number]** を選択してください。従来の形式の my.cnf ファイルを保有している場合は、この画面は表示されません。




6. **[Location of MySQL executables]** の場所を選択または入力します。これは、MySQL データベースサーバデーモンの開始と監視に使用されるバイナリーのフルパス名です。

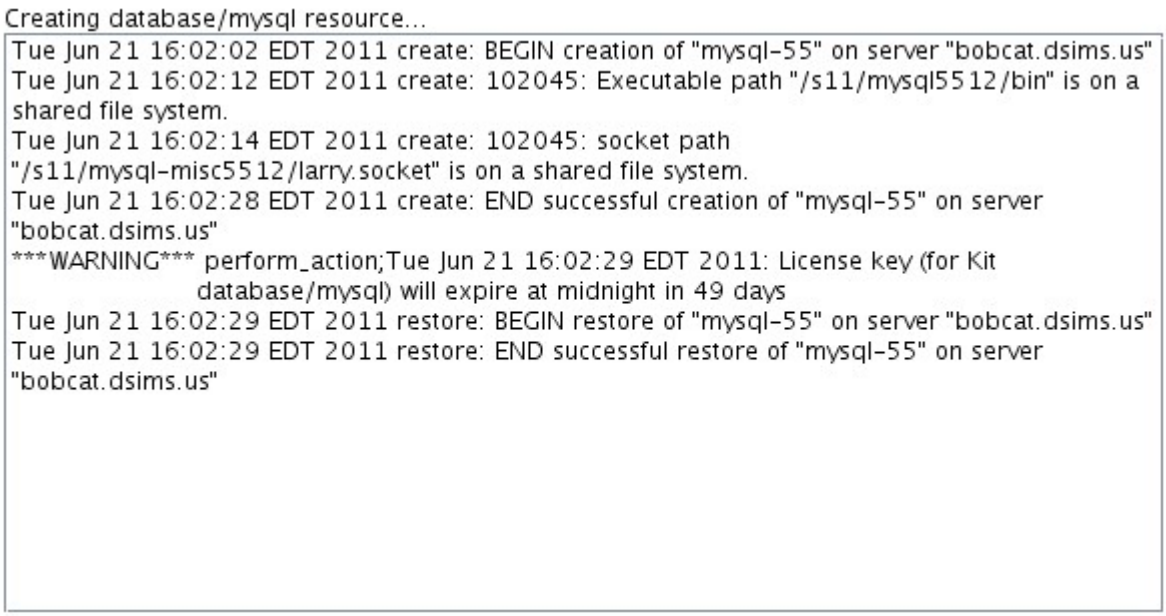


**注記:** この時点で、入力したデータが MySQL リソース階層の作成に妥当かどうかが検査されます。この検査で問題が検出された場合、エラーが画面に表示されます。ディレクトリのパスは妥当で、MySQL 設定自体にエラーがある場合は、ここでそのエラーを訂正してから階層の作成を続けることができます。

**[Next]** をクリックして次のダイアログボックスに進んでください。

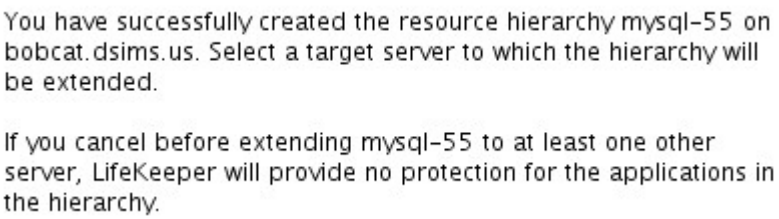
7. **[Database Tag]** を選択または入力します。これは MySQL 階層につけるタグ名です。デフォルトを選択するか、独自のタグ名を入力することができます。

**[Create]** をクリックすると、 **[Create Resource Wizard]** により MySQL リソースが作成されます。



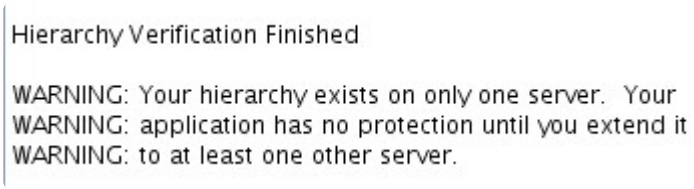

**注記:** この時点で、プライマリーサーバーにMySQL リソース階層が作成されることを確認してください。

8. 別の情報ボックスが表示され、MySQL リソース階層が正常に作成されたこと、MySQL リソース階層を LifeKeeper の保護下に置くには、その階層をクラスタ内の別のサーバに拡張する必要があることが示されます。



**[Continue]** をクリックすると、本セクションで後述する **[Extend Wizard]** が起動されます。

ここで **[Cancel]** をクリックすると、別のダイアログボックスが表示され、いずれかの時点でここに戻り、MySQL リソース階層を別のサーバに拡張して、LifeKeeper の保護下に置く必要があることが警告されます。




9. **[Done]** をクリックして終了します。

# 6.7.3.3. MySQLリソース階層の削除

LifeKeeper 環境内のすべてのサーバからリソース階層を削除するには、以下の手順を実行します。

1. LifeKeeper GUI メニューから **[Edit]** を選択し、次に **[Resouce]** を選択します。ドロップダウンメニューから、 **[Delete Resource Hierarchy]** を選択します。

1. **[Target Server]** で MySQL リソース階層を削除するターゲットサーバの名前を選択します。

   **注記:** 右側ペインで個々のリソースインスタンスを右クリックするか、リソースが 1 台のサーバ上のみにある場合に左側ペインでグローバルリソースを右クリックして **[Delete Resource]** 作業を選択した場合、このダイアログボックスは表示されません。

   **[Next]** をクリックします。

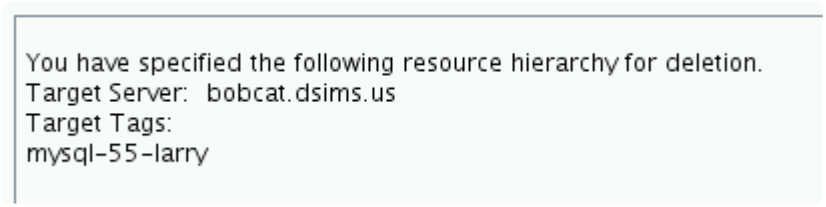
3. **[Hierarchy to Delete]** を選択します。削除するリソース階層を特定して、強調表示にします。

   **注記:** 左側ペインのグローバルリソースまたは右側ペインの個々のリソースインスタンスを右クリックして **[Delete Resource]** 作業を選択した場合、このダイアログボックスは表示されません。

   **[Next]** をクリックします。

4. 選択したターゲットサーバと、削除の対象として選択した階層を確認する情報ボックスが表示されます。




**[Delete]** をクリックします。

5. MySQL リソースが正常に削除されたことを確認する別の情報ボックスが表示されます。

6. **[Done]** をクリックして終了します。

# 6.7.3.4. MySQLリソース階層の拡張

階層の作成後、クラスタ内の別のサーバに拡張する必要があります。3 通りのシナリオで、テンプレートサーバからターゲットサーバにリソースインスタンスを拡張できます。最初のシナリオは、リソース作成後、 **[Continue]** をクリックして、別のサーバにリソースを拡張する場合です。2 番目のシナリオは、以下に示すように **[Edit]** メニューから **[Extend Resource Hierarchy]** を選択する場合です。3 番目のシナリオは、左側または右側のペインから拡張されていない階層を右クリックする場合です。どのシナリオでも同じダイアログボックスが表示されます(いくつかの例外については、以下に詳細を明記する)。

1.
2. LifeKeeperGUI メニューから **[Extend]** ウィザードを開始する場合は、 **[Edit]** を選択し、次に **[Resource]** を選択してください。ドロップダウンメニューから **[Extend Resource Hierarchy]** を選択してください。これで **[Extend Resource Hierarchy]** ウィザードが起動されます。

3. 最初に表示されるダイアログボックスで、 MySQL リソース階層が現在 in service の **[Template Server]** を選択してください。ここで選択する **[Template Server]** と次のダイアログボックスで選択する **[Tag to Extend]** によって、 *in service* のリソース階層が表されることを認識しておくことが重要です選択したテンプレートサーバでサービス中でないリソースタグを選択すると、エラーメッセージが表示されます。このダイアログのドロップダウンボックスには、クラスタ内の全サーバの名前が表示されます。

   **注記:** MySQL リソース階層作成後、直ちに **[Extend Resource Hierarchy]** 作業に入った場合は、作成段階でテンプレートサーバが特定されているため、 このダイアログボックスは表示されません。 これは、GUI ウィンドウの左側ペインの MySQL リソースアイコンまたは右側ペインの MySQL リソースボックスを右クリックして、 **[Extend Resource Hierarchy]** を選択した場合も同様です。

   階層を拡張する手順の間に **[Cancel]** をクリックすると、どの時点であってもそのサーバへの拡張処理がキャンセルされますので注意してください。ただし、すでにリソースを別のサーバに拡張している場合は、明示的に拡張解除するまで、そのインスタンスの拡張は有効です。

   たとえば、Server1 にリソースを作成し、それを Server2 に拡張しているとします。同じリソースを Server3 に拡張している最中に、気が変わっていずれかのダイアログボックス内の **[Cancel]** をクリックしたとします。この場合は、Server3 へのリソース拡張だけがキャンセルされ、Server2 への拡張はキャンセルされません。この階層から Server2 を削除したい場合は、Server2 からリソースを拡張解除する必要があります。

   **[Next]** をクリックして次のダイアログボックスに進んでください。

4. **[Tag to Extend]** を選択してください。これは、テンプレートサーバからターゲットサーバに拡張する MySQL インスタンスの名前です。ウィザードのドロップダウンメニューには、前述のダイアログボックスで選択したテンプレートサーバ上に作成されたすべてのリソースが表示されます。

**注記:** ここでも、MySQL リソース階層作成後、ただちに [Extend Resource Hierarchy] 作業に入った場合は、作成段階で MySQL リソースのタグ名が特定されているため、 このダイアログボックスは表示されません。 GUI ウィンドウの左側ペインの MySQL リソースアイコンまたは右側ペインの MySQL リソースボックスを右クリックして、 **[Extend Resource Hierarchy]** を選択した場合も同様です。

**[Next]** をクリックしてください。

5. MySQL リソース階層を拡張する **[Target Server]** を選択してください。ドロップダウンボックスには、選択された階層に入っていないクラスタ内のサーバ名の一覧が表示されます。

**[Next]** をクリックしてください。

6. **[Switchback Type]** を選択してください。この選択によって、バックアップサーバへのフェイルオーバの後、MySQL インスタンスがサービス中に戻るときに、このサーバにどのようにスイッチバックされるのかが決まります。 ***[intelligent]*** または ***[automatic]*** を選択できます。[intelligent] の場合、インスタンスをプライマリ / オリジナルサーバにスイッチバックするときに管理者の介入が必要になります。[automatic] の場合、プライマリサーバがオンラインに戻り、LifeKeeper コミュニケーションパスを再確立した直後に自動的にスイッチバックが行われます。

スイッチバックタイプは、必要な場合 **[Resource Properties]** ダイアログボックスの **[General]** タブで後から変更できます。

**[Next]** をクリックしてください。

7. **[Template Priority]** を選択または入力してください。これはサーバで現在 in service の MySQL 階層の優先順位です。優先順位は、1 ~ 999 の範囲で未使用の値が有効で、小さい数字ほど優先順位が高くなります(数字 1 が最高の優先順位に相当します)。拡張処理時に、別のシステムですでに使用中の優先順位をこの階層に対して指定することはできません。デフォルト値を推奨します。 **注記:** このフィールドは階層を最初に拡張するときだけ表示されます。

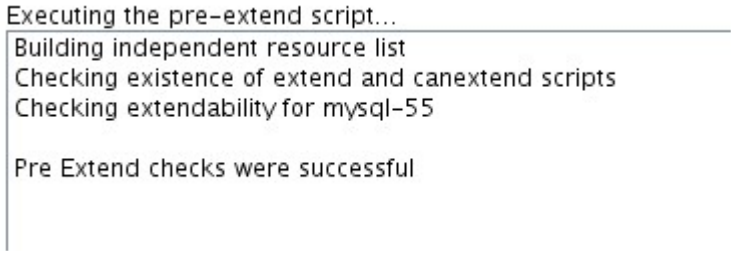
**[Next]** をクリックします。

8. **[Target Priority]** を選択または入力します。これは、別サーバにある同等の階層に対する新しく拡張する MySQL 階層の優先順位です。1 ~ 999 の範囲で、まだ優先順位として使用されていない値が有効で、リソースのカスケーディングフェイルオーバシーケンスにおけるサーバの優先順位を示します。数値が小さいほど優先順位は高くなります(1 は最高の優先順位を表す)。LifeKeeper のデフ

ォルトでは、階層が作成されたサーバに「1」が割り当てられることに注意してください。優先順位は連続している必要はありませんが、特定のリソースについて 2 つのサーバに同じ優先順位を割り当てることはできません。

**[Next]** をクリックしてください。

9. 情報ボックスが表示され、LifeKeeper が環境のチェックを正常に完了し、この MySQL リソースを拡張するためのすべての要件が満たされていることが示されます。満たされていない要件がある場合は、 **[Next]** ボタンは選択できなくなり、 **[Back]** ボタンが有効になります。

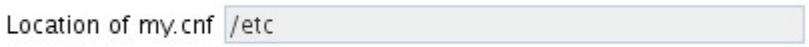



**[Back]** をクリックした場合、情報ボックスに表示されるエラーメッセージの内容に従って、リソースの拡張を変更できます。

ここで **[Cancel]** をクリックした場合は、別の時点でここに戻り、MySQL リソース階層を他のサーバに拡張して、LifeKeeper の保護下に置く必要があります。

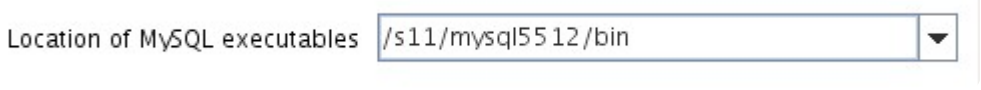
**[Next]** をクリックすると、 **[Extend Resource Hierarchy]** 設定作業に入ります。

10. このダイアログボックスは情報の表示専用です。このボックス内に表示される **[Location of my.cnf]** は変更できません。MySQL インスタンスは設定ファイルから場所情報を取得しています。




**[Next]** をクリックしてください。

11. **[Location of MySQL executables]** を選択または入力してください。これは、MySQL データベースサーバデーモンの開始と監視に使用されるバイナリのフルパス名です。

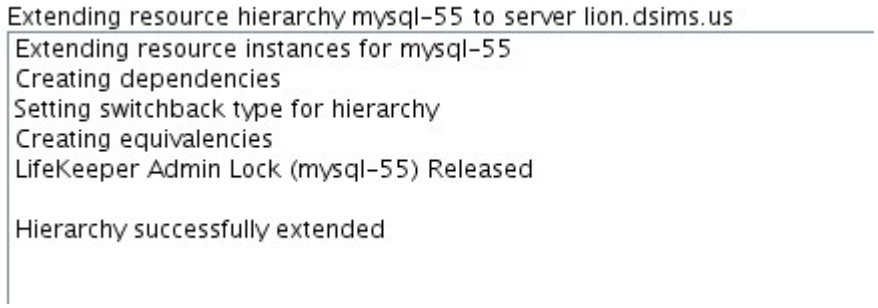



**[Next]** をクリックしてください。

12. **[Database Tag]** を選択または入力してください。これは MySQL 階層につけるタグ名です。デフォルトを選択するか、独自のタグ名を入力することができます。

**[Extend]** をクリックします。

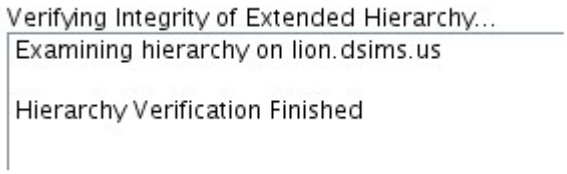
13. 拡張が実行されていることを示す情報ボックスが表示されます。




同じ MySQL リソースインスタンスをクラスタ内の別のサーバに拡張する場合は **[Next Server]** をクリックしてください。その場合は、 **リソース階層の拡張** の操作が繰り返されます。

**[Finish]** をクリックすると、LifeKeeper は MySQL リソースの拡張が正常に完了したことを確認します。

14. **[Finish]** をクリックした場合、次の画面が表示されます。




15. 最後のダイアログボックスで **[Done]** をクリックして終了してください。

**注記:** 必ず *両方の* サーバで新しいインスタンスの機能をテストしてください。

# 6.7.3.5. MySQLリソース階層の拡張解除

1. LifeKeeper GUI メニューから **[Edit]** を選択し、次に **[Resouce]** を選択します。ドロップダウンメニューから、 **[Unextend Resource Hierarchy]** を選択します。

2. **[Target Server]** で、MySQL リソースを拡張解除するターゲットサーバを選択します。MySQL リソースが現在 In Service のサーバは選択できません。

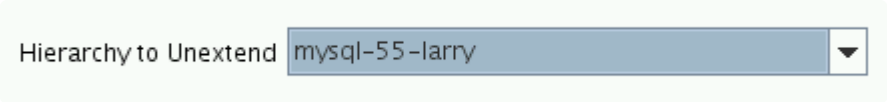
   **注記:** 右側のペインから個々のリソースインスタンスを右クリックして **[Unextend]** 作業を選択した場合、このダイアログボックスは表示されません。

   **[Next]** をクリックします。

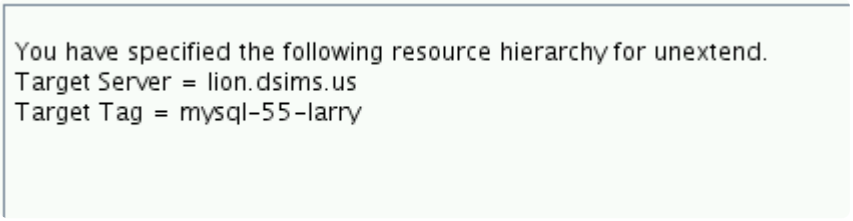
3. **[Hierarchy to Unextend]** で、拡張解除する MySQL 階層を選択してください。

   **注記:** 左側のペインからグローバルリソースを右クリックするか、右側のペインから個々のリソースインスタンスを右クリックして **[Unextend]** 作業を選択した場合、このダイアログボックスは表示されません。

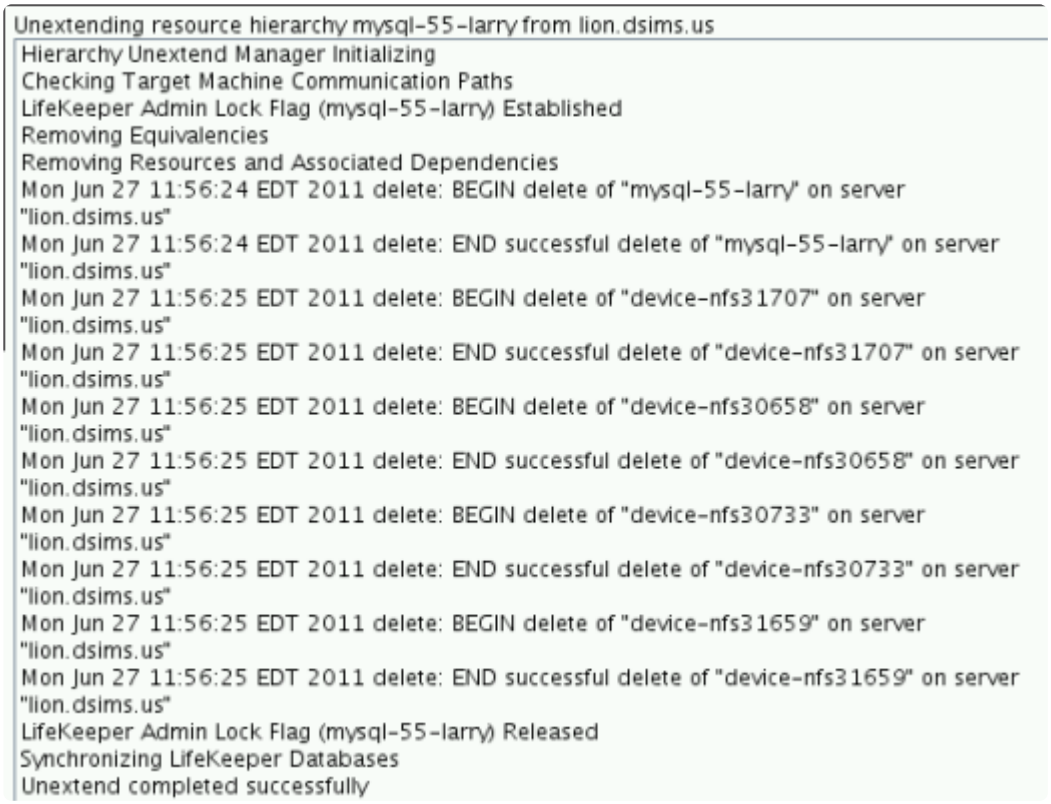
   **[Next]** をクリックします。

4. 拡張解除の対象として選択したターゲットサーバと MySQL リソース階層を確認する情報ボックスが表示されます。

   **[Unextend]** をクリックします。

5. MySQL リソースが正常に拡張解除されたことを確認する別の情報ボックスが表示されます。




6. **[Done]** をクリックして終了します。

# 6.7.4. 管理

## リソース階層のテスト

手動スイッチオーバを開始することによって、MySQL リソース階層をテストできます。これにより、プライマリサーバからバックアップサーバへのリソースインスタンスのフェイルオーバをシミュレートします。

GUI による手動スイッチオーバの実行

# 6.7.4.1. GUI による手動スイッチオーバの実行

手動スイッチオーバを開始することによって、MySQL リソース階層をテストできます。これにより、プライマリサーバからバックアップサーバへのリソースインスタンスのフェイルオーバをシミュレートします。

## GUI による手動スイッチオーバの実行

LifeKeeper GUI で、ドロップダウンメニューから **[Edit] > [Resource] > [In Service]** を選択すると、手動スイッチオーバを開始できます。例えば、バックアップサーバで **[In Service]** 要求を実行すると、アプリケーション階層はバックアップサーバで In Service になり、プライマリサーバでは停止されます。この時点で、元のバックアップサーバがプライマリサーバに、元のプライマリサーバがバックアップサーバに変わります。

**[Out-of-Service]** 要求を実行すると、アプリケーションは他のサーバで In Service にならずに、out of service になります。

LifeKeeper では、ロールバックやバックアップアーカイブのような内部処理の調整や制御は行いません。テープアーカイブやリストアの作業は、アプリケーション管理者の責任です。

## リカバリ動作

プライマリサーバに障害が発生すると、MySQL Recovery Kit ソフトウェアは以下の処理を実行します。

- ファイルシステム (共有もしくは複製) をバックアップサーバ上にマウントします。
- MySQL に関連するデーモンプロセスを開始します。

# 6.7.5. MySQL トラブルシューティング

## 共通のエラーメッセージ

本セクションでは、LifeKeeper MySQL リソース階層の作成時や拡張時、またはリソースの削除時やリストア時に表示される可能性のあるメッセージの一覧を示します。必要に応じて、エラーの原因およびエラー状態を解消するために必要な処置についても説明しています。

他の LifeKeeper コンポーネントからメッセージが出されることもあります。そのような場合は、メッセージカタログ を参照してください。メッセージカタログには、操作、管理、GUI など、LifeKeeper for Linux の使用中に遭遇する可能性のあるすべてのエラーコードが列挙されています。また、エラーコードの原因に関する追加の説明や、問題解決のために必要な処置についても、必要に応じて記載されています。この完全なリストを検索すると、受信したエラーコードを見つけることができます。また、関連する LifeKeeper コンポーネントの個別のメッセージカタログに直接アクセスすることもできます。

## MySQL 固有のエラーメッセージ

注記: エラーメッセージ列で、二重引用符で囲まれたすべて大文字の語はサーバ上のリソースの名前を示します (例えば、"SERVER" は実際には「Server1」という名前のサーバである場合があります)。

| エラー番号 | エラーメッセージ |
|---|---|
| 102001 | Usage: “SCRIPT NAME” sysname dbvarname cnfpath exepath instance |
| 102002 | Usage: “SCRIPT NAME” cnfpath |
| 102003 | Usage: “SCRIPT NAME” exepath cnfpath |
| 102004 | Unable to obtain a valid value for the “socket” variable in “PATH”/my.cnf<br>**処置:** my.cnf 設定ファイルの「mysqld」セクションには、「socket」のエントリがなければなりません。 |
| 102005 | Unable to obtain a valid value for the “port” in “PATH”/my.cnf<br>**処置:** my.cnf 設定ファイルの「mysqld」セクションには、「port」のエントリがなければなりません。 |
| 102006 | Unable to obtain the data directory location “PATH”<br>**処置:** データベースが指定したソケットとポートを使用して稼働していることを確認してください。 |
| 102007 | Must specify the absolute path to the my.cnf configuration file |
| 102008 | Must specify the absolute path to the MySQL executables |
| 102009 | The file my.cnf does not exist in the path specified |
| 102010 | The MySQL executables do not exist in the path specified |

| | |
|---|---|
| 102011 | LifeKeeper was unable to start the MySQL database server |
| 102012 | LifeKeeper successfully started the MySQL database server |
| 102013 | LifeKeeper was unable to stop the MySQL database server |
| 102014 | LifeKeeper successfully stopped the MySQL database server |
| 102015 | The port “PORT NUMBER” is in use on the target server “SERVER” |
| 102016 | The MySQL database server is not running on server “SERVER” |
| 102017 | Unable to open the configuration file “PATH”/my.cnf |
| 102018 | Unable to get the Data Directory information for resource “TAG” on server “SERVER” |
| 102019 | Unable to get the configuration file location information for resource “TAG” on server “SERVER” |
| 102020 | Unable to get the executable location information for resource “TAG” on server “SERVER” |
| 102021 | The argument for the configuration file path is empty |
| 102022 | The argument for the executable path is empty |
| 102023 | The path “PATH” for directive “DIRECTIVE” is not on a shared filesystem |
| 102024 | Unable to get the information for resource “TAG” on system “SYSTEM” |
| 102025 | The MySQL data directory “DATADIR” is already under LifeKeeper protection |
| 102026 | The port variables in the file /etc/my.cnf on “SERVER1” and “SERVER2” do not match |
| 102027 | The socket variables in the file /etc/my.cnf on “SERVER1” and “SERVER2” do not match |
| 102028 | Unable to obtain a valid value for the “user” variable in “PATH”/my.cnf<br>**処置**: my.cnf 設定ファイルの「client」セクションには、「user」変数の有効なエントリがなければなりません。 |
| 102029 | Unable to obtain a valid value for the “password“ variable in “PATH”/my.cnf<br>**処置**: my.cnf 設定ファイルの「client」セクションには、「password」変数の有効なエントリがなければなりません。 |
| 102030 | The user variables in the file /etc/my.cnf on “SERVER1” and “SERVER2” do not match |
| 102031 | The password variables in the file /etc/my.cnf on “SERVER1” and “SERVER2” do not match |
| 102032 | Unable to obtain the pid file location |

| | |
|---|---|
| | **処置**: my.cnf 設定ファイルの mysqld セクションには、「pid-file」変数のエントリがなければなりません。 |
| 102033 | Unable to obtain a valid value for the “user” variable in “PATH”/my.cnf<br>**処置**: my.cnf 設定ファイルの「mysqld」セクションには、「user」変数を使用して OS ユーザを指定する必要があります。 |
| 102034 | 警告: A my.cnf file exists at “PATH”, which may override the values specified in the file at “PATH”/my.cnf. |
| 102035 | The mysql system user “USER” does not exist on target server “SERVER” |
| 102036 | The mysql system user “USER” uids are different on target server “SERVER1” and template server “SERVER2” |
| 102037 | The mysql system user “USER” gids are different on target server “SERVER1” and template server “SERVER2” |
| 102038 | LifeKeeper was unable to stop the MySQL database server using a graceful shutdown. Issuing kill for pid(s): “PROCESS ID LIST”. |
| 102039 | LifeKeeper will ignore failed connection as possible max connections error, due to existence of process pid “PROCESS ID”. |
| 102040 | The mysql action for resource tag :TAG” returned: “COMMAND OUTPUT”. |
| 102041 | LifeKeeper was unable to start the MySQL database server using the defaults-file option. Retrying with individual options. |
| 102042 | The LifeKeeper “ACTION” action detected the flag “FLAG”, and will exit. |
| 102043 | END of “ACTION” action on due to a(n) “SIGNAL” signal. |
| 102044 | The file my.cnf does not exist in the stored path “PATH”. |
| 102045 | “DIRECTIVE” path “PATH” is on a shared filesystem. |
| 102046 | Starting mysqld daemon with databases from “PATH”. |

# 6.7.6. MySQL Recovery Kit 処理概要

## 起動処理 (restore)

以下の順序で MySQL リソースの起動を試みます。

1. 後述する監視処理と同一の項目を確認し、MySQL が起動状態にあることを確認します。起動状態にあると判断された場合は起動処理は成功で終了します。

2. 以下のコマンドで MySQL を起動します。

```
# /usr/bin/systemctl start <サービス名>
```

3. 後述する監視処理と同一の項目を確認し、MySQL が起動状態にあることを確認します。確認に失敗した場合はエラーで終了します。

## 停止処理 (remove)

リソース作成時に指定した MySQL の設定ファイル (my.cnf) より各情報を取得し、MySQL リソースの監視を試みます。

1. 後述する監視処理と同一の項目を確認し、MySQL が起動状態にあることを確認します。起動状態にあると判断された場合は以下のコマンドで MySQL を停止します。戻り値により停止コマンドが成功したと判断された場合は停止処理は成功で終了します。

```
# mysqladimin -u <ユーザー名> --password=<パスワード> --socket=<ソケット> shu
tdown
```

ユーザー名 : my.cnf の [client] セクション内の [user] の値
パスワード : my.cnf の [client] セクション内の [password] の値
ソケット : my.cnf の [mysqld] セクション内の [socket] の値

2. 項目1. において戻り値により停止コマンドが失敗したと判断された場合は、MySQL の PID ファイルの存在の有無により、以下の様に分岐しプロセスが存在しないことを確認します。確認に成功した場合は停止処理は成功で終了します。

   a. PID ファイルが存在する場合は、ps コマンドの出力中に MySQL の PID ファイルに記載された番号のプロセスが存在しないことを確認します。

      PID ファイル : my.cnf の [mysqld] セクション内の [pid-file] の値

   b. PID ファイルが存在しない場合は、ps コマンドの出力中に MySQL の ポート番号とソケットを使用するプロセスが存在しないことを確認します。

ポート番号 : my.cnf の [mysqld] セクション内の [port] の値
ソケット : my.cnf の [mysqld] セクション内の [socket] の値

3. 項目2. のいずれかより確認したプロセス ID をもとに、ps コマンドより親プロセスを確認し、親プロセスに対し SIGTERM を発行し、5秒待機します。

4. ps コマンド出力中に項目3. より確認した親プロセスが存在する場合は、未だに停止状態にないと判断し、親プロセスに対し SIGKILL を発行し、5秒待機します。

5. 項目2. のいずれかより確認したプロセス ID に対し SIGTERM を発行し5秒待機します。SIGTERM 発行の成功または失敗により、停止処理の成功または失敗として終了します。

# 監視処理 (quickCheck)

リソース作成時に指定した MySQL の設定ファイル (my.cnf) より各情報を取得し、MySQL リソースの監視を試みます。

1. 以下のコマンドで VARIABLES を参照します。参照できた場合は成功で終了します。

```
# mysqladimin -u <ユーザー名> --password=<パスワード> --socket=<ソケット> variables
```

ユーザー名 : my.cnf の [client] セクション内の [user] の値
パスワード : my.cnf の [client] セクション内の [password] の値
ソケット : my.cnf の [mysqld] セクション内の [socket] の値

2. 項目1. において VARIABLES が参照できない場合は、MySQL の PID ファイルの存在の有無により、以下の様に分岐しプロセスが存在することを確認します。確認に失敗した場合はエラーで終了します。

   a. PID ファイルが存在する場合は、ps コマンドの出力中に MySQL の PID ファイルに記載された番号のプロセスが存在することを確認します。

      PID ファイル : my.cnf の [mysqld] セクション内の [pid-file] の値

   b. PID ファイルが存在しない場合は、ps コマンドの出力中に MySQL のポート番号とソケットを使用するプロセスが存在することを確認します。

      ポート番号 : my.cnf の [mysqld] セクション内の [port] の値
      ソケット : my.cnf の [mysqld] セクション内の [socket] の値

# 回復処理 (recover)

1. 上述の停止処理と起動処理を順に実行し、MySQL リソースの回復を試みます。

# 6.8. WebSphere MQ Recovery Kit 管理ガイド

## MQ Recovery Kit テクニカルドキュメンテーション

SIOS Protection Suite for Linux WebSphere MQ Recovery Kit は、WebSphere MQ キューマネージャとキューマネージャのストレージ場所に、耐障害性保護を提供します。このキットを使用すると、プライマリ WebSphere MQ サーバまたはキューマネージャ上で障害が発生した場合、人手の介入なしに迅速にプライマリサーバまたは指定のバックアップサーバで復旧することが可能です。

## ドキュメントの内容

このガイドでは次の事項について説明します。

- SIOS Protection Suite ドキュメンテーション 。SPS for Linux のドキュメンテーションの一覧と入手場所を記載します。
- 略語 。本ドキュメンテーション全体で使用される略語の一覧およびその意味を記載します。
- 要件 。WebSphere MQ Recovery Kit を適切に設定、インストール、運用する上で必要なハードウェアとソフトウェアを説明します 。SPS for Linux ソフトウエアをインストールまたはアンインストールする方法の具体的な手順については、SIOS Protection Suite インストールガイド を参照してください。
- WebSphere MQ Recovery Kit の概要 。WebSphere MQ Recovery Kit の特徴および機能の概要と、この Recovery Kit でサポートされる WebSphere MQ ソフトウェアのバージョンの一覧を示します。
- WebSphere MQ を設定する際の考慮事項 。設定上の問題の概要と、WebSphere MQ Recovery Kit でサポートされるファイルシステムのレイアウトを示します。
- LifeKeeper 使用時の WebSphere MQ の設定。 WebSphere MQ のインストール手順と LifeKeeper とともに使用するための設定手順を示します。
- リソース作成後の設定変更 。WebSphere MQ の設定変更が LifeKeeper WebSphere MQ のリソース階層に与える影響に関する情報を記載します。
- WebSphere MQ の設定例 。一般的な WebSphere MQ 設定の例と、WebSphere MQ リソースの設定手順を示します。
- LifeKeeper の設定作業。 LifeKeeper GUI を使用して WebSphere MQ リソース階層を作成および管理するための作業を説明します。
- WebSphere MQ のトラブルシューティング。 情報メッセージとエラーメッセージ、および推奨される対処方法の一覧を示します。
- 付録 。WebSphere MQ の設定ファイルの例と、WebSphere MQ インストールを計画するために使

用できる設定シートが含まれます。

# 参考ドキュメント

MQドキュメンテーションは、以下の WebSphere MQ Library にあります。

http://www.ibm.com/software/integration/wmq/library/

# SPS/LifeKeeper ドキュメンテーション

以下は、SIOS Technology Corp が公開している SPS 関連情報の一覧です。

- SPS for Linux リリースノート
- SPS for Linux テクニカルドキュメンテーション
- SIOS Protection Suite インストールガイド
- Optional Recovery Kit ドキュメンテーション
- SPS for Linux IP Recovery Kit 管理ガイド

このドキュメンテーションと、その他の SPS Recovery Kit に関連するドキュメンテーションは、以下の Web サイトで入手できます。

http://jpdocs.us.sios.com

# 6.8.1. 略語

本ドキュメンテーション全体で、使用される略語は次のとおりです。

| 略語 | 意味 |
|---|---|
| HA | Highly Available, High Availability (高可用性) |
| QMDIR | WebSphere MQ queue manager directory.このディレクトリは、キューマネージャのパーシステントキューデータを格納します。一般的に、 */var/mqm/qmgrs* 内に、キューマネージャー名をサブディレクトリ名として配置されます。このディレクトリの正確な場所は、グローバル設定ファイル *mqs.ini* 内で指定されます。<br><br>DataPath パラメータを定義する場合は、キューマネージャー名を含む DataPath 値でキューマネージャのパーシステントデータの場所が指定されます。定義しない場合は、上記のデフォルトの場所が使用されます。 |
| QMLOGDIR | WebSphere MQ queue manager log directory.このディレクトリは、キューマネージャのログデータを格納します。一般的に、 */var/mqm/log* 内に、キューマネージャー名をサブディレクトリ名として配置されます。このディレクトリの正確な場所は、キューマネージャ設定ファイル (*QMDIR/qm.ini*) 内で指定されます。 |
| MQUSER | すべての WebSphere MQ コマンドを実行するオペレーティングシステムのユーザ。このユーザは、 QMDIR の所有者です。このユーザは、MQGROUP 管理グループ mqm (下記を参照) のメンバである必要があります。 |
| MQGROUP | MQUSER が参加すべきオペレーティングシステムのユーザーグループ。このグループの名前は、mqm である必要があります。 |
| UID | オペレーティングシステムユーザの数値 ID。 |
| GID | オペレーティングシステムユーザグループの数値グループ ID。 |

# 6.8.2. MQ Recovery Kit の要件

WebSphere MQ Recovery Kit をインストールする前に、LifeKeeper の設定が次の要件を満たしている必要があります。LifeKeeper ハードウェアおよびソフトウェアの設定に関する具体的な手順については、LifeKeeper for Linux 管理ガイド を参照してください。

# 6.8.2.1. MQ ハードウェア要件とソフトウェア要件

## ハードウェア要件

- サーバー ：Recovery Kit には、LifeKeeper for Linux インストールガイド に記載の要件に従って設定された 2 台以上のサーバが必要です。 サポートされる Linux ディストリビューションについては、Linux の設定 を参照してください。

- データストレージ：WebSphere MQ Recovery Kit は、共有ストレージと DataKeeper 製品によって提供される複製されたストレージの両方と組み合わせて使用できます。 また、Network Attached Storage (NAS) と使用することもできます。

## ソフトウェア要件

- **LifeKeeper** ソフトウェア ：各サーバーにインストールする **LifeKeeper** ソフトウェアおよびパッチのバージョンは同じにする必要があります。

- **LifeKeeper WebSphere MQ Recovery Kit** ：WebSphere MQ v7.1 以降を実行するシステムには、WebSphere MQ Recovery Kit バージョン 7.5.1 以降が必要です。

- **LifeKeeper IP Recovery Kit** ：各サーバーの LifeKeeper IP Recovery Kit のバージョンは同じにする必要があります。

- **IP** ネットワークインターフェース ：各サーバーは、イーサーネット TCP/IP をサポートするネットワークインターフェースを 1 つ以上必要とします。IP スイッチオーバが正しく動作するには、ローカルネットワークに接続されているユーザシステムが標準の TCP/IP 仕様に準拠している必要があります。

  注記: 各サーバが必要とするネットワークインターフェースが 1 つだけであっても、異種メディア要件、スループット要件、単一障害点の排除、ネットワークのセグメンテーションといった多くの理由のために、複数のインターフェースを使用すべきです。

- **TCP/IP** ソフトウェア ：各サーバは TCP/IP ソフトウェアも必要とします。

- **WebSphere MQ** ソフトウェア ：IBM WebSphere MQ は、別途 IBM から注文する必要があります。 サポートされる WebSphere MQ のバージョンについては、SPS リリースノート を参照してください。WebSphere MQ ソフトウエアは、WebSphere MQ Recovery Kit をインストールする前にクラスタの各サーバにインストールする必要があります。WebSphere MQ Recovery Kit を正常にインストールするには、次の WebSphere MQ パッケージがインストールされている必要があります。

  MQSeriesServer、MQSeriesSamples、MQSeriesClient、MQSeriesRuntime、MQSeriesSDK

  IBM WebSphere MQ バージョン 7.0.1 Fix Pack 6 以降、複数のバージョンの WebSphere MQ (例: MQ バージョン 7.0.1 Fix Pack 6 と 7.1) を同一サーバ上でインストールおよび実行できるようにする新機能が導入されました。 この機能 (マルチインスタンスサポート) は、WebSphere MQ

Recovery Kit 9.0.1 のバージョン以降でサポートされるようになりました。マルチインスタンス機能の一部として導入された単一の IBM WebSphere MQ インストールバージョン内で複数のキューマネージャを保護する機能と、複数の IBM WebSphere MQ インストールバージョン内でキューマネージャーを保護する機能、mqs.iniファイル 内での DataPath パラメータの使用は、このバージョンのリカバリキットでいずれもサポートされています。

- **オプションの C コンパイラ** ：WebSphere MQ Recovery Kit には、WebSphere MQ サンプルパッケージ内の修正版 amqsget0.c サンプルプログラムが含まれています。このプログラムは、デフォルトの 15 秒のタイムアウトではなく、0 秒のタイムアウトに対応するよう修正されました。このプログラムは、キューマネージャの PUT/GET テストを実行するために使用されます。このプログラムは RPM のインストール中にコンパイルされるため、C コンパイラがインストールされ、root ユーザの PATH に配置されている必要があります。

- **Syslog.pm**。 WebSphere MQ リソースで syslog ログを使用するには、Perl モジュール の Syslog.pm がインストールされている必要があります。このモジュールは、標準の Perl ディストリビューションに含まれているため、別途インストールする必要はありません。

# 6.8.2.2. LifeKeeper MQ クラスタのアップグレード

1. WebSphere MQ Recovery Kit を含むクラスタ内のすべてのノード上の LifeKeeper を、LifeKeeper インストールガイド の Upgrading セクションに記載の手順に従ってアップグレードします。

2. クラスタ内の各ノード上の IBM WebSphere MQ ソフトウェアを以下の手順でアップグレードします。

   a. 1 つまたは複数の LifeKeeper IBM WebSphere MQ リソース階層がアップグレードされるノード上で in service の場合、IBM WebSphere MQ ソフトウェアのアップグレードを実行する前に out of service にしなければなりません。この設定は、スタンドバイノードへのスイッチオーバによって可能です。

   b. IBM WebSphere アップグレード手順に従ってください。

3. IBM WebSphere ソフトウェアがノードにインストールされたら、LifeKeeper IBM WebSphere MQ リソース階層を in service にし (リストア)、各 Queue Manager の動作を確認します。

4. 各 Queue Manager の動作を確認したら、クラスタ内の他のすべてのノードをアップグレードします。

# 6.8.3. WebSphere MQ Recovery Kit の概要

WebSphere MQ (旧 MQSeries) は、IBM のソフトウェア製品で、一度だけの信頼性の高いメッセージ配信を行います。WebSphere MQ の中核要素は、メッセージを送受信するための1 つ以上のキューを処理するキューマネージャです。メッセージがキューに追加されると、メッセージは確実にパーシステントになり、一度だけ配布されます。

WebSphere MQ Recovery Kit を使用すると、コマンドサーバ、リスナー、およびパーシステントキューマネージャデータを含む WebSphere MQ キューマネージャを LifeKeeper で保護できます。キューマネージャリスナーの保護は、オプションでキューマネージャごとに無効化できます。これにより、クライアント接続を扱わない設定に対応したり、LifeKeeper のリカバリ試行を発動させることなく管理者がリスナーをシャットダウンしたりできます。

WebSphere MQ Recovery Kit は、保護対象の WebSphere MQ キューマネージャを、障害の発生したプライマリサーバからバックアップサーバにリカバリする仕組みを提供します。LifeKeeper はサーバレベル (ハートビートによる) またはリソースレベル (WebSphere MQ デーモンの監視による) で障害を検知して、保護対象の WebSphere MQ サービスの管理をバックアップサーバに移譲します。

また、WebSphere MQ Recovery Kit は、複数のバージョンの WebSphere MQ のインストールをサポートしており、同じシステム上にインストールして実行することができます。MQ のマルチバージョンはMQ ソフトウェアのバージョン 7.x、およびバージョン 8.x のキューマネージャをサポートし、いずれも Recovery Kit により保護することが可能です。WebSphere MQ Recovery Kit の 9.0.2 のバージョンにおける本サポート以前は、同じシステムにインストールして実行できる MQソフトウェアバージョンは 1 つのみでした (注記: インストールは、デフォルトの場所 /opt/mqm に行う必要がありました)。

# 6.8.3.1. WebSphere MQ リソース階層

典型的なWebSphere MQ階層はWebSphere MQキューマネージャーリソースによって構成されます。また、ファイルシステム構成や IP リソースの有無に応じて、1 つまたは複数のファイルシステムリソースが含まれます。具体的な階層構成は、保護する内容によって変わります。IP リソースを WebSphere MQ リソース階層に含める場合、WebSphere MQ キューマネージャリソースを作成する前に管理者はその IP を作成し、またその IP リソースをプライマリサーバー上でアクティブにする必要があります。ファイルシステム階層は、WebSphere MQ キューマネージャリソースの作成中に自動的に作成されます。しかし、ファイルシステムリソースにレプリケーション (SIOS DataKeeper)ファイルシステムを使用する場合は、MQ リソースを作成する前に作成する必要があります。

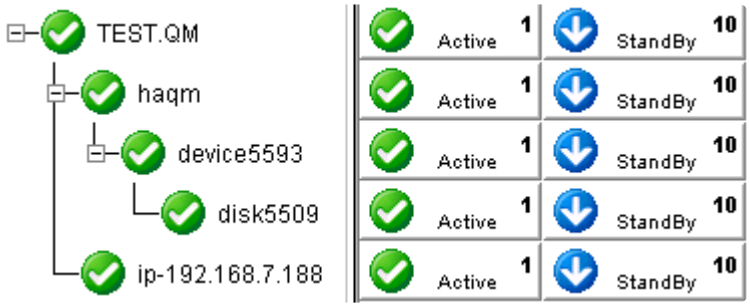



図 1 一般的な WebSphere MQ 階層 (シンボリックリンク)

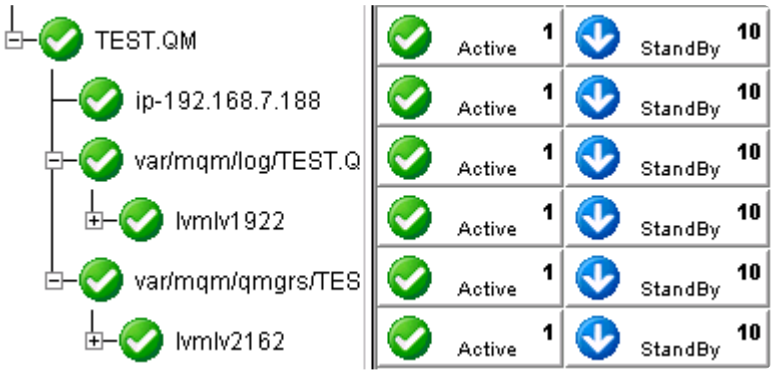



図 2 一般的な WebSphere MQ 階層 (LVM 設定)

# 6.8.3.2. MQ Recovery Kit の機能

WebSphere MQ Recovery Kit は、次の機能を提供します。

- マルチインスタンスのキューマネージャ (MQ ソフトウェアの複数のバージョンで作成されたキューマネージャ) のサポート
- アクティブ / アクティブ設定のサポート
- LINEAR および CIRCULAR ログ (自動検出) のサポート
- サーバ接続およびクライアント接続によるエンドツーエンドアプリケーション健全性チェックのサポート
- オプションの PUT/GET テスト (テストキューを GUI とコマンドラインで定義可能) のサポート
- カスタマイズ可能なログレベルのサポート
- LifeKeeper でサポートされるすべてのストレージタイプをサポート
- オプションのリスナー保護のサポート (デフォルトで有効)
- 追加の syslog メッセージログ (ログ機能 local7) のサポート
- 複数レベルのコマンドサーバ保護 (デフォルトは「full」レベル)

# 6.8.4. WebSphere MQ を設定する際の考慮事項

本セクションでは、WebSphere MQ の設定を開始する前に考慮すべき事項について説明します。また、WebSphere MQ を LifeKeeper キューマネージャで設定および保護するための操作手順を示します。

WebSphere MQ を LifeKeeper で Linux ディストリビューションにインストールする方法については、WebSphere MQ ドキュメンテーション を参照してください。

# 6.8.4.1. MQ 設定の要件

LifeKeeper 使用時の WebSphere MQ の設定 セクションには、LifeKeeper でキューマネージャを保護するための手順が記載されています。一般的に、LifeKeeper とともに使用する WebSphere MQ キューマネージャを正しく設定するには、次の要件が満たされている必要があります。

1. **カーネルパラメータの設定** ：Linux カーネルパラメータ（共有メモリやその他のカーネルリソースなど）の設定方法については、WebSphere MQ ドキュメンテーション を参照してください。

2. **MQUSER** および **MQGROUP** ：MQGROUP および MQUSERは、クラスタのすべてのサーバー上に存在する必要があります。Websphere MQ ソフトウェアでは、MQGROUP mqm が存在し、それが、プライマリグループメンバーシップがMQGROUP mqmに設定されていると定義されたMQUSER mqmを持っている必要があります。 もし、Websphere MQソフトウェアがインストールされるときにmqm userとmqm groupが存在しない場合には、mqm userとmqm groupは自動的に生成されます。 WebSphere MQソフトウェアをインストールする時には、 ほとんどのファイルとディレクトリは、mqm userとmqm groupに対して設定されたユーザーおよびグループの所有権を持ちます。Queue Managerデータとログディレクトリの中のファイルとディレクトリのユーザーおよびグループの所有権は、mqm userとmqm groupに対しても設定されます。さらに、Queue Managerが開始される時には、それがmqm userとして動作します。 そのため、MQUSER ユーザーID (uid) とMQGROUPグループID (gid) は、すべてのクラスターのサーバー上で同一でなくてはなりません。MQ Recovery Kitはリソースを拡張しようとする際に、この点を検証します。もし、この二つのIDが一致しない場合には、リソースの拡張に失敗します。注：もしローカルパスワードやグループファイル以外にNIS、LDAPもしくは他の認証ツールを使用している場合には、Websphere MQとLifeKeeperソフトウェアのインストールの前にMQUSERとMQGROUPを設定しなくてはなりません。また、ホームディレクトリを作成する必要があるかもしれません。もし、今回が以前にリリースされたWebSphere MQ Recovery Kitからのアップグレードである場合、MQUSER PATH環境変数の設定の変更が必要な場合があります。このMQUSER PATH環境変数は、WebSphere MQ ソフトウェアのデフォルトインストールロケーション、 /opt/mqmを含めるために変更する必要があります。もし、この変更が以前のリリースで行われている場合、今回のRecovery Kitのバージョンが正しく動作するために解除する必要があります。

3. 別のMQユーザーのサポート Websphere MQ ソフトウェアは常にmqm ユーザーとして動作しますが、別のユーザーがプライマリもしくはセカンダリメンバーシップをmqmグループの中に持つことを前提に、全てのMQコマンドを動作させるのに別のユーザー名を指定することが可能です。他のMQ Toolと統合する時には、WebSphere MQの開始に別のユーザー名が必要かもしれません。別のユーザーへの変更については、本文書の“LifeKeeper WebSphere MQ Recovery Kit のデフォルト設定の変更”の MQS_ALT_USER_NAME設定 を参照してください。

4. **コマンドサーバーの手動起動** ：LifeKeeper によってコマンドサーバーを起動するには、プライマリサーバー上で、次のコマンドを使用してコマンドサーバーの自動起動を無効にします。この手順を行わない場合、キューマネージャの起動時にコマンドサーバーが自動的に起動されます。

```
runmqsc QUEUE.MANAGER.NAME

ALTER QMGR SCMDSERV(MANUAL)
```

5. 別のMQユーザーのサポート **が共有ストレージ上にあること** 。キューマネージャディレクトリ

QMDIR およびキューマネージャログディレクトリ *QMLOGDIR* は、LifeKeeper でサポートされた共有ストレージ上にある必要があります。これにより、バックアップサーバー上の WebSphere MQ がデータにアクセスできます。詳細については、サポートされるファイルシステムレイアウト を参照してください。

6. **QMDIR および QMLOGDIR の権限** ：*QMDIR* および *QMLOGDIR* ディレクトリは、MQUSER と MQGROUP グループに所有されている必要があります。ARK は、ディレクトリの所有者を参照して MQUSER を自動的に決定します。また、シンボリックリンクを検出し、最終ターゲットに到達します。必要な場合、システムコマンド chown を使用して、これらのディレクトリの所有者を変更してください。

7. **キューマネージャの自動起動を無効化** ：LifeKeeper で保護するキューマネージャでは自動起動を無効化する必要があります。 systemctl コマンドを使って無効化してください。

8. **サーバー接続チャネルの認証** ：WebSphere MQ バージョン 7.1 以降では、チャネルの認証方法が変更されました。デフォルトでは、MQADMIN ユーザー (mqm) は匿名認証 (パスワードなし) を行うことはできないため、リソース階層の作成に失敗します (WebSphere MQ 7.1 以前のリリースで作成されたキューマネージャの認証は引き続き有効)。WebSphere MQ 7.1 以降、MQADMIN ユーザーによる認証を許可するには、チャネルの認証を無効にする方法があります。WebSphere MQ 8.0 では、追加の変更が (現在の設定を取得する「display authinfo(system.default.authinfo.idpwos)」を runmqsc コマンドで実行して) system.default.authinfo.idpwosのauthinfoに対して必要となります。「reqdamd」のchckclnt 設定は、「optional」に変更する必要があります。MQADMIN ユーザーによる匿名認証の許可に失敗すると、リソースの作成時に「 **MQCONNX ended with reason code 2035** 」エラーが発生します。チャネルの認証については、WebSphere MQ ドキュメンテーション を参照してください。

9. **MQSeriesSamples、MQSeriesSDK、およびMQSeriesClient パッケージ** ：LifeKeeper は、WebSphere MQ へのクライアント接続を使用して、リスナーとチャネルイニシエータが完全に機能していることを確認します。これは、リモートのキューマネージャおよびクライアントがキューマネージャに接続するための要件です。そのため、MQSeriesClient パッケージが、WebSphere MQ を実行するすべての LifeKeeper クラスタノード上にインストールされている必要があります。また、MQSeriesSDK および MQSeriesSamples パッケージは、クライアント接続テストと PUT/GET テストを実行するためにインストールされている必要があります。

10. **オプションの C コンパイラ** ：オプションの PUT/GET テストを実行するためには、C コンパイラがマシン上にインストールされている必要があります。C コンパイラがない場合、インストール中に警告が発生します。

11. **LifeKeeper のテストキュー** ：WebSphere MQ Recovery Kit は、オプションで PUT/GET テストを実行してキューマネージャの動作を確認します。リカバリキットはすべてのメッセージをこのキューから取得して破棄するため、専用のテストキューが作成されている必要があります。このキューでは、デフォルトのパーシスタンス設定が「**yes**」(DEFPSIST=yes) に設定されている必要があります。キューマネージャを LifeKeeper で保護している場合、「LIFEKEEPER.TESTQUEUE」という名前のテストキューが自動的に作成されます。また、次のコマンドを使用して、キューマネージャを保護する前に手作業でテストキューを作成することもできます。

```
su - MQUSER
```

```
runmqsc QUEUE.MANAGER.NAME

define qlocal(LIFEKEEPER.TESTQUEUE) DEFPSIST(YES) DESCR('LifeKeeper
test queue')
```

注記：デフォルトの「LIFEKEEPER.TESTQUEUE」以外の名前をテストキューに付けるには、このテストキューの名前を設定する必要があります。詳細については、リソースプロパティ設定の編集 を参照してください。

12. **リスナーオブジェクト用の TCP ポート** ：リスナーオブジェクトを runmqsc を介して変更して使用中の TCP ポートを反映します。次のコマンドを使用して、デフォルトリスナーの TCP ポートを変更します。

```
su - MQUSER
runmqsc QUEUE.MANAGER.NAME

alter LISTENER(SYSTEM.DEFAULT.LISTENER.TCP) TRPTYPE(TCP) PORT(1414)
IPADDR(192.168.1.100)
```

注記: リスナーオブジェクトは、デフォルトの MQ リスナー TCP ポート 1414 を使用する場合も変更が必要ですが、特定の IP アドレス (IPADDR) を指定する必要はありません。IPADDR を設定しない場合、リスナーはサーバー上のすべてのインターフェースにバインドされます。IPADDR を設定した場合は、IPADDR で定義されたアドレスを使用して LifeKeeper で仮想 IP リソースを作成することを強く推奨します。そうすることによって、MQ リスナーの起動時に確実に IP アドレスを使用できます。

13. **TCP ポート番号** 。各 WebSphere MQ リスナーは、異なるポート (デフォルトで 1414) を使用するか、すべてのインターフェースにバインドされたリスナーがない状態で異なる仮想 IP にバインドする必要があります。これには、クラスタ内の保護されたキューマネージャと保護されていないキューマネージャが含まれます。

14. ***mqs.ini* で設定されたキューマネージャ** ：アクティブ / アクティブ設定では、各サーバーは、グローバルキューマネージャー設定ファイル *mqs.ini* のコピーをそれぞれ保持しています。保護されたキューマネージャをクラスタ内のすべてのサーバー上で実行するには、キューマネージャがクラスタ内のすべてのサーバー上の *mqs.ini* 設定ファイルに設定されている必要があります。適切な QueueManager: スタンザをプライマリサーバーからコピーして、すべてのバックアップサーバー上の *mqs.ini* 設定ファイルに追加します。

# 6.8.4.1.1. MQ でサポートされるファイルシステムレイアウト

共有ストレージシステムとファイルシステムレイアウトに応じて、4 つの異なる設定がサポートされています。各設定の違いは、ファイルシステムレイアウトにあります。次のセクションでは、サポートされるファイルシステムレイアウトについて説明します。

# 6.8.4.1.1.1. 設定 1 – 共有ストレージ上の /var/mqm

この設定では、 */var/mqm* ディレクトリ全体が、LifeKeeper でサポートされる共有ストレージ (SCSI、SAN、NAS、または複製されたストレージ) にマウントされます。

注記：これは、アクティブ / パッシブ設定でのみ動作します。

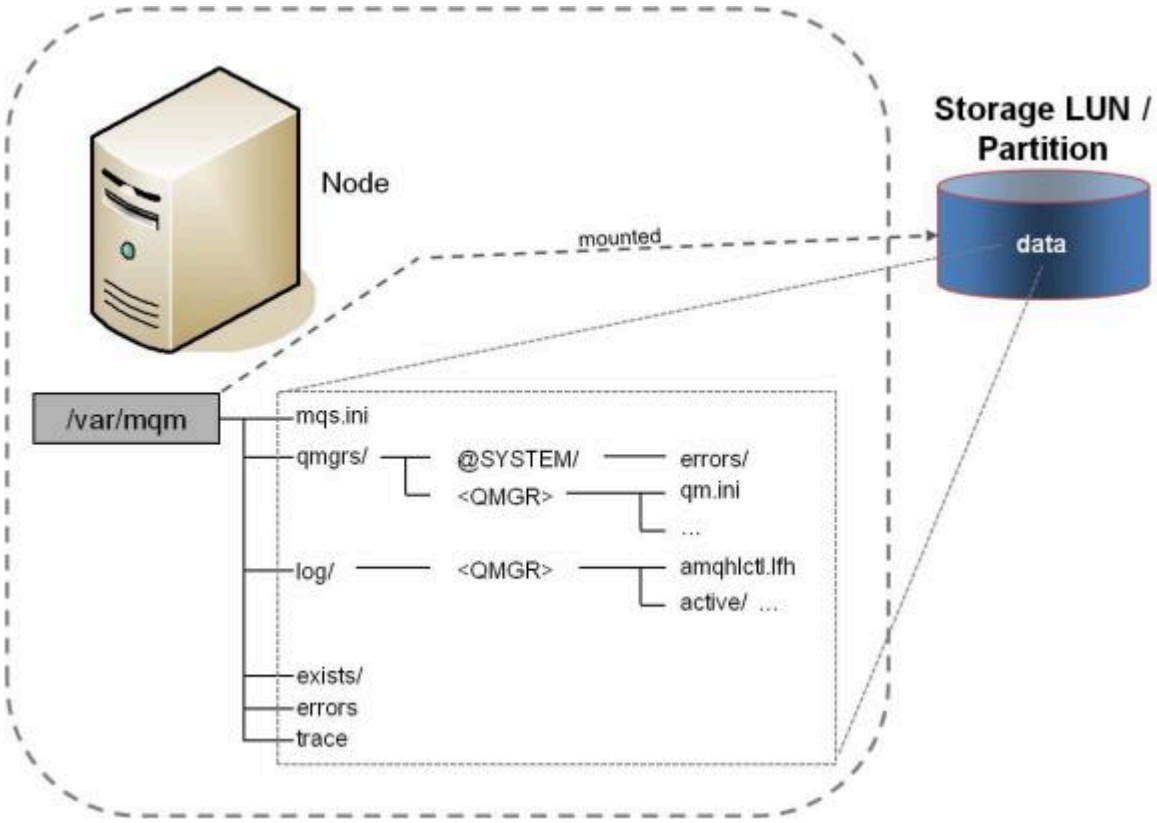



**図 3 -** ファイルシステムレイアウト **1 -** 共有ストレージ上の ***/var/mqm***

# 6.8.4.1.1.2. 設定 2 – 直接マウント

この設定では、*QMDIR* ディレクトリと *QMLOGDIR* ディレクトリは、共有ストレージ上にあります。これには、2 つの専用 LUN またはパーティションを用意するか、各キューマネージャで LVM を使用する必要があります。LVM を使用する場合、同一の LUN から 2 つの論理ボリュームを作成でき、2 つのディレクトリに個別にマウントできます。

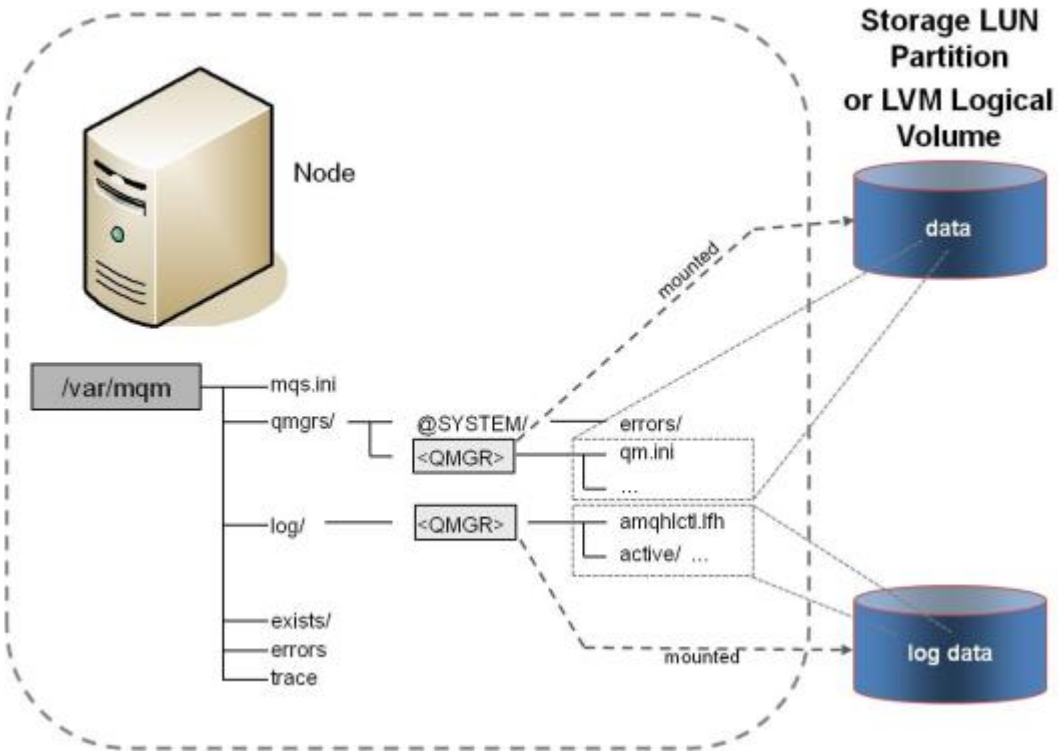



図 4 - ファイルシステムレイアウト 2 - 直接マウント

# 6.8.4.1.1.3. 設定 3 – シンボリックリンク

LVM を使用せず、多数のキューマネージャを使用するアクティブ / アクティブ設定では、シンボリックリンクの使用を推奨します。この場合、1 つまたは複数の専用マウントポイントが作成されます (例: */mq*)。LifeKeeper で保護された 1 つのファイルシステムをそこにマウントし、各キューマネージャ用にサブディレクトリを作成します (例: */mq/QUEUE!MANAGER!NAME/log* および */mq/QUEUE!MANAGER!NAME/qmgrs*)。その後、*QMDIR* ディレクトリおよび *QMLOGDIR* ディレクトリを、この場所にリンクします。

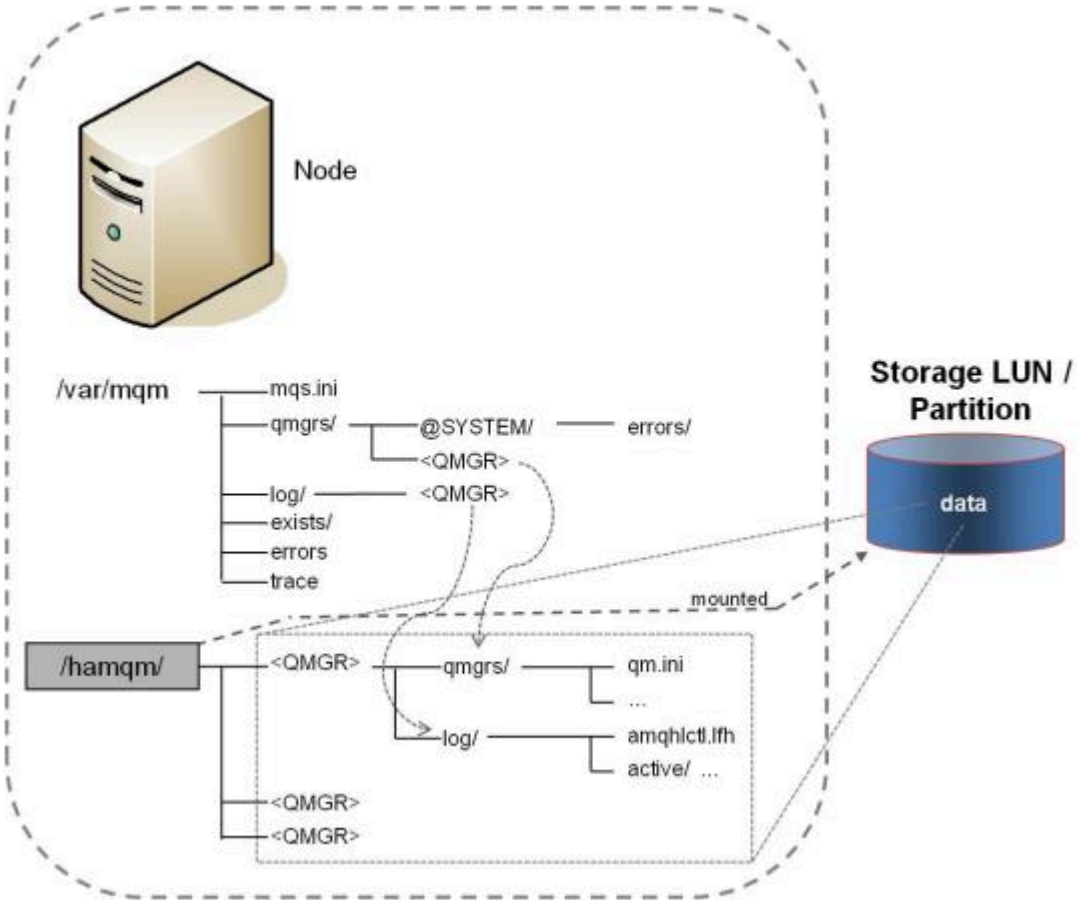



図 5 - ファイルシステムレイアウト 3 - シンボリックリンク

# 6.8.4.1.2. LifeKeeper 使用時の WebSphere MQ の設定

LifeKeeper for Linux WebSphere MQ リソース階層の作成を開始する前に実施すべき WebSphere MQ の設定上の考慮事項は多数あります。これらの変更は、Recovery Kit が PUT/GET テストを実行し、WebSphere MQ パーシステントデータに高可用性を提供するために必要です。WebSphere MQ キューマネージャは、リモートクライアントリクエストを TCP/IP を使って処理するため、仮想 IP リソースを WebSphere MQ リソース階層を作成する前に作成しておく必要があります。次の作業を実行し、LifeKeeper WebSphere MQ リソース階層を作成できるようにします。

1. インストールの計画をします (付録 C を参照)。

   WebSphere MQ をインストールする前に、インストールの計画をする必要があります。これには、MQUSER、MQUSER UID、MQGROUP GID の選択が含まれます。また、使用するファイルシステムレイアウトも決定しなければなりません (サポートされるファイルシステムレイアウト を参照)。このプロセスをスムーズに行うため、SIOS Technology Corp. では、必要なすべての情報を記入できるフォームを提供しています。付録 C – WebSphere MQ 設定シート を参照してください。インストール手順の準備として、このフォームに記入してください。

2. 各サーバでカーネルパラメータを設定します。

   WebSphere MQ では、共有メモリなど、特別な Linux カーネルパラメータ設定を必要とする場合があります。お使いのリリースのWebSphere MQの WebSphere MQ ドキュメンテーション を参照し、WebSphere MQ を実行するための最小要件を確認してください。カーネルパラメータの変更が再起動後も残るようにするためには、*/etc/sysctl.conf* 設定ファイルを使用できます。コマンド sysctl -p を起動スクリプト (boot.local) に追加する必要がある場合があります。SuSE では、insserv boot.sysctl を実行して *sysctl.conf* ファイル内のパラメータの自動設定を有効化できます。

3. 各サーバ上で MQUSER および MQGROUP を作成します。

   オペレーティングシステムコマンド groupadd および adduser を使用して、MQUSER と MQGROUP を作成します。その UID と GID は、手順 1 で使用した『WebSphere MQ 設定シート』に記載されています。

   選択した MQUSER の名前が *mqm* で、UID 1002 を持ち、MQGROUP GID が 1000 の場合、クラスタ内の各サーバ上で次のコマンドを実行できます (実際の設定に合わせて MQUSER、UID、GID の値を変更してください)。

   ```
   groupadd -g 1000 mqm
   useradd -m -u 1002 -g mqm mqm
   ```

   **注記**:これらの設定は、クラスタの全ノードで常に同一の内容である必要があります。NIS または LDAP を実行中の場合、ユーザとグループは一度だけ作成してください。中央ホームディレクトリサーバがない場合、ホームディレクトリを作成する必要がある場合があります。

4. PATH 環境変数（アップグレードの場合のみ) の設定を解除します。

   WebSphere MQ Recover Kit の以前のリリースをアップグレードする場合は、MQUSER PATH 環境変数の設定を変更しなければならない場合があります。Recovery Kit の以前のリリースでは、WebSphere MQ ソフトウェアのデフォルトのインストールの場所 (/opt/mqm) を含むように MQUSER PATH 環境変数を変更する必要がありました。この変更を以前のリリースで行った場合、Recover Kit が正常に動作するには、本バージョンでこの設定を解除する必要があります。

5. 必要なパッケージをインストールして、WebSphere MQ を各サーバ上にインストールします。

   MQSeries インストールでは、ライセンスのアクティベーション (`mqlicense_lnx.sh`) のために X11 ライブラリと Java が必要です。必要なソフトウェアパッケージをインストールします。

6. WebSphere MQ ソフトウェアと WebSphere MQ の Fix Pack を各サーバにインストールします。

   お使いの WebSphere MQ リリースの『WebSphere MQ ドキュメンテーション』に記載されている手順に従ってください。

7. **サーバ接続チャネルの認証。** WebSphere MQ バージョン 7.1 以降では、チャネルの認証方法が変更されました。デフォルトでは、MQADMIN ユーザー (mqm) は匿名認証 (パスワードなし) を行うことはできないため、リソース階層の作成に失敗します (WebSphere MQ 7.1 以前のリリースで作成されたキューマネージャの認証は引き続き有効)。WebSphere MQ 7.1 以降、MQADMIN ユーザーによる認証を許可するには、チャネルの認証を無効にする方法があります。WebSphere MQ 8.0 では、追加の変更が (現在の設定を取得する「display authinfo(system.default.authinfo.idpwos)」を runmqsc コマンドで実行して) system.default.authinfo.idpwosのauthinfoに対して必要となります。「reqdamd」のchckclnt 設定は、「optional」に変更する必要があります。MQADMIN ユーザーによる匿名認証の許可に失敗すると、リソースの作成時に「 **MQCONNX ended with reason code 2035** 」エラーが発生します。チャネルの認証については、WebSphere MQ ドキュメンテーション を参照してください。

8. MQ バージョン 7.1 以降を使用している場合、Queue Managerが使用されている指定チャネルで MQADMIN ユーザを有効化してください。

9. LifeKeeper と WebSphere MQ Recovery Kit を各サーバにインストールします。

   SPS をインストールする方法については、SIOS Protection Suite インストールガイド を参照してください。

10. 共有ストレージを準備し、その共有ストレージをマウントします。

    サポートされるファイルシステムレイアウトについては、サポートされるファイルシステムレイアウト を参照してください。ファイルシステムレイアウトとストレージタイプに応じて、ボリュームグループの作成、論理ボリューム、ファイルシステムの作成、または NFS 共有のマウントが必要です。

    以下は、NAS ストレージを使用するファイルシステムレイアウト 2 の例を示しています。

    ```
    node1:/var/mqm/qmgrs # mkdir TEST\!QM
    ```

```
node1:/var/mqm/qmgrs # mkdir ../log/TEST\!QM

node1:/var/mqm/qmgrs # mount 192.168.1.30:/raid5/vmware/shared_NFS/
TEST.QM/qmgrs ./TEST\!QM/

node1:/var/mqm/qmgrs # mount 192.168.1.30:/raid5/vmware/shared_NFS/
TEST.QM/log ../log/TEST\!QM/
```

11. *QMDIR* と *QMLOGDIR* の所有者とグループを MQUSER と MQGROUP に設定します。

    *QMDIR* と *QMLOGDIR* は、MQUSER と MQGROUP に所有されている必要があります。次のコマンドを使用して、ファイルシステム権限を設定してください。

    ```
    chown MQUSER QMDIR
    chgrp mqm QMDIR
    chown MQUSER QMLOGDIR
    chgrp mqm QMLOGDIR
    ```

    MQUSER、QMDIR、QMLOGDIR の値は、お使いのファイルシステムレイアウトと MQUSER のユーザ名によって異なります。手順 1 のシートを使用して、正しい値を決定してください。

    以下は、MQUSER mqm、およびキューマネージャ TEST.QM で、デフォルトの QMDIR および QMLOGDIR を使用する場合の例を示しています。

    ```
    node1:/var/mqm/qmgrs # chown mqm TEST\!QM/
    node1:/var/mqm/qmgrs # chgrp mqm TEST\!QM/
    node1:/var/mqm/qmgrs # chown mqm ../log/TEST\!QM/
    node1:/var/mqm/qmgrs # chgrp mqm ../log/TEST\!QM/
    ```

12. キューマネージャをプライマリサーバ上に作成します。

    『WebSphere MQ ドキュメント』に記載されている手順に従って、使用されている WebSphere MQ のバージョンに応じてキューマネージャを作成します。

    以下は、MQUSER mqm とキューマネージャ TEST.QM の例を示しています。

    ```
    node1:/var/mqm/qmgrs # su - mqm
    mqm@node1:~> crtmqm TEST.QM
    WebSphere MQ queue manager created.
    Creating or replacing default objects for TEST.QM.
    Default objects statistics : 31 created. 0 replaced. 0 failed.
    Completing setup.
    Setup completed.
    ```

    **注記**: すでに存在するキューマネージャを保護する場合、次の手順に従って、キューマネージャを共有ストレージに移動してください。

a. キューマネージャを停止します (`endmqm -i QUEUE.MGR.NAME`)。

b. キューマネージャディレクトリおよびキューマネージャログディレクトリの中身を手順 10 で作成した共有ストレージにコピーします。

c. グローバル設定ファイル (*mqs.ini*) およびキューマネージャ設定ファイル (*qm.ini*) を必要に応じて変更し、`QMDIR` と `QMLOGDIR` の新しい場所を設定します。

d. キューマネージャを起動して、機能を確認します (`strmqm QUEUE.MGR.NAME`)。

e. キューマネージャを停止します (`endmqm -i QUEUE.MGR.NAME`)。

13. オプション: 仮想 IP リソースをプライマリサーバ上の LifeKeeper で設定します。

SPS for Linux IP Recovery Kit 管理ガイド および SIOS Protection Suite インストールガイド に記載の手順とガイドラインに従ってください。

注記：サーバ接続によってのみキューマネージャにアクセスする場合、LifeKeeper 仮想 IP を設定する必要はありません。

14. リスナーオブジェクトをお使いの TCP アドレスおよびポートに合わせて変更します。

```
su - MQUSER
runmqsc QUEUE.MANAGER.NAME

alter LISTENER(SYSTEM.DEFAULT.LISTENER.TCP) TRPTYPE(TCP) PORT(1414)
IPADDR(192.168.1.100)
```

注記 :手順 13 で使用した IP アドレスと同じアドレスを使用して `IPADDR` の値を設定します。`IPADDR` は、WebSphere MQ がすべてのアドレスをバインドするよう設定してはなりません。

15. プライマリサーバ上のキューマネージャを起動します。

プライマリサーバ上で、キューマネージャ、コマンドサーバ (手動起動に設定されている場合)、リスナーを起動します。

```
su - MQUSER
strmqm QUEUE.MANAGER.NAME
strmqcsv QUEUE.MANAGER.NAME
runmqlsr -m QUEUE.MANAGER.NAME -t TCP &
```

16. キューマネージャが正常に起動したことを確認します。

```
su - MQUSER
echo 'display qlocal(*)' | runmqsc QUEUE.MANAGER.NAME
```

17. キューマネージャのスタンザを、バックアップサーバ上のグローバルキューマネージャ設定ファイル

*mqs.ini* に追加します。

**注記**: この手順は、ファイルシステムレイアウト 2 と 3 で必要です。

18. **オプション:** LifeKeeper テストキューを、プライマリサーバ上に作成します。

```
runmqsc TEST.QM

5724-B41 © Copyright IBM Corp. 1994, 2002. ALL RIGHTS RESERVED.

Starting MQSC for queue manager TEST.QM.

define qlocal(LIFEKEEPER.TESTQUEUE) defpsist(yes) descr('LifeKeeper
test queue')

1 : define qlocal(LIFEKEEPER.TESTQUEUE) defpsist(yes)
descr('LifeKeeper test queue')

AMQ8006: WebSphere MQ queue created.
```

19. LifeKeeper によってコマンドサーバを起動するには、プライマリサーバ上で、次のコマンドを使用してコマンドサーバの自動起動を無効にします。この手順を行わない場合、キューマネージャの起動時にコマンドサーバが自動的に起動されます。

```
su - MQUSER
runmqsc TEST.QM
ALTER QMGR SCMDSERV(MANUAL)
```

20. キューマネージャリソース階層をプライマリサーバ上に作成します。

    詳細については、 LifeKeeper の設定作業 を参照してください。

21. キューマネージャリソース階層をバックアップシステムに拡張します。

    詳細については、 LifeKeeper の設定作業 を参照してください。

21. 設定をテストします。

    HA WebSphere MQ インストールをテストするには、WebSphere MQ リソース階層のテスト に記載されている手順に従ってください。

# 6.8.4.1.3. MQ リソース作成後の設定変更

LifeKeeper WebSphere MQ Recovery Kit は、WebSphere MQ コマンドを使用してキューマネージャーを起動および停止します。以下に示すのは、このルールに対するいくつかの例外です。

---

QMDIR と QMLOGDIR の再配置

リスナーポートの変更

キューマネージャーの IP の変更

# 6.8.4.1.3.1. QMDIR と QMLOGDIR の再配置

*QMDIR* と *QMLOGDIR* の場所が変わった場合、LifeKeeper の設定を変更する必要があります。設定は、次のいずれかの方法で変更できます。

1. キューマネージャのリソース階層を再作成する。

   この方法を行うには、キューマネージャ階層の削除と、キューマネージャ階層の作成が必要です。詳細については、「WebSphere MQ リソース階層の削除 」および「WebSphere MQ リソース階層の作成 」を参照してください。

2. ファイルシステム階層を手作業で新規作成し、その新しいファイルシステム階層を WebSphere MQ 階層に追加する。古いファイルシステム階層を WebSphere MQ 階層から削除し、古いファイルシステム階層を削除します。ファイルシステム階層を作成および削除する方法については、LifeKeeper インストールガイド を参照してください。

# 6.8.4.1.3.2. リスナーポートの変更

キューマネージャのリスナーポートを変更するには、以下の手順を実行してください。

runmqsc 内のリスナーオブジェクトを変更した後、リスナーを停止し、起動してください。

```
su - MQUSER
runmqsc QUEUE.MANAGER.NAME

alter LISTENER(SYSTEM.DEFAULT.LISTENER.TCP) TRPTYPE(TCP) PORT(1415)
stop LISTENER(SYSTEM.DEFAULT.LISTENER.TCP)
start LISTENER(SYSTEM.DEFAULT.LISTENER.TCP)
```

詳細については、リソースプロパティ設定の編集 を参照してください。

# 6.8.4.1.3.3. キューマネージャーの IP の変更

WebSphere MQ キューマネージャーに関連する LifeKeeper で保護されている IP を変更するには、次の手順を実行します。

1. LifeKeeper GUI で新しい LifeKeeper 仮想 IP を作成します。
2. 新しい仮想 IP を WebSphere MQ 階層に追加します。
3. 古い仮想 IP を WebSphere MQ 階層から削除します。
4. 古い仮想 IP リソースを削除します。
5. 必要な場合、 runmqsc 内のリスナーオブジェクトを変更し、リスナーを再起動します。

```
su - MQUSER
runmqsc QUEUE.MANAGER.NAME

alter LISTENER(SYSTEM.DEFAULT.LISTENER.TCP) TRPTYPE(TCP) PORT(1414)
IPADDR(192.168.1.101)
stop LISTENER(SYSTEM.DEFAULT.LISTENER.TCP)
start LISTENER(SYSTEM.DEFAULT.LISTENER.TCP)
```

別の方法として、LifeKeeper の lk_chg_value 機能を使用して、IP を変更できます。詳細については、lk_chg_value(8) マニュアルページを参照してください。

# 6.8.4.1.4. WebSphere MQ の設定例

本セクションでは、一般的な WebSphere MQ 設定の定義と例を示します。各例には、LifeKeeper に適用される設定ファイルエントリが含まれています。

# 6.8.4.1.4.1. 共有ストレージ上の /var/mqm によるアクティブ / スタンバイ設定

アクティブ / スタンバイ設定では、Node1 はプライマリ LifeKeeper サーバです。このノードは、WebSphere MQ キューマネージャを保護します。すべてのストレージは、クラスタサーバ間の共有アレイに置かれます。Node2 は他のアプリケーションやサービスを処理することができますが、LifeKeeper から見た場合は、WebSphere MQ リソースのバックアップとしてのみ機能します。ディレクトリ */var/mqm* は、共有ストレージ上にあります。プライマリサーバは、処理可能な数のキューマネージャを上限なしで実行できます。

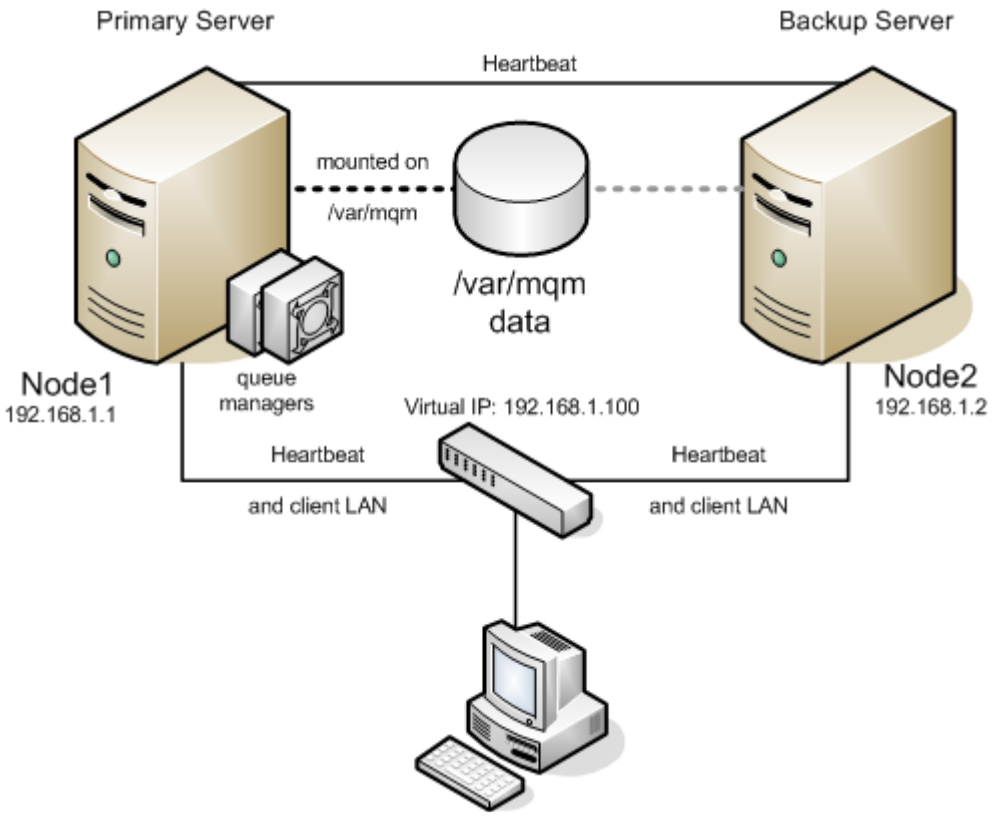



図 6 ローカルストレージにおけるアクティブ / スタンバイ設定

## 設定に関する注記

- クライアントは、LifeKeeper で保護され、クラスタ内のサーバ間で浮動するよう指定された IP 192.168.1.100 を使用して、WebSphere MQ サーバに接続します。
- ディレクトリ */var/mqm* は、共有ストレージ上にあります。
- 各キューマネージャにおいて、リスナーオブジェクトは一意のポート番号を含むよう変更されました。

# 6.8.4.1.4.2. NAS ストレージにおけるアクティブ / スタンバイ設定

アクティブ / スタンバイ設定では、Node1 はプライマリ LifeKeeper サーバです。このノードは、WebSphere MQ キューマネージャを保護します。すべてのストレージは、IP 10.0.0.100 の NAS サーバ上にあります。Node2 は他のアプリケーションやサービスを処理することができますが、LifeKeeper から見た場合は、WebSphere MQ リソースのバックアップとしてのみ機能します。ディレクトリ */var/mqm* は、NAS サーバの IP 10.0.0.100 から配置され、アクティブサーバのみにマウントされます。プライマリサーバは、処理可能な数のキューマネージャを上限なしで実行できます。

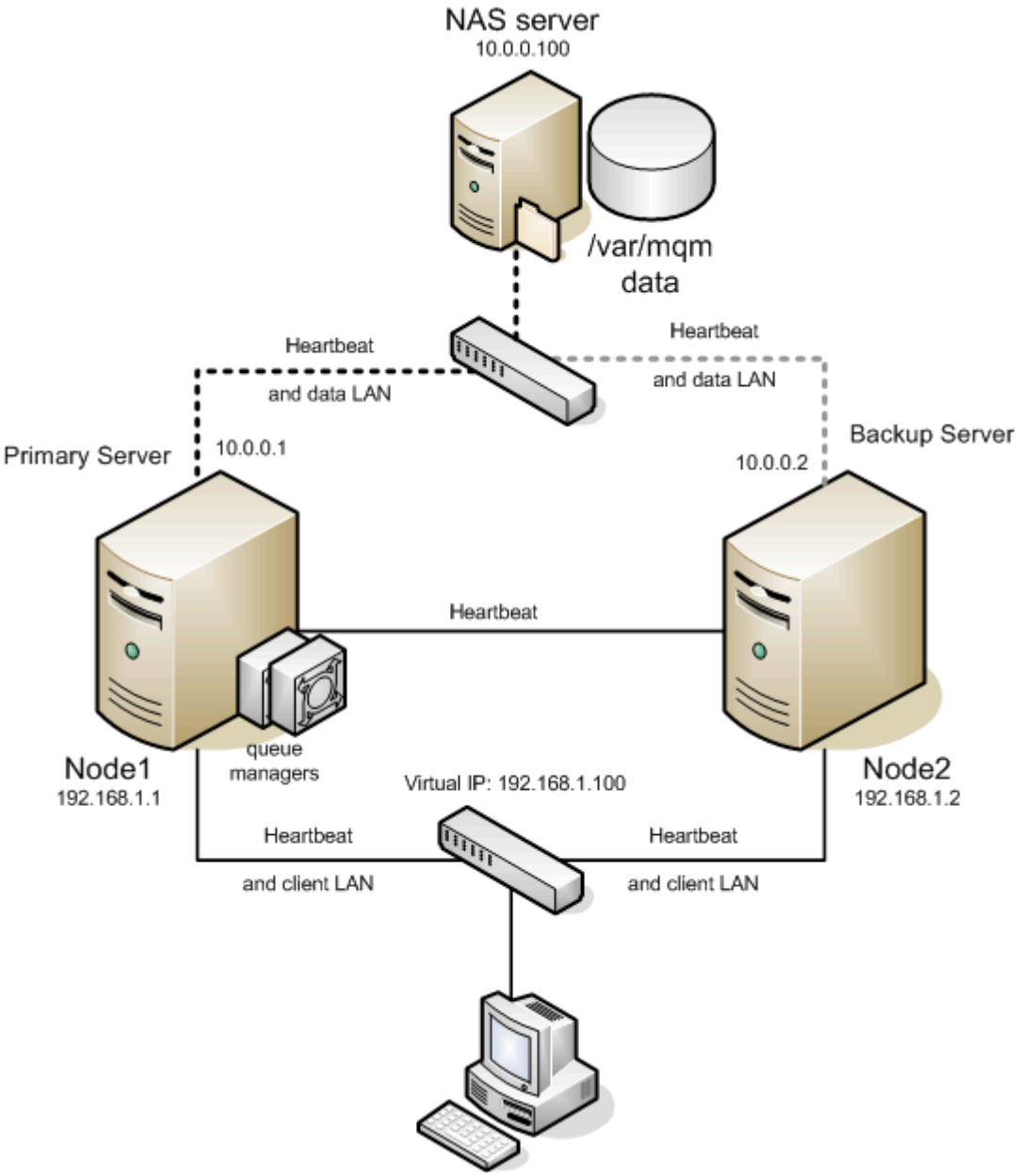



図 7 NFS ストレージにおけるアクティブ / スタンバイ設定

## 設定に関する注記

- クライアントは、LifeKeeper で保護され、クラスタ内のサーバ間で浮動するよう指定された IP 192.168.1.100 を使用して、WebSphere MQ サーバに接続します。
- ディレクトリ */var/mqm* は、NAS サーバ上にあります。
- アクティブサーバは、専用のネットワークインターフェースを使用して、ディレクトリ */var/mqm* を IP 10.0.0.100 を持つ NAS サーバからマウントします。
- 各ネットワークインターフェースには、ハートビートが設定されています。
- 各キューマネージャにおいて、リスナーオブジェクトは一意のポート番号を含むよう変更されました。

# 6.8.4.1.4.3. 共有ストレージにおけるアクティブ / アクティブ設定

以下のアクティブ / アクティブ設定において、Node1 と Node2 の両方が WebSphere MQ リソースに対するプライマリ LifeKeeper サーバです。各サーバは、互いにとってのバックアップサーバでもあります。この例では、Node1 はキューマネージャ QMGR1 のための共有ストレージアレイを保護します。Node2 はキューマネージャ QMGR2 のための共有ストレージアレイをプライマリサーバとして保護します。さらに、各サーバは互いのためのバックアップサーバとしても動作します。この例では、Node2 は Node1 上のキューマネージャ QMGR1 のためのバックアップであり、Node1 は、Node2 上のキューマネージャ QMGR2 のためのバックアップです。

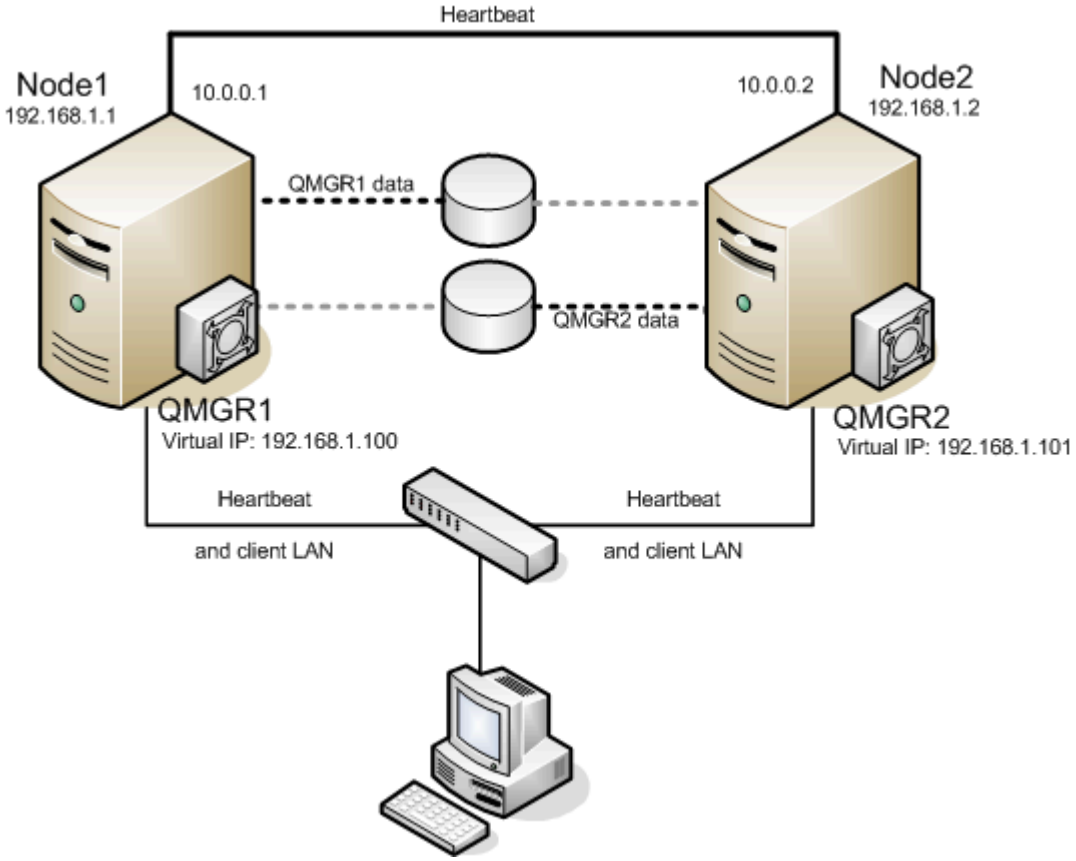



図 8 共有ストレージにおけるアクティブ / アクティブ設定

## 設定に関する注記

- クライアントは、LifeKeeper 仮想 IP 192.168.1.100 を使用して、キューマネージャ QMGR1 に接続します。
- クライアントは、LifeKeeper 仮想 IP 192.168.1.101 を使用して、キューマネージャ QMGR2 に接続します。
- 各ネットワークインターフェースには、ハートビートが設定されています。
- 各キューマネージャにおいて、リスナーオブジェクトは一意のポート番号を含むよう変更されまし

た。

- QMGR1 データは 2 つの論理ボリュームが設定された共有ストレージ上の 1 つのボリュームグループにあります。各論理ボリュームには、QMDIR または QMLOGDIR にマウントされるファイルシステムが含まれます。

- QMGR2 データは 2 つの論理ボリュームが設定された共有ストレージ上の 2 次ボリュームグループにあります。各論理ボリュームには、QMDIR または QMLOGDIR にマウントされるファイルシステムが含まれます。

# 6.8.4.1.4.4. NAS ストレージにおけるアクティブ / アクティブ設定

以下のアクティブ / アクティブ設定において、Node1 と Node2 の両方が WebSphere MQ リソースに対するプライマリ LifeKeeper サーバです。各サーバは、互いにとってのバックアップサーバでもあります。この例では、Node1 はキューマネージャ QMGR1 のための NFS マウントを保護します。Node2 はキューマネージャ QMGR2 のための NFS マウントをプライマリサーバとして保護します。さらに、各サーバは互いのためのバックアップサーバとしても動作します。この例では、Node2 は Node1 上のキューマネージャ QMGR1 のためのバックアップであり、Node1 は、Node2 上のキューマネージャ QMGR2 のためのバックアップです。

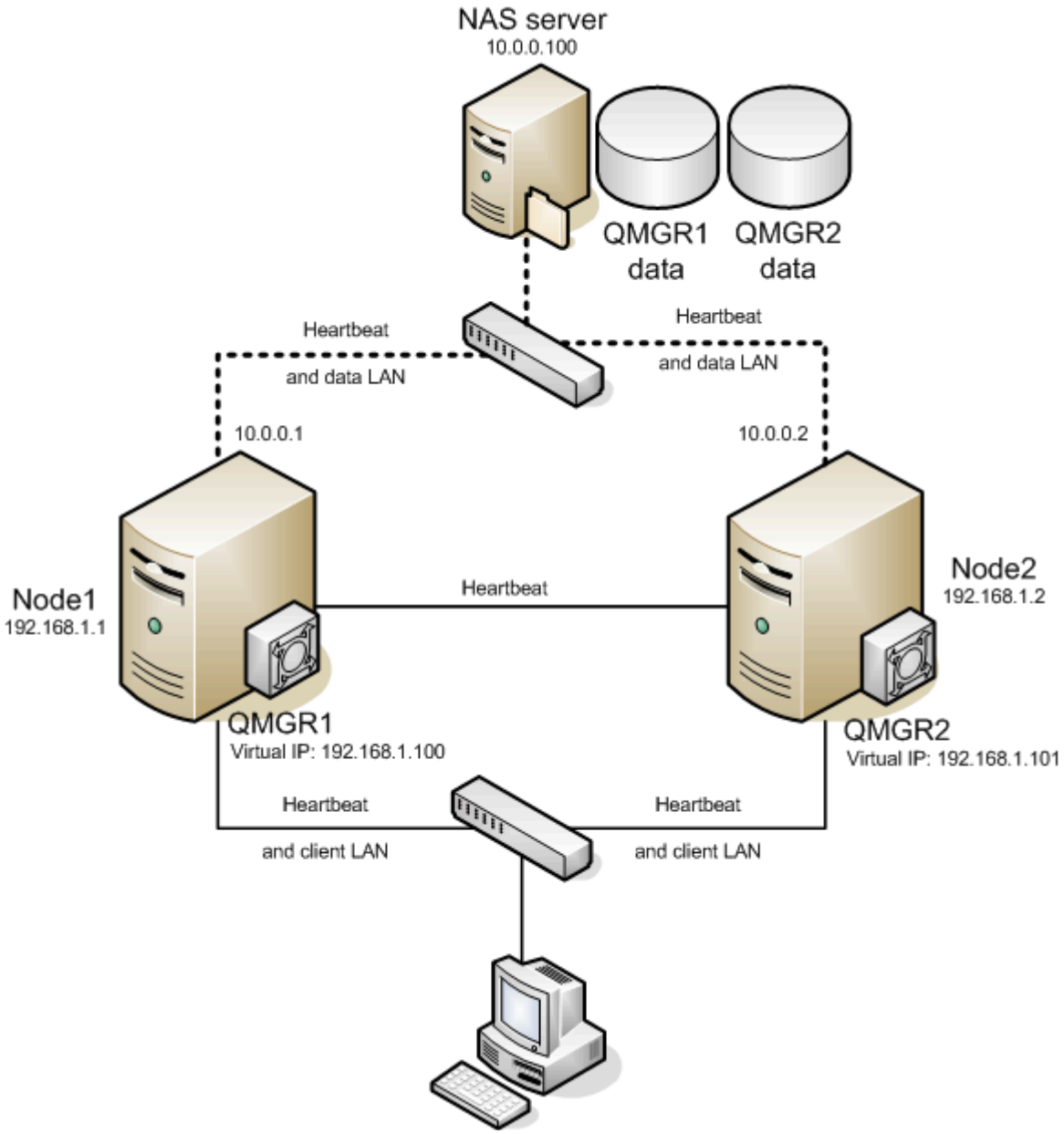



図 9 NFS ストレージにおけるアクティブ / アクティブ設定

## 設定に関する注記

- クライアントは、LifeKeeper 仮想 IP 192.168.1.100 を使用して、キューマネージャ QMGR1 に接続します。
- クライアントは、LifeKeeper 仮想 IP 192.168.1.101 を使用して、キューマネージャ QMGR2 に接続します。
- 各サーバは、NAS サーバにアクセスするために専用のネットワークインターフェースを持っています。
- 各ネットワークインターフェースには、ハートビートが設定されています。
- 各キューマネージャにおいて、リスナーオブジェクトは一意のポート番号を含むよう変更されました。
- QMGR1 データは、NAS サーバ上の 2 つの NFS エクスポートに配置されています。これらのエクスポートは、*QMDIR* または *QMLOGDIR* にマウントされています。NAS サーバの IP は、10.0.0.100. です。
- QMGR2 データは、NAS サーバ上の 2 つの NFS エクスポートに配置されています。これらのエクスポートは、*QMDIR* または *QMLOGDIR* にマウントされています。NAS サーバの IP は、10.0.0.100. です。

# 6.8.5. MQ用 LifeKeeper の設定作業

LifeKeeper for Linux WebSphere MQ Recovery Kit の管理作業はすべて、LifeKeeper のグラフィカルユーザインターフェース (GUI) で実行できます。LifeKeeper の GUI では、WebSphere のリソースの設定、管理、監視の作業をガイド付きで行うことができます。

## 概要

以下の作業は、WebSphere MQ リソースインスタンス特有のものであり、Recovery Kit ごとに異なるため、本書で説明しています。

- WebSphere リソース階層の作成 - WebSphere MQ リソース階層を作成します。
- WebSphere リソース階層の削除 - WebSphere MQ リソース階層を削除します。
- WebSphere リソース階層の拡張 - WebSphere MQ リソース階層をプライマリサーバからバックアップサーバに拡張します。
- WebSphere リソース階層の拡張解除 - LifeKeeper クラスタ内にある 1 台のサーバの WebSphere MQ リソース階層を拡張解除（削除）します。
- MQ リソースプロパティ設定の編集 – WebSphere MQ リソース階層の作成後に、LifeKeeper テストキュー、リスナー管理、停止タイムアウトなどの WebSphere MQ リソースパラメータを再設定します。

次の作業は、すべてのリカバリキットに共通する手順のため、LifeKeeper for Linux テクニカルドキュメンテーション の管理 セクションに記載されています。

- リソース依存関係の作成 ：既存のリソース階層と別のリソースインスタンスとの間に親子の依存関係を作成し、クラスタ内のすべての対象サーバに依存関係の変更を反映します。
- リソース依存関係の削除 ：リソースの依存関係を削除して、クラスタ内のすべての対象サーバに依存関係の変更を反映します。
- In Service ：リソース階層を特定のサーバで in service にします。
- Out of Service ：リソース階層を特定のサーバで out of service の状態にします。
- プロパティの表示 / 編集 。特定のサーバでリソース階層のプロパティを表示または編集します。

**注記:** このセクションの残りの部分では、設定作業はすべて **[Edit]** メニューから実行されます。これらの設定作業の大半は次の方法でも実行できます。

- ツールバーから実行
- ステータス表示の左ペインのグローバルリソースを右クリックして実行

- ステータス表示の右ペインのリソースインスタンスを右クリックして実行

右クリックを使用すると、 **[Edit]** メニューを使用するときに要求される情報を入力せずにすみます。

# 6.8.5.1. WebSphere MQ リソース階層の作成

必要な設定作業を完了した後、次の手順に従って WebSphere MQ リソース階層を作成します。

1. LifeKeeper GUI メニューから **[Edit]** を選択し、次に **[Server]** を選択します。表示されたメニューから、 **[Create Resource Hierarchy]** を選択します。

   ***Create Resource Wizard*** ダイアログボックスが表示され、クラスタ内にインストールされ認識されているすべての Recovery Kit が、ドロップダウンリストボックスに一覧で示されます。

2. **[IBM WebSphereMQ]** を選択し、 **[Next]** をクリックします。

3. 以下の情報を入力するようにプロンプトが表示されます。ダイアログボックスで **[Back]** ボタンが有効な場合は、前のダイアログボックスに戻ることができます。これは、エラーが発生して、前に入力した情報を修正する必要がある場合に便利な機能です。いつでも **[Cancel]** をクリックして、作成処理全体を取り消すことができます。

| フィールド | ヒント |
|---|---|
| スイッチバックタイプ | **[Intelligent]** または **[Automatic]** を選択します。バックアップサーバへのフェイルオーバー後にサーバが復帰したときに、このサーバに WebSphere MQ インスタンスを戻す方法を指定します。スイッチバックタイプは、必要な場合、 **[Resource Properties]** ダイアログボックスの **[General]** タブで後から変更できます。<br><br>**注記:** スイッチバック方法は、WebSphere MQ リソースが使用する IP またはファイルシステムリソースのスイッチバック方法と一致する必要があります。この設定が WebSphere MQ リソースの設定と一致しない場合、作成処理によって、WebSphere MQ リソースの設定と一致するよう設定がリセットされます。 |
| Server | 階層を作成したい **サーバ** を選択します。 |
| Queue Manager Name | 保護したい WebSphere MQ キューマネージャを選択します。キューマネージャは、リソース階層を作成する前に作成しておく必要があります。すでに LifeKeeper によって保護されているキューマネージャはこの一覧から除外されます。キューマネージャは、グローバル設定ファイル *mqs.ini* から取得されます。 |
| Manage Listener | **[YES]** を選択すると、WebSphere MQ キューマネージャリスナーを保護および管理します。 **[NO]** を選択すると、LifeKeeper は WebSphere MQ リスナーを管理しません。<br><br>**注記:** この設定は後から変更できます。詳細については、リソースプロパティ設定の編集 を参照してください。 |
| Server Connection Channel | 接続テストに使用するサーバ接続チャネルを選択してください。デフォルトでは、チャネル SYSTEM.DEF.SVRCONN が使用されますが、MQ バージョン 7.1 以降では、MQ の チャネル認証を変更するには、デフォルト以外のチャネルが使用され、MQADMIN ユーザーが指定のチャネルで有効になっていることが必要になりました。<br><br>**注記:** リソースを作成する前に、サーバ接続チャネルが作成済みであることを確認して |

| | |
|---|---|
| | ください。詳細については、LifeKeeper 使用時の WebSphere MQ の設定 を参照してください。<br><br>**注記:** この設定は、後から変更できます。詳細については、リソースプロパティ設定の編集 を参照してください。 |
| Virtual IP | 階層に含める LifeKeeper の仮想 IP リソースを選択します。LifeKeeper 仮想 IP を WebSphere MQ 階層に含めない場合は、 **[None]** を選択してください。<br><br>**注記:** 仮想 IP が選択リストに表示されるには、プライマリノード上で ISP (アクティブ) である必要があります。 |
| IBM WebSphere MQ Resource Tag | LifeKeeper が提供するデフォルトのルートタグを選択するか、このサーバ上のリソースインスタンスに独自の名前を入力します。デフォルトは、キューマネージャの名前です。アルファベット、数字、および次の特別な文字を使用できます。- _ ./ |

4. **[Create]** をクリックします。 ***[Create Resource Wizard]*** が表示され、WebSphere MQ リソース階層が作成されます。LifeKeeper によって入力データが検査されます。問題が検出された場合、情報ボックスにエラーメッセージが表示されます。

5. 情報ボックスが表示され、WebSphere MQ リソース階層が正常に作成されたことと、フェイルオーバ保護を提供するには階層をクラスタ内の別のサーバに拡張する必要があることが示されます。 **[Next]** をクリックしてください。

6. **[Continue]** をクリックします。次に、LifeKeeper は、 ***[Pre-Extend Wizard]*** を開始します。リソース階層を別のサーバに拡張する方法の詳細については、 WebSphere MQ 階層の拡張 の手順 2 を参照してください。

# 6.8.5.2. WebSphere MQ 階層の拡張

この操作は **[Edit]** メニューから開始できます。または **[Create Resource Hierarchy]** オプションの動作が完了すると自動的に開始されます。その場合は、手順 2 を参照してください。

1. **[Edit]** メニューの **[Resource]** から **[Extend Resource Hierarchy]** を選択します。 **[Pre-Extend Wizard]** が表示されます。 **拡張** 操作に慣れていない場合は、 **[Next]** をクリックしてください。 **LifeKeeper** の **[Extend Resource Hierarchy]** のデフォルト値が分かっていて、入力と確認を省略する場合は **[Accept Defaults]** をクリックしてください。

2. ***[Pre-Extend Wizard]*** に以下の情報を入力します。

   注記: 最初の 2 つのフィールドは **[Edit]** メニューから **拡張** を開始した場合にのみ表示されます。

| フィールド | ヒント |
|---|---|
| Template Server | WebSphere MQ リソースが現在 in service になっているサーバを入力します。 |
| Tag to Extend | 拡張する WebSphere MQ リソースを選択します。 |
| Target Server | 拡張先のサーバを入力するか、選択します。 |
| Switchback Type | **[Intelligent]** または **[Automatic]** のいずれかを選択します。スイッチバックタイプは、必要な場合 **[Resource Properties]** ダイアログボックスの **[General]** タブで後から変更できます。<br><br>**注記:** スイッチバック方法は、WebSphere MQ リソースが使用する依存リソースのスイッチバック方法と一致する必要があることに注意してください。 |
| Template Priority | テンプレート階層の優先順位を選択または入力します。優先順位は、1 ~ 999 の範囲で未使用の値が有効で、小さい数字ほど優先順位が高くなります(数字 1 が最高の優先順位に相当します)。拡張処理時に、別のシステムですでに使用中の優先順位をこの階層に対して指定することはできません。デフォルト値を推奨します。<br><br>**注記:** このフィールドは、階層をはじめて拡張するときにだけ表示されます。 |
| Target Priority | ターゲットサーバの階層の優先順位を選択または入力します。 |
| Queue Manager Name | この情報フィールドには、拡張されるキューマネージャの名前が表示されます。この値を変更することはできません。 |
| Root Tag | LifeKeeper は、ターゲットサーバ上の新しい WebSphere MQ リソースインスタンスに対して、デフォルトのタグ名を提供します。デフォルトのタグ名は、テンプレートサーバ上のこのリソースのタグ名と同一です。新しい名前を入力する |

| | |
|---|---|
| | 場合、ターゲットサーバ上で一意であることを必ず確認してください。アルファベット、数字、および以下の特別な記号を使用できます。- _ ./ |
| **注記:** リスナー管理、LifeKeeper テストキューの名前、シャットダウンタイムアウト値などのキューマネージャ管理パラメータはすべて、テンプレートサーバから取得されます。 | |

3. 拡張前のチェックが正常に終了したというメッセージが表示されたら、 **[Next]** をクリックします。

4. 拡張する階層に応じて、拡張されるリソースタグ（編集不可）を示す一連の情報ボックスが表示されます。 **[Extend]** をクリックします。

5. 「**Hierarchy extend operations completed (階層の拡張操作が完了しました)**」というメッセージが表示されたら、階層を別のサーバに拡張する場合は、 **[Next Server]** をクリックし、拡張作業が必要ない場合は、 **[Finish]** をクリックします。

6. 「**Hierarchy Verification Finished (階層の検査が完了しました)**」というメッセージが表示されたら、 **[Done]** をクリックします。

# 6.8.5.3. WebSphere MQ 階層の拡張解除

LifeKeeper クラスタ内にある 1 台のサーバからリソース階層を削除するには、次の手順を実行します。

1. **[Edit]** メニューの **[Resource]** から **[Unextend Resource Hierarchy]** を選択します。

2. **[Target Server]** で、WebSphere MQ リソースを拡張解除するターゲットサーバを選択します。WebSphere MQ リソースが現在 in service になっているサーバは選択できません。(このダイアログボックスは、右側のペインでリソースインスタンスを右クリックして、 **[Unextend]** 作業を選択した場合には、表示されません。) **[Next]** をクリックします。

3. 拡張解除する WebSphere MQ 階層を選択し、 **[Next]** をクリックします。 (このダイアログは、いずれかのペインでリソースインスタンスを右クリックして、 **[Unextend]** を選択した場合には表示されません。)

4. 拡張解除のために選択したターゲットサーバと WebSphere MQ リソース階層を確認する情報ボックスが表示されます。 **[Unextend]** をクリックします。

5. WebSphere MQ リソースが正常に拡張解除されたことを示す別の情報ボックスが表示されます。 **[Done]** をクリックして、 **[Unextend Resource Hierarchy]** メニューを終了します。

# 6.8.5.4. WebSphere MQ リソース階層の削除

WebSphere 階層を削除することによる、依存関係と保護対象のサービスへの影響を理解することは重要です。

- **依存関係**：リソース階層を削除する前に、依存関係を削除すると良い場合があります。依存しているファイルシステムは、削除されます。ファイルシステム以外の依存しているリソース (IP や Generic Application など) は、LifeKeeper GUI もしくは WebSphere MQ の削除スクリプトで削除しない限り、削除されません。WebSphere MQ キューマネージャの依存しているファイルシステムが LifeKeeper によって削除されないようにするには、WebSphere MQ 階層を削除する前に、手作業で依存関係を削除してください。

- **保護されるサービス**：WebSphere リソース階層を削除する前に out of service にした場合、そのキューマネージャに対する WebSphere デーモンは停止されます。階層が in service 中に削除された場合、WebSphere MQ デーモンは、階層が削除された後も (LifeKeeper 保護なしで) 実行とサービスの提供を継続します。

LifeKeeper 環境内のすべてのサーバからリソース階層を削除するには、以下の手順を実行します。

1. **[Edit]** メニューの **[Resource]** から **[Delete Resource Hierarchy]** を選択します。

2. **[Target Server]** で WebSphere MQ リソース階層を削除するターゲットサーバを選択し、 **[Next]** をクリックします。(このダイアログボックスは、いずれかのペインでリソースインスタンスを右クリックして [Delete Resource] を選択した場合には、表示されません。)

3. **[Hierarchy to Delete]** を選択します。(このダイアログは、右ペインまたは左ペインのリソースインスタンス上で右クリックして [Delete Resource] を選択した場合には表示されません。) **[Next]** をクリックします。

4. 選択したターゲットサーバと、削除の対象として選択した階層を確認する情報ボックスが表示されます。 **[Delete]** をクリックしてください。

5. WebSphere リソースが正常に削除されたことを確認する別の情報ボックスが表示されます。

6. **[Done]** をクリックして終了します。

# 6.8.5.5. WebSphere MQ リソース階層のテスト

手動スイッチオーバを開始することによって、WebSphere MQ リソース階層をテストできます。これにより、プライマリサーバからバックアップサーバへのリソースインスタンスのフェイルオーバをシミュレートします。

**[Edit]** メニューの **[Resource]** から **[In Service]** を選択します。例えば、バックアップサーバで **In Service** 要求を実行すると、アプリケーション階層はプライマリサーバでは out of service になり、バックアップサーバで in service になります。この時点で、元のバックアップサーバがプライマリサーバに、元のプライマリサーバがバックアップサーバになります。

**Out of Service** 要求を実行すると、アプリケーションは他のサーバで in service にならずに、out of service になります。

## 共有ストレージの設定のテスト

WebSphere MQ の共有ストレージの動作をテストするには、次の手順を行います。

1. プライマリサーバ上に一時的なテストキューを作成します。このとき、パーシスタンスはデフォルトの「**yes**」にします。

```
mqm@node1:/opt/mqm/samp/bin> runmqsc TEST.QM
5724-B41 © Copyright IBM Corp. 1994, 2002. ALL RIGHTS RESERVED.
Starting MQSC for queue manager TEST.QM.

define qlocal(TEST) defpsist(yes)
1 : define qlocal(TEST) defpsist(yes)
AMQ8006: WebSphere MQ queue created.
end

2 : end
One MQSC command read.
No commands have a syntax error.
All valid MQSC commands were processed.
```

2. 次のコマンドで、メッセージをプライマリノード上に作成されたテストキューに追加します。

```
mqm@node1:/opt/mqm/samp/bin> echo “HELLO WORLD on NODE1” | ./amqsput
TEST TEST.QM
Sample AMQSPUT0 start
target queue is TEST
Sample AMQSPUT0 end
```

3. 次のコマンドでテストキューを表示し、メッセージが格納されたことを確認します。

```
mqm@node1:/opt/mqm/samp/bin> ./amqsbcg TEST TEST.QM
```

「HELLO WORLD on NODE1」とその他の出力内容を含むメッセージが表示されます。次の行を探し、Persistence が 1 であることを確認します。

```
[…]
Priority :0 Persistence :1
[…]
```

4. リソース階層をスタンバイノードに切り替えます。

5. キューマネージャが現在アクティブになっているスタンバイサーバ上で、手順 3 を再び実行します。メッセージがスタンバイサーバ上でアクセス可能になります。アクセス可能でない場合、ストレージの設定を確認してください。

6. キューマネージャが現在アクティブになっているスタンバイサーバ上で、テストキューからメッセージを取得します。

```
mqm@node1:/opt/mqm/samp/bin> ./amqsget TEST TEST.QM
Sample AMQSGET0 start
message <HELLO WORLD on NODE1>
<now wait 15 seconds>
no more messages
Sample AMQSGET0 end
```

7. 手順 1 で作成したテストキューを削除します。

```
mqm@node1:/opt/mqm/samp/bin> runmqsc TEST.QM
5724-B41 © Copyright IBM Corp. 1994, 2002. ALL RIGHTS RESERVED.
Starting MQSC for queue manager TEST.QM.

delete qlocal(TEST)
1 : delete qlocal(TEST)
MQ8007: WebSphere MQ queue deleted.
end
2 : end
One MQSC command read.
No commands have a syntax error.
All valid MQSC commands were processed.
```

# 6.8.5.5.1. MQ クライアント接続のテスト

クライアント接続をテストするには、次の手順を行ってください。

1. プライマリサーバ上で、amqsbcgc コマンドを使用してキューマネージャに接続します。

   ```
   export MQSERVER='SYSTEM.DEF.SVRCONN/TCP/192.168.1.90(1414)'
   ```

   注記: IP 192.168.1.90 を LifeKeeper が保護するキューマネージャの仮想 IP に置き換えます。キューマネージャが 1414 以外のポートを使用している場合、1414 を使用されているポート番号に置き換えます。使用されているサーバ接続チャネルがデフォルトの SYSTEM.DEF.SVRCONN チャネルではない場合、SYSTEM.DEF.SVRCONN を使用されているサーバ接続チャネルに置き換えます。

   次の出力が表示されます。

   ```
   mqm@node1:/opt/mqm/samp/bin> ./amqsbcgc LIFEKEEPER.TESTQUEUE TEST.QM

   AMQSBCG0 - starts here
   **********************

   MQOPEN - 'LIFEKEEPER.TESTQUEUE'

   No more messages
   MQCLOSE
   ```

   次のようなメッセージが表示された場合、テストキュー LIFEKEEPER.TESTQUEUE が設定されていません。「LifeKeeper 使用時の WebSphere MQ の設定」の説明に従ってテストキューを作成し、テストを再度実行します。

   ```
   AMQSBCG0 - starts here
   **********************

   MQOPEN - 'LIFEKEEPER.TESTQUEUE'
   MQOPEN failed with CompCode:2, Reason:2085
   ```

2. リソース階層のスイッチオーバを実行します。

3. 手順 1 を前と同じサーバ (スイッチサーバ後はスタンバイサーバ) に対して再度実行します。

## PUT/GET テスト実行の確認

WebSphere MQ Recovery Kit が PUT/GET テストを含むすべてのチェックを実行するかどうかを確認するには、次の手順を行ってください。

1. キューマネージャがいずれかのサーバ上で in service (ISP) であることを確認します。

2. キューマネージャのログレベルをログレベルの変更 の説明に従って「**FINE**」に変更します。

3. キューマネージャが実行されているマシン上でログダイアログを開き、次のチェックが実行されるまで待ちます (最大 2 分間)。

4. ログを分析し、1 つのテストも欠かすことなく、すべてのチェックが実行されたことを確認します。PUT/GET テストは、次の理由で省略されることがあります。

   a. LifeKeeper テストキューが設定されていない (その場合、「LifeKeeper テストキュー名の変更」の説明に従ってテストキューを設定します)。

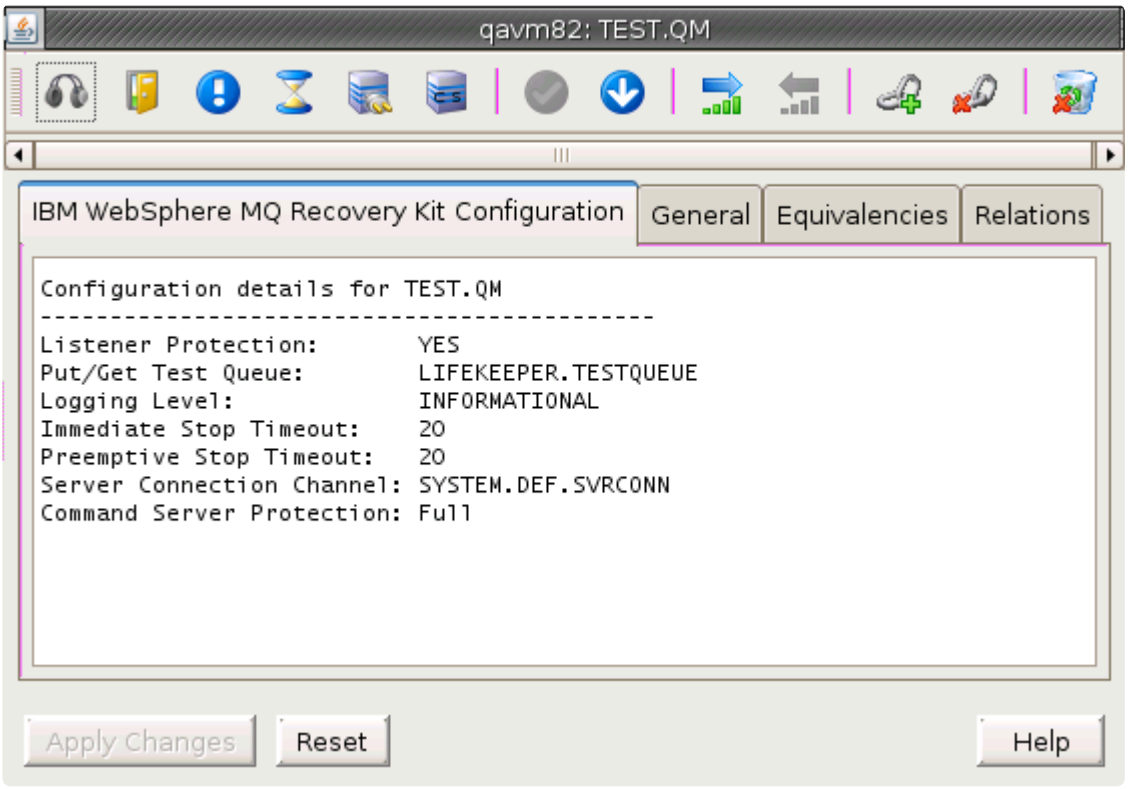
   b. LifeKeeper テストキューが存在しない (その場合、「LifeKeeper 使用時の WebSphere MQ の設定」の説明に従ってテストキューを作成します)。

   c. 変更された amqsget(c) 実行可能ファイルが使用できない (その場合、C コンパイラをインストールし、スクリプト */opt/LifeKeeper/lkadm/subsys/appsuite/mqseries/bin/compilesamples* を再実行します)。

5. デバッグレベルを「INFORMATIONAL」に再度設定します。

# 6.8.5.6. MQ リソースのプロパティの表示

IBM WebSphere MQ リソースのプロパティを表示するには、表示するリソース / サーバの組み合わせのアイコンを右クリックします。リソースコンテクストメニューが表示されたら、 **[Properties]** をクリックします。以下のダイアログが表示されます。



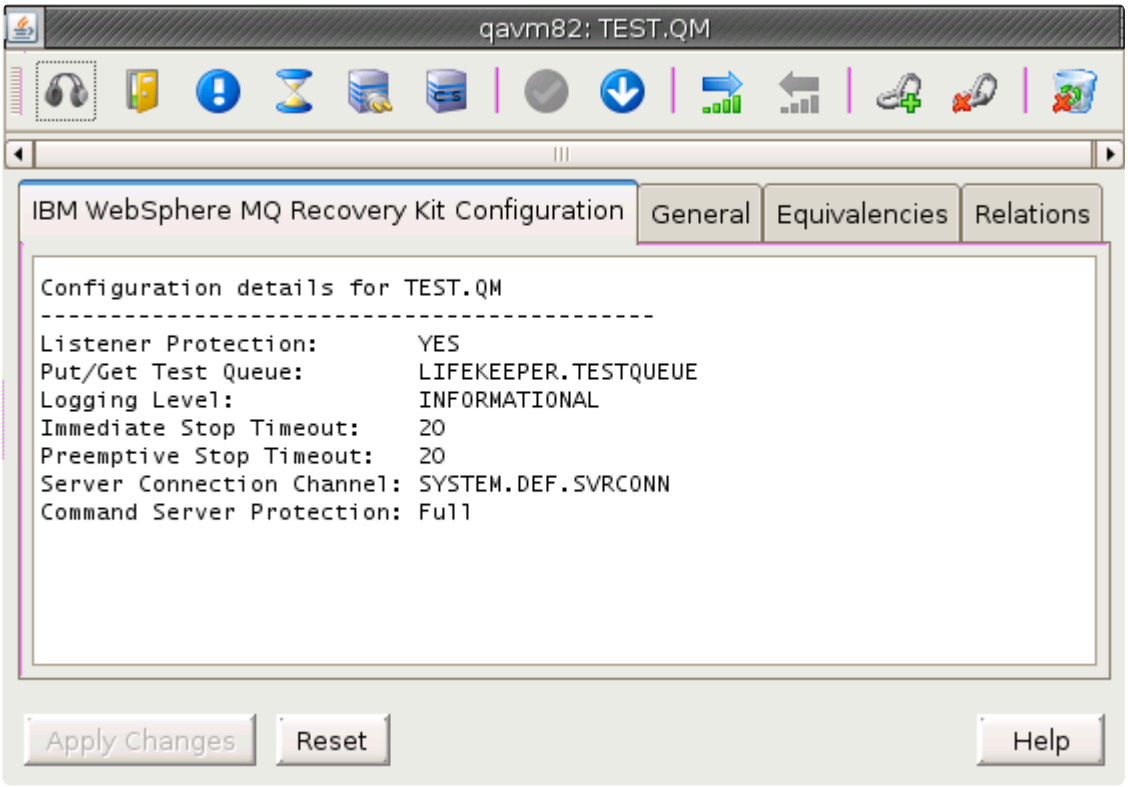

リソースのプロパティは、プロパティ画面(有効になっている場合) に表示されます。プロパティを表示するグローバルリソースのアイコンを右クリックすることもできます。 **[Resource Context]** メニュー が表示されたら、 **[Properties]** をクリックします。ダイアログが表示されたら、表示するリソースが存在するサーバを [Server] リストから選択します。

# 6.8.5.7. MQ リソースプロパティ設定の編集

WebSphere MQ のプロパティ画面では、プロパティパネル (有効な場合) を使用して特定の WebSphere MQ リソースの詳細設定を表示および変更できます。特定の WebSphere MQ リソース設定プロパティは、 **[Resource Context Menu]** を使用して変更することもできます。

LifeKeeper GUI の [Properties Panel] から開く WebSphere MQ の [Configuration Properties] 画面で設定の詳細を編集する前に、GUI の [Properties Panel] が有効になっていることを確認する必要があります。GUI の [Properties Panel] を有効化するには、 **[View]** 、 **[Properties Panel]** を順にクリックします (チェックマークを入れて有効にします)。有効化後に **WebSphere MQ** リソース上で左クリックすると、その設定の詳細が LifeKeeper GUI の [Properties Panel] に表示されます。

以下に、LifeKeeper GUI の [Properties Panel] に表示される WebSphere MQ リソースのプロパティ画面の例を示します。




プロパティ画面にはタブが 4 つあります。1 つ目のタブ、 **[IBM WebSphere MQ Recovery Kit Configuration]** には、WebSphere MQ の各リソースの設定情報が表示され、リソースごとのアイコンを使用して変更できます。他の 3 つのタブは、LifeKeeper のすべてのリソースタイプで共通のため、その内容は SPS for Linux テクニカルドキュメンテーション の [Resource Properties] ダイアログ で説明されています。

次の表には、WebSphere MQ リソース固有のアイコンと、アイコンをクリックすることで変更可能な設定コンポーネントを示しています。

| | | |
|---|---|---|
| | リスナー保護の設定 | IBM WebSphere MQ リスナーの保護を、その他の保護対象の IBM WebSphere MQ キューマネージャコンポーネントに含めるかどうかを指定できます。 |

| | | |
|---|---|---|
| | **PUT/GET テストキューの設定** | 保護対象のキューマネージャに対して IBM WebSphere MQ Recovery Kit が PUT/GETテストを実行するために使用するキューの名前を変更できます。 |
| | **ログレベルの設定** | IBM WebSphere MQ Recovery Kit が保護対象のキューマネージャに使用するログレベルを変更できます。 |
| | **シャットダウンタイムアウトの設定** | 保護対象の IBM WebSphere MQ キューマネージャの即時シャットダウンタイマーおよびプリエンプティブシャットダウンタイマーのタイムアウトを秒単位で変更できます。 |
| | **サーバ接続チャネルの設定** | 保護対象の IBM WebSphere MQ キューマネージャにおいて、クライアント接続および PUT/GET テストに使用するサーバ接続チャネルを変更できます。 |
| | **コマンドサーバ保護の設定** | 保護対象の IBM WebSphere MQ キューマネージャのコマンドサーバコンポーネントの保護/リカバリレベルを指定できます。 |

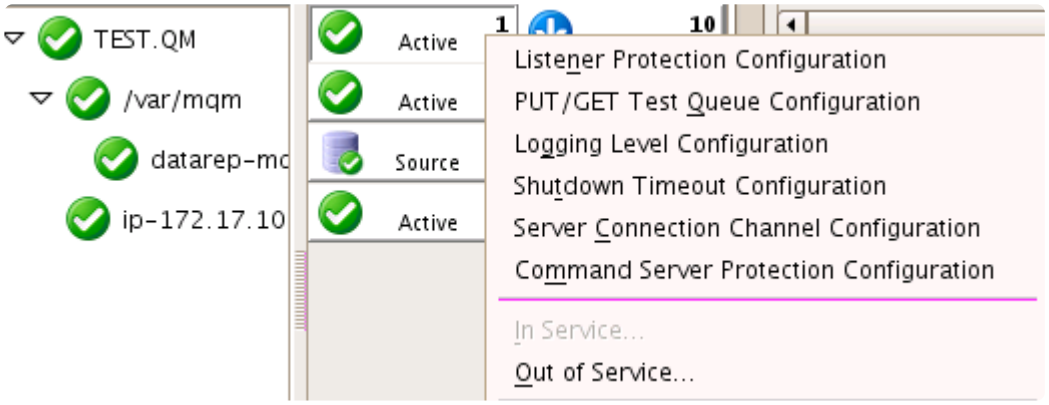
上記の各設定オプションの詳細を以下に示します。

| | |
|---|---|
| **リスナー管理** | LifeKeeper によるキューマネージャの保護の有無を指定します。リスナー管理が無効 (値: NO) の場合、LifeKeeper はリスナーを監視しないため、LifeKeeper リカバリアクションを発動させることなくリスナーを停止できます。リスナー管理が有効 (値: YES) の場合、LifeKeeper はリスナーを監視し、リスナーが実行中でない場合にリスナーを再起動します。リカバリが失敗すると、WebSphere MQ 階層のバックアップサーバへのフェイルオーバが開始されます。 |
| **LifeKeeper のテストキュー** | LifeKeeper は、キューマネージャの動作を監視するために PUT/GET テストを実行します。WebSphere MQ Recovery Kit は、 **専用の** テストキューに対してメッセージを追加および再取得します。障害が検出された場合、リカバリやフェイルオーバは実行されません。その代わり、Recovery Kit は受信するよう登録可能なイベントを送信します。これらのイベントは、putgetfail および putgetcfail と呼ばれます。これらのイベントに対応するため、ディレクトリ */opt/LifeKeeper/events/mqseries/putgetfail* および */opt/LifeKeeper/events/mqseries/putgetcfail* に通知スクリプトを追加できます。<br><br>**注記 1** : LifeKeeper テストキューがキューマネージャに設定されていない場合、PUT/GET テストは行われません。リカバリやフェイルオーバは実行されません。<br><br>**注記 2** : リスナーが保護されている場合、2 回目のクライアント接続チェックが実行されます。このチェックが失敗すると、キューマネージャのリカバリまたはフェイルオーバが試行されます。 |
| **ログレベル** | WebSphere MQ Recovery Kit のログレベルを 4 つのプリセット値に設定できます。<br><br>• ERROR<br><br>このログレベルでは、エラーのみがログに記録されます。情報メッセージはログに記録されません。<br><br>• INFORMATIONAL (デフォルト) |

| | |
|---|---|
| | このログレベルでは、リソースの起動、停止、リカバリに関する LifeKeeper の情報メッセージがログに記録されます。<br><br>• DEBUG<br><br>このログレベルでは、LifeKeeper の情報メッセージと、restore、remove、および recovery スクリプト内のすべての WebSphere MQ コマンドによるコマンド出力がログ記録されます。<br><br>• FINE<br><br>このログレベルでは、start、stop、recovery、および quickCheck スクリプトで実行された WebSphere MQ コマンドによるすべてのコマンド出力がログ記録されます。その他のデバッグメッセージもログ記録されます。<br><br>このデバッグレベルは、デバッグ時のみ設定することを推奨します。`quickCheck` アクションもログ記録されるため、WebSphere MQ キューマネージャで `quickCheck` が実行されるたびにログファイルが一杯になります。<br><br>デフォルトは、 *INFORMATIONAL* です。これは、他のリカバリキットの通常の LifeKeeper ログと同等です。<br><br>**注記:** ログレベルの設定に関わらず、起動、停止、リカバリ、またはチェックルーチン中に発生した WebSphere MQ エラーは、直前に実行されたコマンドの完全な出力とともに常にログに記録されます。 |
| **停止タイムアウト値** | WebSphere MQ Recovery Kit は、キューマネージャを 3 段階で停止します。<br><br>1. 即時停止<br>2. プリエンプティブ停止<br>3. すべてのキューマネージャプロセスを停止<br><br>指定されたタイムアウト値は、Recovery Kit が手順 1 と 2 において正常完了を待機する時間を決定します。このタイムアウトに達すると、シャットダウンプロセスの次の手順が開始します。即時シャットダウンタイムアウトとプリエンプティブシャットダウンタイムアウトのデフォルト値は ***20*** 秒です。 |
| **サーバ接続チャネル** | WebSphere MQ Recovery Kit では、サーバ接続チャネルを指定できます。キットはチャネル `SYSTEM.DEF.SVRCONN` をデフォルトで使用しますが、リソース作成中またはリソース作成後の任意のタイミングで代替チャネルを指定することができます。 |
| **コマンドサーバ** | WebSphere MQ Recovery Kit では、保護対象のキューマネージャのコマンドサーバコンポーネントについて、2 つのレベルの保護およびリカバリを提供します。すなわち、 **Full** レベルと **Minimal** レベルです。<br><br>**Full 保護の場合、コマンドサーバは起動、停止、監視、および復旧され、またリカバリが失敗した場合はフェイルオーバされます。 Full保護のリカバリ手順は、次のとおりです。** |

| | |
|---|---|
| | • コマンドサーバプロセスのみを再起動する。<br>• 上記が失敗した場合、コマンドサーバプロセスを含むキューマネージャ全体を再起動する。<br>• 上記のコマンドサーバ再起動がいずれも失敗した場合、スタンバイノードへのフェイルオーバを開始する。<br><br>**Minimal** 保護の場合、コマンドサーバはリストア中の起動、または、削除中の停止のみ実行されます。コマンドサーバの監視とリカバリは行われません。<br><br>**注記:** リストア中の **Recovery Kit** によるコマンドサーバの起動は、キューマネージャ **SCMDSERV** パラメータが手動起動に設定されている場合のみ行われます。リカバリ中、障害の発生したコマンドサーバの再起動は **SCMDSERV** の設定に関わらず行われます。ただし、コマンドサーバの保護レベルが **Minimal**に設定されている場合を除きます。 |

前述したように、これらの WebSphere MQ リソース設定コンポーネントは、プロパティーパネルでリソース固有のアイコンを使用するか **[Resource Context Menu]** を使用して変更できます。




上記のパラメータは、**LifeKeeper GUI** またはコマンドラインユーティリティを使用して、キューマネージャごとに独立して設定できます。

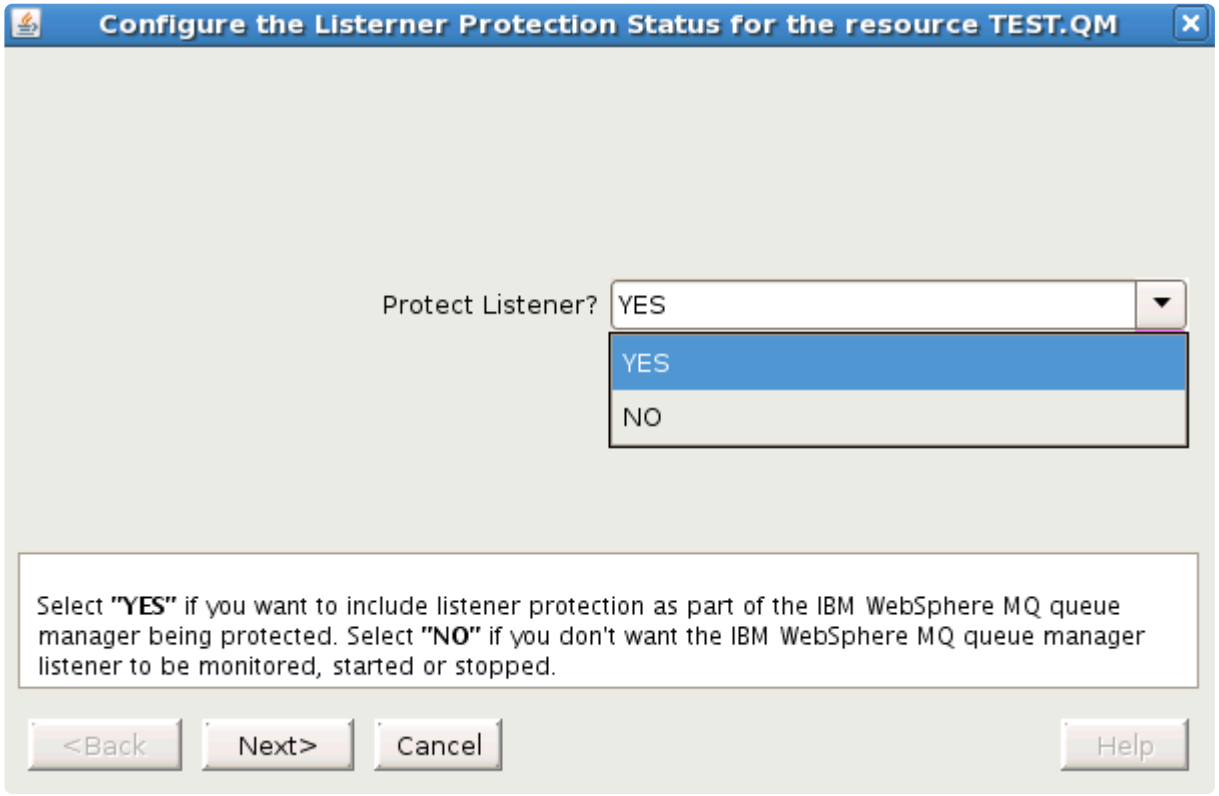
コマンドラインでパラメータを設定するには、次のスクリプトを使用します。

```
$LKROOT/lkadm/subsys/appsuite/mqseries/bin/mq_modqmgrparam
```

# 6.8.5.7.1. リスナー保護の有効化 / 無効化

## GUI

まず、 **WebSphere MQ** のリソースプロパティ画面 または上述の リソースコンテクストメニュー を開きます。リスナー保護の値を変更するには、リソースが in service である必要があります。 **[Listener Protection Configuration]** アイコンまたはメニュー項目をクリックします。以下のダイアログが表示されます。




**[YES]** を選択すると、LifeKeeper は WebSphere MQ リスナーを起動、停止、および監視します。 **[NO]** を選択すると、LifeKeeper は WebSphere MQ リスナーを起動、停止、および監視しません。 **[Next]** をクリックします。リスナー保護を有効化または無効化するかを確認するメッセージが表示されたら、 **[Continue]** をクリックします。リスナー管理を有効化する場合、LifeKeeper GUI はリスナーがすでに実行中であるかチェックします。実行されていない場合、リスナーを起動します。リスナーの起動に成功すると、LifeKeeper GUI は、クラスタ内の各サーバ上でリスナー管理を有効にします。リスナーが実行中でなく、起動できなかった場合、LifeKeeper GUI はクラスタ内の各サーバ上でリスナー管理を有効にしません。

## コマンドライン

LifeKeeper リスナー管理をコマンドラインで設定するには、次のコマンドを使用します。

```
/opt/LifeKeeper/lkadm/subsys/appsuite/mqseries/bin/mq_modqmgrparam -c -s
-i TEST.QM -p LISTENERPROTECTION -v YES
```

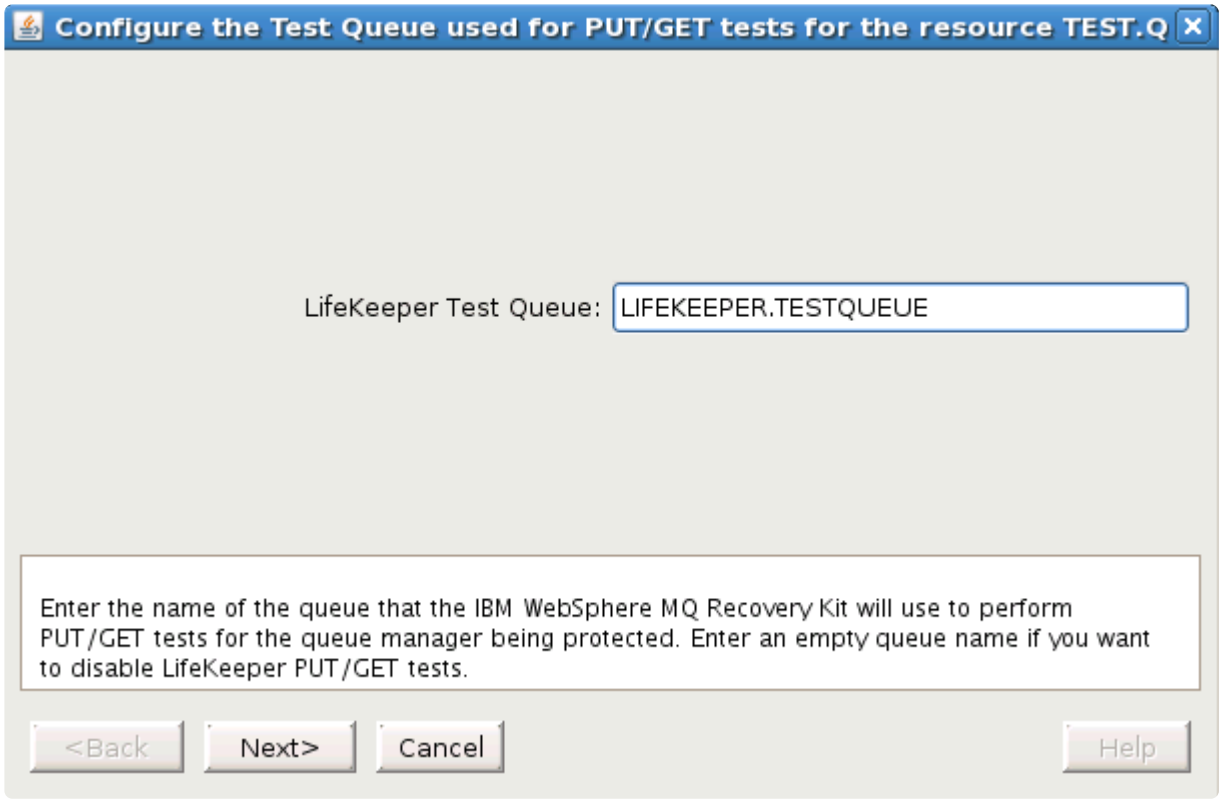
このコマンドを使用すると、キューマネージャ TEST.QM (-i) において、クラスタ (-c) 内の各ノード上の LifeKeeper リスナー管理 (-p) が **YES** (-v) に設定 (-s) されます。

**注記:** キューマネージャの名前（`-i`）または LifeKeeper TAG（`-t`）の名前のいずれかを使用できます。

# 6.8.5.7.2. LifeKeeper テストキュー名の変更

## GUI

まず、 **WebSphere MQ** のリソースプロパティ画面 または上述のリソースコンテクストメニューを開きます。 **[PUT/GET TESTQUEUE Configuration]** アイコンまたはメニュー項目をクリックします。以下のダイアログが表示されます。




LifeKeeper テストキューの名前を入力し、 **[Next]** をクリックします。新しい LifeKeeper テストキューを設定するかを確認するメッセージが表示されたら、 **[Continue]** をクリックします。LifeKeeper GUI により、クラスタ内の各サーバ上で選択されたキューマネージャに LifeKeeper テストキューが設定されます。テストキューに空の値を設定した場合、PUT/GET テストは実行されません。

## コマンドライン

LifeKeeper テストキューをコマンドラインで設定するには、次のコマンドを使用します。

```
/opt/LifeKeeper/lkadm/subsys/appsuite/mqseries/bin/mq_modqmgrparam -c -s
-i TEST.QM -p TESTQUEUE -v "LIFEKEEPER.TESTQUEUE"
```

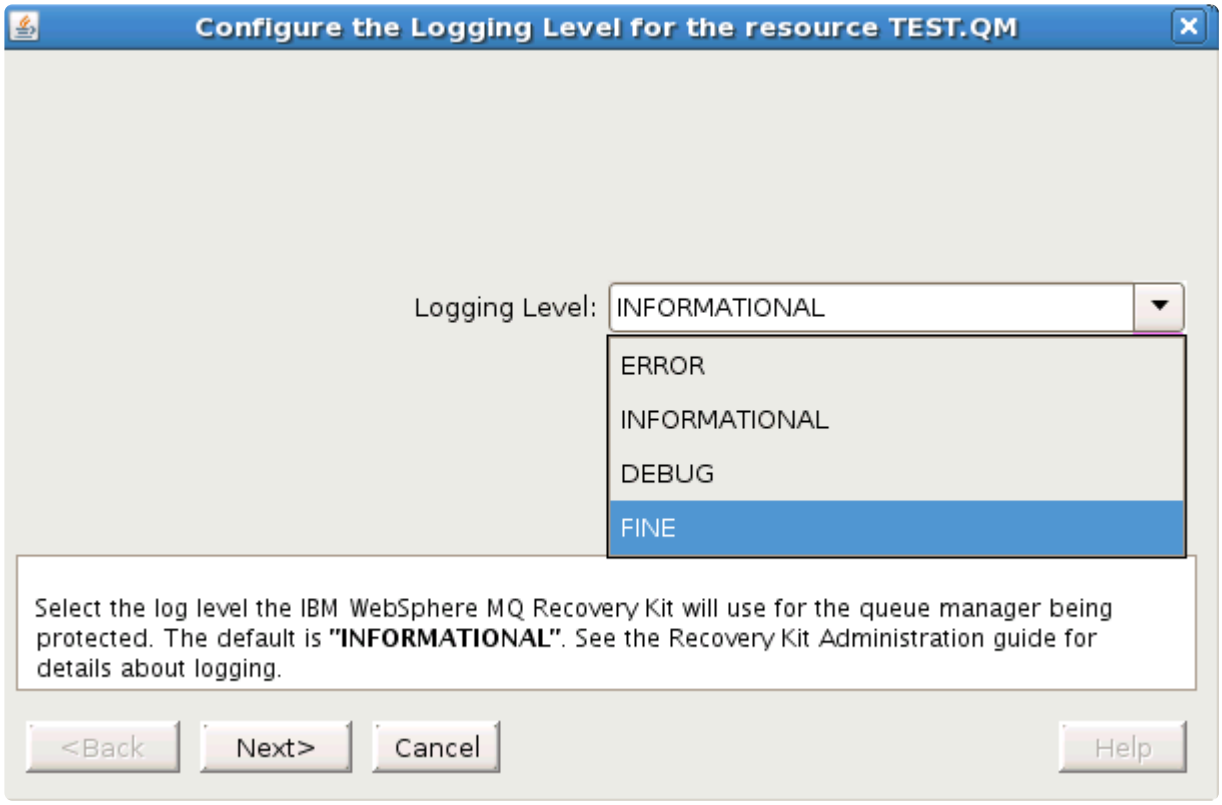
このコマンドを使用すると、キューマネージャ TEST.QM （-i）において、クラスタ (-c) 内の各ノード上の LifeKeeper テストキュー (-p) が LIFEKEEPER.TESTQUEUE （-v）に設定 (-s) されます。

注記: キューマネージャの名前 (-i) または LifeKeeper TAG (-t) の名前のいずれかを使用できます。

# 6.8.5.7.3. ログレベルの変更

## GUI

まず、 **WebSphere MQ** のリソースプロパティ画面 または上述の リソースコンテクストメニュー を開きます。 **[Logging Level Configuration]** アイコンまたはメニュー項目をクリックします。以下のダイアログが表示されます。




**[Logging Level]** を選択して **[Next]** をクリックしてください。新しい LifeKeeper ログレベルを設定するかを確認するメッセージが表示されたら、 **[Continue]** をクリックします。LifeKeeper GUI により、クラスタ内の各サーバ上で選択されたキューマネージャに LifeKeeper ログレベルが設定されます。

## コマンドライン

LifeKeeper ログレベルをコマンドラインで設定するには、次のコマンドを使用します。

```
/opt/LifeKeeper/lkadm/subsys/appsuite/mqseries/bin/mq_modqmgrparam -c -s
-i TEST.QM -p DEBUG -v DEBUG
```

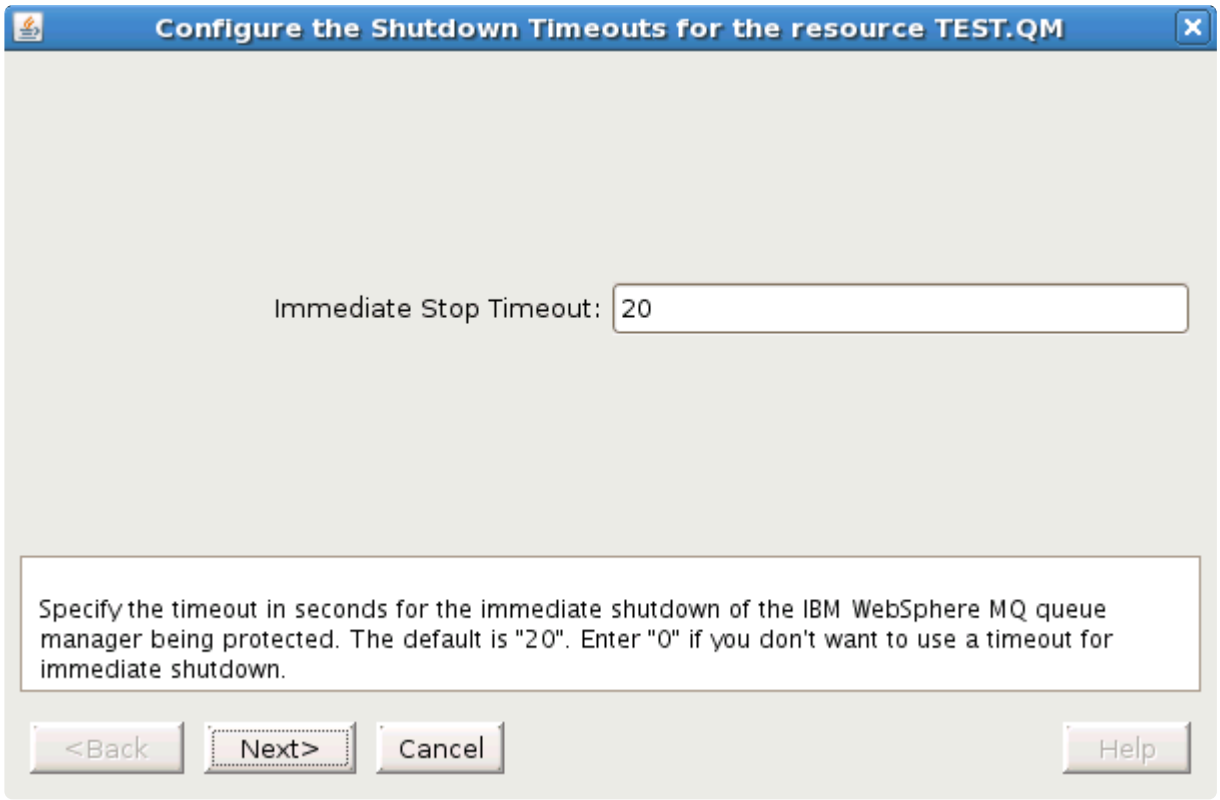
このコマンドを使用すると、キューマネージャ `TEST.QM` （`-i`）において、クラスタ (-c) 内の各ノード上の LifeKeeper ログレベル (-p) が `DEBUG` （`-v`）に設定 (-s) されます。

注記: キューマネージャの名前 (-i) または LifeKeeper TAG (-t) の名前のいずれかを使用できます。

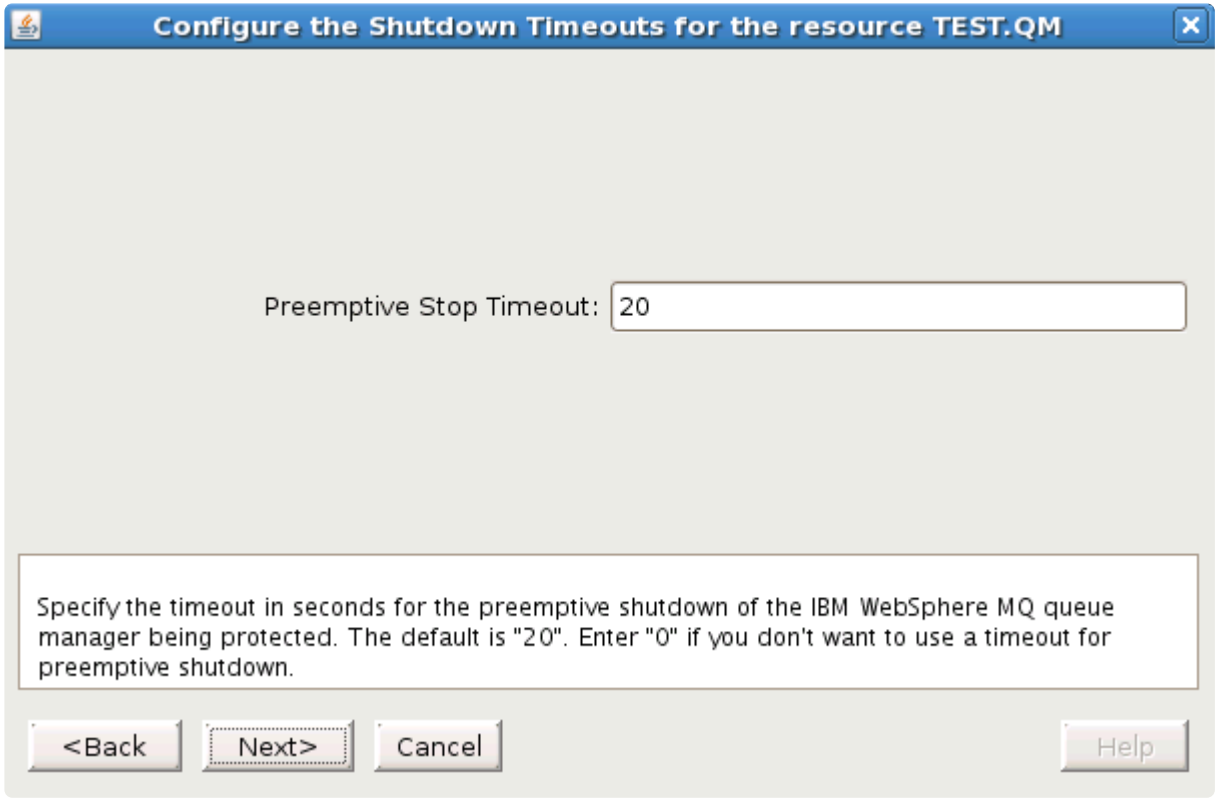
# 6.8.5.7.4. シャットダウンタイムアウト値の変更

## GUI

まず、WebSphere MQ のリソースプロパティ画面または上述のリソースコンテクストメニューを開きます。 **[Shutdown Timeout Configuration]** アイコンまたはメニュー項目をクリックします。以下のダイアログが表示されます。




即時シャットダウンタイムアウト値を秒単位で入力し、 **[Next]** をクリックします。即時シャットダウンタイムアウトを無効にするには、 **0** を入力します。以下のダイアログが表示されます。




プリエンプティブシャットダウンタイムアウト値を秒単位で入力し、 **[Next]** をクリックします。プリエンプティブシャットダウンタイムアウトを無効にするには、0 を入力します。新しい LifeKeeper タイムアウトパラメータを設定するかを確認するメッセージが表示されたら、 **[Continue]** をクリックします。LifeKeeper GUI により、クラスタ内の各サーバ上で、LifeKeeper 即時およびプリエンプティブタイムアウト値が設定されます。

## コマンドライン

LifeKeeper **プリエンプティブシャットダウン** タイムアウト値をコマンドラインで設定するには、次のコマンドを使用します。

```
/opt/LifeKeeper/lkadm/subsys/appsuite/mqseries/bin/mq_modqmgrparam -c -s
-i TEST.QM -p PREEMPTIVE_TIMEOUT -v 20
```

このコマンドを使用すると、キューマネージャ TEST.QM (-i) において、クラスタ (-c) 内の各ノード上の LifeKeeper プリエンプティブシャットダウンタイムアウト (-p) が *20* 秒 (-v) に設定 (-s) されます。

LifeKeeper **即時シャットダウン** タイムアウト値をコマンドラインで設定するには、次のコマンドを使用します。

```
/opt/LifeKeeper/lkadm/subsys/appsuite/mqseries/bin/mq_modqmgrparam -c -s
-i TEST.QM -p IMMEDIATE_TIMEOUT -v 20
```

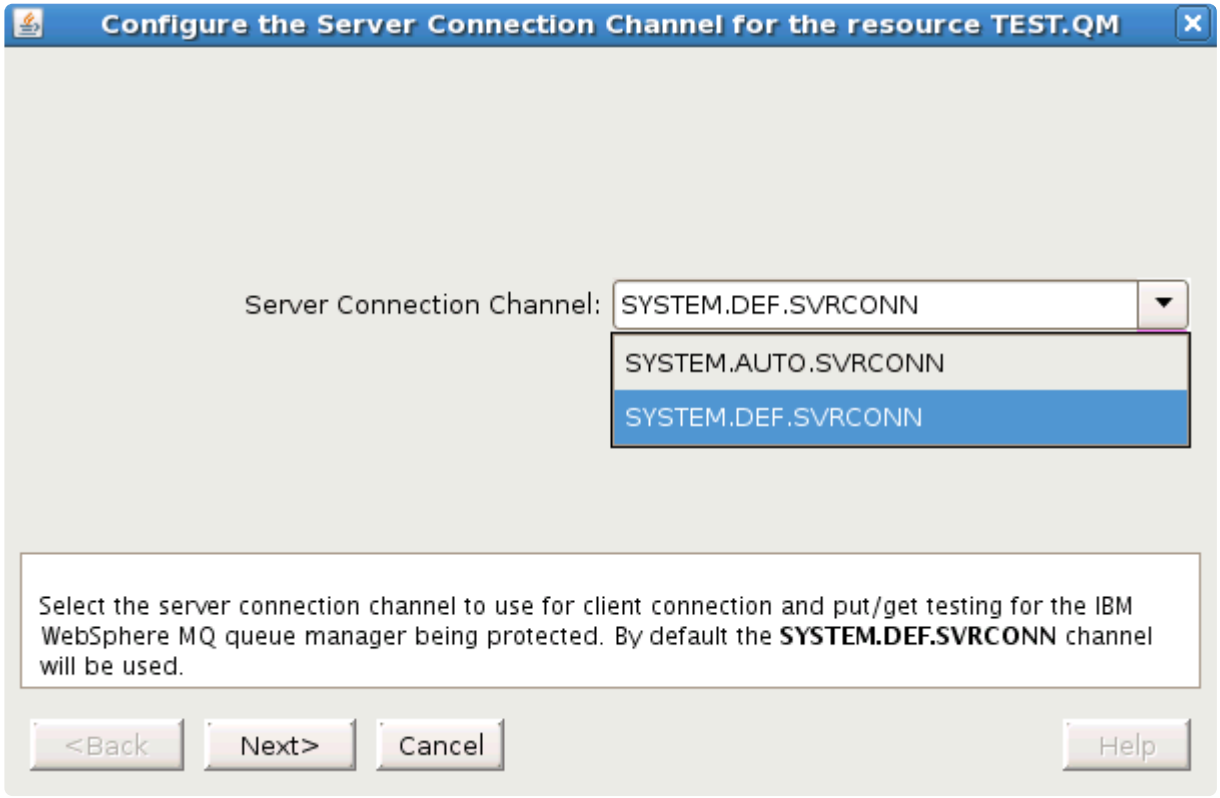
このコマンドを使用すると、キューマネージャ TEST.QM (-i) において、クラスタ (-c) 内の各ノード上の LifeKeeper 即時シャットダウンタイムアウト (-p) が *20* 秒 (-v) に設定 (-s) されます。

**注記**：キューマネージャの名前 (-i) または LifeKeeper TAG (-t) の名前のいずれかを使用できます。

# 6.8.5.7.5. サーバ接続チャネルの変更

## GUI

まず、WebSphere MQ のリソースプロパティ画面または上述のリソースコンテクストメニューを開きます。 **[Server Connection Channel]** の値を変更するには、リソースが in service である必要があります。次に、 **[Server Connection Channel Configuration]** アイコンまたはメニュー項目をクリックします。以下のダイアログが表示されます。




使用するサーバ接続チャネルを **[Server Connection Channel]** で選択し、 **[Next]** をクリックします。新しい **サーバ接続チャネル** に変更するかどうかを確認するメッセージが表示されたら、 **[Continue]** をクリックします。LifeKeeper GUI により、クラスタ内の各サーバ上で選択したキューマネージャのサーバ接続チャネルが設定されます。

## コマンドライン

サーバ接続チャネルをコマンドラインで設定するには、次のコマンドを使用します。

```
/opt/LifeKeeper/lkadm/subsys/appsuite/mqseries/bin/mq_modqmgrparam -c -s
-i TEST.QM -p CHANNEL -v LK.TEST.SVRCONN
```

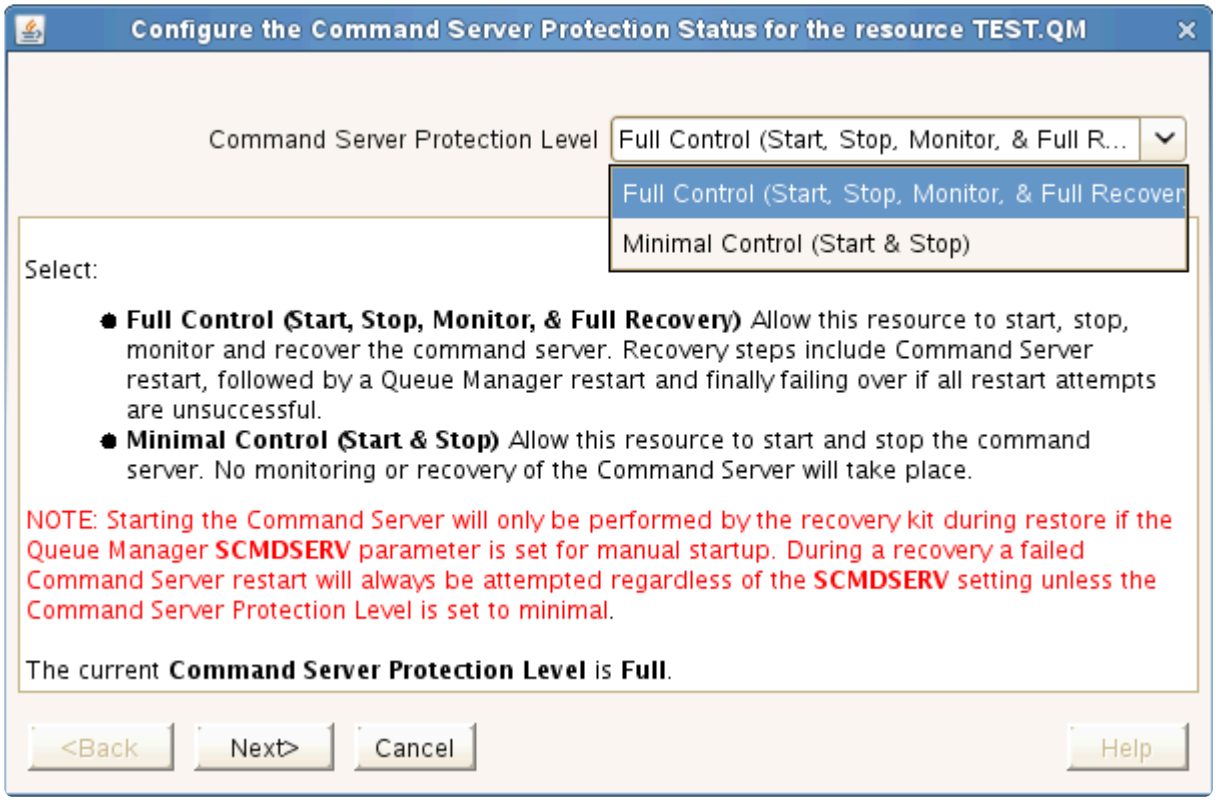
このコマンドを使用すると、キューマネージャ `TEST.QM` (-i) において、クラスタ (-c) 内の各ノード上のサーバ接続チャネル (-p) が *`LK.TEST.SVRCONN`* (-v) に設定 (-s) されます。

注記: キューマネージャの名前 (-i) または LifeKeeper TAG (-t) の名前のいずれかを使用できます。

# 6.8.5.7.6. コマンドサーバ保護設定の変更

## GUI

まず、WebSphere MQ の **リソースプロパティ画面** または上述の **リソースコンテクストメニュー** を開きます。 **[Command Server Protection Configuration]** アイコンまたはメニュー項目をクリックします。以下のダイアログが表示されます。




LifeKeeper が起動、停止、監視、およびリカバリが失敗した場合にフェイルオーバするようにするには、WebSphere MQ キューマネージャのコマンドサーバコンポーネントの **[Full Control]** を選択します。

LifeKeeper が起動と停止のみを行い、監視やリカバリは行わないようにするには、WebSphere MQ キューマネージャのコマンドサーバコンポーネントの **[Minimal Control]** を選択します。

詳細については、上述の表 を参照してください。保護制御を選択したら、 **[Next]** をクリックします。コマンドサーバ保護を現在の設定から新しい設定に変更するか確認するメッセージが表示されたら、 **[Continue]** をクリックして、クラスタ内の全てのノードを変更します。

## コマンドライン

**コマンドサーバ保護設定** をコマンドラインで設定するには、次のコマンドを使用します。

```
/opt/LifeKeeper/lkadm/subsys/appsuite/mqseries/bin/mq_modqmgrparam -c -s
-i TEST.QM -p CMDSERVERPROTECTION -v LEVEL
```

上記の LEVEL は **Full** または **Minimal** です。

このコマンドを使用すると、キューマネージャ TEST.QM (-i) において、クラスタ (-c) 内の各ノード上の LifeKeeper コマンドサーバ保護設定 (-p) が LEVEL (-v) に設定 (-s) されます。

注記: キューマネージャの名前 (-i) または LifeKeeper TAG (-t) の名前のいずれかを使用できます。

# 6.8.5.7.7. LifeKeeper WebSphere MQ Recovery Kit のデフォルト設定の変更

IBM WebSphere MQ Recovery Kit は、多数のデフォルト値を使用しています。デフォルト設定に問題がある場合、これらを調整および変更できます。デフォルト設定は、ほとんどの環境で問題無くできます。タイムアウト値に問題がある場合、次の表を使用して、調整可能なパラメータを特定できます。WebSphere MQ リソース階層に問題が発生するまでは、パラメータを変更しないことをお勧めします。

| /etc/default/LifeKeeper 内の変数名 | デフォルト値 | 説明 |
|---|---|---|
| MQS_QUICKCHECK_TIMEOUT_SC | 10 (秒) | サーバ接続チェックのタイムアウト。 |
| MQS_QUICKCHECK_TIMEOUT_CC | 10 (秒) | クライアント接続チェックのタイムアウト。 |
| MQS_QUICKCHECK_TIMEOUT_PUTGET | 10 (秒) | PUT/GET チェックのタイムアウト。 |
| MQS_QUICKCHECK_TIMEOUT_PS | 5 (秒) | publish/subscribe の使用有無チェックのタイムアウト |
| MQS_QUICKCHECK_TIMEOUT_CLUSTER | 5 (秒) | キューマネージャが WebSphere MQ クラスタの一部であるかのチェックのタイムアウト |
| MQS_QUICKCHECK_TIMEOUT | 40 (秒) | quickCheck スクリプトのタイムアウト (最低 10 秒以上) |
| MQS_QMGR_START_TIMEOUT | 60 (秒) | キューマネージャの起動コマンド完了のタイムアウト |
| MQS_CMDS_START_TIMEOUT | 30 (秒) | コマンドサーバの起動コマンド完了のタイムアウト |
| MQS_LISTENER_START_TIMEOUT | 30 (秒) | リスナーの起動コマンド完了のタイムアウト |
| MQS_LISTENER_LIST_TIMEOUT | 10 (秒) | リスナーの list コマンド完了のタイムアウト |
| MQS_CHECK_TIMEOUT_ACTION | ignore | サーバ接続チェックまたはクライアント接続チェックがタイムアウトになった場合のアクション。デフォルト値の ignore は、タイムアウトに関するメッセージはログに記録されるが、リカバリは開始されないことを表します。この変数を sendevent にすると、サーバ接続チェックがタイムアウトした際にローカルリカバリが開始されます。 |
| MQS_LISTENER_CHECK_DELAY | 2 (秒) | リスナーの起動から、リスナー正常起動 |

| | | |
|---|---|---|
| | | チェックを行うまでの時間 (秒)。デフォルトの 2 秒は、使用状態のポートを検出するのに十分です。 |
| NO_AUTO_STORAGE_DEPS | 0 | この変数を 1 に設定すると、リカバリキットは、キューマネージャとログディレクトリが共有ストレージにあるかどうかを確認しません。1 に設定した場合、リカバリキットは、リソース設定時のファイルシステム階層の作成も行いません。 |
| MQS_DSPMQVER_TIMEOUT | 5 (秒) | dspmqver コマンド (WebSphere MQ のバージョンを知るために必要) のタイムアウト。最低 2 秒必要。 |
| MQS_SKIP_CRT_MISSING_Q | 0 | 1 に設定すると、欠落したテストキューを自動作成しません。 |
| MQS_ALT_USER_NAME | 設定されていない、もしくはユーザーが”mqm”グループにメンバーシップを持っていない場合、mqm | すべてのWebSphere MQコマンドに別のユーザーネームを使用。デフォールトでは、ユーザー”mqm”が使用される。もし設定されている場合には、ユーザーはそのプライマリーグループをグループ”mqm”に設定するか、そのグループにセカンダリーメンバーシップを持たなくてはならない。 |

パラメータを変更するには、上の表内の適切な変数を */etc/default/LifeKeeper* に追加してください。行のシンタックスは、次のとおりです。

```
[…]
MQS_CHECK_TIMEOUT_ACTION=sendevent
[…]
```

カスタム設定を無効化してデフォルト値に戻すには、その行を /etc/default/LifeKeeper から削除するか、コメントアウトします。

# 6.8.6. WebSphere MQ のトラブルシューティング

## WebSphere MQ ログの場所

キューマネージャの名前が既知であり、キューマネージャが使用可能な場合、WebSphere MQ エラーログは、キューマネージャ設定ファイル (*qm.ini*) で定義される LogPath パラメータで指定されたディレクトリにあります。キューマネージャが使用できない場合、エラーログは、(*/var/mqm/qmgrs/@SYSTEM/errors*) にあります。クライアントアプリケーションでエラーが発生した場合、エラーログはクライアントのルートドライブ (*/var/mqm/errors*) にあります。

Message Queue Interface (MQI) 呼び出しが失敗したことを示すリターンコードがアプリケーションに返された場合、リターンコードの詳細について *WebSphere MQ Application Programming Reference Manual* を参照してください。

# 6.8.6.1. MQ エラーメッセージ

このセクションでは、LifeKeeper MQ Recovery Kit の使用中に表示される可能性のあるメッセージの一覧を示します。必要に応じて、エラーの原因およびエラー状態を解消するために必要な処置についても説明しています。

MQ Recovery Kit は、階層の作成および拡張時に他の LifeKeeper コンポーネントを利用するため、それらの他のコンポーネントによるメッセージが表示されることもあります。その場合、メッセージカタログ を参照してください。メッセージカタログでは、操作、管理、GUI など、LifeKeeper for Linux を使用しているときに出会う可能性がある、すべてのエラーコードを列挙します。また、エラーコードの原因に関する補足説明や、問題解決のために必要な処置についても、必要に応じて記載します。この完全なリストを検索すると、受信したエラーコードを見つけることができます。また、適切な LifeKeeper コンポーネントの個別のメッセージカタログに直接アクセスすることもできます。

## 共通のエラーメッセージ

| エラー番号 | エラーメッセージ | 処置 |
|---|---|---|
| 119001 | Queue manager with TAG "TAG" failed to start on server "SERVER" with return code "Code" | start コマンドは正常に実行されましたが、開始後のチェックが失敗しました。<br><br>SERVER 上の IBM WebSphere MQ アラートログを確認し、エラーを修正してください。 |
| 119002 | Queue manager with TAG "TAG" start command failed on server "SERVER" with return code "Code". | キューマネージャ TAG に対する start コマンドが 0 以外の値を返しました。<br><br>SERVER 上の IBM WebSphere MQ アラートログを確認し、エラーを修正してください。<br><br>リターンコード Code は、strmqm コマンドのリターンコードです。 |
| 119006 | Command server start command for queue manager "TAG" failed on server "SERVER" with return code "Code". | コマンドサーバに対する起動コマンドが、0 以外の値を返しました。<br><br>SERVER 上の IBM WebSphere MQ アラートログを確認し、エラーを修正してください。<br><br>リターンコード Code は、runmqsc コマンドのリターンコードです。<br><br>WebSphere MQ v6.0 の場合、コマンドサー |

| | | |
|---|---|---|
| | | バの起動タイプが「MANUAL」になっていることを確認してください。詳細については、設定の要件 を参照してください。 |
| 119007 | Listener for queue manager "TAG" failed to start on server "SERVER". | SERVER 上の IBM WebSphere MQ アラートログを確認し、エラーを修正してください。 |
| 119008 | Listener start command for queue manager with TAG "TAG" failed on server "SERVER" with return code "CODE". | SERVER 上の IBM WebSphere MQ アラートログを確認し、エラーを修正してください。 |
| 119013 | Could not create queue manager object for queue manager “QMGR” with TAG “TAG”. | LifeKeeper と WebSphere MQ のエラーログを確認してください。 |
| 119014 | Could not create listener object for queue manager “QMGR” with TAG “TAG”. | LifeKeeper と WebSphere MQ のエラーログを確認してください。 |
| 119015 | No value forthe “PARAMETER” specified. | LifeKeeper MQ Recovery Kit スクリプトを、正しい引数を使用して実行してください。 |
| 119016 | Instance with ID “ID” does not exist on server “SERVER”. | リソース階層を確認してください。 |
| 119017 | Instance with TAG “TAG” does not exist on server “SERVER”. | リソース階層を確認してください。 |
| 119018 | Invalid parameters specified. | 正しいオプションを使用してスクリプトを実行してください。 |
| 119019 | Too few parameters specified | 正しいオプションを使用してスクリプトを実行してください。 |
| 119021 | Failed to set “VALUE” for resource instance “TAG” on server “SERVER”. | LifeKeeper ログで、値の設定に関するエラーを確認してください。 |
| 119025 | Failed to update instance info for queue manager with TAG “TAG” on server “SERVER”. | サーバが動作中の状態に戻ったら、操作を再度実行して設定を同期してください。 |
| 119026 | The following program required does not exist or is not executable:"EXECUTABLE".Check failed. | プログラム EXECUTABLE が見つかりませんでした。すべてのインストール要件が満たされていることを確認し、必要なパッケージをすべてインストールしてください。<br><br>詳細については、設定の要件 を参照してください。 |
| 119032 | Script: usage error (error message) | 正しい引数を使用して、スクリプト Script を開始してください。 |
| 119033 | Script: error parsing config file "ConfigFile". | ConfigFile が存在し、読み取り可能であることを確認してください。 |

| | | |
|---|---|---|
| 119034 | CHECKTYPE check for queue manager with TAG "TAG" failed on server "SERVER" because the MQUSER could not be determined.This is probably because of a removed configuration file - ignoring. | タグ TAG のキューマネージャに対する CHECKTYPE チェックが失敗しました。<br><br>グローバル設定ファイル (mqs.ini) が存在し、読み取り可能であることを確認してください。<br><br>削除されている場合、mqs.ini 設定ファイルを再作成してください。 |
| 119035 | CHECKTYPE check for queue manager with TAG "TAG" failed on server "SERVER" because no TCP PORT directive found in config file "CONFIGFILE" - ignoring. | キューマネージャ設定ファイル (qm.ini) が存在し、インストール中に必要な TCP セクションが含まれていることを確認してください。<br><br>TCP セクションをキューマネージャ設定ファイルに追加してください。 |
| 119042 | “CHECKTYPE” check for queue manager with TAG “TAG” failed on server “SERVER” because no TCP PORT information was found via runmqsc. | リスナーオブジェクトのポート情報が定義済みであり、runmqsc でアクセス可能であることを確認してください。 |
| 119043 | TCP Listener configuration could not be read, reason:“REASON”. | MQ が実行中であり、リスナーオブジェクトのポート情報が定義済みで、runmqsc でアクセス可能であることを確認してください。 |
| 119044 | No TCP Listener configured, no TCP PORT information was found via runmqsc:“MESSAGE”. | リスナーオブジェクトのポート情報が定義済みであり、runmqsc でアクセス可能であることを確認してください。 |

## 作成

| エラー番号 | エラーメッセージ | 処置 |
|---|---|---|
| 001022 | END failed hierarchy “CREATE” of resource “TAG” on server “SERVER” with return value of “VALUE”. | サーバ SERVER 上の LifeKeeper ログに、リソース階層の作成に関するエラーがないか確認します。このエラーは、キューマネージャが起動していないことに関係する場合があります。 |
| 119020 | Create MQSeries queue manager resource with TAG “TAG” for queue manager “QMGR” failed. | LifeKeeper ログにリソースの作成に関するエラーがないか確認します。このエラーは、キューマネージャが起動していないことに関係する場合があります。 |
| 119022 | Failed to create dependency between “PARENT” and “CHILD”. | LifeKeeper ログに依存関係の作成に関するエラーがないか確認します。 |
| 119023 | Creating the filesystem hierarchies for | LifeKeeper ログにファイルシステム階層の作成に関 |

| | | |
|---|---|---|
| | queue manager with TAG “TAG” failed.File systems:“Filesystems”. | するエラーがないかを確認します。 |
| 119029 | No TCP section configured in "CONFIGFILE" on server "SERVER". | サーバ SERVER 上のキューマネージャ設定ファイルに TCP セクションを追加します。詳細については、「設定の要件 」を参照してください。 |
| 119031 | Queue manager "DIRTYPE" directory ("DIRECTORY") not on shared storage. | ディレクトリ DIRECTORY を共有ストレージに移動して、操作を再度実行してください。 |
| 119038 | Creation of queue manager resource with TAG "TAG" failed on server "SERVER". | サーバ SERVER 上の LifeKeeper ログのエラーを確認および修正し、操作を再度実行してください。 |
| 119039 | TCP section in configuration file "FILE" on line "LINE1" is located before LOG section on line "LINE2" on server "SERVER". | キューマネージャ設定ファイル内で、TCP セクションは LOG: セクションより後に配置することを推奨します。<br><br>TCP セクションをキューマネージャ設定ファイルの末尾に移動し、操作を再度実行してください。 |
| 119040 | Creation of MQSeries queue manager resource by create_ins was successful but no resource with TAG “TAG” exists on server “SERVER”.Sanity check failed. | LifeKeeper ログにリソース作成中のエラーがないか確認してください。 |
| 119041 | Creation of MQSeries queue manager resource was successful but no resource with TAG “TAG” exists on server “SERVER”.Final sanity check failed. | LifeKeeper ログにリソース作成中のエラーがないか確認してください。 |

## 拡張

| エラー番号 | エラーメッセージ | 処置 |
|---|---|---|
| 119024 | Instance "TAG" can not be extended from "TEMPLATESYS" to "TARGETSYS".Reason:REASON | REASON に記載されているエラーを修正し、操作を再度実行してください。 |
| 119027 | The user "USER" does not exist on server "SERVER". | プライマリサーバ上と同じ UID を持つユーザ USER を SERVER 上に作成し、操作を再度実行してください。 |
| 119028 | The user "USER" has a different numeric UID on server "SERVER1" (SERVER1UID) then it should be (SERVER2UID). | USER が全サーバ上で同一の UID を持つように UID を変更し、UID を変更したサーバ上の WebSphere MQ を再インストールしてから、操作を再度実行してください。 |
| 119029 | No TCP section configured in "CONFIGFILE" on server "SERVER". | サーバ SERVER 上のキューマネージャ設定ファイルに TCP セクションを追加します。 |

| | | |
|---|---|---|
| | | 詳細については、「設定の要件 」を参照してください。 |
| 119030 | Queue manager "QMGR" not configured in "CONFIGFILE" on server "SERVER". | 拡張対象のキューマネージャ QMGR は、ターゲットサーバ SERVER 上のグローバル設定ファイルに設定されていません。キューマネージャのスタンザをサーバ SERVER 上の設定ファイル CONFIGFILE に追加し、操作を再度実行してください。 |
| 119036 | Link "LINK" points to "LINKTARGET" but should point to "REALTARGET" on server "SERVER". | ファイルシステムレイアウト上、3 つのシンボリックリンクがテンプレートおよびターゲットサーバ SERVER 上の同じ場所を指している必要があります。<br><br>サーバ SERVER 上のリンク LINK を REALTARGET を指すように修正し、操作を再度実行してください。 |
| 119037 | Link "LINK" that should point to "REALTARGET" does not exist on system "SERVER". | ファイルシステムレイアウト上、3 つのシンボリックリンクは、ターゲットサーバにも存在する必要があります。<br><br>サーバ SERVER 上に REALTARGET への必要なリンク LINK を作成し、操作を再度実行してください。 |

## 削除

| エラー番号 | エラーメッセージ | 処置 |
|---|---|---|
| 119003 | Failed to stop queue manager with TAG “TAG” on server “SERVER". | サーバ SERVER 上のキューマネージャ TAG が Recovery Kit によって停止できませんでした。詳細な情報を確認し調査するには、ログレベルを DEBUG に変更してください。マシン負荷によっては、シャットダウンタイムアウト値を大きくする必要がある場合があります。 |
| 119004 | Some orphans of queue manager with TAG “TAG” could not be stopped on server "SERVER".Tried it “tries” times. | オーファンを手動で停止し、Queue Manager を再起動してください。より詳細な情報を得るには、ログレベルを DEBUG に変更します。 |
| 119010 | Listener for queue manager with TAG "TAG" failed to stop on server "SERVER". | このメッセージは、リスナーの監視が有効な場合のみ表示されます。より詳細な情報を得るには、ログレベルを DEBUG に変更します。 |

## リソース監視

| エラー番号 | エラーメッセージ | 処置 |
|---|---|---|
| 119005 | Queue manager with TAG “TAG” on server | SERVER 上の IBM WebSphere MQ アラートログでエラーを確認します。このメッセージは、キューマネージャがクラッシュしたことを示 |

| | | |
|---|---|---|
| | "SERVER" failed. | しています。 |
| 119009 | Listener for queue manager with TAG “TAG” failed on server "SERVER". | このメッセージは、リスナーの監視が有効な場合のみ表示されます。より詳細な情報を得るには、ログレベルを FINE に変更します。 |
| 119011 | “CHECKTYPE" PUT/GET Test for queue manager with TAG "TAG" failed on server "SERVER" with return code "Code" | このメッセージは、PUT/GET テストが有効で、テストキューが存在する場合のみ表示されます。より詳細な情報を得るには、ログレベルを FINE に変更し、SERVER 上の IBM WebSphere キューマネージャエラーログ (/var/mqm/errors) でエラーを確認し、修正します。ファイルシステムの容量が不足していないか確認します。 |
| 119012 | Client connect test for queue manager with TAG "TAG" on server "SERVER" failed with return code "Code". | このメッセージは、リスナーの管理が有効な場合のみ表示されます。<br><br>このメッセージは、リスナーまたはキューマネージャの問題を示しています。<br><br>ログでエラーを確認し、修正します。<br><br>リターンコード Code は、amqscnxc コマンドのリターンコードです。 |

## 警告メッセージ

| エラー番号 | エラーメッセージ | 処置 |
|---|---|---|
| 119201 | Listener for queue manager with TAG "TAG" is NOT monitored on server "SERVER". | これは、リスナーの管理が有効になっていないことを示す警告です。 |
| 119202 | Queue manager with TAG “TAG” is not running on server “SERVER” but some orphans are still active.This is attempt number “ATTEMPT” at stopping all orphans processes. | これは、MQ が正しく停止されなかったことを示す警告です。 |
| 119203 | Another instance of recover is running, exiting “EXITCODE”. | リカバリプロセスを開始したが、別のリカバリプロセスがすでに実行中のため、このリカバリプロセスを継続できません。 |
| 119204 | Queue manager server connect check for queue manager with TAG "TAG%" timed out after "SECONDS" seconds on server "SERVER". | このメッセージがよく表示される場合、/etc/defaul/LifeKeeper の MQS_QUICKCHECK_TIMEOUT_SC の値を大きくしてください。<br><br>詳細については、サーバ接続チャネルの変更を参照してください。 |
| 119205 | Queue manager client connect check for queue manager with TAG "TAG" timed out after | このメッセージがよく表示される場合 /etc/defaul/LifeKeeper の |

| | | |
|---|---|---|
| | "SECONDS" seconds on server "SERVER". | MQS_QUICKCHECK_TIMEOUT_CC の値を大きくしてください。<br><br>詳細については、サーバ接続チャネルの変更を参照してください。 |
| 119206 | Server "SERVER" is not available, skipping. | キューマネージャ設定の更新中に、サーバがオンラインではありませんでした。<br><br>サーバがオンラインに戻るのを待ってから、設定手順を再度実行してください。 |
| 119207 | "CHECKTYPE" PUT/GET test for queue manager with TAG "TAG" failed because test queue "QUEUE" does not exist (reason code "REASONCODE") - ignoring. | テストキュー QUEUE の設定を作成するか、既存のキューにテストキューを再設定します。<br><br>テストキューの作成については、「設定の要件」を参照してください。 |
| 119208 | Channel "CHANNEL" does not exist for queue manager with TAG "TAG" (reason code "REASONCODE") - ignoring. | 存在していないチャネル CHANNEL を作成してください。デフォルトでは、チャネル SYSTEM.DEF.SVRCONN が使用されます。<br><br>チャネルの作成方法については、WebSphere MQ ドキュメンテーションを参照してください。 |
| 119209 | PUT/GET test for queue manager with TAG "TAG" skipped because no test queue is defined. | キューマネージャ TAG に LifeKeeper テストキューを設定してください。 |
| 119210 | The following program required to perform the PUT/GET test does not exist or is not executable:"EXECUTBALE".Test skipped. | C コンパイラをシステムにインストールし、それが root ユーザの PATH 環境変数に配置されていることを確認してください。スクリプト LKROOT/lkadm/subsys/appsuite/mqseries/bin/compilesamples を実行し、修正済みサンプルプログラムの amqsget および amqsgetc をコンパイルします。 |
| 119211 | Queue manager "CHECKTYPE" PUT/GET test for queue manager with TAG "TAG" timed out after "SECONDS" seconds on server "SERVER". | このメッセージがよく表示される場合、/etc/default/LifeKeeper の MQS_QUICKCHECK_TIMEOUT_PUTGET の値を大きくしてください。<br><br>詳細については、サーバ接続チャネルの変更を参照してください。 |
| 119212 | QuickCheck for queue manager with TAG “TAG” timed out after SECONDS seconds on server “SERVER”. | このメッセージがよく表示される場合、/etc/default/LifeKeeper の MQS_QUICKCHECK_TIMEOUT の値を大きくしてください。 |

| | | |
|---|---|---|
| 119213 | mqseriesQueueManager::getMQVersion::ERROR unexpected dspmqver output (OUTPUT) – reading cached value instead (Queue QUEUE, Queuemanager QMGR). | runmqsc による MQ バージョンの読み取りに失敗しました。このメッセージがよく表示される場合、/etc/default/LifeKeeper の MQS_DSPMQVER_TIMEOUT の値を大きくしてください。 |
| 119214 | mqseriesQueueManager::getMQVersion:: ERROR unexpected output retrieving MQ version information (Queue QUEUE, Queuemanager QMGR). Unexpected results * | mqm ユーザとして次のコマンドを実行して、出力が生成されるかを確認してください: dspmqver -b -p1 -f2. また、mqm ユーザとして次のコマンドを実行して、出力を確認してください: dspmqinst |

# 6.8.7. 付録 A – mqs.ini 設定ファイルの例

```
#********************************************************************#
#*                                                                  *#
#* <START_COPYRIGHT>                                                *#
#* Licensed Materials - Property of IBM                             *#
#* 63H9336                                                          *#
#* (C) Copyright IBM Corporation 1994, 2000                         *#
#*                                                                  *#
#* <END_COPYRIGHT>                                                  *#
#********************************************************************#
#* Module Name: mqs.ini                                               *#
#* Type       : WebSphere MQ Machine-wide Configuration File          *#
#* Function   : Define WebSphere MQ resources for an entire machine   *#
#********************************************************************#
#* Notes      :                                                       *#
#* 1) This is the installation time default configuration             *#
#*                                                                    *#
#********************************************************************#
AllQueueManagers:
   #****************************************************************#
   #* The path to the qmgrs directory, below which queue manager data  *#
   #* is stored                                                      *#
   #****************************************************************#
   DefaultPrefix=/var/mqm

LogDefaults:
   LogPrimaryFiles=3
   LogSecondaryFiles=2
   LogFilePages=1024
   LogType=CIRCULAR
   LogBufferPages=0
   LogDefaultPath=/var/mqm/log
QueueManager:
   Name=TEST.QM
   Prefix=/var/mqm
   Directory=TEST!QM
DefaultQueueManager:
   Name=TEST.QM
QueueManager:
   Name=TEST.QM.NEW
   Prefix=/var/mqm
   Directory=TEST!QM!NEW
QueueManager:
   Name=TEST.QM2
   Prefix=/var/mqm
```

```
   Directory=TEST!QM2
QueueManager:
   Name=MULTIINS_1
   Prefix=/var/mqm
   Directory=MULTIINS_1
   DataPath=/opt/webmq/MULTIINS_1/data

   InstallationName=Installation1


QueueManager:
   Name=MULTIINS_2
   Prefix=/var/mqm
   Directory=MULTIINS_2
   DataPath=/opt/webmq/MULTIINS_2/data

   InstallationName=Installation2
```

# 6.8.8. 付録 B – qm.ini 設定ファイルの例

```
#********************************************************************#
#* Module Name: qm.ini                                              *#
#* Type       : WebSphere MQ queue manager configuration file       *#
#  Function   : Define the configuration of a single queue manager *#
#*                                                                  *#
#********************************************************************#
#* Notes      :                                                     *#
#* 1) This file defines the configuration of the queue manager      *#
#*
*#
#********************************************************************#
ExitPath:
   ExitsDefaultPath=/var/mqm/exits/
   ExitsDefaultPath64=/var/mqm/exits64
#*                                                                  *#
#*                                                                  *#
Log:
   LogPrimaryFiles=3
   LogSecondaryFiles=2
   LogFilePages=1024
   LogType=CIRCULAR
   LogBufferPages=0
   LogPath=/opt/MQ_log/MULTIINS_1
   LogWriteIntegrity=TripleWrite
Service:
   Name=AuthorizationService
   EntryPoints=14
ServiceComponent:
   Service=AuthorizationService
   Name=MQSeries.UNIX.auth.service
   Module=amqzfu
   ComponentDataSize=0
```

# 6.8.9. 付録 C – WebSphere MQ 設定シート

| | | |
|---|---|---|
| クラスタ名 | | |
| 連絡先<br><br>(クラスタ責任者のメールアドレスまたは電話番号) | | |
| LifeKeeper のバージョン | | |
| オペレーティングシステム | | |
| クラスタノード | 名前 | パブリック IP / ネットマスク |
| | | |
| | | |
| | | |
| | | |

| | | |
|---|---|---|
| キューマネージャーの名前 | | |
| LifeKeeper によるリスナー管理 | [ ] YES [ ] NO | |
| WebSphere MQ オペレーティングシステムユーザ | 名前 | 数値 (UID/GID) |
| ユーザ<br><br>(例: mqm/1002) | | |
| グループ<br><br>(例: mqm/200) | | |
| 仮想 IP / ネットマスク / ネットワークドライブ<br><br>(例: 192.168.1.1/24/eth0) | | |
| ファイルシステムレイアウト | __ 設定 1 – 共有ストレージ上の/var/mqm | |
| | __ 設定 2 – 直接マウント | |
| | __ 設定 3 – シンボリックリンク<br><br>__ 設定 4 – マルチインスタンスのキューマネージャ | |
| | __ その他 | |
| 共有ストレージのタイプ | __ NAS<br>(IP:_____) | |

| | |
|---|---|
| | __ SCSI/FC (タイプ:_____)<br>__ SDR |
| キューマネージャ */var/mqm/qmgrs/QM.NAME* の物理的場所<br>(デバイス、マウントポイント、または論理ボリューム)<br><br>(例: LVM /dev/mqm_test_qm/qmgrs ) | |
| キューマネージャ */var/mqm/log/QM.NAME* の物理的場所<br><br>(デバイス、マウントポイント、または論理ボリューム)<br>(例: LVM /dev/mqm_test_qm/log ) | |

# 6.9. NAS Recovery Kit 管理ガイド

## NAS Recovery Kit テクニカルドキュメンテーション

LifeKeeper for Linux Network Attached Storage Recovery Kit（以下、NAS Recovery Kit と呼ぶ）は、LifeKeeper 環境においてネットワークファイルシステム（NFS）ソフトウェアの障害回復能力を高めるものです。NAS Recovery Kit は、LifeKeeper ユーザに対し、 エクスポートされた NFS ファイルシステムを LifeKeeper 階層の基本ストレージとして使用する機会を提供します。

## ドキュメントの内容

本書は以下のトピックについて説明しています。

- NAS ドキュメンテーションとリファレンス ：LifeKeeper for Linux のドキュメントと所在の一覧です。
- NAS Recovery Kit ハードウェア要件とソフトウェア要件 ：NAS Recovery Kit を適切に設定、導入、運用するうえで必要なハードウェアとソフトウェアについて説明しています。LifeKeeper for Linux ソフトウェアの具体的なインストール / アンインストール方法については、LifeKeeper for Linux インストールガイド を参照してください。
- NAS Recovery Kit 概要 ：NAS Recovery Kit の特徴と機能について説明しています。
- LifeKeeper for Linux NAS Recovery Kit の設定 ：NAS Recovery Kit を適切に設定するうえで必要な手順について説明しています。
- LifeKeeper 設定作業 ：LifeKeeper GUI を使用して NAS リソース階層を作成し管理する作業について説明しています。
- NAS トラブルシューティング ：LifeKeeper for Linux のエラーメッセージの一覧とその説明です。

# 6.9.1. NAS ドキュメンテーションとリファレンス

以下に、SIOS Technology Corp が発行している LifeKeeper 関連情報の一覧を示します。

- LifeKeeper for Linux リリースノート
- LifeKeeper for Linux テクニカルドキュメンテーション
- LifeKeeper for Linux インストールガイド
- Optional Recovery Kit ドキュメンテーション

このドキュメンテーション、および他の LifeKeeper Recovery Kit に関連するドキュメンテーションは、以下の Web サイトで入手できます。

http://jpdocs.us.sios.com

# 6.9.2. NAS Recovery Kit ハードウェア要件とソフトウェア要件

LifeKeeper for Linux NAS Recovery Kit をインストールする前に、LifeKeeper の構成が次の要件を満たしている必要があります。LifeKeeper for Linux ハードウェアおよびソフトウェアの設定に関する具体的な手順については、LifeKeeper for Linux インストールガイド を参照してください。

## ハードウェア要件

- サーバ ：製品媒体に付属する LifeKeeper for Linux リリースノート および LifeKeeper for Linux インストールガイド (製品メディアに収録) に記載の要件に従って設定されたLifeKeeper for Linux 対応サーバ。

- **IP ネットワークインターフェースカード** ：各サーバーは、イーサネット TCP/IP をサポートするネットワークインタフェースを 1 つ以上必要とします。ただし、LifeKeeper クラスターには 2 つの通信パスが必要で、独立した 2 つのサブネットを使用する 2 つの分離した LAN ベース通信パスが推奨され、これらの 1 つ以上をプライベートネットワーク として構成する必要があることを忘れないでください。TCP と TTY のハートビートの組み合わせもサポートされています。

## ソフトウェア要件

- **TCP/IP** ソフトウェア ：LifeKeeper 環境の各サーバには TCP/IP ソフトウェアが必須です。

- **LifeKeeper** ソフトウェア ：クラスタの各サーバにインストールした LifeKeeper for Linux ソフトウェアのバージョン、および適用した LifeKeeper for Linux ソフトウェア・パッチのバージョンが同一であることが絶対条件です。

- **LifeKeeper for Linux NAS Recovery Kit** - NAS Recovery Kit は CD で提供されます。 Red Hat Package Manager（rpm）でパッケージされており、rpm を使用してインストール / アンインストールを行います。LifeKeeper for Linux NAS Recovery Kit CD には、次の rpm ファイルが格納されています。

  **steeleye-lkNAS**

- **Linux** ソフトウェア - クラスタ内の各サーバに、 **util-linux** パッケージがインストールされている必要があり、LifeKeeper および LifeKeeper NAS Recovery Kit の構成前に構成しておく必要があります。NAS Recovery Kit が正常に機能するには、バージョン 2.9u 以降の **util-linux** パッケージが必要です。

  LifeKeeper for Linux ソフトウェアの具体的なインストール / アンインストール方法については、LifeKeeper for Linux インストールガイド を参照してください。

# 6.9.3. NAS Recovery Kit 概要

## LifeKeeper for Linux NAS Recovery Kit

LifeKeeper for Linux NAS Recovery Kit の第 1 の目的は、LifeKeeper ユーザに対し、共有ストレージとデータレプリケーションに代わる代替ストレージ方式を提供することです。NAS Recovery Kit を使用すると、LifeKeeper 保護下のサーバまたはクライアント上に LifeKeeper リソース階層を作成できます。このリソース階層には、Network Attached Storage デバイスまたはクラスタ内の NFS サーバからエクスポートしたネットワークファイルシステム（NFS）をインポート（マウント）できます。エクスポートしたファイルシステムがマウントされたクラスタ内のノードで障害が検出されると、NAS Recovery Kit は、事前判別されたバックアップノードへのフェイルオーバを開始します。

そのため、エクスポートしたファイルシステムが LifeKeeper サーバまたはクライアントにマウントされたら、それを LifeKeeper 階層のストレージの基礎として完全に使用できます。

エクスポートしたファイルシステムをストレージメディアとして使用する場合、 LifeKeeper では、ファイルシステムがエクスポートされたサーバーの保護を必要としません。ただし、可用性を高めるには、ユーザが LifeKeeper for Linux NFS Server Recovery Kit を使用して、ファイルシステムがエクスポートされたサーバで障害が発生しないように保護することが推奨されます。

NAS Recovery Kit のリソース階層は、LifeKeeper Core 製品（steeleye-lk パッケージ）から 入手可能な現存の File System Recovery Kit を使用して作成されます。

NAS Recovery Kit にはいくつかの利点がありますが、最も重要な 2 つの利点は、コストのかかる共有ストレージデバイスを使用する必要がなくなる点と、複数ノードのクラスタ構成が可能になる点です。

## NAS Recovery Kit の制限事項

- このバージョンの NAS Recovery Kit では、NAS デバイスへのアクセスが失敗したときのローカルリカバリーをサポートしていません。障害が検出されると、デフォルトアクションによって、バックアップサーバへの階層の転送が開始されます。 リソース階層の構成に応じて、このアクションはハングプロセスにつながることがあります。ハングプロセスを回避するには、サーバーを停止してバックアップサーバーへのフェイルオーバーを強制するように、デフォルトアクションを変更します。デフォルトの切り替え動作を変更するには、LifeKeeper デフォルトファイルで LKNASERROR の設定を変更します。LKNASERROR の詳細については、このドキュメントのセクション「LifeKeeper for NAS Recovery Kit の設定」を参照してください。

- NAS Recovery Kit は、Network Attached Storage デバイスを保護しません。このキットの目的は、LifeKeeper ストレージオプションを Network Attached Storage 領域まで拡張することです。

- NAS Recovery Kitでは、NFSファイルシステムを異なるマウントポイントへ複数回マウントすることは許可されません。ファイルシステムが /etc/fstab ファイルで複数回検出されると、階層の作成試行は失敗します。

- NAS Recovery Kit は以下の形式の NAS デバイスまたは NFS をサポートしておりません。
  - export されているパスに":“が含まれている (例: 192.168.1.10:/export/pa:th)

- IPv6が使用されている (例: fc00::10:/export/data)

# 6.9.4. LifeKeeper for Linux NAS Recovery Kit の設定

本セクションでは、LifeKeeper for Linux NAS Recovery Kit の設定の詳細について説明します。NAS Recovery Kit の設定と管理を開始する前に考慮すべき事項についても記載しています。LifeKeeper Core リソース階層の設定方法については、LifeKeeper for Linux テクニカルドキュメンテーション を参照してください。

# 6.9.4.1. NAS 構成に関する考慮事項

LifeKeeper for Linux NAS Recovery Kit を操作する前に、次の事項を検討する必要があります。

1. エクスポートしたファイルシステムをマウントし、かつ NAS リソース階層を拡張するクラスタ構成内のサーバへ、NAS Recovery Kit をインストールします。ファイルシステムは、LifeKeeper で保護できる NFS サーバからエクスポートすることもできますし（推奨設定）、Network Attached Storage デバイスからエクスポートすることもできます。

2. このキットを適切に実行するために、サーバ名ではなくサーバの IP アドレスを使用して、エクスポートされた NFS ファイルシステムをマウントすること、およびファイルシステムを LifeKeeper 保護下に入れる前にマウント操作を実行することを強くお勧めします。さらに、LifeKeeper for Linux NFS Server Recovery Kit によって現在保護されているファイルシステムをマウントする場合は、NFS Server 階層を作成するために使用した IP アドレスを、LifeKeeper NAS サーバへファイルシステム をマウントするために使用することを強くお勧めします。

3. スプリットブレイン（つまり、クラスタ内の複数のノードにサービス中の保護された（ISP）階層がある）に関連する問題が発生しないようにするため、エクスポートしたファイルシステムへアクセスするために使用する同じネットワーク上のクラスタ内のノード間に、通信パスの 1 つを確立することを強くお勧めします。この推奨事項に従わないと、通信パスの障害が発生したときに、複数のノードで階層が ISP スプリットブレインになる可能性があります。スプリットブレインのシナリオからリカバリーするには、1 つの ISP 階層を除いたすべての階層をサービス休止状態にします。これにより、1 つのノードのみが、エクスポートされたファイルシステムへアクセスできるようになります。

4. NAS 階層をビルドするために使用するビルトインファイルシステムのリカバリーキットでは、フェイルオーバ条件でマウントされたファイルシステムを使用 している LifeKeeper 保護下に入っていないプロセスを検出したり、除去したりすることはできません。そのため、LifeKeeper 保護下に入れるプロセスのみが、NAS 保護下のファイルシステムを使用するように設定することを強くお勧めします。

5. LKNFSTIMEOUT という調整可能パラメータは、NFS マウントファイルシステムのステータスを判別するときに NAS Recovery Kit が使用するタイムアウトを秒単位で表します。この調整可能パラメータのデフォルト値は、2 分に設定されます。 LKNFSSYSCALLTO という調整可能パラメータは、マウントポイントのステータスを判別するときにアラームがシステムコールを中断するように NAS Recovery Kit が使用するタイムアウトを秒単位で表します。下記の式を使用して、この調整可能パラメーターの値を判別します。LKNFSSYSCALLTO の 3 倍の値に 5 を加算した値が、LKNFSTIMEOUT の値より小さくなければなりません。

6. 調整可能パラメータ LKNASERROR は、NAS デバイスへのアクセスが失敗したときに NAS Recovery Kit が実行するアクションを制御します。調整可能パラメータには、2 つの 値（ **switch** および **halt** ）があり、 **switch** がデフォルトです。値が **switch** に設定され、アクセスが失敗すると、NAS Recovery Kit は障害の検出時に、バックアップ･サーバへのリソース階層の転送を開始します。バックアップサーバーへのリソース階層の転送試行は、NAS リソースより上にあるリソースが NAS ファイルシステム上の項目へアクセスしようとすると、停止する場合があります。この問題を回避するには、調整可能値を **halt** に設定します。これにより、アクセス障害が検出されると、即時にシステムが停止します。このアクションによって、バックアップサーバへのすべてのリソース

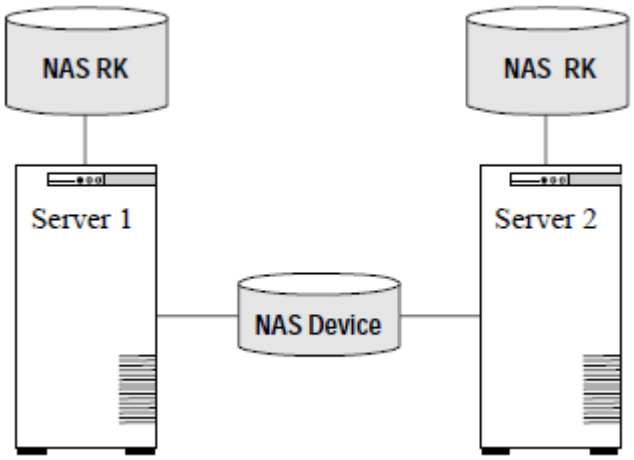
階層のフェイルオーバが強制されます。

7. ノード障害（すべてのコミュニケーションパスがダウンする）によるスプリットブレインが発生しないように、STONITH デバイスや Quorum/Witness パッケージの使用を推奨します。スプリットブレインが発生して、クラスタ内の複数のノードで NAS リソースが in service になるとデータ破損を引き起こす可能性があります。詳細は、LifeKeeper for Linux のテクニカルドキュメンテーションを参照してください。

# 6.9.4.2. NAS の構成例

フェイルオーバーが発生したときに、 **LifeKeeper for Linux NAS Recovery Kit** を使用している とどのような処理が行われるか、その例をいくつか示します。

## 構成 1：アクティブ／スタンバイ設定の例



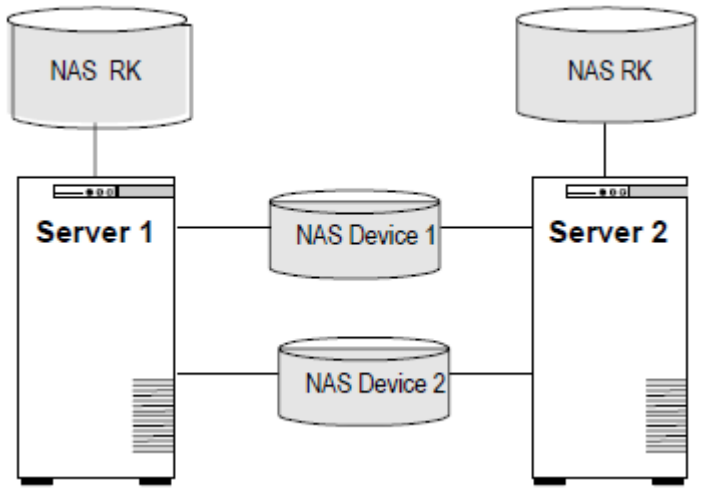

この構成では、サーバ 1 が NAS Recovery Kit ソフトウェアを実行しており、NAS デバイスからファイル･システムをインポート（マウント）したため、アクティブとみなされます。 サーバ 2 は別の処理を実行します。 サーバ 1 に障害が発生した場合、サーバ 2 が ファイル･システムへのアクセス権を獲得し、LifeKeeper のセカンダリー階層を使用して、 クライアントがファイルシステムへアクセスできるようにします。

**設定の注意点：**

- NAS ソフトウェアを両方のサーバーにインストールしておく必要があります。
- ファイルシステムが NAS デバイスからインポートされました。
- サーバ 1 がアクティブである間、サーバ 2 は NAS デバイス上のファイルおよびディレクトリにアクセスできません。

**注記：** アクティブ / スタンバイ設定では、サーバ 2 でも NAS Recovery Kit が実行されていることがありますが、他の NAS リソースは LifeKeeper によって保護されていません。

# 構成 2：アクティブ / アクティブ設定の例




アクティブ / アクティブ設定は、NAS Recovery Kit ソフトウェアをアクティブに実行し、 かつファイルシステムを NAS デバイスからインポートする複数のシステムで構成されます。

**設定の注意点：**

- NAS ソフトウェアを両方のサーバにインストールしておく必要があります。
- 最初に、サーバ 1 がファイルシステムをインポートし、サーバ 2 が別のファイ ルシステムをインポートします。切り替え時には、1 つのシステムで両方のファイルシステムをインポートできます。

# 6.9.5. NAS の LifeKeeper の設定作業

LifeKeeper for Linux NAS Recovery Kit の管理作業はすべて、LifeKeeper グラフィカルユーザインタフェース（GUI）で実行できます。LifeKeeper GUI では、NAS リソースの設定、管理、監視作業をガイド付きで行えます。

LifeKeeper for Linux NAS Recovery Kit の設定に関しては、次の作業が可能です。

- NAS リソース階層の作成 ：NAS リソース階層を作成します。
- NAS リソース階層の削除 ：NAS リソース階層を削除します。
- NAS リソース階層の拡張 ：NAS リソース階層をプライマリサーバーからバックアップサーバーに拡張します。
- NAS リソース階層の拡張解除 ：LifeKeeper クラスタの 1 つのサーバの NAS リソース階層を拡張解除（削除）します。
- 従属関係の作成 ：既存のリソース階層と別のリソースインスタンスとの間に子従属関係を作成し、クラスタ内のすべての対象サーバに従属関係の変化を通知します。
- 従属関係の削除 ：リソースの従属関係を削除して、クラスタ内のすべての対象サーバ に従属関係の変化を通知します。
- In Service ：リソース階層をアクティブにします。
- Out of Service ：リソース階層を非アクティブにします。
- 表示 / 編集 ：リソース階層のプロパティを表示または編集します。

**注記：** このセクションの残りの部分では、設定作業はすべて **[Edit]** メニューから実行することになっています。この設定作業の大半は次の方法でも実行できます。

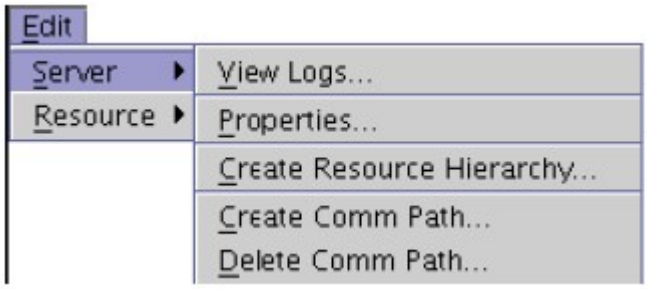
1. ツールバーから状態表示の左ペインにあるグローバルリソースを右クリック。
2. 状態表示の右ペインにあるリソースインスタンスを右クリック。

   ※右クリックの方法を使用すると、 **[Edit]** メニューを使用したときに必要な情報を入力しないですみます。

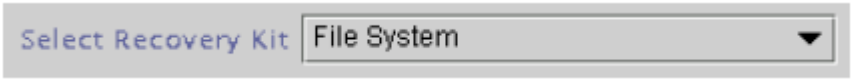
# 6.9.5.1. NAS リソース階層の作成

プライマリサーバーで以下を実行して、**[リソース作成ウィザード]** を開始します。

1. **[Edit] > [Server] > [Create Resource Hierarchy]** を選択します。



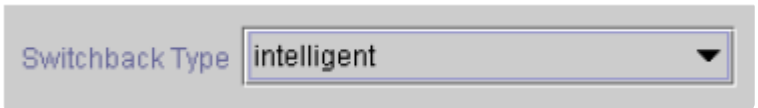

2. **[リカバリーキットの選択]** ダイアログが表示されます。ドロップダウンリストから **[ファイルシステム]** オプションを選択します。NAS Resource Hierarchy を、NFS マウントシステムファイルを使用して作成したファイルシステム階層にします。




**[Next]** をクリックして次に進みます。

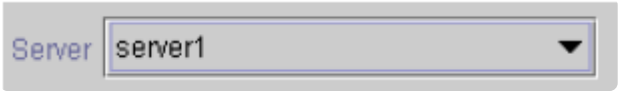
注記：階層作成手順のどの段階でも、**[Cancel]** ボタンをクリックすると、作成プロセス全体がキャンセルされます。

3. **[スイッチバックタイプ]** ダイアログが表示されます。スイッチバックタイプによって、フェイルオーバー後、バックアップサーバで NAS リソースがサービス中（アクティブ）になったとき、NAS リソースをプライマリサーバーにスイッチバックする方法が決まります。スイッチバックタイプは intelligent または automatic です。インテリジェントスイッチバックの場合、リソースをプライマリーサーバに戻すために管理者の操作が必要ですが、自動スイッチバックの場合は、プライマリサーバーがオンラインに戻り、LifeKeeper との通信パスが再確立されるとすぐにスイッチバックが実行されます。




**[Next]** をクリックして次に進みます。

4. **[サーバ]** ダイアログが表示されます。NAS リソースを作成するサーバの名前を選択します（通常は、これがプライマリーサーバです）。クラスター内のすべてのサーバーがドロップダウンリストボックスに表示されます。



**[Next]** をクリックして次に進みます。

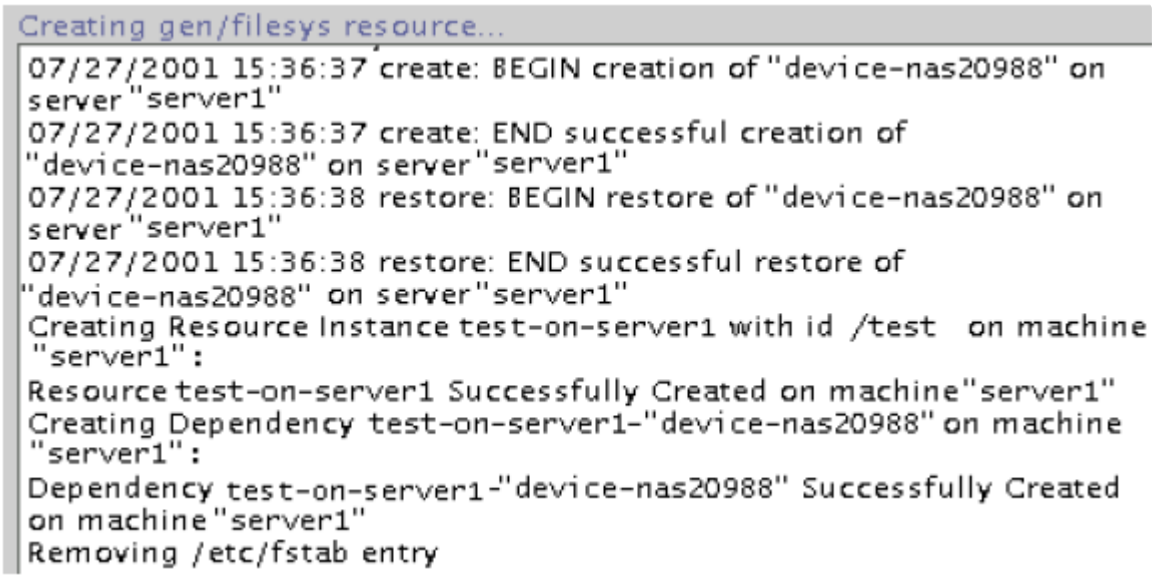
5. NAS (File System) Resource Hierarchy で保護する **[マウントポイント]** パスを選択します。すべての「ローカル」ファイルシステム（つまり、共有ストレージを使用するファイルシステム） および NFS マウントファイルシステムがリストされます。ドロップダウンリストから[NFS マウントファイルシステム]オプションを選択します。

**[Next]** をクリックして次に進みます。

6. **[ルートタグ]** ダイアログに、ターゲットサーバー（つまり上記で選択したサーバー）上のリソースインスタンスの固有名が自動的に表示されます。デフォルトを 受け入れることもできますし、文字、数字、および特殊文字-、_、.、または/で構成 された固有のタグ名を入力することもできます。

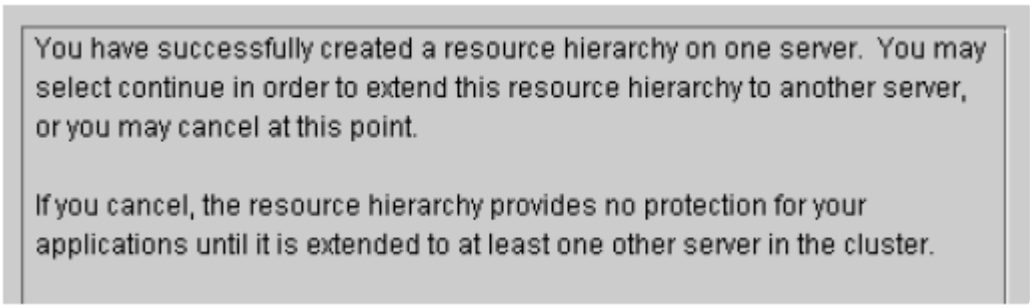
**[Create Instance]** をクリックします。

7. 階層の作成を開始することを示す情報ボックスが表示されます。

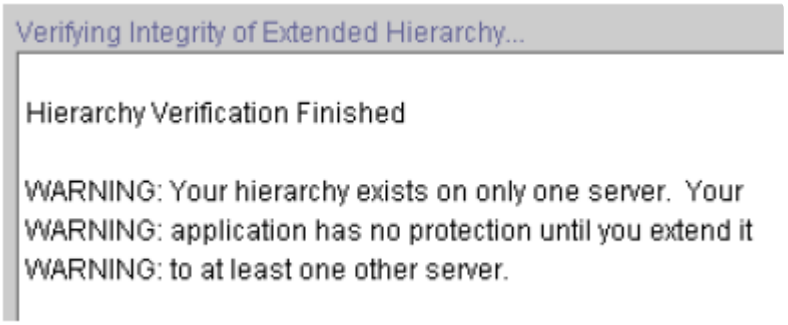



**[Next]** をクリックして次に進みます。

8. NAS リソース階層の作成に成功したことを示す情報ボックスが表示されます。リソース階層を LifeKeeper で保護するには、クラスタ内の別のサーバにリソース階層を 拡張する必要があります。




リソースを拡張する場合は、 **[Continue]** をクリックします。 後でリソースを拡張する場合は、 **[Cancel]** をクリックします。




9. **[Done]** をクリックして、[リソース階層の作成]メニューを終了します。 を LifeKeeper で保護するには、クラスター内の別のサーバーにリソース階層を拡張する必要があります。

# 6.9.5.2. NAS リソース階層の削除

プライマリーサーバーで次の作業を行います。LifeKeeper 構成内のすべてのサーバから NAS リソースを削除するには、次の手順を実行します。

1. **[LifeKeeper GUI]** メニューから **[Edit]** を選択し、次に **[Resource]** を選択します。ドロップダウンメニューから、 **[Delete Resource Hierarchy]** を選択します。

2. NAS リソース階層を削除する **ターゲットサーバー** の名前を選択します。

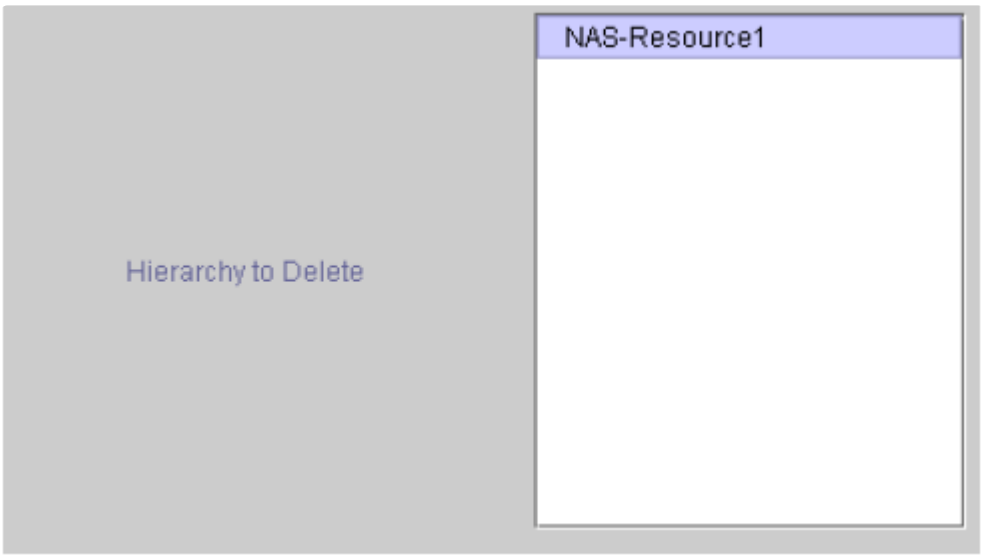
   **注記：** 左側のペインからグローバルリソースを右クリックするか、右側のペインから個々のリソースインスタンスを右クリックして **[リソース階層の削除]** を選択した場合、このダイアログ・ボックスは表示されません。

   **[Next]** をクリックして次に進みます。

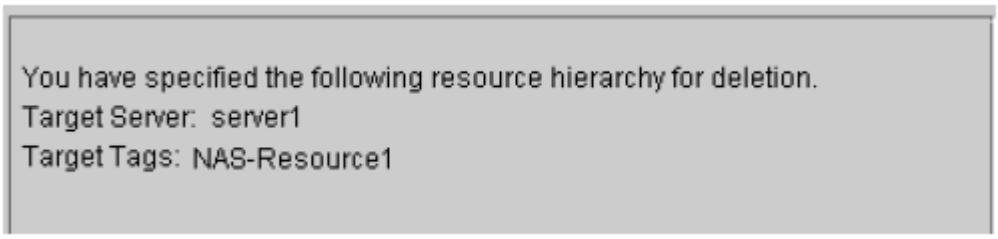
3. **[削除する階層]** を選択します。削除するリソース階層を特定して、強調表示にします。

   **注記：** 左側のペインからグローバルリソースを右クリックするか、右側のペインから個々のリソースインスタンスを右クリックして **[リソース階層の削除]** を選択した場合、このダイアログボックスは表示されません。

   **[Next]** をクリックして次に進みます。

4. 選択したターゲットサーバーと、削除の対象として選択した階層を確認する情報ボ ックスが表示されます。




**[Delete]** をクリックして次に進みます。

5. File System NAS リソースインスタンスの削除に成功したことを示す情報ボックス が表示されます。

6. **[Done]** をクリックして、 **[リソース階層の削除]** メニューを終了します。

# 6.9.5.3. NAS 階層の拡張

階層は、作成後に、クラスタ内の別のサーバに拡張する必要があります。リソースインスタンスを拡張するには、次の 3 つの方法が可能です。

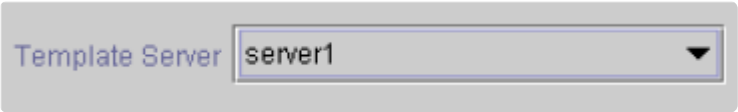
1. NAS リソース階層の作成に成功したときに、**[Continue]** を選択して、バックアップサーバーにリソース階層を拡張する処理に進むことができます。

2. LifeKeeper GUI の左ペインまたは右ペインで拡張されていない階層を右クリックします。

3. LifeKeeper GUI で、ドロップダウン・メニューから **[Edit] > [ Resource ] > [ Extend Resource Hierarchy ]** を選択します。選択すると、**[リソース階層の拡張]** ウィザードが起動されます。**[ Extend Resource Hierarchy ]** オプションの **[ Accept Defaults ]** ボタンは、LifeKeeper の **[リソース階層の拡張]** の既定値をよく知っていて、値の入力や確認をしないで素早く LifeKeeper リソース階層を拡張したいユーザー向けです。GUI ダイアログを使用して対話的に段階を追って LifeKeeper リソース階層を拡張する場合は、**[ Next ]** ボタンを選択します。

   a. 最初に表示されるダイアログボックスでは、NAS リソース階層が現在サービス中であるテンプレートサーバーを選択します。**[テンプレートサーバー]** で選択するテンプレートサーバーと、次のダイアログボックスの **[拡張するタグ]** で選択する拡張するタグが、サービス中（アクティブ）のリソース階層を表すことを忘れないでください。選択したテンプレートサーバーでサービス中になっていない リソースタグを選択すると、エラーメッセージが表示されます。このダイアログのドロップダウンボックスには、クラスタ内の全サーバの名前が示されます。

   **注記：** NAS リソース階層作成後、引き続きリソース階層の拡張作業に入った場合は、作成段階でテンプレートサーバーが特定されているため、このダイアログボックスは表示されません。これは、GUI ウィンドウの左ペインの NAS リソースアイコンまたは右ペインの NAS (File System) リソースボックスを右クリックして、**[リソース階層の拡張]** を選択した場合も同様です。

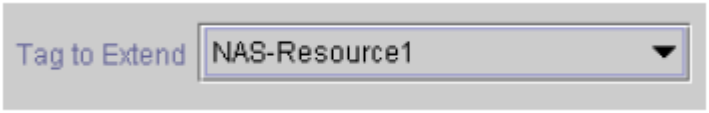
   **注記：** 階層拡張の途中で **[Cancel]** ボタンをクリックすると、拡張処理が取り消されます。ただし、すでにリソースを別のサーバに拡張している場合は、明示的に拡張解除するまで、そのインスタンスの拡張は有効です。

   **[Next]** をクリックして次に進みます。

   b. **[拡張するタグ]** を選択します。これは、テンプレートサーバーからターゲットサーバーに拡張する NAS インスタンスの名前です。ウィザードのドロップダウンリストボックスには、テンプレートサーバー上に作成されているすべてのリソースが表示されます。

   注記： ここでも、NAS リソース階層の作成後すぐにリソース階層の拡張作業に進んだ場合は、作成段階ですでにリソースのタグ名が特定されているため、このダイアログボックスは表

示されません。これは、GUI ウィンドウの左ペインの NAS (File System) リソースアイコンまたは右ペインの NAS (File System) リソースボックスを右クリックして、[リソース階層の拡張]を選択した場合も同様です。

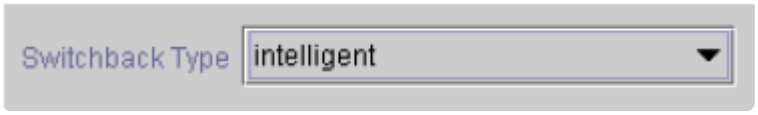



**[Next]** をクリックして次に進みます。

c. NAS リソース階層を拡張するターゲットサーバーを選択します。

**[Next]** をクリックして次に進みます。

d. **[スイッチバック・タイプ]** ダイアログが表示されます。スイッチバックタイプによって、フェイルオーバー後、バックアップサーバーで NAS リソースがサービス中（アクティブ）になったとき、NAS リソースをプライマリーサーバーにスイッチバックする方法が決まります。スイッチバックタイプは intelligent または automatic です。インテリジェントスイッチバックの場合、リソースをプライマリサーバーに戻すために管理者の操作が必要ですが、自動スイッチバックの場合 は、プライマリーサーバーがオンラインに戻り、LifeKeeper との通信パスが再確立されるとすぐにスイッチバックが実行されます。




**[Next]** をクリックして次に進みます。

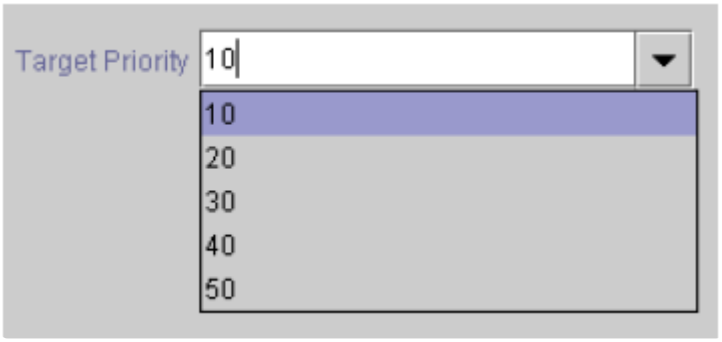
e. テンプレートの優先順位を選択または入力します。これはサーバ＾で現在サービス中の NAS 階層の優先順位です。優先順位は、1～999 の範囲で未使用の値が有効で、小さい数字ほど優先順位が高くなります（数字 1 が最高の優先順位に相当し ます）。拡張操作時には、別のシステムで既に使用中の優先順位をこの階層に対 して指定することはできません。デフォルト値を推奨します。

**注記：** このフィール ドは階層を最初に拡張するときだけ表示されます。

**[Next]** をクリックして次に進みます。

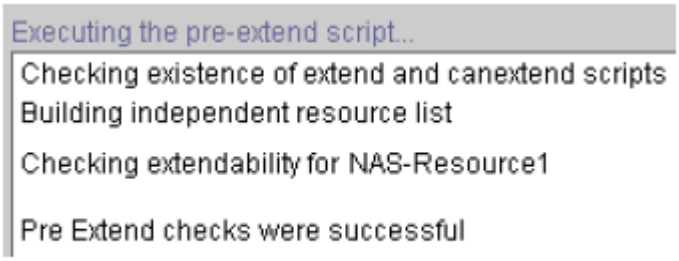
f. ターゲットの優先順位を選択または入力します。これは、別サーバにある同等の階層に対する新しく拡張する NAS 階層の優先順位です。1～999 の範囲の未使用の値が有効で、リソースの連鎖フェイルオーバーシーケンスにおけるサーバの優先順位を示します。数値が小さいほど優先順位は高くなります（1 は最高の優先 順位を表わします）。LifeKeeper のデフォルトでは、階層が作成されたサーバに「1」が割り当てられることに注意してください。優先順位は連

続している必要はありませんが、特定のリソースについて 2 つのサーバに同じ優先順位を割り当てることはできません。




**[Next]** をクリックして次に進みます。

g. 環境がチェックされ、このリソースを拡張するのに必要な要件がすべて満たされていることを示す情報ボックスが表示されます。満たされていない要件がある場合、 **[Next]** ボタンが使用できなくなり、 **[戻る]** ボタンが有効になります。




リソースの拡張の設定を変更する場合は、 **[Back]** ボタンをクリックします。 後でリソースを拡張する場合は、[キャンセル]をクリックします。リソース階層の拡張を実行する場合は、 **[Next]** をクリックします。

**[Finish]** をクリックして、NAS リソース･インスタンスの拡張に成功したことを確認します。

4. **[Done]** をクリックして、 **[リソース階層の拡張]** メニューを終了します。

**注記：** 必ず両方のサーバで新しいインスタンスの機能をテストしてください。

# 6.9.5.4. NAS 階層の拡張解除

1. LifeKeeper GUI メニューで、 **[Edit] > [Resource] > [Unextending Resource Hierarchy]** を選択します。

2. NAS リソースを拡張解除する **[ターゲットサーバー]** を選択します。リソースが現在サービス中（アクティブ）のサーバは選択できません。

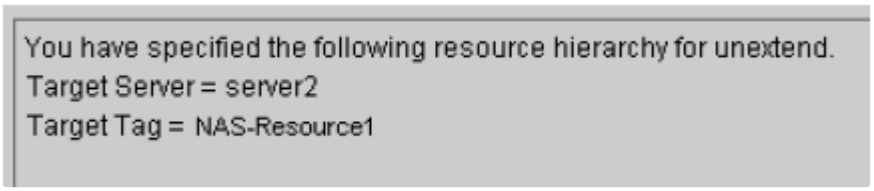
   **注記：** 左側のペインからグローバルリソースを右クリックするか、右側のペインから個々のリソースインスタンスを右クリックして拡張解除作業を選択した場合、 このダイアログボックスは表示されません。

   **[Next]** をクリックして次に進みます。

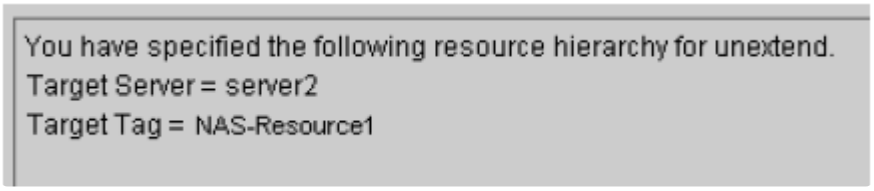
3. **[拡張解除する階層]** を選択します。

   **注記：** 左側のペインからグローバルリソースを右クリックするか、右側のペインから個々のリソースインスタンスを右クリックして拡張解除作業を選択した場合、 このダイアログボックスは表示されません。

   **[Next]** をクリックして次に進みます。

4. 拡張解除するために選択したターゲットサーバーと NAS リソース階層を確認する情報ボックスが表示されます。

   **[Unextend]** をクリックします。

5. NAS リソースの拡張解除に成功したことを示す別の情報ボックスが表示されます。

6. **[Done]** をクリックして、 **[リソース階層の拡張解除]** メニューを終了します。

# 6.9.5.5. NAS リソース階層のテスト

NAS リソース階層のテストは、手動でサーバを切り替えて、プライマリーサーバーからバックアップサーバーへのリソースインスタンスのフェイルオーバをシミュレートすることによって行えます。

## LifeKeeper GUI からの手動切り替えの実行

LifeKeeper GUI で **[Edit] > [Resource] > [In Service]** を選択すると、手動切り替えを開始できます。たとえば、バックアップサーバーでサービス中リクエストが実行されると、NAS リソース階層がバックアップサーバー側でサービス中になり、プライマリーサーバー側ではサービス休止になります。この時点で、元のバックアップサーバーがプライマリーサーバーになり、元のプライマリーサーバーがバックアップサーバーになります。

**[Our of Service]** を実行した場合、リソース階層はサービス休止になりますが、別のサーバーでサービス中にはなりません。

# 6.9.6. NAS トラブルシューティング

| 症状 | 考えられる要因 |
|---|---|
| LifeKeeper のフェイルオーバ操作 がアンマウント・ビジー・エラーで失敗しました。 | NAS 階層をビルドするために使用するファイル・システム・キットでは、フェイルオーバ条件でマウントされたファイル・システムを使用している LifeKeeper 保護下に入っていないプロセスを検出したり、除去したりすることはできません。そのため、LifeKeeper 保護下に入れるプロセスのみが、NAS 保護下のファイル・システムを使用するように設定することを強くお勧めします。この障害が発生したら、ファイル・システムを使用してプロセスを識別し、消滅させる必要があります。 **fuser -m** コマンドを使用すると、ファイル・システムへ現在アクセスしているプロセスを判別できます。この使用法の詳細については、 **fuser** のマニュアル・ページを参照してください。 |
| LifeKeeper が、サーバ名を使用してマウントされたファイル・システムの ローカル・リカバリーを行います。 | NAS Recovery Kit の保護下にあるファイル・システムが IP アドレスではなくホスト名を使用してマウントされたら、 NAS リソースの作成後に、LifeKeeper が次のようなメッセージをログに記録します。<br><br>`. . . WARNING: Mon Aug 26 11:27:01 2002:`<br><br>LifeKeeper protected filesystem resource “tmp/mnt-on-tom.brown.com” (/tmp/mnt) is in service but not mounted<br><br>. . . Attempting Local Recovery of resource<br><br>LifeKeeper は、この時点で IP アドレスを使用してファイル・システムを再マウントします。ただし、問題が発生したら、LifeKeeper は NAS リソースをバックアップ・サーバへフェイルオーバするか（拡張された場合）、またはリソースをサービス休止状態にします（リソースが拡張されていない場合）。<br><br>**推奨される対策：** ローカル・リカバリーが成功したら、それ以上処置は必要ありません。ただし、ローカル・リカバリーが失敗したら、次の操作を行う必要があります。<br><br>1. LifeKeeper で NAS リソースを削除します。<br><br>2. ホスト名ではなく IP アドレスを使用してファイル・システムを再マウントします。<br><br>3. NAS リソースを作成し直します。 |

# 6.9.6.1. NAS エラーメッセージ

本セクションでは、LifeKeeper NAS リソース階層の作成、拡張時、またはリソースの削除、復元時に表示される可能性のあるメッセージの一覧を示します。必要に応じて、エラーの原因およびエラー状態を解消するために必要な処置についても説明しています。

その他の LifeKeeper コンポーネントからもメッセージが表示される可能性があります。 そのような場合は、適切な LifeKeeper コンポーネントのマニュアルを参照してください。

このセクションのメッセージは、次のトピックに分類されています。

- 共通エラー・メッセージ
- 階層の作成
- 階層の拡張
- 階層の削除、復元、リカバリー

## 共通エラーメッセージ

| エラー番号 | エラーメッセージ |
|---|---|
| 000002 | Usage error （使用法のエラー） |
| 000010 | Error getting resource information （リソース情報取得時のエラー） |
| 000011 | Both Tag and ID name not specified （タグ名および ID 名が指定されていません） |
| 000019 | Resource not found on local server （ローカル・サーバにリソースが見つかりません） |
| 000022 | END failed hierarchy <tag name> in service on server <server name> （サーバ <サーバ名> でサービス中の階層 <タグ名> の END が失敗しました） |
| 000026 | END failed ACTION for <tag name> on server <server name> due to <signal> signal （<シグナル> シグナルのためにサーバ <サーバ名> の <タ グ名> ACTION の END が失敗しました） |

## 階層の作成

| エラー番号 | エラーメッセージ |
|---|---|
| 000012 | Usage error（使用法のエラー） |
| 000013 | Error getting resource information（リソース情報取得時のエラー） |
| 000014 | Resource with either matching tag <tag name> or ID exists （タグ <タグ名> または ID が一致するリソースが存在します） |
| 000015 | ins_create failed on server <server name> （サーバ <サーバ名> で ins_create に失敗しました） |

| | |
|---|---|
| 000018 | Error creating resource <tag name> on server <server name> （サーバ <サーバ名> でリソース <タグ名> の作成エラー） |
| 000021 | Removing resource instance < tag name> from server <server name> due to an error during creation （作成時のエラーにより、リソース・インスタンス <タグ 名> をサーバ <サーバ名> から削除しています） |
| 000023 | Error bringing resource < tag name> in service on server <server name> （サーバ <サーバ名> でリソース <タグ名> をサービス中 にするときにエラー） |
| 000024 | Failed resource creation of resource < tag name> on server <server name> （サーバ <サーバ名> でリソース <タグ名> の作成に失敗しました） |
| 000027 | Removing file system dependency from <parent tag> to <child tag> on server <server name> due to an error during creation （作成時のエラーにより、サーバ <サーバ名> の <親タグ> から <子タグ> へのファイル・システム従属関係を削除し ています） |
| 000028 | Removing file system hierarchy <filesys tag> created by <parent tag> on server <server name> due to an error during creation （作成時のエラーにより、サーバ <サーバ名> の <親タグ> によって作成されたファイル・システム階層 <ファイルシステム・タグ> を削除しています） |
| 000029 | Switchback type mismatch between parent <parent tag> and child <child tag> on server <server name> （サーバ <サーバ名> の親 <親タグ> と子 <子タグ> のスイ ッチバック・タイプが一致していません）<br><br>**処置：** スイッチバック・タイプの不一致は予想外の動作を 引き起こすことがあります。ins_setas コマンドを使用して スイッチバック・タイプを手動で変更することによって、 この不一致をなくすことができます。 |
| 000030 | create: tag name not specified （作成時：タグ名が指定されていません）<br><br>または<br><br>extend: tag name not specified （拡張時：タグ名が指定されていません） |

## 階層の拡張

| エラー番号 | エラーメッセージ |
|---|---|
| 000003 | Template resource < tag name> on server <server name> does not exist （サーバ <サーバ名> にテンプレート・リソース <タグ名> が存在しません） |
| 000004 | Template resource < tag name> cannot be extended to server <server name> because it already exists there （テンプレート・リソース <タグ名> がサーバ <サーバ名> に すでに存在するため拡張できません） |
| 000005 | Cannot access canextend script on server <server name> （サーバ <サーバ名> の canextend スクリプトにアクセスできません） |
| 000006 | Cannot access extend script <path to extend> on server <server name> （サーバ <サーバ名> の extend スクリプト <extend へのパス> にアクセスできません） |

| | |
|---|---|
| 000007 | Cannot access depstoextend script <path to depstoexend> on server <server name> （サーバ <サーバ名> の depstoextend スクリプト <depstoextend へのパス> にアクセスできません） |
| 000008 | Cannot extend resource < tag name> to server <server name> （サーバ <サーバ名> にリソース <タグ名> を拡張できません） |
| 000009 | Either <templatesys> or <templatetag> argument missing （<テンプレート・システム> または <テンプレート・タグ> 引数が指定されていません） |
| 000014 | Resource with either matching tag <tag name> or ID exists （タグ <タグ名> または ID が一致するリソースが存在します） |
| 000015 | ins_create failed on server <server name> （サーバ <サーバ名> で ins_create に失敗しました） |
| 000018 | Error creating resource <tag name> on server <server name> （サーバ <サーバ名> でリソース <タグ名> の作成エラー） |
| 000025 | END failed resource extension of < tag name> on server <server name> due to a "<signal>" signal - backing out changes made to server （"<シグナル>" シグナルのためにサーバ <サーバ名> の <タグ名> リソース拡張の END が失敗しました - サーバに加えた変更を元に戻しています） |
| 000030 | create: tag name not specified （作成時：タグ名が指定されていません）<br><br>または<br><br>extend: tag name not specified （拡張時：タグ名が指定されていません） |

### 復元

| エラー番号 | エラーメッセージ |
|---|---|
| 000023 | Error bringing resource < tag name> in service on server <server name> （サーバ <サーバ名> でリソース <タグ名> をサービス中にするときにエラー） |

### リソース監視

| エラー番号 | エラーメッセージ |
|---|---|
| 000001 | Calling sendevent for resource < tag name> on server <server name> （サーバ <サーバ名> でリソース <タグ名> の sendevent を呼び出 しています） |

## NAS Recovery Kit のエラーメッセージ

| エラー番号 | エラーメッセージ |
|---|---|

| | |
|---|---|
| 107001 | Creation of NAS device with tag id <tag id> on server <LifeKeeper server name> failed. （サーバ <LifeKeeper サーバ名> でタグ ID <タグ ID> を使用してNAS デバイスを作成しようとしましたが、失敗しました。） |
| 107002 | Error getting list of IP addresses for NFS server device <NFS server name> on server <LifeKeeper server name>.（サーバ <LifeKeeper サーバ名> で NFS サーバ・デバイス <NFSサーバ名> の IP アドレスのリストを取得中にエラーが発生しました。） |
| 107003 | Error attempting to find active address to NFS server <NFS server name> on server <LifeKeeper server name>. （サーバ <LifeKeeper サーバ名> で NFS サーバ <NFS サーバ名> の アクティブ・アドレスを検出試行中にエラーが発生しました。） |
| 107004 | Error in format of device ID <resource device>. （デバイス ID <リソース・デバイス> のフォーマット・エラー。） |
| 107005 | Cannot bring NAS resource <tag id> in service on server <LifeKeeper server name>. （サーバ <LifeKeeper サーバ名> で NAS リソース <タグ ID> をサービス中の状態にできません。）<br><br>**処置：** 問題を修正したあと、手動でリソースをサービス中にしてください。 |
| 107006 | create: Device not specified. （作成：デバイスが指定されていません。） |
| 107007 | Null Device returned by getId on <LifeKeeper server name> （<LifeKeeper サーバ名> で getId によってヌル・デバイスが戻されました。） |
| 107008 | Cannot open /etc/mtab file on <LifeKeeper server name>. （<LifeKeeper サーバ名> で/etc/mtab ファイルを開けません。） |
| 107009 | Illogical settings for NAS defaults on <LifeKeeper server name>. Using defaults of 120 for LKNFSTIMEOUT and 5 for LKNFSSYSCALLTO. （<LifeKeeper サーバ名> 上の NAS デフォルトの設定が非論理的 です。LKNFSTIMEOUT には 120 のデフォルトを、 LKNFSSYSCALLTO には 5 を使用しています。）<br><br>**処置：** LKNFSSYSCALLTO の 3 倍の値に 5 を加算した値が LKNFSTIMEOUT より小さくなるように、NAS デフォルト値をリセットします。 |
| 107010 | Error: detected conflict in expected tag name <tag id> on target machine <LifeKeeper server name>. （エラー：ターゲット・マシン<LifeKeeper サーバ名> 上の予想さ れたタグ名 <タグ ID> で競合が検出されました。）<br><br>**処置：** 競合するリソースを削除し、階層を再拡張してください。 |
| 107011 | Error: mkdir of "/tmp/nas_mntpt.2915" on "mouse" failed: "permission denied". （エラー："mouse"上の"/tmp/nas_mntpt.2915"の mkdir が失敗しました："権限拒否"） |
| 107012 | Error: Exported file system <NFS exported file system name> cannot be accessed on <server name>. （エラー：<サーバ名> で、エクスポートされたファイル・システム <NFS エクスポート・ファイル・システム名> にアクセスできません。）<br><br>考えられる原因<br><br>- LifeKeeper ノードが、NFS サーバのエクスポートされたシステムのリストに含まれていない。 |

| | |
|---|---|
| | - エクスポートされたシステムのリストに、 **showmount** コマンド で表示されない矛盾したエントリが含まれている。 (つまり、エクスポートされたシステムのリストによってシステムが全システムと特定のシステムの両方にエクスポートされる場合、 **showmount** では特定のシステムのみが報告されます。)<br><br>**処置：** エクスポートされたファイル･システムのアクセス問題を訂正し、階層を再拡張してください。 |
| 107013 | Error: Mount authorization check for "172.25.113.25:/ export" on "fred" appears to be hung. Exiting. (エラー："fred"での"172.25.113.25:/ export"のマウント認証チェックが停止した可能性があります。終了します。)<br><br>**処置：** アクセス問題を訂正し、階層を再拡張してください。 |
| 107015 | Exported file system <tag> cannot be accessed on <LifeKeeper server name>. (サーバ <LifeKeeper サーバ名> でエクスポートされたファイルシステム <tag名> にアクセスできませんでした。) |

# 6.9.6.2. LifeKeeper GUI 関連のエラー

| エラー番号 | エラーメッセージ |
|---|---|
| 104901 | The mount point %s is mounted （マウント･ポイント %s はマウントされています）<br><br>**処置：** マウントされていないマウント･ポイントを指定してくだ さい。 |
| 104902 | The mount point %s is not an absolute path（マウント･ポイント %s が絶対パスでありません）<br><br>**処置：** スラッシュで始まるマウント･ポイントを指定してくださ い。 |
| 104903 | The mount point %s is not empty.（マウント･ポイント %s が空でありません）<br><br>**処置：** 存在しないマウント･ポイントまたは空のマウント･ポイ ントを指定してください。 |

# 6.10. NFS Recovery Kit 管理ガイド

## NFS Server Recovery Kit のドキュメンテーション

SIOS Protection Suite for Linux NFS Server Recovery Kit は、SIOS Protection Suite 環境でネットワークファイルシステム(NFS)ソフトウェアの障害回復能力を高めます。プライマリ NFS サーバに障害が発生した場合には、人手の介入なしに迅速に指定のバックアップサーバでシステムを復旧することが可能です。

SIOS Protection Suite for Linux NFS Server Recovery Kit の概要

## SIOS Protection Suite/LifeKeeper のドキュメンテーション

以下は、SIOS Technology Corp が発行している SIOS Protection Suite for Linux 関連ドキュメントの一覧です。

- SPS for Linux テクニカルドキュメンテーション
- SPS for Linux リリースノート
- SIOS Technology Corp. ドキュメンテーション

## 参考ドキュメント

以下は SIOS Protection Suite NFS Server Recovery Kit に関連する参考ドキュメントの一覧です。

- NFS オンラインドキュメンテーション
- *Managing NFS and NIS, Hal Stern, O'Reilly & Associates, Inc. 1991*

# 6.10.1. NFS Server Recovery Kit 概要

NFS Server Recovery Kit は、Filesystem Recovery Kit (steeleye-lk パッケージの一部として提供) および IP Recovery Kit (steeleye-lkIP) と階層的に連携して、高可用性 NFS サービスを実現します。

NFS Server Recovery Kit は、IP リソースと共有マウントポイントを含むファイルシステムリソースがクラスター内の同じサーバーで常に In Service であることを保証します。LifeKeeper が保護する IP リソースを使用してファイルシステムをマウントするクライアントは、実際のエクスポートサービスが手動で、または障害に対する対処としてクラスター内のサーバー間で切り替えられても、事実上中断なしにボリューム上のファイルの処理を継続できます。NFSセッションの復帰時間はクライアント - サーバー間の状態に依存します。例としてNFSv3においては、TCPのプロトコルタイムアウトはUDPに比べて長くなります。NFSで使用する適切なトランスポート層プロトコルを決定するには、OSベンダーからの推奨や各トランスポート層プロトコルのメリットとデメリット、ご利用の環境を踏まえてご検討ください。

※NFS v4 においては、OSベンダーが TCP トランスポートの使用を強く推奨しているため、NFS Server Recovery Kit でも NFS v4 については TCP トランスポートのみ動作検証しています。

スイッチオーバーまたはフェイルオーバーを実行している間、ファイルシステム上のファイルはすべて一時的に使用できなくなりますが、リソースの転送が完了すると再び使用可能になり、クライアント側に切り替えは意識されません。スイッチオーバーの場合、これには 5 ～ 30 秒かかります。フェイルオーバーの場合、リカバリー時間は、ファイルシステムの修復にかかる時間に左右されます。ディスクボリュームは、障害に極めて強く、数秒で修復が可能なジャーナリングファイルシステム (JFS) でフォーマットすることを強く推奨します。

LifeKeeper v9.6.0 から NFS v4 擬似ファイルシステムはサポートされなくなりました。fsid=0 のエクスポートポイントおよびそのサブディレクトリーのエクスポートポイントは LifeKeeper で保護できません。LifeKeeper v9.6.0 より前のバージョンで fsid=0 のリソースを作成している場合は、リソースを削除してから LifeKeeper をアップグレードしてください。

OS のデフォルト設定では、NFS v3 も v4 もどちらも有効になっています。クライアントからマウントする時のオプションでバージョンを指定してください。

# 6.10.2. NFS Server Recovery Kit 要件

LifeKeeper NFS Server Recovery Kit をインストールして設定する前に、次の要件を満たしていることを確認します。

- **サーバー** この Recovery Kit には、LifeKeeper for Linux テクニカルドキュメンテーション および LifeKeeper for Linux リリースノート で説明されている要件に従って設定した 2 台以上のサーバーが必要です。サポート対象の Linux ディストリビューションについては、LifeKeeper for Linux サポートマトリックス を参照してください。

- **LifeKeeper ソフトウェア** クラスターの各サーバーにインストールする LifeKeeper ソフトウェアのバージョン、および、適用する LifeKeeper ソフトウェアパッチのバージョンが同一であることが絶対条件です。具体的な LifeKeeper の要件については、LifeKeeper for Linux テクニカルドキュメンテーション および LifeKeeper for Linux リリースノート を参照してください。

- **LifeKeeper IP Recovery Kit** 各サーバーの LifeKeeper IP Recovery Kit のバージョンは同じにする必要があります。

- **IP ネットワークインターフェース** 各サーバーは、イーサーネット TCP/IP をサポートするネットワークインターフェースを 1 つ以上必要とします。IP リソースのスイッチオーバーが正しく動作するには、ローカルネットワークに接続されているユーザーシステムが標準の TCP/IP 仕様に準拠している必要があります。

> 注意：各サーバーが必要とするネットワークインターフェースが 1 つだけであっても、異種メディア要件、スループット要件、単一障害点の排除、ネットワークのセグメンテーションといった多くの理由のために、複数のインターフェースを使用すべきです。

- **TCP/IP ソフトウェア** 各サーバーは TCP/IP ソフトウェアも必要とします。

- **NFS ソフトウェア** LifeKeeper インストールサポートセットアップスクリプトは HA クラスター環境で使用する設定を行います。各サーバーでは、LifeKeeper NFS Server Recovery Kit の設定前に、次のソフトウェアをインストールして設定しておく必要があります。各サーバーに同じバージョンをインストールする必要があります。
  - rpcbind
  - rpc.idmapd
  - gssproxy または rpc.svcgssd
  - rpc.nfsd
  - rpc.mountd

LifeKeeper NFS Server Recovery Kit をインストールまたは削除する具体的な方法については、LifeKeeper for Linux インストールガイド を参照してください。

# 6.10.3. NFS Server Recovery Kit 設定上の考慮事項

下記セクションでは、典型的な LifeKeeper NFS 設定の例も含めて NFS Server Recovery Kit の設定と管理を開始する前に考慮すべき事項についても記載しています。

LifeKeeper Core のリソース階層の設定方法については、 LifeKeeper for Linux テクニカルドキュメンテーション を参照してください。

- LifeKeeper における NFS サーバーの設定
- NFS 固有の構成に関する考慮事項
- NFS 構成例

# 6.10.3.1. LifeKeeper における NFS サーバーの設定

本セクションでは、NFS Server Recovery Kit の設定と管理を開始する前に考慮すべき事項および典型的な LifeKeeper NFS 設定例について記載しています。

LifeKeeper Core のリソース階層の設定に関する手順については、SPS for Linux テクニカルドキュメンテーション を参照してください。

## NFS

次の表は、NFS Server Recovery Kit で重要な NFS ファイル、コマンド、デーモンについて説明しています。

| NFS コンポーネント | 説明 |
|---|---|
| *exports(5) (/etc/exports)* | NFS クライアントにエクスポートされたファイルシステムのアクセス制御リストです。ファイルの各行に、エクスポートポイント、ファイルシステムをマウントできるクライアントのリスト(オプション)、マウントパラメータのリスト(オプション)が記述されています。<br><br>**注記:** LifeKeeper が保護する NFS リソースを作成すると、ファイルシステムのエクスポート情報は *exports* ファイルから削除され、LifeKeeper によって管理されます。NFS リソースを削除すると、 *exports* ファイルにエクスポート情報がリストアされます。 |
| */var/lib/nfs* | 現在のエクスポート、クライアントマウント、ロックの状態およびその他の NFS 情報を含んだディレクトリーです。 |
| */var/lib/nfs/etab* | NFS にエクスポートしたファイルシステムの現在の表を含むファイルです。このファイルは **`exportfs`** コマンドによって管理され、ユーザーがこのファイルを直接編集することはありません。<br><br>**注記:** バックアップサーバーで NFS リソースのサービスを起動すると、その NFS ファイルシステムは、プライマリーサーバーの *etab* ファイルから削除され、バックアップサーバーの *etab* ファイルに挿入されます。 |
| */var/lib/nfs/rpc_pipefs* | NFS のユーザー空間の通信に対するカーネルに使用されます。このディレクトリーは、LifeKeeper のインストール中に */var/lib* に再配置されます。 |

| | |
|---|---|
| **exportfs(8)** *(/usr/sbin/exportfs)* | */var/lib/nfs/etab* のエクスポートされたファイルシステムの表を管理するコマンドです。 |
| **rpc.mountd(8)** *(/usr/sbin/rpc.mountd)* | マウント要求を認証し、クライアントにファイルシステムのマウントを許可する場合にはファイルハンドルを返すデーモンです。 |
| **rpc.nfsd(8)** *(/usr/sbin/rpc.nfsd)* | クライアントのファイルシステム要求を処理するデーモンです。 |
| **rpc.quotad(8)** *(/usr/sbin/rpc.rquotad)* | NFS 経由でリモートマウントされたローカルファイルシステムのユーザーのクォータを返す rpc サーバーです。 |
| rpc.lockd(8) *(/sbin/rpc.lockd)* | クライアントのファイルロック要求を処理するデーモンです。 |
| rpc.statd(8) *(/usr/sbin/rpc.statd)* | NFS クライアントとサーバーのステータスを監視し、ステータスを通知するデーモンです。このデーモンは、NFS ファイルロックを適切に機能させるために、実行しておく必要があります。 |
| rpcbind | RPC プログラム番号をポート番号に変えるデーモンプロセス。このデーモンは、NFS のために実行しておく必要があります。このプロセスが失敗するとスタンバイノードへのスイッチオーバーを強制します。<br>また、LifeKeeper も NFS 監視のために使っています。 |
| rpc.idmapd | NFS v4 で、ユーザー名 / グループ名とユーザー ID/ グループ ID を一致させるデーモンプロセスです。 |

## エクスポートの考慮事項

特定のエクスポートされたファイルシステムに対する LifeKeeper の保護は、exports(5) のマニュアルページで説明されている形式のエクスポートオプションによって決まります。特に、ホスト制限のフォーマットに注意してください。正当なホスト制限は、単独ホスト、ネットグループ、ワイルドカードで指定されたホスト名およびネットマスクの 4 つのみです。

特に、ワイルドカードで指定された IP アドレス (172.13.4.* など) は不正であり、スイッチオーバーまたはフェイルオーバーでファイルハンドルの陳腐化を招く恐れがあります。
`exportfs -v` を実行し、返されたエクスポートの記述とマニュアルページに説明されているフォーマットを手動で比較することで、慎重にチェックを行ってください。

（exportfs では完全にチェックは行えず、一部の不正なエクスポートフォーマットを受け入れてしまいます。）

## エクスポートオプション fsid=0 非サポート

エクスポートのオプションに “fsid=0” を指定すると、NFSv4 では擬似ファイルシステムのルートとして扱われます。しかし、LifeKeeper ではサポートされません。LifeKeeper で保護するエクスポートポイントのオプションには ”fsid=0” を指定しないでください。また、LifeKeeper は ”fsid=0” が指定されたエクスポートポイントのサブディレクトリーのエクスポートポイントも保護できません。

クライアントからの接続に影響がありますので、LifeKeeper で保護していないエクスポートについても “fsid=0” を指定してエクスポートしないでください。

## bind マウント 非サポート

bind マウントは LifeKeeper ではサポートされません。エクスポートポイントを bind マウントしないでください。

## RPC.MOUNTD 再起動

複数の NFS リソース階層がある特定の条件下では、rpc.mountd は、使用可能なエクスポートの一覧を適切に公表できません。そのため、リストア時の NFS Recovery Kit は停止し、rpc.mountd を再開して、エクスポートの適切な一覧をすべてのクライアントが利用できるようにします。rpc.mountd を停止し、再開するアクションは、*/etc/default/LifeKeeper* 内の RESTARTMOUNTD エントリーによって制御されます。デフォルトでは、このエントリーは次のように true に設定され、NFS リストア時に必ず rpc.mount の停止と再起動が行われます。

**RESTARTMOUNTD=true**

このアクションをオフにするには、次のように設定します。

**RESTARTMOUNTD=false**

## NFS リソース階層

NFS リソースを作成する前に、IP アドレスリソースを作成する必要があります。

LifeKeeper が保護する NFS リソースを作成すると、LifeKeeper によって次の階層が作成されます。

- NFS ファイルシステムリソース (親またはルート)
  - IP リソース
    - HA-NFS リソース
      - ファイルシステムリソース (基礎ファイルシステム)

NFS リソースを作成する前に、ファイルシステムリソースを作成することができます。この場合は、ファイルシステムリソースに割り当てる名前を選択できます。そうでない場合は、NFS リソース作成時に、

NFS Server Recovery Kit によって自動的にファイルシステムリソースが作成されます。

## NFS サブシステムの停止と起動による悪影響

NFS Server Recovery Kit がNFS エクスポートを保護しているときに NFS サブシステム (`nfs-server.service`) が停止された場合、NFS の停止動作ですべてのディレクトリーのエクスポート解除が実行されるため、保護しているすべてのエクスポート済みディレクトリーが影響を受けます。quickCheck が停止した NFS プロセスを検出し、ローカルリカバリーを実行してプロセスを再起動してから、ディレクトリーを再エクスポートします。ただし、すべてを復旧するためには、保護されたエクスポートのすべてについて quickCheck が実行される必要があります。例えば、5 個のエクスポートが保護されている場合、保護するすべてのエクスポート済みディレクトリーを復旧するために、quickCheck が 5 回実行されます。デフォルトの quickCheck の間隔は 2 分なので、すべてのエクスポート済みディレクトリーを復旧するためには、10 分かかります。

NFS Server Recovery Kit がシステム上のエクスポート済みディレクトリーをアクティブに保護しているときには、NFS サブシステムを停止しないでください。NFS サブシステムを停止する必要がある場合は、すべての NFS リソースをスタンバイノードに切り替えてから NFS サブシステムを停止してください。`exportfs` コマンドの使用も検討する必要があります。このコマンドラインユーティリティは 1 つのディレクトリーのエクスポート / エクスポート解除の機能を有しているので、NFS サブシステム全体を停止する必要がなくなります。

# 6.10.3.2. NFS 固有の構成に関する考慮事項

LifeKeeper NFS Server Recovery Kit を使用する前に、次の事項を考慮する必要があります。

- LifeKeeper で保護する NFS ファイルシステムはプライマリーサーバー (NFS リソースを作成するサーバー) によってエクスポートする必要があります。これは、NFS が実行中で、基礎ファイルシステムがマウントされていることを意味します。

> 注意：NFS 経由で */home* ディレクトリーを共有している場合は、 */home* が基礎ファイルシステムです。

- **読み書き可能なファイルシステムをエクスポートするときは、 *sync* オプションを使用してください。** このオプションを使用すると、ファイルシステムのすべての書き込みをディスクにコミットしてから書き込み要求が完了します。LifeKeeper の保護下にある NFS ファイルシステムで *sync* オプションを使用しない場合、フェイルオーバー時にデータが失われる可能性があります。

- 基礎ファイルシステムは共有デバイス上にある必要があり、書き込み権限でマウントする必要があります。

- 基礎ファイルシステムがすでに LifeKeeper によって保護されている場合、そのファイルシステムはプライマリーサーバー上で In Service で、最高の優先順位を持っている必要があります。基礎ファイルシステムが LifeKeeper の保護下にない場合、Recovery Kit によって保護下に置かれます。

- NFS Server Recovery Kit は、プライマリーサーバーで作成され In Service の IP リソースを必要とします。また、この IP リソースは、プライマリーサーバーで最高の優先順位を持っている必要があります。

- NFS リソースを作成する前に、クライアントは LifeKeeper が保護している IP アドレスを使用して NFS ファイルシステムをマウントできる必要があります。

- NFS ファイルシステムリソースを拡張するときには、各サーバーでファイルシステムを同じマウントポイントでマウントする必要があります。

- Red Hat Enterprise Linux 7以降、CentOS 7以降、Oracle Linux 7以降では NFSv2 はサポートされません。

- Red Hat Enterprise Linux 8以降では NFS over UDP はサポートされません。

# 6.10.3.3. NFS 構成例

本セクションの設定例は、共有(または複製) ディスク上での NFS インスタンスの設定方法を示しています。それぞれの図は、設定の種類と NFS パラメータとの関係を示しています。また、各設定は、このセクションで説明している設定の規則と要件に従って、NFS 設定と LifeKeeper ソフトウェアとの互換性を確保しています。

本セクションに挙げたものは可能な設定の例にすぎませんが、これらの設定例を理解し、設定規則に従うことは、ユーザのコンピューティング環境で実現可能なソリューションを明確にし、設定する上で役立ちます。

## 設定の要件

NFS タグ名は、LifeKeeper が保護するファイルシステムを示す任意の名前です。LifeKeeper によって示される標準タグ名は「 *nfs-<export point>* 」です。

設定例について理解するには、基礎のファイルシステムが必ず共有または複製ディスク上になければならないことを考慮に入れる必要があります。このファイルシステムはクラスター内のシステムのそれぞれからマウントできる必要があります。

## 設定例

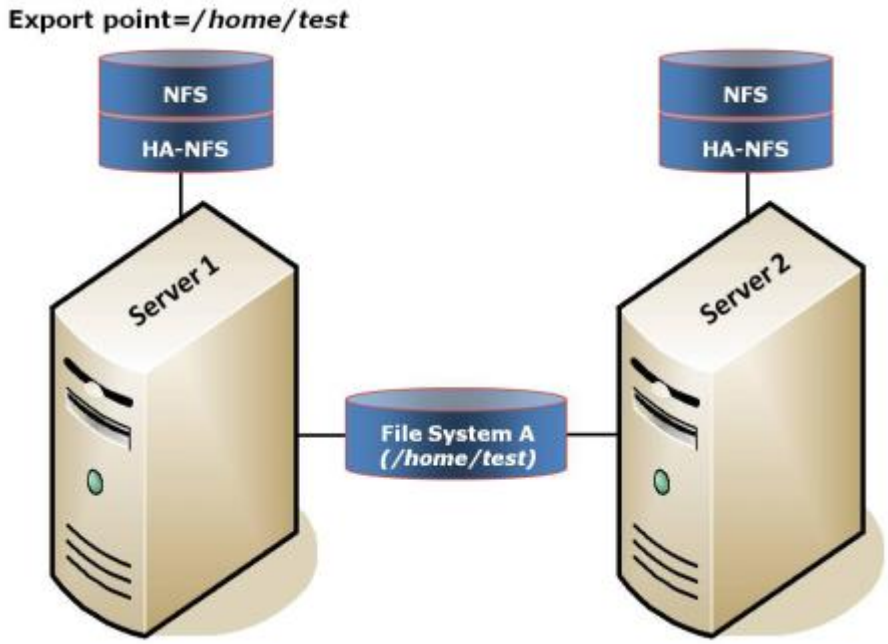
- アクティブ / スタンバイ設定
- アクティブ / アクティブ設定

# 6.10.3.3.1. アクティブ / スタンバイ設定

本セクションでは、アクティブ / スタンバイ設定の例を示します。この設定では、NFS を実行し、ファイルシステムをエクスポートしている *Server 1* がアクティブと見なされます。 *Server 1* に障害が発生した場合、 *Server 2* がファイルシステムをマウントしてエクスポートします。クライアントは切り替えを意識することなく使用することができます。

注意：アクティブ / スタンバイ設定では、 *Server 2* でも NFS が実行されていることがありますが、他の NFS リソースは LifeKeeper によって保護されていません。

## アクティブ / スタンバイ構成例




### 設定に関する注記:

- NFS ソフトウェアを両方のサーバーにインストールしておく必要があります。
- 基礎ファイルシステム (*File System A*) は共有 (または複製) ディスク上に置く必要があります。
- NFS エクスポートポイントは */home/test* です。
- エクスポートされたファイルシステムは、 *Server 1* と *Server 2* の両方で同じマウントポイントにする必要があります。
- *Server 2* は *Server 1* がアクティブである間は、共有ディスク上のファイルとディレクトリーにアクセスできません。

## Server 1 でのリソース階層の作成

| | |
|---|---|
| Server: | *Server1* |
| Export Point: | */home/test* |
| IP Tag: | ip-172.17.100.202 |
| NFS Tag: | *nfs-/home/test* |

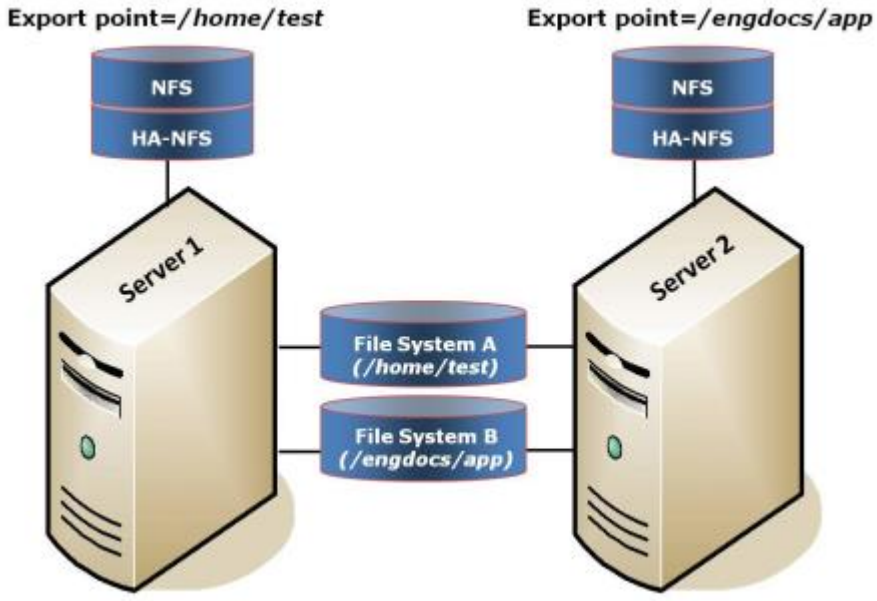
## Server 2 へのリソース階層の拡張

| | |
|---|---|
| Template Server: | *Server1* |
| Tag to Extend: | *nfs-/home/test* |
| Target Server: | *Server2* |
| Target Priority: | 10 |

# 6.10.3.3.2. アクティブ / アクティブ設定

アクティブ / アクティブの例では、複数のシステムが NFS を実行し、ファイルシステムをエクスポートします。

## アクティブ / アクティブ構成例




### 設定に関する注記

- NFS ソフトウェアを両方のサーバーにインストールしておく必要があります。
- 最初は、 *Server 1* が */home/test* をエクスポートし、 *Server 2* が */engdocs/app* をエクスポートしています。スイッチオーバー時には、1 つのシステムで両方のファイルシステムをエクスポートできます。
- ファイルシステム A は、エクスポートポイント、 */home/test* の基礎ファイルシステムです。ファイルシステム B は、エクスポートポイント、 */engdocs/app* の基礎ファイルシステムです。
- 基礎ファイルシステムは異なる共有ディスク上にあります。

### *Server 1* での 1 つ目のリソース階層の作成

| | |
|---|---|
| Server: | *Server1* |
| Export Point: | */home/test* |
| IP tag: | ip-172.17.100.202 |
| NFS Tag: | *nfs-/home/test* |

### 1 つ目のリソース階層の Server 2 への拡張

| | |
|---|---|
| Template Server: | *Server1* |
| Tag to Extend: | *nfs-/home/test* |
| Target Server: | *Server2* |
| Target Priority: | 10 |

### Server 2 での 2 つ目のリソース階層の作成

| | |
|---|---|
| Server: | *Server2* |
| Export Point: | */engdocs/app* |
| IP Tag: | ip-172.17.100.203 |
| NFS Tag: | *nfs-/engdocs/app* |

### 2 つ目のリソース階層の Server 1 への拡張

| | |
|---|---|
| Template Server: | *Server2* |
| Tag to Extend: | *nfs-/engdocs/app* |
| Target Server: | *Server1* |
| Target Priority: | 10 |

# 6.10.4. NFS 設定作業

以下の設定作業は LifeKeeper GUI から実行できます。次の 4 つの作業は、NFS サーバーリソースインスタンスに特有で、Recovery Kit ごとに異なるため、本セクションで説明しています。

> 注意：本セクションでは、LifeKeeper GUI の **[Edit]** ニューから作業を選択することによって、Recovery Kit を設定する方法を説明します。設定作業はツールバーから選択することもできます。状況表示ウィンドウの **[Resource Hierarchy Tree]** (左側のペイン)のグローバルリソースを右クリックして、**[Edit]** メニューと同じドロップダウンメニューの選択項目を表示することもできます。言うまでもなく、これは階層がすでに存在している場合にだけ可能な方法です。
> 状況表示ウィンドウの **[Resource Hierarchy Table]** (右側のペイン)のリソースインスタンスを右クリックして、サーバーおよび特定リソースの状況に応じて、リソース階層の作成を除くすべての設定作業を実行することもできます。

- NFS リソース階層の作成　アプリケーションリソース階層を LifeKeeper クラスターに作成します。
- NFS リソース階層の削除　リソース階層を LifeKeeper クラスター内のすべてのサーバーから削除します。
- NFS リソース階層の拡張　リソース階層をプライマリーサーバーからバックアップサーバーへ拡張します。
- NFS リソース階層の拡張解除　リソース階層を LifeKeeper クラスター内の 1 つのサーバーから拡張解除(削除)します。
- NFS リソース階層のテスト　手動でスイッチオーバーを行って、NFS リソース階層をテストします。

以下の作業については、すべての Recovery Kit で同じ手順を使用する共通の作業であるため、 SPS for Linux テクニカルドキュメンテーション の 管理 セクションを参照してください。

- リソース依存関係の作成　既存のリソース階層と別のリソースインスタンスとの間に親子の依存関係を作成し、クラスター内のすべての対象サーバーに依存関係の変更を反映します。
- リソース依存関係の削除　リソースの依存関係を削除して、クラスター内のすべての対象サーバーに依存関係の変更を反映します。
- In Service　リソース階層を特定のサーバーで In Service の状態にします。
- Out of Service　リソース階層を特定のサーバーで Out of Service の状態にします。
- プロパティの表示 / 編集　特定のサーバーでリソース階層のプロパティを表示または編集します。

# 6.10.4.1. NFS リソース階層の作成

プライマリーサーバーからリソースインスタンスを作成するには、以下の手順を実行します。

1. LifeKeeper GUI メニューから **[Edit]** を選択し、次に **[Server]** を選択します。ドロップダウンメニューから、 **[Create Resource Hierarchy]** を選択します。

   すでに行った選択を変更する場合、または NFS リソース階層の作成中にエラーメッセージが表示された場合、 **[Back]** ボタンを使用して選択を変更するか、修正を行ってください ( **[Back]** ボタンが有効な場合)。

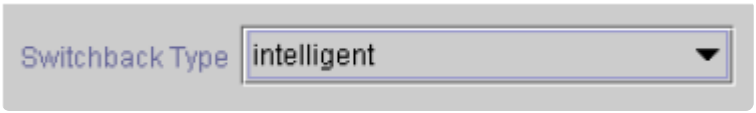
   ダイアログボックスが表示され、ドロップダウンメニューに、クラスターにインストールされ認識されているすべての Recovery Kit が一覧で示されます。ドロップダウンメニューから **[NFS]** を選択します。

   **[Next]** をクリックして次のダイアログボックスに進んでください。

> 注意：階層作成の途中で **Cancel** ボタンをクリックすると、作成処理全体が取り消されます。

2. **[Switchback Type]** を選択します。これによって、バックアップサーバーへのフェイルオーバー後、NFS インスタンスがサービスに戻ったときにどのようにプライマリーサーバーにスイッチバックされるかが決まります。 **[intelligent]** または **[automatic]** を選択します。 **[Intelligent switchback]** の場合、インスタンスをプライマリー / オリジナルサーバーにスイッチバックするときに管理者の介入が必要になります。 **[Automatic switchback]** の場合、プライマリーサーバーがオンラインに戻り、LifeKeeper の保護下に入るとすぐに自動的にスイッチバックが行われます。

   スイッチバックタイプは、必要な場合 **[Resource Properties]** ダイアログボックスの **[General]** タブで後から変更できます。

   **[Next]** をクリックして次のダイアログボックスに進んでください。

3. **[Server]** で NFS リソースを作成するサーバー (通常、これをプライマリーサーバーまたはテンプレートサーバーと呼ぶ) を選択します。ドロップダウンメニューには、クラスター内の全サーバーが表示されます。

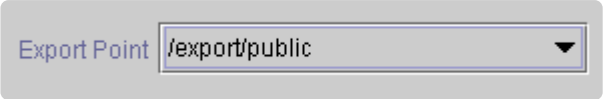
   

**[Next]** をクリックして次のダイアログボックスに進んでください。

4. **[Export Point]** ダイアログボックスのドロップダウンリストに、次の条件に合致する NFS ファイルシステムのエクスポートポイントが表示されます。

   - エクスポートポイントが NFS によってエクスポートされている。
   - エクスポートポイントが共有ドライブ上にある。
   - 基礎ファイルシステムが LifeKeeper に保護されている場合、基礎ファイルシステムは In Service で、かつ、 **Server** ダイアログボックスで選択したサーバー上で最高の優先順位を持っている必要がある。
   - fsid=0 でエクスポートされたエクスポートポイントまたはそのサブディレクトリーのエクスポートポイントは表示されません。

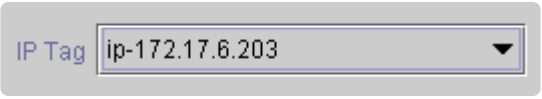
   ドロップダウンリストから保護する NFS エクスポートポイントを選択します。

   **[Next]** をクリックして次のダイアログボックスに進んでください。

5. **[IP Tag]** ダイアログボックスのドロップダウンリストに、現在 LifeKeeper が保護していて、NFS リソースを作成するサーバーで in service の仮想 IP アドレスに対応するタグが表示されます。

   保護されている NFS ファイルシステムにアクセスするためにクライアントが使用する仮想 IP アドレスの **タグ** を選択します。

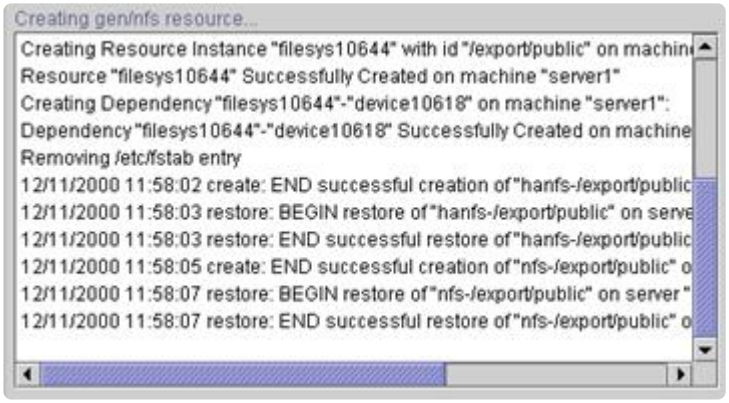
   **[Next]** をクリックして次のダイアログボックスに進んでください。

> 注意：この時点で、保護される IP リソースが利用可能かどうかが LifeKeeper によってチェックされます。

> 注意：仮想 IP アドレス依存関係を持つ別の LifeKeeper Recovery Kit を使用している場合は、NFS リソース用に別の仮想 IP アドレスを作成することもできます。そうしない場合、仮想 IP リソースがバックアップサーバーにフェイルオーバーされると、その IP リソースに依存するすべてのリソースが同時にフェイルオーバーされます。

6. **[NFS Tag]** を選択または入力します。これは NFS 階層に付けるタグ名です。デフォルトを選択するか、独自のタグ名を入力することができます。

**[Create]** ボタンをクリックすると、***[Create Resource Wizard]*** によって NFS リソースが作成されます。




**[Next]** をクリックすると階層の拡張 で説明する ***[Pre-Extend Wizard]*** が起動します。

> **注意：**この時点で、NFS リソース階層が正常に作成されます。ただし、新しい NFS インスタンスの起動に失敗したというエラーメッセージが表示されることがあります。新しい NFS 階層を別のシステムに拡張するには、起動する(in service にする) 必要があることに注意してください。起動に失敗した場合、この時点で表示されるエラーに基づいて問題を修正することも可能です。エラーの修正が不可能であれば、リソース作成のキャンセルのみを選択することになります。

階層の拡張 に進む前に、新しい階層を In Service にしてください。

# 6.10.4.2. NFS リソース階層の削除

LifeKeeper 環境内のすべてのサーバーからリソース階層を削除するには、以下の手順を実行します。

1. LifeKeeper GUI メニューから **[Edit]** を選択し、次に **[Resouce]** を選択します。ドロップダウンメニューから **[Delete Resource Hierarchy]** を選択します。

2. **[Target Server]** で、NFS リソース階層を削除するサーバーの名前を選択します。

**[Next]** をクリックして次のダイアログボックスに進んでください。

> 注意：次のどちらかの場所で右クリックして **[Delete Resource]** を選択した場合、このダイアログボックスは表示されません。
> ・個々のリソースインスタンスの右ペイン
> ・リソースが 1 台のサーバーだけにあるとき、グローバルリソースの左ペイン

3. **[Hierarchy to Delete]** を選択します。削除するリソース階層を特定して、強調表示にします

**[Next]** をクリックして次のダイアログボックスに進んでください。

> 注意：左側ペインのグローバルリソースまたは右側ペインの個々のリソースインスタンスを右クリックしてリソースの削除作業を選択した場合、このダイアログボックスは表示されません。

4. 選択したターゲットサーバーと、削除の対象として選択した階層を確認する情報ボックスが表示されます。

**[Delete]** をクリックして、次のダイアログボックスに進みます。

5. NFS リソースの削除に成功したことを示す別の情報ボックスが表示されます。

6. **[Done]** をクリックして、 **[Delete Resource Hierarchy]** メニューの選択を終了します。

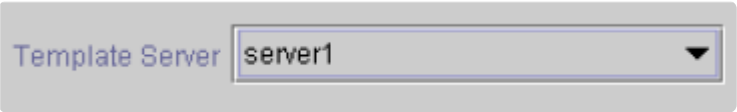
# 6.10.4.3. NFS リソース階層の拡張

階層の作成後、クラスター内の別のサーバーに拡張する必要があります。リソースインスタンスをテンプレートサーバーからターゲットサーバーに拡張するには方法が 2 つあります。最初の方法は、リソース作成後、[Continue] をクリックして、別のサーバーにリソースを拡張する方法です。もう 1 つの方法は、次に説明するように [Edit] メニューから [Extend Resource Hierarchy] を選択する方法です。どちらの方法でも、同じダイアログボックスが表示されます (一部例外があり、これについては次で説明)。

注意：階層拡張の途中で [Cancel] ボタンをクリックすると、その特定のサーバーへの拡張処理が取り消されます。また、取り消しポイントを超えると、現在拡張されている階層 (IP アドレスまたはファイルシステム) の依存リソースもすべて拡張解除されます。ただし、すでに NFS リソース階層を別のサーバーに拡張している場合は、明示的に拡張解除するまで、そのインスタンスの拡張は有効です。
例えば、Server 1 にリソースを作成し、そのリソースを Server 2 へ拡張したとします。同じリソースを Server 3 に拡張している最中に、拡張を取り消すためいずれかのダイアログボックス内の [Cancel] ボタンをクリックしたとします。この場合は、Server 3 へのリソース拡張だけがキャンセルされ、Server 2 への拡張はキャンセルされません。この階層から Server 2 を削除したい場合は、Server 2 からリソースを拡張解除する必要があります。

1. LifeKeeper GUI メニューから **[Extend Wizard]** に入っている場合は、 **[Edit]** を選択し、次に **[Resource]** を選択します。ドロップダウンメニューから **[Extend Resource Hierarchy]** を選択します。これで **[Extend Resource Hierarchy]** ウィザードが起動されます。

2. 最初に表示されるダイアログボックスで、NFS リソース階層が現在 In Service の **[Template Server]** を選択します。ここで選択するするテンプレートサーバーと次のダイアログボックスで選択する拡張するタグによって、 In Service のリソース階層が表されることを認識しておくことが重要です。選択したテンプレートサーバーで In Service でないリソースタグを選択すると、エラーメッセージが表示されます。このダイアログのドロップダウンボックスには、クラスター内の全サーバーの名前が示されています。

   **[Next]** をクリックして次のダイアログボックスに進んでください。

注意：NFS リソース階層作成後、直ちに **[Extend Resource Hierarchy]** 作業に入った場合は、作成段階でテンプレートサーバーが特定されているため、このダイアログボックスは表示されません。これは、GUI ウィンドウの左ペインの NFS リソースアイコンまたは右ペインの NFS リソースボックスを右クリックして **[Extend Resource Hierarchy]** を選択した場合も同様です。

3. **[Tag to Extend]** を選択します。これは、テンプレートサーバーからターゲットサーバーに拡張する NFS インスタンスの名前です。ウィザードのドロップダウンメニューには、前述のダイアログボックスで選択したテンプレートサーバー上に作成されたすべてのリソースが表示されます。

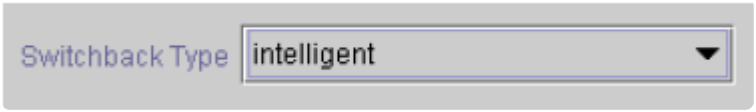
**[Next]** をクリックして次のダイアログボックスに進んでください。

> 注意：ここでも、NFS リソース階層作成後、ただちに **[Extend Resource Hierarchy]** 作業に入った場合は、作成段階で NFS リソースのタグ名が特定されているため、このダイアログボックスは表示されません。これは、GUI ウィンドウの左ペインの NFS リソースアイコンまたは右ペインの NFS リソースボックスを右クリックして、**[Extend Resource Hierarchy]** を選択した場合も同様です。

4. NFS リソース階層の拡張先の **[Target Server]** を選択します。ドロップダウンボックスに、選択された階層にまだ入っていないクラスター内のサーバー名の一覧が表示されます。

**[Next]** をクリックして、次のダイアログボックスに進んでください。

5. **[Switchback Type]** を選択します。これによって、バックアップサーバーへのフェイルオーバー後、NFS インスタンスがサービスに戻ったときに、どのようにプライマリーサーバーにスイッチバックされるかが決まります。[intelligent] または [automatic] を選択できます。[intelligent] の場合、インスタンスをプライマリー / オリジナルサーバーにスイッチバックするときに管理者の介入が必要になります。[automatic] の場合、プライマリーサーバーがオンラインに戻り、LifeKeeper の保護下に入るとすぐに自動的にスイッチバックが行われます。




スイッチバックタイプは、必要な場合 **[Resource Properties]** ダイアログボックスの **[General]** タブで後から変更できます。

**[Next]** をクリックして次のダイアログボックスに進んでください。

6. **[Template Priority]** を選択または入力します。これはサーバーで現在 In Service の NFS 階層の優先順位です。優先順位は、1 ～ 999 の範囲で未使用の値が有効で、小さい数字ほど優先順位が高くなります (数字 1 が最高の優先順位に相当)。拡張処理時に、別のシステムですでに使用中の優先順位をこの階層に対して指定することはできません。デフォルト値を推奨します。

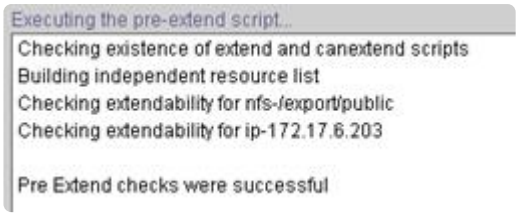
**[Next]** をクリックします。

> 注意：このフィールドは階層を最初に拡張するときだけ表示されます。

7. **[Target Priority]** を選択または入力します。これは、別サーバーにある同等の階層に対する新しく拡張する NFS 階層の優先順位です。1 ～ 999 の範囲で、まだ優先順位として使用されていない値が有効で、リソースのカスケーディングフェイルオーバーシーケンスにおけるサーバーの優先順位を示します。数値が小さいほど優先順位は高くなります (1 は最高の優先順位を表す)。LifeKeeper のデフォルトでは、階層が作成されたサーバーに「1」が割り当てられることに注意してください。優先順位は連続している必要はありませんが、特定のリソースについて 2 つのサーバーに同じ優先順位を割り当てることはできません。

**[Next]** をクリックします。

8. 環境がチェックされ、この NFS リソースを拡張するための要件がすべて満たされていることを示す情報ボックスが表示されます。満たされていない要件がある場合、**[Next]** は選択できず、**[Back]** ボタンが有効になります。




**[Back]** をクリックした場合、情報ボックスに表示されるエラーメッセージの内容に従って、リソースの拡張を変更できます。

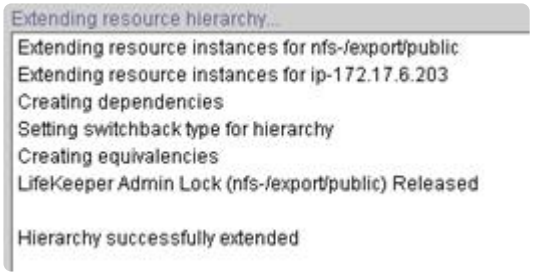
ここで **[Cancel]** をクリックした場合、NFS リソース階層を LifeKeeper で保護するには、いずれかの時点でここに戻り、別のサーバーに NFS リソース階層を拡張する必要があります。

**[Next]** をクリックすると、**[Extend Resource Hierarchy]** 設定作業に入り **[NFS Tag]** ダイアログボックスが表示されます。

9. この画面にはテンプレートサーバー、拡張するタグ、ターゲットサーバー、標準の NFS タグに関する情報が表示されます。NFS タグは、NFS 階層の拡張に付けるタグ名です。デフォルトを選択するか、独自のタグ名を入力することができます。

**[Next]** をクリックして次のダイアログボックスに進んでください。

10. 拡張を実行中であることを確認する情報ボックスが表示されます。

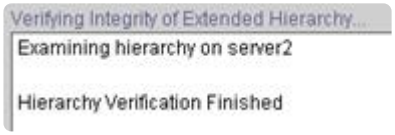



同じ NFS リソースインスタンスをクラスター内の別のサーバーに拡張する場合は、 **[Next Server]** をクリックします。その場合は、リソース階層の拡張の操作が繰り返されます。

**[Finish]** をクリックした場合、NFS リソースの拡張処理が成功したか検証が行われます。

注意：IP リソースをターゲットサーバーにまだ拡張していない場合、NFS リソースの拡張処理中に NFS Server Recovery Kit によって拡張されます。この場合、拡張の実行に関する情報ボックスが表示される前に、IP リソースの拡張に関連したダイアログボックスがいくつか表示されます。

11. **[Finish]** をクリックした場合、次の画面が表示されます。




12. **[Done]** をクリックして終了します。

注意：必ず両方のサーバーで新しいインスタンスの機能をテストしてください。

# 6.10.4.4. NFS リソース階層の拡張解除

リソース階層の拡張解除を行うには、次の手順を実行します。

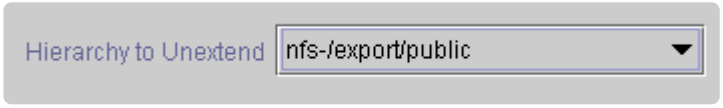
1. LifeKeeper GUI メニューから **[Edit]** を選択し、次に **[Resouce]** を選択します。ドロップダウンメニューから **[Unextend Resource Hierarchy]** を選択します。

2. **[Target Server]** で、NFS リソースを拡張解除するサーバーを選択します。NFS リソースが現在 In Service のサーバーは選択できません。

**[Next]** をクリックして次のダイアログボックスに進んでください。

> 注意：右側のペインから個々のリソースインスタンスを右クリックして **[Unextend]** 作業を選択した場合、このダイアログボックスは表示されません。

3. **[Hierarchy to Unextend]** で、拡張解除する NFS 階層を選択します。



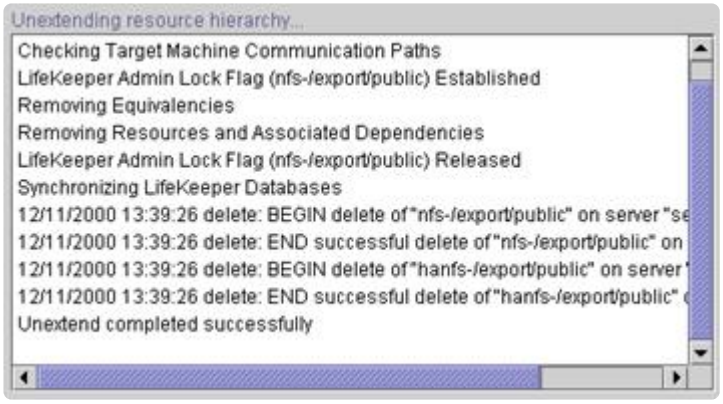

**[Next]** をクリックして次のダイアログボックスに進んでください。

> 注意：左側のペインからグローバルリソースを右クリックするか、右側のペインから個々のリソースインスタンスを右クリックして **[Unextend]** 作業を選択した場合、このダイアログボックスは表示されません。

4. 拡張解除するために選択したターゲットサーバーと NFS リソース階層を確認する情報ボックスが表示されます。

**[Unextend]** をクリックします。

5. NFS リソースの拡張解除に成功したことを示す別の情報ボックスが表示されます。




6. **[Done]** をクリックして、 **[Unextend Resource Hierarchy]** メニューを終了し、LifeKeeper GUI に戻ります。

   階層が拡張解除されて 1 つのサーバーだけになった場合、「**One or More Resources Unprotected**」という警告メッセージが表示されます。

# 6.10.4.5. NFS リソース階層のテスト

NFS リソース階層をテストする前に、次に説明しているようにクライアントの設定を検証する必要があります。その後、手動でスイッチオーバーを行って、NFS リソース階層のテストを行うことができます。これにより、プライマリーサーバーからバックアップサーバーへのリソースインスタンスのフェイルオーバーをシミュレートします。

## クライアントの設定の検証

一般に、クライアントは、NFS リソース階層の作成作業中に選択した LifeKeeper が保護する IP アドレスを使用してファイルシステムをマウントする必要があります。正しい IP アドレスを選択しているかクライアント側で確認することはできません。したがって、クライアントがファイルシステム用の正しい IP アドレスを確実に使用するように、慎重に次の検証手順を実行する必要があります。

クライアントの設定を検証するには、次の手順に従います。

1. バックアップサーバーで In Service の NFS インスタンスがないことを確認します。
2. LifeKeeper が保護する正しい IP アドレスを使用して、クライアントにファイルシステムをマウントします。
3. 手動でバックアップサーバーへのスイッチオーバーを実行して、切り替えた NFS インスタンスをバックアップサーバーで現在 In Service の唯一の NFS インスタンスにします。
4. スイッチオーバーが完了したら、クライアントがそのファイルシステムにアクセスできることを確認します。

## GUI による手動スイッチオーバーの実行

依存関係を定義すると、自動的に LifeKeeper によってアプリケーションの起動と終了が制御され、障害が検出された場合にはフェイルオーバーが開始されます。保守作業など、管理上の理由からスイッチオーバーを手動で実行することもできます。

LifeKeeper GUI で、ドロップダウンメニューから [Edit] > [Resource] > [In Service] を選択すると、手動スイッチオーバーを開始できます。例えば、バックアップサーバーで In Service 要求を実行すると、アプリケーション階層はバックアップサーバーで In Service になり、プライマリーサーバーでは Out of Service になります。

[Out of Service] 要求を実行すると、アプリケーションは他のサーバーで In Service にならずに、Out of Service になります。

> 注意：NFS リソースを Out of Service にするには、NFS リソースと関連する HA-NFS リソースの両方を Out of Service にする必要があります。

アプリケーション側の動作については、すべての動作はアプリケーションのドキュメントで規定されてい

ます。LifeKeeper 側では、ロールバックやバックアップアーカイブのような内部処理の調整や制御は行いません。テープアーカイブやリストアの作業は、アプリケーション管理者の責任です。

## ローカルリカバリー動作

プライマリーサーバーに障害が発生すると、NFS Server Recovery Kit は次の作業を実行します。

1. NFS デーモンが実行されていない場合、それらを開始します。
2. NFS ファイルシステムをエクスポートします。

# 6.10.5. NFS トラブルシューティング

本セクションでは、LifeKeeper NFS リソース階層の作成時や拡張時、またはリソースの削除時やリストア時に表示される可能性のあるメッセージの一覧を示します。必要に応じて、エラーの原因およびエラー状態を解消するために必要な処置についても説明しています。

他の LifeKeeper コンポーネントからメッセージが出されることもあります。そのような場合は、メッセージカタログ を参照してください。メッセージカタログには、操作、管理、GUI など、LifeKeeper for Linux を使用しているときに遭遇する可能性がある、すべてのエラーコードが列挙されています。また、エラーコードの原因に関する追加の説明や、問題解決のために必要な処置についても、必要に応じて記載されています。この完全なリストを検索すると、受信したエラーコードを見つけることができます。また、関連する LifeKeeper コンポーネントの個別のメッセージカタログに直接アクセスすることもできます。

下記トラブルシューティングのヘルプをクリックしてください。

- HA nfs-utils のインストールと設定
- NFS 階層作成時のエラー
- NFS 階層拡張時のエラー
- NFS 階層の起動、停止、ローカルリカバリー時のメッセージとエラー
- NFS 階層削除のメッセージとエラー

# 6.10.5.1. HA nfs-utils のインストールと設定

LifeKeeper NFS Server Recovery Kit は、高可用性の有効な nfs-utils パッケージがインストールおよび設定されている必要があります。(いくつかの OS のディストリビューションバージョンでは nfs-utils が別のパッケージで提供されます。)LifeKeeper NFS Server Recovery Kit のインストールでは、HA が有効な nfs-utils パッケージの存在が検査されます。正しく設定された nfs-utils パッケージを検出できなかった場合は、LifeKeeper インストールサポートのセットアップスクリプトを再実行するか、サーバーをリブートする必要があります。

LifeKeeper インストールサポートのセットアップスクリプトによって、NFS を HA 環境で使用できるように設定を行います。この設定変更を反映させるために NFS サービスの再起動が必要なので、セットアップスクリプトで “Restart NFS Service” を選択するか、インストール作業完了後に手動で NFS サービスを再起動してください。

**NFS v4** の設定において、 *rpc_pipefs* を */var/lib/nfs* から */var/lib* へ移動させる必要あります。この設定を行うためには、カーネルモジュールのアンロードおよび設定とブート時のスクリプトの追加もしくは修正が必要とされます。変更後、インストールサポートがカーネルモジュールのアンロードおよびリロードを行うことができない場合はシステムのリブートが必要になる可能性があります。この事象が発生した場合、システムのリブートが必要であることがユーザに通知されます。LifeKeeper の操作を正常に行うためにはシステムのリブートを含めた `rpc_pipefs` のセットアップを完了させることが必要です。

# 6.10.5.2. NFS リソース階層作成時のエラー

NFS リソース階層の作成時に表示されるエラーメッセージを、説明とともに挙げます。リストするメッセージは、nfs リソースと hanfs リソースの作成時に表示されるメッセージです。LifeKeeper Core および他の Application Recovery Kit が表示するエラーメッセージはこのガイドには記載していません。ここに記載されている問題を解決するために、リソース階層作成の手順を中断し、問題解決後に中断したところから再開することができます（NFS 構成に必要な LifeKeeper リソースの作成も含みます）。

注意：次のエラーメッセージの説明で、コマンドラインのみというのは、コマンドラインでコマンドを入力した場合にだけそのメッセージが表示されることを示しています。LifeKeeper GUI を使用している場合、そのメッセージは表示されません。さらに、リソース階層作成の終わりに、リソースの起動が開始されます。そのプロセス中に発生する可能性のあるメッセージとエラーについては、 NFS リソース階層の起動、停止、quickCheck、ローカルリカバリー時のメッセージとエラー を参照してください。

| エラー番号 | エラーメッセージ | 説明 |
|---|---|---|
| 106000 | create: Export point not specified | リソース階層を作成するときには、NFS ファイルシステムのエクスポートポイントを指定する必要があります。 *コマンドラインのみ* |
| 106001 | create: The path "EXPORT POINT" is not exported by NFS | 指定したエクスポートポイントが現在 NFS によってエクスポートされていません。 **exportfs(8)** を使用してパスをエクスポートし、パスが */var/lib/nfs/etab* ファイルに含まれていることを確認します。 *コマンドラインのみ* |
| 106002 | create: The export point “EXPORT POINT” on server “SERVER” for client “CLIENT” either does not contain an FSID export option or the value is not unique. A unique FSID will be generated and "EXPORT POINT" will be re-exported using the new FSID value. | LifeKeeper 保護下のすべてのエクスポートポイントでは、高可用性の NFS に固有の fsid=エクスポートオプションを使用する必要があります。選択したエクスポートは、この要件を満たしていなかったため、固有の値が生成されてから、選択したエクスポートポイントの再エクスポートが行われました。 **注記:**「*」または「world」のクライアントは、エクスポートポイントをすべてのクライアントが使用できることを示します。 |
| 106003 | create: Unable to create the export entry file for "EXPORT POINT" in LifeKeeper | 考えられる原因:<br>• ターゲットサーバーのファイルシステムに空き容量がありません。<br>• ファイルシステムの問題。 |
| 106004 | The export point "EXPORT POINT" is not on a shared file system on server "SERVER" | エクスポートポイントが共有ファイルシステムのものか確認します。 *コマンドラインのみ* |
| 106005 | Unable to create the HA-NFS hierarchy | 他のエラーメッセージを参考にして、実行するべき |

| | | |
|---|---|---|
| | "TAG" with child resource ID "ID" on server "SERVER" | 処置を決定します。 |
| 106006 | Unable to remove entry for export point "EXPORT POINT" from /etc/exports on server "SERVER" | */etc/exports* ファイルが存在しているか、読み込み可能かを確認します。 |
| 106014 | Usage: USAGE STRING | 誤った引数で実行されたコマンドの使用。 コマンドラインのみ |
| 106016 | "REQUIRED SOFTWARE" cannot be found or does not have the expected permissions on server "SERVER" | NFS は、プライマリーサーバーとすべてのバックアップサーバーにインストールする必要があります。 **nfs-utils** がインストールされていることを確認してください。 |
| 106019 | Executing command: "COMMAND" | このメッセージは、LifeKeeper が NFS デーモンを再起動するとき、またはエクスポートポイントをエクスポート、アンエクスポートするときに表示されます。問題が発生した場合、この情報が役に立ちます。 |
| 106020 | The generated id "ID" conflicts with an existing resource id | nfs または hanfs リソース用に内部で生成されたリソース ID は、重複しています。 |
| 106021 | An entry for export point “EXPORT POINT” already exists in /etc/exports. The entry that was being used by the NFS Server Recovery Kit has been placed in the file “FILENAME” | /etc/exports のエクスポートエントリーが正しいか確認します。 |
| 106025 | An unknown error has occurred while running "NEW TAG" on server "SERVER" | hanfs リソースの nfs のタグを生成するためにコマンド newtag を実行しているときに、予想外のエラーが発生しました。 |
| 106026 | Adding dependency between HA-NFS resource "HANFS TAG" and filesys resource "FILESYS TAG" on server "SERVER" failed. | 選択した hanfs リソースと filesys リソース間の依存関係の作成が、不明な理由で失敗しました。詳細については、出力を参照してください。 |
| 106027 | Open of "FILE NAME" on server "SERVER" failed: "ERROR MSG" | 記述された理由により、ファイルを開けませんでした。 |
| 106031 | Re-export of "EXPORT POINT" to add FSID option failed on server "SERVER" | エクスポートポイントには fsid 引数が含まれておらず、引数を生成した後に再エクスポートしようとしましたが失敗しました。fsid 引数を手動で /etc/exports エントリーに追加し、再エクスポートして、失敗内容を判別してください。 |
| 106032 | create: Dependent IP resource tag name not specified | NFS リソース階層を作成するときには、保護する IP アドレスのリソースタグを指定する必要があります。 *コマンドラインのみ* |
| 106033 | Selected IP resource "TAG" does not exist | NFS リソースを作成する前に、指定のサーバーに |

| | | |
|---|---|---|
| | on server “SERVER" | IP リソースを作成する必要があります。また、コマンドを入力するときに、IP リソースを正しく入力したか確認します。 コマンドラインのみ |
| 106034 | Adding dependency between NFS resource "TAG" and IP resource "TAG" on server "SERVER" failed | NFS リソースを作成しているサーバーで IP リソースが In Service か確認します。 コマンドラインのみ |
| 106035 | Creation of HA-NFS resource "TAG" on server "SERVER" failed | 他のエラーメッセージを参考にして、実行するべき処置を決定します。 |
| 106036 | Adding dependency between IP resource "TAG" and HA-NFS resource "TAG" on server "SERVER" failed | HA-NFS リソースを作成するサーバーで IP リソースが In Service か確認します。 コマンドラインのみ |
| 106037 | Attempts to get exclusive lock on "FILE NAME" on server "SERVER" failed: "ERROR MSG" | 更新のための排他ロックを取得できません。原因については、エラーメッセージを参照してください。 |
| 106038 | Unable to create directory "DIR NAME" on server "SERVER": "ERROR MSG" | エクスポートされたファイルシステムでディレクトリーを作成しようとしましたが、失敗しました。原因については、エラーメッセージを参照してください。 |
| 106039 | Open of "FILE" on server "SERVER" failed: “ERROR MSG”<br>or<br>Attempt to get exclusive lock on "FILE" on server "SERVER" failed: "ERROR MSG" | ファイルを開くかファイルの排他ロックを取得しようとしましたが、失敗しました。原因については、エラーメッセージを参照してください。 |
| 106041 | The selected IP resource "IP TAG" is not ISP on server "SERVER" | 選択した IP リソースはサーバーに存在していますが、現在 In Service ではありません。サーバーで IP リソースを In Service の状態にして、作成を再試行してください。 コマンドラインのみ |
| 106042 | Unable to find HA-NFS resource instance “HA-NFS TAG” on server “SERVER” | 何らかの理由で HA-NFS リソースが作成されませんでした。他のエラーメッセージを参考にして、問題を解決してください。 |
| 106048 | Multiple NFS v4 root exports found on "SERVER". | 1 つの NFS v4 擬似ファイルシステムのみサポートされている SERVER で、複数の NFS v4 擬似ファイルシステムが見つかりました。 |
| 106057 | Failed to open new exportInfo file “TMP FILENAME”. Reason: “ERROR MSG”. | 記述された理由により、一時ファイルのオープンに失敗しました。 |
| 106058 | Failed to add the entry “ENTRY” to “TMP FILENAME”. Reason: “ERROR MSG”. | 記述された理由により、一時ファイルへの書き込みに失敗しました。 |
| 106059 | Failed to close new exportInfo file “TMP FILENAME”. Reason: “ERROR MSG”. | 記述された理由により、一時ファイルのクローズに失敗しました。 |
| 106060 | Failed to create the new exportInfo file | 新しく作成した exportinfo ファイルが予期したサイ |

| | | |
|---|---|---|
| | “TMP FILENAME” as it’s size of “NEW SIZE” is less than the size of the current exportInfo with a size of “OLD SIZE”. The size should be at least the same as the old file size. | ズではありません。 |
| 106061 | Failed to rename temp exportInfo file from “TMP FILENAME” to “FILENAME”. Reason: “ERROR MSG”. | 記述された理由により、ファイル名の変更に失敗しました。 |
| 106062 | fsid(FSID) does not match between resource info and etab on “SERVER”. | 期待した fsid が etab に含まれていません。 |

# 6.10.5.3. NFS リソース階層拡張時のエラー

NFS リソース階層の拡張時に表示されるエラーメッセージを、説明とともに挙げます。これらのエラーメッセージは、GUI が「Executing the pre-extend script....」というメッセージを表示して、新しいシステムにリソース階層を拡張する前にリソース階層の検証を行っているときに表示されることに注意してください。

ターゲットサーバーへの **NFS** リソース階層の作成時

| エラー番号 | エラーメッセージ | 説明 |
|---|---|---|
| 106016 | "REQUIRED SOFTWARE" cannot be found or does not have the expected permissions on server "SERVER" | NFS は、プライマリーサーバーとすべてのバックアップサーバーにインストールする必要があります。 **nfs-utils** がインストールされていることを確認してください。 |
| 106017 | The file system "FILE SYSTEM" on template server "SERVER" has a different mount point "MOUNT POINT" on server "SERVER" | リソースは各サーバーに同じマウントポイントで作成する必要があります。ターゲットサーバーからファイルシステム階層を拡張解除するか、同じマウントポイントでテンプレートサーバーとターゲットサーバーに再作成します。 |
| 106018 | Unable to copy the file "FILENAME" from server "SERVER" to server "SERVER" | 考えられる原因:<br>• サーバー間のコミュニケーションパスがダウンしています。<br>• ターゲットサーバーのファイルシステムに空き容量がありません。<br>• ファイルシステムの問題。 |
| 106020 | The generated id "ID" conflicts with an existing resource id | nfs または hanfs リソース用に内部で生成されたリソース ID は、重複しています。 |
| 106022 | The export point “EXPORT POINT" is in /etc/exports on the target server "SERVER" | ターゲットサーバーの */etc/exports* ファイルからエクスポートポイントを削除してからリソースを拡張してください。 |
| 106023 | The export point "EXPORT POINT" is exported on the target server "SERVER" | リソース拡張前にターゲットサーバーのエクスポートポイントをエクスポート解除してください。 |
| 106051 | Unable to create active/active configurations with NFS v4 exports. Either "TEMPLATE SERVER" or "TARGET SERVER" currently protects a NFS v4 root export. | すでに片方もしくは両方のサーバーが NFS v4 エクスポートを保護しており、NFS v4 エクスポートでのアクティブ / アクティブをサポートしていないため TEMPLATE SERVER から TARGET SERVER へ NFS リソースを拡張することができません。 |

# 6.10.5.4. NFS リソース階層の起動、停止、quickCheck、ローカルリカバリー時のメッセージとエラー

## NFS リソースを In Service にする (起動)

| エラー番号 | エラーメッセージ | 説明 |
|---|---|---|
| 106003 | restore: Unable to create the export entry file for “EXPORT POINT” in LifeKeeper | 考えられる原因:<br>• ターゲットサーバーのファイルシステムに空き容量がありません。<br>• ファイルシステムの問題。 |
| 106007 | Cannot bring “NFS or HA-NFS” resource “TAG” In Service on server “SERVER” | 他のエラーメッセージを参考にして、実行するべき処置を決定します。問題を修正した後、手動でリソースを In Service にしてください。 |
| 106010 | NFS is not running on server “SERVER”. LifeKeeper will attempt to restart NFS | このダイアログボックスは情報の表示専用です。LifeKeeper は自動的に NFS デーモンの再起動を試みます。デーモンの再起動時に LifeKeeper に問題が発生した場合は、NFS の起動に失敗したというメッセージが表示されます。 |
| 106011 | Starting NFS on server “SERVER" failed | NFS デーモンの再起動中に問題が発生しました。手動で NFS の再起動をしてください。 |
| 106012 | The export point “EXPORT POINT” is not exported on server “SERVER”. LifeKeeper will attempt to export the entry | LifeKeeper によって、エクスポートポイントがエクスポートされていないことが検出され、エクスポートが試みられます。 |
| 106013 | Unable to export "EXPORT POINT" on server "SERVER" | ファイルシステムを手動でエクスポートします。 |
| 106019 | Executing command: "COMMAND" | このメッセージは、LifeKeeper が NFS デーモンを再起動するとき、またはエクスポートポイントをエクスポート、アンエクスポートするときに表示されます。問題が発生した場合、この情報が役に立ちます。 |
| 106024 | Unable to stop and restart rpc.mountd on “SERVER”. | 階層リストア中に rpc.mountd デーモンプロセスを再開する必要がありましたが、このプロセスが失敗しました。プロセスを手動で停止および再開して、エラーとその処置を判別してください。 |
| 106027 | Open of "FILE NAME" on server "SERVER" failed: "ERROR MSG" | 記述された理由により、ファイルを開けませんでした。 |
| 106028 | Mount of /proc/fs/nfsd failed on server | 2.6 以降のカーネルでは、クライアント認証に |

| | | |
|---|---|---|
| | “SERVER” | /proc/fs/nfsd が使用されますが、そのマウント試行が失敗しました。手動で /proc/fs/nfsd をマウントし、失敗内容を判別してください。 |
| 106030 | Unable to restore client info for "CLIENT" on server "SERVER": "ERROR MSG" | クライアントのロックのフェイルオーバーに失敗しました。障害を修正して、再度階層をリストアしてください。 |
| 106039 | Open of "FILE" on server "SERVER" failed: “ERROR MSG”<br>or<br>Attempt to get exclusive lock on "FILE" on server "SERVER" failed: "ERROR MSG" | ファイルを開くかファイルの排他ロックを取得しようとしましたが、失敗しました。原因については、エラーメッセージを参照してください。 |
| 106040 | Multiple virtual IP addresses detected. In this release NFS lock failover only supports one virtual IP address. | NFS リソース階層を作成し直して、1 つの仮想 IP アドレスのみを使用するか、または LifeKeeper デフォルトファイルで FAILOVERNFSLOCKS を false に設定してください。 |
| 106052 | Unable to mount rpc_pipefs on "SERVER". Reason: "REASON". | REASON により rpc_pipefs は、SERVER でマウントされませんでした。 |
| 106053 | *rpc_pipefs* successfully mounted on "SERVER" | SERVER で *rpc_pipefs* が正常にマウントされました。 |
| 106054 | Ping test of the nfsd processes on “SERVER” failed. | NFS サーバーの応答確認に失敗しました。 |
| 106055 | The attempted restart of the nfsd processes on “SERVER” failed. | nfsd プロセスを再起動しましたが、NFS サーバーの応答がありません。 |
| 106056 | Attempting to kill hung nfsd processes on “SERVER”. | NFS サーバーの応答がなかったので、nfsd プロセスを kill して再起動します。 |
| 106057 | Failed to open new exportInfo file “TMP FILENAME”. Reason: “ERROR MSG”. | 記述された理由により、一時ファイルのオープンに失敗しました。 |
| 106058 | Failed to add the entry “ENTRY” to “TMP FILENAME”. Reason: “ERROR MSG”. | 記述された理由により、一時ファイルへの書き込みに失敗しました。 |
| 106059 | Failed to close new exportInfo file “TMP FILENAME”. Reason: “ERROR MSG”. | 記述された理由により、一時ファイルのクローズに失敗しました。 |
| 106060 | Failed to create the new exportInfo file “TMP FILENAME” as it’s size of “NEW SIZE” is less than the size of the current exportInfo with a size of “OLD SIZE”. The size should be at least the same as the old file size. | 新しく作成した exportinfo ファイルが予期したサイズではありません。 |
| 106061 | Failed to rename temp exportInfo file from “TMP FILENAME” to “FILENAME”. Reason: “ERROR MSG”. | 記述された理由により、ファイル名の変更に失敗しました。 |

| | | |
|---|---|---|
| 106062 | fsid(FSID) does not match between resource info and etab on “SERVER”. | 期待した fsid が etab に含まれていません。 |
| 106063 | Failed due to mismatch between InfoFile and etab on “SERVER”. | 期待通りにエクスポート出来ていません。 |
| 106064 | Pseudo file system (fsid=0) protected by NFS RK is not supported in LifeKeeper v9.6.0 and later. Please delete the “TAG” resource whose export point is “EXPORT POINT”. | LifeKeeper v9.6.0 から 擬似ファイルシステム (fsid=0) のリソースはサポートされなくなりました。リソース階層を削除してください。 |
| 106065 | The exportfs command appears to have hung during the “restore” action. Aborting the “restore”. | exportfs コマンドがハングアップしたために、restore を中止しました。exportfs のハングアップを解消してください。ハングアップが解消しない場合は、システムを再起動してください。<br>なお、exportfs コマンドのハングアップ判定については、 NFSパラメータ一覧 の NFSEXPORTFSTIMER を参照してください。 |
| 106066 | Forcefully killing hung process:ps=“PROCESS REPORT”. | exportfs コマンドがハングアップしたため、KILL シグナルを送信しました。記述されている PROCESS REPORT は、KILL シグナルを送信する前の “ps -f” コマンドの出力です。 |

## NFS リソースを Out of Service にする(停止)

| エラー番号 | エラーメッセージ | 説明 |
|---|---|---|
| 106008 | Unable to unexport the export point "EXPORT POINT" on server "SERVER" | **exportfs(8)** コマンドを使用して、エクスポートを解除します。 |
| 106014 | Usage: USAGE STRING | 誤った引数で実行されたコマンドの使用。コマンドラインのみ。 |
| 106019 | Executing command: "COMMAND" | このメッセージは、LifeKeeper が NFS デーモンを再起動するとき、またはエクスポートポイントをエクスポート、アンエクスポートするときに表示されます。問題が発生した場合、この情報が役に立ちます。 |
| 106047 | nfsd stalled on “SERVER”. | nfsd プロセスの停止に失敗しました。 |
| 106065 | The exportfs command appears to have hung during the “remove” action. Aborting the “remove”. | exportfs コマンドがハングアップしたために、remove を中止しました。exportfs のハングアップを解消してください。ハングアップが解消しない場合は、システムを再起動してください。<br>なお、exportfs コマンドのハングアップ判定については、 NFSパラメータ一覧 の NFSEXPORTFSTIMER を参照してください。 |
| 106066 | Forcefully killing hung process:ps=“PROCESS REPORT”. | exportfs コマンドがハングアップしたため、KILL シグナルを送信しました。記述されている PROCESS REPORT は、KILL シグナルを送信する前の “ps -f” コマンドの出力です。 |

## NFS リソースの quickCheck

| エラー番号 | エラーメッセージ | 説明 |
|---|---|---|
| 106040 | Multiple virtual IP addresses detected. In this release NFS lock failover only supports one virtual IP address. | NFS リソース階層を作成し直して、1 つの仮想 IP アドレスのみを使用するか、または LifeKeeper デフォルトファイルで FAILOVERNFSLOCKS を false に設定してください。 |
| 106054 | Ping test of the nfsd processes on “SERVER” failed. | NFS サーバーの応答確認に失敗しました。 |
| 106063 | Failed due to mismatch between InfoFile and etab on “SERVER”. | 期待通りにエクスポート出来ていません。 |

## NFS リソースを In Service に戻す (ローカルリカバリー)

| エラー番号 | エラーメッセージ | 説明 |
|---|---|---|
| 106010 | NFS is not running on server “SERVER”. LifeKeeper will attempt to restart NFS | このダイアログボックスは情報の表示専用です。LifeKeeper は自動的に NFS デーモンの再起動を試みます。デーモンの再起動時に LifeKeeper に問題が発生した場合は、NFS の起動に失敗したというメッセージが表示されます。 |
| 106011 | Starting NFS on server “SERVER" failed | NFS デーモンの再起動中に問題が発生しました。手動で NFS の再起動をしてください。 |
| 106012 | The export point “EXPORT POINT” is not exported on server “SERVER”. LifeKeeper will attempt to export the entry | LifeKeeper によって、エクスポートポイントがエクスポートされていないことが検出され、エクスポートが試みられます。 |
| 106013 | Unable to export "EXPORT POINT" on server "SERVER" | ファイルシステムを手動でエクスポートします。 |
| 106019 | Executing command: "COMMAND" | このメッセージは、LifeKeeper が NFS デーモンを再起動するとき、またはエクスポートポイントをエクスポート、アンエクスポートするときに表示されます。問題が発生した場合、この情報が役に立ちます。 |
| 106027 | Open of "FILE NAME" on server "SERVER" failed: "ERROR MSG" | 記述された理由により、ファイルを開けませんでした。 |
| 106028 | Mount of /proc/fs/nfsd failed on server “SERVER” | 2.6 以降のカーネルでは、クライアント認証に /proc/fs/nfsd が使用されますが、そのマウント試行が失敗しました。手動で /proc/fs/nfsd をマウントし、失敗内容を判別してください。 |
| 106030 | Unable to restore client info for | クライアントのロックのフェイルオーバーに失敗しました。障 |

| | | |
|---|---|---|
| | "CLIENT" on server "SERVER": "ERROR MSG" | 害を修正して、再度階層をリストアしてください。 |
| 106039 | Open of "FILE" on server "SERVER" failed: “ERROR MSG”<br>or<br>Attempt to get exclusive lock on "FILE" on server "SERVER" failed: "ERROR MSG" | ファイルを開くかファイルの排他ロックを取得しようとしましたが、失敗しました。原因については、エラーメッセージを参照してください。 |
| 106040 | Multiple virtual IP addresses detected. In this release NFS lock failover only supports one virtual IP address. | NFS リソース階層を作成し直して、1 つの仮想 IP アドレスのみを使用するか、または LifeKeeper デフォルトファイルで FAILOVERNFSLOCKS を false に設定してください。 |
| 106052 | Unable to mount rpc_pipefs on "SERVER". Reason: "REASON". | REASON により rpc_pipefs は、SERVER でマウントされませんでした。 |
| 106053 | *rpc_pipefs* successfully mounted on "SERVER" | SERVER で *rpc_pipefs* が正常にマウントされました。 |
| 106054 | Ping test of the nfsd processes on “SERVER” failed. | NFS サーバーの応答確認に失敗しました。 |
| 106055 | The attempted restart of the nfsd processes on “SERVER” failed. | nfsd プロセスを再起動しましたが、NFS サーバーの応答がありません。 |
| 106056 | Attempting to kill hung nfsd processes on “SERVER”. | NFS サーバーの応答がなかったので、nfsd プロセスを kill して再起動します。 |
| 106063 | Failed due to mismatch between InfoFile and etab on “SERVER”. | 期待通りにエクスポート出来ていません。 |
| 106065 | The exportfs command appears to have hung during the “recover” action. Aborting the “recover”. | exportfs コマンドがハングアップしたために、recover を中止しました。exportfs のハングアップを解消してください。ハングアップが解消しない場合は、システムを再起動してください。なお、exportfs コマンドのハングアップ判定については、 NFS パラメーター一覧 の NFSEXPORTFSTIMER を参照してください。 |
| 106066 | Forcefully killing hung process:ps=“PROCESS REPORT”. | exportfs コマンドがハングアップしたため、KILL シグナルを送信しました。記述されている PROCESS REPORT は、KILL シグナルを送信する前の “ps -f” コマンドの出力です。 |

# 6.10.5.5. NFS リソース階層削除のメッセージとエラー

| エラー番号 | エラーメッセージ | 説明 |
|---|---|---|
| 106015 | Unable to restore the entry for export point "EXPORT POINT" in /etc/exports on server “SERVER" | エントリーを手動でリストアします。 |
| 106021 | An entry for export point "EXPORT POINT" already exists in /etc/exports. The entry that was being used by the NFS Server Recovery Kit has been placed in the file "FILENAME" | /etc/exports のエクスポートエントリーが正しいか確認します。 |
| 106049 | Restore statedir from /var/lib/.nfs.LK to /var/lib/nfs failed on server “SERVER” | NFS ステートディレクトリー */var/lib/nfs* のリストアに失敗しました。 */var/lib/.nfs.LK* を */var/lib/nfs* へ移動させ、手動でディレクトリーをリストアしてください。 |

# 6.11. Recovery Kit for Oracle® Cloud Infrastructure 管理ガイド

## Recovery Kit for Oracle Cloud Infrastructure (RK for OCI)

RK for OCI は、Oracle Cloud Infrastructure (OCI) 上で IP Recovery Kit による仮想 IP アドレスを利用するための仕組みを提供します。

RK for OCI の機能、シナリオ、および運用の詳細については、運用の原則 を参照してください。

### SIOS Protection Suite/LifeKeeper ドキュメンテーション

以下は、SIOS Technology Corp が発行している SIOS Protection Suite for Linux 関連ドキュメントの一覧です。

- SPS for Linux テクニカルドキュメンテーション
- SPS for Linux リリースノート
- SIOS Technology Corp. ドキュメンテーション

「Oracle Cloud」は、Oracle Corporation およびその関連会社の登録商標です。なお、本文および図表中では、「™」、「®」は明記しておりません。(一部記載がございます。)

# 6.11.1. 運用の原則

Recovery Kit for Oracle Cloud Infrastructure (RK for OCI) は、LifeKeeper が保護する IP リソース（仮想 IP アドレス）に OCI 上のクライアントから接続できるように、セカンダリー・プライベート IP アドレスの仮想ネットワーク・インタフェース・カード (VNIC) への割り当てを管理します。

https://docs.oracle.com/ja-jp/iaas/Content/Network/Tasks/managingIPaddresses.htm

OCI 上で仮想 IP アドレスを利用するには、OS 上での IP アドレスの割り当てだけでなく、対象の IP アドレスをアクティブノードの VNIC へ割り当てる必要があります。

LifeKeeper では、OS 上での IP アドレスの割り当ては IP Recovery Kit が行います。

RK for OCI は、クラスターノード上で OCI CLI を実行し、セカンダリー・プライベート IP アドレスを OCI 上のアクティブノードの VNIC への割り当てを管理します。

また RK for OCI は、保護対象の IP アドレスがアクティブノード上の指定した VNIC に割り当てられているかを監視します。

アクティブノードに障害が発生すると、 RK for OCI は IP アドレスをスタンバイノードの VNIC に割り当て、仮想 IP アドレスを利用できる状態にします。

より詳細な動作については リソースの監視とローカルリカバリー動作 をご参考ください。

# 6.11.2. リソースの監視とローカルリカバリー動作

## Recovery Kit for Oracle Cloud Infrastructure (RK for OCI) のクイックチェック

RK for OCI は次のコマンドを実行して、セカンダリー・プライベート IP アドレスがアクティブノードの VNIC に割り当てられていることを確認します。

```
oci network private-ip list --vnic-id <vnicid> --query 'data[?"is-primary" == `false`]."ip-address"'
```

成功するケース

- 保護対象の IP アドレスが返り値に含まれている

失敗するケースの例

- `oci` コマンドの実行に失敗する
- 保護対象の IP アドレスが返り値に含まれていない

RK for OCI のクイックチェックに失敗した場合、RK for OCI のローカルリカバリーを実行します。

## RK for OCI のローカルリカバリー

RK for OCI のクイックチェックに失敗すると、RK for OCI のローカルリカバリーを実行します。

ローカルリカバリーでは、次のコマンドを実行して、セカンダリー・プライベート IP アドレスをアクティブノードの VNIC に割り当てます。

```
oci network vnic assign-private-ip --vnic-id <vnicid> --ip-address <ip>
```

`oci` コマンドの失敗や、IP アドレスが既に他のインスタンスに割り当てられているなどの理由でローカルリカバリーに失敗すると、
LifeKeeper は RK for OCI リソース、および依存関係を持つすべてのリソースをスタンバイノードにフェイルオーバーします。
フェイルオーバー時にスタンバイノードで実施される In Service 処理では、IP アドレスを奪取するコマンド

```
oci network vnic assign-private-ip --unassign-if-already-assigned --vnic-id <vnicid> --ip-address <ip>
```

を実行するため、クラスターノード外のインスタンスに IP アドレスが割り当てられていても、フェイルオ

ーバーに成功します。

# 6.11.3. 要件

## Oracle Cloud Infrastructure およびソフトウェアの要件

Recovery Kit for Oracle Cloud Infrastructure (RK for OCI) をインストールして設定する前に、使用している環境が以下の要件を満たしていることを確認してください。

### Oracle Cloud Infrastructure コンピュート・インスタンス:

- OCI が提供するベア・メタル・インスタンス、または仮想マシン・インスタンスであること
- クラスターノードとして使用するインスタンス上で OCI が提供する コマンドライン・インタフェース 3.0.0 以上を利用できること
- クラスターノードとして使用するインスタンス上で OCI が提供する インスタンス・メタデータ・サービス v1 を利用できること
    - IMDSv2 には未対応です。

### Oracle Cloud Infrastructure コマンドライン・インターフェース (CLI):

- 設定ファイル */etc/default/LifeKeeper* のパラメーター PATH に `oci` コマンドのパスを追加してください。
- RK for OCI が OCI を操作するために、 クラスターノードの root ユーザーが以下のコマンドを実行できるように、 CLI、および権限を設定してください。CLI 、および権限の設定に関しては OCI のドキュメントを参照してください。

```
oci network vnic assign-private-ip --vnic-id <vnicid> --ip-address <ip>
oci network vnic unassign-private-ip --vnic-id <vnicid> --ip-address <i
p>
oci network private-ip list --vnic-id <vnicid>
oci network vnic get --vnic-id <vnicid>
oci network private-ip list --subnet-id <subnetid>
```

<ip> は LifeKeeper で保護する IP アドレスです。
<vnicid> は IP アドレスを割り当てる仮想ネットワーク・インタフェース・カード (VNIC) の OCID です。
<subnetid> は VNIC が関連付けられているサブネットの OCID です。

### 利用する IP アドレス:

- RK for OCI で保護するプライベート IP アドレスは、IPv4 である必要があります。
    - IPv6 はサポート対象外です。
- RK for OCI で保護するプライベート IP アドレスは、VNIC に割り当て可能な IP アドレスにする必要があります。
    - 割り当て対象の VNIC が関連付けられているサブネット内の IP アドレスである。
    - OCI が予約済みの IP アドレスでない。
    - セカンダリー・プライベート IP アドレスの割り当て可能数を超えていない。

# LifeKeeper ソフトウェア:

各サーバーに同じバージョンの LifeKeeper ソフトウェアとパッチをインストールする必要があります。具体的な LifeKeeper の要件については、SPS for Linux テクニカルドキュメンテーション および SPS for Linux リリースノート を参照してください。

## LifeKeeper Recovery Kit for Oracle Cloud Infrastructure:

各サーバーに同じバージョンの RK for OCI ソフトウェアとパッチをインストールする必要があります。

## LifeKeeper IP Recovery Kit:

各サーバーに同じバージョンの LifeKeeper for Linux IP Recovery Kit ソフトウェアとパッチをインストールする必要があります。

最新リリースの互換性と発注情報については、 SPS for Linux リリースノート を参照するか、営業担当者に問い合わせてください。
LifeKeeper RK for OCI のインストール / アンインストール方法の具体的な手順については、 SIOS Protection Suite インストールガイド を参照してください。

RK for OCI を使う場合には、Quorum/Witness の使用を推奨します。

# 6.11.4. Recovery Kit for Oracle Cloud Infrastructure を利用する際の注意事項

- Recovery Kit for Oracle Cloud Infrastructure (RK for OCI) リソースを作成すると、対応する IP リソースが作成されます。
  RK for OCI リソース作成前に、既に IP リソースが存在している場合、
  - RK for OCI リソース の設定と一致していれば既存の IP リソースとの依存関係が自動で作成されます。
  - RK for OCI リソース の設定と一致していない、つまりデバイス、またはネットマスクの値が一致していない場合、 IP リソースと RK for OCI リソースの依存関係は作成されません。 IP リソースを削除してから RK for OCI リソースを再度作成するか、 IP リソースの設定を適切に変更してから依存関係を作成してください。
- LifeKeeper IP リソースのブロードキャスト PING チェックを無効にする必要があります。 */etc/default/LifeKeeper* の設定ファイルの NOBCASTPING エントリーを以下のように設定してください。

```
NOBCASTPING=1
```

- ocid.service を無効にする必要があります。

# 6.11.5. 設定

このページでは、設定上の考慮事項に加え、Recovery Kit for Oracle Cloud Infrastructure (RK for OCI) の設定作業を説明します。

## RK for OCI 固有の設定上の考慮事項

RK for OCI を利用する前に、以下のページを見直して、要件を満たしていることを確認してください。

- 要件
- Recovery Kit for Oracle Cloud Infrastructure を利用する際の注意事項

## RK for OCI の設定作業

- リソース階層の作成 : RK for OCI リソースを LifeKeeper クラスターに作成します。
- リソース階層の削除 : RK for OCI リソースを LifeKeeper クラスターを構成するすべてのノードから削除します。
- リソース階層の拡張 : RK for OCI リソースをプライマリーノードからセカンダリーノードへ拡張します。
- リソース階層の拡張解除 : RK for OCI リソースを LifeKeeper クラスターの 1 つのノードから拡張解除 (削除) します。
- Recovery Kit for Oracle Cloud Infrastructure のチューニング: RK for OCI で利用できるパラメーターを記載します。

すべての Recovery Kit に共通の設定作業は、SPS for Linux テクニカルドキュメンテーション の 管理 セクションに記載されています。

- リソース依存関係の作成 : 既存のリソースと別のリソースとの間に親子の依存関係を作成し、リソースが拡張されているノードに依存関係の変更を反映します。
- リソース依存関係の削除 : リソースの依存関係を削除し、リソースが拡張されているノードに依存関係の変更を反映します。
- In Service : リソース階層を特定のノードで In Service にします。
- Out of Service : リソース階層を特定のノードで Out of Service にします。
- プロパティの表示 / プロパティの編集 : 特定のノードでリソース階層のプロパティーを表示または編集します。

# 6.11.5.1. Recovery Kit for Oracle Cloud Infrastructure リソース階層の作成

Recovery Kit for Oracle Cloud Infrastructure (RK for OCI) リソースの作成は、以下の手順で行います。

1. リソース階層の作成 を参考に **[Create Resource Wizard]** を起動します。
2. Recovery Kit として **[OCIVIP]** を選択します。
3. 以下のパラメーターを入力します。

> 階層作成の途中で **[Cancel]** ボタンをクリックすると、作成処理全体が取り消されます。

| フィールド | ヒント |
|---|---|
| Switchback Type | フェイルオーバー後にこのノードが復帰したときに、このノードに RK for OCI リソースを戻す方法を指定します。[intelligent] または [automatic] を選択できます。<br><br>• [intelligent] の場合、リソースのスイッチバックを手動で行う必要があります。<br>• [automatic] の場合、他のノードとの LifeKeeper コミュニケーションパスが再確立したら、自動的にスイッチバックが行われます。<br><br>**注記:** スイッチバックタイプは、[Resource Properties] ダイアログボックスの [General] タブで後から変更できます。 |
| Server | RK for OCI リソースを作成するノードを選択します。クラスター内のすべてのノードがドロップダウンリストに表示されます。 |
| Secondary IP address | セカンダリー・プライベート IP アドレスを入力します。ここで指定した IP アドレスが VNIC に割り当てられます。 |
| Network Interface | IP アドレスを割り当てるネットワークインターフェース名を選択します。IP アドレスを割り当てることができる VNIC に対応するネットワークインターフェース名から選択できます。 |
| OCIVIP Resource Tag | 作成している RK for OCI リソースに固有の タグ名を入力します。このフィールドには、デフォルトのタグ名である ocivip-<ip> が自動的に表示されます。<ip> は指定したセカンダリー・プライベート IP アドレスです。 |

4. 全てのパラメーターを入力すると、リソースを作成し In Service にします。LifeKeeper が問題を検知した場合は、情報ボックス、あるいはログファイルにエラーが表示されます。
5. 対応する IP リソースを作成し In Service にします。IP リソースの IP アドレス、ネットワークインターフェース名は RK for OCI リソースと同じです。またネットマスクの値は、 IP アドレスを割り当てる VNIC が関連付けられているサブネットのサブネットマスク（CIDR）の値です。
   a. 既に IP リソースが存在している場合、対応する IP リソースの作成をスキップします。
6. IP リソースを親、RK for OCI リソースを子として依存関係を作成します。
   a. 5. で新規に IP リソースを作成した場合、依存関係を作成します。
   b. 5. で既に IP リソースが存在しており、IP リソースの作成をスキップした場合、既存の IP リソースの IP アドレスとネットワークインターフェース名、ネットマスクの値が、

i. RK for OCI リソースの設定値と完全一致していれば、依存関係を作成します。
ii. RK for OCI リソースの設定値と矛盾していれば、その IP リソースは RK for OCI リソース の親リソースとして不適当なリソースであるため、依存関係を作成しません。 IP リソースの設定を変更するか、 IP リソースを削除し RK for OCI リソースを再度作成してください。

リソースの作成に成功すると、 **[Pre-Extend configuration task]** が起動します。

リソース階層を別のノードに拡張する方法の詳細については、リソース階層の拡張 を参照してください。

# 6.11.5.2. Recovery Kit for Oracle Cloud Infrastructure リソース階層の削除

Recovery Kit for Oracle Cloud Infrastructure (RK for OCI) リソースの削除は、すべてのサーバからの階層の削除 を参照してください。

> ! RK for OCI リソースを削除する際、リソースの Out of Service、すなわち VNIC からの IP アドレスの割り当て解除が行われます。

# 6.11.5.3. Recovery Kit for Oracle Cloud Infrastructure リソース階層の拡張

Recovery Kit for Oracle Cloud Infrastructure (RK for OCI) リソース階層の拡張は、以下の手順で行います。

1. リソース階層の拡張 を参考に **[Extend Resource Hierarchy]** ウィザードを起動します。
2. **[Pre-Extend Wizard]** で以下の情報を入力します。

| フィールド | ヒント |
|---|---|
| Switchback Type | フェイルオーバー後に拡張先ノードが復帰したときに、そのノードに RK for OCI リソースを戻す方法を指定します。[intelligent] または [automatic] を選択できます。<br><br>• [intelligent] の場合、リソースのスイッチバックを手動で行う必要があります。<br>• [automatic] の場合、他のノードとの LifeKeeper コミュニケーションパスが再確立したら、自動的にスイッチバックが行われます。<br><br>**注記**: スイッチバックタイプは、[Resource Properties] ダイアログボックスの [General] タブで後から変更できます。 |
| Template Priority | 拡張元ノードのリソース階層の優先順位を入力します。1 ～ 999 の範囲で、まだ優先順位として使用されていない値を指定できます。小さい数値ほど優先順位が高くなります。デフォルト値を使用することを推奨します。この数字はリソースのカスケーディングフェイルオーバシーケンスにおけるサーバの優先順位を示します。<br><br>**注記**: このフィールドは、リソース階層を最初に拡張するときにだけ表示されます。 |
| Target Priority | 拡張先ノードのリソース階層の優先順位を入力します。1 ～ 999 の範囲で、まだ優先順位として使用されていない値を指定できます。数値が小さいほど優先順位が高くなります。この数字はリソースのカスケーディングフェイルオーバシーケンスにおけるサーバの優先順位を示します。<br><br>**注記**: LifeKeeper のデフォルトでは、階層が作成されたサーバに「1」が割り当てられます。優先順位は連続している必要はありませんが、特定のリソースについて 2 つのサーバに同じ優先順位を割り当てることはできません。 |

3. 対象の RK for OCI リソースの拡張要件を満たしていることを確認するために CanExtend が実行されます。
   a. 拡張先ノードに、 IP アドレスを割り当て可能な VNIC が存在していることを確認します。
4. 以下のパラメーターを入力し、リソースの拡張を行います。

| フィールド | ヒント |
|---|---|
| Secondary IP address | 拡張元リソースの IP アドレスが表示されます。値の変更はできません。 |

| | |
|---|---|
| Network Interface | IP アドレスを割り当てるネットワークインターフェース名を選択します。IP アドレスを割り当てることができる VNIC に対応するネットワークインターフェース名から選択できます。 |
| OCIVIP Resource Tag | RK for OCI リソースタグを入力します。これは、スタンバイノードに拡張される RK for OCI リソースが使用するリソースタグ名です。デフォルトでは、拡張元ノードでのリソースタグ名が表示されます。 |

5. 同じ RK for OCI リソースをクラスター内の別のノードに拡張する場合は、 **[Next Server]** をクリックし、同様の操作を繰り返します。 **[Finish]** をクリックすると、LifeKeeper は RK for OCI リソースの拡張が正常に完了したことを確認します。

> ! すべてのノードで RK for OCI リソースの機能をテストしてください。

# 6.11.5.4. Recovery Kit for Oracle Cloud Infrastructure リソース階層の拡張解除

Recovery Kit for Oracle Cloud Infrastructure リソースの拡張解除は、階層の拡張解除 を参照してください。

# 6.11.5.5. Recovery Kit for Oracle Cloud Infrastructure のチューニング

Recovery Kit for Oracle Cloud Infrastructure で設定可能なパラメータは Recovery Kit for Oracle Cloud Infrastructure パラメーター一覧 を参照してください。

# 6.11.6. トラブルシューティング

現在、FAQ はありません。

# 6.11.6.1. 既知の問題と制限事項

## 既知の問題

| 説明 |
| --- |
| リソースのリストア時に、32 bit のサブネットマスクの IP が付与されることがある。<br><br>**原因**: RK for OCI と IP リソースのリストア処理の間に、ocid.service が担う VNIC のスキャンと更新処理が実行されることで発生します。<br><br>**解決方法**: ocid.service を無効化してください。 |
| Recovery Kit for Oracle Cloud Infrastructure (RK for OCI) リソース作成時に、対応する IP リソースの起動処理に失敗した場合、依存関係が作成されない。<br><br>**解決方法**: IP リソースを正しく起動できるように設定した後、手動で RK for OCI リソースと IP リソースの依存関係を作成してください。IP リソースが親リソース、RK for OCI リソースが子リソースに指定してください。 |
| lkcli import 時に RK for OCI リソースに対応する IP リソース に関して、以下のメッセージが出力される。<br><br>`Resource '<ip-res>' is already exists. skipping <ip-res>.`<br><br>**原因**: IP リソースの作成より先に実行される RK for OCI リソース作成時に、対応する IP リソースを一緒に作成しているためです。<br><br>**解決方法**: 設定ファイル (YAML 形式) に記載されている全リソースが過不足なく作成されていれば、問題はありません。そうでない場合、残りのリソースを手動でセットアップしてください。あるいは、RK for OCI リソースを除外した import を再度実施し、RK for OCI リソースのみを手動で作成してください。 |
| BM インスタンスに LifeKeeper をインストールする際、セットアップメニューの Recovery Kit 選択画面において、”Recovery Kit for Oracle Cloud Infrastructure” が表示されず、インストールする Recovery Kit として選択できない。<br><br>**解決方法**: セットアップ実行時に以下のオプションを指定してください。<br><br>`--env oci`<br><br>あるいは、create_response_file スクリプトで構成情報ファイルを作成 し、非対話的インストールを行ってください。 |

## 制限事項

| 説明 |
| --- |

RK for OCI が利用する OCI CLI において任意の IPv6 アドレスを割り当てできないバグにより、 IPv6 アドレスは利用できません。
https://github.com/oracle/oci-cli/issues/421

# 6.11.6.2. エラーメッセージ

Recovery Kit for Oracle Cloud Infrastructure のエラーメッセージは Recovery Kit for Oracle Cloud Infrastructure メッセージカタログ を参照してください。

# 6.12. Oracle® Recovery Kit 管理ガイド

## Oracle Recovery Kit ドキュメンテーション

SIOS Protection Suite for Linux Oracle® Recovery Kit は、 SPS 環境で Oracle ソフトウェアの障害回復能力を高めます。Recovery Kit ソフトウェアは、Oracle データベースのデータ整合性と LifeKeeper SIOS Protection Suite によって実現される可用性の向上を結び付けるメカニズムを提供します。

## ドキュメントの内容

このガイドでは次の事項について説明します。

SPS for Linux テクニカルドキュメンテーション (LifeKeeper GUI 内の [Help] メニューからも参照できます) 。 SPS for Linux のすべてのドキュメントおよび情報が提供されている場所のリスト。

要件 。リカバリソフトウェアをインストールおよびセットアップするには、サーバが一定のハードウェア／ソフトウェア要件を満たしている必要があります。LifeKeeper Oracle Recovery Kit のインストールまたは削除の詳しい手順については、SIOS Protection Suite Installation Guide を参照してください。

Recovery Kit の設定 。 必要とする保護と柔軟性が得られるように SPS を設定するには、設定ルールを把握している必要があります。設定を適切にプランニングするには、ネットワーク設定、インターフェースの選択、ユーザシステムのセットアップ、階層オプション、Oracle 設定作業についての理解が必要です。この「要件」セクションでは、設定のプランニングだけでなく、設定の例を示し、Recovery Kit を設定するために必要な特定のタスクについて説明します。

トラブルシューティング このセクションでは、情報メッセージとエラーメッセージ、および推奨される対処方法の一覧を示します。

# 6.12.1. Oracle Recovery Kit のハードウェアとソフトウェアの要件

LifeKeeper Oracle Recovery Kit のインストール / アンインストールを試みる前に、ハードウェア / ソフトウェア要件とインストール / アンインストールの手順を理解しておく必要があります。
設定が以下の要件を満たしていることを確認してください。

**サーバー** この Recovery Kit には、LifeKeeper テクニカルドキュメンテーション および LifeKeeper リリースノート で説明されている LifeKeeper の要件に従って設定した 2 台以上のサポートされているコンピュータが必要です。

**共有ストレージ** Oracle データベースは LifeKeeper環境では共有ディスク上に存在する必要があります。共有ストレージのアーキテクチャーに応じて、適切な LifeKeeper 共有ストレージまたは multipath Recovery Kitをクラスター内の各ノードにインストールする必要があります。NFSでバックアップされたデーターベースストレージの例では、LifeKeeper NAS Recovery Kit のインストールが必要です。データレプリケーション（ミラーリング）環境でLifeKeeperを使用する場合は、SIOS DataKeeper管理ガイド を参照してください。Network Attached Storage（NAS）環境で LifeKeeperを使用する場合は、LifeKeeper Network Attached Storage(NAS) Recovery Kit 管理ガイド を参照してください。

**LifeKeeper ソフトウェア** クラスタの各サーバーにインストールする LifeKeeper ソフトウェアのバージョン、および適用する LifeKeeper ソフトウェアパッチのバージョンが同一であることが絶対条件です。LifeKeeperの特定の要件については、LifeKeeperリリースノート および LifeKeeper テクニカルドキュメンテーション を参照してください。

**LifeKeeper IP Recovery Kit** リモートクライアントが Oracle データベースにアクセスする場合に、この Recovery Kit が必要になります。同じバージョンの Recovery Kit を各サーバにインストールする必要があります。

**IP ネットワークインターフェース** 各サーバーは、イーサーネット TCP/IP をサポートするネットワークインターフェースを 1 つ以上必要とします。IP 切り替えが正しく動作するには、ローカルネットワークに接続されているユーザシステムが標準の TCP/IP 仕様に準拠している必要があります。

**注記:** 各サーバが必要とするネットワークインターフェースが 1 つだけであっても、異種メディア要件、スループット要件、単一障害点の排除、ネットワークのセグメンテーションといった多くの理由のために、複数のインターフェースを使用すべきです。

**TCP/IP ソフトウェア** 各サーバーは TCP/IP ソフトウェアも必要とします。

**Oracle ソフトウェア** Oracle ソフトウェアが各サーバーにインストールおよび設定されていなければ、LifeKeeper と LifeKeeper Oracle Recovery Kit の設定はできません。各サーバーには同じバージョンのをインストールする必要があります。最新のリリースの互換性および購入方法については、LifeKeeperリリースノート を参照するか、営業担当者にお問い合わせください。

LifeKeeper Oracle Recovery Kit をインストールまたはアンインストールする具体的な手順については、

LifeKeeperインストールガイド を参照してください。

# 6.12.2. LifeKeeper での Oracle の設定

本セクションでは、Oracle の設定を開始する前に検討すべき情報と、一般的な Oracle の設定例を示します。

以下の作業は、すべての Recovery Kit を通じて同一の手順からなる共通の作業のため、SPS for Linux テクニカルドキュメンテーション の 管理 セクションで説明しています。

- リソース依存関係の作成 。既存のリソース階層と別のリソースインスタンスとの間に親子の依存関係を作成し、クラスタ内のすべての対象サーバに依存関係の変更を反映します。
- リソース依存関係の削除 。リソースの依存関係を削除して、クラスタ内のすべての対象サーバに依存関係の変更を反映します。
- In Service 。リソース階層を特定のサーバで in service にします。
- Out of Service 。リソース階層を特定のサーバで out of service にします。
- プロパティの表示 / 編集 。特定のサーバでリソース階層のプロパティを表示または編集します。

2.6 カーネルを使用した Linux ディストリビューションへの Oracle のインストール手順については、Linux ディストリビューションの Web サイトを参照してください。

LifeKeeper Core のリソース階層の設定に関する手順については、SIOS Technology の Web サイトにある SPS for Linux テクニカルドキュメンテーション も参照してください。

# 6.12.2.1. Oracle 特有の設定上の考慮事項

注記 : Raw I/O で Oracle を使用する場合は、あらかじめ Raw I/O デバイスを適切にセットアップしておく必要があります。手順については、付録 を参照してください。(Raw I/O は LifeKeeper Single Server Protection のオプションではありません。)

Oracle Recovery Kit を設定する前に以下の準備作業を完了し、必ず各サーバから Oracle が使用するファイルシステムとディスクパーティションをアクセスできるようにしてください。

1. **個人用初期化ファイルのプロンプトの削除** 。Oracle Recovery Kit が正しく動作するためには、Oracle ユーザの個人用初期化ファイル (つまり、.profile、.bash_profile) のプロンプトをすべて削除 (またはコメントとして除外) する必要があります。このファイルは Oracle ユーザが使用するシェル専用です。対話型にはできません。

注記 :「stty」ステートメントを個人用初期化ファイルで使用する場合は、対話型ターミナルが使用されているかどうか確認する「if」ステートメントに入れる必要があります。

2. **カーネルパラメータの設定** 。linux カーネルパラメータ（共有メモリやその他のカーネルリソースなど）の設定方法については、Oracle ドキュメンテーションを参照してください。これらのパラメータの設定方法の例を次に示します。

クラスタ内の *各サーバ* で次のようにします。

a. LifeKeeper を設定する前に、 */etc/sysctl.conf* で次のように ipcs の限度を設定します。

```
# changes for Oracle
kernel.shmmax = <value>
kernel.shmmni = <value>
kernel.shmall = <value>
kernel.sem = <value>
```

b. `sysctl -p` を実行して、上記の変更をカーネルに反映します。

c. 一部のディストリビューションでは、システム初期化ファイル (boot.local または rc.local) に `sysctl -p` を追加して、再起動のたびに変更をカーネルに反映させる必要があります。

3. **$ORACLE_HOME ディレクトリ** 。 *$ORACLE_HOME* ディレクトリおよび関連ファイルをローカルディスク上に設定するときは、 *$ORACLE_HOME* ディレクトリと各ファイルがすべてのサーバで同じになるように設定します。Linux の標準ユーティリティを使用して、ディレクトリとファイルを作成し、各サーバにコピーしてください。

注記 : 一部のアクティブ / アクティブ設定では、 *$ORACLE_HOME* の場所が異なります。

4. **場所** 。 *$ORACLE_HOME* ディレクトリは共有ディスクまたは非共有ディスク上に置くことができます。ディレクトリを共有メディアに置く利点は、 *$ORACLE_HOME* にも同じ共有ディスクが使用されていれば (例えば、アクティブ / スタンバイ設定など)、パラメータファイル*Oracle_HOME/dbs/<initSID.ora* や *spfileSID.ora>* の設定が 1 回で済むことです。共有ディレクトリの

不利な点は、一度に 1 台のサーバしかファイルシステムへの直接アクセスができないことです。SCSI リザベーションは、LifeKeeper によって保護された共有ドライブへのアクセスを一度に 1 台のサーバにしか許可しません。2 つ以上の Oracle インスタンス (SID) がクラスタ内で個々に保護されるアクティブ / アクティブ設定のクラスタを作成する場合は、 *$ORACLE_HOME* をローカルの非共有ストレージ上に設定する必要があります。

5. **ユーザおよびグループ ID** 。Oracle ユーザ (oracle) とグループ (dba) をすべてのサーバに作成する必要があります。ユーザ ID とグループ ID の番号は、すべてのサーバで同じになるようにしてください。

6. **データベース、アーカイブファイル、ログファイル、制御ファイル** 。すべてのデータベース、アーカイブファイル、ログファイル、制御ファイルを共有ファイルシステムまたはディスクパーティションに作成する必要があります。これらのロケーションは、Oracle パラメータファイル *init<SID>.ora* または *spfileSID.ora* に設定されます。データベースパラメータの編集については、Oracle ドキュメンテーションを参照してください。このパス名は全サーバで同じになるようにしてください。Oracle は内部的にこの情報を制御ファイルに保持しているため、Oracle が稼動中でなければ、SYSTEMS データベースのスペースとパスは変更できません。

**注記** : Oracle ログのアーカイブは、デフォルトでは有効になっていません。LifeKeeper Oracle 階層の作成前に有効になっている場合は、LifeKeeper がアーカイブファイルの場所を検知し、必要に応じて別のファイルシステム階層を作成します。しかし、ログアーカイブが LifeKeeper Oracle 階層作成後に有効になる場合は、共有アーカイブロケーションを保護するために手動でファイルシステム階層を作成および拡張し、Oracle リソースからこの新しいファイルシステム階層へ依存関係を作成してください。

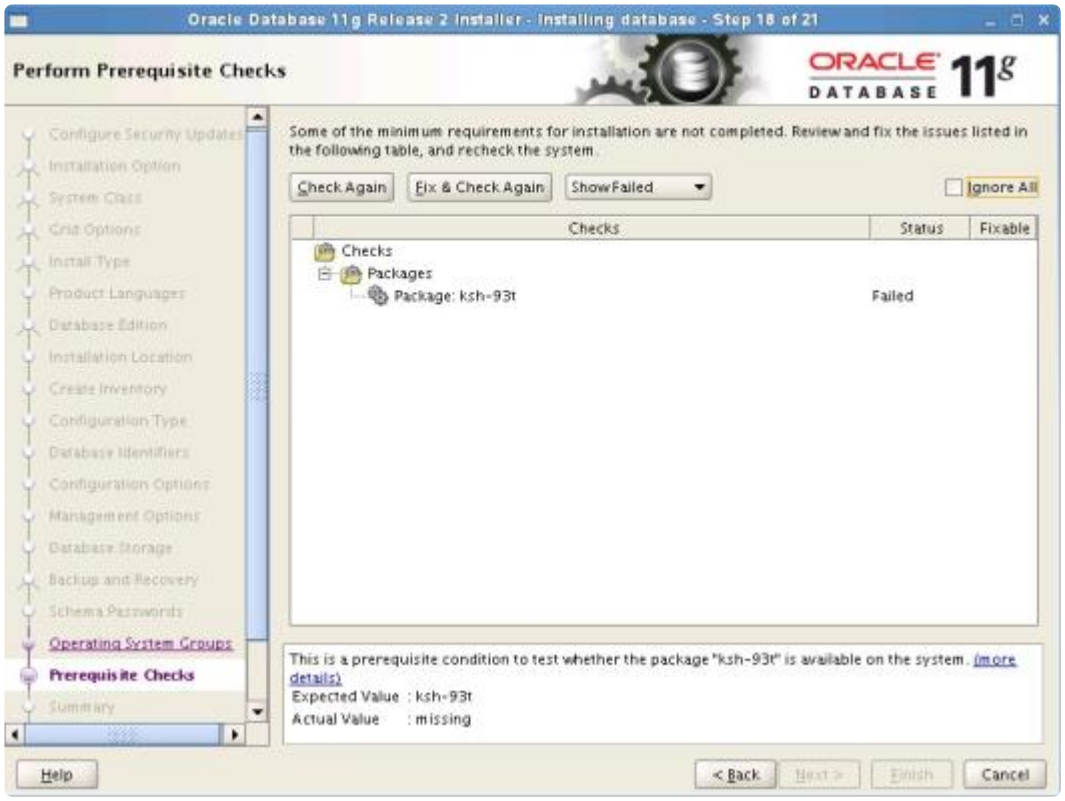
**注記** : ロック付きのストレージアプリケーションを使用し、NFS マウントオプションへの推奨設定を行う場合には、 *rw,nolock,bg,hard,nointr,tcp,nfsvers=3,timeo=600,rsize=32768,wsize=32768,actimeo=0* など追加の nolock オプションを SPS で設定する必要があります。

7. Oracle の新規インストール時にデータベースインストールプログラムでデータベースを作成しない場合、データベースインスタンスの最後の設定が容易になります。インストーラによってデータベースの作成が求められた場合は、 **[No]** を選択してください。インストールが完了したら、 **Oracle Database Creation Assistant** (dbca) を起動してください。dbca は、データベースコンポーネント作成時のインストール先の制御がより優れています。dbca の起動時に、Flash Recovery Area が LifeKeeper で保護するストレージ上に作成されるように指定してください 。

**重要:** Flash Recovery Destination は、共有ドライブ上に配置してください。

runInstaller でデータベースの作成を行った場合は、手動で Flash Recovery Area を再配置する必要があります。( **注記** : `runInstaller` でデータベースを作成できるようにはしないでください。)

8. `runInstaller` ユーティリティを使用した Oracle のインストール中、Oracle データベースのインストールに進む前にインストーラが Linux のパッケージおよび設定を確認するポイントがあります。LifeKeeper 7.2 (もしくは、それ以降のバージョン) がインストールされている場合に、ksh パッケージが見つからないというメッセージが表示されます。




このメッセージが表示された場合、画面右上の **[Ignore All]** にチェックを入れてください。LifeKeeper のインストールにより ksh パッケージが削除され、Public Domain Korn Shell の `pdksh` に置き換えられます。Oracle のインストールは `pdksh` を使用してください。

注記：バージョン 8.0 以降、Oracle ARK では `pdksh` は必要ありません。ただし、LifeKeeper Core ではまだ `pdksh` は必要であるため、 **[Ignore All]** にチェックを入れる必要があります。

注記：バージョン 8.1 以降の LifeKeeper では、独自の `pdksh` パッケージが提供されているため、Oracle の ksh の要件と競合しなくなりました。

9. データベースエンジンの調整。データの整合性とパフォーマンスのためにデータベースエンジンを調整するときのガイドラインについては、Oracle のマニュアルを参照してください。特に、障害回復能力の向上のためにアプリケーションを最適化するときは、メモリキャッシングとチェックポイントの頻度の調整が重要です。チェックポイントの間隔はコミットされていないデータベーストランザクションの数を判定します。その結果、システム障害の発生時に失われるデータベーストランザクションの数が決まります。

10. oratab ファイルのデータベースエントリ。/etc/oratab ファイルにはデータベースに対応したエントリが必要です。LifeKeeper 設定ルーチンは、このファイルの内容を使用して、 *$ORACLE_HOME* と *$ORACLE_SID* の値を関係付けます。通常は、Oracle インストールプログラムが必要なエントリを作成します。ただし、Oracle ソフトウェアが共有ファイルシステムにインストールされた設定では、管理者が oratab ファイルを、Oracle のインストールが実行されたサーバから、他の各サーバの /etc ディレクトリにコピーして、すべてのサーバで利用可能にする必要があります。

注記：サーバごとに設定可能な oratab は 1 つだけです。ファイルフォーマットについては、「Oracle Product Manual」を参照してください。

注記：*oratab* ファイルは */etc* などの場所に収容できます。デフォルトでは、Oracle ARK は *oratab* ファイルをまず */etc* で検索した後、*/var/opt/oracle* でこのファイルを検索します。このいずれかのデフォルトの場所に *oratab* ファイルがない場合は、 ORACLE_ORATABLOC を */etc/default/LifeKeeper* で *oratab* が含まれるディレクトリに設定する必要があります。

11. 自動スタートアップの無効化。LifeKeeper により制御されるデータベースは LifeKeeper により起動されるため、自動スタートアップアクションはすべて無効にしてください。LifeKeeper は、階層が作成されたときに自動スタートアップを無効にします。これは、oratab ファイルを変更することによって実現されます。

12. Listener 設定ファイル、listener.ora。エントリに新しい行を組み込まないでください（例えば、SID_NAME=xx は 1 行にします）。

13. Oracle データベースのユーザ名およびパスワード。LifeKeeper はローカルセッションおよび OS 認証を使用して、Oracle データベースを制御します。セキュリティ上の理由からローカル OS 認証をオフにする場合、LifeKeeper は指定されたユーザ名およびパスワードを使用できます。Oracle データベースユーザは、保護されるデーターベースに対して sysdba 権限として接続可能である必要があり、各サーバの Oracle データベースが同一のユーザ名とパスワードを保有している必要があります。この構成がリソースの作成中にスキップされた場合、LifeKeeper は Oracle データベースリソースを制御するためにユーザ名およびパスワードを使用しません。このパラメータはリソース作成後にいつでも追加、変更、削除することが可能です。

LifeKeeper の保護下に置かれたら、LifeKeeper およびデータベースのユーザ権限を sysdba から sysoper に下げることができます。詳細については、Oracle データベースアカウントに対するユーザ名 / パスワードの変更 を参照してください。

**Oracle ユーザ名およびパスワードの作成に関する情報**

a. Oracle データベースが起動しているノード上で、dba グループの一部であるユーザで Linux にログインしてください。(「oracle」アカウントが最も一般的です。) sqlplus ユーティリティを使用して以下のコマンドを実行し、管理ユーザとしてデーターベースに接続してください。

```
$ sqlplus / as sysdba
```

b. この機能に対する新規ユーザを作成してください。

```
SQL> CREATE USER lkdba IDENTIFIED BY “password”;
```

c. このユーザに SYSDBA 権限を与えてください。

```
SQL> GRANT SYSDBA to lkdba;
```

d. Oracle が設定されると、LifeKeeper クラスタの各ノードが $ORACLE_HOME のローカルコピーを保有するので、クラスタの各ノードでこれらのコマンドを実行してください。LifeKeeper Oracle 階層の作成後、ノード上でデータベースを In Service にし、 CREATE および GRANT コマンド (上記) を実行してOracle でユーザをセットアップしてください。

注意：同じファイルシステムに 2 つのデータベースを設定しないでください。どうしてもそうしなければ

ならない場合は、十分な注意が必要です。その場合には、両方のデータベースを LifeKeeper の保護下に置き、両方の階層が同じプライマリサーバとバックアップサーバを持つようにする必要があります。

# 6.12.2.2. LifeKeeper 保護用の Oracle Net Listener の設定

Oracle データベースがリモートクライアント接続に対応する場合、Oracle データベースサーバに加えて、Oracle Listener を保護することができます。Oracle ネットワーク設定ユーティリティを使用して Oracle ネットワーク設定ファイル（例えば、listener.ora 、tnsnames.ora など）を作成する方法については、Oracle ドキュメンテーションを参照してください。

注記: 複数のリソースに対しての共有 Oracle Listener の作成方法については、このドキュメントの付録の「複数のリソースに対する共有 Oracle Listener の作成」を参照してください。

## Listener の設定

1. クライアントが接続するための切り替え可能 IP アドレスを選択する必要があります。このアドレスは DNS に登録すると良いでしょう。(IP のリソース階層を作成する方法の詳細については、**LifeKeeper IP Recovery Kit ドキュメンテーション** を参照してください。リソース依存関係を作成する方法の詳細については、「GUI Administration Tasks」の リソース依存関係の作成 トピックを参照してください。)

2. *listener.ora*ファイルで、データベースサービス名の HOST にこの切り替え可能な IP アドレスを指定してください。(*listener.ora* ファイルの詳細については、Oracle ドキュメンテーションを参照してください。)データベースサービス名の HOST には、切り替え可能な IP アドレスの代わりに DNS 名を使用できますが、LifeKeeper のベストプラクティスとしては推奨されません。切り替え可能な IP アドレスを使用することにより、DNS ルックアップの問題の影響によって quickCheck や restore または remove 処理中に LifeKeeper がリスナーの実行ステータスを決定できなくなることを防止できます。また、リスナーが 1 つしか定義されていない場合でも SID_LIST_LISTENER を定義する必要があります。

listener.ora のサンプルフォーマット:

```
.
.
.
SID_LIST_LISTENER =
     (SID_LIST =
     (SID_DESC =
             (SID_NAME = <SID Name>)
       )
)
.
.
.
<listener name>=
     (DESCRIPTION_LIST =
         (DESCRIPTION =
```

```
               (ADDRESS = (PROTOCOL = TCP)(HOST = <vip>)(PORT = <port number>))
        )
)
.
.
.
```

tnsnames.ora ファイルまたは Oracle Names の HOST に切り替え可能な IP アドレスを指定してください。(tnsnames.ora ファイルの詳細については、Oracle ドキュメンテーションを参照してください。)データベースサービス名の HOST には、切り替え可能な IP アドレスの代わりに DNS 名を使用できますが、LifeKeeper のベストプラクティスとしては推奨されません。切り替え可能な IP アドレスを使用することにより、DNS ルックアップの問題の影響によって quickCheck や restore または remove 処理中に LifeKeeper がリスナーの実行ステータスを決定できなくなることを防止できます。

```
    .
    .
    .
    <SID Name>=
       (DESCRIPTION =
          (ADDRESS_LIST =
             (ADDRESS = (PROTOCOL = TCP)(HOST = <vip>)(PORT = <port number>))
             )
             (CONNECT_DATA =
             (SID = <SID Name>)
          )
       )
```

これらのサンプルファイルは、Oracle 10g および 11g の両方で正常に動作します。

listener.ora

```
SID_LIST_LISTENER =
   (SID_LIST =
   (SID_DESC =
      (SID_NAME = ORA11A)
   )
)
LISTENER =
   (DESCRIPTION_LIST =
      (DESCRIPTION =
         (ADDRESS = (PROTOCOL = TCP)(HOST = 192.0.2.0)(PORT = 1521))
      )
   )
```

tnsnames.ora

```
ORA01 =
   (DESCRIPTION =
      (ADDRESS_LIST =
         (ADDRESS = (PROTOCOL = TCP)(HOST = 192.0.2.0)(PORT = 1521))
      )
         (CONNECT_DATA =
         (SID = ORA01)
      )
   )
```

通常の listener.ora の場所は *$ORACLE_HOME/network/admin* になります。最も一般的なポート番号は 1521 です。データベースのグローバル名は、作成時に定義されます。 *$ORACLE_HOME* ディレクトリが非共有ストレージ上にインストールされている場合は、listener.ora のコピーは両システム上に必要となるので注意してください。

注記: Oracle Net は、サービスのリスナーに障害が発生した場合に、クライアント接続を自動的に他のリスナーにフェイルオーバするオプションを提供しています。この機能を利用するには、*tnsnames.ora* ファイルで FAILOVER パラメータを「**ON**」に設定します。こうしておけば、LifeKeeper によって保護された Oracle SID のリスナーに障害が発生しても、LifeKeeper が保護下にあるリスナーを回復するまで、クライアント接続は別のリスナーを通じて継続されます。

## エラー

下記エラーが発生した場合は、以下に記載の解決策を確認してください。

### Oracle listener fails with TNS-00511: No Listener Linux Error:111: Connection refused

listener.ora ファイルを下記の例のように設定した場合、LifeKeeper はリスナーを起動することができません。

```
 (ADDRESS_LIST=
        (ADDRESS=(PROTOCOL=tcp)(HOST=<hostname>)(PORT=1521))
        (ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521))
        (ADDRESS=(PROTOCOL=tcp)(HOST=<vip>)(PORT=1521))
  )
```

<hostname> は、ホストのホスト名で、 <vip> は仮想 IP アドレスです。

エラーメッセージをスクロールしていくと下記エラーが表示されます。

TNS-12542: TNS:address already in use TNS-12560: TNS:protocol adapter error

### 解決策

listener.ora を下記のように変更してください。

```
(ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=tcp)(HOST=<ip>)(PORT=1521))
      (ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521))
      (ADDRESS=(PROTOCOL=tcp)(HOST=<vip>)(PORT=1521))
)
```

<ip> はホストの IP アドレスで、 <vip> は、仮想 IP アドレスです。

**注記:** セキュリティ上の問題がある場合は、 vipname が仮想 IP の DNS 名でも設定可能です。

```
(ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=tcp)(HOST=<vipname>)(PORT=1521))
)
```

これは tnsnames.ora ファイルの変更にも適用されます。

# 6.12.2.3. LifeKeeper での透過的アプリケーションフェイルオーバの設定

サーバがフェイルオーバしたり、Oracle データベースに障害が発生したりすると、ユーザは深刻な影響を受けてしまいます。通常は、データベースへの接続が切断されるだけでなく、作業中のデータもほとんど失われます。フェイルオーバが完了（または Oracle データベースが回復）すると、クライアントはアプリケーションを再起動してデータベースに再接続する必要があります。Oracle の透過的アプリケーションフェイルオーバ（TAF）機能を使用すると、障害によってはそれにマスクをかけて、その影響を減らしたりなくしたりすることができます。LifeKeeper 環境で TAF を設定する際には、LifeKeeper サーバ側で行う作業と Oracle クライアント側で行う作業があります。

この TAF 機能の効果的な利用をお考えの場合は、クライアントアプリケーションで Oracle Call Interface（OCI）から API を呼び出す際に、フェイルオーバを意識した呼び出しを行う必要があります。また、tnsnames.ora ファイルに Oracle Net のパラメータを指定して、適切な TAF の設定を行う必要があります。TAF モードは、tnsnames.ora の CONNECT_DATA セクションに FAILOVER_MODE パラメータを指定して設定します。TAF のメカニズムでは、フェイルオーバ時のクライアント接続の動作を制御するサブパラメータをいくつかサポートしています。LifeKeeper for Linux Oracle Recovery Kit では、TAF 設定時に以下のサブパラメータを指定できます。

**TYPE= (SELECT または SESSION)**

> クライアント接続を TAF でどのようにフェイルオーバするかを指定します。 **SELECT** を指定すると Oracle は、移行中に発行されたすべての select ステートメントを記録します。新しい接続が確立されるとすべての select ステートメントが再実行され、カーソルを元の位置に配置してクライアントが行のフェッチを継続できるようにします。SESSION を指定すると接続のみが新規に作成され、作業中のデータは失われます。

**METHOD= (BASIC)。**

> このメソッドを指定すると TAF は、一次接続が失敗したときのみ再接続を試みます。他に PRECONNECT メソッドがありますが、LifeKeeper では現在のところ **PRECONNECT** メソッドの使用はサポートしておりません。

**DELAY=（秒数）**

> 障害発生後、TAF が再接続を試みる間隔を秒数で指定します。クライアントアプリケーションや環境に合った値を指定する必要があります。

**RETRIES=（試行回数）**

> 障害発生後、TAF が再接続を試みる回数を指定します。 **DELAY** と **RETRIES** を組み合わせて使用する際には、サーバで障害が発生したときに Oracle が完全に回復できる、十分な時間を設定する必要があります。そうすることで TAF は、サーバのフェイルオーバの完了後、十分に時間的余裕を持って再起動できるようになります。

以下は、クライアントシステム用サンプル tnsnames.ora ファイルからの抜粋です。

```
LKproDB=
    (DESCRIPTION=
        (ADDRESS_LIST=
        (ADDRESS = (PROTOCOL=TCP) (HOST=<switchableIP>) (PORT=<port number>))
        )
        (CONNECT_DATA=
            (SID=LKroDB)
            (SERVER=DEDICATED)
            (FAILOVER_MODE=
                (TYPE=SELECT)
                (METHOD=BASIC)
                (DELAY=5)
                (RETRIES=30)
            )
        )
    )
```

通常の tnsnames.oraの場所は *$ORACLE_HOME/network/admin* になります。最も一般的なポート番号は 1521 です。 tnsnames.oraファイルはユーザのホームディレクトリにも同様に配置されます。 *$ORACLE_HOME* ディレクトリが非共有ストレージ上にインストールされている場合は、 listener.ora および tnsnames.ora のコピーが両システム上に必要になるので注意してください。

Oracle データベースを保護している LifeKeeper サーバでは、LifeKeeper によって保護されている切り替え可能な IP アドレスを使用してリスナーを設定する必要があります。Oracle Net とリスナーの設定の詳細については、前述の「LifeKeeper 保護用の Oracle Net Listener の設定」セクションを参照してください。

# 6.12.2.4. Oracle Multitenant 構成による Pluggable Database の設定

Oracle データベースが Oracle Multitenant アーキテクチャに対応し、Oracle データベースがコンテナ・データベース（以下、CDB と呼ぶ）を保護する場合、Oracle データベースサーバに加えて、プラガブル・データベース（以下、PDB と呼ぶ）を保護することができます。

## CDB と PDB の確認

1. PDB を保護するためには Oracle リソースが作成されている必要があります。また、保護している Oracle リソースが CDB である必要があります。CDB であるかの確認はデータベースへ接続後に以下のコマンドを使って確認することができます。

   ```
   SQL> select CDB from V$DATABASE;
   ```

2. PDB を保護するにあたって、CDB の中に PDB がマウントされている必要があります。PDB がマウントされているかの確認はデータベース接続後に以下のコマンドで確認することができます。

   ```
   SQL> show pdbs;
   ```

## Oracle PDB リソース階層の作成

1. LifeKeeper GUI メニューから **[Edit]** を選択し、次に **[Server]** を選択してください。ドロップダウンメニューから、 **[Create Resource Hierarchy]** を選択してください。

   **重要:** リソースを作成するときは、Oracle リソースが作成されており、In Serviceである必要があります。

   ダイアログボックスが表示され、クラスタ内にインストール済みの認識されているすべての Recovery Kit がドロップダウンリストボックスに表示されます。ドロップダウンリストから **[Oracle Pluggable Database]** を選択してください。 **[Next]** をクリックして次のダイアログボックスに進んでください。

   **注記:** ダイアログボックスで **[Back]** ボタンが有効な場合は、前のダイアログボックスに戻ることができます。これは、エラーが発生して、以前に入力した情報を訂正する必要がある場合に特に役立ちます。

   階層作成手順のどの段階でも、 **[Cancel]** をクリックすると、作成プロセス全体がキャンセルされます。

2. 以下の情報を入力するように要求されます。ダイアログボックスで **[Back]** ボタンが有効な場合は、前のダイアログボックスに戻ることができます。これは、エラーが発生して、前に入力した情報を修正する必要がある場合に便利な機能です。いつでも **[Cancel]** をクリックして、作成処理全体を取り消すことができます。

| フィールド | ヒント |
|---|---|
| **Server** | Oracle PDB リソースを作成する LifeKeeper サーバを選択します。 |
| **Switchback Type** | intelligent または automatic を選択します。フェイルオーバ後、バックアップサーバで Oracle PDB リソースが In Service (アクティブ) になるとき、プライマリサーバにスイッチバックされる方法が決まります。インテリジェントスイッチバック (intelligent) の場合、リソースをプライマリサーバにスイッチバックするときに管理者の介入が必要ですが、自動スイッチバック (automatic) の場合は、プライマリサーバがオンラインに戻り、LifeKeeper のコミュニケーションパスが再確立されるとすぐにスイッチバックが実行されます。<br><br>**注記:** スイッチバック方法は、Oracle PDB リソースが使用する依存リソースのスイッチバック方法と一致する必要があります。 |
| **ORACLE_SID** | 保護しているOracleデータベースのSIDを指定します。 |
| **Oracle PDBs** | 保護したいPDBを指定します。このフィールドでは複数選択が可能です。 |
| **PDB Tag** | プライマリサーバ上の新しい Oracle PDB リソースに対する固有のタグ名です。デフォルトのタグ名は「pdb-<ORACLE_SID>」となっています.。別の固有のタグ名を入力することもできます。タグ名に対する有効な文字は、アルファベット、数字および以下の特別な記号になります。- _ ./ |

3. **[Next]** をクリックしてください。 **[Create Resource Wizard]** が表示され、Oracle PDB リソース階層が作成されます。LifeKeeper によって入力データが検査されます。問題が検出された場合、情報ボックスにエラーメッセージが表示されます。

4. Oracle PDB リソース階層が正常に作成されたこと、フェイルオーバによる保護を実現するにはその階層をクラスタ内の別のサーバに拡張する必要があることを示すメッセージが表示されます。**[Next]** をクリックします。

5. **[Continue]** をクリックします。 **[Pre-extend Wizard]** が起動されます。リソース階層を別のサーバに拡張する方法の詳細については、「Oracle PDB リソース階層の拡張」の **手順 2** を参照してください。

# Oracle PDB リソース階層の拡張

この操作は **[Edit]** メニューから開始できます。または、 **[Create Resource Hierarchy]** オプションが完了すると自動的に開始されます。その場合は下の手順 2 を参照してください。

1. **[Edit]** メニューの **[Resource]** から **[Extend Resource Hierarchy]** を選択します。 **[Pre-Extend Wizard]** が表示されます。拡張操作に慣れていない場合は、 **[Next]** をクリックします。LifeKeeper のリソース階層の拡張作業のデフォルト値が分かっていて、入力と確認を省略する場合は、**[Accept Defaults]** をクリックします。

2. **[Pre-Extend Wizard]** で以下の情報を入力します。

   **注記:** 最初の 2 つのフィールドは **[Edit]** メニューの **[Extend]** から操作を開始した場合にだけ表示さ

れます。

| フィールド | ヒント |
|---|---|
| **Template Server** | Oracle PDB リソースが現在 In Service のサーバを選択します。 |
| **Tag to Extend** | 拡張する Oracle PDB リソースを選択します。 |
| **Target Server** | 拡張先のサーバを入力または選択します。 |
| **Switchback Type** | フェイルオーバ後、バックアップサーバで Oracle PDB リソースが In Service (アクティブ) になるとき、プライマリサーバにスイッチバックされる方法が決まります。intelligent または automatic を選択できます。スイッチバックタイプは、必要な場合 **[Resource Properties]** ダイアログボックスの **[General]** タブで後から変更できます。<br><br>**注記:** スイッチバック方法は、Oracle PDB リソースが使用する依存リソースのスイッチバック方法と一致する必要があることに注意してください。 |
| **Template Priority** | テンプレートの優先順位 を選択または入力します。これはサーバで現在 In Service の Oracle PDB 階層の優先順位です。優先順位は、1 ~ 999 の範囲で未使用の値が有効で、小さい数字ほど優先順位が高くなります (数字 1 が最高の優先順位に相当)。拡張処理時に、別のシステムですでに使用中の優先順位をこの階層に対して指定することはできません。デフォルト値を推奨します。<br><br>**注記:** このフィールドは階層を最初に拡張するときだけ表示されます。 |
| **Target Priority** | これは、新しく拡張する Oracle PDB 階層が、他のサーバに存在する同等の階層に対して持つ相対的な優先順位です。1 ~ 999 の範囲で、まだ優先順位として使用されていない値が有効で、リソースのカスケーディングフェイルオーバシーケンスにおけるサーバの優先順位を示します。LifeKeeper のデフォルトでは、階層が作成されたサーバに「1」が割り当てられることに注意してください。優先順位は連続している必要はありませんが、特定のリソースについて 2 つのサーバに同じ優先順位を割り当てることはできません。 |

3. 拡張前のチェックが正常に終了したというメッセージが表示されたら、 **[Next]** をクリックします。

4. 拡張する階層に応じて、拡張されるリソースタグ (一部編集不可) を示す一連の情報ボックスが表示されます。

5. **[Extend Wizard]** でTag名が正しいかを確認し **[Extend]** をクリックします。

6. 「Hierarchy extend operations completed (階層の拡張操作が完了しました)」というメッセージが表示されたら、階層を別のサーバに拡張する場合は、 **[Next Server]** をクリックし、拡張作業が必要ない場合は、 **[Finish]** をクリックします。

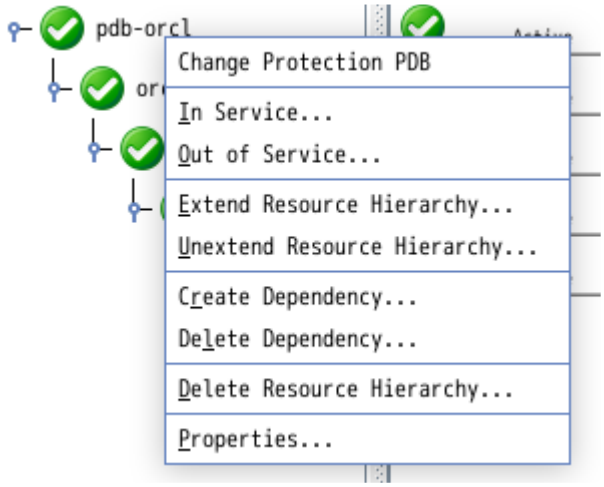
7. 「Hierarchy Verification Finished (階層の検査が完了しました)」というメッセージが表示されたら、

**[Done]** をクリックします。

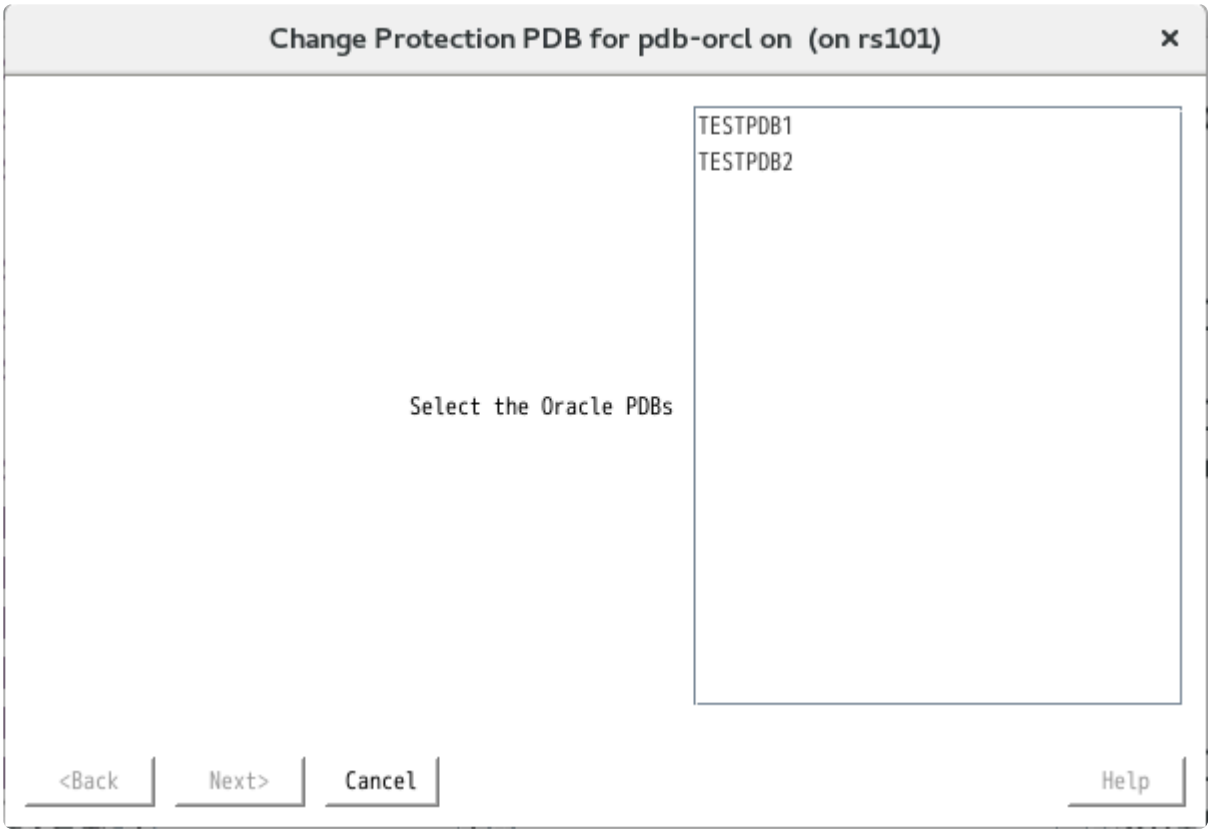
## 保護する PDB の変更

階層作成後、以下の手順で保護する PDB を変更してください。

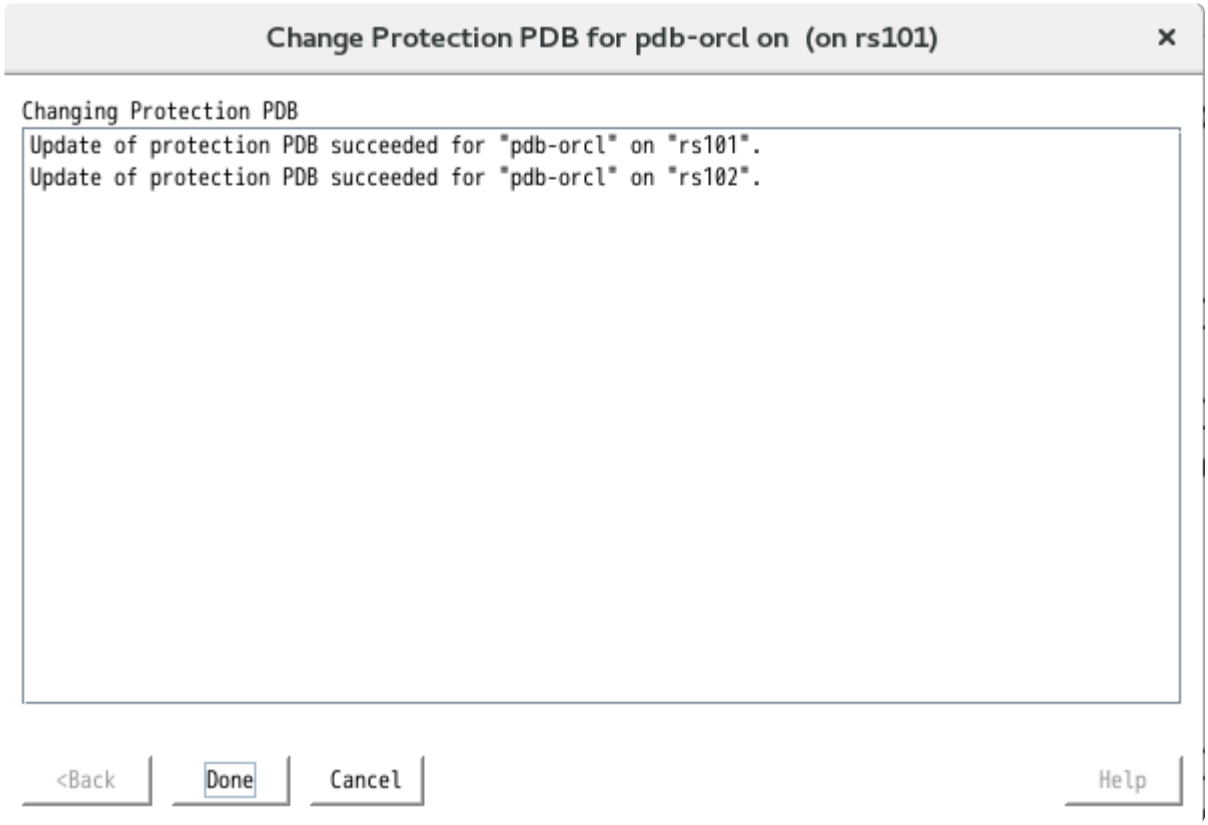
1. LifeKeeper GUI から Oracle PDB リソース階層を右クリックし、 **[Change Protection PDB]** を選択してください。




2. 保護したい PDB を選択してください。（複数選択可能です）




3. **[Next]** をクリックすると設定が変更されます。




4. **[Done]** をクリックして設定変更を完了してください。

# 6.12.2.5. Oracle の構成例

以下の図は、LifeKeeper LifeKeeper 環境における Oracle のアクティブ / スタンバイおよびアクティブ / アクティブの設定例を示しています。

本セクションの例を見れば、Oracle データベースインスタンスをローカルディスクおよび共有ディスク上に設定する方法がわかります。それぞれの表は、設定タイプと Oracle パラメータの関係を示しています。これらの設定は、本書で説明している、Oracle の設定と LifeKeeper ソフトウェアの互換性を保証する設定ルールおよび要件にも適合しています。

本セクションでは、まず設定の要件を説明し、次にそれぞれの設定例を示します。

- アクティブ / スタンバイ
- アクティブ / アクティブ

本セクションの例は、設定のサンプルにすぎませんが、これらの設定を理解し、設定ルールに従うことにより、利用する環境に適した効果的なソリューションの定義およびセットアップが容易にできるようになります。

# 6.12.2.5.1. Oracle 設定の要件

それぞれの例では、1 つまたは 2 つのデータベース （ **databaseA** 、 **databaseB** ）が使用されています。デフォルトでは、Oracle データベースのシステム識別子（SID）と一致するタグ名が LifeKeeper から示されます。ただし、以下の例では、databaseA-on-server1 のように SID とサーバ名で構成されたタグ名が使用されています。

設定例を理解するには、次のような設定要件を留意しておいてください。

- **LifeKeeper 階層** ：LifeKeeper の管理を実行するときに、プライマリサーバは Oracle インスタンスが現在実行されている場所を参照します。LifeKeeper 階層を作成するときのシステム管理はこのサーバで行われます。この設定例では、プライマリサーバは Server 1、バックアップサーバまたは代替サーバは Server 2 です。

- **1 台のサーバのみによってロックされる共有ディスク** 。共有ストレージリソースが LifeKeeper の保護下にあると、そのリソースに一度にアクセス可能なサーバは 1 つだけになります。共有デバイスがディスクアレイである場合、LUN 全体が保護されます。共有デバイスがディスクである場合、ディスク全体が保護されます。共有デバイスがディスクの場合は、ディスク全体がリザーブされます。サーバに障害が発生すると、最も優先順位の高いバックアップサーバが自身の保護を確立し、他のサーバをすべてロックします。

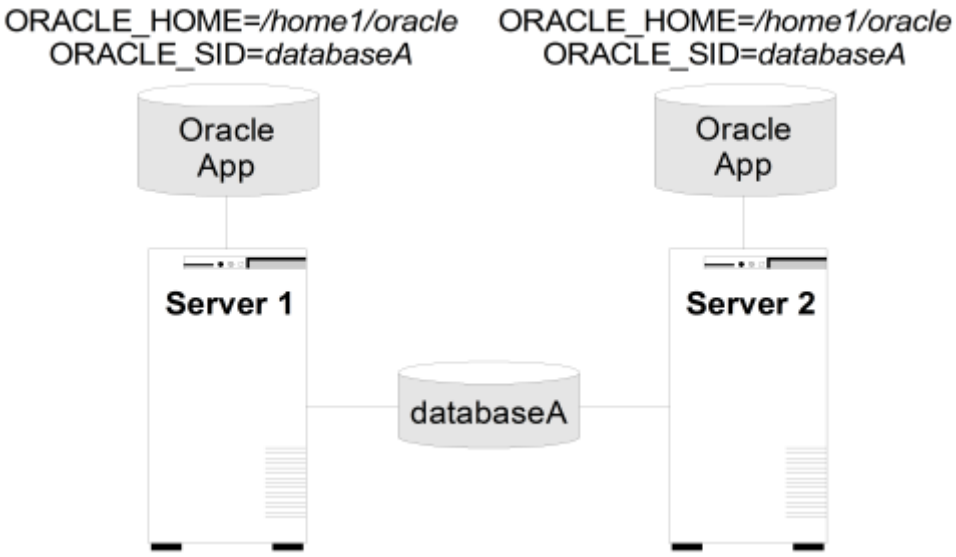
- **共有ディスク上のデータベース** ：LifeKeeper Oracle Recovery Kit が正しく機能するには、データベースが常に共有デバイス上にあることが必要です。データベースは 1 つまたは複数のファイルシステムおよびディスク上に存在することができます。

**注記:** それぞれの設定図に付随する表は、プライマリサーバ（Server 1）の場合は [Create Resource Hierarchy] ウィザードに、バックアップサーバ（Server 2）の場合は [Extend Resource Hierarchy] ウィザードに入力される適切な情報の例を示しています。ウィザードに入力する情報の詳細については、LifeKeeper の設定作業 を参照してください。これらの表は、Recovery Kit を設定するときに役立つ参照情報となります。

# 6.12.2.5.2. アクティブ / スタンバイ設定

このセクションでは、2 つのアクティブ / スタンバイ設定の例（図 1 と図 2）を紹介します。これらの構成では、Server 1 がデータベースへの排他アクセスを持っているため、アクティブと見なされます。Server 2 は別の処理を実行します。Server 1 に障害が発生すると、Server 2 がデータベースにアクセス可能になり、LifeKeeper がデータベースの稼動を再確立します。

**図 1アクティブ / スタンバイ設定、例 1**




**設定に関する注記:**

各サーバは独自の $ORACLE_HOME ディレクトリを非共有ディスク上に置いています。各サーバには同じバージョンの Oracle アプリケーションが導入されています。

$ORACLE_HOME パスはどちらのサーバでも同じです。

データベース databaseA は共有ディスク上にあります。

Server1 でのリソース階層の作成:

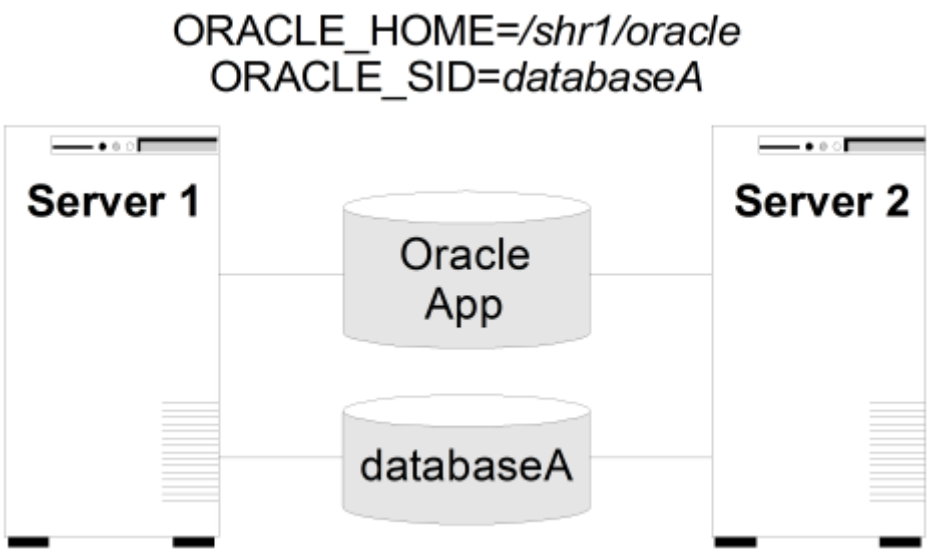
| | |
|---|---|
| サーバ: | Server1 |
| データベースの ORACLE_SID | databaseA |
| データベースのユーザ名 | system |
| ユーザ名のパスワード | ********** |
| データベースの ORACLE_HOME | /home1/oracle |
| データベースタグ: | databaseA-on-server1 |

Server 2 へのリソース階層の拡張:

| | |
|---|---|
| テンプレートサーバ: | Server1 |

| | |
|---|---|
| 拡張するタグ: | databaseA-on-server1 |
| ターゲットサーバ: | Server2 |
| ターゲットの優先順位: | 10 |
| データベースタグ: | databaseA-on-server1 |

図 2アクティブ / スタンバイ設定、例 2




**設定に関する注記:**

1.
2. どちらのサーバも共有ディスク上の $ORACLE_HOME ディレクトリを使用します。
3. $ORACLE_HOME パスはどちらのサーバでも同じです。
4. データベース databaseA は共有ディスク上にあります。
5. Server 1 がアクティブである間、Server 2 は共有ディスクのファイルおよびディレクトリにアクセスできません。
6. $ORACLE_HOME はデータベースと同じ共有ディスク上または別のディスク上に置くことができます。

Server1 でのリソース階層の作成:

| | |
|---|---|
| サーバ: | Server1 |
| データベースの ORACLE_SID | databaseA |
| データベースのユーザ名 | system |
| ユーザ名のパスワード | ********** |
| データベースの ORACLE_HOME | /shr1/oracle |

| | |
|---|---|
| データベースタグ: | databaseA-on-server1 |

Server 2 へのリソース階層の拡張:

| | |
|---|---|
| テンプレートサーバ: | Server1 |
| 拡張するタグ: | databaseA-on-server1 |
| ターゲットサーバ: | Server2 |
| ターゲットの優先順位: | 10 |
| データベースタグ: | databaseA-on-server1 |

# 6.12.2.5.3. アクティブ / アクティブ設定

アクティブ / アクティブ設定は、それぞれが別々のデータベースインスタンスを実行する 2 台以上のサーバで構成されます。各データベースは、別々の共有ディスク上にある必要があります。

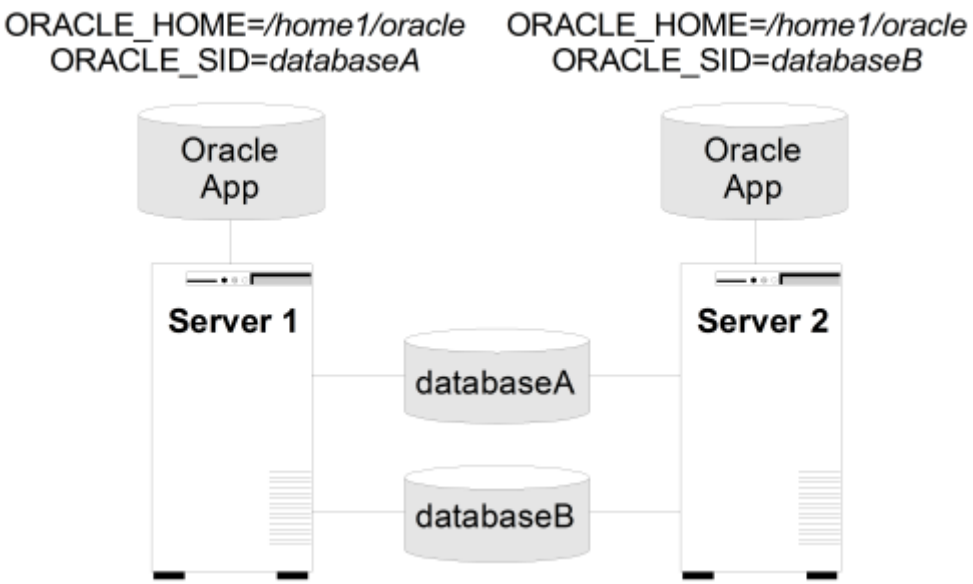
$ORACLE_HOME は、設定の要件に応じて、非共有ディスクまたは共有ディスク上に置くことができます。例えば、共通の$ORACLE_HOME を使用する任意のサーバ上の複数のデータベースインスタンスは、$ORACLE_HOME が非共有ディスク上にある必要があります。$ORACLE_HOME ディレクトリが共有ディスク上にある場合は、別々の共有ディスクでなければなりません。

本セクションでは、図 3 と図 4 に示されている 2 種類のアクティブ / アクティブ設定の例を検討します。

- 共有リソース上のデータベースと、非共有リソース上の一般的な $ORACLE_HOME。
- 共有リソース上のデータベースと、同じ共有リソース上の適切な $ORACLE_HOME。

**注記:** 非共有ストレージ上で複数の $ORACLE_HOME を使用しているサーバの複数のデータベースインスタンスは除外されます。

図 3アクティブ / アクティブ設定の例




**設定に関する注記:**

1. サーバは独自の $ORACLE_HOME ディレクトリを非共有ディスク上に配置します。各サーバには同じバージョンの Oracle アプリケーションが導入されています。
2. $ORACLE_HOME パスはどちらのサーバでも同じです。
3. 2 つのデータベース、databaseA と databaseB は共有ディスク上にあります。

4. 両方の Oracle インスタンスのエントリが含まれている oratab ファイルは、両サーバの /etc/ にあります。

5. 最初は、Server 1 で databaseA が、Server 2 で databaseB が稼動します。スイッチオーバが行われると、1 つのシステムが両方のデータベースを実行できます。

6. 詳細については、ローカルストレージでの Oracle バイナリのインストール後の Oracle データベース階層の作成 を参照してください。

Server 1 での 1 つ目のリソース階層の作成:

| | |
|---|---|
| サーバ: | Server1 |
| データベースの ORACLE_SID: | databaseA |
| データベースのユーザ名 | system |
| ユーザ名のパスワード | ********** |
| データベースの ORACLE_HOME | /home1/oracle |
| データベースタグ: | databaseA-on-server1 |

1 つ目のリソース階層の Server 2 への拡張:

| | |
|---|---|
| テンプレートサーバ: | Server1 |
| 拡張するタグ: | databaseA-on-server1 |
| ターゲットサーバ: | Server2 |
| ターゲットの優先順位: | 10 |
| データベースタグ: | databaseA-on-server1 |

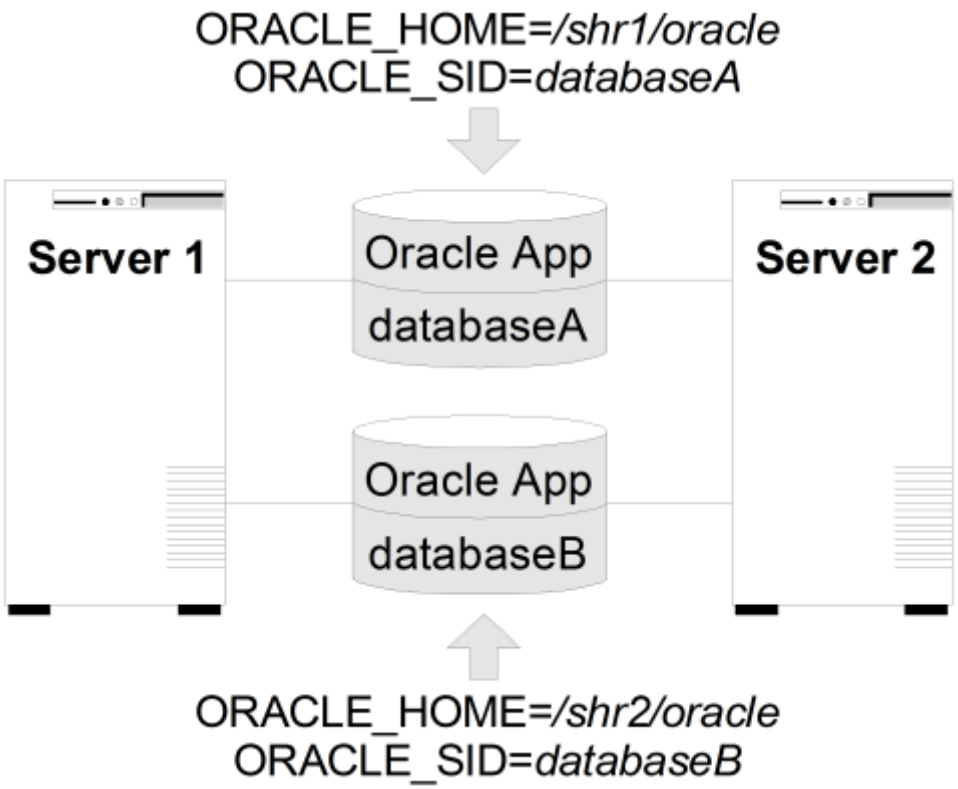
Server 2 での 2 番目のリソース階層の作成:

| | |
|---|---|
| サーバ: | Server2 |
| データベースの ORACLE_SID: | databaseB |
| データベースのユーザ名 | system |
| ユーザ名のパスワード | ********** |
| データベースの ORACLE_HOME | /home1/oracle |
| データベースタグ: | databaseB-on-server2 |

2 つ目のリソース階層の Server 1 への拡張:

| | |
|---|---|
| テンプレートサーバ: | Server2 |
| 拡張するタグ: | databaseB-on-server2 |
| ターゲットサーバ: | Server1 |
| ターゲットの優先順位: | 10 |
| データベースタグ: | databaseB-on-server2 |

図 4アクティブ / アクティブ設定、例 2




**設定に関する注記:**

1. 両サーバが別々の共有ディスク上の $ORACLE_HOME ディレクトリを使用します。
2. Oracle アプリケーションはどちらのサーバでも同じです。$ORACLE_HOME は、サーバに定義されたインスタンスごとに異なります。
3. 2 つのデータベース、databaseA と databaseB は共有ディスク上にあります。
4. 両方の Oracle インスタンスのエントリが含まれている oratab ファイルは /etc/ にあります。

5. Oracle インスタンスごとに一意のログインが必要です。それぞれのログインに対する Id および gid は Server 1 および Server 2 で同一である必要があります。

6. 最初は、Server 1 で databaseA が、Server 2 で databaseB が稼動します。スイッチオーバが行われると、1 つのシステムが両方のデータベースを実行できます。

Server 1 での 1 つ目のリソース階層の作成:

| | |
|---|---|
| サーバ: | Server1 |
| データベースの ORACLE_SID: | databaseA |
| データベースのユーザ名 | system |
| ユーザ名のパスワード | ********** |
| データベースの ORACLE_HOME | /shr1/oracle |
| データベースタグ: | databaseA-on-server1 |

1 つ目のリソース階層の Server 2 への拡張:

| | |
|---|---|
| テンプレートサーバ: | Server1 |
| 拡張するタグ: | databaseA-on-server1 |
| ターゲットサーバ: | Server2 |
| ターゲットの優先順位: | 10 |
| データベースタグ: | databaseA-on-server1 |

Server 2 での 2 番目のリソース階層の作成:

| | |
|---|---|
| サーバ: | Server2 |
| データベースの ORACLE_SID: | databaseB |
| データベースのユーザ名 | system |
| ユーザ名のパスワード | ********** |
| データベースの ORACLE_HOME | /shr2/oracle |
| データベースタグ: | databaseB-on-server2 |

2 つ目のリソース階層の Server 1 への拡張:

| | |
|---|---|
| テンプレートサーバ: | Server2 |
| 拡張するタグ: | databaseB-on-server2 |
| ターゲットサーバ: | Server1 |
| ターゲットの優先順位: | 10 |
| データベースタグ: | databaseB-on-server2 |

# 6.12.3. Oracle Recovery Kitの設定タスク

設定作業は LifeKeeper GUI を使用して実行できます。以下の 4 つの作業は、Oracle リソースインスタンス特有のものであり、Recovery Kit ごとに異なるため本書で説明しています。

- リソース階層の作成 アプリケーションリソース階層を LifeKeeper クラスターに作成します。レプリケーション（SIOS DataKeeper）ファイルシステムリソースは、Oracleリソースを作成する前に作成する必要があります。
- リソース階層の削除 リソース階層を LifeKeeper クラスター内のすべてのサーバーから削除します。
- リソース階層の拡張 リソース階層をプライマリサーバーからバックアップサーバーへ拡張します。
- リソース階層の拡張解除 リソース階層を LifeKeeper クラスター内の 1 つのサーバーから拡張解除（削除）します。
- Oracle 設定の参照 [Resource Properties] ダイアログを参照できます。
- ユーザ名 / パスワードの変更 Oracle データベースを保護するためにログインするユーザー名およびパスワードの変更します。
- リソース階層のテスト 手動スイッチオーバーを実施することで、Oracle リソース階層をテストできます。

以下のタスクはすべての Recovery Kit で手順が同じであるためLifeKeeper for Linux テクニカルドキュメンテーション の管理 セクション で説明します。

- リソース依存関係の作成 既存のリソース階層と別のリソースインスタンスとの間に親子の依存関係を作成し、クラスター内のすべての対象サーバーに依存関係の変更を反映します。
- リソース依存関係の削除 .リソースの依存関係を削除して、クラスター内のすべての対象サーバーに依存関係の変更を反映します。
- In Service リソース階層を特定のサーバーで In Service にします。
- Out of Service リソース階層を特定のサーバで Out of Service にします。
- 表示 / 編集 特定のサーバでリソース階層のプロパティを表示または編集します。

**注記:** 本セクションの残りの部分では、LifeKeeper GUI のメニューから特定の作業を選択して Recovery Kit を設定する方法について説明します 。設定作業はツールバーから選択することもできます。状況表示ウィンドウの [Resource Hierarchy Tree]（左側のペイン） のグローバルリソースを右クリックして、[Edit] メニューと同じドロップダウンメニューの選択項目を表示することもできます。言うまでもなく、これは階層がすでに存在している場合にだけ可能な方法です。

状況表示ウィンドウの [Resource Hierarchy Table]（右側のペイン） のリソースインスタンスを右クリックして、サーバおよび特定リソースの状況に応じて、 *リソース階層の作成* を除くすべての設定作業を実行

することもできます。

# 6.12.3.1. リソース階層の作成

**注記:** Oracle Net リモートクライアントアクセスを利用するには、クライアントの接続に使用する IP アドレスが、Oracle リソース階層に依存関係を持っている LifeKeeper の保護下にある必要があります。（詳細については、Lifekeeper 保護用の Oracle Net Listener の設定 を参照してください。）

プライマリサーバからリソースインスタンスを作成するには、以下の手順を実行します。

1. LifeKeeper GUI メニューから **[Edit]** を選択し、次に **[Server]** を選択してください。ドロップダウンメニューから、 **[Create Resource Hierarchy]** を選択してください。

   **重要:** リソースを作成するときは、Oracle アプリケーションが実行されている必要があります。

   ダイアログボックスが表示され、クラスタ内にインストール済みの認識されているすべての Recovery Kit がドロップダウンリストボックスに表示されます。ドロップダウンリストから **[Oracle Database]** を選択してください。 **[Next]** をクリックして次のダイアログボックスに進んでください。

   **注記:** ダイアログボックスで **[Back]** ボタンが有効な場合は、前のダイアログボックスに戻ることができます。これは、エラーが発生して、以前に入力した情報を訂正する必要がある場合に特に役立ちます。

   階層作成手順のどの段階でも、 **[Cancel]** をクリックすると、作成プロセス全体がキャンセルされます。

2. **[Switchback Type]** を選択してください。この選択によって、バックアップサーバへのフェイルオーバの後、Oracle インスタンスが In Service に戻ったときに、このサーバにどのようにスイッチバックされるのかが決まります。intelligent または automatic を選択できます。Intelligent switchback では、インスタンスをプライマリ／オリジナルサーバにスイッチバックするときに、管理者の介入が必要になります。自動スイッチバックの場合は、プライマリサーバがオンラインに戻り、LifeKeeper コミュニケーションパスを再確立した直後にスイッチバックが行われます。

   スイッチバックタイプは、必要な場合 **[Resource Properties]** ダイアログボックスの **[General]** タブで後から変更できます。 **[Next]** をクリックして次のダイアログボックスに進んでください。

3. Oracle データベースを配置したい **サーバ** を選択してください（通常これは、プライマリサーバまたはテンプレートサーバと呼ばれます）。クラスタ内のすべてのサーバがドロップダウンリストに表示されます。 **[Next]** をクリックして次のダイアログボックスに進んでください。

4. このOracleデータベースの **ORACLE_SID** を選択してください。これは、設定するデータベースの Oracle システム識別子を指定するタグ名です。 */etc/oratab* にこのデータベースのエントリが存在しなければなりません。 **[Next]** をクリックして次のダイアログボックスに進んでください。

5. **ORACLE_SID** の **[Username]** を入力してください。これは ORACLE_SID にログインする間に指定する Oracle データベースのユーザ名になります。このユーザ名は、フルコントロールを与えるためにデータベースに対して sysdba 権限で接続可能である必要があります。 **[Next]** をクリックして次のダイアログボックスに進んでください。（このフィールドは空欄のままでも問題ありません。

空欄にした場合は、LifeKeeper は Oracle データベースユーザリソースを制御するためにユーザ名およびパスワードを使用しません。次のステップの **[Input Password]** がスキップされます。)

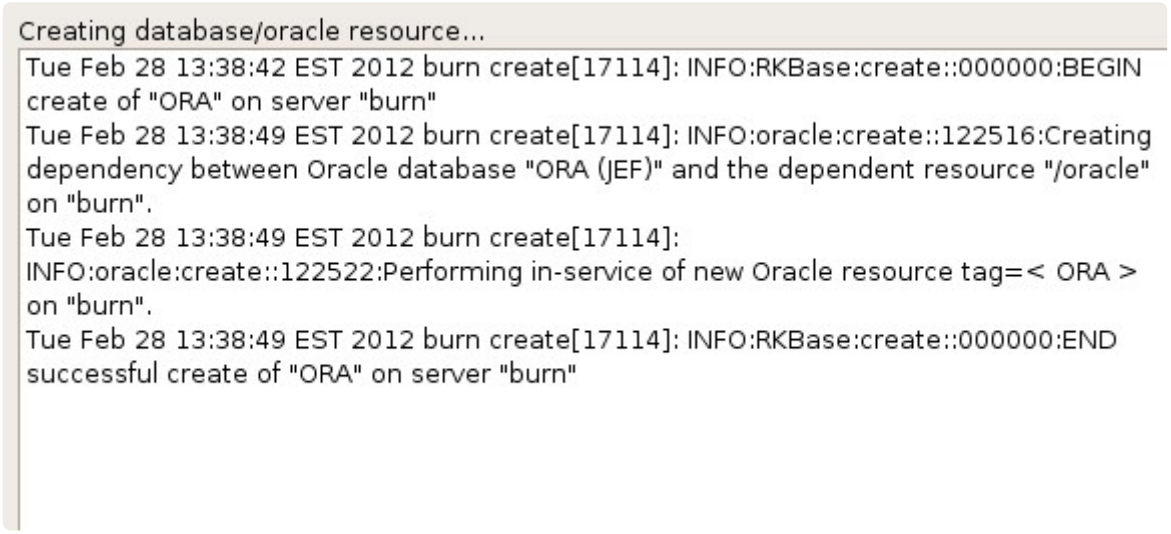
6. **[Password]** を入力してください。このパスワードは ORACLE_SID にログインする際に指定するパスワードです。このパスワードは、LifeKeeper により暗号化されて保存されます。 **[Next]** をクリックして次のダイアログボックスに進んでください。

7. Oracle リソースに依存して含まれる Listener の タグ名 を選択してください。リストが、現在サーバ上で保護されているすべての Listener リソースを表示します。Oracle SID に対して要求される Listener に一致した Listener リソースタグを選択してください。Listener リソースが存在しない場合は、 **[None]** を選択してください。

8. **[Database Tag]** を選択または入力します。これは、LifeKeeper が Oracle 階層に与えるタグ名です。デフォルトを選択するか、独自のタグ名を入力することができます。

   **[Create]** をクリックすると、 **[Create Resource]** ウィザードによって Oracle リソースが作成されます。

9. この時点で情報ボックスが表示され、LifeKeeper は、Oracle リソース階層を作成するのに有効なデータが提供されているか検査します。LifeKeeper が問題を検知した場合は、インフォメーションボックスにエラーが表示されます。認証に成功すると、リソースが作成されます。




10. **[Next]** をクリックして **[Pre-extend]** ダイアログボックス （このドキュメントで後述）に進んでください。リソース階層を LifeKeeper で保護するには、クラスタ内の別のサーバにリソース階層を拡張する必要があります。

# 6.12.3.2. リソース階層の削除

リソース階層を SPS 環境から削除するには、次の手順を実行します。

1.
2. [LifeKeeper GUI] メニューから **[Edit]** を選択し、次に **[Resouce]** を選択してください。ドロップダウンメニューから、 **[Delete Resource Hierarchy]** を選択してください。

3. Oracle リソース階層を削除する **[Target Server]** の名前を選択してください。

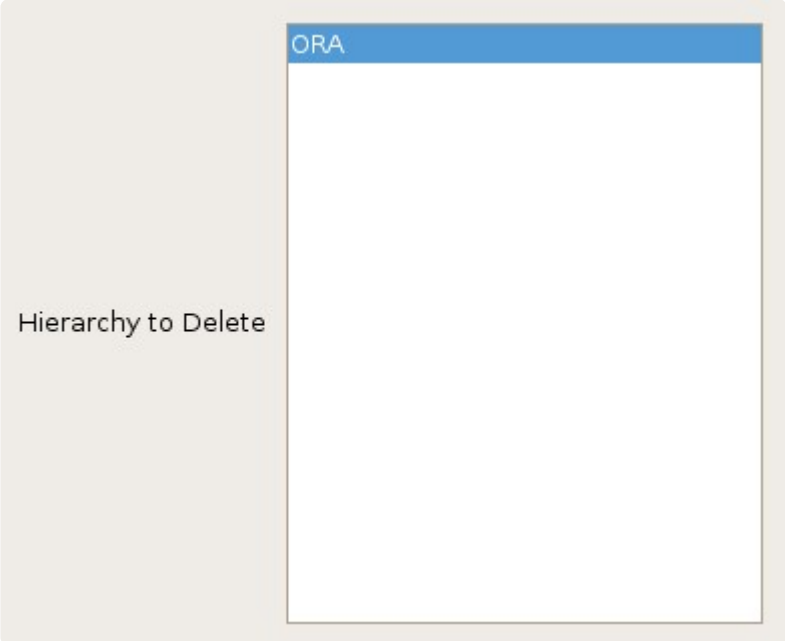
   **注記:** 右側ペインで個々のリソースインスタンスを右クリックした場合、またはリソースが 1 台のサーバ上のみにあり、左側ペインでグローバルリソースを右クリックし、[Delete Resource] 作業を選択した場合、このダイアログボックスは表示されません。

   **[Next]** をクリックして次のダイアログボックスに進んでください。

4. [Hierarchy to Delete] を選択してください。削除するリソース階層を指定して、強調表示にしてください。

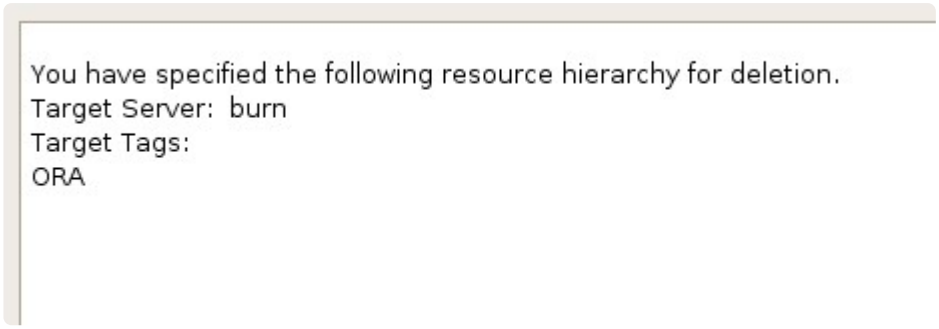
   **注記:** 左側のペインからグローバルリソースを右クリックするか、右側のペインから個々のリソースインスタンスを右クリックして [Delete Resource] 作業を選択した場合、このダイアログボックスは表示されません。

   **[Next]** をクリックして次のダイアログボックスに進んでください。

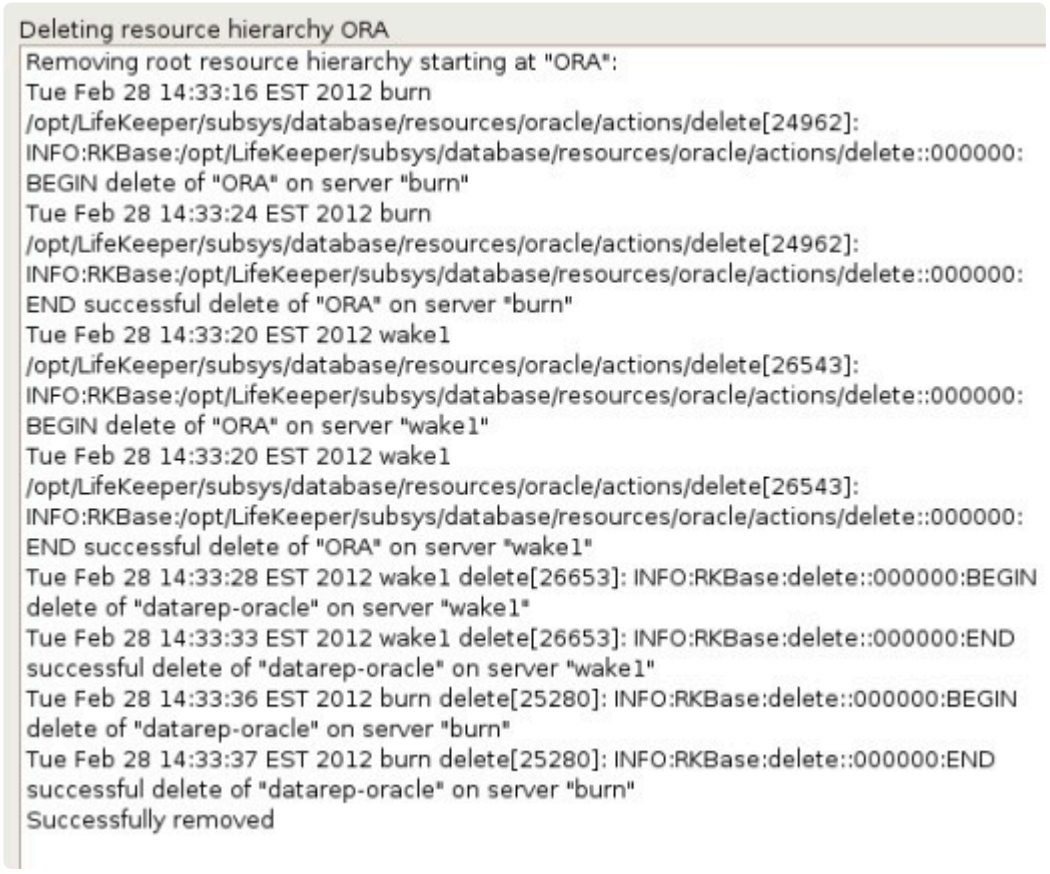
5. 選択したターゲットサーバと、削除の対象として選択した階層を確認する情報ボックスが表示され

ます。




**[Delete]** をクリックしてリソースを削除し、最後のダイアログボックスに進んでください。

6. Oracle リソースが正常に削除されたことを示す別の情報ボックスが表示されます。




7. **[Done]** をクリックして、 **[Delete Resource Hierarchy]** メニューを終了します。

**注記:** 複数のリソースに対する共有 Oracle Listener の作成方法については、付録の「複数のリソースに対する共有 Oracle Listener の作成」を参照してください。

# 6.12.3.3. リソース階層の拡張

階層の作成後、クラスタ内の別のサーバに拡張する必要があります。3 通りのシナリオで、テンプレートサーバからターゲットサーバにリソースインスタンスを拡張できます。最初のシナリオは、リソース作成後、表示された情報ダイアログボックスで **[Next]** をクリックすることで、「**続けて**」、そのリソースを別のサーバに拡張する場合です。2 番目のシナリオは、以下に示すように [Edit] メニューから **[Extend Resource Hierarchy]** 作業を入力する場合です。3 番目のシナリオは、左側または右側のペインから拡張されていない階層を右クリックする場合です。どのシナリオでも同じダイアログボックスが表示されます（いくつかの例外については、以下に詳細を明記します）。

1.
2. LifeKeeper GUI メニューから **Extend ウィザード** を起動している場合は、 **[Edit]** を選択し、次に **[Resource]** を選択してください。ドロップダウンメニューから **[Extend Resource Hierarchy]** を選択してください。これで ***Pre-Extend Resource Hierarchy ウィザード*** が起動されます。

3. 最初のダイアログボックスが表示され、現在 Oracle リソース階層がサービス中の **テンプレートサーバ** を選択するように指示されます。ここで選択する **テンプレートサーバ** と次のダイアログボックスで選択する **[Tag to Extend]** によって、サービス中のリソース階層が表されることを認識しておくことが重要です。選択したテンプレートサーバでサービス中でないリソースタグを選択すると、エラーメッセージが表示されます。このダイアログのドロップダウンボックスには、クラスタ内の全サーバの名前が示されています。

   **注記** : Oracle リソース階層の作成に続いて **[Pre-Extend Resource Hierarchy]** 作業に入った場合、ウィザードが作成段階にあるテンプレートサーバをすでに確認しているため、このダイアログボックスは表示されません。GUI ウィンドウの左側ペインの Oracle リソースアイコンまたは右側ペインの Oracle リソースボックスを右クリックして、 **[Extend Resource Hierarchy]** を選択した場合も同様です。

   階層を拡張する手順の間に **[Cancel]** をクリックすると、どの時点であってもそのサーバへの拡張プロセスがキャンセルされるので注意してください。ただし、すでにリソースを別のサーバに拡張している場合は、明示的に拡張解除するまで、そのインスタンスの拡張は有効です。

   例えば、Server 1 にリソースを作成し、そのリソースを Server 2 へ拡張したとします。同じリソースを Server 3 に拡張している最中に、気が変わっていずれかのダイアログボックス内の **[Cancel]** ボタンをクリックしたとします。この場合は、Server3 へのリソース拡張だけがキャンセルされ、Server2 への拡張はキャンセルされません。この階層から Server2 を削除したい場合は、Server2 からリソースを拡張解除する必要があります。

   **[Next]** をクリックして次のダイアログボックスに進んでください。

4. **[Tag to Extend]** を選択してください。これは、テンプレートサーバからターゲットサーバに拡張する Oracle インスタンスの名前です。ウィザードのドロップダウンメニューには、前のダイアログボックスで選択した、テンプレートサーバ上に作成されているすべてのリソースが表示されます。

**注記:** ここでも、Oracle リソース階層の作成に続いて [Pre-Extend Resource Hierarchy] 作業に入った場合、ウィザードが作成段階にある Oracle リソースのタグ名をすでに確認しているため、 **このダイアログボックスは表示されません** 。GUI ウィンドウの左側ペインの Oracle リソースアイコンまたは右側ペインの Oracle リソースボックスを右クリックして、 **[Extend Resource Hierarchy]** を選択した場合も同様です。

**[Next]** をクリックして次のダイアログボックスに進んでください。

5. Oracle リソース階層を拡張する **ターゲットサーバ** を選択してください。ドロップダウンボックスに、選択された階層にまだ入っていないクラスタ内のサーバ名の一覧が表示されます。

**[Next]** をクリックして次のダイアログボックスに進んでください。

6. **[Switchback Type]** を選択します。この選択によって、バックアップサーバへのフェイルオーバの後、Oracle インスタンスがサービス中に戻ったときに、このサーバにどのようにスイッチバックされるのかが決まります。 *intelligent* または *automatic* を選択できます。Intelligent switchback では、インスタンスをプライマリ / オリジナルサーバにスイッチバックするときに、管理者の介入が必要になります。automatic switchback の場合は、プライマリサーバがオンラインに戻り、LifeKeeper コミュニケーションパスを再確立した直後にスイッチバックが行われます。

スイッチバックタイプは、必要な場合 [Resource Properties] ダイアログボックスの [General] タブで後から変更できます。

**[Next]** をクリックして次のダイアログボックスに進んでください。

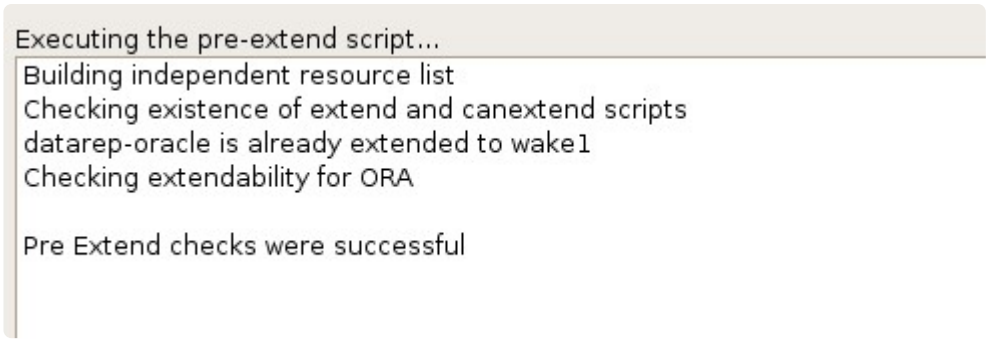
7. **[Template Priority]** を選択または入力します。これはサーバ上で現在サービス中の Oracle 階層の優先順位です。優先順位は、1～999 の範囲で未使用の値が有効で、小さい数字ほど優先順位が高くなります（数字 1 が最高の優先順位に相当します）。拡張処理時に、別のシステムですでに使用中の優先順位をこの階層に対して指定することはできません。デフォルト値を推奨します。注記:このフィールドは階層を最初に拡張するときだけ表示されます。

8. **[Target Priority]** を選択または入力します。これは、他のサーバにある同等の階層に対する、新しく拡張する Oracle 階層の優先順位です。1～999 の範囲で、まだ優先順位として使用されていない値が有効で、リソースのカスケーディングフェイルオーバシーケンスにおけるサーバの優先順位を示します。数値が小さいほど優先順位は高くなります（1 は最高の優先順位を表します）。LifeKeeper のデフォルトでは、階層が作成されたサーバに「1」が割り当てられることに注意してください。優先順位は連続している必要はありませんが、特定のリソースについて 2 つのサーバに同じ

優先順位を割り当てることはできません。

**[Next]** をクリックします。

9. 情報ボックスが表示され、LifeKeeper が環境のチェックを正常に完了し、この Oracle リソースを拡張するためのすべての要件が満たされていることが示されます。満たされていない要件がある場合は、[Next] ボタンは選択できなくなり、[Back] ボタンが有効になります。



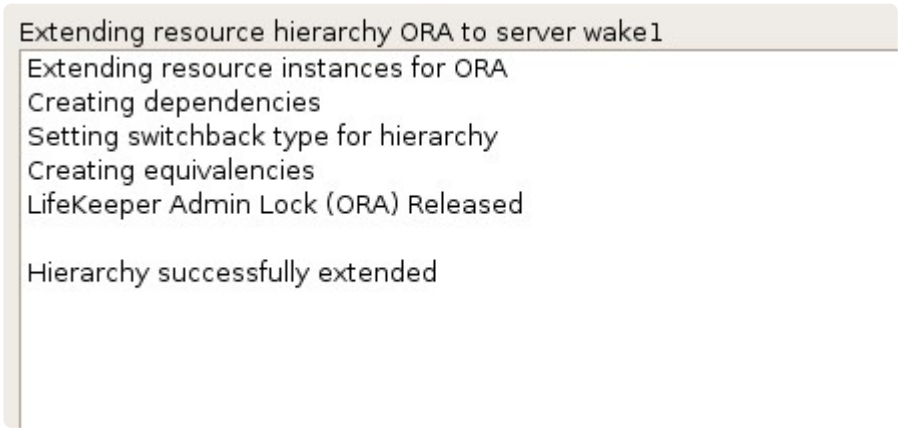

**[Back]** をクリックした場合、情報ボックスに表示されるエラーメッセージの内容に従って、リソースの拡張を変更できます。

ここで **[Cancel]** をクリックすると、いずれかの時点でここに戻り、Oracle リソース階層を別のサーバに拡張して、LifeKeeper の保護下に置く必要があります。

**[Next]** をクリックすると、[Extend Resource Hierarchy] 設定作業に入ります。

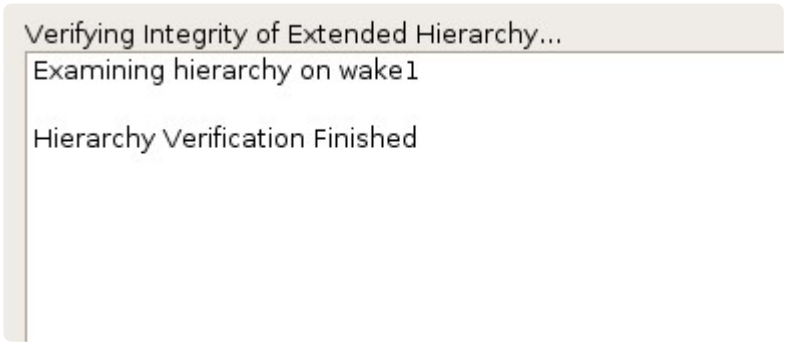
10. **[Database Tag]** が表示されたら、 **[Extend]** をクリックしてください。

11. 拡張が実行されていることを示す情報ボックスが表示されます。



同じ Oracle リソースインスタンスをクラスタ内の別のサーバに拡張したい場合は、 **[Next Server]** をクリックします。その場合は、 **[Extend Resource Hierarchy]** 操作が繰り返されます。

**[Finish]** をクリックすると、Oracle リソースの拡張が正常に完了したことを示す別のダイアログボックスが表示されます。




12. **[Done]** をクリックして、 **[Extend Resources Hierarchy]** メニューを終了します。

    **注記**：必ず両方のサーバで新しいインスタンスの機能をテストしてください。

# 6.12.3.4. リソース階層の拡張解除

1.
2. **[LifeKeeper GUI]** メニューから **[Edit]** を選択し、次に **[Resouce]** を選択してください。ドロップダウンメニューから **[Unextend Resource Hierarchy]** を選択してください。

3. Oracle リソースを拡張解除したい **Target Server** を選択してください。Oracle が現在 In Service のサーバは選択できません。

   **注記:** 右側のペインから個々のリソースインスタンスを右クリックして **拡張解除** 作業を選択した場合、このダイアログボックスは表示されません。

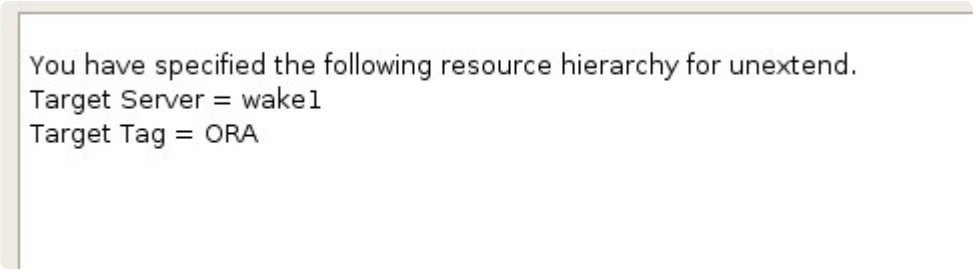
   **[Next]** をクリックして次のダイアログボックスに進んでください。

4. **拡張解除する Oracle 階層** を選択してください。 **注記:** 左側のペインからグローバルリソースを右クリックするか、右側のペインから個々のリソースインスタンスを右クリックして **拡張解除** 作業を選択した場合、このダイアログボックスは表示されません。

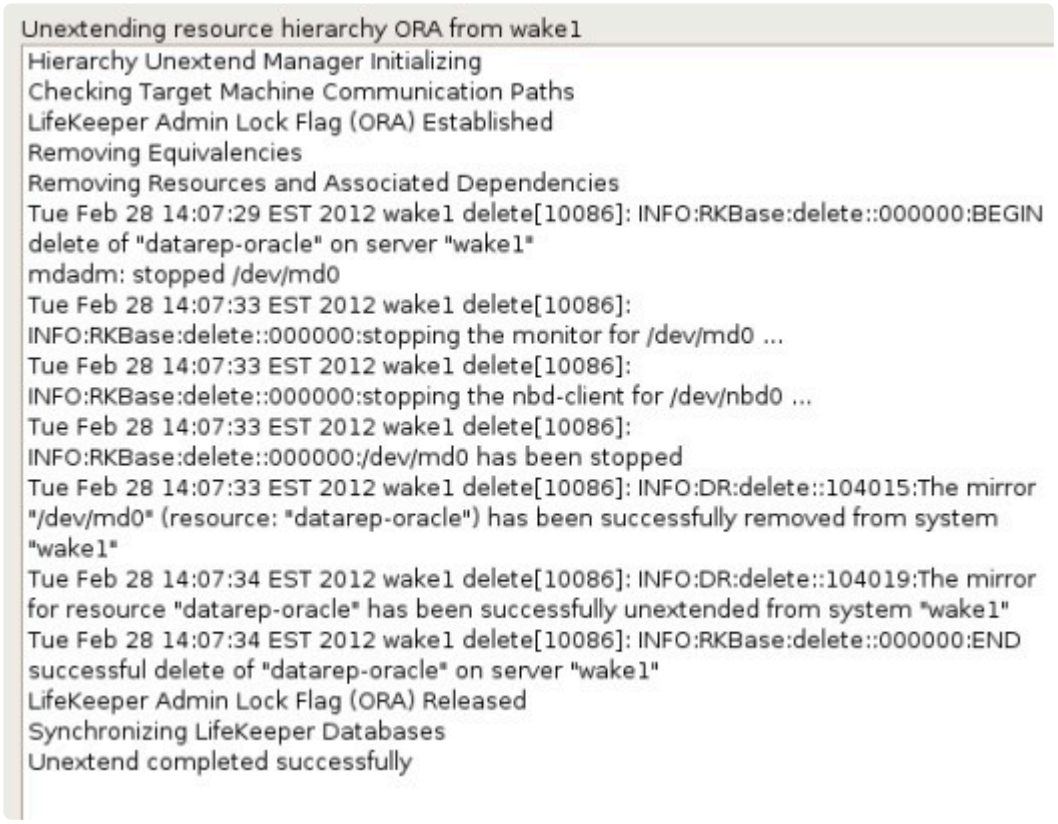
   **[Next]** をクリックして次のダイアログボックスに進んでください。

5. 拡張解除のために選択したターゲットサーバと Oracle リソース階層を確認する情報ボックスが表示されます。

   **[Unextend]** をクリックしてください。

6. Oracle リソースが正常に拡張解除されたことを示す別の情報ボックスが表示されます。



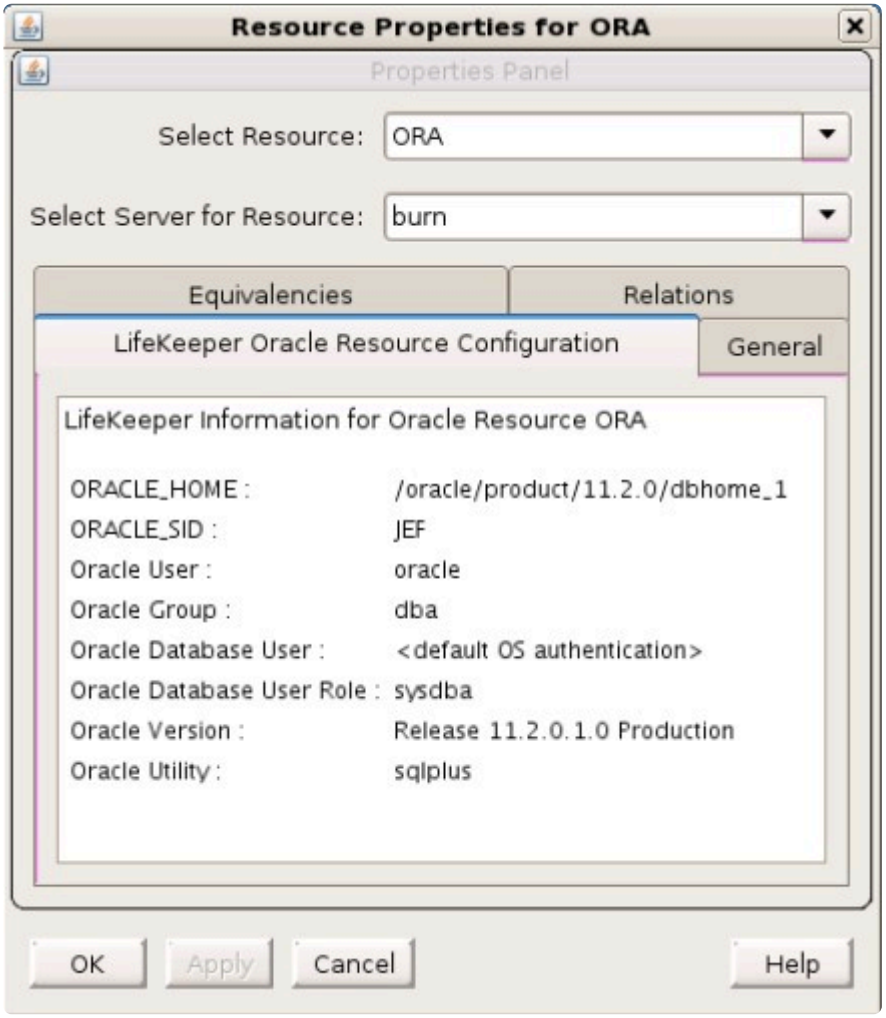

7. **[Done]** をクリックして、 **[Unextend Resource Hierarchy]** メニューの選択を終了します。

# 6.12.3.5. Oracle 設定の参照

**[Resource Properties]** ダイアログは **[Edit]** メニューもしくはリソースコンテキストメニューから参照可能です。このダイアログには、サーバ上の特定のリソースに対するプロパティが表示されます。 **[Edit]** メニューからアクセスした場合は、リソースおよびサーバを選択することができます。リソースコンテキストメニューからアクセスする場合はサーバを選択することができます。

**[Configuration]** タブからは以下の Oracle 設定を参照することができます。

- ORACLE_HOME
- ORACLE_SID
- Oracle User
- Oracle Group
- Oracle Database User
- Oracle Database User Role
- Oracle Version
- Oracle Utility



# 6.12.3.6. Oracle データベースアカウントに対するユーザ名 / パスワードの変更

階層作成後、以下のいずれかの手順でユーザ名およびパスワードを変更してください。

$ORACLE_HOME が共有（もしくは複製）ストレージ上にある場合（一般的なアクティブ-パッシブ構成）：

1.
2. Oracle データベースインスタンスが動作可能なシステム上で LifeKeeper の設定ファイル /etc/default/LifeKeeper を編集し、以下の行をファイルに追加してください。

   ```
   LK_ORA_NICE=1
   ```

   Oracle リソースが定義されているクラスタ内の各システムで同様の処理を行ってください。

3. sqlplusを使用して、Oracle ユーザのパスワードを変更してください。

   ```
   SQL> ALTER USER {username} IDENTIFIED BY {newpassword};
   ```

4. LifeKeeper GUI から、Oracle データベースリソース階層を右クリックし、 **[Change Username / Password]** を選択してください。

5. **[Username]** を入力し、 **[Next]** を選択してください。

6. **[Password]** を入力し、 **[Next]** を選択してください。

7. **[database user role]** を選択し、 **[Apply]** をクリックしてください。認証後、ユーザ名およびパスワードが更新されます。

8. **[Done]** を選択してください。

9. すべてのクラスタノード上で LifeKeeper の設定ファイルを編集し、以下の変更を行ってください。

   ```
   LK_ORA_NICE=0
   ```

$ORACLE_HOME がローカルストレージ上にあり、クラスタの各ノードがそれらの $ORACLE_HOME のコピーを保有している場合（一般的なアクティブ-アクティブ構成）：

1.
2. Oracle データベースインスタンスが動作可能なシステム上で LifeKeeper の設定ファイル /etc/default/LifeKeeper を編集し、以下の行をファイルに追加してください。

   ```
   LK_ORA_NICE=1
   ```

   Oracle リソースが定義されているクラスタ内の各システムで同様の処理を行ってください。

3. sqlplus を使用して、Oracle ユーザのパスワードを変更してください。

   ```
   SQL> ALTER USER {username} IDENTIFIED BY {newpassword};
   ```

4. LifeKeeper GUI から、Oracle データベースリソース階層を右クリックし、 **[Change Username / Password]** を選択してください。

5. sys ユーザのように一時的に代用するユーザ名を **[Username]** に入力し、 **[Next]** を選択してください。

6. **[Password]** を入力し、 **[Next]** を選択してください。

7. **[database user role]** を選択し、 **[Apply]** をクリックしてください。認証後、ユーザ名およびパスワードが更新されます。

8. **[Done]** を選択してください。

9. バックアップシステムで Oracle データベースリソースを「In Service」にしてください。

10. バックアップシステムでデータベースが起動したら、sqlplus を使用して、Oracle アカウントパスワードを変更してください。

    ```
    SQL> ALTER USER {username} IDENTIFIED BY {newpassword};
    ```

    このパスワードを変更する場合、ステップ 2 で設定した新しいパスワードを使用します。この手順は $ORACLE_HOME のセキュリティトークンをリセットします。

11. データベースをクラスタ内の各ノード上で「In Service」にし、手順 8 を繰り返してください。

12. すべてのクラスタノード上でパスワードが変更されたら、適切なノードで Oracle データベースを「In Service」に戻してください。

13. LifeKeeper GUI から、Oracle データベースリソース階層を右クリックし、 **[Change Username / Password]** を選択してください。

14. **[Username]** を入力し、 **[Next]** を選択してください。

15. **[Password]** を入力し、**Next** を選択してください。

16. **[database user role]** を選択し、 **[Apply]** をクリックしてください。認証後、ユーザ名およびパスワードが更新されます。

17. **[Done]** を選択してください。

18. すべてのクラスタノード上で LifeKeeper の設定ファイルを編集し、以下の変更を行ってください。

    ```
    LK_ORA_NICE=0
    ```

# 6.12.3.7. リソース階層のテスト

まず手動スイッチオーバを実施することで、Oracle リソース階層をテストできます。これにより、プライマリサーバからバックアップサーバへのリソースインスタンスのフェイルオーバをシミュレートします。

## GUI による手動スイッチオーバの実行

LifeKeeper GUI メニューで [Edit] の [Resource] から [In Service] を選択すると、手動でのスイッチオーバを開始できます。例えば、バックアップサーバで In Service 要求を実行すると、アプリケーション階層はバックアップサーバで In Service になり、プライマリサーバでは out of service になります。この時点で、元のバックアップサーバがプライマリサーバに、元のプライマリサーバがバックアップサーバに変わります。

Out of Service 要求を実行すると、アプリケーションは他のサーバで In Service にならずに、out of service になります。

## リカバリ動作

プライマリサーバに障害が発生すると、Oracle Recovery Kit ソフトウェアは以下の作業を実行します。

- バックアップサーバのいずれかの物理ネットワークインターフェースで論理インターフェースを In Service にすることによって、Oracle をバックアップサーバで In Service にします。（注記: これは、IP リソースインスタンスが Oracle 階層の子として定義されている場合にのみ実行されます。）
- ファイルシステムをそのサーバの共有ディスク上にマウントします。
- Oracle に関連するデーモンプロセスを開始します。

リカバリ後はセッションコンテキストが失われているため、Oracle ユーザは最初の接続に使用したのとまったく同じ手順で再接続する必要があります。

# 6.12.3.8. DataKeeper を使用した Oracle ノード（SAP/Oracle）へのパッチ適用

ほとんどの Oracle のアップデート/パッチ適用では、システムテーブルドライブへのアクセスのみが必要であり、データードライブ（保護されたDataKeeperのデータードライブ）へのアクセスは必要ありません。

一般的には以下の手順で行うことができます。

手順を開始する前に、プライマリーでブロックフェイルオーバーを設定し、ターゲットでフェイルオーバーを確認します。

1. スタンバイ/ターゲットサーバーで、以下を実施します。
    a. 適切なアップグレード、パッチなどを適用します。
    b. サーバーを再起動します。
2. 次に、リソースをアクティブ/ソースサーバーからスタンバイ/ターゲットサーバーにスイッチオーバーします。
    a. Oracle データーベースにアクセスできることを確認します。
3. もう一方のノード（この時点ではスタンバイノード、手順の当初はアクティブ/ソースサーバー）で、以下を実施します。
    a. 適切なアップグレード、パッチなどを適用します。
    b. サーバーを再起動します。
4. （オプション）再度スイッチオーバーを実行し、元のソース/アクティブサーバーの Oracle への接続とアクセスを確認します。

パッチが DataKeeper ボリュームへのアクセスを必要とする場合（例：catsbp の実行など）は、ミラーを一時停止/ロック解除して、スタンバイ/ターゲットシステムでアップグレードを実行する必要があります。

バックアップで Oracle を起動できることを確認してから、バックアップノードで Oracle を停止し、ミラーリングを続行します。次に、ソースシステムでアップグレードを繰り返します。

手順の最後に、プライマリーの「block failover」フラグとターゲットの「confirm failover」フラグを削除します。

# 6.12.4. Oracle トラブルシューティング

メッセージカタログ は、LifeKeeper for Linux の使用時に検出されるすべてのメッセージのリストを示すとともに、適宜、エラーの原因とエラー状態を解決するために必要な処置の説明を提供します。このリストから受け取ったエラーコードを探すことができます。また、Oracle Recovery Kit の使用時に検出されるすべてのメッセージがリストされた Oracle Kit メッセージカタログ または Oracle Listener メッセージカタログ を直接参照してもかまいません。

- Oracle 既知の問題と制限

# 6.12.4.1. Oracle 既知の問題と制限

## 制御ファイルのスイッチオーバの失敗

$ORACLE_HOME ディレクトリが回復されない場合、データベース制御ファイルが正しくセットアップされていない可能性があります。自動スイッチオーバの場合は、データベースの作成中に制御ファイルを共有デバイス上に設定する必要があります。制御ファイルを別々のサーバに置く場合は、変更が必要になったときに両方のサーバを手動でアップデートする必要があります。

## 出力の打ち切り

Oracle の一部のバージョンでは、show parameters control_files を sqldba モードで実行するときに、出力が打ち切られることがあります。使用している Oracle のバージョンがそのような動作をする場合は、以下の点を確認してください。

- controlfile パラメータ。パラメータが、*$Oracle_HOME/dbs/init#SID.ora* ファイルにあることを確認します。
- controlfile デバイス。controlfile デバイスが、新規行ではなく、継続行で指定され、各デバイスがコンマで区切られていることを確認します。

Oracle 関連のデバイスが正しく設定されていない場合は、LifeKeeper Application 管理の下で利用可能なファイルシステムアプリケーションを使用して、手動で設定できます。

## 共有ドライブ上に配置された Flash Recovery Destination

構成のセクションでも記載されているように、Flash Recovery destination が、共有ドライブ上に配置されていることが重要です。Flash Recovery Area がどこに配置されているかを確認する場合は、SYSDBA として以下のクエリーを発行してください。

```
SQL> SELECT substr(Name,1,30) Name,
  (SPACE_LIMIT/1024/1024/1024) Space_Limit_GB,
       SPACE_USED/1024/1024/1024 Space_Used_GB,
       SPACE_RECLAIMABLE, NUMBER_OF_FILES
FROM V$RECOVERY_FILE_DEST;

NAME                SPACE_LIMIT_GB SPACE_USED_GB SPACE_RECLAIMABLE
------------------- -------------- ------------- -----------------
NUMBER_OF_FILES
---------------
/U01/flash_recovery_area    3.76171875    .156448364           0
              4
```

下記は、このタスクを完了させるために、 *$ORACLE_HOME/dbs/spfile<sid>* に変更を行う場合の例です。

```
SQL> ALTER SYSTEM SET DB_RECOVERY_FILE_DEST='/oracledb/oracle/flash_recovery_a
rea' scope=both;
System altered.
SQL> show parameter DB_RECOVERY_FILE_DEST;
NAME                          TYPE        VALUE
------------------------------------------------------------------------
db_recovery_file_dest         string      /oracledb/oracle/flash_recovery_area
db_recovery_file_dest_size    big integer 2G
SQL> commit;
```

## データベースに接続できない場合のフェイルオーバの回避

リソースのヘルスチェックの実行時に、Oracle ARK は実行中のデータベースプロセスをチェックし、データベースへの接続を試行します。最大許容接続に達したことで生じるヘルスチェック障害を回避するには、 */etc/default/LifeKeeper* で以下を設定します。

```
LK_ORA_NICE=1
```

**注記:** LK_ORA_NICE を設定することで、非稼働データベースによって生じる他のタイプの接続エラーをマスクできます。この設定を使用する場合は、注意してください。

## *oratab* の非伝統的な場所

デフォルトでは、Oracle ARK は *oratab* ファイルをまず */etc* で検索した後、 */var/opt/oracle* でこのファイルを検索します。このいずれかのデフォルトの場所に *oratab* ファイルがない場合は、 ORACLE_ORATABLOC を */etc/default/LifeKeeper* で *oratab.* が含まれるディレクトリに設定する必要があります。

## サポートされていない NFS バージョン 4

Oracle Recovery Kit は共有データベースストレージで NFSv3 をサポートしています。現時点では、NFSv4 のファイルロックメカニズムにより NFSv4 はサポートされていません。

## ローカルストレージでの Oracle バイナリのインストール後の Oracle データベース階層の作成

各 LifeKeeper クラスタノードのローカルストレージに Oracle バイナリ *（$ORACLE_BASE）* をインストールした場合、LifeKeeper GUI で Oracle データベース階層の作成時に以下のようなメッセージが表示されます。

```
BEGIN create of <SID> on server <server1（サーバ server1 でSID の作成を開
始）>
Creating resource instance <SID> on server <server1（サーバ server1 でリソー
スインスタンス SID を作成）>
```

Setting resource state for <SID> on server <server1> to "ISP". (サーバ server1 で SID のリソース状態を「ISP」に設定)

ORACLE_HOME「/opt/oracle/app/oracle/product/11.2.0/dbhome_1」が共有ファイルシステムにありません。ORACLE_HOME ディレクトリおよび関連ファイルがすべてのサーバで同じであるか確認してください。詳細については、LifeKeeper Oracle Recovery Kit のドキュメントを参照してください。
Creating dependent file system resource "/u00" on <server1>. (server1 で依存ファイルシステムリソース「/u00」を作成)
Creating dependency between Oracle database "SID (SID)" and the dependent resource "/u00" on <server1>. (server1 で Oracle データベース「SID (SID) 」と依存リソース「/u00」間の依存関係を作成)
Creating dependency between Oracle database "SID (SID)" and the listener resource "LSNR.LISTENER" on <server1>. (server1 で Oracle データベース「SID (SID) とリスナーリソース「LSNR.LISTENER」間の依存関係を作成)
Performing in-service of new Oracle resource tag=< SID > on <server1>. (server1 で新しい Oracle リソースタグ = SID の In Service を実行)
END successful create of <SID> on server <server1 (サーバ server1 で SID の作成が正常終了) >

また、LifeKeeper ログにも似たような警告が示されます。この警告が不要で、階層を拡張して作業を続行した後、データベースリソースを別のノードで In Service にしようとした場合、[LifeKeeper GUI] ダイアログに以下のようなメッセージが表示されます。

Put resource "OST" in-service (リソース「OST」を In Service にする)
BEGIN restore of "OST" on server "cae-qa-v39" (サーバ「cae-qa-v39」で「OST」のリストアを開始)
Begin the "start [ start.normal ]" of the database "OST" on "cae-qa-v39". (「cae-qa-v39」でデータベース「OST」の「start[start.normal]」を開始)
The "start [ start.normal ]" attempt of the database "OST" appears to have failed on "cae-qa-v39". (「cae-qa-v39」でデータベース「OST」の「start [start.normal]」試行が失敗したようです)
ORA-01078: failure in processing system parameters (ORA-01078: システムパラメータの処理障害)
LRM-00109: could not open parameter file ‘/opt/oracle/app/oracle/product/11.2.0/dbhome_1/dbs/initOST.ora’ (LRM-00109: パラメータファイル「/opt/oracle/app/oracle/product/11.2.0/dbhome_1/dbs/initOST.ora」を開くことができませんでした)
Begin the "start [ start.force ]" of the database "OST" on "cae-qa-v39". (「cae-qa-v39」でデータベース「OST」の「start[start.force]」を開始)
The "start [ start.force ]" attempt of the database "OST" appears to have failed on "cae-qa-v39". (「cae-qa-v39」でデータベース「OST」の「start [start.force]」試行が失敗したようです)
select ‘alter database datafile ‘||file#||’ end backup;’ from v\$backup where status = ‘ACTIVE’ (ステータス =「ACTIVE」の v\$backup から「alter

database datafile '||file#||' end backup;」を選択してください)

また、ダイアログボックスに以下のようなメッセージが表示されることもあります。

Put resource "OST" in-service (リソース「OST」を In Service にする)
BEGIN restore of "OST" on server "cae-qa-v39" (サーバ「cae-qa-v39」で「OST」のリストアを開始)
Begin the "start [ start.normal ]" of the database "OST" on "cae-qa-v39". (「cae-qa-v39」でデータベース「OST」の「start[start.normal]」を開始)
122528 Initial inspection of "COMMAND" failed, verifying failure or success of received output.」(122528 「COMMAND」の初期点検が失敗し、受信出力の成否を検証できませんでした)
Logon failed with "" for "OST" on "cae-qa-v39". (「cae-qa-v39」の「OST」で「」でログオンが失敗しました) Please check username/password and privileges. (ユーザ名 / パスワードおよび権限をチェックしてください)
The "start [ start.normal ]" attempt of the database "OST" appears to have failed on "cae-qa-v39". (「cae-qa-v39」でデータベース「OST」の「start [start.normal]」試行が失敗したようです)
ERROR: (エラー: )

ORA-01031: insufficient privileges (ORA-01031: 不十分な権限)

Enter password: (パスワードを入力してください: )

ERROR: (エラー: )

ORA-01005: null password given; logon denied (ORA-01005: null パスワードが返されました。ログオンが拒否されます)

この問題を解決するには、***$ORACLE_BASE/admin*** をデータベースインスタンスが作成されたプライマリシステムからバックアップシステム（階層が拡張された場所）の ***$ORACLE_BASE/admin*** にコピーしてください。また、このディレクトリの所有権を Oracle ユーザ名および Oracle グループ（通常 **oracle:oinstall**）に変更してください。

また、すべての ****{$ORACLE_SID}**** (この例では ***OST***) ファイルをプライマリファイルの ***$ORACLE_BASE/product/11.2.0/dbhome_1/dbs***からバックアップシステムの ***ORACLE_BASE/product/11.2.0/dbhome_1/dbs*** にコピーしてください。

例えば、これらはプライマリシステムからバックアップへコピーされたファイルであり、ORACLE SID が OST であるとします。

```
-rw-r—— 1 oracle oinstall 1544 2012-05-09 11:02 hc_OST.dat
-rw-r—— 1 oracle oinstall 24 2012-01-31 10:22 lkOST
-rw-r—— 1 oracle oinstall 1536 2012-03-05 09:02 orapwOST
-rw-r—— 1 oracle oinstall 2560 2012-05-09 10:58
spfileOST.ora
```

SID が ORA01 のもう 1 つの例では、以下のファイルがコピーされます。

```
-rw-r--- 1 oracle oinstall 1536 2010-09-08 18:25 orapwORA01
-rw-r--- 1 oracle oinstall 24 2010-09-08 18:25 lkORA01
-rw-r--- 1 oracle oinstall 2560 2010-09-08 18:30 spfileORA01.ora
-rw-r--- 1 oracle oinstall 1544 2010-09-08 18:30
hc_ORA01.dat
```

およびディレクトリ

```
peshm_ORA01_0/:
```

## フェイルオーバ後も Oracle リスナーがプライマリサーバ上で稼動したままになる

ネットワーク障害に起因した Oracle アプリケーションのフェイルオーバ後に、プライマリサーバ上でリスナー・プロセスが稼動し続けることがあります。

# 6.12.4.1.1. データベース作成時の問題

| | |
|---|---|
| 問題: | dbca を使用してデータベースを作成したときに、次のメッセージが表示された。「ORA-00439 feature not enabled: string」(ORA-00439 この機能は使用できません：文字列) |
| 処置: | 環境変数 $ORACLE_SID の値を調べて、作成している SID と同じか確認してください。 |
| 問題: | スクリプトからデータベースを作成したときに、次のメッセージが表示された。「ORA-01092 ORACLE instance terminated.Disconnection forced」(ORA-01092 Oracle インスタンスが終了しました。強制的に切断されます) |
| 処置: | bdump ディレクトリにある警告を参照してください。「ORA-12714 invalid national character set specified」(ORA-12714 無効な各国語キャラクタセットが指定されました) というメッセージが表示される場合には、環境変数 $ORA_NLS33 の値を調べて、正しいロケーションに設定されているか確認してください。 |
| 問題: | dbca で生成されたスクリプトでデータベース作成中に問題が発生したら、次の手順を実行してください。 |
| 処置: | 1. 次のディレクトリがなければ、必ず作成してください<br><br>*bdump*<br>*cdump*<br>*udump*<br>*<oracle data base directory>/oradata sid*<br>*<oracle data base directory>/dbs*<br>*<oracle data base directory>/admin/<SID>*<br><br>*bdump* ディレクトリと *udump* ディレクトリのパスを調べる場合は、初期設定ファイル（*init<SID>.ora*）を確認してください。<br><br>2. ファイル *$ORACLE_HOME/dbs/orapw* が存在するか確認してください。存在しない場合には、orapwd ユーティリティを使用して作成してください。 |

# 6.12.4.1.2. データベース起動時の問題

| | |
|---|---|
| **問題:** | sqlplus を使用してデータベースを起動中に、次のメッセージが表示された。「ORA-03113 end-of-file on communication channel」(ORA-03113 通信チャネルでファイルが終わりになりました) |
| **処置::** | 初期設定ファイル (*init<SID>ora*) とパスワードファイル (*orapw<SID>*) がディレクトリ *$ORACLE_HOME/dbs* にあるか確認してください。 |
| **問題:** | データベースの起動時に、次のメッセージが表示された。「ORA-01092 ORACLE instance terminated.Disconnection forced」(ORA-01092 Oracle インスタンスが終了しました。強制的に切断されます) |
| **処置::** | *bdump* ディレクトリにある警告を参照してください。「ORA-12701 CREATE DATABASE character set is not known」(ORA-12701 CREATE DATABASE で指定したキャラクタセットが不明です) というメッセージが表示される場合には、環境変数 $ORA_NLS33 の値を調べて、正しいロケーションに設定されているか確認してください。 |

# 6.12.4.1.3. LifeKeeper の inqfail エラーログ

フェイルオーバーに続いて、inqfail エラーが LifeKeeper エラーログに表示された場合は、filesystemio の設定を変更する必要があります。

**注記:** ディスク ID およびサーバ名はそれぞれの設定で異なります。

この問題を解決するには、filesystemio=「SETALL」 を filesystemio=「ASYNCH」へ設定変更する必要があります。

この設定を探すには、以下の SQL コマンドでオプションを確認してください。

1. SQL> show parameter filesystemio;
2. 以下のコマンドを使用して設定を変更してください。

   SQL> alter system set filesystemio_options=<XXXXXXX> scope=spfile;

   <XXXXXXX> は以下に設定できます。

   <XXXXXXX> = {none | setall | direction | asynch}

   NONE – 最適化なし

   ASYNC - 非同期 I/O を有効化

   DIRECTIO – ダイレクト I/O を有効化

   SETALL – 使用可能なすべての機能を有効化

   **重要:** パラメータの再設定の後、Oracle を再起動する必要があります。

# 6.12.5. 付録

Raw I/O を使用するための Oracle のセットアップ

階層作成後のテーブルスペースの追加

複数のリソースに対する共有 Oracle Listener の作成

Listener 保護レベルのアップデート

Listener リカバリレベルのアップデート

保護された Listener のアップデート

# 6.12.5.1. Raw I/O を使用するための Oracle のセットアップ

以下の手順に従って、ファイルの代わりに共有 Raw I/O デバイスを使用する Oracle データベースを作成します。

1. データベースを作成するために最低限必要なファイルの数とサイズを決めます。これには、制御ファイル、テーブルスペース、REDO ログが含まれます。それらのアイテムの一部をファイルとし、それ以外を Raw I/O デバイスとする混合セットアップも可能です。すべての Raw I/O デバイスが共有ディスクパーティションを使用する必要があります。

2. 必要な数の Raw I/O デバイスを使用して、Raw I/O セットアップを作成します。

   a. Oracle データベースを作成するときに指定するサイズ以上のサイズで RAW デバイスを作成します。

   b. `raw` コマンドを使用して、システム初期化ファイル（つまり、`boot.local` または `rc.local`）にRAW デバイスマッピングを作成します。どの RAW デバイスがどの Oracle ファイルを表すのかを識別するために、意味のあるコメントを追加してください。これは、システムのリブートを行う場合に、マッピングを再確立できるようにするために必要です。Raw I/O デバイスが LifeKeeper の保護下に置かれたら、これらのマッピングは手動でファイルから削除してください。

3. 以下のコマンドを使用して、RAW デバイスを Oracle データベースに書き込み可能にします。
   `chown oracle:dba /dev/raw[0-9]*`

   ここで、オーナおよびグループは、Oracle インスタンスの構成に固有です。

4. マッピングを含むファイルを実行して、RAW デバイスの設定をアクティブにします。

5. データベース作成スクリプトがすでに存在している場合は、ステップ 6 へ進みます。データベース作成スクリプトがない場合は、Oracle Java GUI ツールの 1 つ（`dbassist` または `dbca`）を使用して作成できます。どちらのツールを使用する場合も「**Save As Script**」を選択してください。データベースの作成を選択しないでください。

   **注記:**

   - `dbca` で、スクリプトを生成するには、「**New Database**」テンプレートを選択する必要があります。必要であれば、ファイル名を共有デバイスに変更し、構成の値を調整します。
   - この時点で DB 作成プロセスを開始しないでください。`dbassist` ツールは、各テーブルスペースに指定されたファイルがすでに存在しているかどうかをチェックし、ファイルが存在している場合は処理を行いません。`dbca` ツールはファイルを上書きするかどうかを確認するメッセージを表示しますが、RAW デバイスでは失敗します。どちらの場合も、これらのツールからRAW デバイスを直接使用することはできません。

6. データベース作成スクリプト（既存のもの、もしくは dbassist または dbca で作成したもの）を編集する必要があります。該当するファイル名（パスを含む）を、Raw I/O デバイスのフルパス名に置き換えます。対象となるファイルには（最低限）データベース作成ファイル（CREATE DATABASE コマンド用）とテーブルスペース作成ファイル（CREATE TABLESPACE コマンド用）があります。選択した dbassist または dbca のオプションによっては、さらに別のファイルを編集しなければならない場合があります。また、必要に応じて、初期化ファイルを編集し、制御ファイルを Raw I/O デバイスに変更します。初期化ファイルは、作成スクリプトが存在するディレクトリにあります。データファイルの場合、編集結果は以下のようになります。

```
. . .

CREATE DATABASE “LK”

     maxdatafiles 254

     maxinstances 8

     maxlogfiles 32

     character set US7ASCII

     national character set US7ASCII

DATAFILE ‘/dev/raw/raw1’ SIZE 260M AUTOEXTEND ON NEXT 10240K

logfile ‘/ora/LK/redo01.log’ SIZE 500K,

       ‘/ora/LK/redo02.log’ SIZE 500K,

       ‘/ora/LK/redo03.log’ SIZE 500K;

. . .
```

Raw I/O デバイスのサイズは、格納するデータのために Oracle が必要とする最低限のサイズにします。

7. ここで、ステップ 5 で作成したスクリプトを実行してデータベースを作成します。

8. 作成ログをチェックして、データベースまたはテーブルスペースのエラーが発生していないか確認します。

9. 作成スクリプトでデータベースを作成中に問題が発生するか、後でRAW デバイスにテーブルスペースを追加する場合、適切なツール（dbassist または dbca）でデータベースを作成する必要があります。その後、sql ユーティリティから次のようなコマンドを実行して、Raw I/O デバイスデータファイルを追加します。

```
tablespace RAWTS
```

```
DATAFILE '/dev/raw/raw217' SIZE 50M REUSE
DEFAULT STORAGE (INITIAL 50K NEXT 50K
MINEXTENTS 1 MAXEXTENTS 4) ONLINE
```

10. udev の規則を追加して、RAW デバイスの権限および所有権をリブート、切り替え、フェイルオーバがあっても変わらないようにします。

    Linux 2.6 カーネルでは、udev システムは、RAW デバイスなどの特殊ファイルの作成をカーネルが制御するためのデフォルトの方法です。RAW デバイスを Oracle で使用する場合、固有の所有権および権限の設定が必要です。このような固有の設定は、カーネルのデフォルトの設定と競合します。固有の設定をアドレス指定するには、udev の規則を使用して、LifeKeeper が保護する Oracle SID で使用する Oracle ユーザおよび Oracle グループにRAW デバイスの所有権を設定する必要があります。

    **注記:** 一部の OS ディストリビューションでは、デバイスを作成するための規則とデバイス権限を設定するための規則を分ける必要があります。詳細については、OS ディストリビューションの udev ドキュメントを参照してください。

    OS ディストリビューションで動作する udev 規則の例を下記に示します。

    ```
    KERNEL=="raw10", RUN+="/bin/chown oracle:oinstall /dev/raw/raw10"

    KERNEL=="raw[3-5]*", OWNER="oracle", GROUP="oinstall", MODE="660"
    ```

    リソース階層を In Service にする前に、作成した udev の規則を LifeKeeper クラスタのすべてのノードに適用する必要があります。

# 6.12.5.1.1. 階層作成後のテーブルスペースの追加

LifeKeeper で Oracle 階層を作成した後に Raw I/O デバイスでテーブルスペースを追加する場合は、GUI を使用して LifeKeeper Raw I/O 階層を作成し、Oracle リソース（親）と Raw I/O リソース（子）の間に依存関係を手動で作成する必要があります。

# 6.12.5.2. 複数のリソースに対する共有 Oracle Listener の作成

ご利用のシステム構成で次のいずれかが当てはまる場合、Oracle Listener を作成できます。

- 複数の Oracle SID に対して、複数の Listener が定義されている。
- Oracle Listener が設定において重要なコンポーネントである。
- 複数の Oracle SID に対して 1 つの Listener が定義されている。

> **注記:** 複数の Listener を SPS で保護する場合、LISTENER/SID_LIST_LISTENER スタンザのコレクションはすべて一意である必要があります。（例：11g インストールの listener.ora ファイルにはスタンザ LISTENER_11G_1/SID_LIST_LISTENER_11G_1 が含まれます。）

このプロセスにより、LifeKeeper 内の Listener を保護し、さまざまな Listener と SID の組み合わせに対応できるようになります。

複数のリソースに対して 1 つの共有 Oracle Listener を作成する場合は、下記の手順に従ってください。

1. LifeKeeper GUI メニューから **[Edit]** を選択し、次に **[Server]** を選択します。ドロップダウンメニューから、 **[Create Resource Hierarchy]** を選択します。

   **重要:** リソースを作成するときは、Oracle アプリケーションが実行されている必要があります。

2. ダイアログボックスが表示され、クラスタ内にインストール済みの認識されているすべての Recovery Kit がドロップダウンリストボックスに表示されます。ドロップダウンリストから **[Oracle Database Listener]** を選択します。 **[Next]** をクリックして次のダイアログボックスに進んでください。

3. 次の表で説明している情報を入力します。ダイアログボックスで **[Back]** ボタンが有効な場合は、前のダイアログボックスに戻ることができます。これは、エラーが発生して、以前に入力した情報を訂正する必要がある場合に特に役立ちます。階層作成手順のどの段階でも、 **[Cancel]** をクリックすると、作成プロセス全体がキャンセルされます。

| フィールド | ヒント |
|---|---|
| スイッチバックタイプ | intelligent または automatic を選択します。これは、サーバがフェイルオーバした後、どのように Listener リソースがサーバにスイッチバックされるかを指示します。データレプリケーションを使用する場合は、スイッチバックタイプは intelligent に選択してください。<br><br>**注記:** スイッチバックタイプは Listener リソースによって使用される依存リソース（IP およびボリュームリソース）と一致している必要があります。そうでない場合は、作成に失敗します。 |
| サーバ | 階層を作成したいサーバを選択してください。 |

| | |
|---|---|
| Listener 設定ファイルのパス | Oracle Listener 設定ファイルへのフルパスを選択してください。 |
| Listener 名 | このリソースインスタンスで保護する Oracle Listener の名前を選択してください。 |
| Listener の実行パス | Oracle Listener プログラムの実行パスを選択します。このパスは Oracle Listener を起動･停止･監視･回復する際に必要となります。 |
| Listener の保護レベル | 以下のレベルの 1 つを選択してください。<br><br>Full Control（開始、停止、監視および回復）<br><br>Intermediate Control（開始、監視および回復）<br><br>Minimal Control（開始および監視のみ） |
| Listener のリカバリレベル | 指定された listener に対するリカバリレベルを選択してください。<br><br>Standard (On) - 標準の LifeKeeper リカバリを有効にします。Standard (On) が選択されている場合は、すべての listener 障害がローカルで発生し、必要であれば、有効なバックアップサーバへのフェイルオーバを行います。<br><br>Optional (off) - オプションの LifeKeeper リカバリを有効にします。Optional (Off) を選択した場合は、すべての listener 障害がローカルで発生しますが、有効なバックアップサーバへのフェイルオーバは行いません。<br><br>**注記** ：どちらのリカバリレベルもlistener障害が発生した際に、ローカルリカバリは実施しますが、フェイルオーバ実施の有無に違いがあります。 |
| IP アドレス名 | このリソース階層の依存関係として保護される IP アドレスのリソース名を選択してください。選択された listener に関連した IP アドレスは選択リストに表示されます。IP リソースが設定に必要がない場合は [None] を選択してください。 |
| Listener タグ | サーバ上のリソースに対するユニークな名前を入力してください。名前に対する有効な文字は、アルファベット、数字および特殊記号 - _ . / になります。 |

4. **[Create]** ボタンを選択し、階層の作成を開始します。インフォメーションボックスが表示され、LifeKeeper がデータベース listener リソース階層を作成するために提供した有効なデータを確認します。LifeKeeper が問題を検知した場合は、インフォメーションボックスにエラーが表示されます。認証に成功すると、リソースが作成されます。

5. ***[Pre-Extend Wizard]*** ダイアログはリソース階層の作成が完了した場合に表示され、以下の情報を入力することになります。 **Extend** のオペレーションに慣れていない場合は、それぞれのダイアログで選択をした後 **[Next]** を選択してください。LifeKeeper の **[Extend Resource Hierarchy]** 作業

のデフォルト値が分かっていて、入力と確認を省略する場合は、 **[AcceptDefaults]** をクリックします。

| フィールド | ヒント |
|---|---|
| ターゲットサーバ | 階層が拡張されるターゲットサーバを選択してください。リソースが少なくとも 1 つのサーバへ拡張される前にキャンセルを選択した場合は、LifeKeeper は階層のアプリケーションに対して保護をしません。 |
| スイッチバックタイプ | バックアップサーバへフェイルオーバした後、 in service にする場合に、どのように Oracle Listener インスタンスがこのサーバにスイッチバックするかを指示します。intelligent または automatic を選択できます。スイッチバックタイプは、必要な場合 [Resource Properties] ダイアログボックスの [General] タブで後から変更できます。<br>注記: スイッチバックストラテジーは、Oracle Listener リソースによって使用される依存リソースのものと一致している必要があります。 |
| テンプレートの優先順位 | このフィールドは、作成時に拡張しなかった場合に表示されます。<br>リソースに対するカスケーディングフェイルオーバシーケンスにおける、テンプレートサーバの優先順位を 1 から 999 までの間の値を入力してください。低い数値は高い優先順位となります。LifeKeeper は、階層が作成されたサーバの値を「1」に割り当てています。2 つのサーバは与えられたリソースに対して同じ優先順位をもつことはできません。 |
| ターゲットの優先順位 | リソースに対するカスケーディングフェイルオーバシーケンスにおける、ターゲットサーバの優先順位を 1 から 999 までの間の値を入力してください。低い数値は高い優先順位となります。LifeKeeper は、階層が拡張される最初のサーバのデフォルト値を 10 にすることを推奨します。 |

6. pre-extend チェックに成功したというメッセージを受けた後、 [Next] をクリックし、以下の情報を入力してください。

| フィールド | ヒント |
|---|---|
| Listener 設定ファイルのパス | Oracle Listener 設定ファイルに対するフルパスを選択してください。 |
| Listener の実行パス | Oracle Listener プログラムの実行パスを選択します。このパスは Oracle Listener を起動･停止･監視･回復する際に必要となります。 |
| Listener | このフィールドはプライマリサーバ上の新しい Oracle Listener リソースに対してユニークな名前 |

| | |
|---|---|
| タグ | で自動的に割り当てられます。デフォルトのネーミングパターンが表示されます。他のユニークな名前を入力することもできます。Listener タグに対する有効な文字は、アルファベット、数字および特殊記号 - : . / になります。 |

7. **[Extend]** をクリックします。 **[Hierarchy Integrity Verification]** ウィンドウが、 **Hierarchy Verfication Finished（階層完了）** のメッセージと共に表示されます。 **[Next Server]** または **[Finish]** をクリックしてください。

# 6.12.5.2.1. Listener 保護レベルのアップデート

1. リソース を選択後、 **[Resource]** ツールバーから ボタンを選択し、リソースの保護レベルをアップデートしてください。

2. 以下の情報を入力してください。

| フィールド | ヒント |
|---|---|
| **Listener** の保護レベル | 以下の 1 つを選択してください。<br><br>Full Control（開始、停止、監視および回復）<br><br>Intermediate Control（開始、監視および回復）<br><br>Minimal Control（開始および監視のみ） |

3. **[Update]** を選択し、現在の状態から新しい状態へと **保護レベル** を変更してください。現在の状態を変更しない場合は、 **[Cancel]** を選択してください。

# 6.12.5.2.2. Listener リカバリレベルのアップデート

1. **listener** を選択後、 **[Resource]** ツールバーから ボタンを選択し、リソースのリカバリレベルをアップデートしてください。

2. 以下の情報を入力してください。

| フィールド | ヒント |
|---|---|
| **Listener のリカバリレベル** | 指定された listener に対するリカバリレベルを選択してください。<br><br>Standard (On) - 標準の LifeKeeper リカバリを有効にします。Standard (On) が選択されている場合は、すべての listener 障害がローカルで発生し、必要であれば、有効なバックアップサーバへのフェイルオーバを行います。<br><br>Optional (off) - オプションの LifeKeeper リカバリを有効にします。Optional (Off) を選択した場合は、すべての listener 障害がローカルで発生しますが、有効なバックアップサーバへのフェイルオーバは行いません。<br><br>**注記** ：どちらのリカバリレベルもlistener障害が発生した際に、ローカルリカバリは実施しますが、フェイルオーバ実施の有無に違いがあります。 |
| **アップデートの確認** | [Update] ボタンを選択し、リカバリレベルを現在の状態から新しい状態へと変更してください。<br><br>現在の状態を変更しない場合は [Cancel] を選択してください。 |

3. **[Update]** をクリックし、リカバリレベルを現在の状態から新しい状態へと変更してください。現在の状態を変更しない場合は **[Cancel]** を選択してください。

✻ 注意: tnsnames.ora の FAILOVER パラメーターは、LifeKeeper のフェールオーバーやリカバリーとは異なります。接続の移動はOracle のソフトウェアが実行しますが、listener のリカバリーは LifeKeeper Oracle Listener Recovery Kit が行います。これらはそれぞれ独立しています。

✻ 注意: SIOS Protection Suite for Linux または Oracle のどちらにおいても“FAILOVER”パラメーターをオンにすることでのフェールオーバーはサポートされていません。oracleのフェイルオーバーに対応するために、LifeKeeperに特別なパラメータや設定をする必要はありません。LifeKeeper でのフェールオーバーはクライアント接続の障害では発生しません。quickCheck およびイベントベースのエラー後に発生します。

# 6.12.5.2.3. 保護された Listener のアップデート

1. **Listener** を選択後、 **[Resource]** ツールバーから ボタンを選択し、保護された Listener のアップデートを行ってください。

2. 以下の情報を入力してください。

| フィールド | ヒント |
|---|---|
| **Listener 名** | リソースインスタンスに対して、保護する Oracle Listener の名前を選択してください。 |
| **IP アドレス名** | このリソース階層の依存関係として追加する IP アドレスのリソース名を選択してください。<br><br>この設定に対して追加する IP リソースが存在しない場合は [None] を選択してください。 |

3. **[Update]** を選択し、 **保護された Listener** および **IP の割り当て** を現在の状態から新しい状態へと変更してください。現在の状態を変更しない場合は **[Cancel]** を選択してください。

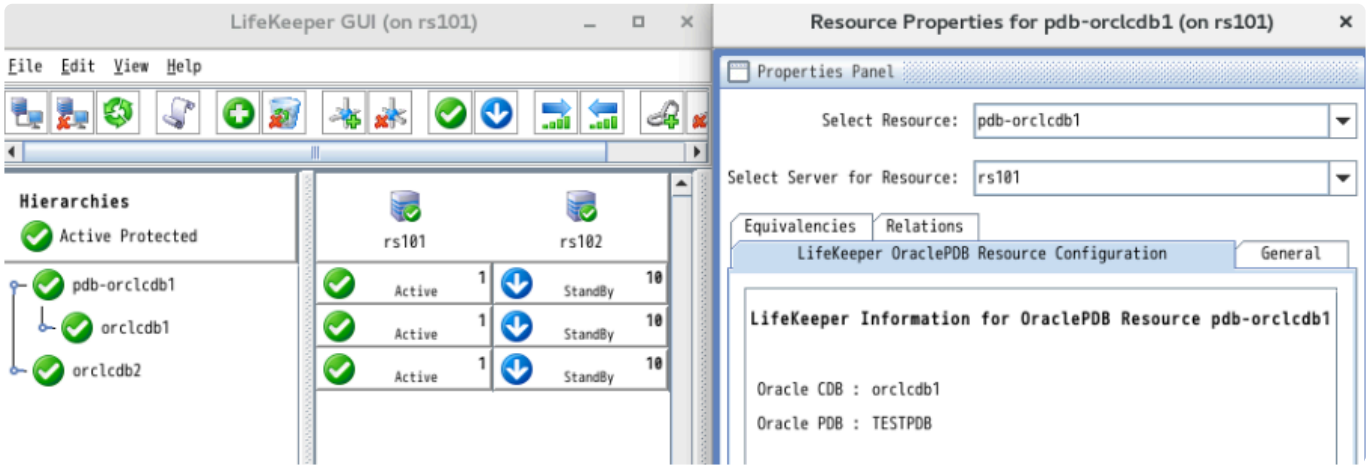
   **[Update]** を選択した場合、特定のリソースに対する保護された Listener がアップデートされている状態がダイアログに表示されます。 **[Finish]** をクリックしてください。

# 6.12.5.3. Pluggable Database の移行

LifeKeeper で保護されている、コンテナ・データベース（以下、CDB）間でプラガブル・データベース（以下、PDB）を移行する手順を紹介します。事前にデータベースのバックアップを取得しておくことをおすすめします。

## 移行手順

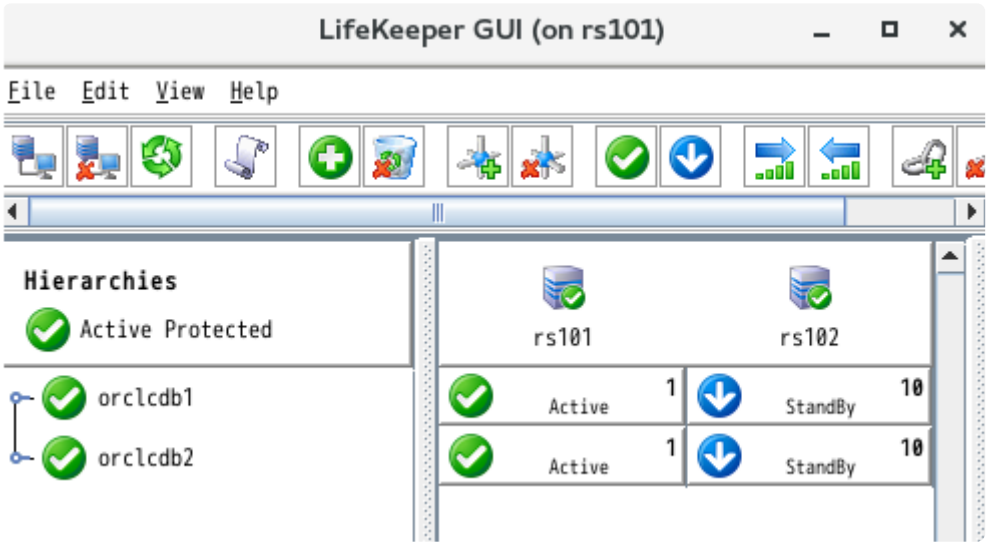
本手順では、PDB のプラグ/アンプラグを使用した方法による PDB の移行を示します。
下図の構成の場合を例に、ソース CDB （ORCLCDB1）から ターゲット CDB （ORCLCDB2）へ PDB （TESTPDB） を移行する手順です。




1. 移行前にソース CDB に接続し、対象の PDB の情報を確認します。

```
SQL> COLUMN NAME FORMAT A8

     SELECT NAME, DBID, GUID FROM V$CONTAINERS WHERE NAME='<PDB>';
```

2. Oracle PDB リソースを Out of Service にし、リソースを削除します。
   複数の PDB を保護している Oracle PDB リソースがある場合は、リソース設定 [Change Protection PDB] （保護するPDBの変更 を参照）で対象の PDB を保護対象から削除します。




3. ソース CDB に接続し対象の PDB をアンプラグします。（PDB が停止していない場合、PDB を停止します。）

```
SQL> ALTER PLUGGABLE DATABASE <PDB> UNPLUG INTO '/home/
oracle/<PDB>.xml';

SQL> DROP PLUGGABLE DATABASE <PDB>;
```

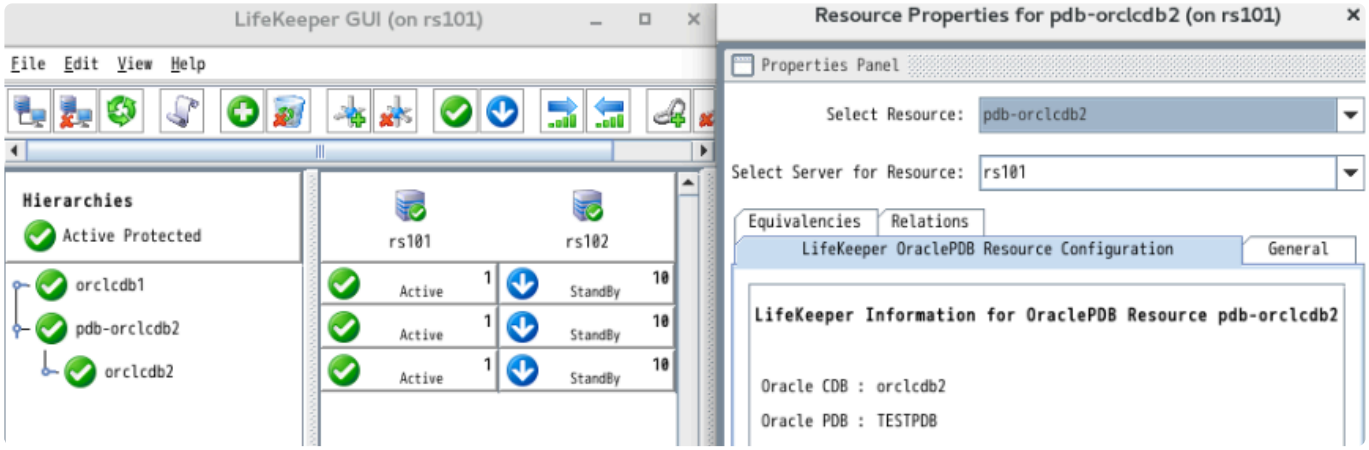
4. ターゲット CDB に接続し対象の PDB をプラグし、PDB を起動します。

```
SQL> CREATE PLUGGABLE DATABASE <PDB> USING '/home/oracle/.xml'
<COPY|NOCOPY>;

SQL> ALTER PLUGGABLE DATABASE <PDB> OPEN;
```

5. 移行後の PDB の情報が移行前と一致することを確認します。

```
SQL> COLUMN NAME FORMAT A8

     SELECT NAME, DBID, GUID FROM V$CONTAINERS WHERE NAME='<PDB>';
```

6. ターゲット CDB を指定して Oracle PDB リソースを作成します。
既に PDB を保護している Oracle PDB リソースがある場合は、リソース設定 [Change Protection PDB]（保護するPDBの変更 を参照）で対象の PDB を保護対象に追加して一つのリソースで管理することも可能です。


LifeKeeper GUI (on rs101)
File Edit View Help
Hierarchies
Active Protected
orclcdb1
pdb-orclcdb2
orclcdb2
rs101
rs102
Active 1
StandBy 10
Active 1
StandBy 10
Active 1
StandBy 10
Resource Properties for pdb-orclcdb2 (on rs101)
Properties Panel
Select Resource: pdb-orclcdb2
Select Server for Resource: rs101
Equivalencies
Relations
LifeKeeper OraclePDB Resource Configuration
General
LifeKeeper Information for OraclePDB Resource pdb-orclcdb2
Oracle CDB : orclcdb2
Oracle PDB : TESTPDB

# 6.13. PostgreSQL Recovery Kit 管理ガイド

## PostgreSQL Recovery Kit のドキュメンテーション

## 概要

SIOS Protection Suite for Linux PostgreSQL Recovery Kit は、SQLに準拠した、POSTGRESをベースとするオブジェクトリレーショナルデータベース管理システム (ORDBMS) です。PostgreSQL は誕生以来進化を続け、最も高度なオープンソースリレーショナルデータベース管理システムの 1 つになりました。

SPS for Linux PostgreSQL Recovery Kit は、LifeKeeper 内の PostgreSQL インスタンスを保護する機能を提供します。PostgreSQL ソフトウェア、LifeKeeper Core、および PostgreSQL Recovery Kit はクラスタ内の 2 台以上のサーバにインストールされます。PostgreSQL データベースインスタンスが LifeKeeper の保護下に入ったら、クライアントは LifeKeeper が保護する IP アドレスを使用してデータベースに接続します。LifeKeeper が保護する IP アドレスは別々に作成し、親 PostgreSQL リソースインスタンスと子 IP アドレスリソースの間に依存関係を手動で作成する必要があります。PostgreSQL サーバに障害が発生した場合、LifeKeeper はまずローカルサーバ上での復旧を試みます。ローカルリカバリに失敗した場合、バックアップサーバに切り替えます。

PostgreSQL リソース階層

## SIOS Protection Suite/LifeKeeper のドキュメンテーション

SIOS Technology Corp の Web サイトでは以下の SIOS Protection Suite 製品のドキュメンテーションを提供しています。

- SPS for Linux リリースノート
- SPS for Linux テクニカルドキュメンテーション
- Optional Recovery Kit ドキュメンテーション

## PostgreSQL のドキュメンテーション

*Administration Guide* 、 *User Guide* 、 *Reference Guide* を含む PostgreSQL のドキュメンテーションは以下の Web サイトで入手できます。

http://www.postgresql.org/docs

# 6.13.1. PostgreSQL リソース階層

以下の例は典型的な PostgreSQL リソース階層を示しています。

上記の例の依存関係は、以下の保護されるリソースに対応しています。

| リソース | **PostgreSQL** ソフトウェアコンポーネント |
|---|---|
| *LKIP.EXAMPLE.COM* | クライアント接続に使用される切り替え可能な IP アドレスを保護します。 |
| *var/lib/pgsql/data* | データベースデータディレクトリ (PGDATA) を保護します。 |
| *var/lib/pgsql/exec* | PostgreSQL サーバとクライアントの実行可能ファイルを保護します (実行可能ファイルが共有ファイルシステムにインストールされている場合)。 |
| *var/lib/pgsql/log* | データベースログファイルディレクトリを保護します (ログパスが共有ファイルシステム上にある場合)。 |
| *var/lib/pgsql/pg_xlog* | データベーストランザクションログディレクトリ (PGDATA/pg_xlog) を保護します。トランザクションディレクトリは Write-Ahead-Log ディレクトリとしても参照されます。 |
| *var/lib/pgsql/ socket_path* | データベースソケットディレクトリを保護します(ソケットパスが共有ファイルシステム上にある場合)。 |

フェイルオーバが発生した場合、LifeKeeper はバックアップサーバでファイルシステム、IP アドレス、およびデータベースリソース(すべてのリソース依存関係を含む)を In Service の状態にします。クライアントは切断され、サーバに再接続する必要があります。まだコミットされていない SQL ステートメントはすべて再入力する必要があります

# 6.13.2. PostgreSQLのハードウェアおよびソフトウェア要件

LifeKeeper for Linux PostgreSQL Recovery Kitをインストールする前に、LifeKeeper設定が次の要件を満たしている必要があります。LifeKeeperのハードウェアとソフトウェアのインストールと設定に関する具体的な手順については、 LifeKeeper for Linux インストールガイド を参照してください。

## ハードウェア要件

- **サーバー** — サーバーは、要件にしたがって設定する必要があります。要件は、 LifeKeeper for Linux テクニカルドキュメンテーション 、および LifeKeeper for Linux リリースノート に記載されています。
- **IPネットワークインターフェースカード** — 各サーバーは、イーサネットTCP/IPをサポートするネットワークインターフェースカードを1つ以上必要とします。ただし、LifeKeeperクラスターのベストプラクティスでは2つ以上のコミュニケーションパスが必要であることに注意してください。LANを使用した2つのコミュニケーションパスは、ハートビート用に別々のサブネットに接続することをお勧めします。また、少なくともどちらか1つのサブネットは、プライベートネットワークとして設定します。TCPとTTYのハートビートの組み合わせもサポートされています。

## ソフトウェア要件

- **TCP/IPソフトウェア** — LifeKeeper環境の各サーバーにはTCP/IPソフトウェアが必要です。
- **PostgreSQLまたはPostgreSQLベースのソフトウェア** — クラスター内の全サーバーに必ず同じバージョンのPostgreSQLまたはPostgreSQLをベースにしたソフトウェアをインストールします。サポートされるソフトウェアは、 サポートマトリックス の「サポート対象のアプリケーション」を参照ください。PostgreSQLソフトウェアは、 http://www.postgresql.org/download に記載されているいずれかのミラーサイトからダウンロードできます。
- **LifeKeeperソフトウェア** — クラスター内の各サーバーに同じバージョンのLifeKeeperソフトウェアとパッチをインストールする必要があります。具体的なLifeKeeperの要件については、 LifeKeeper for Linux テクニカルドキュメンテーション および LifeKeeper for Linux リリースノート を参照してください。
- **LifeKeeper for Linux PostgreSQL Recovery Kit** — LifeKeeperソフトウェアと同じバージョンのRecovery Kitを各サーバーにインストールする必要があります。

# 6.13.3. PostgreSQL 設定上の考慮事項

本セクションでは、PostgreSQL Recovery Kit の設定と管理を開始する前に考慮すべき事項について説明します。

PostgreSQL の保護: ベストプラクティス

DataKeeper でのミラーファイルシステムの使用

# 6.13.3.1. Postgre SQL の保護: ベストプラクティス

アクティブ / スタンバイ設定では、バックアップサーバは PostgreSQL をアクティブに実行せず、プライマリサーバに障害が発生した場合に備えてスタンバイします。アクティブ / アクティブ設定では、障害時、他のサーバに対するバックアップとして各サーバがアクティブに PostgreSQL インスタンスを実行します。以下に、アクティブ / スタンバイもしくはアクティブ / アクティブ設定で PostgreSQL リソースインスタンスを保護するときに従わなければならない具体的な要件を示します。

1. PostgreSQL の DataDir および Write-Ahead-LogPath (*PGDATA/pg_xlog*) を 1 つ以上の共有ファイルシステムにインストールしてください。DataDir および WAL-Path は、リソースインスタンスを保護するすべてのサーバ間で共有してください。

- ◦ PostgreSQL オペレーティングシステムユーザは、データディレクトリと、Write-Ahead-LogPath を含むディレクトリを所有している必要があります。

- ◦ PostgreSQL データベースは、`initdb` ユーティリティを使用して作成してください。`initdb` ユーティリティは、PostgreSQL オーナとして、-D <datadir> オプションを使用して実行する必要があります。

- ◦ デフォルトの PostgreSQL インスタンスの自動スタートアップは無効にするか、またはデフォルトの PostgreSQL インスタンスが LifeKeeper で使用するポート以外のポートで稼働するように制限する必要があります。

- ◦ LifeKeeper で保護する PostgreSQL インスタンスの自動スタートアップは無効にする必要があります。LifeKeeper は、保護するインスタンスの開始と停止を制御します。

- ◦ PostgreSQL インスタンスは、階層の作成に先だって手動で開始する必要があります。インスタンスは、`pg_ctl` ユーティリティにバックエンドオプション – o “ – p <port>” を指定して開始してください。

2. `StartupLogPath`、`SocketPath` 、`ExecutablePath` は、プライマリサーバの任意の共有ファイルシステム上か、各クラスタノードのローカルファイルシステム上に配置することができます。

- ◦ PostgreSQL オペレーティングシステムユーザはソケットパスを含むディレクトリを所有している必要があります。

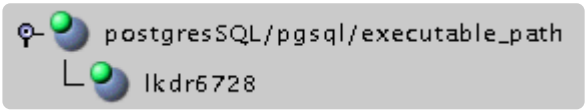
- ◦ PostgreSQL オペレーティングシステムユーザは StartupLogPath を含むディレクトリでの書き込み権限を所有している必要があります。

3. アクティブ / スタンバイもしくはアクティブ / アクティブのどちらかで複数のインスタンスを起動する場合は、各インスタンスに一意のポートおよびソケットパスを割り当ててください。

# 6.13.3.2. DataKeeperによるミラーファイルシステムの使用

PostgreSQL Recovery Kit では、SIOS DataKeeper を共有ファイルシステムとして使用できます。ミラーファイルシステムは、PostgreSQL インストールパス、ログパス、データディレクトリ、および実行可能パスに使用できます。

例えば、PostgreSQLリソースに依存するファイルシステムは以下のようになります。これは、データディレクトリーとその依存ファイルであるDataKeeperリソースミラーのファイルシステムを示しています。




レプリケーション (SIOS DataKeeper) ファイルシステムリソースを使用する場合は、PostgreSQLリソースを作成する前に作成する必要があります。

# 6.13.4. PostgreSQL インストール

## LifeKeeper での PostgreSQL のインストール / 設定

PostgreSQL データベースと LifeKeeper ソフトウェアのインストールと設定には以下の手順をお勧めします。各ステップは、詳細作業にリンクしています。

1. PostgreSQL ソフトウェアのインストール
2. PostgreSQL データベースの作成
3. LifeKeeper Core と PostgreSQL Recovery Kit のインストール
4. PostgreSQL リソースのための LifeKeeper 設定項目

これらの作業を実行すると、LifeKeeper リソース階層を作成して PostgreSQL データベースを保護できるようになります。

## リソース設定作業

前のセクションで説明したセットアップ作業を終了したら、PostgreSQL リソース階層を作成して拡張できるようになります。

LifeKeeper for Linux PostgreSQL Recovery Kit の設定に関して、以下の作業を行うことができます。

- Create Resource Hierarchy - PostgreSQL リソース階層を作成します。
- Delete Resource Hierarchy - PostgreSQL リソース階層を削除します。
- Extend Resource Hierarchy - PostgreSQL リソース階層をプライマリサーバからバックアップサーバに拡張します。
- Unextend Resource Hierarchy - LifeKeeper クラスタ内の 1 つのサーバから PostgreSQL リソース階層を拡張解除 (削除) します。
- Viewing PostgreSQL Configuration Settings - Resource Properties ダイアログを参照します。

LifeKeeper Core のリソース階層 (ファイルシステムや IP リソースなど) の設定方法については、LifeKeeper for Linux テクニカルドキュメンテーション の 管理作業 セクションを参照してください。

以下の作業については、 すべての Recovery Kit で同じ手順を使用する共通の作業であるため、LifeKeeper for Linux テクニカルドキュメンテーション の 管理 セクションを参照してください。

- Create a Resource Dependency - 既存のリソース階層と別のリソースインスタンスとの間に親子の依存関係を作成し、クラスタ内のすべての対象サーバに依存関係の変更を反映します。

- Delete a Resource Dependency - リソースの依存関係を削除して、クラスタ内のすべての対象サーバに依存関係の変更を反映します。
- In Service - リソース階層を特定のサーバで In Service にします。
- Out of Service - リソース階層を特定のサーバで Out of Service にします。
- View / Edit - 特定のサーバでリソース階層のプロパティを表示または編集します。

**注記:** このセクションでの設定作業は **[Edit]** メニューを使用して実行します。この設定作業の大半は次の方法でも実行できます。

- ツールバーから
- 状況表示の左側ペインにあるグローバルリソースを右クリックする
- 状況表示の右側ペインにあるリソースを右クリックする

右クリックの方法を使用すると、Edit メニューを使用するときに要求される情報を入力せずにすみます。

## アップグレード

PostgreSQL Recovery Kit の旧バージョンからのアップグレード

# 6.13.4.1. PostgreSQL ソフトウェアのインストール

同一のパラメータ / 設定を使用してクラスタ内の全サーバに PostgreSQL ソフトウェアをインストールする必要があります。詳細については PostgreSQL のサイト を参照してください。以下は、LifeKeeper を PostgreSQL と連携させるための追加の推奨事項および注意事項です。

- PostgreSQL クライアントソフトウェアパッケージがインストールされている必要があります。これらのパッケージには、PostgreSQL `psql` クライアントユーティリティが含まれていなければなりません。
- PostgreSQL サーバソフトウェアパッケージがインストールされている必要があります。これらのパッケージには、PostgreSQL `pg_ctl` および `initdb` ユーティリティが含まれていなければなりません。
- PostgreSQL クライアントパッケージとサーバパッケージのバージョンは、全サーバで同じでなければなりません。
- 全サーバに、PostgreSQL オペレーティングシステムユーザが以下のように存在している必要があります。
    - ° PostgreSQL オペレーティングシステムユーザを PostgreSQL ソフトウェアのインストールとサブディレクトリの所有者として指定します。
    - ° PostgreSQL オペレーティングシステムユーザには

`pg_ctl` ユーティリティを使用する権限が必要です。PostgreSQL オペレーティングシステムユーザは `pg_ctl` コマンドを使用してポストマスタインスタンスを開始および停止できなければなりません。

- ° PostgreSQL オペレーティングシステムユーザ名には英数字 (a-z,A-Z, 0-9) のみを使用する必要があります。
- ° PostgreSQL オペレーティングシステムユーザのユーザ ID とグループ ID は、全サーバで同じでなければなりません。

- PostgreSQL データベース管理者ユーザは、`psql` ユーティリティを使用した LifeKeeper クライアント接続のため、PostgreSQL のデータベース内に存在している必要があります。
    - ° また PostgreSQL データベース管理者ユーザは、データベース (

*template1* ) への接続権限を有している必要があり、またインスタンスに定義されたデータベースのリストを取得できる必要があります。

- ° この PostgreSQL データベース管理者ユーザは、システムテーブルの閲覧権限および一般的なクエリを作成できる権限を保有している必要があります。
- ° PostgreSQL データベース管理者ユーザは PostgreSQL オペレーティングシステムユーザと同じ名前を持つことができますが、異なるものです。

    - º 例:PostgreSQL オペレーティングシステムユーザ=postgres で PostgreSQL データベース管理者ユーザ=lkpostgres、もしくは、PostgreSQL オペレーティングシステムユーザ=postgres で PostgreSQL データベース管理者ユーザ=postgres。

- PostgreSQL サーバーデーモンは LifeKeeper が制御するので、システム起動時の自動起動を無効にする必要があります。

# 6.13.4.2. PostgreSQLデータベースの作成

データベースを作成するには、『 PostgreSQLのサイト 』の手順に従います。さらに次の推奨事項にも留意してください。

> レプリケーション（SIOS DataKeeper）ファイルシステムリソースを使用する場合は、PostgreSQLリソースを作成する前に作成する必要があります。

- PostgreSQLデータディレクトリーは、`initdb` ユーティリティを使用し、 *-D <data dir>* オプションを指定して初期化する必要があります。`initdb` コマンドはPostgreSQLオペレーティングシステムユーザーとして実行する必要があります。
- PostgreSQLインスタンスデータディレクトリーは共有ファイルシステムに存在している必要があります。
- PostgreSQLトランザクションログディレクトリーは共有ファイルシステム上に存在している必要があります。
- PostgreSQLデータベース名には英数字のみを使用する必要があります。
- データベースを作成したら、PostgreSQLデータベースインスタンスの自動スタートアップを無効にする必要があります。LifeKeeperの保護下に入ったら、LifeKeeperがデータベースの開始と停止を行います。
- PostgreSQLインスタンスは、階層の作成に先だって手動で開始する必要があります。インスタンスは、`pg_ctl` ユーティリティーにバックエンドオプション *-o “ -p <port>”* を指定して開始してください。

## パスワードの保護を無効にする場合（インスタンスへの接続に、パスワード保護を使用しない）

- PostgreSQLデータベースインスタンスがパスワード保護されない場合や、PostgreSQLの管理ユーザーからローカルクライアント接続にパスワードを必要としない場合は、エントリーがローカルトラスト接続を許可した状態で存在している必要があります。以下はPostgreSQLのデータベース管理ユーザーのローカルクライアント接続を有効にするために *pg_hba.conf* に記述するエントリーのサンプルです。

```
====================
.
.
Local all postgres trust
.
.
======================
```

## パスワード保護を有効にする場合（インスタンスへの接続にパスワードを使用）

- パスワードで保護されるデータベースインスタンスは、リソースが保護されるクラスターの各サーバー上の *.pgpass* 認証ファイルに存在するためのPostgreSQLデータベース管理ユーザーに対するパスワードエントリーが必要になります。 *.pgpass* ファイルはパスワードを要求する各PostgreSQLデータベース管理ユーザーに対して有効でテスト済みのエントリーを保有している必要があります。
- *.pgpass* ファイルはPostgreSQLオペレーティングシステムユーザーのホームディレクトリーに配置してください。ファイルへのアクセス権を制限するために適切な権限を設定してください。
- 以下は、*<hostname>:<port>:<database>:<user>:<password>*のフォーマットで示した有効な *.pgpass* ファイルの一例です。

```
==================
*:5443:*:lifekeeper:jh43tmp2009
====================
```

> 注意： *.pgpass* ファイルは、ユーティリティーpsqlによる自動の（非ターミナルまたはスクリプト）接続のために必要です。 *.pgpass* ファイルは、パスワード保護されたインスタンスが保護される各サーバー上に存在している必要があります。

## PowerGres Plusで透過的データ暗号化（TDE）を使用する場合

透過的データ暗号化を使用すると、暗号化されたデータにアクセスするためにはキーストアをオープンする必要があります。quickCheckによる監視のため、クラスター内の各サーバーでキーストアの自動オープンを有効にしてください。

# 6.13.4.3. LifeKeeper ソフトウェアのインストール

PostgreSQL ソフトウェアをインストールし、データベースを作成したら、LifeKeeper Core ソフトウェアと必要なパッチをインストールし、続いて PostgreSQL Recovery Kit をインストールできます。

LifeKeeper パッケージのインストール方法については、SPS for Linux インストールガイド を参照してください。

# 6.13.4.4. PostgreSQL リソースのための LifeKeeper 設定項目

PostgreSQL Recovery Kit には、特定のシナリオでリソース保護をカスタマイズできるように、設定項目のための環境変数が用意されています。これらの変数の値を変更するには、ファイル */etc/default/LifeKeeper* を編集します。新しい設定を反映するためにプロセスを再起動する必要はありません。デフォルト値は PostgreSQL Recovery Kit がインストールされるほとんどの環境で動作する値となっています。

## ·LKPGSQL_CONN_RETRIES

この設定項目は、データベースが開始するまで PostgreSQL Recovery Kit が待機する時間を制御します。この時間は、「(LKPGSQL_CONN_RETRIES * 5) = データベースインスタンスの開始まで待機する合計時間(秒単位)」の式を用いて Recovery Kit により計算されます。この変数の設定は、リソースの In Service 要求とリソースのローカルリカバリの両方に影響します。

## ·LKPGSQL_DISCONNECT_CLIENT

この設定項目は、postmaster がクラッシュした場合にアクティブなクライアントが切断されるかどうかを制御します。値を 1 (true) に設定すると、アクティブなクライアントはリソースのローカルリカバリが進行している間は切断されます。値を 0 (false) に設定すると、アクティブなクライアントはリソースのローカルリカバリが進行している間も切断されません。この変数は、リソースのローカルリカバリイベントのみに影響し、postmaster プロセスが稼働していない場合のローカルリカバリイベントの場合にのみ適用されます。

## ·LKPGSQL_SDIRS

この設定項目は、PostgreSQL データベースをシャットダウンする際のクライアント切断動作を制御します。カンマで区切られた設定項目をデフォルトファイルに追加する必要があります。このオプションを設定することにより、指定されたリソースインスタンスもしくは保護されるデータディレクトリと一致するインスタンスが、クライアントに対してシャットダウン時の切断を要求しません。

```
LKPGSQL_SDIRS=/protected/pgsql-datadir

LKPGSQL_SDIRS=/protected/pgsql-datadir,/otherprotected/pgsql-datadir
```

*/protected/pgsql-datadir* および */otherprotected/pgsql-datadir* は、LifeKeeper の保護下の PostgreSQL データディレクトリです。

**注記:** このオプション LKPGSQL_SDIRS および LKPGSQL_IDIRS は排他的なオプションです。LKPGSQL_SDIRS もしくは LKPGSQL_IDIRS の値は、階層の作成中に選択した保護対象の datadir の値と一致していなければなりません。

## ·LKPGSQL_IDIRS

この設定項目は、PostgreSQL データベースをシャットダウンする際のクライアント切断動作を制御します。カンマで区切られた設定項目をデフォルトファイルに追加する必要があります。このオプションを設定することにより、指定されたリソースインスタンスもしくは保護されるデータディレクトリと一致するインスタンスが、クライアントに対してシャットダウン時の即時切断を強制します。

```
LKPGSQL_IDIRS=/protected/pgsql-datadir
```

```
LKPGSQL_IDIRS=/protected/pgsql-datadir,/otherprotected/pgsql-datadir
```

*/protected/pgsql-datadir* および */otherprotected/pgsql-datadir* は、LifeKeeper の保護下の PostgreSQL データディレクトリです。

**注記:** このオプション LKPGSQL_SDIRS および LKPGSQL_IDIRS は排他的なオプションです。LKPGSQL_SDIRS もしくは LKPGSQL_IDIRS の値は、階層の作成中に選択した保護対象の datadir の値と一致していなければなりません。

# 6.13.4.5. PostgreSQL リソース階層の作成

プライマリーサーバーで次の手順を実行します。

1. **[Edit]** メニューの **[Server]** から **[Create Resource Hierarchy]** を選択します。

   **[Create Resource Wizard]** ダイアログボックスが表示されます。

2. ドロップダウンリストから **[PostgreSQL Database]** を選択し、 **[Next]** をクリックします。.

3. 以下の情報を入力するように要求されます。ダイアログボックスで **[Back]** ボタンが有効な場合は、前のダイアログボックスに戻ることができます。これは、エラーが発生して、前に入力した情報を修正する必要がある場合に便利な機能です。いつでも **[Cancel]** をクリックして、作成処理全体を取り消すことができます。

| フィールド | ヒント |
|---|---|
| **Switchback Type** | **[intelligent]** または **[automatic]** を選択します。フェイルオーバー後、バックアップサーバーで PostgreSQL リソースが In Service (アクティブ) になるとき、プライマリーサーバーにスイッチバックされる方法が決まります。インテリジェントスイッチバック (intelligent) の場合、リソースをプライマリーサーバーにスイッチバックするときに管理者の介入が必要ですが、自動スイッチバック (automatic) の場合は、プライマリーサーバーがオンラインに戻り、LifeKeeper のコミュニケーションパスが再確立されるとすぐにスイッチバックが実行されます。<br><br>**注記:** スイッチバック方法は、PostgreSQL リソースと依存関係を構築しているリソースのスイッチバック方法と一致する必要があります。 |
| **PostgreSQL Executable Location** | このフィールドは、PostgreSQL 実行可能ファイルを含むディレクトリパスを指定するために使用されます。パス名に対する有効な文字は、アルファベット、数字および以下の特別な記号になります。- _ ./ |
| **PostgreSQL Client Executable Location** | このフィールドは、PostgreSQL クライアント実行可能ファイルのフルパスを指定します。 デフォルトは {PostgreSQL 実行可能ファイルの場所}/psql です。 パス名に使用できる有効な文字は、文字、数字、および次の特殊文字です。- _ ./ |
| **PostgreSQL Administration Executable Location** | このフィールドは、PostgreSQL 管理実行可能ファイルの完全パスを指定します。 デフォルトは {PostgreSQL 実行可能ファイルの場所}/pg_ctl です。 パス名に使用できる有効な文字は、文字、数字、および次の特殊文字です。- _ ./ |
| **PostgreSQL Data Directory** | このフィールドは、LifeKeeper の保護下に置かれる PostgreSQL データディレクトリ (datadir) の場所を指定するために使用されます。指定されたディレクトリは、共有ファイルシステムに存在し、常駐している必要があります。パス名に対する有効な文字は、アルファベット、数字および以下の特別な記号になります。- _ ./ |
| **PostgreSQL Port** | このフィールドは、ポストマスタデーモンがクライアントアプリケーションからの接続をリスニングする TCP/IP ポート番号を指定するために使用されます。 |

| | |
|---|---|
| **PostgreSQL Socket Path** | このフィールドは、ポストマスタデーモンがクライアントアプリケーションからの接続をリスニングする UNIX ドメインソケットへのフルパスを指定するために使用されます。パス名に対する有効な文字は、アルファベット、数字および以下の特別な記号になります。- _ ./ |
| **PostgreSQL Database Administrator User** | このフィールドは、指定したデータベースインスタンスとそのインスタンスへの接続および管理者権限に対する PostgreSQL データベース管理ユーザ名を指定するために使用されます。 |
| **PostgreSQL Logfile** | このフィールドは、PostgreSQL の起動や停止に使用する `pg_ctl` の -l オプションで使われるログファイルパスを指定するために使用されます。 |
| **PostgreSQL Database Tag** | プライマリーサーバー上の新しい PostgreSQL データベースリソースに対する固有のタグ名です。デフォルトのタグ名は「pgsql」の後にデータベースインスタンスのポート番号を付けたものです。別の固有のタグ名を入力することもできます。タグ名に対する有効な文字は、アルファベット、数字および以下の特別な記号になります。- _ ./ |

4. **[Create]** をクリックします。 **[Create Resource Wizard]** によって、PostgreSQL リソース階層が作成されます。LifeKeeper によって入力データが検査されます。問題が検出された場合、情報ボックスにエラーメッセージが表示されます。

5. PostgreSQL リソース階層が正常に作成されたこと、フェイルオーバーによる保護を実現するにはその階層をクラスタ内の別のサーバーに拡張する必要があることを示すメッセージが表示されます。**[Next]** をクリックします。

6. **[Continue]** をクリックします。 **[Pre-extend Wizard]** が起動されます。リソース階層を別のサーバーに拡張する方法の詳細については、PostgreSQL リソース階層の拡張 の手順 2 を参照してください。

# 6.13.4.6. PostgreSQL リソース階層の削除

LifeKeeper 設定内のすべてのサーバから PostgreSQL リソース階層を削除するには、次の手順を実行します。

1. **[Edit]** メニューの **[Resource]** から **[Delete Resource Hierarchy]** を選択します。
2. **[Target Server]** で、PostgreSQL リソース階層を削除するサーバの名前を選択します。

   **注記:** 左側ペインのグローバルリソースまたは右側ペインの個々のリソースインスタンスを右クリックして **[Delete Resource]** 作業を選択した場合、このダイアログボックスは表示されません。
3. **[Hierarchy to Delete]** を選択します。(このダイアログは、右ペインまたは左ペインのリソースインスタンス上で右クリックして **[Delete Resource]** を選択した場合には表示されません。) **[Next]** をクリックします。
4. 選択したターゲットサーバと、削除の対象として選択した階層を確認する情報ボックスが表示されます。 **[Next]** をクリックします。
5. PostgreSQL リソースが正常に削除されたことを確認する別の情報ボックスが表示されます。
6. **[Done]** をクリックして終了します。

# 6.13.4.7. PostgreSQL リソース階層の拡張

この操作は **[Edit]** メニューから開始できます。または **[Create Resource Hierarchy]** の処理を完了すると自動的に開始されます。その場合は、下記の **手順 2** を参照してください。

1. **[Edit]** メニューの **[Resource]** から **[Extend Resource Hierarchy]** を選択します。 ***[Pre-Extend Wizard]*** が表示されます。 **拡張** 操作に慣れていない場合は、 **[Next]** をクリックします。 LifeKeeper のリソース階層の拡張作業のデフォルト値が分かっていて、入力と確認を省略する場合は、 **[Accept Defaults]** をクリックします。
2. ***[Pre-Extend Wizard]*** で以下の情報を入力します。

   **注記:** 最初の 2 つのフィールドは **[Edit]** メニューの **[Extend]** から操作を開始した場合にだけ表示されます。

| フィールド | ヒント |
|---|---|
| Template Server | PostgreSQL リソースが現在 In Service のサーバを選択します。 |
| Tag to Extend | 拡張する PostgreSQL リソースを選択します。 |
| Target Server | 拡張先のサーバを入力または選択します。 |
| Switchback Type | フェイルオーバ後、バックアップサーバで PostgreSQL リソースが In Service (アクティブ) になるとき、プライマリサーバにスイッチバックされる方法が決まります。 **[intelligent]** または **[automatic]** を選択できます。スイッチバックタイプは、必要な場合 **[Resource Properties]** ダイアログボックスの **[General]** タブで後から変更できます。<br><br>**注記:** スイッチバック方法は、PostgreSQL リソースと依存関係を構築しているリソースのスイッチバック方法と一致する必要があることに注意してください。 |
| Template Priority | **テンプレートの優先順位** を選択または入力します。これはサーバで現在 In Servic の PostgreSQL 階層の優先順位です。優先順位は、1 ~ 999 の範囲で未使用の値が有効で、小さい数字ほど優先順位が高くなります(数字 1 が最高の優先順位に相当します)。拡張処理時に、別のシステムですでに使用中の優先順位をこの階層に対して指定することはできません。デフォルト値を推奨します。<br><br>**注記:** このフィールドは階層を最初に拡張するときだけ表示されます。 |
| Target Priority | これは、別サーバにある同等の階層に対する、新しく拡張する PostgreSQL 階層の優先順位です。1 ~ 999 の範囲の未使用の値が有効で、リソースのカスケーディングフェイルオーバシーケンスにおけるサーバの優先順位を示します。LifeKeeper のデフォルトでは、階層が作成されたサーバに「1」が割り当てられることに注意してください。優先順位は連続している必要はありませんが、特定のリソースについて 2 つのサーバに同じ優先順位を割り当てることはできません。 |

3. 拡張前のチェックが正常に終了したというメッセージが表示されたら、 **[Next]** をクリックします。

4. 拡張する階層に応じて、拡張されるリソースタグ(一部編集不可) を示す一連の情報ボックスが表示されます。

5. ***[Extend Wizard]*** で以下の情報を入力します。

| フィールド | ヒント |
|---|---|
| PostgreSQL Executable Location | このフィールドは、PostgreSQL 実行可能ファイルを含むディレクトリパスを指定するために使用されます。パス名に対する有効な文字は、アルファベット、数字および以下の特別な記号になります。- _ ./ |
| PostgreSQL Database Tag | プライマリサーバ上の新しい PostgreSQL データベースリソースに対する固有のタグ名です。デフォルトのタグ名は語「pgsql」の後にデータベースインスタンスのポート番号を付けたものです。別の固有のタグ名を入力することもできます。タグ名に対する有効な文字は、アルファベット、数字および以下の特別な記号になります。- _ ./ |

6. 「Hierarchy extend operations completed」というメッセージが表示されたら、階層を別のサーバに拡張する場合は、 **[Next Server]** をクリックし、これ以上拡張作業が必要ない場合は、 **[Finish]** をクリックします。

7. 「Hierarchy Verification Finished」のメッセージが表示されたら、 **[Done]** をクリックします。

# 6.13.4.8. PostgreSQL リソース階層の拡張解除

LifeKeeper クラスタ内にある 1 つのサーバからリソース階層を削除するには、次の手順を実行します。

1. **[Edit]** メニューの **[Resource]** から **[Unextend Resource Hierarchy]** を選択します。

2. **[Target Server]** で、PostgreSQL リソースを拡張解除するサーバを選択します。リソースが現在 In Service のサーバは選択できません。(このダイアログボックスは、右側のペインでリソースインスタンスを右クリックして、拡張解除作業を選択した場合には、表示されません。) **[Next]** をクリックします。

3. 拡張解除する PostgreSQL 階層を選択し、 **[Next]** をクリックします。(このダイアログボックスは、いずれかのペインでリソースインスタンスを右クリックして、拡張解除作業を選択した場合には、表示されません。)

4. ターゲットサーバと拡張解除の対象として選択した PostgreSQL リソース階層を確認する情報ボックスが表示されます。 **[Unextend]** をクリックします。

5. PostgreSQL リソースが正常に拡張解除されたことを確認する別の情報ボックスが表示されます。 **[Done]** をクリックして、 **[Unextend Resource Hierarchy]** メニューを終了します。

# 6.13.4.9. PostgreSQL 設定の参照

リソースプロパティダイアログは [Edit] メニューもしくはリソースコンテキストメニューから参照可能です。このダイアログはサーバ上の特定のリソースに対するプロパティを表示します。 **[Edit]** メニューからアクセスした場合は、リソースおよびサーバを選択することができます。 **リソースコンテキスト** メニューからアクセスする場合はサーバを選択することができます。

**[Configuration]** タブからは以下の PostgreSQL 設定を参照することができます。

- Executable Path
- Client Executable Name
- Admin Executable Name
- Bind Setting
- Startup Log Location
- PostgreSQL Operating System User Name
- PostgreSQL Database Administrator User
- Version Number
- Data Directory
- Socket Location
- Port Number
- OS Daemon Name

# 6.13.4.10. PostgreSQL Recovery Kit の旧バージョンからのアップグレード

旧バージョンの SPS for Linux の PostgreSQL ソフトウェアからのアップグレード時、既存の SPS PostgreSQL リソースのインスタンスは変更されます。SPS ソフトウェアがサーバ上でアップデートされる際に、次の保持される値が、SPS 内部の情報のフィールドに自動的に追加されます。

- **Client Executable Location ( *psql* )** -保護対象のデータベースインスタンスへの接続に使用される *psql* もしくは同等のクライアントユーティリティの場所。アップグレード後、この値は LifeKeeper GUI のプロパティ表示で確認することが可能です。また、この値は LifeKeeper コマンドラインの `set_value` ユーティリティを使用して確認することも可能です。

  `set_value` は、LifeKeeper PostgreSQL Recovery Kit が内部のリソース情報の項目の値をアップデートするために提供される LifeKeeper ユーティリティの名前です。このユーティリティの使用は本トピックで説明されている問題、もしくは SIOSTechnology Corp. 社のサポートチームによる依頼および指示に限るものとします。

  **注記**：`set_value`ユーティリティは、正確なエラーチェックは行わないので、一般的な使用には向いていません。

- **Administration Executable Location (*pg_ctl*)** - 保護対象のデータベースインスタンスの起動、停止、ステータス確認に使用する *pg_ctl* または同等の管理ユーティリティの場所です。アップグレード後、この値は LifeKeeper GUI のプロパティ表示で確認することが可能です。また、この値は LifeKeeper コマンドラインの set_value ユーティリティを使用して確認することも可能です。

- **PostgreSQL Database Administrator User** - LifeKeeper が保護するインスタンスの PostgreSQL データベース管理者ユーザです。このユーザは保護対象のデータベースインスタンスへの接続および管理権限を持っている必要があります。デフォルト値は、PostgreSQL のデータディレクトリを保持している PostgreSQL オペレーティングシステムユーザになります。アップグレード後、この値は LifeKeeper GUI のプロパティ表示で確認することが可能です。また、この値は SPS コマンドラインの set_value ユーティリティを使用して確認することも可能です。

- **PostgreSQL Daemon Name (postmaster)** - バックエンドで起動しているデーモン名です。この値は、データベースの最初のステータス確認時に決定されます。デフォルト値は postmaster です。アップグレード後、この値は LifeKeeper GUI のプロパティ表示で確認することが可能です。また、この値は LifeKeeper コマンドラインの set_value ユーティリティを使用して確認することも可能です。

- **Default Test Database (template1)** - データベースインスタンスが基本的な接続確認を行っている間に LifeKeeper が使用するデータベースです。アップグレード後、デフォルトのテストデータベースは **template1** に設定されます。

- **PostgreSQL Maximum Monitoring Hangs (`LKPGSQL_QCKHANG_MAX`)** - この設定は、リストアやリカバリのためのフェイルオーバアクションが開始される前に、無制限な数のコネクションがハングすることからの保護を提供します。PostgreSQL Recovery Kit の監視の一部は、保護されたデータベースへの接続を必要とします。どれだけの数のコネクションのハングを許容するかは、リソー

ス作成時に **LKPGSQL_QCKHANG_MAX** によって決定されます。8.1.2 以前のバージョンでは、デフォルトの値は 15 でした。バージョン 8.1.2 （またはそれ以降のバージョン）へのアップグレード後は、デフォルトの値は 2 です。この値はリソース作成時にリソース情報と紐付けて保存されますので、手動でアップデートしない限り、バージョン 8.1.2 にアップグレードする前に作成されたリソースはいずれもデフォルト値 15 のままになります。バージョン 8.1.2 （またはそれ以降のバージョン）で作成されたリソースはデフォルト値の 2 となります。値は set_value ユーティリティ を使用して確認することもできます。

重要な注記: SPS for Linux PostgreSQL Recovery Kit ソフトウェアをアップグレードした後、プライマリサーバからバックアップサーバへリソースインスタンスのフェイルオーバをシュミレーションするために 手動でスイッチオーバを行い PostgreSQL リソース階層のテストを行ってください。

| | |
|---|---|
| アップグレードに関する重要事項 | アップグレードを行った後、リソースを起動できない場合は、以下の事項を確認してください:<br><br>**Client Executable name is not found or incorrect (クライアントの実行可能ファイル名が見つからない、または不正です)**<br><br>この値は set_value ユーティリティを使用してアップデートできます。クライアントの実行可能ファイルをアップデートする構文は以下の通りです。<br><br>• /opt/LifeKeeper/lkadm/subsys/database/pgsql/bin/set_value <tag> 'clientexe' <full path to the psql utility><br><br>• 例: /opt/LifeKeeper/lkadm/subsys/database/pgsql/bin/set_value pgsql-5443 'clientexe' '/pgsql/clientutils/psql'<br><br>• **Administration Executable name is not found or incorrect (管理実行可能ファイル名が見つからない、または不正です)**<br><br>• この値は set_value ユーティリティを使用してアップデートできます。管理実行可能ファイルをアップデートする構文は以下の通りです。<br><br>• /opt/LifeKeeper/lkadm/subsys/database/pgsql/bin/set_value <tag> 'osexe' <full path to the pg_ctl utility> |

| | |
|---|---|
| | • 例: `/opt/LifeKeeper/lkadm/subsys/database/pgsql/bin/set_value pgsql-5443 'osexe' '/pgsql/adminutils/pg_ctl'` |
| | 複数のハング･イベントからの復旧までのインターバルの短縮 ( LKPGSQL_QCKHANG_MAX )<br><br>**8.1.2 以前のバージョンでは、Maximum Monitoring Hangs の値は大きすぎました。**<br><br>既存の PostgreSQL リソースインスタンスの **Maximum Monitoring Hangs** の値は、set_value ユーティリティを使用して見る、またはセットすることができます。<br><br>**Maximum Monitoring Hangs** ( LKPGSQL_QCKHANG_MAX ) の値をセットする構文は以下の通りです。<br>• `/opt/LifeKeeper/lkadm/subsys/database/pgsql/bin/set_value <tag> 'hangmax' <number>.`<br><br>• 例: `/opt/LifeKeeper/lkadm/subsys/database/pgsql/bin/set_value pgsql-5443 'hangmax' 3.`<br><br>• 注記: クラスタ内の全てのノードで値を更新するためには、`-c` オプションを付与してください。(`set_value -c <tag>…`).<br><br>この値を **確認する** 構文は以下の通りです。<br>• `/opt/LifeKeeper/lkadm/subsys/database/pgsql/bin/set_value -l <tag> 'hangmax'`<br><br>• **例:** `/opt/LifeKeeper/lkadm/subsys/database/pgsql/bin/set_value -l pgsql-5443 'hangmax'` |

# 6.13.5. 管理

## データベース管理者ユーザの更新

ユーザの更新 オプションにより、LifeKeeper の管理者は、LifeKeeper PostgreSQL リソースインスタンスに対する現在の PostgreSQL データベース管理者ユーザを変更することができます。

データベース管理者ユーザの更新

## PostgreSQL リソース階層のテスト

PostgreSQL リソース階層のテストは、手動でサーバをスイッチオーバすることで、プライマリサーバからバックアップサーバへのリソースインスタンスのフェイルオーバをシミュレートして実行できます。

LifeKeeper GUI からの手動スイッチオーバの実行

## EnterpriseDB Postgres Plus Advanced Server 環境

EnterpriseDB Postgres Plus Advanced Server の保護

## FUJITSU Software Symfoware/Enterprise Postgres 環境

Symfoware Server / Enterprise Postgres の保護

# 6.13.5.1. LifeKeeper GUI からの手動スイッチオーバの実行

LifeKeeper GUI で **[Edit] > [Resource] > [In Service]** を選択すると、手動スイッチオーバを開始できます。例えば、バックアップサーバで [In Service] 要求 が実行されると、PostgreSQL リソース階層がバックアップサーバ側で In Service になり、プライマリサーバ側では Out of Service になります。この時点で、元のバックアップサーバがプライマリサーバに、元のプライマリサーバがバックアップサーバに変わります。

[Out of Service] 要求を実行した場合、リソース階層は Out of Service になりますが、別のサーバで In Service にはなりません。

**重要:** バックアップサーバでリソース階層を In Service にしたら、データベースに接続できるかどうか試す必要があります。パスワード保護されたインスタンスに関しては *.pgpass* ファイルがバックアップ上で有効であることが重要です。 *.pgpass* ファイルが有効であるか確認するためには、データベースへのクライアント接続が `psql` ユーティリティと PostgreSQL データベース管理者ユーザの両方を使用して実行できることを確認してください。パスワード認証を要求されなければ、接続が有効な *.pgpass* ファイルが存在していることが分かります。

# 6.13.5.2. EnterpriseDB Postgres Plus Advanced Server の保護

EnterpriseDB Postgres Plus Advanced Server リソースを保護するにあたって、追加の LifeKeeper 設定は必要ありません。

| 問題 | 解決方法 |
| --- | --- |
| EnterpriseDB Postgres Plus Advanced Server のインストール中に、 **[Configuration Mode]** ダイアログでオプションの **[PostgreSQL-compatible defaults and samples]** を選択した場合は、LifeKeeper が使用する「edb」データベースが作成されません。 | 「createdb」ユーティリティを使用して「*edb* 」データベースを手動で追加してください。<br><br>`createdb -p <port> -h <socket path> edb` コマンドは PostgreSQL オペレーティングシステムユーザとして実行する必要があります。以下はサンプルです。<br><br>`su - postgres`<br><br>`postgres@server1 ~>createdb -p 5435 -h /var/lib/postgres edb` |

# 6.13.5.3. Symfoware Server/Enterprise Postgres の保護

Symfoware Server/Enterprise Postgres を保護するにあたって、サポート範囲及びご利用の環境について以下に解説します。

| サポート範囲 |
| --- |
| • PostgreSQL と互換性のある機能をサポートします。<br>• ミラーリング機能はサポートされません。ミラーを作成しデータに冗長性を持たせたい場合は、DataKeeper をご利用ください。<br>• Symfoware Server の Native インターフェースはサポートされません。Open インターフェース (Symfoware 12.2)、Postgres (Symfoware 12.3 以降) をご利用ください。<br>• WebAdmin は非対応です。詳細は Fujitsu Software の Symfoware Server クラスタ運用ガイドをご参照ください。<br>• 以下の機能はサポートされません。<br>◦ WAL の2重化<br>◦ 暗号化機能（格納データの暗号化）<br>◦ データ秘匿化<br>◦ 並列検索<br>◦ インメモリ機能 |

| 構成上の注意 |
| --- |
| DB 管理ユーザー (OS ユーザー) のログイン時の環境設定ファイル (.bash_profile 等) に、Symfoware のコマンド (pg_ctl,psql 等) の実行に必要な環境変数 LD_LIBRARY_PATH を設定してください。 |

# 6.13.5.4. データベース管理者ユーザの更新

**ユーザの更新** オプションは、保護されたリソースが存在する全システム上の PostgreSQL データベース管理者ユーザに対して保存された値を更新します。 **[Update User]** オプションは **LifeKeeper のリソースツールバー** もしくは **LifeKeeper のリソースコンテキストメニュー** から実行することが可能です。

PostgreSQL データベース管理者ユーザを更新するためには、プライマリサーバで以下の手順を実行してください。

注記 : **[Update User]** メニューおよびツールバーオプションは out of service のリソースに対して無効になります。

1. ツールバーの **[Update User]** アイコンを選択するか、リソースコンテキストメニューから [Update User] を選択してください。

   **[Update User Wizard]** のダイアログが表示されます。

2. 以下の情報を入力するように要求されます。ダイアログボックスで **[Back]** ボタンが有効な場合は、前のダイアログボックスに戻ることができます。これは、エラーが発生して、前に入力した情報を修正する必要がある場合に便利な機能です。いつでも **[Cancel]** をクリックして、作成処理全体を取り消すことができます。

| フィールド | ヒント |
|---|---|
| **[PostgreSQL Database Administrator User]** と入力 | このダイアログは、指定したデータベースインスタンスとそのインスタンスへの接続および管理者権限に対する PostgreSQL データベース管理者ユーザ名を指定するために使用されます。<br><br>**注記:** 認証スクリプトは、指定した値を使用して接続確認を行います。パスワード保護されたインスタンスは PostgreSQL データベース管理者ユーザの *.pgpass* ファイルの有効なエントリを要求します。 |
| **Confirm Update Action** | このダイアログでは、以前のユーザから新しいユーザに対して行う変更について確認をします。 |

3. **[Update]** をクリックしてください。現在保護されているリソースがあるすべてのサーバ上で PostgreSQL データベース管理者ユーザが更新されます。

# 6.13.6. PostgreSQL トラブルシューティング

本セクションでは、LifeKeeper PostgreSQL リソース階層の作成時や拡張時、またはリソースの削除時やリストア時に表示される可能性のあるメッセージの一覧を示します。必要に応じて、エラーの原因およびエラー状態を解消するために必要な処置についても説明しています。

他の LifeKeeper コンポーネントからメッセージが出されることもあります。そのような場合は、メッセージカタログ を参照してください。メッセージカタログには、操作、管理、GUI など、LifeKeeper for Linux を使用しているときに遭遇する可能性がある、すべてのエラーコードが列挙されています。また、エラーコードの原因に関する追加の説明や、問題解決のために必要な処置についても、必要に応じて記載されています。この完全なリストを検索すると、受信したエラーコードを見つけることができます。また、関連する LifeKeeper コンポーネントの個別のメッセージカタログに直接アクセスすることもできます。

- 全般項目
- 設定項目

# 6.13.6.1. PostgreSQL 全般項目

以下のエラーメッセージおよび状態は PostgreSQL Recovery Kit 使用時に表示される可能性があります。

| エラー | 解決方法 |
|---|---|
| Unable to protect PostgreSQL database using the same port as another LK protected PostgreSQL database. | PostgreSQL のこのバージョンに *postgresql.conf* ファイルが含まれていることを確認してください。 *postgresql.conf* ファイルで、エントリ listen_address= に、データベースインスタンスで使用される IP アドレスを設定してください。<br><br>**注記:** *postgresql.conf* ファイルの listen_address= のフォーマットは重要です。構文エラーにより、データベースサーバの起動に失敗することもあります。 |
| Unable to perform a manual switchover of version 8.X when clients are connected. | デフォルトの (スマート) シャットダウンオプションがスイッチオーバでクライアントの切断に失敗しました。接続されたクライアントにより、シャットダウンに失敗し続けてしまう場合は、LKPGSQL_SDIRS が設定されていないかどうかを確認してください。障害が継続するようであれば、LifeKeeper の設定項目 LKPGSQL_IDIRS を設定してください。 |
| Unable to connect from a remote client to the database server. | PostgreSQL のリモートホストログインを有効にするには、 *PostgreSQL Administration Guide* の *pg_hba.conf* ファイルの設定に関する説明を参照してください。 |
| psql: connectDBStart() — connect() failed: No such file or directory. Is the postmaster running at 'localhost' and accepting connections on Unix socket '<port>'?" | ソケットファイルが存在し、インスタンスが現在実行されていることを確認します。ソケットファイルが */tmp* にある場合、 */tmp* ディレクトリをクリーンアップする cron ジョブによって削除された可能性があります。リソースを out of service にしてから、 in service に戻します。次に、PostgreSQL ソケットファイルを残すように cron ジョブを修正します。 |
| PostgreSQL resource hierarchy fails to come in service but the database is running. | データベースが、指定された期間内に LifeKeeper クライアント要求に応答できなかった可能性があります。 */etc/default/LifeKeeper* の設定項目 LKPGSQL_CONN_RETRIES を調整して、PostgreSQL データベースインスタンスのリカバリと再起動に許容される秒数を増やします。 |
| PostgreSQL resource hierarchy fails local recovery following a postmaster crash with active client connections. | 多数のアクティブクライアントが PostgreSQL に接続しているとき、データベースはクライアント接続が終了するまで正しく再起動できない場合があります。この場合、ローカルリカバリが正常に行われるよう強制的にクライアント接続を終了させることをお勧めします。<br>LKPGSQL_DISCONNECT_CLIENT 変数を */etc/default/LifeKeeper* に設定して、この場合の PostgreSQL リソース階層の動作を制御することができます。値を 1 ( **true** ) に設定すると、クライアントプロセスに SIGTERM シグナルが送信され、データベースから強制的に切断されます。この処置は、ポストマスタプロセスがローカルリカバリ中に稼働していない場合のみに取ることができます。 |

| | |
|---|---|
| Unable to connect to database with error “WARNING: password file “*/home/<osuser>/.pgpass*” has world or group read access” | *.pgpass* ファイルの権限を、u=rw(0600) にしてください。 *.pgpass* ファイルの権限および所有者を変更してください。 |
| FATAL: syntax error in file “*/<datadir>/postgresql.conf*” line 50, near token “.17” | *postgresql.conf* ファイルの listen_address= のエントリが正しい引用を含んでいません。エントリが正しいかどうか、またはエントリが正しい引用を含んでいるかを確認してください。 |

# 6.13.6.2. PostgreSQL 設定項目

| 設定項目 | 機能 |
|---|---|
| LKPGSQL_KILLPID_TIME | プロセス ID が停止した後、そのプロセスに対する再チェックを行うまでの時間 |
| LKPGSQL_CONN_RETRIES | 旧 LKPGSQLMAXCOUNT – 操作 (開始もしくは停止) を行った後、クライアント接続を試みる回数 |
| LKPGSQL_ACTION_RETRIES | アクションコマンドに失敗するまで、開始と停止を試行する回数 |
| LKPGSQL_STATUS_TIME | status コマンドのタイムアウト値 |
| LKPGSQL_QCKHANG_MAX | データベースインスタンスがフェールオーバ / sendevent を発生させるまでに許容される、クイックチェックがハングする回数 |
| LKPGSQL_CUSTOM_DAEMON | postgres デーモン (デフォルト postmaster) に対する追加の別名の指定を許可します。 |
| LKPGSQL_IDIRS | 旧 LKPGSQL_IPORTS – immediate オプションのみを使用してシャットダウンするインスタンスに対する datadir のエントリを含みます。 |
| LKPGSQL_SDIRS | Smart オプションを使用してシャットダウンするインスタンスに対する datadir のエントリを含みます。 |
| LKPGSQL_DISCONNECT_CLIENT | データベースに障害が発生している間の PostgreSQL リソース階層の振る舞いを制御します。値を 1 **(true)** に設定すると、クライアントプロセスに SIGTERM シグナルが送信され、データベースから強制的に切断されます。この処置は、postmasterプロセスがローカルリカバリ中に稼働していない場合のみに取ることができます。<br><br>※注:PostgreSQL 8.2以降では、本パラメータをご利用いただけません。 |
| LKPGSQL_DISCONNECT_CLIENT_BYTAG | LKPGSQL_DISCONNECT_CLIENT と類似していますが、この設定は、処置を、この設定項目で指定したカンマで区切られたタグのリストに限定します。<br><br>※注:PostgreSQL 8.2以降では、本パラメータをご利用いただけません。 |
| LKPGSQL_RESUME_PROC | プロセスの停止状態が検出されたときに (プロセス状態が T であるとき) 再開するか無視するかを決めます。 |
| LKPGSQLDEBUG | PostgreSQL database kit および postgres データベースのデバッグを有効にします。0 ~ 5 の値が有効です。数字が大きくなるほどデバッグ情報が増えます。 |

| | |
|---|---|
| | この設定項目は、オプション -d <LKPGSQLDEBUG> を使用して postmaster データベースへと渡されます。 |

# 6.14. Postfix Recovery Kit 管理ガイド

> **重要** ：SIOS は次の Recovery Kit のサポートを終了します。これらは LifeKeeper for Linux の次のメジャーリリースでサポートされなくなります。
> (なお、新規受注は2022年12月31日をもって終了しています。)
> • NEC iStorage StoragePathSavior
> • Samba
> • Postfix
> • HULFT HUB Recovery Kit

Postfix はさまざまな役割を演じます。 そのすべてが、電子メールの適切な流れには不可欠なものです。 Postfix はメールの着信がないかネットワークをリスニングし、メールを別のサーバに送信し、ローカルメールをローカルプログラムに配信します。

LifeKeeper for Linux Postfix Recovery Kit は、LifeKeeper 環境で障害の発生したプライマリーサーバからバックアップサーバーに Postfix を回復する仕組みを提供します。LifeKeeper もPostfix も、 大幅な時間の損失や人手の介入なしにフェイルオーバ処理を通じてデータの整合性を保証します。

## ドキュメントの内容

本書は以下のトピックについて説明しています。

- ドキュメンテーションとリファレンス ：LifeKeeper for Linux のドキュメントと所在の一覧です。
- 要件 ：Postfix Recovery Kit を適切に設定、導入、運用するうえで必要なハードウェアとソフトウェアについて説明しています 。LifeKeeper for Linux ソフトウェアの具体的なインストール / アンインストール方法については、LifeKeeper for Linux インストールガイド を参照してください。
- LifeKeeper for Linux Postfix Recovery Kit の設定 ：Postfix Recovery Kit を適切に設定するうえで必要な手順について説明しています。
- Postfix の 構成確認 ：Postfix のリソース階層を構成する前に、Postfix の構成確認方法を 記述 します。
- LifeKeeper 設定作業 ：LifeKeeper GUI を使用して Postfix リソース階層を作成し管理する作業について説明しています。
- スプール (メールボックス) の依存関係作成 ：Postfix リソースと Mailbox Spool ファイル･システムリソースの間で手動で依存関係をつくる方法を記述します。
- リソース階層のテスト ：LifeKeeper GUI とコマンドラインインターフェースを使用して、Postfix リソース階層をテストするためのステップを記述します。
- トラブルシューティング ：LifeKeeper for Linux のエラーメッセージの一覧とその説明です。

# ドキュメンテーションとリファレンス

以下に、SIOS Technology Corp が発行している LifeKeeper 関連情報の一覧を示します。

- LifeKeeper for Linux リリースノート
- LifeKeeper for Linux テクニカルドキュメンテーション
- LifeKeeper for Linux インストールガイド
- Optional Recovery Kit ドキュメンテーション

このドキュメンテーション、および他の LifeKeeper Recovery Kit に関連するドキュメンテーションは、以下の Web サイトで入手できます。

http://jpdocs.us.sios.com

# Postfix のドキュメンテーションとリファレンス

以下は、Postfix アプリケーションおよび LifeKeeper Postfix Recovery Kit に関連する参考資料の一覧です。

- Postfix マニュアルページ
- Red Hat Postfix リファレンスマニュアル

# 6.14.1. Postfix ハードウェア要件とソフトウェア要件

LifeKeeper for Linux Postfix Recovery Kit をインストールする前に、LifeKeeper の構成が次の要件を満たしている必要があります。LifeKeeper for Linux ハードウェアおよびソフトウェアの設定に関する具体的な手順については、LifeKeeper for Linux インストールガイド を参照してください。

## ハードウェア要件

- サーバ ： 製品媒体に付属する LifeKeeper for Linux リリースノート および LifeKeeper for Linux インストールガイド (製品メディアに収録) に記載の要件に従って設定されたLifeKeeper for Linux 対応サーバ。

- データストレージ ： Postfix Recovery Kit は共有ストレージ、LifeKeeper DataReplication が提供するデータレプリケーションストレージのどちらもデータストレージとして使用する事ができます。

## ソフトウェア要件

- **TCP/IP** ソフトウェア ：LifeKeeper 環境の各サーバには TCP/IP ソフトウェアが必須です。

- **LifeKeeper** ソフトウェア ：クラスタの各サーバにインストールした LifeKeeper for Linux ソフトウェアのバージョン、および適用した LifeKeeper for Linux ソフトウェア・パッチのバージョンが同一であることが絶対条件です。

- **LifeKeeper for Linux IP Recovery Kit** ：同じバージョンのRecovery Kit を各サーバにインストールする必要があります。

- **IP** ネットワークインタフェース ： 各サーバには、イーサーネットTCP/IP をサポート するネットワークインタフェースが1 つ以上必要です。IP 切り替えが正しく機能するためには、ローカルネットワークに接続したユーザシステムが標準TCP/IP 仕様に準拠している必要があります。

  **注記：** 各サーバに必要なネットワークインタフェースが1 つだけであっても、異種メディア要件、スループット要件、シングルポイントオブフェイラの排除、ネットワークのセグメンテーションといった多くの理由で、複数のインタフェースを使用すべきです。

- **Postfix** ソフトウェア ： 各サーバには、LifeKeeper とPostfix Recovery Kit を設定する前に、PostfixソフトウェアのLinuxディストリビューション版をインストールして設定しておく必要があります。各サーバには同じバージョンをインストールする必要があります。最新リリースの互換性および発注情報については、SPS for Linux リリースノート を参照するか販売代理店にお問い合わせください。

  LifeKeeper for Linux ソフトウェアの具体的なインストール / アンインストール方法については、LifeKeeper for Linux インストールガイド を参照してください。

# 6.14.1.1. Postfix Recovery Kit のインストール

LifeKeeper for Linux ソフトウェアの具体的なインストール / アンインストール方法については、
LifeKeeper for Linux インストールガイド を参照してください。

# 6.14.2. LifeKeeper for Linux Postfix Recovery Kit の設定

本セクションでは、LifeKeeper for Linux Postfix Recovery Kit の設定の詳細について説明します。Postfix Recovery Kit の設定と管理を開始する前に考慮すべき事項についても記載しています。LifeKeeper Core リソース階層の設定方法については、SPS for Linux テクニカルドキュメンテーション を参照してください。

# 6.14.2.1. Postfix の保護対象

Postfix Recovery Kit では以下のオブジェクトを保護対象とします。

- Postfix サービスデーモン
- Postfix のネットワークソケット

  仮想 IP アドレスを一つまたは複数作成し Postfix の設定ファイルで指定します。all を指定した場合はローカルループバックアドレスを死活監視に使用します。また、 Postfix の smtpd プロセスが使用する SMTP および SMTPS ポートは 25(465) をサポートします。

- Postfixが利用するキュー領域

  スプール(メールボックス)領域がキュー領域と別のファイルシステム上であり、そちらも合わせて保護する場合は別途ファイル･システムリソースを作成し、Postfix Recovery Kitで作成したリソース階層との依存関係を作成する必要があります。こちらは、スプール(メールボックス)の依存関係作成を参照してください。

# 6.14.2.2. Postfix の設定要件

- SMTP サービスが使用する IP アドレスを指定している場合、その IP アドレスは LifeKeeper で保護している仮想 IP アドレスである必要があります。
- main.cf

  以下の項目はPostfix Recovery Kit から参照されるため指定する必要があります。

  - mail_owner
  - setgid_group
  - daemon_directory
  - command_directory
  - process_id_directory
  - inet_interfaces

    監視する仮想 IP アドレスを指定します。1 つまたは複数を指定できます。すべての IP アドレスを指定する場合は、“all” を使用します。
  - queue_directory
  - mail_spool_directory
- master.cf

  以下の項目は必須で指定する必要があります。

  - smtpd を起動するための、smtp サービスエントリ
- directory 値のディレクトリは共有ストレージ上にある必要があります。そのディレクトリのファイルシステムは LifeKeeper により保護(ファイルシステムリソース) されている必要があります。
- メールボックスに共有されるデータが格納される場合、mail_spool_directory 値のディレクトリも共有ストレージ上にある必要があります。
- postfix オーナーID は全てのクラスタサーバ上で一致している必要があります。
- postdrop(setgid group) のグループID は全てのクラスタサーバ上で一致している必要があります。
- PostfixサービスはLifeKeeperが制御するので、システム起動時の自動起動を無効にする必要があります。

# 6.14.2.3. ポート定義とPostfix Recovery Kit

Postfix Recovery Kit はPostfix 設定ファイルのmaster.cf 内のエントリでSMTP サービス プロ セスで指定されたポートをリスニングします。ポートがサービス名(たとえば smtp) として指定 されている場合は、/etc/services ファイルのポート番号が調べられます。smtp であれば25 番ポ ート、smtps であれば465 ポートが使用されます。

```
smtp      inet  n       -       n       -       -       smtpd
```

# 6.14.2.4. DNS、Postfix とLifeKeeper

DNSは、メール配信用のバックアップ(代替) ホストを指定する仕組み(MX レコード) を備えています。この仕組みによって、ホストはnullクライアントのように、メールを受信するように設定されていない別のホストのメール処理を代行することもできます。 MXレコードは、すべてのメールをハブマシンまたはメールサーバに強制的に送信する仕組みも可能にします。 MXレコードでは、あるドメイン名のメール交換者(つまり、指定されたホスト名のメールを処理、または転送するホスト)が指定されます。これは、たとえばDNSサーバに次のような項目を追加することによって行います。

```
himalaya.sc.steeleye.com IN     MX      10  relay.steeleye.com.
```

この例では、サーバhimalaya.sc.steeleye.comは、このサーバのメールをrelay.steeleye.comに配信することを指定したMXレコードを持っています。 LifeKeeperで保護するサーバはMXレコードを持っていてはいけません。代わりに、Postfixリソース階層作成時に使用されるLifeKeeperが保護する仮想IPアドレスをMXレコード用として使用する必要があります。

# 6.14.2.5. 構成例

## 構成 1：アクティブ／スタンバイ設定の例

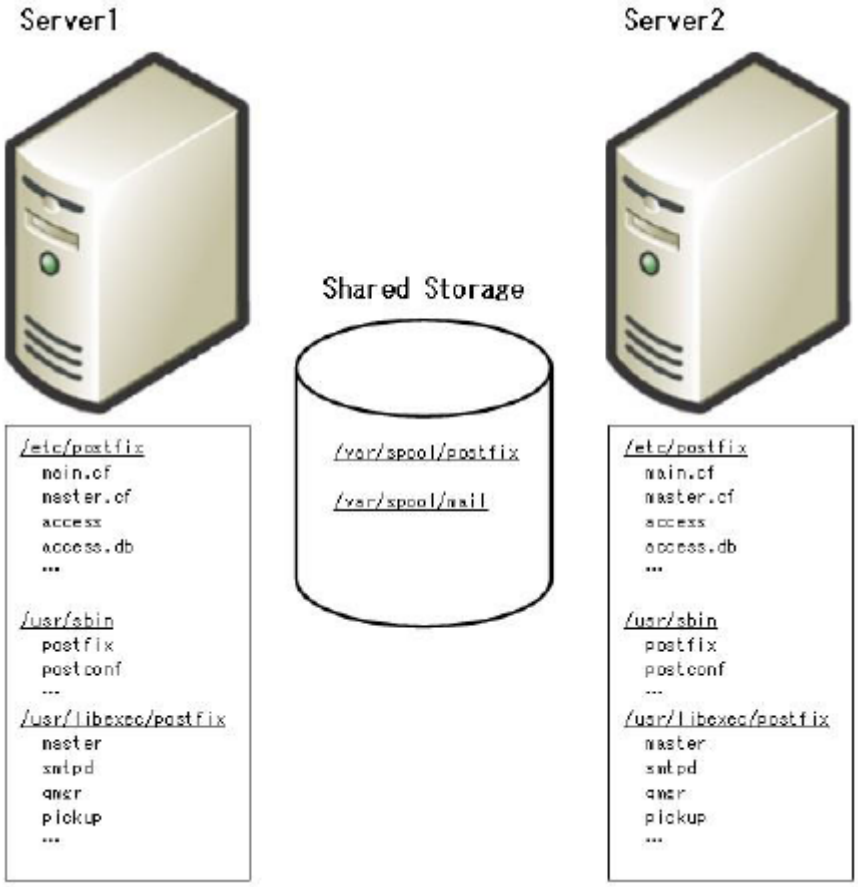



**図1: 典型的なLifeKeeper アクティブ/スタンバイPostfix 環境 1**

- Postfix の設定ファイルは個々のサーバ内に設定
- Postfix の実行ファイルおよび、デーモンプログラムは個々のサーバ内に設定
- キュー領域 /var/spool/postfix は共有ストレージ上に設定
- スプール(メールボックス)領域 /var/spool/mail は共有ストレージ上に設定

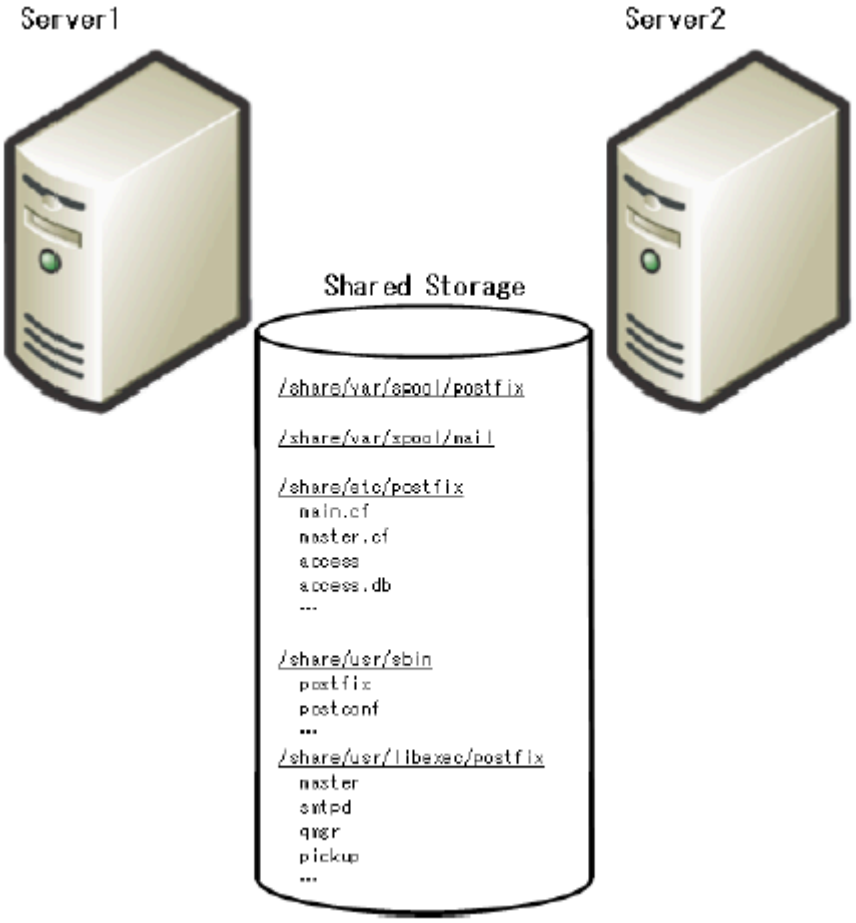



**図2: 典型的なLifeKeeper アクティブ/スタンバイPostfix 環境 2**

- Postfix の設定ファイルは共有ストレージ上に設定
- Postfix の実行ファイルおよび、デーモンプログラムは共有ストレージ上に設定
- キュー領域 /var/spool/postfix は共有ストレージ上に設定
- スプール(メールボックス)領域 /var/spool/mail は共有ストレージ上に設定

# 構成 2：アクティブ／アクティブ設定の例

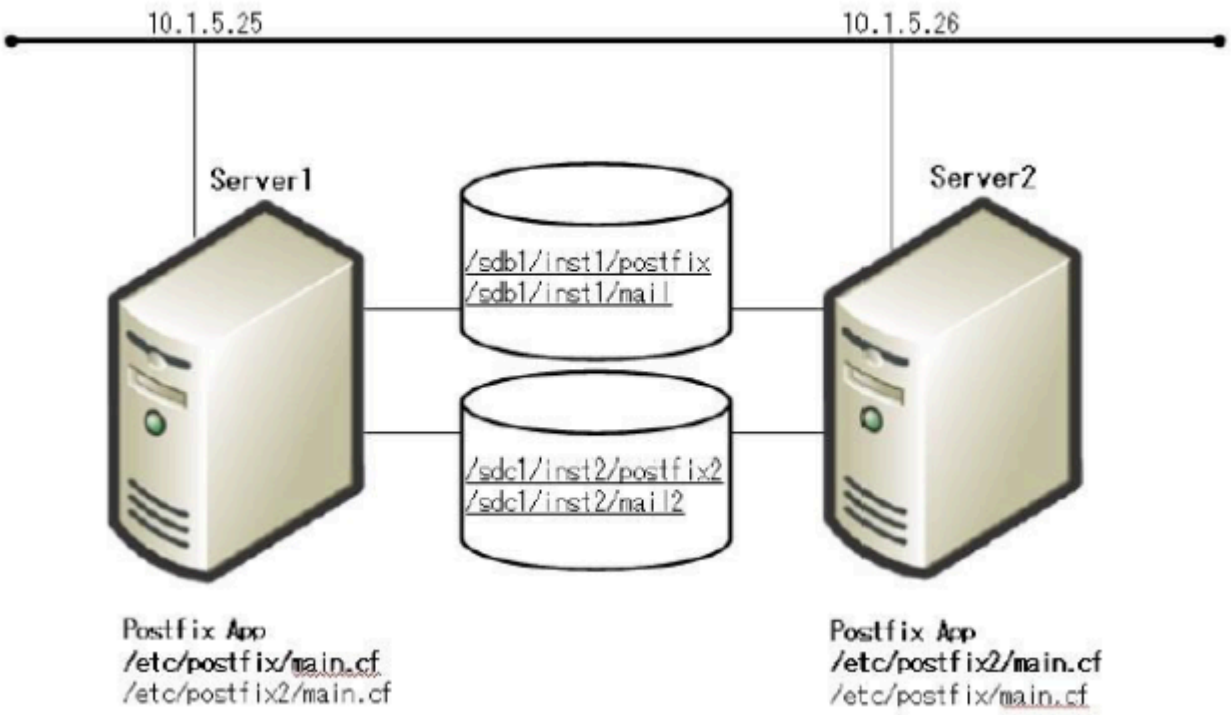



**図3: 典型的なLifeKeeper アクティブ/アクティブPostfix 環境 3**

[インスタンス1 はServer1 上でアクティブ]

Postfixの設定ファイルパス：/etc/postfix

Postfix の実行ファイルパス：/usr/sbin

キュー領域：/sdb1/inst1/postfix

メールスプール領域：/sdb1/inst1/mail

<main.cf>

inet_interfaces = 10.1.5.25, localhost

[インスタンス2 はServer2 上でアクティブ]

Postfix の設定ファイルパス：/etc/postfix2

Postfix の実行ファイルパス：/usr/sbin

キュー領域：/sdc1/inst2/postfix2

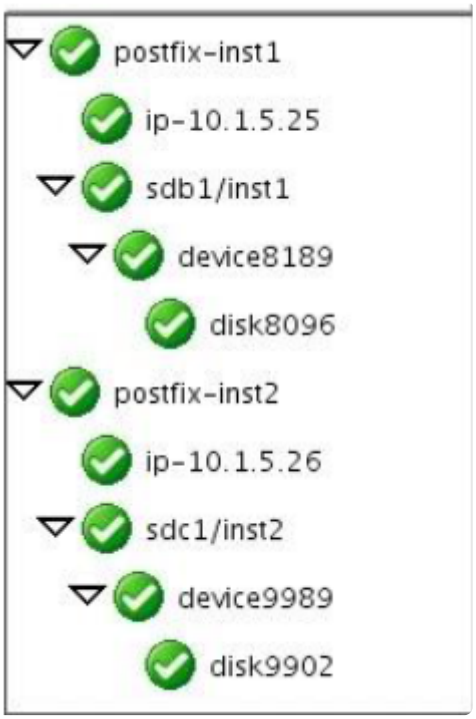
メールスプール領域：/sdc1/inst2/mail2

<main.cf>

inet_interfaces = 10.1.5.26

alternate_config_directories = /etc/postfix2

LifeKeeper GUIで Postfix リソースの階層構造は下記のようになります。



# 6.14.3. Postfix の構成確認

このセクションでは、Postfix Recovery Kit でのリソース作成前、つまりクラスタシステムを構築する前に、単体システムにおける確認作業の実施方法を構成例 1 を例に示します。

## Postfix の構成確認手順

1. Postfix の設定

   Postfix の設定ファイルは各サーバにあります。

```
main.cf (extract)

daemon_directory = /usr/libexec/postfix
command_directory = /usr/sbin
process_id_directory = pid
inet_interfaces = localhost , 192.168.0.10
mail_spool_directory = /var/spool/mail
queue_directory = /var/spool/postfix
```

```
master.cf (extract)

smtp      inet  n       -       n       -       -       smptd
```

2. SMTP サービスが待ち受ける仮想IP アドレスを起動

   Postfix のSMTP サービスが待ち受ける仮想IP アドレスを起動します。 LifeKeeper によりIP リソースを作成するか、以下のように”ifconfig”コマンドに より手動で起動します。

```
# ifconfig eth0:1 192.168.0.10 netmask 255.255.255.0 up
```

3. キュー領域のための共有ファイルシステムをマウント

```
# mkfs.ext3 /dev/sda1
# mount -t ext3 /dev/sda1 /mnt/queue
# mkdir -p /mnt/queue/postfix
```

```
# cp -rp /var/spool/postfix/* /mnt/queue/postfix/
# mv /var/spool/postfix /var/spool/postfix.org
# ln -s /mnt/queue/postfix /var/spool/postfix
# postfix check
```

4. メールスプール領域のための共有ファイルシステムをマウント

```
# mkfs.ext3 /dev/sdb1
# mv /var/spool/mail /var/spool/mail.org
# mkdir -p /var/spool/mail
# mount -t ext3 /dev/sdb1 /var/spool/mail
```

5. Postfix の起動

```
# postfix –c /etc/postfix start
postfix/postfix-script: starting the Postfix mail system
```

6. プロセスとソケットの確認

```
# netstat -pltun | grep master
tcp      0   0   127.0.0.1:25        0.0.0.0:*        LISTEN        15931/master
tcp      0   0   192.168.0.10:25     0.0.0.0:*        LISTEN        15931/master

# ps -ef | grep -v grep | grep postfix
root      15931      1  0  16:11  ?  00:00:00  /usr/libexec/postfix/master
postfix   15932  15931  0  16:11  ?  00:00:00  pickup -l -t fifo -u
postfix   15933  15931  0  16:11  ?  00:00:00  qmgr -l -t fifo -u
```

7. Postfix の停止

```
# postfix -c /etc/postfix stop
postfix/postfix-script: stopping the Postfix mail system
```

5～7 の操作により、正しく起動ができなかった場合は Postfix のエラーメッセージ等を参照し、問題を解決してから再度同じ作業を実施して問題がなくなるまで確認します。また、この操作を全てのクラスタシステムのサーバ上で実施し、確認をします。

# 6.14.3.1. Postfix のLifeKeeper 設定作業

LifeKeeper for Linux Postfix Recovery Kit の管理作業はすべて、LifeKeeper グラフィカル・ユーザ・インタフェース（GUI）で実行できます。LifeKeeper GUI では、Postfix リソースの設定、管理、監視作業をガイド付きで行えます。

LifeKeeper for Linux Postfix Recovery Kit の設定に関しては、次の作業が可能です。

- リソース階層の作成 ：Postfix リソース階層を作成します。
- リソース階層の拡張 ：Postfix リソース階層をプライマリ・サーバからバックアップ・サーバに拡張します。
- リソース階層の拡張解除 ：LifeKeeper クラスタの 1 つのサーバの Postfix リソース階層を拡張解除（削除）します。
- リソース階層の削除 ：Postfix リソース階層を削除します。
- 従属関係の作成 ：既存のリソース階層と別のリソース・インスタンスとの間に子従属関係を作成し、クラスタ内のすべての対象サーバに従属関係の変化を通知します。
- 従属関係の削除 ：リソースの従属関係を削除して、クラスタ内のすべての対象サーバ に従属関係の変化を通知します。
- In Service ：リソース階層をアクティブにします。
- Out of Service ：リソース階層を非アクティブにします。
- 表示 / 編集 ： リソース階層のプロパティを表示または編集します。

**注記：** このセクションの残りの部分では、設定作業はすべて **[Edit]** メニューから実行することになっています。この設定作業の大半は次の方法でも実行できます。

1. ツールバーから状態表示の左ペインにあるグローバルリソースを右クリック。
2. 状態表示の右ペインにあるリソースインスタンスを右クリック。

   ※右クリックの方法を使用すると、 **[Edit]** メニューを使用したときに必要な情報を入力しないですみます。

# 6.14.3.1.1. Postfix リソース階層の作成

必要なセットアップ作業を完了したあと、Postfix リソース階層を作成するために、以下の手 順を行います。

**重要:** Postfix リソース・インスタンスを作成する前に、仮想 IP アドレスを LifeKeeper の保護下に置く必要があります。

1. **[LifeKeeper GUI]** メニューから **[編集]** を選択し、次に **[サーバ]** を選択します。 ドロップダウンメニューから、 **[リソース階層の作成]** を選択します。

   **[リソース作成ウィザード]** ダイアログ・ボックスが表示され、クラスタにインストールされている全ての Recovery Kit がリストボックスに表示されます。

2. **[Postfix Mail Server]** を選択します。

3. 次の表で説明している情報を入力します。 どのダイアログ・ボックスでも、 **[戻る]** がアクティブになっているときには、直前のダイアログ・ボックスに戻ることができます。 これは、エラーが発生して、前に入力した情報を修正する必要がある場合に便利な機能です。 いつでも **[キャンセル]** をクリックして、作成処理全体を取り消すことができます。

| フィールド | ヒント |
|---|---|
| Switchback Type | [インテリジェント]または[自動]を選択します。これによって、バックアップサーバへのフェイルオーバ後、Postfix インスタンスがこのサー バでサービスに戻ったときにどのようにスイッ チバックされるかが決まります。スイッチバック・タイプは、必要な場合、[リソース・プロパティ]ダイアログ・ボックスの[一般]タブで後から変更できます。<br><br>**注記：** スイッチバックの方法は、Postfix リソースで使用するプリントサーバ、IP リソース、フ ァイル・システムリソースのスイッチバック方 法と一致している必要があります。一致していない場合、Postfix リソース作成時に、Postfix リソースに対して選択された設定に一致するように設定がリセットされます。 |
| Server | 階層を作成するサーバを選択します。 |
| Postfix Binary Location | Postfix デーモンが置かれてディレクトリパス名を入力してください。 |
| Postfix server Config File Location | Postfix の設定ファイル(main.cf) が置いてあるディレクトリを入力します。 |
| Queue Root | Postfix キューディレクトリのディレクトリパス名を入力してください。 |

| | |
|---|---|
| Directory | デフォルトは前のウィザードで指定した構成ファイルから決定されます。<br><br>Postfix キューディレクトリは共有ディスクの上になければなりません。Postfix キューディレクトリがシンボリックリンクであるならば、ダイアログ･ボックスはディレクトリの最初の場所を指しているシンボリックリンクのルートディレクトリを示します。 |
| Root Tag | LifeKeeper が示すデフォルト･ルート･タグを選択するか、このサーバ上のリソース･インスタンスの重複しない名前を入力します。デフォルトは postfix-on-<キューディレクトリパス>です。英字、数字、および特殊文字（- _ . /）を使用できます。 |

4. **[作成]** をクリックします。**[リソース作成ウィザード]** が表示され、Postfix リソース階層が作成されます。LifeKeeper によって入力データが検証されます。問題が検出された場合、情報ボックスにエラーメッセージが表示されます。

5. 情報ボックスが表示され、Postfix リソース階層の作成に成功したこと、フェイルオーバによって保護するには、クラスタ内の別のサーバにリソース階層を拡張する必要があることを示すメッセージが表示されます。**[次へ]** をクリックします。

6. **[続行]** をクリックします。**[事前拡張ウィザード]** が起動されます。リソース階層を別のサーバに拡張する方法の詳細については、次の「Postfix リソース階層の拡張 」の手順2 を参照してください。

# 6.14.3.1.2. Postfix リソース階層の拡張

この操作は **[編集]** メニューから開始できます。また、**[リソース階層の作成]** の処理を完了すると自動的に開始されます。その場合は、手順2を参照してください。

1. **[編集]** メニューの **[リソース]** から **[リソース階層の拡張]** を選択します。**[事前拡張ウィザード]** が表示されます。拡張操作に慣れていない場合は、**[次へ]** をクリックします。

2. **[事前拡張ウィザード]** で次の情報を入力します。

   **注記：** 最初の 2 つのフィールドは、**[編集]** メニューの **[拡張]** から操作を開始した場合にだけ表示されます。

| Field | Tips |
|---|---|
| Template Server | Postfix リソースが現在サービス中のサーバを入力します。 |
| Tag to Extend | 拡張するPostfix リソースを選択します。 |
| Target Server | 拡張先のサーバを入力または選択します。 |
| Switchback Type | **[インテリジェント]** または **[自動]** を選択します。スイッチバック・タイプは、必要な場合、**[リソース・プロパティ]** ダイアログ・ボックスの **[一般]** タブで後から変更できます。<br><br>**注記：** スイッチバック方法は、Postfix リソースが使用する従属リソースのスイッチバック方法と一致する必要があることを忘れないでください。 |
| Template Priority | テンプレート階層の優先順位を選択または入力します。優先順位は1～999 の範囲の未使用の値が有効で、数値が小さいほど優先順位は高くなります（1は最高の優先順位を表わします）。 拡張操作時には、別のシステムで既に使用中の優先順位をこの階層に対して指定することはできません。デフォルト値を推奨します。<br><br>**注記** ：これは階層を最初に拡張するときだけ表示されます。 |
| Target Priority | ターゲット・サーバの階層の優先順位を選択、または入力します。 |

3. 拡張前のチェックが成功したというメッセージが表示されたら、**[次へ]** をクリックします。

4. LifeKeeper はPostfix リソースを拡張するための一連のダイアログ・ボックスを示します。そして、その幾つかは編集することはできません。

| Field | Tips |
|---|---|
| Root Tag | ターゲット・サーバの新しいPostfix リソース・インスタンスのデフォルト・タグ名が表示されます。デフォルト・タグ名はテンプレート・サーバ上のこのリソースのタグ名と同じです。新しいタグ名を入力する場合は、ターゲット・サーバ上で重複しない名前にします。英字、数字、および特殊文字（- _ . /）を使用できます。 |
| Binary Directory (Information Only) | このダイアログ・ボックスは、情報の目的だけです。ボックスに現れるBinary Directory を変えることができません。 |
| Configuration Directory (Information Only) | このダイアログ・ボックスは、情報の目的だけです。ボックスに現れるConfiguration Directory を変えることができません。 |

5. 「Hierarchy extend operations completed (階層の拡張操作が完了しました)」という メッセージが表示されたら、階層を別のサーバに拡張する場合は、**[次のサーバ]** を クリックし、拡張作業が必要ない場合は、**[完了]** をクリックします。

6. 「Hierarchy Verification Finished (階層の検査が完了しました)」というメッセージ が表示されたら、**[完了]** をクリックします。

# 6.14.3.1.3. Postfix リソース階層の拡張解除

LifeKeeper クラスタの1 つのサーバのリソース階層を削除するには、次の手順を実行します。

1. **[編集]** メニューの **[リソース]** から **[リソース階層の拡張解除]** を選択します。
2. Postfix リソースを拡張解除する **[ターゲット・サーバ]** を選択します。リソースが現在サービス中の Postfix サーバは選択できません。（このダイアログ・ボックスは、右側のペ インでリソース・インスタンスを右クリックして、**[拡張解除]** を選択した場合には、表示されません。）**[次へ]** をクリックします。

3. 拡張解除するPostfix 階層を選択し、**[次へ]** をクリックします。（このダイアログ・ボックスは、いずれかのペインでリソース・インスタンスを右クリックして、**[拡張解除]** を選択した場合には、表示されません。）

4. 拡張解除するために選択したターゲット・サーバとPostfix リソース階層を確認する情報ボックスが表示されます。**[拡張解除]** をクリックします。

5. Postfix リソースの拡張解除に成功したことを示す別の情報ボックスが表示されます。**[完了]** をクリックして、**[リソース階層の拡張解除]** メニューを終了します。

# 6.14.3.1.4. Postfix リソース階層の削除

Postfix 階層を削除すると、従属関係および保護されているサービスに何が起きるかを理解しておくことが重要です。

- **依存関係:** Postfixリソース階層を削除するとき、Postfixリソースだけは削除されます。従属するIPとファイルシステム資源は削除されません。

- **保護されたサービス:** Postfixリソースを削除する前にPostfixリソースを停止した場合、リソースを削除した後Postfixデーモンは停止します。 Postfixリソースの動作中にリソースを削除した場合、Postfixデーモンは動き続けます。リソース階層の削除をした後はLifeKeeperの保護対象となりません。

LifeKeeper環境内のすべてのサーバからリソース階層を削除するには、以下の手順を実行します。

1. **[編集]** メニューの **[リソース]** から **[リソース階層の削除]** を選択します。

2. Postfixリソース階層を削除する **[ター ゲット・サーバ]** を選択し、**[次へ]** をクリックします。（このダイアログ・ボックスは、いずれかのペインでリソース・インスタンスを右クリックして、**[リソースの削除]** を選択した場合には、表示されません。）

3. **[削除する階層]** を選択します。このダイアログ・ボックスは、左側または右側のペインでリソース・インスタンスを右クリックして、**[リソースの削除]** を選択した場合には、表示されません。）**[次へ]** をクリックします。

4. 選択したターゲット・サーバと、削除の対象として選択した階層を確認する情報ボックスが表示されます。**[削除]** をクリックします。

5. Postfixリソースの削除に成功したことを示す別の情報ボックスが表示されます。

6. **[完了]** クリックして終了します。

# 6.14.3.1.5. スプール (メールボックス) の依存関係作成

Postfix のキューディレクトリとスプール(メールボックスディレクトリが共有ストレージの同じファイルシステム(LUN)上である場合、Postfix リソース階層の作成、Postfix リソース階層の拡張により両ディレクトリは保護対象となります。スプール(メールボックス)ディレクトリが共有ストレージの別のファイルシステム(LUN)上である場合、 スプール(メール ボックス) ディレクトリ配下のファイルシステムを LifeKeeper により保護対象とし、作成されたファイル・システムリソースと、Postfix リソース階層と依存関係を作成する必要があります。

以下のような手順で行います。

1. 対象のファイルシステムをマウントする
2. LifeKeeper GUI の **[Edit]** 、 **[Server]** 、 **[Create Resource Hierarchy]** を選択
3. **[Select Recovery Kit]** で **[File System]** を選択
4. **[Switchback Type]** スイッチバック・タイプを選択
5. **[Server]** プライマリ・サーバを選択
6. **[Mount Point]** マウントポイントを選択
7. **[Root Tag]** タグの名前を入力

これによりプライマリ・サーバでファイル・システムリソースが作成されるため、バックアップ・サーバにも拡張します。そしてこの作成されたファイル・システムリソースを Postfixのリソース階層内に依存させます。Lifekeeper for Linux テクニカルドキュメンテーションの依存関係の作成 を参照してください。

# 6.14.3.1.6. Postfix リソース階層のテスト

Postfix リソース階層のテストは、手動でサーバを切り替えて、プライマリーサーバからバックアップサーバーへのリソースインスタンスのフェイルオーバーをシミュレートすることによって行えます。

## LifeKeeper GUI からの手動切り替えの実行

LifeKeeper GUI で **[Edit] > [Resource] > [In Service]** を選択すると、手動切り替えを開始できます。たとえば、バックアップサーバーでサービス中リクエストが実行されると、NAS リソース階層がバックアップサーバー側でサービス中になり、プライマリサーバー側ではサービス休止になります。この時点で、元のバックアップサーバーがプライマリーサーバーになり、元のプライマリーサーバーがバックアップサーバーになります。

**[Our of Service]** を実行した場合、リソース階層はサービス休止になりますが、別のサーバーでサービス中にはなりません。

## コマンドラインインターフェースによる手作業の切り替えの実行

LifeKeeper GUI による切り替え操作を実施する場合は、以下のコマンドをその操作を実施したいシステム上で入力します。

```
# /opt/LifeKeeper/bin/perform_action -t [tag-name] -a [restore|remove]
```

- **-t**

  起動や停止をする LifeKeeper により保護されているサービスのタグ名を指定します。タグ名は LifeKeeper GUI 画面や、lcdstatus コマンドにより確認することができます。

- **-a**

  実行されるリソースのアクションを指定します。起動を実施する場合 restore、停止する場合 remove とします。

perform_action コマンドの詳細については man オンラインマニュアルをご参照ください。

## リカバリ動作

LifeKeeper はサービスを提供中のシステム(プライマリーサーバー) で以下のような障害が発生した場合、リカバリー動作を実施します。

- Postfix サービス障害 (Postfix リソース)
- IP アドレス障害 (Postfix リソースに依存するIP リソース)
- 共有ファイルシステム障害(Postfix リソースに依存するファイル･システムリ ソース)

- ノード障害 (サーバのハングアップや、電源断等)

プライマリーサーバーに障害が発生すると、Postfix Mail Server Recovery Kit は次の作業を実行します。

- バックアップサーバーの物理ネットワークインタフェースに対し論理インタフェースをサービス中にすることによって、バックアップサーバーでエイリアス IP(仮想 IP) アドレスをサービス中にします。

- バックアップサーバーで共有ディスク上のファイルシステムをマウントします。

- Postfix 関連のデーモン・プロセスを開始します。

リカバリー後はセッションコンテキストが失われるため、Postfix ユーザは最初に接続し たときと正確に同じ手順を使用して再接続する必要があります。

# 6.14.4. Postfix トラブルシューティング

この項には、LifeKeeper Postfix 階層の作成、拡張、削除、復元処理時に表示される可能性のあるメッセージの一覧が記載されています。また、該当する場合には、エラーの原因とエラーの解決に必要な処置についても説明されています。他のLifeKeeperスクリプトやユーティリティがメッセージを表示することもありえます。その場合は、そのスクリプトやユーティリティのマニュアルを参照してください。この項ではメッセージを次のように分類しています。

リソース階層作成時のエラーメッセージ

リソース階層拡張時のエラーメッセージ

サービスを起動、停止、回復時に出力されるエラーメッセージ

# 6.14.4.1. Postfix リソース階層作成時のエラーメッセージ

| エラー | エラーメッセージ |
|---|---|
| No config path | コンフィグレーションパスが指定されていません。コンフィグレーションパスを指定してください。 |
| main.cf not found in the configuration path | 指定されたコンフィグレーションパスにはmain.cf が見つかりません。パスを見直し、正しいパスを指定してください。 |
| master.cf not found in the configuration path | 指定されたコンフィグレーションパスには master.cf が見つかりません。パスを見直し、正しいパスを指定してください。 |
| A value of inet_interfaces must be IPv4 or “all” | main.cf 内の inet_interfaces パラメータには IPv4 アドレスまたはall を指定してください。 |
| No execute path | 実行パスが指定されていません。postfix のコマンドが置かれている実行パスを指定してください。 |
| Postfix command invalid | 指定された実行パスには postfix が見つからないか、有効ではありません。postfix のインストール状況および、コマンドについて確認し正しい実行パスを指定してください。 |
| *<queue directory>* is not found. This directory must exist on a shared filesystem | 指定されたキューディレクトリは共有ファイルシステム上にありません。共有ファイルシステム上にキューを保持するように設定してください。 |
| *<tag name>* not in service on the server | 選択されたタグ名はサーバ上でアクティブではありません。IPリソースを作成し、仮想 IP アドレスがサーバ上でアクティブとなっているかを確認してください。 |
| Could not find IP resouce for “*<IP address>*” | main.cf 内の inet_interfaces パラメータに指定された IP address の IP リソースが見つかりません。IP リソースを作成してください。 |

# 6.14.4.2. Postfix リソース階層拡張時のエラーメッセージ

| エラー | エラーメッセージ |
| --- | --- |
| postfix id does not match between servers | postfixのオーナーID がクラスタサーバ間において一致していません。postfix のオーナーID が全てのサーバで一致するよう設定してください。 |
| postdrop gid does not match between servers | postdrop のグループID がクラスタサーバ間において一致していません。postdrop の グループID が全てのサーバで一致するよう設定してください。 |

# 6.14.4.3. Postfix リソース In-Service / Out-of-Service / Health Monitoring エラーメッセージ

| エラー | エラーメッセージ |
|---|---|
| Master process of postfix is not running | postfixのmasterデーモンが起動していません。 maillogなどからpostfixの動作に異常が生じていなかったか確認してください。 |
| Failed in a check by postfix command | postfixのcheckオプションによるエラーが発生しました。 postfixの環境および、設定内容を確認してください。 |
| Couldn't start postfix resource | postfixのサービスを起動することができませんでした。 maillogなどからpostfixの動作に異常が生じていなかったか確認してください。 |
| Failed in a stop process by kill command | postfixの停止時にkillコマンドによる停止を試みましたが失敗しました。 maillogなどからpostfixの動作に異常が生じていなかったか確認してください。 |
| PID <*pid*> does not exist. postfix may have already stopped | 起動されているはずのpostfixのプロセスIDは存在していません。 maillogやpostfixのプロセス稼動状況を確認してください。既に停止されているか、別のIDにて起動されているなどが原因として考えられます。 |
| Check script was not able to be connected to a socket *(vip:port)* | サービスが待ち受けているソケットに接続することができませんでした。postfixの設定内容を確認してください。 |
| Execute files (postfix or postconf command) is not an executable file | postfixまたはpostconf実行ファイルが存在しない、もしくは実行権がありません。 リソース作成時に指定した実行パス内にて、これらの実行ファイルを確認してください。 |
| Configuration files (main.cf or master.cf) does not exist | main.cfまたはmaster.cf実行ファイルが存在しない、もしくは実行権がありません。 リソース作成時に指定したコンフィグレーションパス内にて、これらのファイルを確認してください。 |
| The postfix owner <*owner name*> does not exist | postfixのオーナーがサーバ上に存在していません。 |
| The postdrop group id does not match an attribute of queue directory | postdropのグループIDがキューディレクトリ内maildropのgidと一致していません。 postfixの設定内容ついて確認してください。 |

# 6.15. Recovery Kit for Route 53 管理ガイド

## Recovery Kit for Route 53

Recovery Kit for Route 53は、障害の発生したプライマリサーバからバックアップサーバに復旧する際、依存関係にあるIPリソースの仮想IPアドレス・実IPアドレス情報に対応するAmazon Route 53 DNS 情報を更新する仕組みを提供します。

## SIOS Protection Suite/LifeKeeper ドキュメンテーション

以下は、SIOS Technology Corp が発行している SIOS Protection Suite for Linux 関連ドキュメントの一覧です。

- SPS for Linux テクニカルドキュメンテーション
- SPS for Linux リリースノート
- SIOS Technology Corp. ドキュメンテーション

詳細については、Amazon Route 53 ドキュメント を参照してください。

「Amazon Web Services」、「Powered by Amazon Web Services」のロゴ、「AWS」、「Amazon EC2」、「EC2」、「Amazon Elastic Compute Cloud」、「Amazon Route 53」、「Amazon Virtual Private Cloud」および「Amazon VPC」は、米国その他の国における Amazon.com, Inc. またはその関連会社の商標です。なお、本文および図表中では、「™」、「®」は明記しておりません。(一部記載がございます。)

# 6.15.1. Recovery Kit for Route 53™ の要件

Recovery Kit for Route 53 をインストールして設定する前に、使用している設定が以下の要件を満たしていることを確認してください。

- AWS Command Line Interface (AWS CLI) を全ての EC2 インスタンスにインストールする必要があります。インストール方法は、「AWS Command Line Interface のインストール」を参照してください。
- Amazon Route 53 サービスのエンドポイント route53.amazonaws.com にプロトコル HTTPS を使用してアクセスできる必要があります。EC2 および OS の設定を適切に行ってください。
- Amazon Route 53 に適切なドメイン名を登録してください。
- LifeKeeper が AWS を操作するために、以下のアクセス権限を持った IAM ユーザーもしくは IAM ロールが必要です。EC2 インスタンスに適切な権限を持つ IAM ロールをアタッチするか、root ユーザーの AWS CLI プロファイルに権限のある IAM ユーザーを登録してください。AWS の IAM ロールと AWS CLI の詳細については AWS のユーザーガイド、Amazon EC2 の IAM ロール および AWS CLI の開始方法 を参照してください。
  - route53:GetChange
  - route53:ListHostedZones
  - route53:ChangeResourceRecordSets
  - route53:ListResourceRecordSets

> 設定ファイル */etc/default/LifeKeeper* のパラメーター PATH に AWS CLI 実行ファイルのパスが設定されていない場合、PATH に AWS CLI 実行ファイルのパスを追加してください。

- ネットワーク構成（クライアントとクラスターノードが存在する VPC 間の通信経路）が以下の条件を満たすことを確認してください。
  1. クライアントと各クラスタノードの実 IP アドレスとの間で、 IP 通信ができること。
  2. クライアントから、 Route53 の DNS サーバーに登録された A レコードの名前解決（ホスト名の FQDN から実 IP アドレスを取得）ができること。

## LifeKeeper ソフトウェア:

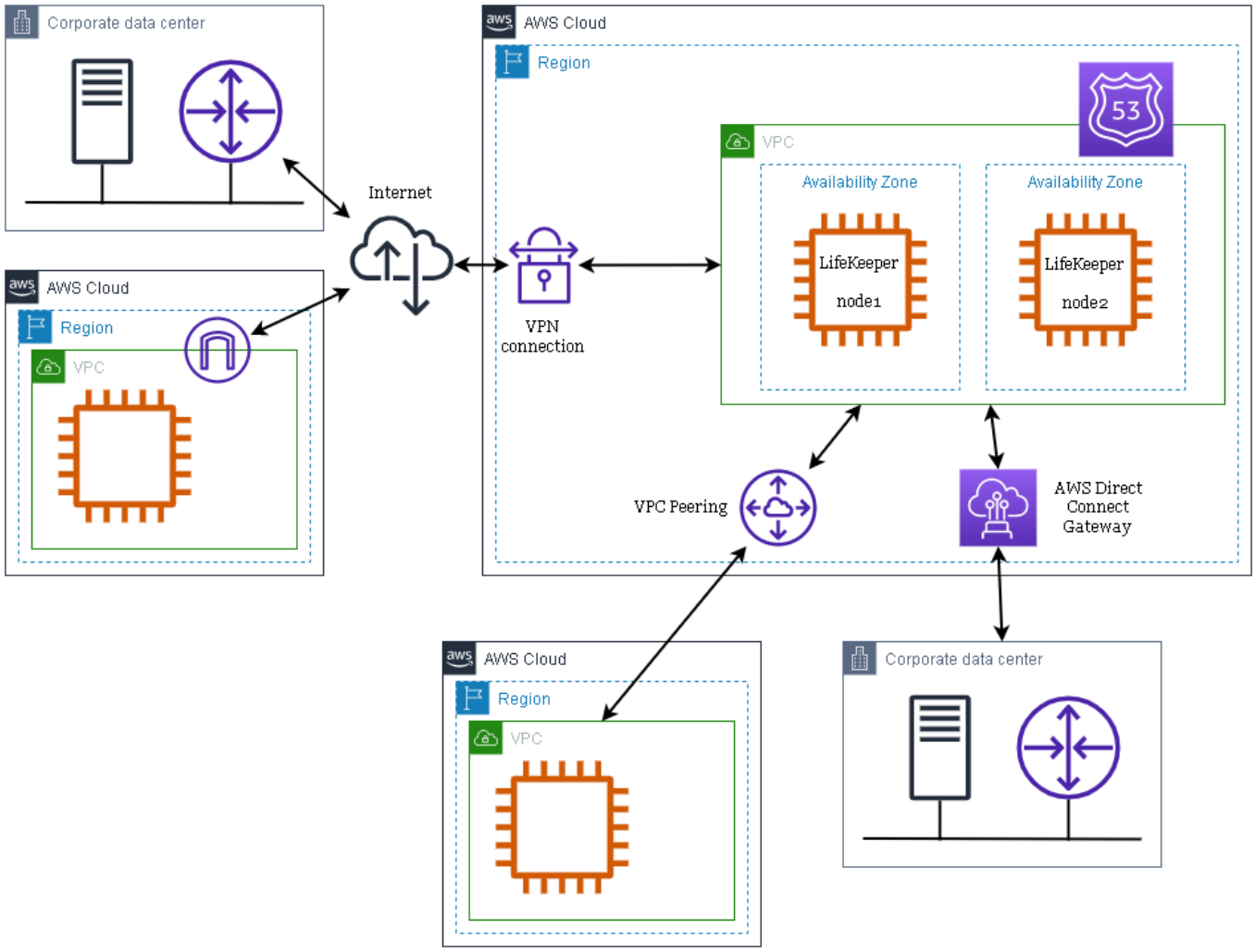
各サーバに同じバージョンの LifeKeeper ソフトウェアとパッチをインストールする必要があります。具体的な LifeKeeper の要件については、LifeKeeper for Linux テクニカルドキュメンテーション および LifeKeeper for Linux リリースノート を参照してください。

Recovery Kit for Route 53 を使う場合には、Quorum/Witnessの使用を推奨します。
詳細については、こちら を参照してください。

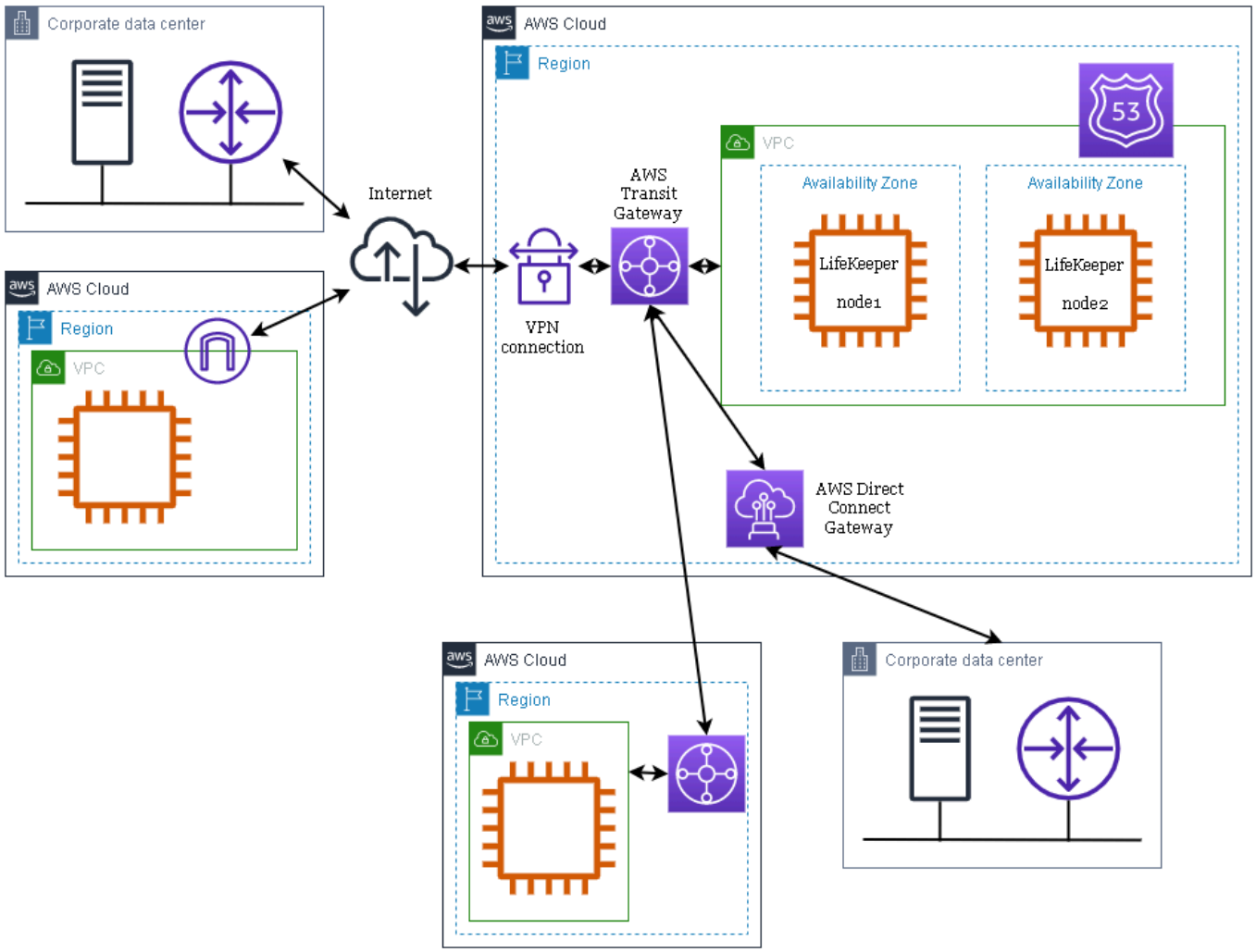
# 6.15.2. 構成例

ネットワーク構成が 要件 に記載している条件を満たす場合に Recovery Kit for Route53 を利用できます。例えば以下のような構成が挙げられます。

## Transit Gateway を使わない構成




## Transit Gateway を使う構成


Corporate data center
Internet
AWS Cloud
Region
VPC
AWS Cloud
Region
VPC
AWS
Transit
Gateway
VPN
connection
53
VPC
Availability Zone
Availability Zone
LifeKeeper
node1
LifeKeeper
node2
AWS Direct
Connect
Gateway
AWS Cloud
Region
VPC
Corporate data center

# 6.15.3. Recovery Kit for Route 53™ の設定

必要とする保護と柔軟性が得られるように LifeKeeper を設定するには、設定要件を把握している必要があります。設定を適切にプランニングするには、Amazon、Amazon Virtual Private Cloud (VPC)、Amazon Elastic Compute Cloud (EC2)、およびユーザシステムの階層設定オプションを理解する必要があります。このセクションでは、設定のプランニングに加え、Recovery Kit を設定するために必要な特定の作業についても説明します。

## Route53 リソース 特有の設定上の考慮事項

このセクションでは、Route53 リソースの以下の設定タスクについて説明します。設定タスクは Route53 リソースインスタンスに特有のものであり、Recovery Kit ごとに異なります。

- リソース階層の作成: アプリケーションリソース階層を LifeKeeper クラスタに作成します。
- リソース階層の削除: リソース階層を LifeKeeper クラスタ内のすべてのサーバから削除します。
- リソース階層の拡張: リソース階層をプライマリサーバからバックアップサーバへ拡張します。
- リソース階層の拡張解除: リソース階層を LifeKeeper クラスタ内の 1 つのサーバから拡張解除 (削除) します。
- Recovery Kit for Route 53のチューニング: Recovery Kit for Route 53 の動作を調整します。

以下の作業は、SPS for Linux テクニカルドキュメンテーション の管理 セクションに記載されています。これらは、すべての Recovery Kit で同じ手順を使用する共通の作業です。

- リソース依存関係の作成: 既存のリソース階層と別のリソースインスタンスとの間に親子の依存関係を作成し、クラスタ内のすべての対象サーバに依存関係の変更を反映します。
- リソース依存関係の削除: リソースの依存関係を削除して、クラスタ内のすべての対象サーバに依存関係の変更を反映します。
- In Service: リソース階層を特定のサーバで In Service にします。
- Out of Service: リソース階層を特定のサーバで Out of Service にします。
- プロパティの表示 / プロパティの編集: 特定のサーバでリソース階層のプロパティを表示または編集します。

このセクションの残りの部分では、LifeKeeper GUI の [Edit] メニューから作業を選択することによって、Recovery Kit を設定する方法を説明します。設定作業はツールバーから選択することもできます。

- ◦ 状況表示ウィンドウのリソース階層ツリー (左側のペイン) のグローバルリソースを右クリックすると、[Edit] メニューと同じドロップダウンメニューの選択項目が表示されます。これは階層がすでに存在している場合にだけ可能な方法です。

- ◦ 状況表示ウィンドウのリソース階層表 (右側のペイン) のリソースインスタンスを右クリックすると、サーバおよび特定リソースの状況に応じて、リソース階層の作成を除くすべての設定作業を実行できます。

# 6.15.3.1. Route53™ リソース階層の作成

プライマリーサーバからリソースインスタンスを作成するには、以下の手順を完了する必要があります。

1. LifeKeeper GUI メニューから **[Edit]** を選択し、次に **[Server]** を選択してください。ドロップダウンメニューから、 **[Create Resource Hierarchy]** を選択してください。

2. ダイアログボックスが表示され、クラスタ内にインストール済みの認識されている Recovery Kit がすべて、ドロップダウンリストボックスに表示されます。ドロップダウンリストから **[Amazon Route53]** を選択し、 **[Next]** をクリックしてください。

3. 以下の情報を入力するようにプロンプトが表示されます。(ダイアログボックスで [Back] ボタンが有効な場合は、前のダイアログボックスに戻ることができます。これは、以前に入力した情報を訂正する必要がある場合に特に役立ちます。)

   **注記:** 階層作成の途中で [Cancel] ボタンをクリックすると、作成処理全体が取り消されます。

| フィールド | ヒント |
|---|---|
| Switchback Type | フェイルオーバ後にサーバが復帰したときに、このサーバに EC2 インスタンスを戻す方法を指定します。[intelligent] または [automatic] を選択できます。<br>• <br>◦ [intelligent] の場合、インスタンスをプライマリ / オリジナルサーバにスイッチバックするときに管理者の介入が必要になります。<br>◦ [automatic] の場合、プライマリサーバがオンラインに戻り、LifeKeeper コミュニケーションパスを再確立した直後に自動的にスイッチバックが行われます。<br><br>**注記:** スイッチバックタイプは、必要な場合、[Resource Properties] ダイアログボックスの [General] タブで後から変更できます。 |
| Server | [Server] で Route53 リソースの作成先のサーバ (通常、これをプライマリサーバまたはテンプレートサーバと呼ぶ) を選択します。クラスタ内のすべてのサーバがドロップダウンリストに表示されます。 |
| Domain name (Route53 hosted zone) | Route53のホストゾーンがドロップダウンリストに表示されます。使用するドメイン名を選択してください。 |
| Host Name (Not FQDN) | 使用するホスト名を入力してください。 |
| IP resource | IP リソースを選択してください。これは LifeKeeper が保護する仮想 IP アドレスもしくは実IPアドレスです。 |
| Route53 Resource Tag | 作成している Route53 リソースインスタンスに固有の Route53 リソースタグ名を選択または入力します。このフィールドには、デフォルトのタグ名である route53-<host name> |

| | |
|---|---|
| | が自動的に表示されます |

4. [Create] をクリックしてください。[Create Resource Wizard] によって、Route53 リソースが作成されます。

5. この時点で情報ボックスが表示され、LifeKeeper は、Route53 リソース階層を作成するために有効なデータが指定されたかどうかを検証します。LifeKeeper が問題を検知した場合は、情報ボックスにエラーが表示されます。検証が正常に完了すると、リソースが作成されます。 **[Next]** をクリックしてください。

Route53 リソース階層が正常に作成されたことを示す情報ボックスが表示されます。リソース階層を LifeKeeper で保護するには、クラスタ内の別のサーバにそのリソース階層を拡張する必要があります。

**[Continue]** をクリックすると、[Pre-Extend configuration task] が起動されます。リソース階層を別のサーバに拡張する方法の詳細については、リソース階層の拡張 を参照してください。

ここで **[Cancel]** をクリックすると、別のダイアログボックスが表示され、後で Route53 リソース階層を別のサーバに手動で拡張して LifeKeeper の保護下に置く必要があることが警告されます。

# 6.15.3.2. Route53™ リソース階層の削除

LifeKeeper 環境のすべてのサーバからリソース階層を削除するには、次の手順を実行してください。

1. LifeKeeper GUI メニューから **[Edit]** を選択し、次に **[Resouce]** を選択してください。ドロップダウンメニューから **[Delete Resource Hierarchy]** を選択してください。

2. Route53 リソース階層から削除するターゲットサーバの名前を選択し、 **[Next]** をクリックしてください。

   **注記:** このダイアログボックスは、いずれかのペインでリソースインスタンスを右クリックして [Delete Resource] を選択した場合には、表示されません。

3. [Hierarchy to Delete] を選択してください。削除するリソース階層を指定して強調表示にしてから、**[Next]** をクリックしてください。

   **注記:** このダイアログボックスは、左右どちらかの インでリソースインスタンスを右クリックして [Delete Resource] を選択した場合には、表示されません。

4. 選択したターゲットサーバと、削除の対象として選択した階層を確認する情報ボックスが表示されます。 **[Delete]** をクリックして次に進んでください。

5. Route53 リソースが正常に削除されたことを確認する情報ボックスが表示されます。

6. **[Done]** をクリックして終了してください。

# 6.15.3.3. Route53™ リソース階層の拡張

階層の作成後、クラスター内の別のサーバに拡張する必要があります。3 通りのシナリオで、テンプレートサーバからターゲットサーバーにリソースインスタンスを拡張できます。

- リソースの作成後、[Continue] をクリックして、別のサーバにリソースを拡張します。
- 次に説明するように [Edit] メニューから [Extend Resource Hierarchy] を選択します。
- 左側または右側のペインから拡張されていない階層を右クリックします。

どのシナリオでも同じダイアログボックスが表示されます (いくつかの例外については、以下に詳細を明記)。

1. LifeKeeperGUI メニューから [Extend] ウィザードを開始する場合は、 **[Edit]** を選択し、次に **[Resource]** を選択します。ドロップダウンメニューから **[Extend Resource Hierarchy]** を選択します。これで [Extend Resource Hierarchy] ウィザードが起動されます。拡張操作に慣れていない場合は、 **[Next]** をクリックします。LifeKeeper の [Extend Resource Hierarchy] のデフォルト値が分かっていて、入力と確認を省略する場合は **[Accept Defaults]** をクリックします。

2. [Pre-Extend Wizard] で以下の情報を入力します。

   **注記:** 最初の 2 つのフィールドは [Edit] メニューの [Extend] から操作を開始した場合にだけ表示されます。階層を拡張する手順の間に [Cancel] をクリックすると、どの時点であってもそのサーバへの拡張処理がキャンセルされるので注意してください。ただし、すでにリソースを別のサーバに拡張している場合は、明示的に拡張解除するまで、そのインスタンスの拡張は有効です。

| フィールド | ヒント |
|---|---|
| Switchback Type | スイッチバックタイプを選択します。ここでは、バックアップサーバへのフェイルオーバの後、Route53 インスタンスが In Service に戻ったときに、このサーバにRoute53 インスタンスをスイッチバックする方法を指定します。[intelligent] または [automatic] を選択できます。<br><br>* intelligent の場合、インスタンスをプライマリ / オリジナルサーバにスイッチバックするときに管理者の介入が必要になります。<br><br>* automatic の場合、プライマリサーバがオンラインに戻り、LifeKeeper コミュニケーションパスを再確立した直後に自動的にスイッチバックが行われます。<br><br>スイッチバックタイプは、必要な場合 [Resource Properties] ダイアログボックスの [General] タブで後から変更できます。 |
| Template Priority | テンプレートの優先順位を選択するか、入力してください。これはサーバで現在 In Service |

| | |
|---|---|
| | の Route53 階層の優先順位です。1 ～ 999 の範囲で、まだ優先順位として使用されていない値が有効で、小さい数値ほど優先順位が高くなります (数値 1 が最高の優先順位)。拡張処理時に、別のシステムですでに使用中の優先順位をこの階層に対して指定することはできません。デフォルト値を使用することを推奨します。<br><br>**注記:** このフィールドは、階層を最初に拡張するときにだけ表示されます。 |
| Target Priority | ターゲットの優先順位を選択するか、入力してください。これは、他のサーバにある同等の階層に対する、新しく拡張する Route53 階層の優先順位です。1 ～ 999 の範囲で、まだ優先順位として使用されていない値が有効で、リソースのカスケーディングフェイルオーバシーケンスにおけるサーバの優先順位を示します。数値が小さいほど優先順位が高くなります (数値 1 が最高の優先順位)。<br><br>**注記:** LifeKeeper のデフォルトでは、階層が作成されたサーバに「1」が割り当てられます。優先順位は連続している必要はありませんが、特定のリソースについて 2 つのサーバに同じ優先順位を割り当てることはできません。 |

3. 情報ボックスが表示され、LifeKeeper が環境のチェックを正常に完了し、この EC2 リソースを拡張するためのすべての要件が満たされていることが示されます。満たされていない要件がある場合は、 **[Next]** ボタンは選択できなくなり、 **[Back]** ボタンが有効になります。 **[Back]** をクリックした場合、情報ボックスに表示されるエラーメッセージの内容に従って、リソースの拡張を変更できます。ここで **[Cancel]** をクリックした場合は、EC2 リソース階層を他のサーバに手動で拡張して、LifeKeeper の保護下に置く必要があります。 **[Next]** をクリックすると、[Extend Resource Hierarchy] 設定作業に入ります。

4. リソース階層構造の拡張を行うために、以下の情報の入力が要求されます。

| フィールド | ヒント |
|---|---|
| Route53 Resource Tag | Route53リソースタグを選択するか、入力してください。これは、ターゲットサーバに拡張される Route53 リソースが使用するリソースタグ名です。<br><br>**注記:** このフィールドは編集できません。 |

5. 拡張が実行中であることを示す情報ボックスが表示されます。同じ Route53リソースインスタンスをクラスタ内の別のサーバに拡張する場合は、 **[Next Server]** をクリックしてください。その場合は、リソース階層を拡張する操作を繰り返します。 **[Finish]** をクリックすると、LifeKeeper は Route53 リソースの拡張が正常に完了したことを確認します。

6. **[Done]** をクリックして、 **[Extend Resources Hierarchy]** メニューの選択を終了します。

   **注記:** 必ずすべてのサーバで新しいインスタンスの機能をテストしてください。

# 6.15.3.4. Route53™ リソース階層の拡張解除

階層全体を拡張解除するには、以下の手順を実行してください。

1. **LifeKeeper GUI メニュー** から **[Edit]** を選択し、次に **[Resouce]** を選択してください。ドロップダウンメニューから **[Unextend Resource Hierarchy]** を選択してください。

2. EC2 リソースから、拡張解除の対象となるターゲットサーバを選択してください。 Route53 リソースが現在 In Service になっているサーバは選択できません。 **[Next]** をクリックしてください。

   **注記:** 右側のペインから個々のリソースインスタンスを右クリックして **[Unextend]** 作業を選択した場合、このダイアログボックスは表示されません。

3. 拡張解除する Route53 階層を選択してください。 **[Next]** をクリックしてください。

   **注記:** 左側のペインにあるグローバルリソースを右クリックするか、右側のペインにある個々のリソースインスタンスを右クリックして **[Unextend]** 作業を選択した場合、このダイアログボックスは表示されません。

4. 拡張解除の対象として選択したターゲットサーバと Route53 リソース階層を確認する情報ボックスが表示されます。 **[Unextend]** をクリックしてください。

5. EC2 リソースが正常に拡張解除されたことを示す情報ボックスが表示されます。

6. **[Done]** をクリックして終了してください。

# 6.15.3.5. Recovery Kit for Route 53 のチューニング

Recovery Kit for Route 53 で設定可能なパラメータは パラメータ一覧 を参照してください。

# 6.15.3.6. Route53™ リソースの監視とリカバリ動作

Route53 リソースは作成時に登録した DNS A レコードの取得と仮想IPアドレスとの関連づけの正常性を監視しています。リソースの監視内容として、以下のような処理を行っています。

1. Route 53 の A レコードで設定されているアドレスを API で取得します。取得に失敗した場合は、デフォルトで 2 秒の間隔を置いてから、情報の再取得を行います。
   3回取得を試行して取得できなかった場合、ログに記録を残し監視処理を終了します。

2. Route53リソースと依存関係がある IP リソースから IP アドレスを取得して、 Route53 の A レコードで設定されているアドレスと IP リソースの IP アドレスを比較します。
   一致した場合、正常であると判断して処理を正常終了します。一致しなかった場合、異常であると判断しリカバリを開始します。

# 6.15.3.7. Route53™ ユーザーシステムのセットアップ

Route53 リソース作成の際必要とするIPリソースは、仮想 IP リソースと実 IP リソース（ネットワークインターフェース用に構成されたプライマリー IP アドレスのためのリソース）のいずれかを使用することができます。

## 仮想 IP リソースを使用する場合

Route53 リソースの子リソースに仮想IPリソースを使用する場合、リソースのスイッチオーバー時に仮想IPアドレスに対する通信がバックアップサーバーに対して行われるよう、ルートテーブルを再設定する必要があります。
そのため、Recovery Kit for EC2 を併用してください。詳細は Recovery Kit for EC2 のマニュアルを参照してください。

## 実 IP リソースを使用する場合

Route53 リソースの子リソースに実 IP リソースを使用する場合、追加で設定する項目はありません。
ただしスイッチオーバー毎に接続先 IP アドレスが変更されてしまうため、接続には必ず Route53 リソースが保護しているホスト名で接続を行わなければならないことに注意してください。

# 6.15.4. Recovery Kit for Route 53™ のトラブルシューティング

メッセージカタログ は、LifeKeeper for Linux の使用時に検出されるすべてのメッセージのリストを示すとともに、適宜、エラーの原因とエラー状態を解決するために必要な処置の説明を提供します。受け取ったエラーコードをこのリストから探すことができます。

## Route53リソース起動にともなうレコードの更新に時間がかる場合があります

Route53のDNSレコードの更新の伝播速度に関して、Amazonでは以下のような情報を提供しています。

Amazon Route 53 よくある質問
Q: Amazon Route 53 での DNS 設定の変更が全体に適用されるには、どのくらいの時間がかかりますか?
http://aws.amazon.com/jp/route53/faqs/

Route53リソースではRoute53 APIを使用してDNSへのレコード更新の状況について確認しています。INSYNCステータスを得られた場合には更新が完了したと判断し、PENDINGステータスが帰ってきた場合には、更新状況の確認のリトライを行います。このような仕様のため、Route53リソースの起動処理としては正しくレコード更新がかけられているにも関わらず、DNSへの更新情報の伝播に時間がかかった場合には、Route53リソースとしては起動失敗の状態になってしまう場合があります。

もし、Route53リソースの起動に失敗した場合には、Route53の管理コンソールをみてAレコードが正しく更新されていることを確認してください。更新されている場合、該当するDNSサービスの更新は完了しており、DNSサービスの更新を反映するには、LifeKeeperのみ更新が必要となります。もう一度LifeKeeper GUIからRoute53リソースを起動してください。

常時前述のようなケースでRoute53リソースの起動ステータスが失敗する場合には、/etc/default/LifeKeeperファイルの「ROUTE53_CHANGEID_TRY_COUNT」の値をデフォルトの５回から１～２回程度値を増やして、状況が改善するかご確認ください。この設定変更にはLifeKeeperやOSの再起動は必要ありません。

## DNSレコードのTTLの値を適切に設定してください

スイッチオーバーやフェイルオーバーが発生した後のクライアントからのアクセスは、TTLの設定によって決められた時間が過ぎるまで各クライアントが保持するDNS情報のキャッシュを用いて行います。そのため、TTLの値が長く設定されている場合、切り替わり前のアドレスへのアクセスが発生する機会が増えることとなり、予期せぬ問題が生じる可能性があります。また、TTLの値が短く設定されている場合、頻繁に名前解決が行われるので、ネットワークへの負荷がかかります。環境に応じて、可能な限り短くTTLの値を設定してください。

TTLの設定は、/etc/default/LifeKeeper ファイルで「ROUTE53_TTL」パラメータを設定してください。単位は秒です。

# 6.15.5. Recovery Kit for Route53™ 処理概要

各動作は Amazon Web Services より提供されている AWS CLI より実行されます。

## 起動処理 (restore)

1. 以下のコマンドにて、Route53 の A レコードに設定されている IP アドレスを AWS CLI で取得します。

```
# aws --output text --cli-connect-timeout AWSCLI_CONNECT_TIMEOUT --cli-
read-timeout AWSCLI_READ_TIMEOUT route53 list-resource-record-sets --ho
sted-zone-id $zoneId --query \"ResourceRecordSets[?Name == \'$fqd
n.\']\" 2>&1
```

   上記コマンドの実行に失敗した場合は、起動処理はエラーで終了します。
   上記コマンドの実行に成功した場合は、次の項目2. の処理へ移行します。

2. Route53 リソースと依存関係がある IP リソースから IP アドレスを取得します。
   Route53 の A レコードで設定されている IP アドレスと IP リソースの IP アドレスを比較して一致した場合は、正常と判断して起動処理は成功で終了します。

   一致しなかった場合は、次の項目3. の処理へ移行します。

3. 以下のコマンドにて、Route53 の A レコードを更新します。何も設定されていない場合は、A レコードを新規作成します。

```
# aws --output text --cli-connect-timeout AWSCLI_CONNECT_TIMEOUT --cli-
read-timeout AWSCLI_READ_TIMEOUT route53 change-resource-record-sets --
hosted-zone-id $zoneId --change-batch \'$upsertJSON\' 2>&1
```

   更新および作成に失敗した場合は、異常と判断し起動処理はエラーで終了します。
   成功した場合は、次の項目4. の処理へ移行します。

4. 以下のコマンドにて、A レコードの更新、新規作成処理のステータスが “INSYNC” となっていることを確認します。
   INSYNC となっていない場合は、ROUTE53_CHANGEID_INTERVAL の値（デフォルト値：20秒）の間隔でで最大で ROUTE53_CHANGEID_TRY_COUNT の値（デフォルト値：5回）の回数実施します。

```
# aws --output text --cli-connect-timeout AWSCLI_CONNECT_TIMEOUT --cli-
read-timeout AWSCLI_READ_TIMEOUT route53 get-change --id /change/$chang
eId 2>&1
```

INSYNC とならなかった場合は、異常と判断し起動処理はエラーで終了します。
INSYNC となった場合は、次の項目5. の処理へ移行します。

5. 再度、項目1. と同様の処理を行います。

6. 再度、項目2. と同様の処理を行います。
前の項目5. で取得した IP アドレスと IP リソースの IP アドレスが一致した場合は、正しい IP アドレスが A レコードに登録できたと判断し起動処理は成功で終了します。
一致しなかった場合は、起動処理はエラーで終了します。

AWS CLI から応答がない場合は ROUTE53_RESTORE_TIMEOUT の値（デフォルト値：140秒）でタイムアウトします。

# 停止処理 (remove)

Route53 リソースに関する処理はありません。

# 監視処理 (quickCheck)

1. 以下のコマンドにて、Route53 の A レコードに設定されている IP アドレスを AWS CLI で取得します。

```
# aws --output text --cli-connect-timeout AWSCLI_CONNECT_TIMEOUT --cli-read-timeout AWSCLI_READ_TIMEOUT route53 list-resource-record-sets --hosted-zone-id $zoneId --query \"ResourceRecordSets[?Name == \'$fqdn.\']\" 2>&1
```

上記コマンドの実行に失敗した場合は、監視処理はエラーで終了します。
上記コマンドの実行に成功した場合は、次の項目2. の処理へ移行します。

2. Route53 リソースと依存関係がある IP リソースから IP アドレスを取得します。
Route53 の A レコードで設定されている IP アドレスと IP リソースの IP アドレスを比較して一致した場合は、正常と判断して起動処理は成功で終了します。

一致しなかった場合は、監視処理はエラーで終了します。

AWS CLI から応答がない場合は ROUTE53_QUICKCHECK_TIMEOUT の値（デフォルト値：25秒）でタイムアウトします。ただし、タイムアウトではリソース障害とは判断しません。

# 回復処理 (recover)

起動処理と同じ処理を実行します。
AWS CLI から応答がない場合は ROUTE53_RECOVERY_TIMEOUT の値（デフォルト値：140秒）でタイムアウトします。

# Recovery Kit for Route53 のパラメーターについて

処理概要の中で使用されている Recovery Kit for Route53 のパラメーターについては、Route53パラメーター一覧 をご参照ください。

# 6.16. Samba Recovery Kit 管理ガイド

**重要** ：SIOS は次の Recovery Kit のサポートを終了します。これらは LifeKeeper for Linux の次のメジャーリリースでサポートされなくなります。
(なお、新規受注は2022年12月31日をもって終了しています。)
- NEC iStorage StoragePathSavior
- Samba
- Postfix
- HULFT HUB Recovery Kit

LifeKeeper for Linux Samba Recovery Kit は異種ネットワーク環境にある Linux サーバ上で 動作する Samba のファイルとプリンタの共用を保護し耐障害性を高めます。この Kit により、プライマリー Samba サーバーで障害が発生しても、わずかな時間で人手を介さずに指定されたバックアップサーバーに現状を回復することができます。

## ドキュメントの内容

本書は以下のトピックについて説明しています。

- ドキュメンテーションとリファレンス ：LifeKeeper for Linux のドキュメントと所在の一覧です。
- 要件 ：Samba Recovery Kit を適切に設定、導入、運用するうえで必要なハードウェアとソフトウェアについて説明しています。LifeKeeper for Linux ソフトウェアの具体的なインストール / アンインストール方法については、LifeKeeper for Linux インストールガイド を参照してください。
- Samba Recovery Kitの概要 ：Samba Recovery Kit の特徴と機能について簡単に説明しています。
- LifeKeeper for Linux Samba Recovery Kit の設定 ：Samba Recovery Kit を適切に設定するうえで必要な手順について説明しています。
- LifeKeeper 設定作業 ：LifeKeeper GUI を使用して Samba リソース階層を作成し管理する作業について説明しています。
- Samba 階層の管理 ：Samba リソース作成後に必要になる可能性がある作業について説明しています。
- トラブルシューティング ：LifeKeeper for Linux のエラーメッセージの一覧とその説明です。

## ドキュメンテーションとリファレンス

以下に、SIOS Technology Corp が発行している SPS 関連情報の一覧を示します。

- LifeKeeper for Linux リリースノート
- LifeKeeper for Linux テクニカルドキュメンテーション

- LifeKeeper for Linux インストールガイド
- Optional Recovery Kit ドキュメンテーション

このドキュメンテーション、および他の LifeKeeper for Linux Recovery Kit に関連するドキュメンテーションは、以下の Web サイトで入手できます。

http://jpdocs.us.sios.com

# 6.16.1. Samba Recovery Kit の要件

LifeKeeper for Linux Samba Recovery Kit をインストールする前に、LifeKeeper の構成が次の要件を満たしている必要があります。LifeKeeper for Linux ハードウェアおよびソフトウェアの設定に関する具体的な手順については、LifeKeeper for Linux インストールガイド を参照してください。

- サーバー ：Recovery Kit は、LifeKeeper for Linux インストールガイド で説明されている要件に従って 2 つ以上のサーバーを設定する必要があります。対応している Linux ディストリビューションについては、LifeKeeper for Linux リリースノート を参照してください。

- **LifeKeeper ソフトウェア** ：各サーバーに同じバージョンの LifeKeeper ソフトウェアと パッチをインストールする必要があります。

- **LifeKeeper IP Recovery Kit** ：各サーバーの LifeKeeper IP Recovery Kit のバージョンは同 じにする必要があります。

- **IP ネットワークインタフェース** ： 各サーバーには、イーサーネットTCP/IP をサポートするネットワークインタフェースが 1 つ以上必要です。IP 切り替えが正しく機能するためには、ローカルネットワークに接続したユーザーシステムが標準TCP/IP 仕様に準拠している必要があります。

  **注記** ：各サーバーが必要とするネットワーク･インタフェースが 1 つだけであっても、異種メディア要件、スループット要件、シングル･ポイント･オブ･フェイルの排除、ネットワークのセグメンテーションといった多くの理由のために、複数のインタフェースを使用すべきです。

- **TCP/IP ソフトウェア** ：各サーバーは TCP/IP ソフトウェアも必要とします。

- **Samba ソフトウェア** ：Samba は LifeKeeper for Linux がサポートしているすべての Linux ディストリビューションに付属しています。付属しているバージョンの Samba を使用してください

# 6.16.2. Samba Recovery Kit のインストール

LifeKeeper for Linux ソフトウェアの具体的なインストール / アンインストール方法については、LifeKeeper for Linux インストールガイド を参照してください。

# 6.16.3. Samba Recovery Kit の概要

Samba は、SMB (Server Message Block) プロトコルを使用するアプリケーションの集まりで、異種ネットワークで Microsoft Windows 製品の動作するサーバおよびクライアント と Linux サーバとの通信を可能にします。

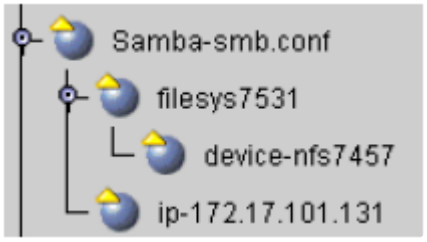
Samba Recovery KitはLinux サーバ上の Samba によるファイルとプリンタの共用を保護します。Samba には、クライアントの認証、ネットワーク・コンピュータの参照支援、WINS ネーム・サーバ解決など、その他のサービスも用意されていますが、LifeKeeperの今回のリリースでは、これらの追加サービスは保護されません。これらのSambaサービスは、

「Samba の複数インスタンスの実行 」のセクションで示されている規則にのっとっている限り、保護されていないSamba インスタンスとしてLifeKeeperサーバ上で共存すること も可能です。

Samba Recovery Kit は、保護された Samba のファイルとプリンタの共用を障害の発生したプライマリ・サーバからバックアップ・サーバに回復する仕組みを提供します。LifeKeeper は、サーバ・レベル（ハートビート経由）またはリソース・レベル（Samba デーモンを監視）で障害を検出して、Samba リソースの制御をバックアップ・サーバに移します。

## Samba リソース階層

一般的な Samba 階層は、1 つの Samba リソース、1 つ以上のファイル・システム・リソ ース、1 つ以上の IP リソースから構成され、1 つのプリント・サービス・リソースが存在することもあります。Samba のファイルの共用を保護しているリソース階層の例を次に示します。




この Samba-smb.conf 階層は 1 つのファイルの共用 filesys7531 (これはパーティション device-nfs7457 に従属しています) と 1 つの IP アドレス 172.17.101.131 を保護しています。次のセクションでは Samba リソースの設定方法について説明します。

# 6.16.4. LifeKeeper for Linux Samba Recovery Kit の設定

LifeKeeper for Linux の Samba リソース階層を作成する前に検討する必要がある設定上の検討事項がいくつかあります。Linux サーバ上の Samba サービスは **smbd** および **nmbd** という 2 つのデーモン・プロセスによって提供されます。このデーモン・プロセスは以下に説明する Samba 設定ファイルで定義された値によって制御されています。

# 6.16.4.1. Samba 設定ファイル

Samba 設定ファイルにはさまざまな命令を多数記述できますが、ここでは LifeKeeper の設定に影響を及ぼす項目に絞って説明します。設定ファイルに関しては、次のようにい くつか重要な点があります。

- 設定ファイルは提供する共用 (すなわちサービス) に対応するセクションから構成 されます。設定ファイルの各セクションには、その共用に特有の設定オプション(すなわち命令) が記述されます。
- 指定された命令は Samba Recovery Kit によって妥当性が検査されます。命令が適切に 設定されていない場合は、Samba リソースの作成に失敗します。
- Samba のデフォルト設定ファイルは、通常、smb.conf という名前で、/etc または /etc/samba（Linux ディストリビューションによって異なる）内にあります。
- 設定ファイルの名前はクラスタ内で重複してはなりません。また、アクティブ／アクティブ設定の場合は、各サーバの異なるディレクトリ内にある必要があります。名前または場所が重複してはいけないのは、Samba Recovery Kit がバックアップ・サーバの同じ場所に拡張するときに、設定ファイルのコピーが作成されるためです。
- アクティブ／スタンバイ設定で Samba デーモンのインスタンスを 1 つだけ実行する 場合は、デフォルト設定を使用できます（これを推奨します）。この場合、Samba デーモンの自動起動を無効にする必要があります。
- アクティブ／スタンバイ設定で複数のSambaインスタンスを実行する場合、またはアクティブ／アクティブ設定を使用する場合は、Samba設定ファイルの名前を重複させることはできません。Sambaの複数インスタンスを実行する場合の要件と詳細については、「Sambaの複数インスタンスの実行 」を参照してください。

次のセクションでは、LifeKeeper で Samba の共用を保護するのに必要なオプションも含めて、設定ファイルのセクションについて説明します。

# 6.16.4.2. 設定ファイルの [global] セクション

[global] セクションは、LifeKeeper Samba リソース階層で使用するすべての設定ファイルに必須の特別なセクションです。名前が示すように、このセクションで設定したオプションは、他のセクションでそのオプションが明示的に指定されない限り、他のすべてのセクションに適用されます。LifeKeeper では、特定の命令を[global]セクションで定義する必要があります。そのうち一部の命令はデフォルトの設定ファイルには記述されていないため、ユーザが追加する必要があります。オプションは次のとおりです。

- **netbios name** - LifeKeeper Samba階層を構成するリソース・セットに付ける一意の名前。クライアントはこの名前を使用して、interfaces命令で定義されたIPアドレス経由で共用に接続します (たとえば、NetBIOS name = server1_print1)。

- **interfaces** - Linux Samba サーバが認識し応答するネットワーク・アドレスの一覧。interfaces 命令を適切に設定するうえで必要な要件は次のとおりです。

  - º Samba　サーバがサービスするすべてのサブネットを記述する必要があります。これは LifeKeeper が保護するアドレスでなくてはならず、クラスタ内で重複してはなりません(その他の Samba 設定ファイルで同じ IP アドレスを使用してはなりません)。

  - º interfaces 命令では複数のフォーマット、IP アドレス(ドット形式またはホスト 名)、ネットワーク・インタフェース名を指定でき、ワイルドカードを使用でき ます。ただし、Samba Recovery Kit の場合、ワイルドカードは使用せず IP アドレスはドット形式 (100.25.104.25) で指定する必要があります。

  - º IP アドレスとあわせてサブネット・マスクを指定することもできますが、LifeKeeper では使用されません。

  - º この命令で指定したアドレス用の LifeKeeper IP リソースは、Samba リソース階層の作成前に作成しておく必要があります。また、この命令のアドレスにネットワ ーク・マスクを適用する場合は、作成時の IP リソースで使用されるマスクと一致している必要があります。

  - º その他の保護されていない Samba インスタンスも、interfaces 命令を使用して、LifeKeeper が使用するものとは異なる IP アドレスを指定します。

    注記： 後述するbind interfaces only 命令を使用するために、interfaces命令で localhost アドレス 127.0.0.1 を指定して、ユーティリティ **smbpasswd** が正しく動作するようにしなくてはならない場合があります。localhost アドレスが必要かどうか判断する方法については、「Samba の複数インスタンスの実行 」を参照してください。

- **lock directory** (または lock dir) - 全サーバの Samba インスタンスで使用する一意のロック・ファイル・ディレクトリの名前と場所。これは、クラスタ内の全サーバに存在する既存のディレクトリでなくてはなりません。

  注記： この命令は **lock dir** と呼ばれることもあります。 Samba Recovery Kit はどちらの名前でも処理します。

- **bind　interfaces only** - この命令は、 **smbd** プロセスと **nmbd** プロセスが interfaces 命令で定義

されたアドレスに対する SMB リクエストだけを処理するようにします。これは Yes に設定する必要があります。 システムで動作している保護されていない Samba インスタンスでも、この命令を yes に設定する必要があります。 yes に設定すると、Samba は interfaces 命令で指定されていないサブネットに対するリクエストは処理せず、サーバ上で動作している別の Samba インスタンスに対するリクエストも処理しません。

# 6.16.4.3. 設定ファイルの [homes] セクション

[homes] セクションは、Samba サーバ上のユーザのホーム・ディレクトリが明示的に共用として定義されていない場合に、ホーム・ディレクトリに対する接続を処理する特別なセクションです。LifeKeeper では、この特別な共用を使用してユーザのホーム・ディレクトリを 保護することはありません 。したがって、 この共用は削除するかコメント・アウトしてください 。 LifeKeeper Samba Recovery Kit で Samba の共用を保護するには、path 命令を指定する必要があります。path 命令は LifeKeeper Samba 階層が保護するファイル・システムを決定するために使用されます。ユーザが Samba サーバに接続したときにパスが決定されるため、[homes] セクションのデフォルトではパスは指定されていません。この特別な共用を削除するかコメント・アウトする必要があるのは、このためです。

# 6.16.4.4. 設定ファイルの [printers] セクション

[printers] セクションは、Samba サーバ上のプリンタが具体的に共用として定義されていない場合に、プリンタへの接続を処理します。LifeKeeper はこの特別なセクションやグローバル命令 load printers を使用してプリンタの共用を保護することはありません。LifeKeeper が保護するプリンタの共用はそれぞれ設定ファイルの独自の共用セクションで定義する必要があります。

# 6.16.4.5. 設定ファイルの共用定義セクション

設定ファイルの残りの全セクションでは、このインスタンスに対してクライアントがアクセスできるファイルとプリンタの共用を定義します。設定ファイルでは共用を 1 つ以上定義する必要があります。Samba 設定ファイルでは、ファイルの共用のみ、プリンタの共用のみ、ファイルとプリンタの共用の両方の組み合わせのいずれも定義可能です。LifeKeeper の場合、定義可能な共用数に制限はありませんが、1 つの共用に関して障害が発生しただけで、階層全体がバックアップ・サーバに切り替えられることに注意する必要があります。各共用に対して次の命令を定義する必要があります。

- **path** - この命令はファイルまたはプリンタの共用のルート・パス名を指定します。この値によって、Samba 階層の一部として保護するファイル・システム・リソースが決まり ます。Samba リソース作成時に LifeKeeper ファイル・システム・リソースがすでに存在していない場合は、 LifeKeeper によって自動的に作成されます。

  注記： この命令は directory と呼ばれることもあります。Samba Recovery Kit はどちらの名前でも命令を処理します。

- **printable** - 値が Yes の場合、Samba の共用は Linux プリンタに印刷するためのプリント・スプール・レポジトリとして使用されます。共用を通常のファイルの共用として使用する場合は、この命令を No に設定するか、命令自体を指定しません。[global] セクションで Yes に設定されていない限り、この値のデフォルトは No です。この命令を Yes に設定した場合、下記の printer name 命令で定義したプリンタを保護する LifeKeeper プリント・サービス・リソースが Samba 階層作成時に存在している必要があります。

  注記： この命令は **print ok** と呼ばれることもあります。Samba Recovery Kit はどちらの名前でも命令を処理します。

- **printer name** - この命令は共用が使用するプリンタ名を定義し、名前付きプリンタを保護するプリント・サービス・インスタンスを探すときに使用されます。プリント・サービス・インスタンスは Samba 階層の子リソースになります。プリンタの共用に対してこ の命令が定義されていない場合、Samba Recovery Kit は共用名をプリンタ名と見なします。

  注記： この命令は **printer** と呼ばれることもあります。Samba Recovery Kit はどちらの名前 でも命令を処理します。

注記： Samba 設定ファイルでは、いくつかの命令で変数置換を使用できます。Samba ユーティリティ testparm で変数を解決できない限り、前述の命令では、変数置換を使用しないでください。

# 6.16.4.6. Samba の複数インスタンスの実行

LifeKeeper クラスタで Samba の複数インスタンスを実行する場合は、追加の設定要件と制限があります。次の Samba 設定の場合、Samba の複数インスタンスが関係してきます。

- アクティブ／スタンバイ設定で複数の LifeKeeper Samba インスタンスが 1 つのサーバ上にある場合
- アクティブ／アクティブ設定で複数の LifeKeeper Samba インスタンスが 1 つ以上のサーバ上にある場合

この設定のどちらの場合でも、LifeKeeper で保護されていない Samba インスタンスを含めることができます。

「LifeKeeperでのSambaの設定」で述べたように、Sambaの複数インスタンスを実行するときには、各インスタンスの設定ファイル名が異なっているか、異なるディレクトリに存在している必要があります。各設定ファイルにいくつか命令を記述する必要があり、その値は異なっていなくてはいけません。具体的には、netbios name、lock directory、pid directory、interfaces、およびlog fileです。これらの命令の値が一意でない場合、Sambaが起動せず、したがってクライアントが接続できなくなります。また、各インスタンスで指定したlock、log file、pidの各directoryはクラスタ内の全サーバに存在している必要があります。

## smbpasswd ユーティリティと Samba の複数インスタンス

LifeKeeper では必須ではありませんが、Samba Recovery Kit で使用する一部の Samba ユーティリティには、smb.conf がデフォルトの場所にあることを前提としているものがあります。Samba Recovery Kit は **smbclient** ユーティリティと **nmblookup** ユーティリティを使用してそれぞれ **smbd** と **nmbd** に接続し、LifeKeeper が保護しているデーモン・プロセスの状態を判断します。この 2 つのユーティリティは、デフォルトの場所に smb.conf が存在しなくてもエラー終了することはありません。しかし、**smbpasswd** ユーティリティの場合は、smb.conf がデフォルトの場所に存在する必要があります。

**smbpasswd** は、セキュリティ・レベルが share または user に設定されているときに、クライアント接続リクエストに対するユーザ認証を行う **smbpasswd** ファイルを管理するのに使用します。デフォルト設定ファイルが存在しない場合、Samba のパスワードの変更はすべて失敗します。この問題を回避するには、セキュリティ・レベルが share または user に設定されている場合、または、Samba セキュリティ・レベルが server に設定されているシステムに対する **smbpasswd** サーバとしてサーバが動作している場合には、Samba インスタンスの 1 つはデフォルト設定ファイルを使用する必要があります。これは、**smbpasswd** がデフォルト設定ファイルを使用して **smbpasswd** ファイルの場所を取得するためです。このため、LifeKeeper クラスタ内で **smbpasswd** ファイルが存在する場所は 1 か所だけに限られます。クラスタ内の Samba の全インスタンスの設定ファイルで、smb passwd file 命令を同じ値に設定する必要があります。また、**smbpasswd** ファイルはクラスタ内の全サーバで同期を保つ必要があります。

**smbpasswd** ユーティリティは、LifeKeeper Samba Recovery Kitでは必須のbind interfaces only命令の影響も受けます。bind interfaces only命令をYesに設定すると、Sambaパスワードを変更する通常ユーザはlocalhostアドレス 127.0.0.1 を使用して **smbd** デーモン・プロセスに接続しようとします。**smbd** デーモンが使用する設定ファイルのinterfaces命令にそのアドレスが追加されていて、**smbd** がそのアドレスに接続してリッスン中の場合、パスワードの変更は成功します。デーモンの設定ファイルのinterfaces命令

にそのアドレスが存在しない場合、パスワードの変更は失敗します。複数インスタンス環境でlocalhostが複数の設定ファイルで指定されている場合、起動して実行できるのは 1 つのインスタンスだ けです。**smbpasswd** で「 -r netbios_name」オプションを使用すると、インタフェース・リストにlocalhostアドレスを追加しなくても済みます(たとえば、 **smbpasswd -r server1 print1** …)。

注記：前述のように、保護されたSambaインスタンスが存在するLifeKeeperサーバ上で実行される保護されていないSambaインスタンスも、bind interfaces only命令をYesに設定する必要があります。

## Samba およびユーザ認証の考慮事項

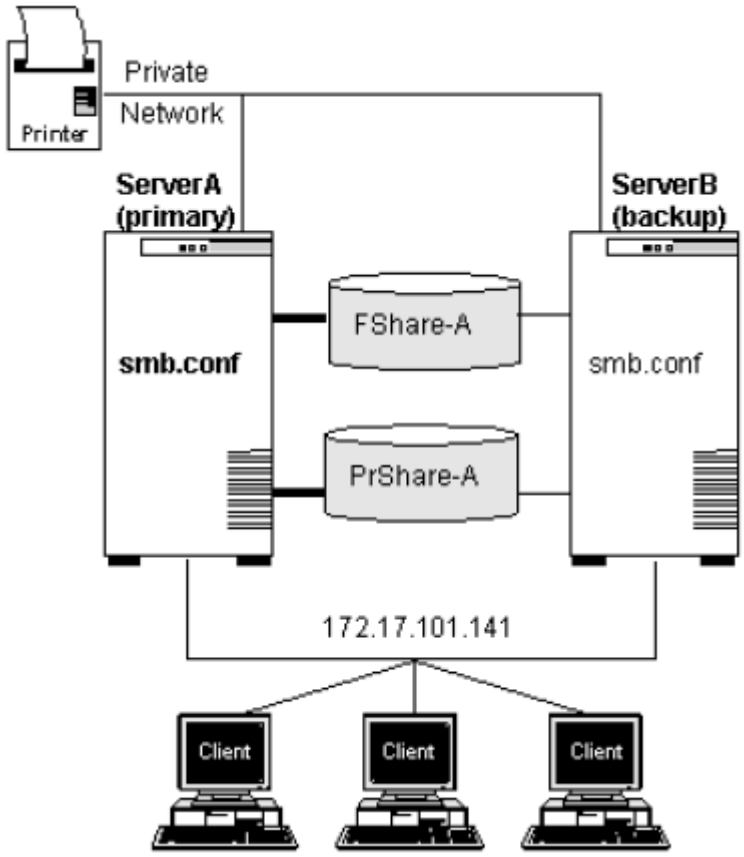
Samba は、ユーザ認証のために、クラスタ内の全 ての サーバ上で /etc/samba/smbpasswd または /etc/samba/secrets.tdb のようなデータファイルが一致しているのを保障するために、LifeKeeperwp 通じて Samba を保護する場合に考慮しなければならないセキュリティ・パラメータを通していくつかのメソッドをサポートします。(例えば、共有, ユーザ, ドメインなど…)。ユーザとしてセキュリティ・メソッドを使用する場合は、クラスタ内の全てのサーバで smbpasswd ファイルが一致していることを確認する必要があります。加えて、ドメインのようなセキュリティ・メソッドは secrets.tdb ファイルの一致が要求されます。 secrets.tdb ファイルの使用かつ LifeKeeper のアクティブ/アクティブ構成は、ファイルの配置場所を指定するために、プライベート dir パラメータの使用が必要となります。このパラメータの値はそれぞれの LifeKeeper Samba インスタンスにおいてユニークでなければなりません。

# 6.16.4.7. 構成例

このセクションでは一般的な Samba 設定の定義と例を示します。各例には、LifeKeeper に当てはまる設定ファイルの項目が含まれています。

## 構成 1：アクティブ／スタンバイ設定の例

アクティブ／スタンバイ設定では、ServerA がプライマリ LifeKeeper サーバです。プライ マリ・サーバは共有ストレージ・デバイスにあるファイルとプリンタの共用をエクスポートします。ServerB は別のアプリケーションやサービスを処理していてもかまいませんが、LifeKeeper に対しては Samba リソースのバックアップとしてだけ機能します。




設定に関する注記：

- クライアントは NetBIOS 名 LKServerA を使用して、保護された IP アドレス（172.17.101.141）経由で Samba サーバに接続します。 IP アドレスは設定ファイルの interfaces 命令で定義されています。

- 設定ファイル smb.conf は Samba リソース階層の拡張時に ServerB にコピーされます。設定ファイルには次の命令が記述されています。

  [global] netbios name = LKServerA

  bind interfaces only = yes

lock directory = /var/lock/samba interfaces = 172.17.101.141 127.0.0.1

log file = /var/log/sambaServA/log

[FShare-A]

path = /FShare-A

read only = no

public = yes

valid users =

printable = no create mode = 0664

directory mode = 0775

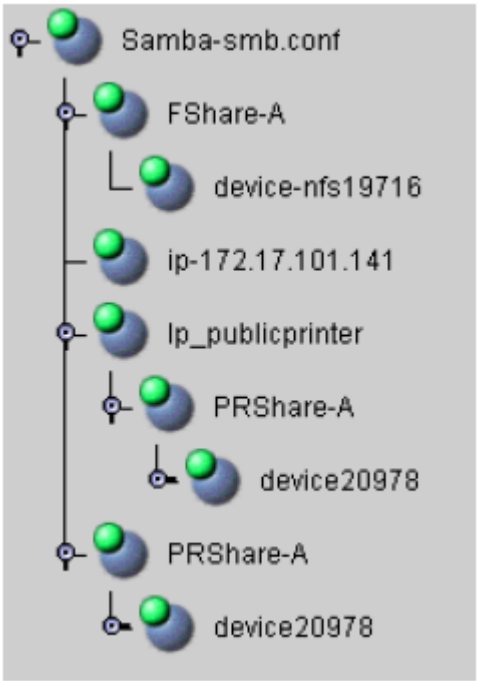
[PRShare-A]

path = /PRShare-A printer = publicprinter

printable = yes

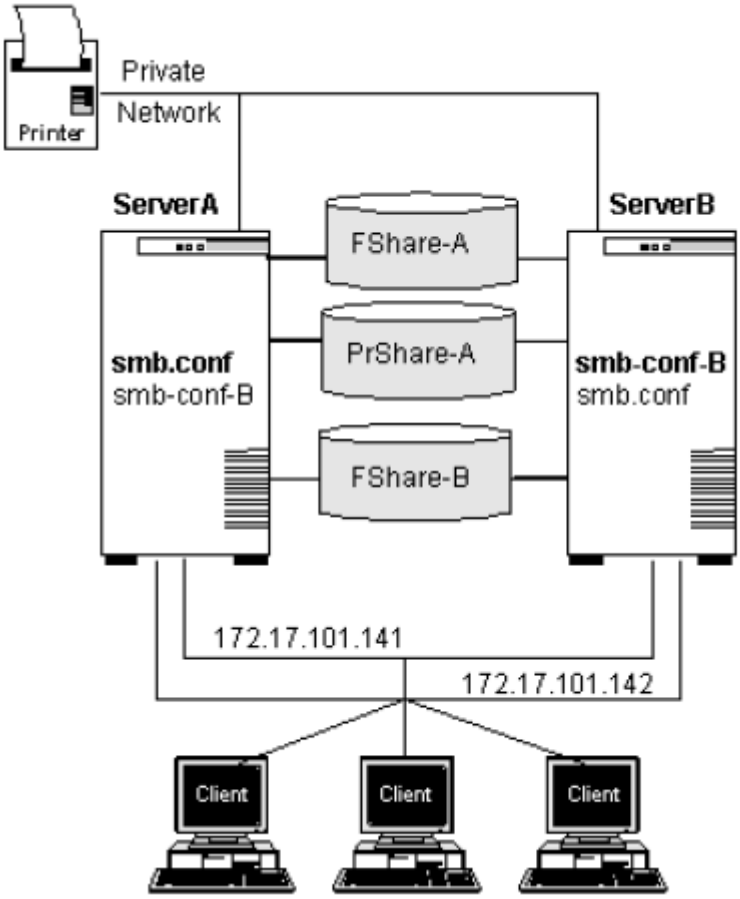
browseable = no

- Samba リソース階層は次のようになります。



# 構成 2：アクティブ／アクティブ設定の例

下図のアクティブ／アクティブ設定の場合、ServerA と ServerB の両方が Samba リソース のプライマリ LifeKeeper サーバです。各サーバは互いのバックアップ･サーバにもなります。



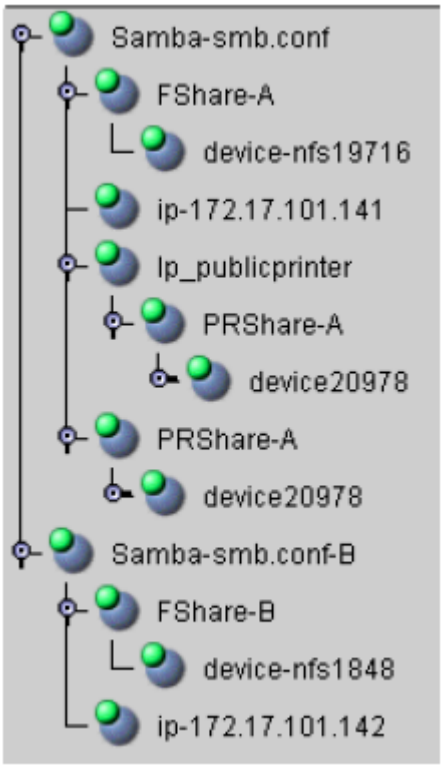

設定に関する注記：

- クライアントは NetBIOS 名 LKServerA と LKServerB を使用して、保護された IP アドレス (172.17.101.141 および 172.17.101.142) 経由で Samba サーバに接続します。IP アドレスは設定ファイルの interfaces 命令で定義されています。
- 設定ファイル smb.conf は Samba リソース階層の拡張時に ServerB にコピーされます。 同様に、設定ファイル smb.conf-B は Samba リソース階層の拡張時に ServerA にコピーされます。
- ServerA はファイルの共用/Fshare-A を保護します。ServerB はファイルの共用/Fshare- B を保護します。
- ServerA はプリンタの共用/publicprinter を保護します。ServerB はプリンタの共用を保護しません。
- 2 つの設定ファイルには次の命令が記述されます。

| smb.conf | smb-conf-B |
|---|---|

| | |
|---|---|
| [global]<br>netbios name = LKServerA bind<br>interfaces only = yes<br>lock directory = /var/lock/sambaServA<br>pid directory = /var/run/sambaServA<br>interfaces = 172.17.101.141 127.0.0.1<br>log file = /var/log/sambaServA/log<br>[FShare-A]<br>path = /FShare-A<br>read only = no<br>public = yes<br>valid users =<br>printable = no create mode = 0664 directory mode = 0775<br>[PRShare-A]<br>path = /PRShare-A<br>printer = publicprinter<br>printable = yes browseable = no | [global]<br>netbios name = LKServerB bind<br>interfaces only = yes<br>lock directory = /var/lock/<br>pid directory = /var/run/sambaServB<br>sambaServB interfaces = 172.17.101.142<br>log file = /var/log/sambaServB/log<br>[FShare-B]<br>path = /FShare-B<br>read only = no<br>public = yes<br>valid users =<br>printable = no create mode = 0664 directory mode = 0775 |

- Samba リソース階層は次のようになります。


Samba-smb.conf
FShare-A
device-nfs19716
ip-172.17.101.141
lp_publicprinter
PRShare-A
device20978
PRShare-A
device20978
Samba-smb.conf-B
FShare-B
device-nfs1848
ip-172.17.101.142

# 6.16.5. Samba の設定手順

このセクションでは Samba リソースの設定に必要な手順について説明します。

1. Samba の設定案を作成します。次のような項目を決定します。

   - 使用する NetBIOS 名
   - 保護して共用へのアクセスを許可するインタフェース
   - Samba の共用で使用し、したがって保護するファイルシステム
   - アクセスし保護するプリンタ(存在する場合)

   ロックディレクトリーとログディレクトリーの場所 使用する設定ファイルの数と設定の種類 (アクティブ／スタンバイかアクティブ／アクティブか)について検討します。たとえば、保護するSambaの共用が4つある場合、1 つの設定ファイルに4つの共用をすべて記述することもできますが、1 つのファイルシステムに障害が発生するだけで4つのファイルの共用すべてを含めてSamba階層全体のフェイルオーバが発生するというデメリットがあります。別の方法として、それぞれ1つのファイルの共用を保護する独立した設定ファイルを4つ作成することもできますが、この場合、4つの NetBIOS 名を定義して管理する必要があります。

2. Samba設定ファイルをステップ 1 で作成した設定案に従ってセットアップします。これには、[global] セクションに必要な命令や、使用するファイルとプリンタの共有などがあります。LifeKeeper Samba階層に必要なグローバル命令と共用命令については、「LifeKeeper での Sambaの設定 」を参照してください。

3. LifeKeeper で保護する IP アドレスを作成します。 この IP アドレスは、NetBIOS 名を使用してクライアントが Sambaサーバーに接続するのに使用されます。保護された IP アドレスは、設定ファイルの interface 命令に一致する必要があります。IP リソースのセットアップの詳細については、『LifeKeeper for Linux IP Recovery Kit 管理ガイド 』を参照してください。すべてのクライアントおよびその他のクラスターサーバー から ping を実行して、保護するIPアドレスをテストします。ローカルホスト(127.0.0.1) 用の保護される IP リソースは不要です。

4. Samba デーモンを起動して、クライアント接続をテストします。

   a. デーモンを起動するコマンドは次のとおりです。
      ```
      s/nmbd -D -s ConfigurationFile
      ```

   b. Samba ユーティリティーsmbclient を使用して、次のように smbd デーモンへの接続をテストします。これは interfaces 命令で定義された各アドレスに対して実行する必要があります。
      ```
      smbclient -L netbios_name -U% -I Protected_IP_Address
      ```

   c. **nmblookup** を使用して **nmbd** デーモンプロセスへの接続をテストします。これは各ブロードキャストアドレスに対して実行する必要があります。interfaces 命令に定義された各アドレスごとに、関連するブロードキャスト･アドレスを使用します。(ブロードキャスト･アドレスは、 **ip addr show** を実行すると取得できます。)
      ```
      nmblookup -B broadcast_address netbios_name
      ```

5. 1 つ前の手順で起動した Samba デーモンを停止します。これは **kill** コマンドを使用して行います。ps コマンドを使用して実行中のデーモン・プロセスを見つけ、 **kill pid** を発行してデーモンプロセスを終了します。

6. LifeKeeper 下で保護されるファイル・システム(Samba ファイルを管理する) を作成し、上記のステップで定義した共有を印刷します。(SPS for Linux テクニカルドキュメンテーションの「ファイルシステムリソース階層の作成 」を参照してください。) ファイルシステムリソースは、プリント・サービス・リソースまたは Samba リソース作成時に自動的に作成されるため、この手順は省略することもできます。Sambaリソースを作成する前に、レプリケーション(SIOS DataKeeper)ファイルシステムリソースを作成する必要があります。

7. 1 つのファイルシステムを複数の Sambaの共用で使用する場合は、保護するファイル・システム上に共用のためのディレクトリを作成します。

8. LifeKeeper 下で保護される印刷サービス階層 (設定ファイルにプリンタ共有が定義さ れている場合に、クライアントの印刷に使用される) を作成します。

9. LifeKeeperでSamba リソース階層を作成し、少なくとも 1 つのバックアップサーバーに拡張します (後述の「LifeKeeper設定作業 」を参照してください)。extend スクリプトによってSamba設定ファイルがテンプレート・サーバーからターゲットサーバーの同じ場所にコピーされます。

10. プライマリサーバー側で、サービス中のSamba階層が保護する共用へのクライアント接続をテストします。たとえば、Windows クライアントから共用ディレクトリにマッ ピングして、共用ファイルにアクセスできるか確認します。クラスタ内の全サーバーに対してテストを繰り返します。Samba リソースは、手動でバックアップサーバーへスイッチオーバする方法でテストすることもできます。(「リソース階層のテスト 」を参照してください。)

11. LifeKeeper で保護するSamba デーモンは LifeKeeper が制御するので、システム起動時の自動起動を無効化する必要があります。

# 6.16.6. Samba の LifeKeeper の設定作業

LifeKeeper for Linux Samba Recovery Kit の管理作業はすべて、LifeKeeper グラフィカル・ユーザ・インタフェース(GUI) で実行できます。LifeKeeper GUI では、Samba リソースの設定、管理、監視作業をガイド付きで行えます。

LifeKeeper for Linux Samba Recovery Kit の設定に関しては、次の作業が可能です。

- リソース階層の作成 ：Samba リソース階層を作成します。
- リソース階層の削除 ：Samba リソース階層を削除します。
- リソース階層の拡張 ：Samba リソース階層をプライマリ・サーバからバックアップ・サーバに拡張します。
- リソース階層の拡張解除 ：LifeKeeper クラスタの 1 つのサーバの Samba リソース階層を拡張解除 (削除) します。
- 従属関係の作成 ：既存のリソース階層と別のリソース・インスタンスとの間に子従属関係を作成し、クラスタ内のすべての対象サーバに従属関係の変化を通知します。
- 従属関係の削除 ：リソースの従属関係を削除して、クラスタ内のすべての対象サーバ に従属関係の変化を通知します。
- In Service ：リソース階層をアクティブにします。
- Out of Service ：リソース階層を非アクティブにします。
- 表示 / 編集 ： リソース階層のプロパティを表示または編集します。

**注記：** このセクションの残りの部分では、設定作業はすべて **[編集]** メニューから実行することになっています。この設定作業の大半は次の方法でも実行できます。

1. ツールバーから状態表示の左ペインにあるグローバル・リソースを右クリック。
2. 状態表示の右ペインにあるリソース・インスタンスを右クリック。

※右クリックの方法を使用すると、**[編集]** メニューを使用したときに必要な情報を入力しないですみます。

# 6.16.6.1. Samba のリソース階層の作成

必要なセットアップ作業を完了したあと、Samba リソース階層を作成するために、以下の手順を行います。

1. **[LifeKeeper GUI]** メニューから **[編集]** を選択し、次に **[サーバ]** を選択します。ドロップダウンメニューから、**[リソース階層の作成]** を選択します。

   **[リソース作成ウィザード]** ダイアログ･ボックスが表示され、クラスタにインストールされている全ての Recovery Kit がリストボックスに表示されます。

2. Samba の共用を選択して、**[次へ]** をクリックします。

3. 次の表で説明している情報を入力します。どのダイアログ･ボックスでも、**[戻る]** がアクティブになっているときには、直前のダイアログ･ボックスに戻ることができます。これは、エラーが発生して、前に入力した情報を修正する必要がある場合に便利な機能です。いつでも **[キャンセル]** をクリックして、作成処理全体を取り消すことができます。

| フィールド | ヒント |
|---|---|
| スイッチバック･タイプ | **[インテリジェント]** または **[自動]** を選択します。これによって、バックアップサーバへのフェイルオーバ後、Samba インスタンスがこのサーバでサービスに戻ったときにどのようにスイッチバックされるかが決まります。スイッチバック･タイプは、必要な場合、**[リソース･プロパティ]** ダイアログ･ボックスの **[一般]** タブで後から変更できます。<br><br>**注記：** スイッチバックの方法は、Samba リソースで使用するプリントサーバ、IP リソース、ファイル･システムリソースのスイッチバック方法と一致している必要があります。一致していない場合、Samba リソース作成時に、Samba リソースに対して選択された設定に一致するように設定がリセットされます。 |
| サーバ | 階層を作成するサーバを選択します。 |
| 設定ファイルのロケーション | Samba 設定ファイルを配置するディレクトリを選択します。 |
| 設定ファイル名 | このリソースの作成に使用する Samba 設定ファイルの名前を入力します。デフォルトは smb.conf です。<br><br>**注記：** LifeKeeper は選択された設定ファイルを読み込み、必要な命令が指定されていない場合、エラー･メッセージを表示します。設定ファイルの最小限のチェックも行います(共用が存在 |

| | |
|---|---|
| | するか、path 命令があるか、ロック・ディレクトリが指定されているか、そのディレクトリが存在するか検証)。その他のチェックも作成処理の途中で実行されます。 |
| ルート・タグ | LifeKeeper が示すデフォルト・ルート・タグを選択するか、このサーバ上のリソース・インスタンスの重複しない名前を入力します。デフォルトは Samba-configfilename です。 ここで、configfilename は対応する設定ファイルの名前です。英字、数字、および特殊文字 (- _ . /) を使用できます。 |

4. **[作成]** をクリックします。**[リソース作成ウィザード]** が表示され、Samba リソース階層が作成されます。LifeKeeper によって入力データが検証されます。問題が検出された場合、情報ボックスにエラーメッセージが表示されます。

5. 情報ボックスが表示され、Samba リソース階層の作成に成功したこと、フェイルオーバによって保護するには、クラスタ内の別のサーバにリソース階層を拡張する必要があることを示すメッセージが表示されます。**[次へ]** をクリックします。

6. **[続行]** をクリックします。**[事前拡張ウィザード]** が起動されます。リソース階層を別のサーバに拡張する方法の詳細については、次の「Samba リソース階層の拡張 」の手順 2 を参照してください。

**注記：** Samba 階層の作成に失敗する場合は 、「トラブルシューティング」の「Samba 階層の復元の失敗 」を参照してください。

# 6.16.6.2. Samba のリソース階層の拡張

この操作は **[編集]** メニューから開始できます。また、**[リソース階層の作成]** の処理を完了すると自動的に開始されます。その場合は、手順2を参照してください。

1. **[編集]** メニューの **[リソース]** から **[リソース階層の拡張]** を選択します。**[事前拡張ウィザード]** が表示されます。拡張操作に慣れていない場合は、**[次へ]** をクリックします。LifeKeeper の[リソース階層の拡張]作業のデフォルト値が分かっていて、入力と確認 を省略する場合は、**[デフォルトを受け入れる]** をクリックします。

2. **[事前拡張ウィザード]** で次の情報を入力します。

   **注記：** 最初の 2 つのフィールドは、**[編集]** メニューの **[拡張]** から操作を開始した場合にだけ表示されます。

| フィールド | ヒント |
|---|---|
| テンプレート・サーバ | Samba リソースが現在サービス中のサーバを入力します。 |
| 拡張するタグ | 拡張するSamba リソースを選択します。 |
| ターゲット・サーバ | 拡張先のサーバを入力または選択します。 |
| スイッチバック・タイプ | **[インテリジェント]** または **[自動]** を選択します。スイッチバック・タイプは、必要な場合、**[リソース・プロパティ]** ダイアログ・ボックスの **[一般]** タブで後から変更できます。<br><br>**注記：** スイッチバック方法は、Samba リソースが使用する従属リソースのスイッチバック方法と一致する必要があることを忘れないでください。 |
| テンプレートの優先順位 | テンプレート階層の優先順位を選択または入力します。優先順位は1～999 の範囲の未使用の値が有効で、数値が小さいほど優先順位は高くなります(1は最高の優先順位を表わします)。 拡張操作時には、別のシステムで既に使用中の優先順位をこの階層に対して指定することはできません。デフォルト値を推奨します。 |

| | |
|---|---|
| | **注記** ：これは階層を最初に拡張するときだけ表示されます。 |
| ターゲットの優先順位 | ターゲット・サーバの階層の優先順位を選択、または入力します。 |
| ルート・タグ | ターゲット・サーバの新しい Samba リソース・インスタンスのデフォルト・タグ名が表示されます。デフォルト・タグ名はテンプレート・サーバ上のこのリソースのタグ名と同じです。新しいタグ名を入力する場合は、ターゲット・サーバ上で重複しない名前にします。英字、数字、および特殊文字（- _ . /）を使用できます。 |

3. 拡張前のチェックが成功したというメッセージが表示されたら、**[次へ]** をクリックします。

4. 拡張する階層に応じて、拡張するリソース・タグを表示する情報ボックスが表示されます。リソース・タグは編集できません。**[拡張]** をクリックします。

5. 「Hierarchy extend operations completed (階層の拡張操作が完了しました)」というメッセージが表示されたら、階層を別のサーバに拡張する場合は、**[次のサーバ]** をクリックし、拡張作業が必要ない場合は、**[完了]** をクリックします。

6. 「Hierarchy Verification Finished (階層の検査が完了しました)」というメッセージが表示されたら、**[完了]** をクリックします。

# 6.16.6.3. Samba のリソース階層の拡張解除

LifeKeeper クラスタの1 つのサーバのリソース階層を削除するには、次の手順を実行します。

1. **[編集]** メニューの **[リソース]** から **[リソース階層の拡張解除]** を選択します。
2. Samba リソースを拡張解除する **[ターゲット・サーバ]** を選択します。リソースが現在サービス中の Samba サーバは選択できません。(このダイアログ・ボックスは、右側のペ インでリソース・インスタンスを右クリックして、 **[拡張解除]** を選択した場合には、表示されません。) **[次へ]** をクリックします。
3. 拡張解除するSamba 階層を選択し、 **[次へ]** をクリックします。(このダイアログ・ボックスは、いずれかのペインでリソース・インスタンスを右クリックして、 **[拡張解除]** を選択した場合には、表示されません。)
4. 拡張解除するために選択したターゲット・サーバとSamba リソース階層を確認する情報ボックスが表示されます。 **[拡張解除]** をクリックします。
5. Samba リソースの拡張解除に成功したことを示す別の情報ボックスが表示されます。 **[完了]** をクリックして、 **[リソース階層の拡張解除]** メニューを終了します。

# 6.16.6.4. Samba のリソース階層の削除

Samba 階層を削除すると、従属関係および保護されているサービスに何が起きるかを理解しておくことが重要です。

- **従属関係:** リソース階層を削除する前に、従属関係を削除したほうがよい場合があ ります。別の階層で使用されていない場合、従属しているファイル・システムは削 除されます。従属している IP リソースとプリント・サービス・リソースは、 LifeKeeper GUI または Samba 削除スクリプトを使用して削除する限り、削除されません。

- **保護されたサービス:** Sambaリソースを削除する前に Samba リソースを停止した場合、リソースを削除した後 Samba デーモンは停止します。Samba リソースの動作中にリソースを削除した場合、Samba デーモンは動き続けます。リソース階層の削除をした後は LifeKeeper の保護対象となりません。

LifeKeeper 環境内のすべてのサーバからリソース階層を削除するには、以下の手順を実行します。

1. **[編集]** メニューの **[リソース]** から **[リソース階層の削除]** を選択します。
2. Samba リソース階層を削除する **[ター ゲット・サーバ]** を選択し、 **[次へ]** をクリックします。 （このダイアログ・ボックスは、いずれかのペインでリソース・インスタンスを右クリックして、 **[リソースの削除]** を選択した場合には、表示されません。）
3. **[削除する階層]** を選択します。このダイアログ・ボックスは、左側または右側のペインでリソース・インスタンスを右クリックして、 **[リソースの削除]** を選択した場合には、表示されません。） **[次へ]** をクリックします。
4. 選択したターゲット・サーバと、削除の対象として選択した階層を確認する情報ボックスが表示されます。 **[削除]** をクリックします。
5. Samba リソースの削除に成功したことを示す別の情報ボックスが表示されます。
6. **[完了]** クリックして終了します。

# 6.16.6.5. Samba のリソース階層のテスト

手動で切り替えを行って、Samba リソース階層のテストを行うことができます。これは、プライマリ・サーバからバックアップ・サーバへのリソース・インスタンスのフェイルオーバをシミュレートします。

**[編集]** メニューの **[リソース]** から **[サービス中]** を選択します。たとえば、バックアップ・サーバ側で **[サービス中]** を実行すると、アプリケーション階層がプライマリ・サーバ側ではサービス休止になり、バックアップ・サーバ側でサービス中になります。この時点で、元のバックアップ・サーバがプライマリ・サーバになり、元のプライマリ・サーバがバックアップ・サーバになります。

**[サービス休止]** 要求を実行すると、アプリケーションは他のサーバでサービス中にならずに、サービスが停止されます。

# 6.16.7. Samba 階層の管理

Samba リソース階層の作成後、以後の Samba リソースの管理では以下の指針に従ってく ださい。

Samba 設定ファイルの修正

smbpasswd ファイルの管理

# 6.16.7.1. Samba 設定ファイルの修正

LifeKeeper Samba インスタンスで使用する Samba 設定ファイルを変更する必要があるときには、保護サービス中 (ISP) のサーバー上で、ここで説明する手順を実行します。設定ファイルの変更には次の 3 種類があります。

- Samba 階層に直接影響を与えない変更
- 階層に直接影響を与え、階層の削除と再作成は必要ない変更
- 階層に直接影響を与え、階層の削除と再作成が必要な変更必要な変更の種類を特定する方法と種類毎の手順については、以下のセクションを参照してください。

## Samba 階層に直接影響を与えない変更

Samba 階層に直接影響を与えない変更LifeKeeper によって使用されない設定ファイルの命令に対する変更は、このカテゴリに属します。(キットで使用する命令のリストについては、「LifeKeeper での Samba の設定 」を参照してください。) LifeKeeper が使用しない命令の例としては、security、hosts allow、hosts deny、valid users などがあります。手順は次のとおりです。

1. 設定ファイルを使用する Samba リソースをサービス休止にします。この手順は Samba デーモンを停止するために必要です。
2. Samba 設定ファイルに必要な変更を加えます。
3. クラスタ内の設定を同期します。この作業は、次のようにユーティリティ **synccfg** を使用して実行します。

   **LKROOT/lkadm/subsys/gen/samba/bin/synccfg -t TargetSys -c ConfigFile**

   ここで LKROOT は LifeKeeper のインストール場所(/opt/LifeKeeper)、TargetSys は更新するノード、ConfigFile はコピーする設定ファイルのフルパスです。
4. クラスタ内の全サーバーに対して 1 つ前の手順を繰り返します。
5. 階層をサービス中に戻して、Samba デーモンを再起動します。

## Samba 階層に直接影響を与える変更

LifeKeeperが使用する設定ファイル命令（一覧は「LifeKeeper での Samba の設定 」を参照）の変更がこの種類に該当します。 ただし、netbios nameまたは設定ファイルの物理的移動を除きます。変更の大きさによっては、3 種類目と同様に階層を再作成するほうが簡単で作業が早く済むこともあります。この種類に該当する典型的な変更としては、新 しいファイルとプリンターの共用の追加、ファイルとプリンターの共用の削除、IPインタフ ェースの追加または削除などがあります。

1. 設定ファイルを使用する Samba リソースをサービス休止にします。この手順は Samba デーモンを停止するために必要です。

2. Samba 設定ファイルに必要な変更を加えます。

3. Samba 階層に必要な変更を加えます。これは設定ファイルに加える変更の種類によって異なります。たとえば、以下のようになります。

   - º interfaces 命令に IP アドレスを追加した場合は、新しい IP リソースを作成して拡張し、従属した子として Samba リソース階層に追加する必要があります。SPS for Linux Technical Documentation の

リソース依存関係の作成 を参照してください。

   - º 設定ファイルにファイルの共用を追加した場合は、ファイルシステムリソースを作成して拡張し、従属した子として Samba リソース階層に追加しなくては ならない場合があります。ファイルシステムリソースがすでに子として階層 に存在している場合は(たとえば、新しい共用のために定義された path 命令が 別のファイルの共用またはプリンターの共用と同じファイルシステムマウント トポイントである場合)、従属した子として作成し追加する必要はありません。

   - º プリンターの共用を追加する場合、ファイルシステムリソースとプリントサービスリソースを作成して拡張し、従属した子として Samba 階層に追加する 必要があります。print share name 命令または printer/printer name 命令で定義され たプリンターを保護するプリントサービスリソースが存在しない場合は、それ を作成する必要があります。ファイルシステムリソースを追加する必要があ るかどうかは、前述のファイルの共用の場合を参照してください。

   - º ファイルの共用またはプリンターの共用を削除した場合、または、interfaces 命令から IP アドレスを削除した場合は、Samba 階層の従属関係を削除してから、個々 のリソースを削除します。

   - º プリンターの共用名を変更する場合は、プリンターの共用を削除してから新しいプリンターの共用を追加します。

4. クラスタ内の設定を同期します。この作業は、次のようにユーティリティ **synccfg** を使用して実行します。

   **LKROOT/lkadm/subsys/gen/samba/bin/synccfg -t TargetSys -c ConfigFile**

   ここで LKROOT は LifeKeeper のインストール場所 (/opt/LifeKeeper)、TargetSys は更新するサーバー、ConfigFile はコピーする設定ファイルのフルパスです。

5. クラスタ内の全サーバーに対して 1 つ前の手順を繰り返します。

6. 階層をサービス中に戻して、Samba デーモンを再起動します。

   **注意：** 多数のリソースの作成と従属関係の追加または削除が必要な変更を行う場合は、Samba 階層をサービス休止にする前に新しいリソースをすべて作成しておき、システムの停止時間を最小限に抑えたほうがよい場合があります。

## 階層に直接影響を与え、階層の削除と再作成が必要な変更

netbios name 命令を変更するか設定ファイルの物理的場所を変更する場合は、次の手順を 実行します。

1. 階層を削除します。（詳細は「Samba 階層の削除」を参照してください。）
2. NetBIOS 名を変更します。 または設定ファイルを移動します。
3. 新しい Samba 階層を作成して、すべてのバックアップサーバーに拡張します。

# 6.16.7.2. smbpasswd ファイルの管理

Sambaでは、security命令で 4 つの異なる認証方法を指定できます。セキュリティ・レベ ルが share の場合もuserの場合も、ローカルの smbpasswd ファイルを参照してアクセスが許可されているか判断する必要があります。「Samba の複数インスタンスの実行 」で述べているように、smbpasswd ファイルは 1 つしか存在できず、そのため LifeKeeper クラスターでは管理上の問題が発生する可能性があります。セキュリティ・レベルとしてshare または userを選択した場合は、全サーバーでファイルを同期して、フェイルオーバー後も認証が成功するようにしておく必要があります。

Samba 階層が 1 つしかないクラスターでは、セキュリティ・レベルを share または user にした場合、smbpasswd ファイルを設定ファイルで定義したファイルの共用に配置することができます。この共用に対しては、管理者だけがアクセスできるようにしておきます。複数インスタンスが存在する場合は、セキュリティ・レベルとして server または domain を使用することを推奨します。

> 注記: ファイアーウォールを使用している場合は、ファイアーウォールが **smbd** デーモンへの 接続を許可し、 **nmblookup** が動作することを確認してください。

# 6.16.8. Samba トラブルシューティング

## Samba 階層の復元の失敗

Samba 階層の復元が失敗しても、デーモン・プロセス **smbd** および **nmbd** は実行されたままである場合があります。(復元操作は、作成、フェイルオーバ、または手動のスイッチオーバの完了によって開始されるか、または **smbd** および **nmbd** への接続障害が原因のローカル・リカバリーによって開始されます。) 問題を修正した後も、デーモンの停止と再起動が実行されない場合、復元をもう一度行おうとしても、復元は再び失敗します。

**推奨される対策：**

接続に失敗する原因を解消します (たとえば、interfaces 命令のマスク設定の間違い)。 次に、手動で **smbd** デーモンと **nmbd** デーモンを停止します。階層をサービス中に戻します。 作成時に復元に失敗した場合は、階層を再作成します。

デーモンを停止すると、復元時に確実に設定ファイルが再読み込みされます。

# 6.16.8.1. Samba のエラーメッセージ

## 共通エラーメッセージ

| エラー番号 | エラーメッセージ | 説明 |
|---|---|---|
| 109009 | Error getting netbios name from the instance information field for tag "Samba-smb.conf" on server "ServerA". (サーバ"ServerA"上のタグ"Samba- smb.conf"用インスタンス情報からの netbios 名の取得時にエラー。) | インスタンス情報から NetBIOS 名を抽出するのに失敗しました。情報フィールドに設定ファイル名と NetBIOS 名が含まれている か確認してください。 |
| 109015 | The Samba utility testparm failed. Unable to parse Samba configuration file.(Samba ユーティリティ、testparm の実行に失敗しました。Samba 構成ファイルの構文解析ができません。) | 設定ファイルの解析に使用した Samba ユーティリティ **testparm** の実行に失敗しました。コマンド･ラインから、階層で使用する設定ファイルを指定して **testparm** を実行し、失敗の内容を確認してください。 |
| 109019 | Failed to initialize for reading of the Configuration File "/tmp/smb.ini.1234". (設定ファイル"/tmp/smb.ini.1234"の読み込み作業の初期化に失敗しました。) | 生成された設定ファイルの読み込みに失敗しました。ユーティリティ **testparm** が生成したファイルに間違いがあります。 |
| 109022 | Error getting configuration name from the instance information field for tag "Samba-smb.conf" on server "ServerA". (サーバ"ServerA"上のタグ "Samba- smb.conf" 用インスタンス情報からの設定名の取得時にエラー。) | インスタンス情報から NetBIOS 名を抽出するのに失敗しました。情報フィールドに設定ファイル名と NetBIOS 名が含まれているか確認してください。 |
| 109030 | Failure opening "/var/lock/samba/smbd.pid" on server "ServerA": "File Not Found". (サーバ"ServerA"上の "/var/lock/samba/smbd.pid"のオープンに 失敗しました："ファイルが見つかりません"。) | 表示している理由によってデーモン･プロセス ID ファイルのオープンに失敗しました。表示されたエラー･コードに基づいて問題を修正してください。 |
| 109050 | Open of the testparm output file failed. (testparm 出力ファイルのオープンに失敗しました。) | ファイルが存在しないか、ファイルにデータが存在しないため、 **testparm** を実行して作成した出力ファイルのオープンに失敗しました。コマンド･ラインから、階層で使用する設定ファイルを指定して **testparm** を実行し、失敗の内容を確認してください。 |

# 6.16.8.2. リソース階層作成時のエラーメッセージ

| エラー番号 | エラーメッセージ | 説明 |
|---|---|---|
| 109001 | Usage: "valid_cf"CfgPath CfgName TemplateSys<br><br>(使用法："valid_cf" CfgPath CfgName TemplateSys) | **valid_cf** スクリプトは、設定ファイルのあるディレクトリ、設定ファイル名、設定ファイルを検証するテンプレート・システム名という 3 つの引数を必要とします。3 つすべてを指定する必要があります。 |
| 109002 | Must specify an absolute path to the "smb.conf" configuration file. ("smb.conf" 設定ファイルの絶対パスを指定してください。) | **choice_cf** スクリプトと **valid_cf** スクリプトを実行するときには、設定ファイルの絶対パスを指定する必要があります。 |
| 109003 | The file "smb.conf" does not exist in "/etc". (ファイル”smb.conf”が”/etc” にありません。) | **valid_cf** スクリプトを実行して指定の設定ファイルを検証するときには、設定ファイルの正しいパスを指定する必要があります。 |
| 109004 | The path "/export/fs" in "/etc/samba/smb.conf" does not reside on a shared file system. (/etc/samba/smb.conf" 内のパス "/export/fs" が共有ファイル・システム上にありません。) | Samba 設定の共有には LifeKeeper がファイル・システム・リソース経由で保護できる path 命令が含まれている必要があります。設定ファイルを編集して、LifeKeeper が保護できるファイル・システムのパスを変更してください。 |
| 109005 | Usage: "choices_cf" CfgFile (使用法：“choices_cf” CfgFile) | **choice_cf** スクリプトは既存の設定ファイルのフルパスを必要とします。正しいパスを指定してください。 |
| 109006 | Samba Configuration file not specified. (Samba 設定ファイルが指定されていません。) | Samba リソース階層の作成に使用する 設定ファイルが指定されていません。 |
| 109007 | Cannot bring hierarchy "Samba-smb.conf" in service on server "ServerA".(階層 "Samba-smb.conf" をサーバ "ServerA" でサービス中にできません。)<br><br>**処置：** 問題を修正した後、手動で階層をサービス中にします。 | 作成の最後に階層をサービス中にできませんでした。ログ・ファイルを参照して、失敗の原因を調べてください。 |
| 109010 | The Samba configuration file does not have the interfaces directive defined. This directive is required to create Samba | 設定ファイルに interfaces 命令が記述されていません。 または、 |

| | | |
|---|---|---|
| | File Share hierarchies.(Samba 設定ファイルに interfaces 命令が定義されていません。この命令は Samba のファイル共有階層を作成するには必須です。) | interfaces 命令で localhost (127.0.0.1) 以外の IP アドレスが指定されていません。 |
| 109011 | The Samba configuration file does not have a correctly formatted interfaces directive. The interfaces directive must be in full dotted decimal IP address format with or without the mask parameter.(Samba 設定ファイルの interfaces 命令の形式が正しくありません。interfaces 命令は、マスク・パラメータの有無にかかわらず、ドット形式 10 進 IP アドレスで記述する必要があります。) | interfaces 命令では、aaa.bbb.ccc.ddd または aaa.bbb.ccc.ddd/mask という形式で 1 つ以上の IP アドレスをスペースで区切って指定する必要があります。 |
| 109012 | The Samba configuration file section "FileShare1" does not have a path directive defined. All Samba shares must have a path directive.(Samba 設定ファイルのセクション "FileShare1"に path 命令が定義されていません。すべての Samba の共有に path 命令を記述する必要があります。) | 設定ファイルで指定された共有に path 命令が記述されていないか、path 命令 の値が指定されていません。 |
| 109014 | No IP resources defined on server "ServerA".(サーバ "ServerA" に IP リソースが定義されていません。)<br><br>**処置：** Samba 設定ファイルの interfaces 命令で定義された IP アドレス用の IP リソースを作成します。 | 指定されたサーバに、Samba リソース階層の作成に必要な LifeKeeper が保護する IP リソースが存在しません。 |
| 109016 | The IP(s) "100.25.104.25,100.35.104.26 defined in the interfaces directive are not under LifeKeeper protection.(interfaces 命令で定義された IP(s)、"100.25.104.25,100.35.104.26"が LifeKeeper によって保護されていません。)<br><br>**処置：** Samba 設定ファイルの interfaces命令で定義された保護されていないアドレス用の IP リソースを作成します。 | interfaces 命令で指定された IP アドレス には、LifeKeeper が保護する IP リソースが存在する必要があります。表示された IP アドレスは LifeKeeper で保護されていません。 |
| 109017 | Missing configuration file name.(設定ファイルが存在しません。) | Samba リソースの作成時に **testparm** を 実行しようとしたときに、設定ファイルが存在していませんでした。 |
| 109018 | No Samba shares found in "/etc/samba/smb.conf".("/etc/samba/smb.conf"に Samba の共有が見つかりませんでした。) | リソースに対して指定された Samba 設定ファイルには少なくとも 1 つファイルの共有またはプリンタの共有を含んでいる必要があります。 |
| 109020 | Bad configuration file. No section information found in the file "/tmp/smb.ini.1234".(設定ファイルが間違っています。ファイル"/tmp/smb.ini.1234" にセクション情報が見つかりません。) | リソースに対して指定された Samba 設定ファイルには少なくとも 1 つファイルの共有またはプリンタの共有を含んでいる必要があります。 |
| 109021 | Creation of Samba hierarchy with tag "Samba-smb.conf" on | Samba 階層の作成に失敗しまし |

| | | |
|---|---|---|
| | server "ServerA" failed.(サーバ "ServerA"上のタグ "Samba-smb.conf" の Samba 階層の作成に失敗しました。) | た。他のエラー・メッセージを参考にして、失敗の原因を判断してください。 |
| 109023 | The file system resource "filesys1328" is not in-service on server "ServerA".(ファイル・システム・リソース"filesys1328"がサーバ"ServerA"でサービス中ではありません。) | Samba 階層の従属している子として必要なファイル・システム・リソースステンプレート・サーバでサービス中ではありません。リソースをサービス中にして、リソースの作成をもう一度行ってください。 |
| 109024 | Selected IP resource "ip-100.25.104.26" does not exist on server "ServerA".(選択した IP リソース"ip-100.25.104.26" がサーバ "ServerA" に存在しません。)<br><br>**処置：** 操作をやり直します。 | 指定された IP リソース・タグはもう存在しませんが、Samba 階層の作成には必要です。IP リソースを再作成して、リソースの作成をもう一度行ってください。 |
| 109025 | LifeKeeper was unable to create a dependency between the Samba hierarchy "Samba-smb.conf" and the IP resource "ip-100.25.104.26" on server "ServerA".(Samba 階層 "Samba-smb.conf"とサーバ "ServerA"上の IP リソース"ip-100.25.104.26" との間に従属関係を作成できませんでした。) | Samba リソースと IP リソース間の従属関係の作成に失敗しました。他のメッセージを参考にして、失敗の原因を判断してください。 |
| 109026 | The Samba configuration file does not have a netbios name directive defined.(Samba 設定ファイルに netbios name 命令が定義されていません。)<br><br>**処置：** 設定ファイルの global セクションに netbios name 命令を追加してください。 | すべての設定ファイルに NetBIOS 名が記述されている必要があります。設定ファイルに NetBIOS name 命令を追加してください。 |
| 109029 | The Samba configuration file "%s" directive defines a directory that does not exist. The %s can contain pid directory,lock dir, or lock directory.(Samba 設定ファイルの "%s" 命令が存 在しないディレクトリを定義しています。%s は pid directory または lock directory (lock dir) である可能性があります。) | すべての設定ファイルには、この命令によって指定される既存のディレクトリが記述されている必要があります。存在しないディレクトリを追加してください。 |
| 109035 | The Samba directive "bind interfaces only" must be set to "Yes".(Samba 命令"bind interfaces only" を "Yes" に設定する必要があります。)<br><br>**処置：** "bind interfaces only"の値を "Yes"に変更して、階層を再作成します。 | すべての設定ファイルで bind interfacesonly 命令が Yes に設定されている必要があります。命令を正しく "Yes" に設定してください。 |
| 109036 | Selected Printer resource "lp-admin" does not exist on server "ServerA". (選択したプリンタ・リソース "lp-admin"がサーバ"ServerA"に存在しません。)<br><br>**処置：** 操作をやり直します。 | 指定されたプリント・サービス・リソース・タグはもう存在しませんが、Samba 階層の作成には必要です。プリント・サービス・リソースを再作成して、リソースの作成をも |

| | | |
|---|---|---|
| | | う一度行ってください。 |
| 109037 | LifeKeeper was unable to create a dependency between the Samba hierarchy "Samba-smb.conf" and the Printer resource "lp-admin" on server "ServerA".(Samba 階層 "Samba-smb.conf" とサーバ "ServerA"上のプリンタ・リソース"lp-admin" との間に従属関係を作成できませんでした。) | Samba リソースとプリント・サービス・リソース間の従属関係の作成に失敗しました。他のメッセージを参考にして、失敗の原因を判断してください。 |
| 109038 | The Printers(s) "lpadmin" defined in the configuration file are not under LifeKeeper protection.(設定ファイルで定義されているプリンタ "lpadmin" が LifeKeeper によって保 護されていません。)<br><br>**処置：** Samba 設定ファイルで定義された保護されていないプリンタ用のプリンタ・インスタンスを作成します。 | 設定ファイルで定義されたプリンタには、LifeKeeper が保護するプリント・サービス・リソースが存在する必要があります。表示されたプリンタは LifeKeeper で保護されていません。 |
| 109041 | The selected configuration file "/etc/samba/smb.conf" is in use by Samba resource "Samba-smb.conf".(選択した設定ファイル "/etc/samba/smb.conf" は Samba リソース "Samba-smb.conf" が使用中です。)<br><br>または<br><br>The selected netbios name "LKServer" is in use by Samba resource "Samba-smb.conf".(選択した netbios 名 “LKServer” は Samba リソース"Samba-smb.conf"が使用中です。) | Samba 設定ファイルまたは netbios 名はクラスタ内で 1 度だけしか保護できません。設定ファイル名を変更するか、新しい NetBIOS 名を選択します。 |

# 6.16.8.3. リソース拡張時のエラーメッセージ

| エラー番号 | エラーメッセージ | 説明 |
|---|---|---|
| 109008 | Replication of config file to target server "ServerB" failed. The "mkdir" of "/etc/samba/config_files" failed. (ターゲット・サーバ"ServerB" に設定ファイルを複製するのに失敗しました。"/etc/samba/config_files"の”mkdir” に失敗しました。)<br><br>または<br><br>Replication of config file to target server "ServerB" failed. The "/etc/samba/config_files" failed. (ターゲット・サーバ"ServerB" に設定ファイルを複製するのに失敗しました。"/etc/samba/config_files" の "lcdrcp" に失敗しました。) | 拡張時に Samba 設定ファイルのコピーに失敗しました。 **mkdir** またはリモート・コピーに失敗しています。 |
| 109042 | WARNING: The configuration file "/etc/samba/smb.conf" currently exists on server "ServerB" and will be overwritten if this resource is extended. (警告：設定ファイル "/etc/samba/smb.conf" は現在サーバ "ServerB"上に存在し、このリソースを拡張すると上書きされます。) | リソース階層で使用する設定ファイルが既にバックアップ・サーバに存在しており、上書きされます。ファイルの上書きをしない場合は、拡張処理をキャンセルします。 |

# 6.16.8.4. リソース復元時のエラーメッセージ

| エラー番号 | エラーメッセージ | 説明 |
|---|---|---|
| 109027 | Failed start of smbd as daemon process. (smbd をデーモン･プロセスとして起動するのに失敗しました。) | **smbd** をデーモンとして起動するのに失敗しました。Samba ログ･ファイルで 追加の情報を確認してください。 |
| 109028 | Failed start of nmbd as daemon process. (nmbd をデーモン･プロセスとして起動するのに失敗しました。) | **nmbd** をデーモンとして起動するのに失敗しました。Samba ログ･ファイルで追加の情報を確認してください。 |

# 6.16.8.5. リソース削除時のエラーメッセージ

| エラー番号 | エラーメッセージ | 説明 |
|---|---|---|
| 109033 | Failed start of smbd as daemon process.Attempting to stop via SIGKILL. (smbd をデーモン・プロセスとして起動するのに失敗しました。SIGKILL による停止を試みます。) | **smbd** デーモン・プロセスの通常終了に失敗したため、デーモンは強制終了されます。 |
| 109034 | Failed to stop nmbd daemon process. Attempting to stop via SIGKILL. (nmbd デーモン・プロセスの停止に失敗しました。SIGKILL による停止を試みます。) | **nmbd** デーモン・プロセスの通常終了に失敗したため、デーモンは強制終了されます。 |

# 6.16.8.6. リソース監視時のエラーメッセージ

| エラー番号 | エラーメッセージ | 説明 |
|---|---|---|
| 109031 | Connection attempt to smbd daemon on address "100.25.104.26" failed. (アドレス"100.25.104.26" の smbd デーモンへの接続に失敗しました。) | 表示された IP アドレスへの接続に失敗したため、 **smbd** デーモン・プロセスのヘルス・チェックに失敗しました。ローカル・リカバリーが試みられます。 ローカル・リカバリーに失敗した場合、階層はバックアップ・サーバに切り替えられます。 |
| 109032 | Connection attempt to nmbd daemon failed for broadcast address "100.25.107.255" using netbios name "FILESERV1". (netbios 名"FILESERV1" を使用したブロードキャスト・アドレス "100.25.107.255" による nmbd デーモンへの接続に失敗しました。) | 表示されたブロードキャスト・アドレスへの接続に失敗したため、 **nmbd** デーモン・プロセスのヘルス・チェックに失敗しました。ローカル・リカバリーが試みられます。ローカル・リカバリーに失敗した場合、階層はバックアップ・サーバに切り替えられます。 |
| 109039 | No dependent IP resources were found for tag "Samba-smb.conf" on server "ServerA". (サーバ "ServerA" 上のタグ "Samba-smb.conf" に従属した IP リソースが見つかりませんでした。) | **nmbd** デーモンのヘルス・チェックを行おうとしたときに、Samba 階層に IP の子リソースが見つかりませんでした。ログを調べて、失敗の原因を判断してください。 |
| 109040 | Unable to determine IP address and/or mask for IP resource "ip-100.25.104.25". (IP リソース"ip-100.25.104.25" の IP アドレスまたはマスクを決定できません。) | IP リソース情報から IP アドレスとマスクを抽出するのに失敗しました。情報フィールドに IP アドレスとマスクが含まれているか確認してください。 |

# 6.16.8.7. 設定ファイル同期ユーティリティのエラーメッセージ

| エラー番号 | エラーメッセージ | 説明 |
|---|---|---|
| 109013 | The hierarchy for the specified configuration file does not have an equivalency with the target system. Select another target system or configuration file. (指定された設定ファイルの階層にターゲット・システムとのイクイバレンシがありません。別のターゲット・システムまたは設定ファイルを選択してください。) | **synccfg** ユーティリティの「-targetSys」引数で指定されたシステム にローカル・システム上の Samba 階層 に対するイクイバレンシがありません。 |
| 109044 | Usage: synccfg -t TargetSys -c ConfigFile(使用法：synccfg -t TargetSys -c ConfigFile) | **synccfg** ユーティリティは引数を 2 つ必要とします。「-t TargetSys」は設定ファイルのコピー先で、「-c ConfigFile」はコピーする設定ファイルの名前です。両方の引数を指定する必要があります。 |
| 109045 | Specified Configuration file does not exist on this server.(指定された設定ファイルはこのサーバに存在しません。) | **synccfg** ユーティリティの「-cConfigFile」引数で指定された設定ファイルがこのシステムには存在しません。 |
| 109046 | The target system specified for updating is the same as the template system. Select a new target system.(更新のために指定されたターゲット・システムがテンプレート・システムと同じです。別のターゲット・システムを選択してください。) | **synccfg** ユーティリティの「-TargetSys」引数で指定されたコピー先がローカル・サーバになっています。 ターゲット・サーバはローカル・システムであってはいけません。 |
| 109047 | The specified configuration file is not used in any Samba instances. Select another configuration file.(指定された設定ファイルは Samba インスタンスで使用されていません。別の設定ファイルを選択してください。) | **synccfg** ユーティリティの「-cConfigFile」引数で指定された設定ファイルがどの LifeKeeper Samba インスタンスでも使用されていません。このユーティリティでコピーできるのは、LikeKeeper で保護された設定ファイルだけです。 |
| 109048 | The syncronization of the Samba configuration file failed with a status of "%s".(Samba 設定ファイルの同期に失敗しました。ステータスは"%s"です。) | ローカル・システムとターゲット・システム間の設定ファイルの同期に失敗しました。ステータスは失敗した理由を示しています。 |
| 109049 | The hierarchy for the specified configuration file is ISP on the target node. Select another target system or configuration file.(ターゲット・ノートで ISP です。別のターゲット・システムまたは設定ファイルを選択してください。) | synccfg ユーティリティの「-tTargetSys」引数で指定されたシステムで Samba 階層が現在 ISP です。ターゲット・システムは ISP であってはなりません。 |

# 6.17. SAP Recovery Kit 管理ガイド

LifeKeeper for Linux SAP Recovery Kit は、LifeKeeper 環境で障害の発生したプライマリサーバからバックアップサーバに SAP NetWeaver を復旧する仕組みを提供します。SAP Recovery Kit は、他の LifeKeeper Recovery Kit (IP Recovery Kit 、NFS Server Recovery Kit 、NAS Recovery Kit 、および Oracle Recovery Kit などのデータベースリカバリキット) と連係して包括的なフェイルオーバ保護を提供します。

本ドキュメンテーションには、SAP リソースの設定と管理に不可欠な情報が記載されています。SPS を正しく実装させるには、設定手順を慎重に実行してください。また、関連するリカバリキット のドキュメンテーションも参照してください。

SAP Recovery Kit の概要

## ドキュメンテーション一覧

以下は、SIOS Technology Corp が発行している LifeKeeper 関連情報の一覧です。

- LifeKeeper for Linux リリースノート
- LifeKeeper for Linux テクニカルドキュメンテーション (LifeKeeper GUI の **[Help]** メニューからも参照できます)
- SIOS Technology Corp. Documentation and Support
- 略語と定義 - 本ドキュメンテーション全体で使用される略語と用語の一覧およびその意味。
- LifeKeeper/SAP のアイコン - 使用されるアイコンの一覧およびその意味。

## 参考ドキュメント

以下は、本ドキュメンテーション全体で参照した SAP 関連のドキュメントです。

- *SAP R/3 in Switchover Environments* (SAP ドキュメント 50020596)
- *R/3 Installation on UNIX:* (データベース固有)
- *SAP Web Application Server in Switchover Environments*
- *Component Installation Guide SAP Web Application Server* (データベース固有)
- *SAP Notes 7316, 14838, 201144, 27517, 31238, 34998 and 63748*

# 6.17.1. SAP 略語と定義

本マニュアル全体で、使用される略語は次のとおりです。

| 略語 | 意味 |
|---|---|
| **AS** | SAP アプリケーションサーバ。一般的には、AS は任意のアプリケーションサーバを指すが、本マニュアルにおいては、CI 以外の冗長化されたアプリケーションサーバを指す。そのため、アプリケーションサーバには LifeKeeper による保護は必要ない。 |
| **ASCS** | ABAP SAP セントラルサービスインスタンス。NetWeaver ABAP 環境向けのメッセージサービスと ENQEUE サービスで構成される SAP インスタンスである。このインスタンスは、単一障害点であり、LifeKeeper で保護する必要がある。 |
| **(ASCS)** | バックアップ ABAP SAP セントラルサービスインスタンスサーバ。プライマリ ASCS サーバに障害が発生した場合の ASCS ホストサーバである。 |
| **DB** | SAP データベースインスタンス。このデータベースは、Oracle またはその他の SAP がサポートするデータベースである。このインスタンスは、単一障害点であり、LifeKeeper で保護する必要がある。CI と DB は同一のサーバ上にあっても、異なるサーバ上にあってもよい。また、DB は、プライマリ DB サーバを指すこともある。 |
| **(DB)** | バックアップデータベースサーバ。プライマリ DB サーバに障害が発生した場合の DB ホストサーバである。1 つのサーバを データベース (DB) とセントラルインスタンス (CI) の両方のバックアップにすることも可能。 |
| **ENSAv1** | スタンドアローンエンキューサーバー バージョン1。これは、7.51以前のSAPカーネルバージョンで使用可能なエンキューサーバーのバージョンです。 |
| **ENSAv2** | スタンドアローンエンキューサーバー バージョン2。このバージョンのエンキューサーバーは、SAPカーネルバージョン7.51以降で使用できます。 |
| **ERS** | ENQUEUE レプリケーションサーバ。 |
| **ERSv1** | エンキューレプリケーションサーバー バージョン1。これは、7.51以前のSAPカーネルバージョンで使用可能なエンキューレプリケーションサーバーのバージョンです。 |
| **ERSv2** | エンキューレプリケーションサーバー バージョン2。このバージョンのエンキューレプリケーションサーバーは、SAPカーネルバージョン7.51以降で使用できます。 |
| **HA** | 高可用性。 |
| **ID** または **<ID>** | SAP インスタンスの数値識別子(2桁）。 |
| **< *INST* >** | SAP インスタンス用のディレクトリ。その名前はインスタンスに含まれるサービスとインスタンス番号に由来する。例えば、CI <INST> は、 *DVEBMGS00* など。 |
| **PAS** | プライマリアプリケーションサーバインスタンス。 |
| **SAP インスタンス** | 同時に開始および停止されるプロセスのグループ。 |
| **SAP シス** | SAP インスタンスのグループ。 |

| | |
|---|---|
| テム | |
| **<*sapmnt*>** | SAP のホームディレクトリ。デフォルトでは */sapmnt* だが、インストール時にユーザにより変更可能。 |
| **SCS** | SAP セントラルサービスインスタンス。NetWeaver Java 環境向けのメッセージサービスと ENQEUE サービスで構成される SAP インスタンスである。このインスタンスは、単一障害点であり、LifeKeeper で保護する必要がある。 |
| **(SCS)** | バックアップ SAP セントラルサービスインスタンスサーバ。プライマリ SCS サーバに障害が発生した場合の SCS ホストサーバである。 |
| **SID** または **<SID>** | システム ID。 |
| **sid** または **<sid>** | SID の小文字バージョン。 |
| **SPOF** | 単一障害点(Sinlge point of Failure) |

# 6.17.2. LifeKeeper – SAP アイコン

以下のアイコンは、LifeKeeper 環境における SAP リソースのステータスを解釈する上で重要です。これらのアイコンは、LifeKeeper UI 上に表示されます。

| | |
|---|---|
| | アクティブ - リソースはアクティブで In Service である (通常状態)。 |
| | スタンバイ - リソースはバックアップノードにあり、プライマリリソースに障害が発生したら引き継ぐ準備ができている (通常状態)。 |
| | 障害 - リソースに障害が発生した。リソースを In Service にする (リソース上で右クリックし、[In Service] を選択して Enter を押す) ことができる。リソースにまだ障害が残っている場合、リカバリは失敗する。 (障害状態)。 |
| | **要注意** - SAP リソースに障害が発生したか、警告状態である。障害が発生しており、自動リカバリが有効な場合 (保護レベルが「Full」または「Standard」)、LifeKeeper は自動的にリソースの復旧を試みる。<br>SAP リソースを右クリックして、 **[Properties]** を選択すると警告状態にあるリソースが表示される。 **黄色** の SAP 状態は通常だが、SAP リソースの実行が遅いか、パフォーマンスボトルネックがあることを示す。 |
| | **保護レベルの更新** - SAP リソースの保護レベルを変更するためのアクションアイコン。 |
| | **リカバリレベルの更新** - SAP リソースのリカバリレベルを変更するためのアクションアイコン。 |
| | **警告の処理** - 警告に障害を設定するためのアクションアイコン。 |
| | **開始** 、 **停止** 、 **移行** 、 **メンテナンスモード** の設定など、さまざまなアクションを選択するためのアクションアイコン。 |

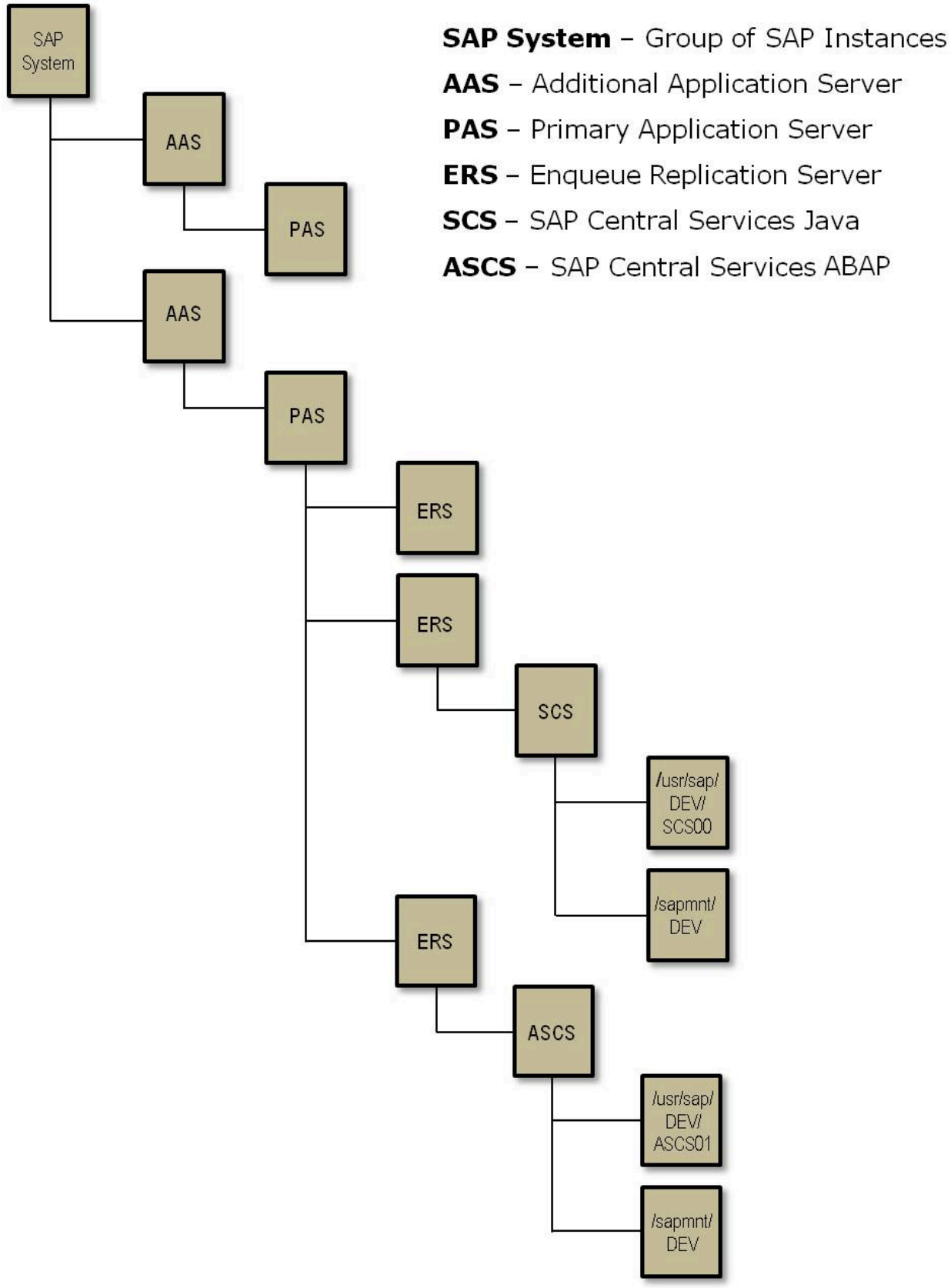
# 6.17.3. SAP Recovery Kit の概要

SAP NetWeaver フレームワークには、冗長化することのできないサービスが一部存在します。これらのサービスは、同一の SAP システム上に複数個存在することができないため、単一障害点になります。LifeKeeper SAP Recovery Kit は、これらの単一障害点に標準的な LifeKeeper 機能による保護を提供します。さらに、このキットでは、SAP インフラストラクチャのその他の部分についても、さまざまなレベルにおいて保護する機能を提供します。各インフラストラクチャコンポーネントの保護は、階層内の単一のリソースで表現されます。

SAP Recovery Kit は、さまざまな SAP インスタンス、すなわち SAP プライマリアプリケーションサーバ (PAS) インスタンス、ABAP SAP セントラルサービス (ASCS) インスタンス、SAP セントラルサービス (SCS) インスタンスに対して監視とスイッチオーバを提供します (セントラルサービスインスタンスは、ENQUEUE およびメッセージサーバを保護する)。SAP Recovery Kit は、適切なデータベースリカバリキットと連携してデータベースを保護し、Network File System (NFS) Server Recovery Kit と連携して NFS マウントを保護します。また、必要に応じてクラスタ内のネットワークカード間で移動できる仮想 IP アドレスを提供するため、IP Recovery Kit も使用されます。Network Attached Storage (NAS) Recovery Kit を使用して、ローカル NFS マウントを保護できます。これらさまざまな Recovery Kit を使用して、アプリケーション環境のすべてのコンポーネントを保護する SAP リソース階層を構築します。

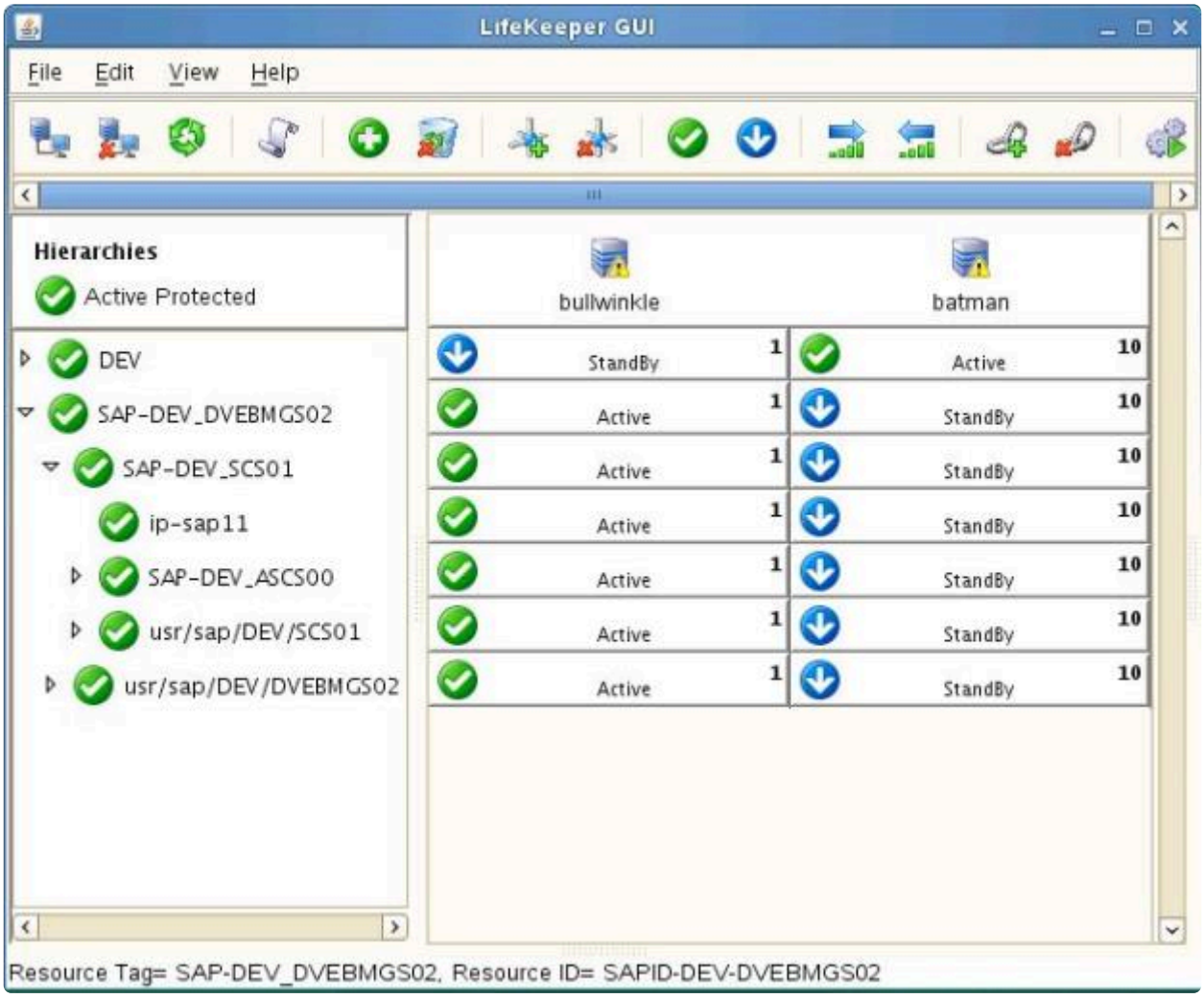
各 Recovery Kit は、保護下にあるアプリケーションの状態を監視し、ローカルおよび別のクラスタサーバ上のアプリケーションを停止し再開できます。

**SAP システム階層のマップ**




典型的な SAP リソース階層は、LifeKeeper GUI に次のように表示されます。

## SAP リソース階層




注記: ディレクトリ */usr/sap/trans* は、SAP 環境では必須ではありません。SAP NetWeaver Java のみの環境では、このディレクトリは存在しません。

注記: SPS-L v9.4.0以降で作成されたERSリソースは、独立した独自の階層で動作します。詳細については、LifeKeeperのERSリソースタイプ を参照してください。

# 6.17.4. LifeKeeper SAP ソリューションページ

## SAP High Availability Interface 7.73

注意: SAPの構築において使用するオペレーティングシステムおよび構成についてはSAP, SIOS およびインフラの提供者 (パブリック/プライベート クラウドまたはオンプレミス)がサポートしている必要があります。SAP NetWeaver および S/4HANA の各バージョンにおけるサポート対象のオペレーティングシステムについては、SAP のProduct Availability Matrix (PAM) を確認してください。

| パッケージ | バージョン | OS / アプリケーションバージョンのサポート |
|---|---|---|
| LifeKeeper Core (SAP) | 9.7.0 | Red Hat Enterprise Linux 7 (7.0 から 7.9 まで)<br>Red Hat Enterprise Linux 8 (8.0 から 8.4 まで)<br>SUSE LINUX Enterprise Server (SLES) 12 (SP1 から SP5 まで)<br>SUSE LINUX Enterprise Server (SLES) 15 (15.0 から SP3 まで)<br>Oracle Linux 7 (7.0 から 7.6 まで、UEK R4 カーネルを含む) |
| LifeKeeper SAP Recovery Kit | 9.7.0 | SAP NetWeaver 7.3、Enhancement Package 1 を含む<br>SAP NetWeaver 7.4<br>SAP NetWeaver 7.5<br>SAP NetWeaver AS for ABAP 7.51 innovation package<br>SAP NetWeaver AS for ABAP 7.52 innovation package<br>SAP S/4HANA 1809 Platform<br>SAP S/4HANA 1909 Platform<br>SAP S/4HANA 2020 Platform<br>SAP S/4HANA 2021 Platform<br>SAP S/4HANA 2022 Platform |
| LifeKeeper NFS Server Recovery Kit | 9.7.0 | カーネルバージョンが 2.6 以降の Linux ディストリビューションのエクスポートされた NFS ファイルシステム |
| LifeKeeper Supported Databases for SAP | 9.7.0 | Oracle, SAP ASE (Sybase), SAP HANA, SAP DB (MaxDB), IBM DB2 |
| SAP HA Interface Connector | 7.73 | SAP High Availability Interface 7.73, 7.77, 7.81, 7.85, 7.89 |

v9.5.0より、SIOS は新しい SAP HANA Application Recovery Kit をリリースしました。SIOS は **2022年3月31日** まで、9.4.x リリースでの SAP HANA gen/app ベースの Recovery Kit のサポートを継続します。LifeKeeper for Linux v9.5 以降をご利用の場合は、新しい（ビルトインの）SAP HANA Application Recovery Kit を使用する必要があります。

! 既存の SAP HANA gen/appベースのRecovery Kit は v9.5.0 以降では サポートされていません 。LifeKeeper for Linux v9.5.0 以降へのアップグレードを希望するユーザーは、既存の SAP HANA gen/app ベースのRecovery Kitから新しい SAP HANA Recovery Kit に 移行する必要があります 。

✻ 注意： 拡張 SAP サポート用に構築されたオペレーティングシステムのバージョン（Red Hat Enterprise Linux for SAP Business Applications、Red Hat Enterprise Linux for SAP Solutions、SUSE Linux Enterprise Server for SAP Applicationsなど）も、実行中の Linux カーネルのバージョンが上記のサポート対象 OS のバージョンと同じであればサポートされます。

# 6.17.5. SAP ハードウェア / ソフトウェア要件

SPS SAP Recovery Kit をインストールして設定する前に、使用している設定が以下の要件を満たしていることを確認してください。

- **サーバ** 。この Recovery Kit には、SPS for Linux テクニカルドキュメンテーション および SPS for Linux リリースノート で説明されている要件に従って設定した 2 台以上のコンピュータが必要です。

- **共有ストレージ** 。SAP プライマリアプリケーションサーバ (PAS) インスタンス、ABAP SAP セントラルサービス (ASCS) インスタンス、SAP セントラルサービス (SCS) インスタンス、およびプログラムファイルは、SPS 環境の共有ディスク上に存在する必要があります。ERSv2 構成の場合、ERS インスタンスのファイルシステムは共有ディスクに存在する場合もあります。

- **IP ネットワークインターフェース** 。各サーバは、イーサーネット TCP/IP をサポートするネットワーク・インターフェースを 1 つ以上必要とします。IP スイッチオーバが正しく動作するには、ローカルネットワークに接続されているユーザシステムが標準の TCP/IP 仕様に準拠している必要があります。

  **注記:** 各サーバが必要とするネットワークインターフェースが 1 つだけであっても、異種メディア要件、スループット要件、単一障害点の排除、ネットワークのセグメンテーションといった多くの理由のために、複数のインターフェースの使用を推奨します (詳細については、IP ローカルリカバリと設定に関する考慮事項 を参照してください)。

- **オペレーティングシステム** 。Linux オペレーティングシステム。(サポート対象のディストリビューションおよびカーネルのバージョンについては、SPS for Linux リリースノート を参照してください。)

- **SELinux** 。無効または permissive モード。 (詳細については、テクニカルノート を参照してください。)

- **TCP/IP ソフトウェア** 。各サーバは TCP/IP ソフトウェアを必要とします。

- **SAP ソフトウェア** 。各サーバは、SPS と SPS SAP Recovery Kit の設定に先だって SAP ソフトウェアをインストールして設定しておく必要があります。各サーバには同じバージョンをインストールする必要があります。最新のリリースの互換性および購入方法については、SPS for Linux リリースノート を参照するか、営業担当者にお問い合わせください。

- **SPS ソフトウェア** 。各サーバには同じバージョンの SPS ソフトウェアおよびパッチをインストールする必要があります。具体的な LifeKeeper の要件については、 SPS for Linux リリースノート を参照してください。

- **SPS IP Recovery Kit** 。この Recovery Kit は、リモートクライアントが SAP PAS、ASCS、または SCS インスタンスにアクセスする場合に必要です。各サーバで同じバージョンの Recovery Kit を使用する必要があります。

- **SPS for Linux NFS Server Recovery Kit** 。この Recovery Kit は、ほとんどの設定で必要です。各

サーバで同じバージョンの Recovery Kit を使用する必要があります。

- **SPS for Linux Network Attached Storage (NAS) Recovery Kit** 。この Recovery Kit は、一部の設定で必要です。各サーバで同じバージョンの Recovery Kit を使用する必要があります。

- **SPS for Linux Database Recovery Kit** 。SAP で利用されているデータベース用の SPS リカバリキットを各データベースサーバにインストールする必要があります。サポート対象のデータベースの詳細については、 SPS for Linux リリースノート を参照してください。SAP を設定する前に、SAP PAS、ASCS、または SCS インスタンスに対する LifeKeeper データベース階層を作成する必要があります。

**重要事項:**

- バージョン 7.3 より前の SAP を使用する場合、 **SAPHOSTAGENT** のダウンロードおよびインストール方法について、SAP ドキュメンテーションおよび注記を参照してください (「設定の計画」トピックの重要事項 を参照)。

- Core ソフトウェアおよび SAP Recovery Kit をインストールまたは削除する手順については、SPS for Linux インストールガイド を参照してください。

- インストール手順は、推奨する順序に従って行う必要があります。LifeKeeper を先にインストールした場合、SAP を正しくインストールできません。

- 必要な SPS Recovery Kit のそれぞれの設定方法については、各キットのドキュメンテーション (IP、NFS Server、NAS、およびデータベース Recovery Kit) を参照してください。

- スワップ領域やメモリ要件などの他のインストール要件については、SAP インストールドキュメンテーションを参照してください。

# 6.17.6. SAP 設定上の考慮事項

**重要:** 業界のベストプラクティスに従い、SIOS では、SAP システムのセットアップ (「ループバック構成」とも呼ばれます) 時に、NFS サーバー/クライアントを SAP サービスと同じノードにインストールすることを推奨しなくなりました。 代わりに、次のような代替構成を使用することをお勧めします。
- EFS in AWS
- Azure Premium File Shares with NFS
- Google NFS Share
- NFS as separate cluster

今後、「ループバック構成」を使用した新規インストールはサポートされません。 問題が SIOS ソフトウェア内にある場合、既存のお客様は可能な範囲でサポートいたしますが、代替構成への移行が推奨されます。 次のリリースでは、既存の構成はサポートされません。

このセクションでは、SAP の設定に着手する前に考慮すべき情報と、SAP の典型的な設定例を示します。また、SAP を LifeKeeper で設定および保護するための操作手順を示します。

SAP を LifeKeeper でサポートされているカーネル 2.4 または 2.6 を使用する Linux ディストリビューションにインストールする方法については、SAP が提供するデータベース固有の SAP インストールガイドを参照してください。

また、 SPS for Linux テクニカルドキュメンテーション を参照して、 SPS Core リソース (ファイルシステムリソースなど) を設定してください。

## サポートされる構成

SAP Highly Available (HA) 環境において取ることのできるデータベースサーバおよびアプリケーションサーバ構成は多数あります。LifeKeeper で保護するために SAP を設定する上で必要な具体的手順は、構成によって異なります。そのため、実際の要件に最も近い構成を見つけることが重要です。サポートされる構成例の一部を以下に示します。

ABAP+Java 構成 (ASCS および SCS)

ABAP のみの構成 (ASCS)

Java のみの構成 (SCS)

上記の例に図示する構成は 2 つのサーバで構成されており、各サーバは、セントラルサービスインスタンスとともに、ERS インスタンス、データベースインスタンス、プライマリアプリケーションサーバインスタンス、0 個以上の追加の冗長性のあるアプリケーションサーバインスタンス (AS) をホストしています。冗長性のあるアプリケーションサーバなしで SAP を設定することは可能ですが、その場合、ASCS インスタンスまたは SCS インスタンスへのユーザによるログインが必要になります。これは、SAP によって推奨されていません。ASCS インスタンス、SCS インスタンス、およびデータベースサーバは、データベースファイルとアプリケーションファイルのために共有ファイルストレージにアクセスできます。

セントラルサービスは多くのリソースを使用するわけではなく、非常に高速でスイッチオーバ可能です

が、データベースはスイッチオーバ速度に大きな影響を与えます。そのため、データベースインスタンスとセントラルサービスインスタンス (ASCS および SCS) は、2 つの異なる LifeKeeper 階層によって保護することを推奨します。これらは、異なるサーバ上で実行することも、同一サーバ上で実行することも可能です。

## 設定に関する注記

以下は、HAクラスタ環境で実行するための SAP の設定に関する技術的な注記です。SAP を LifeKeeper で保護するための操作手順については、関連するトピックを参照してください。

SAP ディレクトリ構造

SAP 仮想サーバ名

SAP ヘルスモニタリング

SAP ライセンス

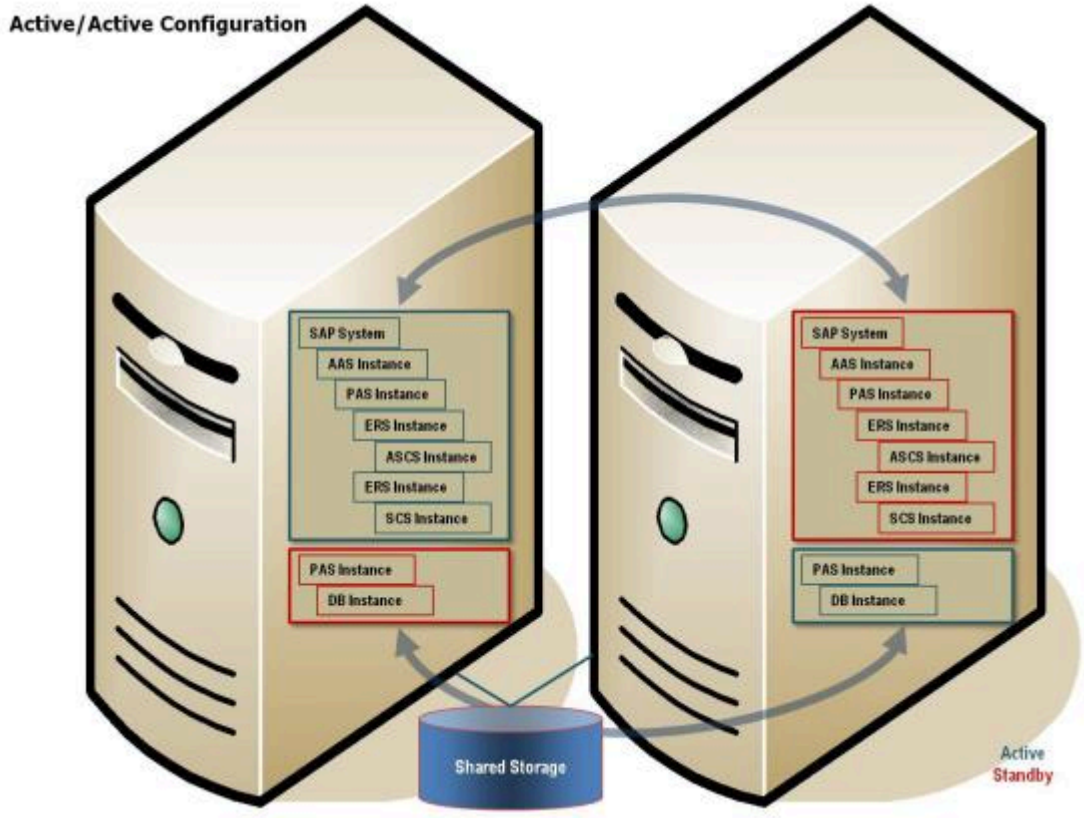
その他の注記

# 6.17.6.1. ABAP+Java の設定 (ASCS および SCS)

ABAP+Java の設定は、次のインストールで構成されます。

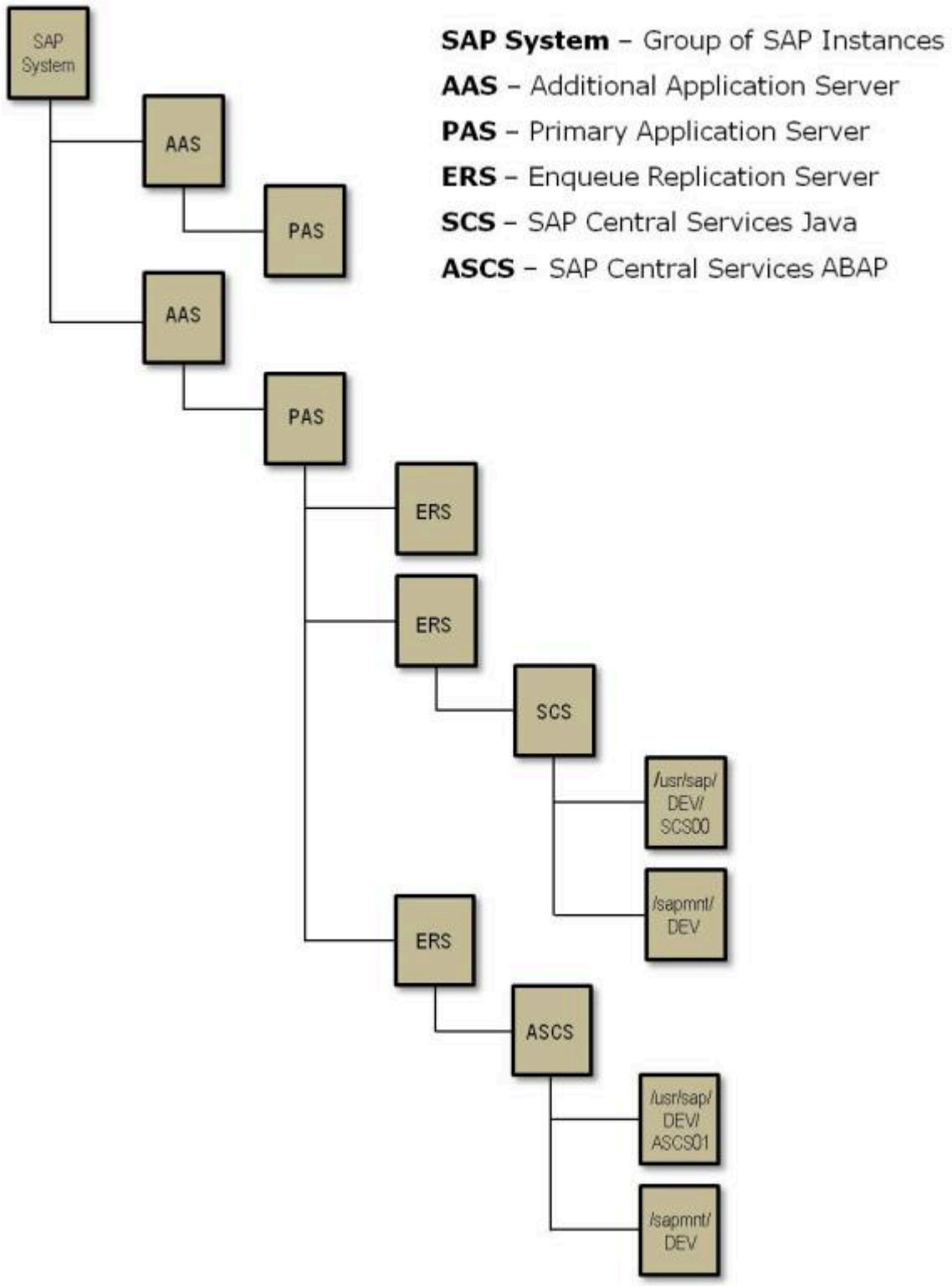
- ABAP セントラルサービスインスタンス (ASCS インスタンス)
- ASCS インスタンス用エンキューレプリケーションサーバインスタンス (ERS インスタンス) (オプション) (ASCS インスタンスと SCS インスタンスの両方は、自身の ERS インスタンスを持つ必要がある)
- Java 用セントラルサービスインスタンス (SCS インスタンス)
- SCS インスタンス用エンキューレプリケーションサーバインスタンス (ERS インスタンス) (オプション)
- データベースインスタンス (DB インスタンス) - ABAP スタックと Java スタックは、同一のデータベース内で独自のデータベーススキーマを使用します。
- プライマリアプリケーションサーバインスタンス (PAS)
- 追加のアプリケーションサーバインスタンス (AAS) - 追加のアプリケーションサーバインスタンス (AAS) を、プライマリアプリケーションサーバインスタンス (PAS) ホストとは別のホストにインストールすることを推奨します。

# SAP デュアルスタック (ABAP+Java) システム向けスイッチオーバクラスタ




上の例において、ASCS と SCS は、データベースとは別のリソース階層にあり、これらのセントラルサービスインスタンスは、データベースとは異なるサーバで起動しています。

# SAP 階層の例



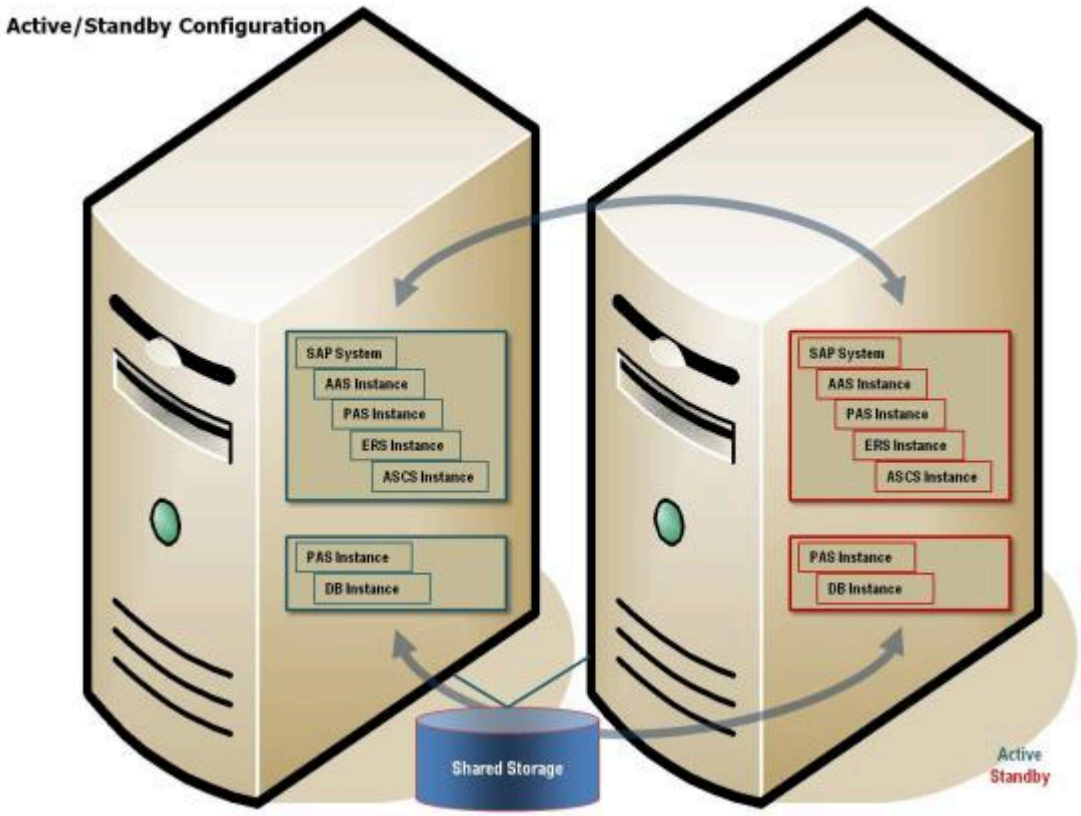

注記: SPS-L v9.4.0 以降で作成された ERS リソースは、独立した独自の階層で動作します。詳細については、LifeKeeper の ERS リソースタイプ を参照してください。

# 6.17.6.2. ABAP SCS (ASCS)

ABAP のみの設定は、次のインストールで構成されます。

- ABAP セントラルサービスインスタンス (ASCS インスタンス)
- ASCS インスタンス用エンキューレプリケーションサーバインスタンス (ERS インスタンス) (オプション)
- データベースインスタンス (DB インスタンス)
- プライマリアプリケーションサーバインスタンス (PAS)
- 追加のアプリケーションサーバインスタンス (AAS) - 追加のアプリケーションサーバインスタンス (AAS) を、プライマリアプリケーションサーバインスタンス (PAS) ホストとは別のホストにインストールすることを推奨します。

## SAP ABAP のみの (ASCS) システム向けスイッチオーバクラスタ




上の例において、ASCS は、データベースとは別のリソース階層にあります。同一サーバ上でデータベースとしてアクティブでも、別々にフェイルオーバできます。

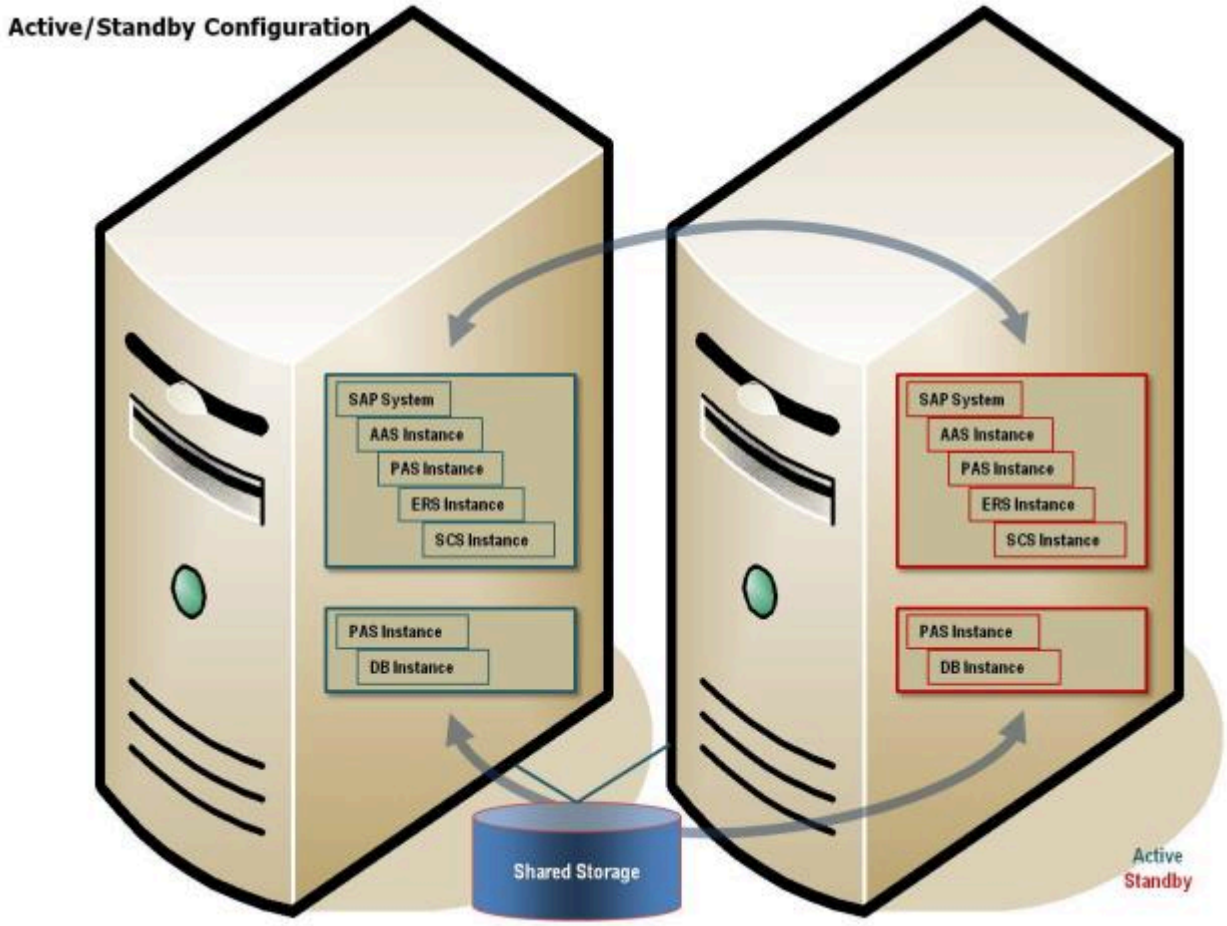
注記: SPS-L v9.4.0以降で作成されたERSリソースは、独立した独自の階層で動作します。詳細については、LifeKeeperのERSリソースタイプ を参照してください。

# 6.17.6.3. Java のみの構成 (SCS)

Java のみの構成は、次のインストールで構成されます。

- Java 用セントラルサービスインスタンス (SCS インスタンス)
- SCS インスタンス用エンキューレプリケーションサーバインスタンス (ERS インスタンス) (オプション)
- データベースインスタンス (DB インスタンス)
- プライマリアプリケーションサーバインスタンス (PAS)
- 追加のアプリケーションサーバインスタンス (AAS) - 追加のアプリケーションサーバインスタンス (AAS) を、プライマリアプリケーションサーバインスタンス (PAS) ホストとは別のホストにインストールすることを推奨します。

## Java のみのシステム (SCS) 向けスイッチオーバクラスタ




上の例において、SCS は、データベースとは別のリソース階層にあります。同一サーバ上でデータベースとしてアクティブでも、別々にフェイルオーバできます。

注記: SPS-L v9.4.0以降で作成された ERS リソースは、独立した独自の階層で動作します。詳細については、 LifeKeeper の ERS リソースタイプ を参照してください。

# 6.17.6.4. SAP ディレクトリ構造

データベースのディレクトリ構造は、SAP システムと合わせて使用する各データベース管理システムによって異なります。使用するデータベースのディレクトリ構造の詳細については、データベース管理システム固有の SAP Installation Guide を参照してください。データベース向け LifeKeeper Recovery Kit で保護するには、すべてのデータベースファイルを共有ディスク上に配置する必要があります。データベースの保護に関する詳細については、データベースごとの Recovery Kit ドキュメンテーション を参照してください。

このセクションで説明する SAP ディレクトリの図解については、下のディレクトリ構造図 を参照してください。

インストール中に作成されるディレクトリの種類は、次の通りです。

**物理的に共有されたディレクトリ** (グローバルホスト上にあり、NFS で共有される)

> */<sapmnt>/<SAPSID>* - 1 つの SAP システム用のソフトウェアとデータ (同一の SAP システムに属するすべてのホストにマウントする必要がある)
>
> */usr/sap/trans* - グローバルトランスポートディレクトリ (エクスポートポイントを持つ必要がある)

以下のローカルディレクトリを持つ */usr/sap* などのノードにバインディングされている **論理的に共有されたディレクトリ** (ローカルホスト上にあり、グローバルホストへのシンボリックリンクを持つ)

> */usr/sap/<SAPSID>*
>
> */usr/sap/<SAPSID>/SYS*
>
> */usr/sap/hostctrl*

以下のような SAP インスタンスを含む **ローカルディレクトリ** (ローカルホスト上にあり、共有されている)

> */usr/sap/<**SAPSID**>/DVEBMGS<**No.**>* — プライマリアプリケーションサーバインスタンスのディレクトリ
>
> */usr/sap/<**SAPSID**>/D<**No.**>* — プライマリアプリケーションサーバインスタンスのディレクトリ
>
> */usr/sap/<**SAPSID**>/ASCS<**No.**>* — ABAP セントラルサービスインスタンス (ASCS) のディレクトリ
>
> */usr/sap/<**SAPSID**>/SCS<**No.**>* — Java セントラルサービスインスタンス (SCS) のディレクトリ
>
> */usr/sap/<**SAPSID**>/ERS<**No.**>* — ASCS および SCS 向けエンキューレプリケーションサーバインスタンス (ERS) のディレクトリ

SAP ディレクトリ */sapmnt/<**SAPSID**>* および */usr/sap/trans* は、NFS からマウントされます。ただし、SAP インスタンスディレクトリ (*/usr/sap/<**SAPSID**>/<**INSTTYPE**><**No.**>*) は、常に現在インスタンスを実行しているクラスターノード上にマウントする必要があります。Quorum または Witness サーバーを使用する場合のみ、これらのディレクトリを NFS でマウントします。必要なディレクトリ構造は、選択した構成によって異なります。必要なディレクトリ構造を決める問題はいくつかあります。

## NFS マウントポイントと inode

LifeKeeper は、inode を使用して NFS 共有情報を管理するため、各 NFS 共有は固有の inode を持つ必要があります。すべてのファイルシステムのルートディレクトリは同じ inode を持つため、NFS 共有を LifeKeeper で保護するためには、ルートから少なくとも 1つ下のディレクトリ階層にある必要があります。例えば、上記の情報に留意して、*/usr/sap/trans* ディレクトリを SAP サーバ上で NFS 共有する場合、共有ストレージデバイス上に */trans* ディレクトリが作成されます。これには、共有ストレージデバイスを */usr/sap* としてマウントする必要があります。しかしながら、この配置で要求されるように、すべてのファイルを共有ストレージ上の */usr/sap* 配下に置くことは必ずしも望ましくありません。この問題を回避するため、NFS 共有されているディレクトリを含むすべての共有ファイルシステムをマウントするための */exports* ディレクトリツリーを作成し、その後で SAP ディレクトリと */exports* ディレクトリ間にソフトリンクを作成するか、この NFS 共有ディレクトリをローカルに NFS マウントすることを推奨します。(注記: ここで */exports* と呼ぶディレクトリの名前は、ユーザの好みによって変更できますが、説明を分かりやすくするため、このドキュメンテーションではこのディレクトリを */exports* と呼びます。)例えば、この SAP プライマリサーバの例に関する次のディレクトリとリンク / マウントは、以下のようになります。

| ***/usr/sap/trans* 共有の場合** | |
|---|---|
| ディレクトリ | **備考** |
| */trans* | 共有ファイルシステム上に作成され、NFS で共有 |
| */exports/usr/sap* | / (共有ファイルシステム上) にマウント |
| */usr/sap/trans* | /exports/usr/sap/trans にソフトリンク |

同様に、 *<sapmnt>/<SAPSID>* 共有に対するディレクトリとリンクは、次のようになります。

| ***<sapmnt>/<SAPSID>* 共有の場合** | |
|---|---|
| ディレクトリ | **備考** |
| */<SAPSID>* | 共有ファイルシステム上に作成され、NFS で共有 |
| */exports/sapmnt* | / (共有ファイルシステム上) にマウント |
| *<sapmnt>/<SAPSID>* | *<virtual SAP server>:/exports/sapmnt/<SAPSID>* に *NFS* マウント |

すべてのディレクトリ構造とリンクを作成するための詳細な手順を、本ドキュメンテーションの設定手順で後述します。inode 競合の詳細と、NFSv4 の新機能を使用するための情報については、NFS Server Recovery Kit ドキュメンテーション を参照してください。

# ローカル NFS マウント

LifeKeeper 環境の SAP に推奨されるディレクトリ構造では、1 つまたは複数の SAP システムディレクトリについてローカルにマウントされた NFS 共有が要求されます。いずれかのローカルにマウントされた NFS 共有の NFS エクスポートポイントが使用不可になると、エクスポートポイントが使用可能になるのを待機する間、システムがハングすることがあります。システムの再起動を含め、多くのシステム操作が正しく動作しません。SAP クラスタの NFS サーバを LifeKeeper で保護する必要があり、ローカルマウントポイントが存在する間は手動でサービス休止状態にしてはならないことに注意してください。

不注意で NFS サーバを停止してクラスタをハングさせないように、NFS の考慮事項 トピックに記載した推奨事項に従ってください。

NFS 共有にアクセスできない場合、アンマウントが失敗することがあります。LifeKeeper は、ファイルシステムのアンマウントを複数回試行します。複数回試行することにより、通常、最終的にリソースを Out of service にすることに成功します。ただし、リソースを Out of service にする際に遅延が発生します。再試行を回避するには、マウントオプション「nfsvers=3, proto=udp 」を使用します。

> ! フェイルオーバとリカバリで重要になるので、udp の使用方法に注意してください。

# NFS マウントと su

LifeKeeper は、多くのデータベース作業および SAP 作業を、`su - <admin name> -c <command>` コマンド構文を使用してデータベース処理および SAP 処理を実行することで実現します。このような方法で呼び出された場合、su コマンドにより、管理者のホームディレクトリにあるログインスクリプトが実行されます。これらのログインスクリプトは、環境変数をさまざまな SAP パスに設定します。その一部は、NFS マウントされた共有に存在する可能性があります。これらの NFS 共有が何らかの理由で利用できない場合、su 呼び出しは、NFS 共有が利用可能になるまで待機してハングします。

ハングスクリプトは LifeKeeper の正常実行を妨げる可能性があるため、この潜在的な問題に対処するよう、サーバを設定することが望まれます。SAP リソースの削除、リストア、監視処理を行う LifeKeeper スクリプトは、これらのスクリプトがいつまでもハングし続けることを防止するタイマを内蔵しています。そのため、SAP Application Recovery Kit では、NFS ハングに対処するための設定作業は必要ありません。

それでも、利用不可能な NFS 共有によって影響を受ける手動の処理は数多く存在することに注意してください。手動で LifeKeeper 処理を実行する前に、必ず、すべての NFS 共有が利用可能であることを確認する必要があります。

> * アクセスできない NFS 共有の処理遅延を回避するには、NFS の考慮事項 に記載されている推奨事項に従ってください。

# < *INST* > ディレクトリの場所

*/usr/sap/<SAPSID>* パスは NFS 共有ではないため、ファイルシステムのルートディレクトリにマウントできます。 */usr/sap/<SAPSID>* パスには、サーバ上で実行可能な SAP インスタンスごとに、 *SYS* サブディレクトリと < *INST* > サブディレクトリが含まれています。構成によっては、< *INST* > ディレクトリは 1 つしかないため、共有ファイルシステム上の */usr/sap/<SAPSID>* 配下に置くことができます。ただし、他の構成では、バックアップサーバもローカル AS インスタンスを含む可能性があり、その < *INST* > ディレクトリは常に利用できるわけではないため、共有ファイルシステム上に置くことはできません。この問題を解決するため、特定の構成では、PAS、ASCS、または SCS の */usr/sap/<SAPSID>/<INST>* 、*/usr/sap/<SAPSID>/<ASCS-INST>* 、または */usr/sap/<SAPSID>/<SCS-INST>* ディレクトリを */usr/sap/<SAPSID>* ではなく共有ファイルシステムにマウントし、AS の */usr/sap/<SAPSID>/SYS* および */usr/sap/<SAPSID>/<AS-INST>* をローカルサーバ上に置いてください。

> ERS ファイルシステムは、ERSv1 の場合はローカルにマウントし、ERSv2 の場合は共有ストレージにマウントする必要があります。

例えば、ABAP+Java 構成では、次のディレクトリとマウントポイントを作成する必要があります。

| ディレクトリ | 注意 |
|---|---|
| */usr/sap/<SAPSID>/DVEBMGS<**Instance #**>* | /(共有ファイルシステム上) にマウント |
| */usr/sap/<SAPSID>/SCS<**Instance #**>* | /(共有ファイルシステム上) にマウント |
| */usr/sap/<SAPSID>/ERS<**Instance #**> (SCS インスタンス用)* | すべてのクラスターノード上にローカルでマウントするか、NAS 共有からマウントする必要がある ( 共有ファイルシステム上にマウントしてはならない ) |
| */usr/sap/<SAPSID>/ASCS<**Instance #**>* | /(共有ファイルシステム上) にマウント |
| */usr/sap/<SAPSID>/ERS<**Instance #**>*<br><br>*(ASCS インスタンス用)* | すべてのクラスターノード上にローカルでマウントするか、NAS 共有からマウントする必要がある ( 共有ファイルシステム上にマウントしてはならない ) |
| */usr/sap/<SAPSID>/AS<**Instance #**>* | バックアップサーバ上の AS に作成 |

**注記:** エンキューレプリケーションサーバ (ERS) リソースは、クラスタのプライマリノードで In Service (ISP) になります。ただし、ERS のアーキテクチャと機能は、インスタンスの実際のプロセスがバックアップノード上で実行される必要があります。これによって、スタンバイサーバはプライマリサーバおよびプライマリエンキューサーバインスタンスのロックテーブル情報の完全なコピーを保持することができます。エンキューサーバを実行するプライマリサーバに障害が発生した場合、ERS プロセスが現在実行されているバックアップサーバ上の SIOS Protection Suite によって再起動されます。ERS 上に保存されてい

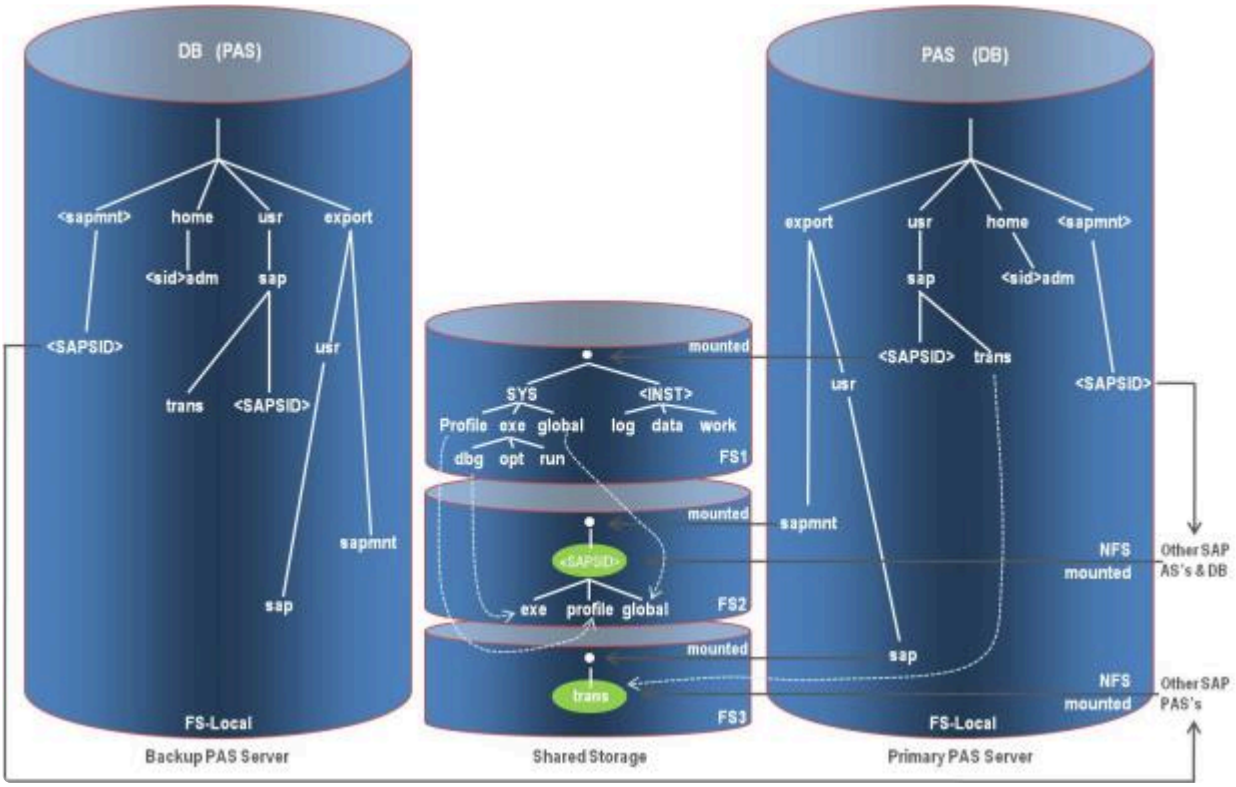
るロックテーブル (レプリケーションテーブル) が復旧中のエンキューサーバプロセスに転送され、それを基に新しいロックテーブルが作成されます。このプロセスが完了すると、アクティブなレプリケーションサーバは非アクティブになります (このサーバはエンキューサーバとの接続を閉じ、レプリケーションテーブルを削除します)。これまで非アクティブだった新しい現在のバックアップノード (元のプライマリ) 上で、SIOS Protection Suite によって ERS プロセスが再起動されます。ERS プロセスがアクティブになると、エンキューサーバに接続され、レプリケーションテーブルが作成されます。ERS プロセスと SAP アーキテクチャの機能の詳細については、http://help.sap.com にアクセスして、 **エンキューレプリケーションサービス** で検索してください。

レプリケーションサーバはバックアップノードで常にアクティブであるため、ファイルシステムはプライマリノードでアクティブになり、レプリケーションサーバプロセスはバックアップノードでアクティブになります。このため、レプリケーションサーバはSIOS Protection Suiteで保護されたファイルシステムには常駐できません。したがって、ERSが使用するファイルシステムは、すべてのクラスタノードでローカルにマウントするか、NAS共有からマウントする必要があります。

# ディレクトリ構造図

ABAP のみの環境を LifeKeeper で保護するために必要なディレクトリ構造を、下図に示します。図内で使用されている略語の説明については、略語と定義 セクションを参照してください。

## ディレクトリ構造の例



凡例

| | |
|---|---|
| | soft link |
| | mount point |
| | NFS shared file system |
| PAS | Primary PAS |
| (PAS) | Backup PAS |
| AS | Application Server |
| DB | Primary DB Server |
| (DB) | Backup DB Server |
| FSn | File System on shared storage |
| FS-local | File System on local disk |

# ディレクトリ構造のオプション

このドキュメンテーションに示す設定手順は、上に示したディレクトリ構造および図に基づきます。これは、SIOS Technology Corp によってテストおよび認証済みの、推奨ディレクトリ構造です。

すべてがテストされているわけではありませんが、SAP Recovery Kit が正しく動作する、その他のディレクトリ構造のバリエーションも存在します。ディレクトリ構造のバリエーションの設定については、以下の指針に従う必要があります。

- */usr/sap/trans* ディレクトリは、ネットワーク上でアクセス可能な任意のサーバ上にホストでき、PAS サーバである必要はない。 */usr/sap/trans* ディレクトリに PAS からリモートでアクセスする場合、このディレクトリへのアクセスがミッションクリティカルであるかどうか判断する必要がある。ミッションクリティカルであれば、LifeKeeper で保護する。これには、ディレクトリが共有または複製ファイルシステムにあり、NFS Server Recovery Kit によって保護されていることが必要である。NFS を使用せずに */usr/sap/trans* ディレクトリをすべての SAP インスタンスから利用可能にする方法があれば、それを使用してもよい。

- */usr/sap/trans* ディレクトリは、PAS サーバに配置されているかどうかに関わらず、NFS 共有されている必要はない。

- */usr/sap/trans* ディレクトリは、NFS 共有または LifeKeeper で保護されていない場合、共有ファイルシステム上に置く必要はない。

- 図示した NFS ファイルシステムをエクスポートするために使用されるディレクトリ構造とパス名は、単なる例である。パス */exports/usr/sap* は、 */exports/sap* または単に */sap* でよい。

- */usr/sap/<SAPSID>/<INST>* パスは、共有ファイルシステム上のいずれかのレベルにある必要がある（ERSv1 インスタンスの場合を除く）。このパスのどの部分がファイルシステムのマウントポイントであるかは関係ない。 */usr*、*/usr/sap*、*/usr/sap/<SAPSID>* 、 */usr/sap/<SAPSID>/<INST>* のいずれも可能である。

- */sapmnt/<SAPSID>* パスは、共有ファイルシステム上のいずれかのレベルにある必要がある。設定図では、このパスは NFS マウントとして図示されている。これは SAP の要件であるが、LifeKeeper の要件ではない。

# 6.17.6.5. SAP 仮想サーバ名

SAP アプリケーションサーバと SAP クライアントは、PAS インスタンスが実行中のサーバの名前を使用して、SAP プライマリアプリケーションサーバ (PAS) と通信します。同様に、SAP PAS は、データベース (DB) と、DB サーバの名前を使用して通信します。HAクラスタでは、PAS は、常にプライマリサーバまたはバックアップサーバのいずれかで実行されます。どちらのサーバでアクティブになっているかに関わらず、その他のサーバとクライアントが PAS にシームレスに接続できるようにするため、PAS とのすべての通信に仮想サーバ名を使用します。また、仮想サーバ名は、PAS が実行されているどちらのサーバでもアクティブにできる切り替え可能 IP アドレスにもマッピングされます。

切り替え可能 IP アドレスは、LifeKeeper によって、IP Recovery Kit を使用して作成および処理されます。仮想サーバ名は、仮想サーバ名 / 切り替え可能 IP アドレスのマッピングを DNS に追加するか、すべてのサーバおよびクライアントのホストファイルに追加することにより、手動で設定されます。この仕組みの詳細については、IP ローカルリカバリー トピックを参照してください。

さらに、仮想サーバ名を物理サーバ名の代わりに使用するよう、SAP 設定ファイルを変更する必要があります。これについては、インストール セクションで詳しく説明されており、その他にも LifeKeeper で SAP を設定するための手順が記載されています。

**注記:** SAP アプリケーションサーバ階層と NFS サーバ階層では、別々の切り替え可能 IP アドレスを使用することを推奨します。これにより、NFS クライアントで使用される IP アドレスと SAP クライアントで使用される IP との分離を維持できます。

# 6.17.6.6. SAP ヘルスモニタリング

LifeKeeper は、プライマリアプリケーションサーバ (PAS) インスタンスの状態を監視し、SAP が正しく機能していないと判断した場合、リカバリ処理を開始します。

そのステータスは、GUI の [Properties Panel] と CLI を通じて、 **グレー** (未知/非アクティブ/オフライン)、 **赤** (障害発生)、 **黄** (問題)、または **緑** (良好) として、ユーザに返されます。

インスタンスのステータスが **グレー** の場合、状態は未知であるか、情報が利用できません。

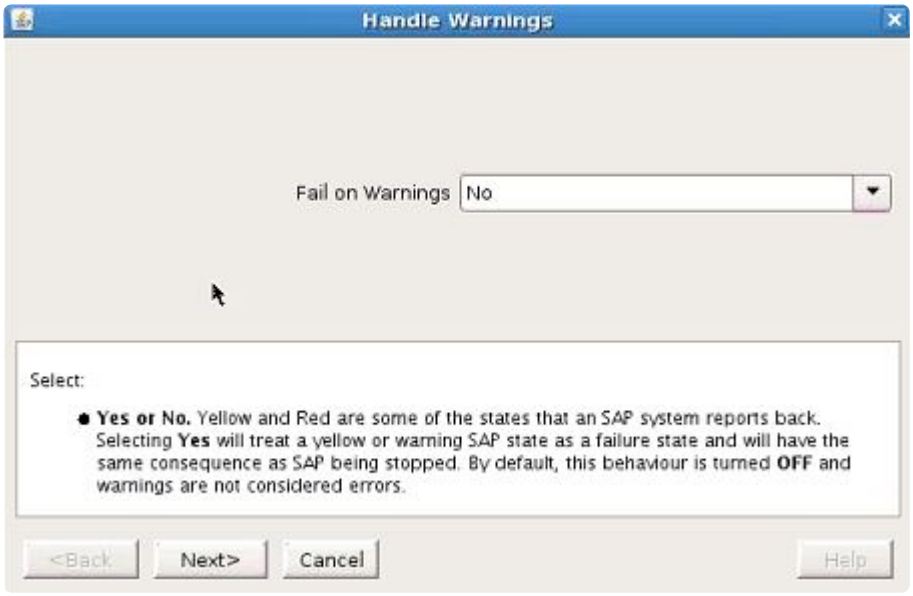
インスタンスのステータスが **赤** の場合、リソースは障害状態とみなされ、LifeKeeper は適切なリカバリ処理を開始します。

インスタンスのステータスが **黄** , の場合、定義されたインスタンスについて、SAP プロセスに問題がある可能性を示します。黄色のステータスのデフォルト動作は、リカバリを開始することなく監視を続行することです。

デフォルト動作は、このオプションを GUI リソースメニューで設定することで変更できます。

1. インスタンス を右クリックします。
2. **[Handle Warnings]** を選択します。

3. 以下の画面が表示され、 **[Fail on Warnings]** を選択できます。




**[Yes]** を選択すると、 **黄色の警告** はエラーとして扱われて、リカバリが開始されます。

**注記:** 黄は過渡的な状態であり、ほとんどの場合は障害を表すものではないため、この設定はデフォルト選択の **[No]** のままにすることを強くお勧めします。

# 6.17.6.7. SAP ライセンス

HAクラスタ環境では、SAP は、プライマリサーバとバックアップサーバの両方で実行できるよう設定されます。SAP ライセンススキームはハードウェア依存であるため、SAP が実行される各サーバについて個別のライセンスが必要になります。そのため、プライマリサーバとバックアップサーバの両方について、SAP ライセンスを取得しインストールする必要があります。

# 6.17.6.8. その他の注記

次の項目については、HAクラスタ環境において特別な設定手順が必要になります。以下の項目それぞれの設定要件に関する情報については、 *SAP Web Application Server in Switchover Environments* ドキュメントを参照してください。

- ログイングループ
- SAP スプーラ
- バッチジョブ
- SAP ルータ
- SAP システムアップグレード

# 6.17.7. SAP のインストール

## 設定 / インストール

LifeKeeper を使用して SAP リソース階層を作成する前に、以下の作業を下記に **推奨する順序で** 実施してください。以下に記載されていない追加の非 HA 固有の設定作業を行う必要があることに注意してください。詳細については、SAP インストールガイドを参照してください。

以下の作業では、「**SAP プライマリサーバ**」および「**SAP バックアップサーバ**」を参照しています。 **SAP プライマリサーバ** は、通常動作時にセントラルサービスが実行されるサーバです。 **SAP バックアップサーバ** は、SAP プライマリサーバに障害が発生したときにセントラルサービスが実行されるサーバです。

必ずしも必須ではありませんが、以下の操作手順には、使用する前にすべての共有ファイルシステムを LifeKeeper で保護する推奨手順が含まれています。共有ファイルシステムは、LifeKeeper で保護する前は両方のサーバからアクセス可能であるため、データが破壊される可能性があります。LifeKeeper でファイルシステムを保護することで、データへのアクセスを 1 つのサーバに留めます。

## SAP をインストールする前に

以下のトピックの作業は、SAP ソフトウェアをインストールする前に行う必要があります。記載されている順序に従って、作業を行ってください。また、NetWeaver 環境でのインストールを予定している場合、SAP ドキュメントの *SAP Web Application Server in Switchover Environments* を参照してください。

設定の計画

## SAP ソフトウェアのインストール

以下の作業は、SAP ソフトウェアを高可用性環境にインストールするために必要です。記載されている順序に従って、作業を行ってください。詳細については、各作業をクリックしてください。SAP のインストール手順の詳細については、関連する SAP インストールガイドを参照してください。

### プライマリサーバのインストール

コアサービス、ABAP セントラルサービス (ASCS)、および Java セントラルサービス (SCS) のインストール

データベースのインストール

プライマリアプリケーションサーバインスタンスのインストール

追加のアプリケーションサーバインスタンスのインストール

### バックアップサーバのインストール

バックアップサーバへのインストール

# LifeKeeper のインストール

LifeKeeper のインストール

ファイルシステムとディレクトリ構造の作成

# LifeKeeper での SAP の設定

## リソース設定作業

以下の作業では、LifeKeeper GUI の **[Edit]** ニューからオプションを選択することによって、Recovery Kit を設定する方法を説明します。それぞれの設定作業は、ツールバーから選択することもできます。または、状況表示ウィンドウの **リソース階層ツリー** (左側のペイン) のグローバルリソースを右クリックして、 **[Edit]** メニューと同じドロップダウンメニューの選択項目を表示することもできます。言うまでもなく、これは階層がすでに存在している場合にだけ可能な方法です。

または、状況表示ウィンドウのリソース階層表 (右側のペイン) のリソースインスタンスを右クリックして、サーバおよび特定リソースの状況に応じて、リソース階層の作成を除くすべての設定作業を実行することもできます。

IP リソース

SAP リソース階層の作成

リソース階層の削除

リソース階層の拡張

リソース階層の拡張解除

共通の Recovery Kit 作業

コマンドラインによる SAP の設定

SAPインスタンスに対してSAP SIOS HA Cluster Connectorを有効にする方法については、 SAP SIOS HA Cluster Connector (SSHCC)のアクティブ化 を参照してください。

LifeKeeperでERSインスタンスを適切に管理するには、ERSプロファイルはERSプロセスの開始に、Restart_Programではなく Start_Programパラメーターを使用する必要があります。ERSインスタンスプロファイルでこのパラメーターを変更する方法の詳細については、ASCS + ERS Restart_Programパラメーター を参照してください。

# SAP リソース階層のテスト

SAP ソフトウェアに対して LifeKeeper 保護を構築した後、その SAP 階層を徹底的にテストする必要があります。記載されている順序に従って、作業を行ってください。

テストの準備

テストの実行

# 6.17.7.1. SAP 構成を計画する

**重要:** 業界のベストプラクティスに従い、SIOS では、SAP システムのセットアップ (「ループバック構成」とも呼ばれます) 時に、NFS サーバー/クライアントを SAP サービスと同じノードにインストールすることを推奨しなくなりました。 代わりに、次のような代替構成を使用することをお勧めします。
- EFS in AWS
- Azure Premium File Shares with NFS
- Google NFS Share
- NFS as separate cluster

今後、「ループバック構成」を使用した新規インストールはサポートされません。 問題が SIOS ソフトウェア内にある場合、既存のお客様は可能な範囲でサポートいたしますが、代替構成への移行が推奨されます。 次のリリースでは、既存の構成はサポートされません。

1. 使用する設定 を決定してください。必要な作業は、設定によって異なります。

2. SAP システム全体の */usr/sap/trans* ディレクトリを SAP プライマリアプリケーションサーバ上にホストするか、ファイルサーバ上にホストするかを決定してください。NFS で共有され完全にアクセス可能である限り、どちらにホストしても構いません。SAP プライマリアプリケーションサーバ上にホストして共有ファイルシステム上に置く場合、LifeKeeper で保護し、SAP 階層内に含める必要があります。

3. *SAP Installation Guide* に記載されている SAP およびデータベースのストレージ要件を考慮してください。大部分の SAP ファイルは、共有ストレージ上にインストールする必要があります。どのデータベースファイルが共有ストレージにインストールされ、どのデータベースファイルがローカルにインストールされるかについては、データベース固有リカバリキットの SPS for Linux テクニカルドキュメンテーション を参照してください。SAP 環境では、SAP がデータベースバイナリにローカルアクセスを必要とするため、これらはローカルにインストールする必要があることに注意してください。以上の要件を満たして、使用する共有ストレージに最適な使用方法を決定してください。

   また、共有ストレージリソースが LifeKeeper の保護下にある場合、そのリソースに一度にアクセス可能なサーバは 1 つだけになることに注意してください。共有デバイスがディスクアレイである場合、LUN 全体が保護されます。共有デバイスがディスクである場合、ディスク全体が保護されます。そのため、単一ボリュームに配置されているすべてのファイルシステムは、LifeKeeper によってまとめて制御されます。これは、データベース用に 1 つ、SAP 用に 1 つの、少なくとも 2 つの論理ボリューム (LUN) を用意する必要があることを意味します。

4. フェイルオーバ時にシステムを識別するため、仮想ホスト名が必要になります。使用する仮想ホスト名 1 つにつき 1 つの新しい IP アドレスが 必要です。この仮想ホスト名が DNS (ドメインネームシステム) 設定で正しく解決できることを確認したら、以下の手順に進みます。

   a. 以下のコマンドを使用して、新しい仮想 IP アドレスを作成してください。

   `ip addr add {IPADDRESS}/{NETMASK} dev eth0` (使用設定に合わせて、適切なネットマスクを使用してください)

**注記:** これらの新しい仮想IPアドレスを確認するには、`ip addr show` コマンドを使用します。

b. ERSv2 を使用する場合は、ERS インスタンス用に個別の仮想 IP も必要になります。

切り替え可能な IP アドレスと仮想サーバー名を関連付けるために、/etc/hosts を編集し、新しい仮想 IP アドレスと関連する仮想ホスト名を追加します。

**注記:** プライマリアプリケーションサーバとデータベースが常に同一のサーバ上で実行され、その通信が常にローカルである場合、この手順は必須ではありません。しかし、いつか異なるサーバで実行する場合に備えて、プライマリアプリケーションサーバとデータベースに対し、別々の切り替え可能 IP アドレスおよび仮想サーバ名を用意することを推奨します。

5. 両方マシンでキャッシングデーモンを停止してください。

```
rcnscd stop
```

6. ソフトウェアをマウントしてください。

```
mount //{path of software} (no password needed)
```

7. X セッション (ssh -X または VNC セッション、Microsoft Windows ユーザの場合、 Hummingbird Exceed X Windows を使用可能) を実行してください。

**注記:** `sapinst` が実行されている場合、ディレクトリは */tmp* 配下に抽出されます。

8. SAP（具体的にはASCSまたはSCS）をインストールするときは、SAPINST_USE_HOSTNAME={仮想ホスト名} オプションを指定する必要があります。 これは、ERSv2を使用している場合はERSインスタンスにも必要ですが、ERSv1を使用している場合は必要ありません。

**注記:** `sapinst SAPINST_USE_HOSTNAME = {仮想ホスト名}` を指定します。ここでは、 **仮想ホスト名** は、ノード間でフロートする仮想IPに解決されるホスト名です。

# 重要事項

LifeKeeper SAP Recovery Kit は、インストールされている SAP Host Agent に依存します。このソフトウェアがインストールされていない場合、LifeKeeper SAP Kit はインストールされません。SAP Netweaver Version 7.3 ではこの Host Agent が提供されていますが、それより前のバージョンでは SAP からダウンロードする必要があります。使用する特定のバージョンについては、SAP のヘルプを参照することをお勧めします。また、詳細なドキュメンテーションをヘルプフォーラム (help.sap.com) で入手できます。

- RPM または SAR フォーマットの saphostexec モジュールを、SAP からダウンロードできます。
- モジュールが正しくインストールされているか確認するには、いくつかのモジュール ( *saposcol*、*saphostexec*、*saphostctrl* ) を検索します。SAP がインストールされている場合、通常これらのモジュールが見つかります (一般的に */usr/sap* ディレクトリ)。

# 6.17.7.2. ファイルシステムとディレクトリ構造の作成

使用するデーターベース管理システムによって多くの異なる設定が可能ですが、準拠すべき基本的なレイアウトを以下に示します。

> レプリケーション（SIOS DataKeeper）ファイルシステムのリソースはSAPリソースを作成する前に作成する必要があります。

- 仮想 IP リソースを *etc/hosts* に追加する
- インスタンスおよびデータベースホストの仮想 IP リソースを作成する
- 共有ディスクを構築する
  - ファイバー チャネルまたは iSCSI 上の共有ディスク
    - ファイルシステムを作成する
  - SIOS DataKeeper を使用して複製されたディスク
  - NFS 共有
  - AWS EFS デバイス
- メインの SAP ファイル システムをマウントする
- PAS、ASCS/SCS、および ERS ディレクトリ (該当する場合) と追加のアプリケーションサーバーをマウントする。

データーベースのディレクトリ構造の詳細については、データーベース管理システム固有の SAP インストールガイドを参照してください。データーベース用の LifeKeeper Recovery Kit の保護下に置くためには、すべてのデーターベースファイルを共有ディスク上に配置する必要があります。データーベースを保護するための追加情報については、データーベース固有の Recovery Kit ドキュメンテーション を参照してください。

仮想ホスト名: sap-shc-nfs、SAP SID: SHC、SAP インスタンス: ASCS10、ERS インスタンス: ERS20 のネットワーク共有デバイスを使用する例です。

1. ネットワーク共有デバイスを作成し、DNS 名または IP アドレスを介してノードに接続する
2. ディレクトリを作成する

```
# mkdir -p /sapmnt/SHC
# mkdir -p /usr/sap/SHC/ASCS10
# mkdir -p /usr/sap/SHC/ERS20
# mkdir -p /usr/sap/SHC/SYS
```

3. すべてのノードの /etc/fstab にエントリを追加する

```
sap-shc-nfs:/sapmnt/SHC                    /sapmnt/SHC              nfs
vers=4,proto=tcp,rw,sync,bg    0 0
sap-shc-nfs:/usr/sap/SHC/ASCS10    /usr/sap/SHC/ASCS10 nfs   vers=4,prot
o=tcp,rw,sync,bg   0 0
```

```
sap-shc-nfs:/usr/sap/SHC/ERS20    /usr/sap/SHC/ERS20    nfs  vers=4,pro
to=tcp,rw,sync,bg   0 0
sap-shc-nfs:/usr/sap/SHC/SYS           /usr/sap/SHC/SYS     nfs  ver
s=4,proto=tcp,rw,sync,bg   0 0
```

4. すべてのノードにファイル システムをマウントする

```
# mount /sapmnt/SHC
# mount /usr/sap/SHC/ASCS10
# mount /usr/sap/SHC/ERS20
```

# 6.17.7.3. コアサービスのインストール

ソフトウェアをインストールする前に、すべてのサーバ上で日付と時刻が同期していることを確認してください。これは、LifeKeeper と SAP の両方にとって重要です。

コアサービス、ABAP セントラルサービス (ASCS) および Java セントラルサービス (SCS) は、単一障害点 (SPOF) であり、LifeKeeper で保護する必要があります。適切な SAP インストールガイドを使用して、コアサービスを SAP プライマリサーバ上にインストールしてください。

## インストール上の注記

- 設定の計画 トピックで作成した必要な仮想ホスト名を使用するには、SAPinst 開始前に SAPinst プロパティの `SAPINST_USE_HOSTNAME` を設定し、必要な仮想ホスト名を指定します。( **注記** :`SAPINST_USE_HOSTNAME` 仮想 IP アドレスは、後で LifeKeeper で SAP リソースを作成するときに使用するため、記録しておいてください。)

  次のコマンドを実行: `./sapinst SAPINST_USE_HOSTNAME={hostname}`

- 7 つの段階で、 **コアサービス** を作成して、起動する必要があります。`jdbcconnect.jar` でパーミッションエラーが発生する場合、 */sapmnt/STC/exe/uc/linuxx86_64* に移動し、ディレクトリとファイル `jdbcconnect.jar` を書き込み可能にしてください (chmod 777 ---)。

- 正常完了のメッセージが表示され、インストールが完了します。

# 6.17.7.4. データベースのインストール

1. dba と oinstall のグループ ID をメモしてください。これは、バックアップマシンで必要になります。

2. ソフトウェアのディレクトリに移動し、次のコマンドを実行してください。

```
./sapinst SAPINST_USE_HOSTNAME={database connectivity ip address}
```

3. 適切な SAP インストールガイドを使用して、SAPinst を実行し、データベースインスタンスをインストールしてください。

## インストール上の注記

- SIOS は、orarun パッケージがデータベースインスタンスのインストール前に既にインストールされている場合、orarun パッケージを削除することを推奨します ( **SAP ノート 1257556** を参照)。

- SAPinst ウィンドウのデータベースインストールオプションは、Oracle を除き、データベースソフトウェアが既にインストールされていることを前提としています。Oracle データベースの場合、SAPinst はインストールを停止し、データベースソフトウェアのインストールを要求します。

- <DBSID> は、データベースインスタンスを識別します。SAPinstは、データベースインスタンスをインストールスするときに <DBSID> の入力を要求します。<DBSID> は、<SAPSID> と同じにすることができます。

- SAP グローバルホスト以外のホストにデータベースをインストールする場合、SAP グローバルホストからグローバルディレクトリをマウントする必要があります。

- Listener が起動される時点で問題が発生した場合、以下のコマンドを使用して終了させてください。

```
(ps -ef | grep lsnrctl)
```

- SAPR3 と SAPR3DB のユーザ ID のパスワードをリセットするには、以下のコマンドを使用してください。

```
brtools
```

データベースのインストールが完了したら、下のダイアログを閉じ、SAP のインストール、すなわちアプリケーションサービスのインストール に進んでください。

# 6.17.7.5. プライマリアプリケーションサーバインスタンスのインストール

1. プライマリアプリケーションサーバインスタンスをインストールするには、先述したディレクトリから sapinst を再実行します。

   ./sapinst SAPINST_USE_HOSTNAME= < vip >

2. 要求された場合、**[Primary Application Server Instance]** を選択し、適切な SAP インストールガイドを使用してインストールを続行します。

## インストール上の注記

- プライマリアプリケーションサーバインスタンスは、既に単一障害点 (SPOF) ではなくなっているため、クラスタの一部にする必要はありません。この時点で、SPOF は、クラスタによって保護されているセントラルサービスインスタンス (SCS インスタンスおよび ASCS インスタンス) 内にあります。

- プライマリアプリケーションサーバインスタンスのディレクトリの名前は、DVEBMGS<*No*> です (<*No*> はインスタンス番号)。

- **[OK]** メッセージを受信すると、アプリケーションサーバのインストールが完了します。

- レプリケートされたエンキューを 7.1 上にインストールする場合、`sapinst` をそのまま実行します。

# 6.17.7.6. 追加のアプリケーションサーバインスタンスのインストール

冗長性を実現するため、追加のアプリケーションサーバインスタンスをインストールすることを推奨します。アプリケーションサーバインスタンスは SPOF ではないため、クラスタに含める必要はありません。

各追加アプリケーションサーバインスタンス上で、次の作業を実行してください。

1. `SAPinst` を実行し、追加のアプリケーションサーバインスタンスをインストールしてください。

2. 要求された場合、 **[Additional Application Server Instance]** を選択し、適切な SAP インストールガイドを使用してインストールを続行します。

# 6.17.7.7. バックアップサーバへのインストール

バックアップサーバでは、プライマリサーバ上で実行した以下のインストール手順をもう一度実行します。

1. コアサービス、ABAP セントラルサービス (ASCS)、および Java セントラルサービス (SCS) のインストール
2. データベースのインストール
3. アプリケーションサービスのインストール

# 6.17.7.8. LifeKeeper のインストール

プライマリー サーバと バックアップ サーバの両方において、LifeKeeper ソフトウェアを次のリカバリキットを含めてインストールしてください。

- SAP
- 適切なデータベース (Oracle、SAP DB など)
- IP
- NFS
- NAS
- Quorum (steeleye-lkQWK package) ※ネットワーク共有デバイスを使用している場合

1. 両方のマシンで **Oracle Listener** と **SAP** を停止してください。

   例えば、Oracle ユーザが *orastc* で、Oracle Listener が *LISTENER_STC* で、SAP ユーザが *stcadm* の場合、次のようになります。

   a. `su` コマンドでユーザ `orastc` に切り替え、コマンド `lsnrctl stop LISTENER_STC` を実行してください。

   b. `su` コマンドでユーザ `stcadm` に切り替え、コマンド `stopsap sap{No.}` を実行してください。

   c. root ユーザとして、SAP または Oracle ユーザプロセスが実行されていないことを確認します。もし実行されている場合、「`killall sapstartsrv`」と入力してください。このコマンドを実行した後もプロセスが存在する場合は、「ps ?ef」を実行して各プロセスを終了してください。

2. 両方のマシンで */etc/hosts* を開き、ホストと DNS のエントリが正しく指定されていることを確認してください。

3. 現在のインターフェースから IP アドレスを **停止** および **削除** してください。 **注記**: この手順は、IP アドレスを LifeKeeper IP Recovery Kit で保護する前に行う必要があります。

   ```
   ip addr delete {A/SCS VIRTUAL IP ADDRESS}/{NETMASK} dev eth0

   ip addr delete {ERSv2 VIRTUAL IP ADDRESS}/{NETMASK} dev eth0
   ```

4. ping などを使用して接続試行を行い、IP アドレスが削除されていることを確認してください。

5. LifeKeeper インストールガイド に記載の手順に従って、プライマリーサーバーとバックアップサーバーの両方に LifeKeeper をインストールしてください (DE、コア、DataKeeper、LVM、およびラ

イセンス)。Recovery Kit を選択するよう求められたら、以下を選択してください。

SAP、適切なデータベース (Oracle)、IP、NFS、NAS

インストールスクリプトが一定のチェックを実行します。以下の例に示すように、環境が正しく設定されない場合、失敗することがあります。

```
SAP Services file /usr/sap/sapservices not found

SAP Installation is not valid; please check environment and retry

error: %pre(steeleye-lkSAP-7.3.1-1.noarch) scriptlet failed, exit
status 2

error: install: %pre scriptlet failed (2), skipping steeleye-
lkSAP-7.3.1-1
```

上記の例では、期待される SAP ファイル、/usr/sap/sapservices が欠落しています。インストールを続行する前に、環境が正しい状態になることはとても重要です。

リカバリキットのインストールおよびリソースを保護するためのサーバの設定に関する詳細は、Recovery Kit ドキュメンテーション を参照してください。

6. すべてのサーバー間で LifeKeeper LifeKeeper コミュニケーションパス をセットアップします。

7. ネットワーク共有デバイスを使用する場合のQuorumの設定 (Quorum-Witness).
この例では、クォーラムは各サーバーの /etc/default/LifeKeeper で次の設定を使用して “Majority Mode” で構成されています。

QUORUM_MODE=majority

WITNESS_MODE=remote_verify

# 6.17.7.9. ASCS および ERS インスタンスプロファイル設定の変更

エンキューサーバーおよびエンキューレプリケータープロセス（それぞれ ASCS および ERS インスタンスプロファイル内に存在）の Restart_Program パラメーターを Start_Program に変更して、sapstart ユーティリティがそれらを自動的に再起動しないようにします。また、両方のインスタンスプロファイルのエントリ「Autostart = 1」を「Autostart = 0」に変更し、システムの再起動時に各インスタンスが自動的に再起動しないようにしてください。

次の手順を実行する前に、アクティブな ASCS および ERS インスタンスプロファイルの場所を特定します。これは、「pf=」パラメーターとして sapstartsrv に提供されたファイルのパスを調べることにより、/usr/sap/sapservices ファイルで見つけることができます。

> ✻ 以下は例です。ここでは ASCS インスタンス番号として00、ERS インスタンス番号として10を使用していますが、ご利用の環境に合わせて番号を変更してください。また、<SID>を実際のシステムIDに置き換えてください。

```
LD_LIBRARY_PATH=/usr/sap/SID/ASCS00/exe:$LD_LIBRARY_PATH; export

LD_LIBRARY_PATH; /usr/sap/SID/ASCS00/exe/sapstartsrv

pf= /usr/sap/SID/SYS/profile/SID_ASCS00_sap1 -D -u SIDadm

LD_LIBRARY_PATH=/usr/sap/SID/ERS10/exe:$LD_LIBRARY_PATH; export

LD_LIBRARY_PATH; /usr/sap/SID/ERS10/exe/sapstartsrv

pf= /usr/sap/SID/SYS/profile/SID_ERS10_sap2 -D -u SIDadm
```

この例では、アクティブな ASCS インスタンスプロファイルは /usr/sap/SID/SYS/profile/SID_ASCS00_sap1 に、アクティブな ERS インスタンスプロファイルは /usr/sap/SID/SYS/profile/SID_ERS10_sap2 にあります。

## 手順

1. 以下のように、ASCSインスタンスプロファイルを編集します。

   a.「Autostart = 1」を「Autostart = 0」に変更するか、「Autostart = 0」という行を手動で追加します。

   b. エンキューサーバープロセスを開始するプロファイル内の行の正確な形式は、スタンドアロンエンキューサーバーフレームワークのバージョン 1 または 2 のどちらが使用されているかによって異なります。次の表に従ってプロファイルを変更してください。

| 変更前（この行が表示されたら、右の列の内容に変更してくだ | 変更後 |
|---|---|

| さい) | |
|---|---|
| Restart_Program_01 = local $(_EN) pf=$(_PF) | Start_Program_01 = local $(_EN) pf=$(_PF) |
| Restart_Program_01 = local $(_ES2) pf=$(_PF) | Start_Program_01 = local $(_ES2) pf=$(_PF) |
| Restart_Program_01 = local $(_ENQ) pf=$(_PF) | Start_Program_01 = local $(_ENQ) pf=$(_PF) |

✻ Start_Program_xx または Restart_Program_xx エントリの番号 xx は、お使いのシステムでは異なる場合があります。
上記の内容に一致するように変更する必要はありません。

2. すべてのエントリが正しいことを確認し、ASCS インスタンスプロファイルへの変更を保存します。

3. ERSインスタンスプロファイルを次のように編集します。

   a.「Autostart = 1」を「Autostart = 0」に変更するか、「Autostart = 0」という行を手動で追加します。

   b. エンキューレプリケーターサーバープロセスを開始するプロファイル内の行の正確な形式は、スタンドアロンエンキューサーバーフレームワークのバージョン1または2のどちらが使用されているかによって異なります。次の表に従ってプロファイルを変更してください。

| 変更前（この行が表示されたら、右の列の内容に変更してください） | 変更後 |
|---|---|
| Restart_Program_00 = local $(_ER) pf=$(_PFL) NR=$(SCSID) | Start_Program_00 = local $(_ER) pf=$(_PFL) NR=$(SCSID) |
| Restart_Program_00 = local $(_ER2) pf=$(_PF) | Start_Program_00 = local $(_ER2) pf=$(_PF) |
| Restart_Program_00 = local $(_ENQR) pf=$(_PF) | Start_Program_00 = local $(_ENQR) pf=$(_PF) |

✻ Start_Program_xx または Restart_Program_xx エントリの番号 xx は、お使いのシステムでは異なる場合があります。
上記の内容に一致するように変更する必要はありません。

4. すべてのエントリが正しいことを確認し、変更を ERS インスタンスプロファイルに保存します。

5. システムを再起動して、更新されたプロファイルが読み込まれ、キャッシュが有効になっていないことを確認することを推奨します。

6. また、本番環境のワークロードに適用する前に、常にテスト環境での変更を検証することを推奨します。

autostart の値を変更した場合は、変更を反映させるため再起動が必要になります。

注記：

1. エントリ「Autostart = 0」と「Start_Program_xx ...」は、各インスタンスプロファイルの別々の行にある必要があります。

2. Start_Program_xx または Restart_Program_xx エントリの番号 xx は、お使いのシステムでは異なる場合があります。この例で指定されている数字と一致するように変更する必要はありません。

3. Java ベースまたはデュアルスタックの Java+ABAP デプロイメントには、SCS セントラルサービスインスタンスがあります。この場合、SCS インスタンスに対しても手順 1〜3 を実行する必要があります。

4. Start_Program と Restart_Program の違いの詳細については、SAP Note 768727（Automatic restart functions in sapstart for processes）を参照してください。

# 6.17.7.10. SAP IP リソース

LifeKeeper リソース階層の設定を続行する前に、SAP リソースがフェイルオーバまたはスイッチオーバに使用する IP アドレスを決定してください。これは、通常、SAP のインストール中にパラメータ SAPINST_USE_HOSTNAME で使用した仮想 IP アドレスです。この IP アドレスは、クラスタ内のノード間で共有される 1 つの仮想 IP アドレスで、一度に 1 つのノード上でアクティブになります。この IP アドレスは、データベース階層を保護するために使用される IP アドレスとは異なります。これらの IP アドレスは、SAP リソースを作成するときにも使用するので、メモしておいてください。

# 6.17.7.11. SAP リソース階層の作成

**重要:** 業界のベストプラクティスに従い、SIOS では、SAP システムのセットアップ (「ループバック構成」とも呼ばれます) 時に、NFS サーバー/クライアントを SAP サービスと同じノードにインストールすることを推奨しなくなりました。 代わりに、次のような代替構成を使用することをお勧めします。
- EFS in AWS
- Azure Premium File Shares with NFS
- Google NFS Share
- NFS as separate cluster

今後、「ループバック構成」を使用した新規インストールはサポートされません。 問題が SIOS ソフトウェア内にある場合、既存のお客様は可能な範囲でサポートいたしますが、代替構成への移行が推奨されます。 次のリリースでは、既存の構成はサポートされません。

SAP システムを保護するには、SAP 階層が必要です。この SAP 階層は、コア (セントラルサービス) リソース、ERS リソース、プライマリーリソース、セカンダリーリソースで構成されます。この階層を作成するには、プライマリーサーバから次の作業を実行します。 **注記:** 以下の例は、階層作成の指針として提供するものです。実際の構成によって、作業は少々異なります。

## コアリソースの作成

1. LifeKeeper GUI メニューから **[Edit]** を選択し、次に **[Server]** を選択します。ドロップダウンメニューから、 **[Create Resource Hierarchy]** を選択します。

   ダイアログボックスが表示され、クラスタ内にインストールされ認識されているすべての Recovery Kit が、ドロップダウンリストボックスに一覧で示されます。ドロップダウンメニューから **[SAP]** を選択します。

   **[Next]** をクリックします。

   ダイアログボックスで **[Back]** ボタンが有効な場合は、前のダイアログボックスに戻ることができます。これは、エラーが発生して、以前に入力した情報を訂正する必要がある場合に特に役立ちます。

   階層作成の途中で **[Cancel]** ボタンをクリックすると、作成処理全体が取り消されます。

2. **[Switchback Type]** を選択します。この選択によって、バックアップサーバへのフェイルオーバの後、SAP インスタンスがサービス中に戻るときに、このサーバにどのようにスイッチバックされるのかが決まります。 **[intelligent]** または **[automatic]** を選択できます。 **[Intelligent switchback]** の場合、インスタンスをプライマリ / オリジナルサーバにスイッチバックするときに管理者の介入が必要になります。 **[Automatic switchback]** の場合、プライマリサーバがオンラインに戻り、LifeKeeper コミュニケーションパスを再確立した直後に自動的にスイッチバックが行われます。

スイッチバックタイプは、必要に応じて **[Resource Properties]** ダイアログボックスの **[General]** タブで後から変更できます。

**[Next]** をクリックします。

3. SAP PAS、ASCS、または SCS を配置するサーバ (通常これは、プライマリーサーバまたはテンプレートサーバと呼ばれます) を選択します。クラスタ内のすべてのサーバがドロップダウンリストに表示されます。

**[Next]** をクリックします。

4. **[SAP SID]** を選択します。これは、保護対象の SAP PAS、ASCS、または SCS システムのシステム識別子です。

**[Next]** をクリックします。

5. 保護対象の SID に対する SAP インスタンス名 (例: *ASCS<No.>* ) (コアインスタンスが先) を選択します。

**[Next]** をクリックします。

**注記**: 保護およびリカバリーレベルのカスタマイズに関して、その他の画面が表示されることがあります。

6. **[IP Child Resource]** を選択します。これは、一般的に、SAP インストール中にメモした仮想ホスト IP アドレス (SAPINST_USE_HOSTNAME) か、フェイルオーバに必要な IP アドレスのいずれかです。

a. LifeKeeperがインスタンスの **依存ファイルシステムの作成を自動化するかどうか** を選択します。 **yes** を選択すると、LifeKeeperは必要なファイルシステムリソースを作成し、それをSAPリソースの下に依存関係として追加します。（複製されたファイルシステムは自動では作成できません）。 **no** を選択すると次のダイアログボックスが表示され、依存関係として追加する既存のLifeKeeperファイルシステムリソースを選択するよう求められます。

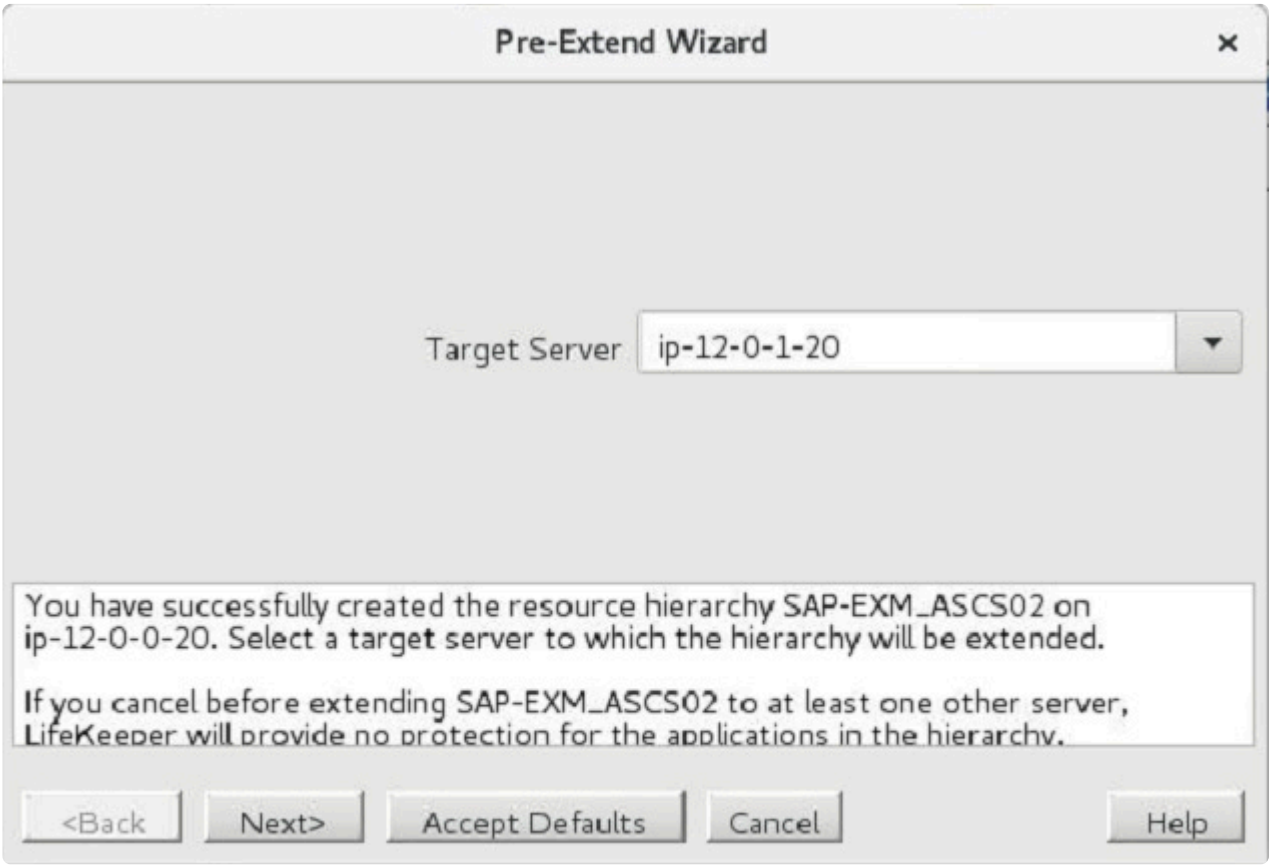
b. 前のダイアログで **no** を選択した場合、SAPリソースの下に依存関係として追加されるファイルシステムリソースを選択するオプションが提供されます。Ctrlキーを押しながら各リソースをクリックすると、複数のリソースを選択できます。 **注記：** このダイアログで選択肢として表示するには、LifeKeeperのこのサーバでファイルシステムリソースがIn Service（ISP）になっている必要があります。

7. **[SAP Tag]** を選択または入力します。これは、LifeKeeper が SAP リソースに与えるタグ名です。デフォルトを選択するか、独自のタグ名を入力することができます。デフォルトのタグは、*SAP-<SID>_<INST>* です。

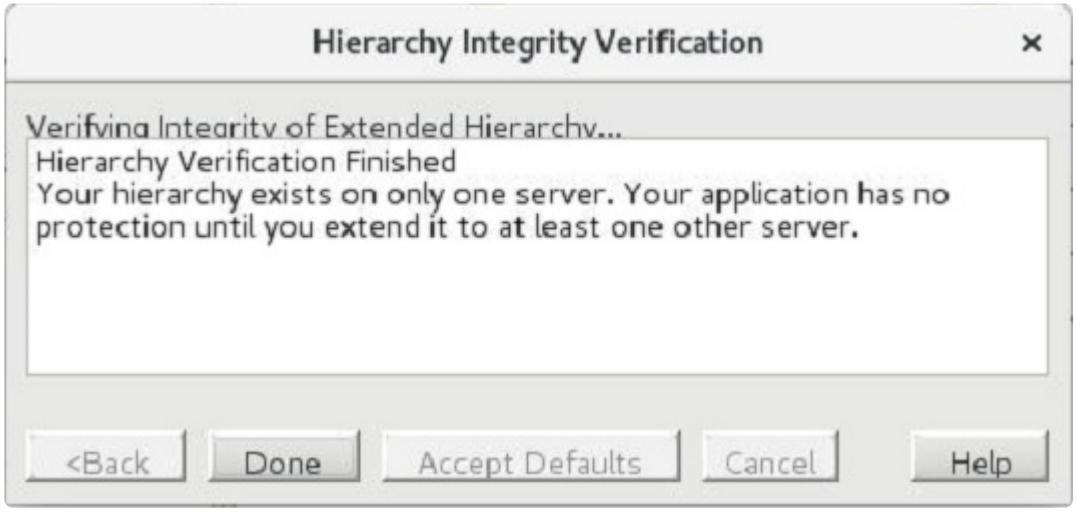
**[Create]** をクリックすると、 ***[Create SAP Resource Wizard]*** によって SAP リソースが作成されます。

8. 情報ボックスが表示され、SAP リソース階層が正常に作成されたことが示されます。SAP リソース階層を LifeKeeper の保護下に置くには、その階層をクラスタ内の別のサーバに **拡張** する必要があります。

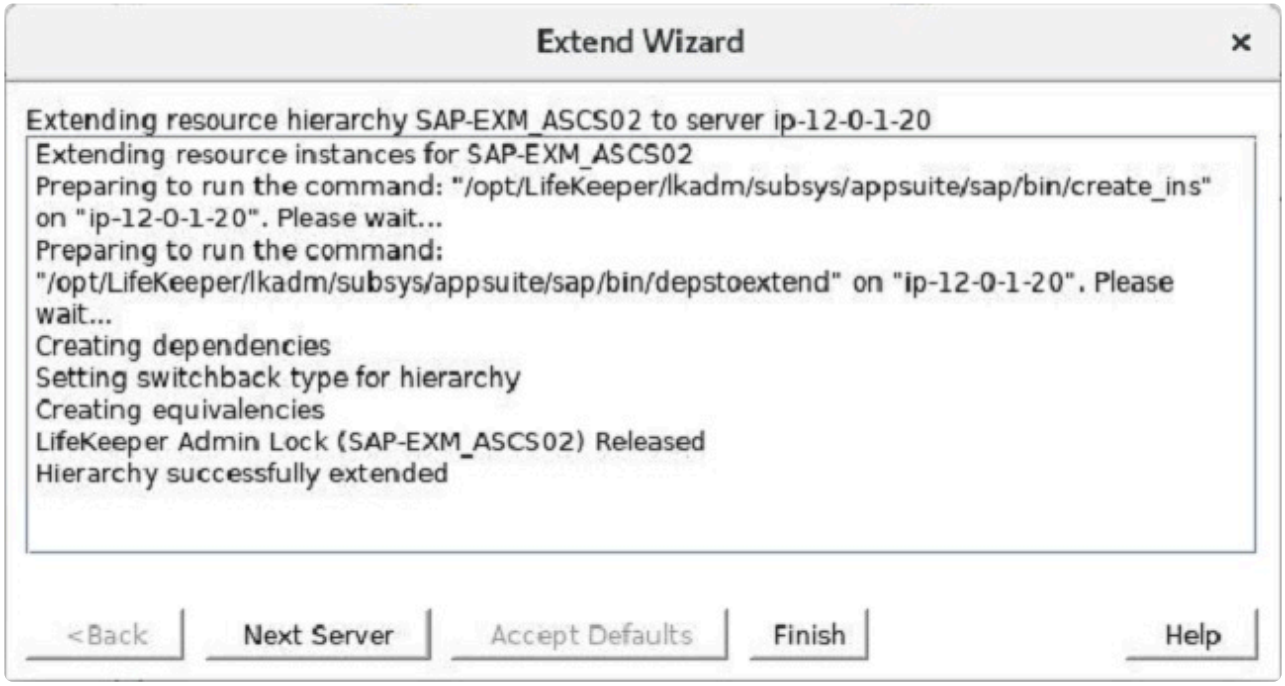
   **[Next]** をクリックすると、 ***[Pre-Extend Wizard]*** が起動します。このウィザードについては、本セクションで後述します。




ここで **[Cancel]** をクリックすると、別のダイアログボックスが表示され、いずれかの時点でここに戻り、SAP リソース階層を別のサーバに拡張して、LifeKeeper の保護下に置く必要があることが警告されます。




9. ***[Extend Wizard]*** ダイアログに「 **Hierarchy successfully extended** 」と表示されます。 **[Finish]** をクリックします。




10. **[Hierarchy Integrity Verification]** ダイアログが表示されます。階層の検証が終了したら、 **[Done]** をクリックして、 **[Create Resource Hierarchy]** メニューを終了します。

**コアが最上位レベルにある階層**

# ERS リソースの作成

ERSリソースは、コアインスタンス（セントラルサービスインスタンス）またはエンキューサーバプロセスの単一障害点に対する追加の保護を提供します。コアインスタンス（セントラルサービスインスタンス）に障害が発生して再起動されると、コアインスタンスはエンキューレプリケーションサーバからエンキューロックテーブル（レプリケーションテーブル）のバックアップコピーを取得します。その結果、エンキューサーバに障害が発生してもトランザクションや更新は失われず、SAPシステムのサービスが継続されます。

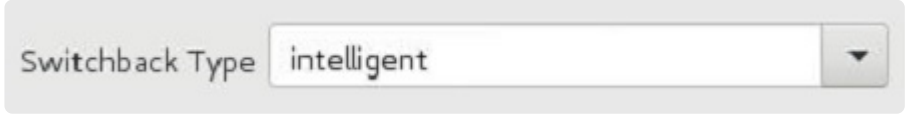
SPS-Lのバージョン9.4.0以降と9.4.0より前でのERSリソースの実装の違いの説明については、LifeKeeperのERSリソースタイプ を参照してください。

> **重要な注記:** ERS階層の作成と拡張は、 **対応するASCS/SCSインスタンスがLifeKeeperでIn Service (ISP)になってないサーバで実施する必要があります。** ERSv2の場合、ベースとなるERSファイルシステムリソースは、リソースが作成されるバックアップサーバで最初にIn Serviceにする必要があります。

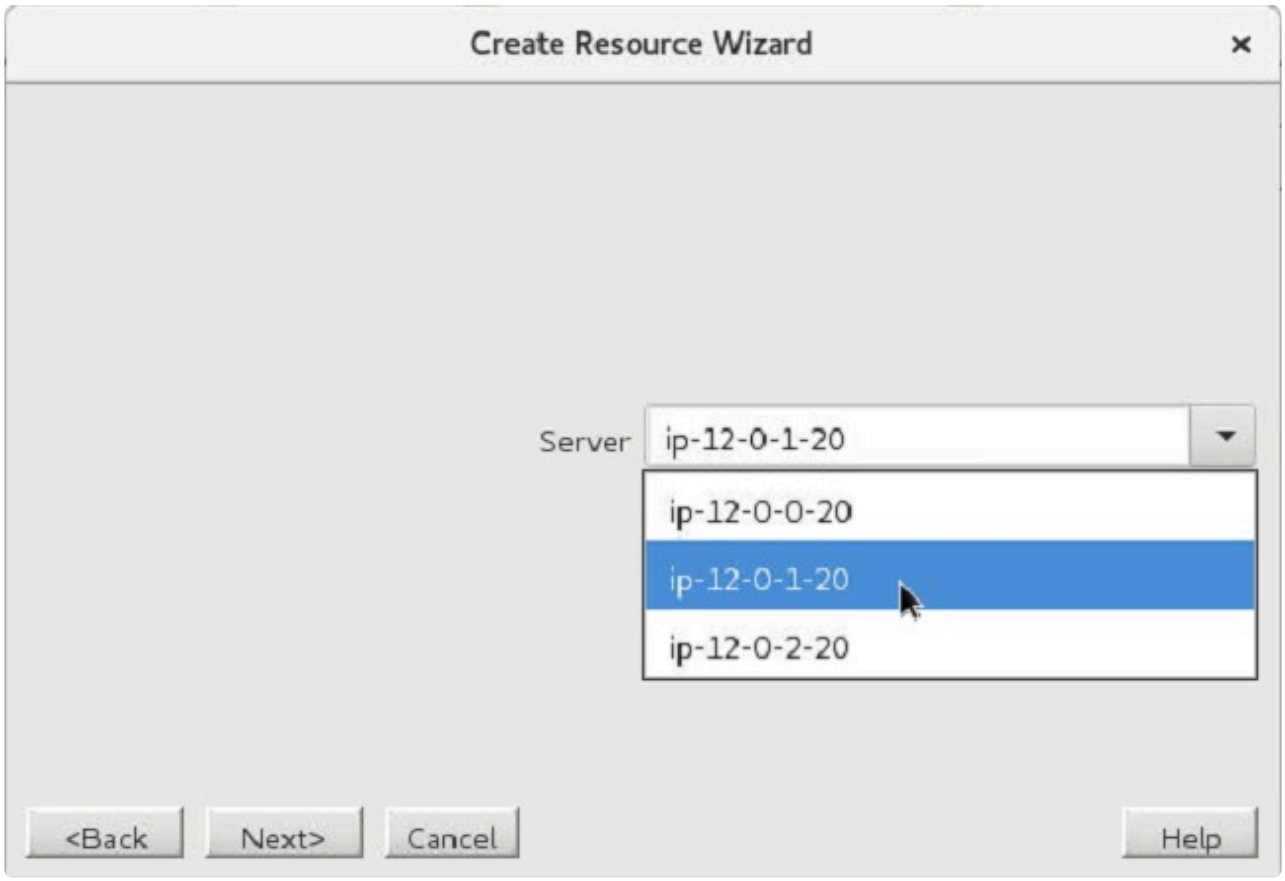
次の手順を実行して、ERSリソースを作成します。

1. LifeKeeper GUIメニューから、 **[Edit]** 、 **[Server]** の順に選択します。ドロップダウンメニューから、 **[Create Resource Hierarchy]** を選択します。クラスタ内にインストールされ、認識されているすべてのリカバリーキットを含むドロップダウンリストボックスのあるダイアログボックスが表示されます。ドロップダウンリストから **SAP** を選択し、 **[Next]** をクリックします。

2. **Switchback Type** を選択し、 **[Next]** をクリックします。



3. **重要:** 対応するASCS/SCSインスタンスが **ISP**ではない クラスタ内の サーバ を選択します。 **[Next]** をクリックします。

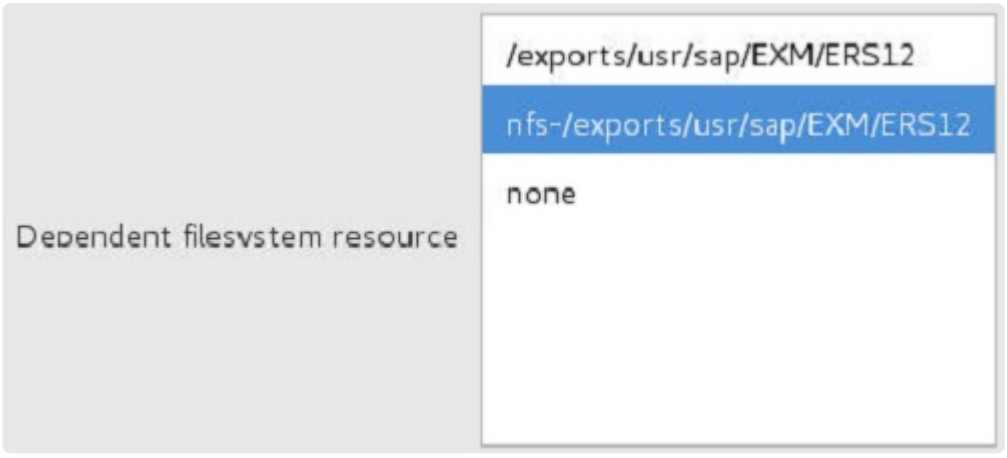



4. ERSインスタンスの **SAP SID** を選択し、 **[Next]** をクリックします。

5. **SAP**インスタンス名 (例. ERS<No.>) を選択し、 **[Next]** をクリックします。

6. ERSv2インスタンスを表すリソースを作成する場合は、 **IP Child Resource** を選択します。ERSv1インスタンスを表すリソースを作成する場合、この選択画面は表示されません。

7. ERSv2インスタンスを表すリソースを作成する場合、ERSリソースの下に依存関係として追加する **Dependent Filesystem Resource** を選択します。このリストに表示させるには、LifeKeeperのこのサーバでファイルシステムリソースがIn Service（ISP）になっている必要があります。ERSv1インスタンスを表すリソースを作成する場合、この選択画面は表示されません。




8. **SAP Tag** を選択するか、入力します。

9. プロンプトに従って、 **リソース階層を拡張します** 。 **注記:** ERSv1インスタンスを表すリソースは、1つのバックアップサーバにのみ拡張できます。ERSv1階層を3番目のクラスタノードに拡張しようとすると、階層の拡張は失敗します。

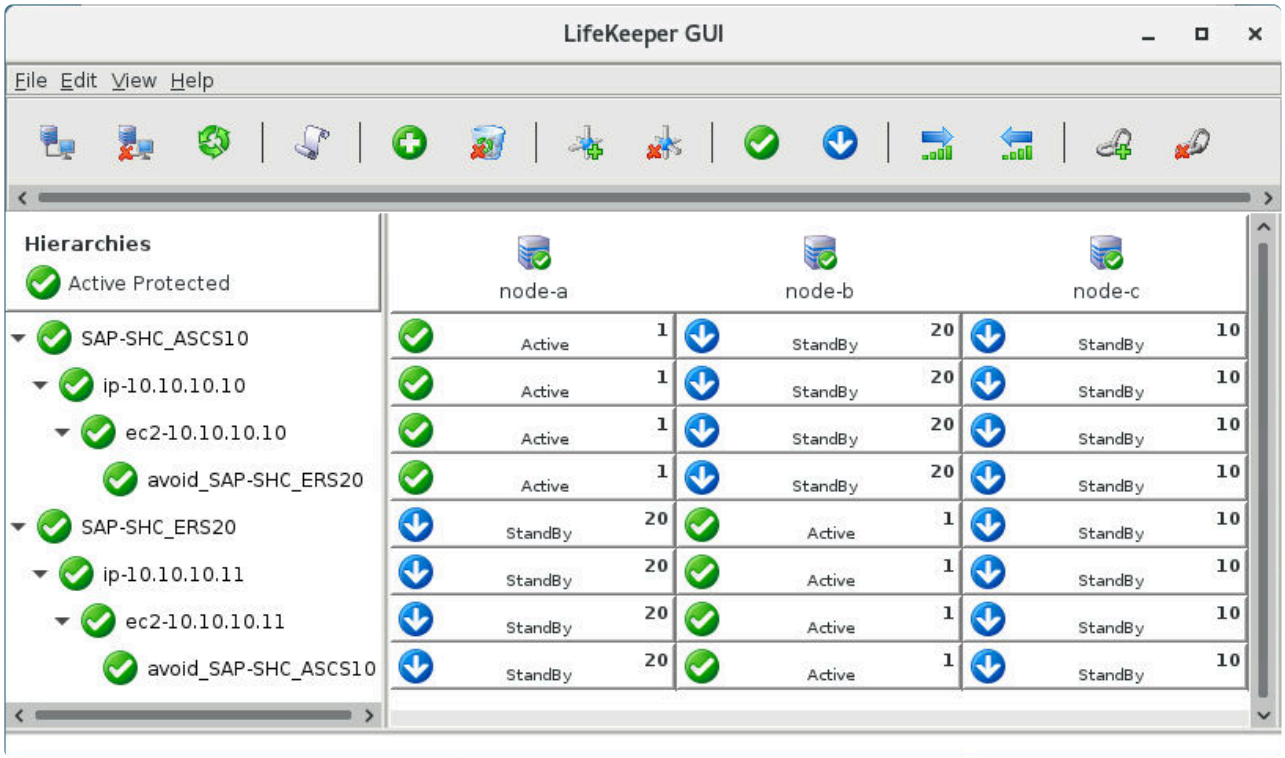
10. **[Hierarchy Successfully Extended]** と表示されたら、 **[Finish]** を選択します。

11. **[Done]** を選択します。

**ASCSおよびERSv1が別の階層にあります**

**ASCSおよびERSv2が別の階層にあります**

ASCSインスタンスがENSAv2を実行し、ERSインスタンスがERSv2を実行しており、3つ以上のノードを持つクラスタ内の同じシステムに階層が拡張されている場合、ASCSおよびERSリソース階層は、特別な「階層回避」Genericアプリケーションリソースを使用することにより、スイッチオーバーとフェイルオーバーで互いを回避するように強制できます。
**注記**：これらの階層回避リソースは、ENSAv1/ERSv1 構成または 2 ノードクラスターでは使用しないでください。 以下は、「階層回避」リソースが構成された 3 ノードクラスターです。




## プライマリアプリケーションサーバリソースの作成

1. ここでも同様に、同じ SAP SID について上記の手順を繰り返し、 **DVEBMGS{ *XX* }** (ここで {XX} はインスタンス番号) を選択して、プライマリアプリケーションサーバリソースを作成します。

2. 画面の指示に従って、 **[Level of Protection]** を選択します (デフォルトは **[FULL]** )。 **[Next]** をクリックします。

3. 画面の指示に従って、 **[Level of Recovery]** を選択します (デフォルトは **[FULL]** )。 **[Next]** をクリックします。

4. **[Dependent Instances]** の入力を要求されたら、「親」インスタンス、すなわち、上記で作成した **ERS** インスタンス を選択します。

5. **[IP Child Resource]** を選択します。

6. 画面の指示に従ってリソース階層を拡張します。

7. 「 **Hierarchy Successfully Extended** 」と表示されたら、 **[Finish]** をクリックします。

8. **[Done]** を選択します。

プライマリアプリケーションサーバが最上位レベルにある階層

## セカンダリアプリケーションサーバリソースの作成

必要な場合、セカンダリアプリケーションサーバリソースを同様の方法で作成します。

**注記:** コマンドラインでの手順については、コマンドラインによる SAP の設定 を参照してください。

# 6.17.7.12. SAP リソース階層の削除

LifeKeeper 設定内の すべての サーバからリソースを削除するには、次の手順を実行します。

> 注記: 階層全体を削除するには、各リソースを個別に削除する必要があります。

1. **[Edit]** メニューの **[Resource]** から **[Delete Resource Hierarchy]** を選択します。
2. リソース階層を削除する **[Target Server]** の名前を選択します。

   注記: 左側ペインのグローバルリソースまたは右側ペインの個々のリソースインスタンスを右クリックして **[Delete Resource]** 作業を選択した場合、このダイアログボックスは表示されません。

   **[Next]** をクリックします。
3. **[Hierarchy to Delete]** を選択します。削除するリソース階層を特定して、強調表示にします

   注記: このダイアログは、右ペインまたは左ペインのリソースインスタンス上で右クリックして **[Delete Resource]** を選択した場合には表示されません。

   **[Next]** をクリックします。
4. 選択したターゲットサーバと、削除の対象として選択した階層を確認する情報ボックスが表示されます。 **[Delete]** をクリックします。
5. リソースが正常に削除されたことを確認する別の情報ボックスが表示されます。 **[Done]** をクリックして終了します。

# 6.17.7.13. SAP Recovery Kit の一般的な作業

以下の作業は、すべての Recovery Kit を通じて同一の手順からなる一般的な作業のため、LifeKeeper for Linux テクニカルドキュメンテーション の管理 セクションで説明しています。

- リソース依存関係の作成 。既存のリソース階層と別のリソースインスタンスとの間に親子の依存関係を作成し、クラスタ内のすべての対象サーバに依存関係の変更を反映します。
- リソース依存関係の削除 。リソースの依存関係を削除して、クラスタ内のすべての対象サーバに依存関係の変更を反映します。
- In Service ：リソース階層を特定のサーバで in service にします。
- Out of Service ：リソース階層を特定のサーバで out of service にします。
- プロパティの表示 /編集 ：特定のサーバでリソース階層のプロパティを表示または編集します。

# 6.17.7.14. コマンドラインからの SAP のセットアップ (LKCLI)

LKCLI (コマンドラインインターフェイス) を使用して、SAP Recovery Kit をセットアップできます。

## 用語解説

- **Switchback Type** - この選択によって、バックアップサーバーへのフェイルオーバーの後、SAP インスタンスが In Service に戻るときに、このサーバーにどのようにスイッチバックされるのかが決まります。 **intelligent** または **automatic** を選択できます。intelligent の場合、インスタンスをプライマリー / オリジナルサーバーにスイッチバックするときに管理者の介入が必要になります。Automatic スイッチバックの場合、プライマリーサーバーがオンラインに戻り、LifeKeeper のコミュニケーションパスを再確立した直後に自動的にスイッチバックが行われます。
- **IP Tag** - SAP リソース階層の依存リソースになる IP リソースを表します。
- Protection Level - **Protection Level** は、各リソースに実行できるアクションを表します。
- Recovery Level - **Recovery Level** は、障害発生時のリソースに対する指示を提供します。
- **Additional SAP Dependents** - この値は、現在作成しようとしている SAP リソースの依存リソースになる LifeKeeper SAP リソースタグを表します。

## コマンドラインからの SAP リソースの作成 / 拡張

以下に、 **LKCLI 経由** でリソースを作成および拡張できるコマンドを示します。
(*SAP 固有のパラメーターは以下の太字で示されています。*)

### lkcli を使用して SAP リソースを作成する

例: `#lkcli resource create sap --tag <tag>` **`--sid <SAP SID> --instance <SAP Instance>`**

オプションパラメーター: `--switchback <switchback type>` **`--protection_level <Protection Level> --recovery_level <Recovery Level> --ip <IP Tag> --dependent_fs <fs, fs, fs> --dependent_cs <SAP central services tag> --automate_fs_creation`**

| オプション | デフォルト | 説明 |
|---|---|---|
| --sid <str> | | SAP インストールの SID |
| --instance <str> | | SAP インスタンス (例: ASCS10) |

| | | |
|---|---|---|
| [--protection_level <FULL\|STANDARD\|BASIC\|MINIMUM>] | FULL | 保護された SAP リソースの Protection Level |
| [--recovery_level <FULL\|REMOTE\|LOCAL\|OFF>] | FULL | 保護された SAP リソースの Recovery Level |
| [--ip <str>] | | LifeKeeper IP リソースタグは、与えられた SAP インスタンスの仮想 IP を保護します。IP リソースは、SAP リソースが作成されるサーバー上で in-service である必要があります。このパラメーターには、IP リソースタグのカンマ区切りのリストが使用できます。この引数で指定された IP リソースは、SAP リソースの子リソースになります。 |
| [--dependent_fs <str>] | | ファイルシステム (gen:filesys) または NFS (gen:nfs) の LifeKeeper リソースタグのカンマ区切りのリストです。(例: /sapmnt、/usr/sap/SID/ASS10) 指定されたすべてのリソースは、SAP リソースが作成されるサーバー上で in-service である必要があります。この引数で指定されたファイルシステムおよび/または NFS リソースは、SAP リソースの子リソースとなります。注意：このパラメーターは、 --automate_fs_ パラメータと併用することはできません。 |
| [--dependent_cs <str>] | | 作成中の SAP リソースの子リソースとして追加されるセントラルサービスリソース (ASCS10 など) の LifeKeeper SAP リソースタグです。 指定されたセントラルサービスリソースは、SAP リソースが作成されているサーバー上でサービス中である必要があります。 注意: このパラメーターは、PAS インスタンスを保護するためのリソースを作成するときに適用される場合があります。 詳細については、SAP リソース階層の作成 の “プライマリーアプリケーションサーバーリソースの作成セクションを参照してください。 |
| [--automate_fs_creation] | | SAP リソースの作成中に、LifeKeeper は既存の SAP ディレクトリ構造をスキャンし、適切な LifeKeeper リソースを自動的に作成して、見つかった共有ファイルシステムのマウントを保護しようとします。 このプロセス中に作成されたリソースは、SAP リソースの子リソースになります。 プロセスが完了したら、結果の階層が正確かどうかを調べます。 注意: このプロセスでは、複製されたファイルシステム階層を自動的に作成することはできません。このパラメーターは、==—==dependent_fs パラメーターと組み合わせて使用することはできません。 |

## lkcli を介して SAP リソースを拡張する

例: `#lkcli resource extend sap --tag <tag> --dest <backupnode>`

オプションパラメーター: `--switchback <switchback type> --template_prioity <priority> --target_priority <prority>`

**注意:** 拡張には SAP 固有のパラメーターはありません。

lkcli コマンドに関する追加情報については、各 ARK の LKCLI (コマンドラインインターフェイス) および LKCLI サブコマンド を参照してください。

# 6.17.7.15. SAP SIOS HA Cluster Connector（SHCC）のアクティブ化

SAP SIOS HA Cluster Connector（SSHCC）は、SAP Start Service（sapstartsrv）とLifeKeeper間のインターフェースを提供します。
SAPインスタンスに対してHA Cluster Connectorがアクティブになっている間、クラスター内のリソースのステータスを最新に保つために、インスタンスの状態に影響するsapcontrolを介した呼び出しはLifeKeeperによってルーティングされます。

SAPインスタンスのSAP SIOS HA Cluster Connectorをアクティブにするには、次の手順を実行します。

1. SAPインスタンスのアクティブなプロファイルの場所を特定します。これは、インスタンスに対応する行を検索することにより、/usr/sap/sapservicesファイルで見つけることができます。たとえば、ASCSインスタンスに対応する次の行を考えます。

```
LD_LIBRARY_PATH=/usr/sap/STC/ASCS00/exe:$LD_LIBRARY_PATH; export
LD_LIBRARY_PATH;
/usr/sap/STC/ASCS00/exe/sapstartsrv pf=/usr/sap/STC/SYS/profile/STC_ASCS00_sap1
-D -u stcadm
```

   この例では、アクティブなASCSインスタンスプロファイルは/usr/sap/STC/SYS/profile/STC_ASCS00_sap1にあります。

2. ステップ1. で見つかったインスタンスプロファイルを編集し、ファイルの下部に次の行を追加します。

```
#SIOS
#-------------------------------------------
service/halib = saphascriptco.so
service/halib_cluster_connector = /opt/LifeKeeper/lkadm/subsys/appsuite/sap/
bin/sap_sios_cluster_connector
service/halib_debug_level = 1
```

3. クラスター内の各システムの/etc/sudoersファイルに次の行を追加して、SAP管理者ユーザーがsap_sios_cluster_connectorスクリプトを実行できるようにします。次の行の<sid>をシステムの小文字のSAP SIDに置き換えます。

```
<sid>adm ALL=NOPASSWD:/opt/LifeKeeper/lkadm/subsys/appsuite/sap/bin/
sap_sios_cluster_connector-main
```

4. プロファイルの変更を反映するには、インスタンスのsapstartsrvプロセスを再起動する必要があります。これは、次のコマンドで実行できます（<sid>を小文字のSAP SIDに、<SID>を大文字のSAP SIDに置き換えます）。

```
su - <sid>adm -c "sapcontrol -nr <Inst#> -function RestartService <SID>"
```

5. SAPインスタンスに対してHA Cluster Connectorが正常にアクティブ化されたことを確認するに

は、次のコマンドを実行します。

```
su – <sid>adm -c "sapcontrol -nr <Inst#> -function HAGetFailoverConfig"
```

- インスタンスに対してHA Cluster Connectorが正常に構成されている場合、このコマンドの出力には、次のサンプル出力のようにHAActive: TRUEと表示されます。

```
HAGetFailoverConfig
OK
HAActive: TRUE
HAProductVersion: "SIOS Protection Suite for Linux" steeleye-lk 9.3.2-6863
HASAPInterfaceVersion: "SIOS Protection Suite for Linux" steeleye-lkHACONNECTO
R-for-SAP 7.5.0-6855
HADocumentation: docs.us.sios.com/Linux/current/LK4L/SAP/index.htm
HAActiveNode: ip-12-0-0-20
HANodes: ip-12-0-0-20, ip-12-0-1-20, ip-12-0-2-20
```

- HA Cluster Connectorが正常に初期化されなかった場合、関連するエラーメッセージは、通常/usr/sap/<

SID>/<INST>/work/sapstartsrv.log内にあるインスタンスのsapstartsrv.logファイルに出力されます。

# 6.17.7.16. SAP テストの準備

1. 仮想 SAP サーバ名を使用して SAP にログインできるよう、SAP GUI を SAP クライアントに設定します。

2. リダンダント AS にログインできるよう、SAP GUI を SAP クライアントに設定します。

3. 必要な場合、追加の AS をクラスタ内の別のサーバにインストールして、PAS を除くすべてのアプリケーションサーバ間にログイングループを設定します。インストールされているすべての AS で、上述したようにプロファイルファイルを変更する必要があります。

# 6.17.7.17. SAP テストの実行

以下の一連のテストを実行してください。テスト手順は、各設定で異なります。一部の手順では、SAP が正常に実行されていることを求めますが、実行すべき具体的なテスト内容は記載されていません。SAP が正常に設定され実行されていることを検証するために行うことのできるテストの一覧については、SAP ドキュメント、 *SAP R/3 in Switchover Environments* の付録を参照してください。

## アクティブ / アクティブ設定のテスト

1. SAP 階層の作成時、SAP と DB は、異なるサーバ上で In Service になります。SAP GUI から、SAP にログインします。ログインが成功し、SAP が正常に実行されていることを確認します。

2. ログアウトし、リダンダント AS からログインします。ログインが成功することを確認します。

3. ログイングループを設定した場合、そのグループを使用してログインが成功することを確認します。

4. LifeKeeper GUI を使用して、SAP バックアップサーバ上で SAP のリソース階層を In Service にします。これで、SAP と DB の両方が同一のサーバ上で In Service になります。

5. 同様に、SAP 仮想サーバ名、リダンダント AS、ログイングループを使用して SAP にログインできることを検証します。SAP が正常に実行されていることを確認します。

6. LifeKeeper GUI を使用して、DB バックアップサーバ上で DB 階層を In Service にします。これで、各階層はバックアップサーバ上で In Service になります。

7. 同様に、すべてのログイン方法を使って SAP にログインできることと、SAP が正常に実行されていることを確認します。トランザクション SM21 を実行すると、ログにおいて PAS が DB との通信を失った記録と再取得した記録を確認できます。

8. SAP にログイン中に、SAP が現在 In Service である SAP バックアップサーバの電源スイッチを押してシャットダウンします。SAP 階層が SAP プライマリサーバ上で In Service になること、そしてフェイルオーバ後、PAS に再度ログインできることと正常に実行されていることを確認します。

9. 切断したサーバの電源をリストアします。LifeKeeper GUI を使用して、DB プライマリ サーバ上で DB 階層をもう一度 In Service にします。同様に、SAP にログイン中に、DB が現在 In Service である DB プライマリサーバの電源スイッチを押してシャットダウンします。DB 階層が DB バックアップサーバ上で In Service になること、そしてフェイルオーバ後に SAP にログインでき、トランザクションを正常に実行できることを確認します。

10. 切断したサーバの電源をリストアします。LifeKeeper GUI を使用して、DB プライマリ サーバ上で DB のリソース階層をもう一度 In Service にします。

## アクティブ / スタンバイ設定のテスト

1. 階層作成時、SAP と DB の両方がプライマリサーバ上で In Service になります。リダンダント AS がバックアップサーバ上で開始されます。SAP GUI から、SAP にログインします。ログインが成功

し、SAP が正常に実行されていることを確認します。トランザクション SM51 を実行し、SAP サーバのリストを表示します。このリストに、PAS または ASCS と AS の両方が含まれている必要があります。

2. ログアウトし、バックアップサーバ上のリダンダント AS を介して再ログインします。ログインが成功することを確認します。

3. ログイングループを設定した場合、そのグループを使用してログインが成功することを確認します。

4. LifeKeeper GUI を使用して、バックアップサーバ上で SAP/DB のリソース階層を In Service にします。

5. 同様に、SAP 仮想サーバ名、リダンダント AS、ログイングループを使用して SAP にログインできることを検証します。SAP が正常に実行されていることを確認します。

6. SAP にログイン中に、階層が現在 In Service になっている SAP/DB バックアップサーバの電源スイッチを押してシャットダウンします。SAP/DB 階層がプライマリサーバ上で In Service になること、そしてフェイルオーバ後、PAS に再度ログインできることと PAS が正常に実行されていることを確認します (サーバがダウンするときに接続が切れるため、再ログインする必要があります)。

7. 切断したサーバの電源をリストアします。同様に、SAP にログイン中に、DB が現在 In Service である SAP/DB プライマリサーバの電源スイッチを押してシャットダウンします。SAP/DB 階層がバックアップサーバ上で In Service になること、そしてフェイルオーバ後に SAP に再度ログインでき、SAP が正常に実行されていることを確認します。

8. 同様に、切断したサーバの電源をリストアします。LifeKeeper GUI を使用して、プライマリサーバ上で SAP/DB のリソース階層を In Service にします。

# 6.17.8. SAP の管理

## 管理のヒント

このセクションでは、特定の設定の管理およびメンテナンスに役立つヒントやその他の情報を提供します。

NFS の考慮事項

クライアントの再接続

LifeKeeper SAP の設定可能パラメータ

保護レベルの更新

リカバリレベルの更新

プロパティの表示

## Oracle データベース

Oracle に関する特別な考慮事項

# 6.17.8.1. NFS の考慮事項

**重要** ：業界のベストプラクティスに従い、v9.6.2 の新規環境、v9.7.0 以降のすべての環境において、SIOS では SAP システムのセットアップ時に、NFS サーバー/クライアントを SAP サービスと同じノードにインストールすること (「ループバック構成」とも呼ばれます) をサポートしません。 代わりに、次のような代替構成を使用してください。
- EFS in AWS
- Azure Premium File Shares with NFS
- Google NFS Share
- NFS as separate cluster

# 6.17.8.2. SAP クライアントの再接続

SAP クライアントは、特定の SAP インスタンスにログオンするよう設定することも、ログオングループにログオンするよう設定することもできます。ログオングループを介してログオンするよう設定した場合、SAP は、クライアントが実行中のどのインスタンスに実際に接続するかを決定します。

クライアントが接続されているインスタンスがダウンした場合、クライアント接続が失われ、クライアントは再ログインする必要があります。データベースが一時的に失われたものの、クライアントが接続しているインスタンスは動作中である場合、データベースが復帰するまでクライアントは一時的に使用不可になりますが、再ログオンする必要はありません。

パフォーマンス上の理由から、クライアントは、PAS、ASCS、または SCS ではなく、リダンダントアプリケーションインスタンスにログオンする必要があります。ただし、管理者は、ログの表示などのために PAS にログオンできます。SAP を LifeKeeper で保護した後、クライアントログインは仮想 SAP サーバ名を使用して設定でき、SAP インスタンスが SAP プライマリサーバまたはバックアップサーバ上でアクティブであるかに関わらずクライアントがログオンできるよう設定できます。

# 6.17.8.3. SAP Recovery Kit のチューニング

SAPスクリプトのいくつかはタイムアウト機能を備えており、スクリプトをハングさせて自動的に自分自身を終了させることができます。この機能は、使用不可のNFS共有による問題に備えて要求されます。これについては、NFSマウントとsu トピックでより詳細に説明します。この機能を備えた各スクリプトは、デフォルトのタイムアウト値 (秒単位) を含んでおり、必要に応じて、その値をオーバーライドできます。タイムアウトの待ち時間を短縮するには、必要な NFSファイルシステムをカンマ区切りリストとして「SAP_NFS_CHECK_DIRS」に追加します。これにより、pingでファイルシステムがエクスポートされていることを確認できます。エクスポートされていない場合はすぐに戻ります。

SAP_CREATE_NAS の有効 / 無効を切り替えできます。SAP_CREATE_NAS のデフォルトは 1 (有効) です。これは、NASマウントされたファイルシステムのNASリソースを自動的に含めるために使用されます。無効にするには、このパラメータを0に設定してください。

チューニング可能な SAP_CONFIG_REFRESH のデフォルトは LKCHECKINTERVAL/2 です。ユーザーは呼び出し間隔の秒数を変更して、LifeKeeper GUIの SAPリソースのプロパティパネルを更新できます。

以下の表は、スクリプト名、変数名、およびデフォルト値を示しています。デフォルト値をオーバーライドするには、 */etc/default/LifeKeeper* ファイルに行を追加し、そのスクリプトに必要な値を設定してください。例えば、removeスクリプトが終了されるまで1分間実行されるようにするには、 */etc/default/LifeKeeper* に次の行を追加します。

```
SAP_REMOVE_TIMEOUT=60
```

> **注意:** スクリプトは、実際には、終了前にタイムアウト値よりもわずかに長く実行されることがあります。

> **注意:** これらの値を変更するとき、LifeKeeperを停止したり再起動する必要はありません。

| スクリプト名 | 変数名 | デフォルト値 |
|---|---|---|
| restore, recover | SAP_START_WAIT | 228秒 |
| remove | SAP_STOP_WAIT | 228秒 |
| remove | SAP_REMOVE_TIMEOUT | 804秒 |
| restore | SAP_RESTORE_TIMEOUT | 516秒 |
| recover | SAP_RECOVER_TIMEOUT | 1320秒 |
| quickCheck | SAP_QUICKCHECK_TIMEOUT | 60秒 |
| create NAS | SAP_CREATE_NAS | 1 (無効にするには 0 に設定) |
| GUI Properties Panel refresh | SAP_CONFIG_REFRESH | LKCHECKINTERVAL の値の 1/2 |

| | | |
|---|---|---|
| NFS shares to check | SAP_NFS_CHECK_DIRS | なし |

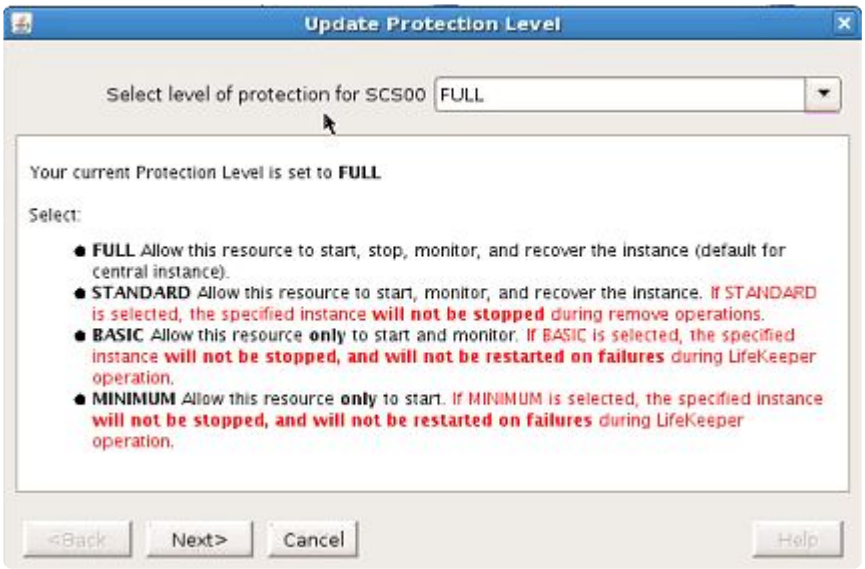
注意: NetWeaver Javaのみの環境では、SCSインスタンスに加えて Java PASを起動するよう設定した場合、SAP_RESTORE_TIMEOUTとSAP_RECOVER_TIMEOUTの値を増加させる必要がある場合があります。

# 6.17.8.4. SAP 保護レベルの更新

保護レベル は、各リソースに実行できるアクションを表します。現在の保護レベルを確認するか、このオプションを変更するには、 **[Update Protection Level]** を使用します。保護のレベルは、 **[FULL]** 、 **[STANDARD]** 、 **[BASIC]** 、または **[MINIMUM]** に設定できます。

1. インスタンス を右クリックします。
2. **[Update Protection Level]** をクリックします。

3. 以下の画面が表示され、保護レベルを選択できます。




**[FULL]**。 デフォルトのレベルです。完全な保護を提供し、インスタンスを起動、停止、監視、および復旧できます。

**[STANDARD]**。このレベルを選択すると、リソースによりインスタンスを起動、監視、および復旧できますが、リソースの停止中は、インスタンスの停止はできません。

**[BASIC]**。このレベルを選択すると、リソースによる起動と監視のみが行えます。障害発生時に停止したり再起動したりすることはできません。

**[MINIMUM]**。 このレベルを選択すると、リソースによる起動のみが行えます。障害発生時に停止し

たり再起動したりすることはできません。

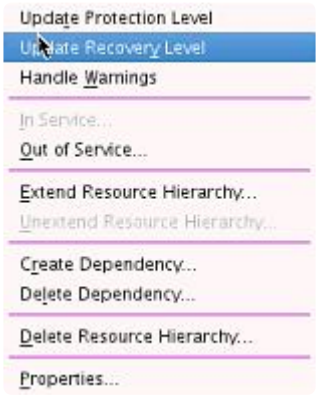
SAP Recovery Kit の ASCS/ERS リソース タイプの設定については、 SAP Protection / Recovery Level レベルの推奨事項 を参照してください。

**注意:** 保護レベル **[BASIC]** と **[MINIMUM]** は、LifeKeeper 保護アプリケーションを一時的なメンテナンスモードにするために使用します。 **[BASIC]** または **[MINIMUM]** をリソースの保護レベルの継続的な状態として使用することは推奨しません。詳細については、「トラブルシューティング」セクションの階層削除時のエラー を参照してください。

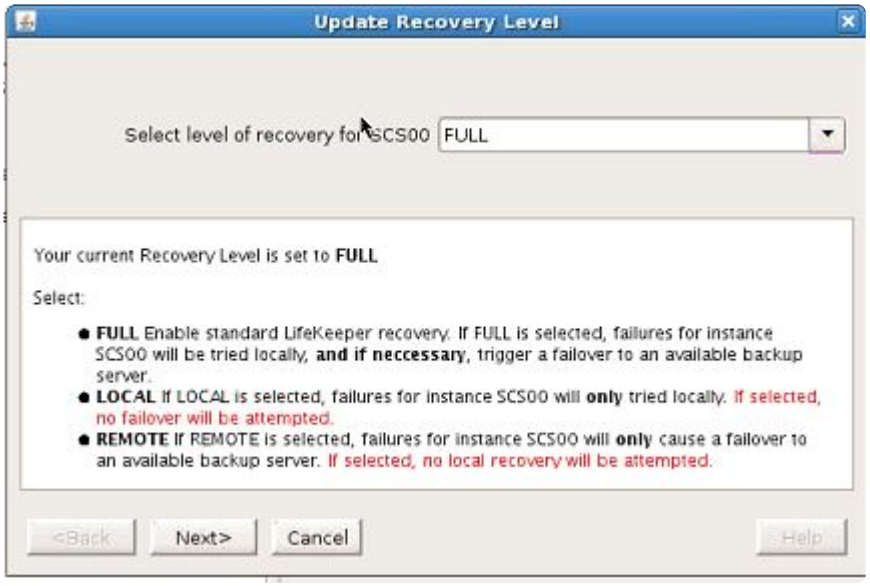
# 6.17.8.5. SAP リカバリレベルの更新

**リカバリレベル** は、障害発生時のリソースに対する指示を提供します。現在のリカバリレベルを確認するか、このオプションを変更するには、 **[Update Recovery Level]** を使用します。リカバリレベルは、 **[FULL]** 、 **[LOCAL]** 、または **[REMOTE]** に設定できます。

1. インスタンス を右クリックします。
2. **[Update Recovery Level]** をクリックします。




3. 以下の画面が表示され、 **リカバリレベル** を選択できます。




**[FULL]**。 リカバリレベルが **[FULL]** に設定されている場合、リソースはローカルリカバリを試みます。それが失敗した場合、対向ノードでの復旧を、成功するまで試みます。

**[LOCAL]**。 リカバリレベルが **[LOCAL]** に設定されている場合、リソースはローカルリカバリのみを試みます。フェイルオーバは行いません。

**[REMOTE]**。 リカバリレベルが **[REMOTE]** に設定されている場合、対向ノードでの起動を試みます。ローカルリカバリを試みることはありません。

SAP Recovery Kit の ASCS/ERS リソース タイプの設定については、 SAP Protection / Recovery Level レベルの推奨事項 を参照してください。

# 6.17.8.6. SAP Protection / Recovery Level レベルの推奨事項

## SAP Recovery Kit ASCS/ERS リソース タイプの設定

| LifeKeeper バージョン | SAP Kit リソースタイプ | ENSA/ERS バージョン | デフォルト Protection Level | デフォルト Recovery Level |
|---|---|---|---|---|
| 9.3.2 以前 | TYPE_CS (1) | ENSAv1 | FULL | REMOTE |
| | TYPE_ERS (2) | ERSv1 | STANDARD | FULL |
| 9.4.0 以降 | TYPE_CS (1) | ENSAv1 | FULL | REMOTE |
| | TYPE_CS (1) | ENSAv2 | FULL | FULL |
| | TYPE_ERS (2) | ERSv1 | STANDARD | FULL |
| | TYPE_NEW_ERS (5) | ERSv1 | FULL | FULL |
| | TYPE_NEW_ERS (5) | ERSv2 | FULL | FULL |

### 注意

1. **SAP Kit** リソースタイプ は、SAPリソース情報フィールドの 7番目のエントリです。 例えば以下のようになります。

```
#/opt/LifeKeeper/bin/ins_list -t SAP-SHC_ASCS10 -f: | cut -d: -f6 | tr '\002'
':' | cut -d: -f7
1
```

```
#/opt/LifeKeeper/bin/ins_list -t SAP-SHC_ERS20 -f: | cut -d: -f6 | tr '\002'
':' | cut -d: -f7
5
```

2. SAP Recovery Kit によって検出された **ENSA/ERS**バージョン は、TYPE_CS および TYPE_NEW_ERS リソースごとに INFO_<tag> ファイルに保存されます。 例えば以下のようになります。

```
# grep SAPENQ_VERSION /opt/LifeKeeper/subsys/appsuite/resources/sap/INFO_SAP-S
HC_ASCS10
SAPENQ_VERSION=2
```

```
# grep SAPENQREP_VERSION /opt/LifeKeeper/subsys/appsuite/resources/sap/INFO_SA
```

```
P-SHC_ERS20
SAPENQREP_VERSION=2
```

3. **TYPE_ERS** リソースタイプは、2ノードの ERSv1 クラスターのみをサポートします。 このタイプのリソースは、SAPリソース階層の最上位に位置し、対応するCSリソースに依存します。様々なERSリソースタイプについては ERS Resource Types in LifeKeeper を参照してください。

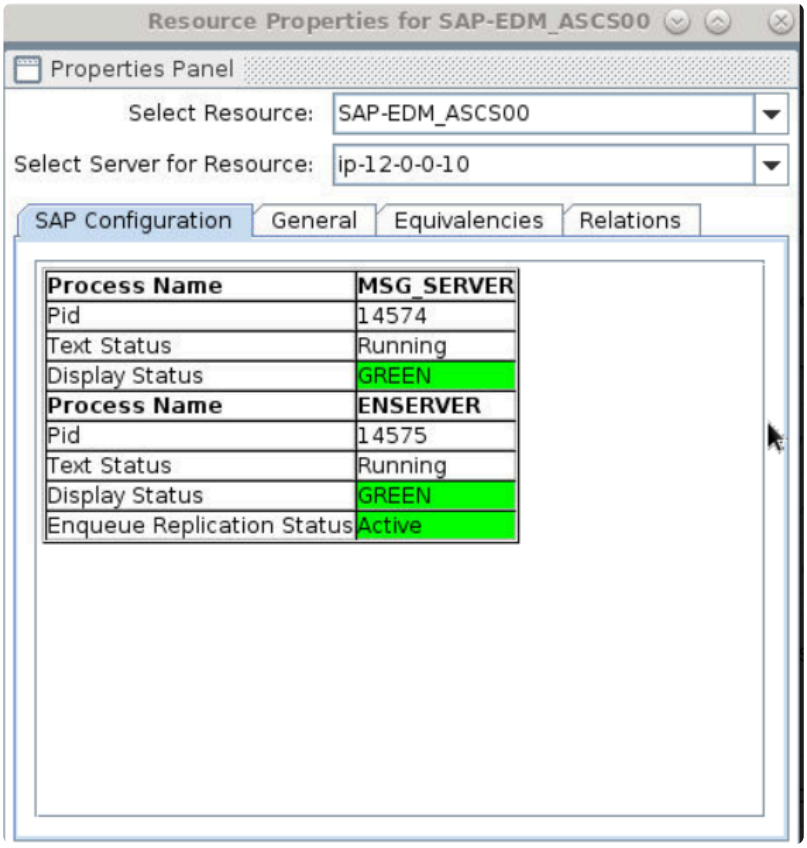
4. **TYPE_NEW_ERS** リソースタイプは、2ノードのERSv1クラスターと、任意の数のノードを含むERSv2クラスターをサポートします。 このタイプのリソースは、対応するCSリソースとは別の独立した階層に存在します。様々な ERS リソースタイプについては LifeKeeperのERSリソースタイプ を参照してください。

# 6.17.8.7. SAP プロパティの表示

**[Resource Properties]** ページでは、特定の SAP リソースの設定の詳細を確認できます。特定のサーバ上のリソースのプロパティを表示するには、または SAP プロセスの状態を表示するには、**[Properties]** 画面を開きます。

1. インスタンス を右クリックします。
2. **[Properties]** を選択します。

3. 以下のような **[Properties]** 画面が表示されます。




表示された **[Properties]** 画面にはタブが 4 つあります。1 番目のタブは [SAP Configuration] で, 特定の SAP リソースの設定情報が設定されています。残り 3 つのタブは、LifeKeeper のすべてのリソースタイプで共通の項目として使用できます。

# 6.17.8.8. Oracle に関する特別な考慮事項

一度 SAP プロセスが LifeKeeper クラスタ内のシステムで機能し始めると、主要な SAP 機能について、LifeKeeper でリソースを作成する必要が生じます。これには、ASCS システム、DVEBMGS システム、SCS システム、および Oracle データベースが含まれます。

このトピックでは、Oracle を LifeKeeper 環境で保護するためのいくつかの特別な考慮事項を説明します。

- LifeKeeper for Linux Oracle Application Recovery Kit がインストールされていることを確認してください。
- Oracle Recovery Kit ドキュメンテーション を参照してください。
- SAP のインストール中、SAPinst プロセスは、通常、データベースソフトウェアが既にインストールされ設定済みであることを前提としています。ただし、SAP で使用するデータベースが Oracle の場合、SAPinst プロセスはインストーラから Oracle インストールツール (RUNINSTALLER) を起動し、Oracle のインストールを完了させます。
- SAP のインストール中に Oracle をインストールすると、Oracle SID が作成されます。この SID は、Oracle Recovery Kit によって要求されます。そのため、Oracle リソースを LifeKeeper で作成するときに提供できるよう準備してください。
- Oracle を使用して 標準的な SAP インストールを作成する場合、Oracle インスタンスが使用する 13 個の個別のファイルシステムが作成されます。一般的に、これらのファイルシステムは、それぞれ LVM 論理ボリュームの最上位階層に構築され、それぞれに多くの別々の物理ボリュームが含まれることがあります。LifeKeeper が正しくこれらのファイルシステムを表現するため、各物理ボリュームと論理ボリュームおよびそれらのボリュームグループに対して個別のリソースが作成されます。この大規模なリソース集合を LifeKeeper 階層にまとめる必要があるため、Oracle 階層の作成と拡張を完了するにはある程度の時間がかかることがあります。作成処理に 1時間かかり、さらに拡張処理に 10～20分かかることは珍しくありません。
- 必要な Oracle (および SAP) ファイルシステムを LVM の最上位階層に構築することは必須ではなく、LifeKeeper 内の SAP と Oracle の Recovery Kit は、標準の Linux ファイルシステムで良好に動作します。
- LifeKeeper Oracle Recovery Kit は、Oracle SAP インストールが標準の Oracle 依存関係として使用する 13 個のファイルシステムのうち、10 個を識別することができ、これらのファイルシステムについて自動的に階層に依存関係を作成します。Oracle Recovery Kit は、 *saptrace* 、*sapreorg*、*saparch* ファイルシステムを自動的に認識しません。LifeKeeper を設定する管理者は、これらの追加のファイルシステムについて、手動で リソース依存関係の作成 を行う必要があります。

# 6.17.8.9. SSHCC HA アクション

SAP SIOS HA Cluster Connector (SSHCC) アクション は、SAP SIOS HA Cluster Connector と併用されるリソースに対して有効なアクションです。 次の SSCC HA Actions は、スタートインスタンス、ストップインスタンス、マイグレートインスタンスのそれぞれにリソースに対して有効です。

- **Start Instance：** Start Instance を選択することで、現在のノード上の特定のリソースタグに対してSAP SIOS HA Cluster Connectorのスタートアクションを実行します。
- **Stop Instance：** Stop Instance を選択することで、現在のノード上の特定のリソースタグに対してSAP SIOS HA Cluster Connectorのストップアクションを実行します。
- **Migrate Instance：** Migrate Instance を選択することで、現在のノード上の特定のリソースタグに対してSAP SIOS HA Cluster Connectorのマイグレートアクションを実行します。
- **Maintenance Mode：** Maintenance Mode を選択することで、すべてのクラスタノード上の指定されたリソースタグに対して、SAP SIOS HAクラスタコネクタのメンテナンスモード操作を実行します。

> 注記 ：これらのアクションは高度な操作設定であり、稼働中、もしくは待機中のLifeKeeperの標準的な操作を置き換えるものとして使用されるものではありません。

この操作を行うには、次の手順を実行してください。

1. インスタンスを右クリックします。
2. 利用可能なメニューから SSHCC HA アクションを選択します。
3. 選択肢から目的の操作を選択します。
4. 指定されたタグに対して選択されたSSHCC HA操作を確認します。
5. [Update] を選択して、選択した操作を続行します。

# 6.17.8.10. LifeKeeper の ERS リソースタイプ

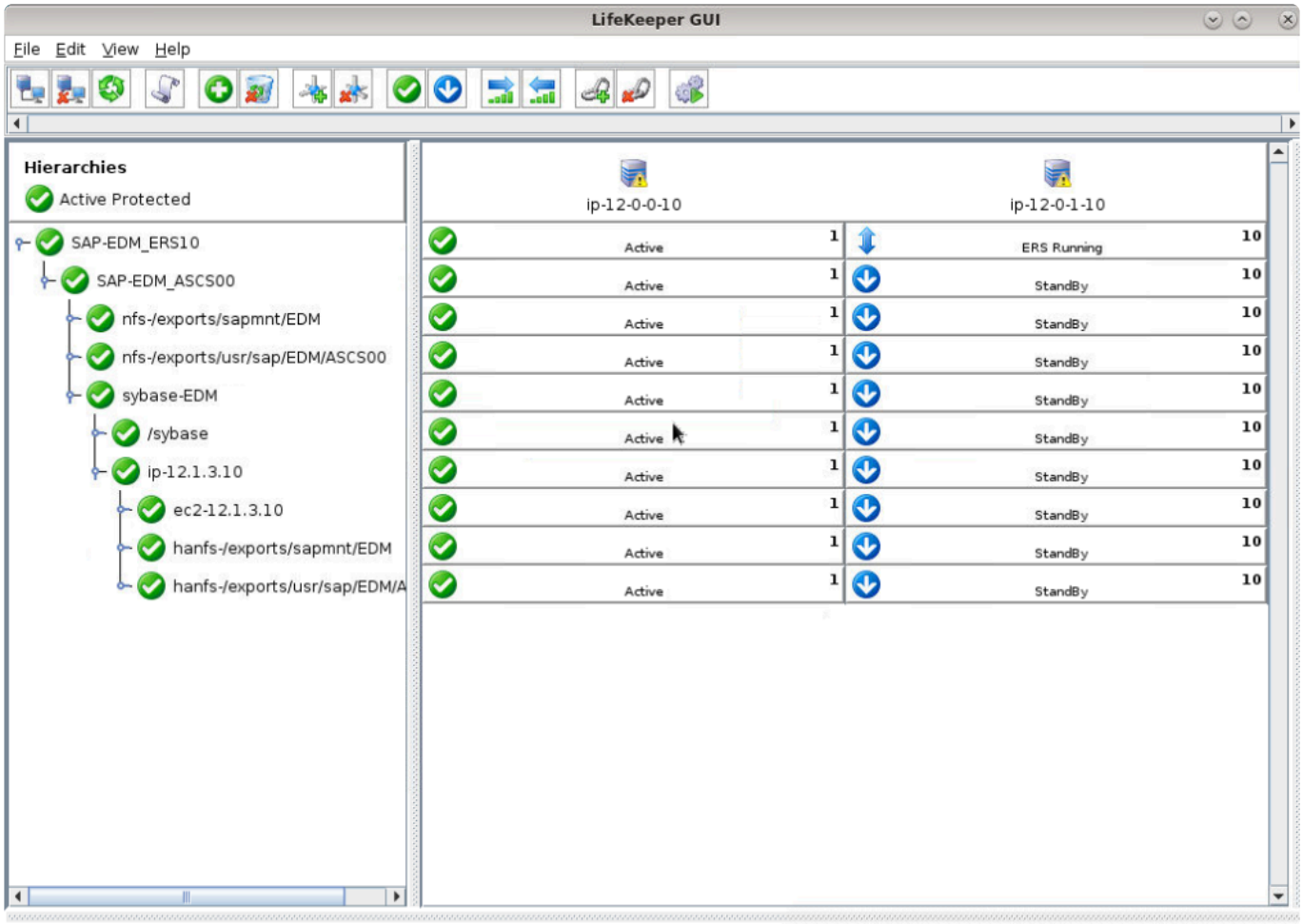
重要: SAP Recovery Kit – 既知の問題と制限 を参照してください。

ERS リソースタイプの設計と実装は、SPS-L 9.4.0で変更されました。このページでは、バージョン9.4.0より前の実装とバージョン9.4.0以降の実装の違い、システムに存在するバージョンの判別方法、および新しいリソースタイプへのアップグレード方法について説明します。

## SPS-L 9.4.0より前（9.3.2以前）のERSリソース

9.4.0より前（9.3.2以前）のバージョンの SPS-L では、ERS リソースは、ロックテーブルの冗長性を提供するセントラルサービス（ASCS/SCS）リソースに依存するSAP階層の最上位に位置するように設計されていました。

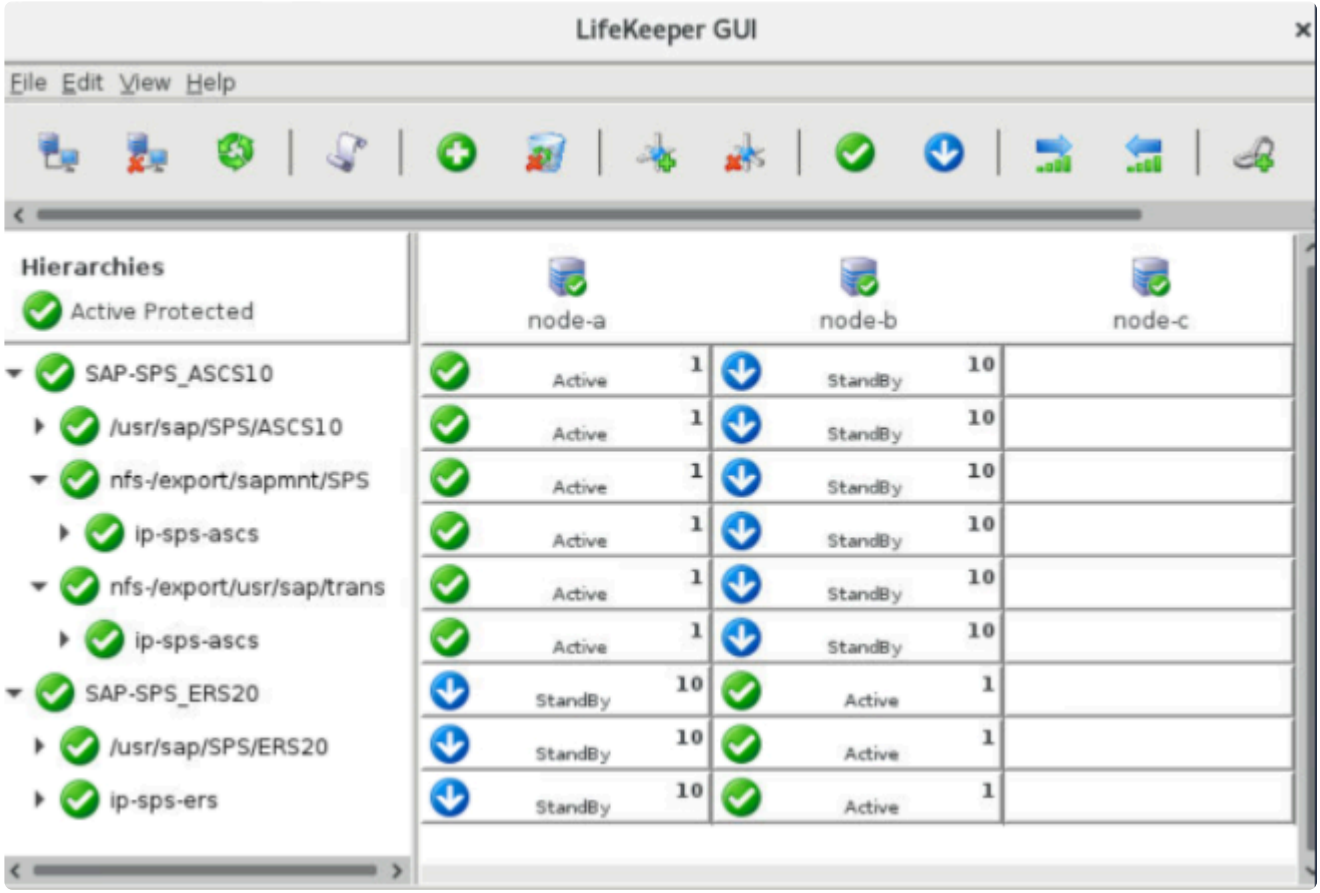
このリソースの動作は、バックアップノード（ERS リソースが LifeKeeper GUI でスタンバイとしてリストされていた場所）で ERS インスタンスを開始するように設計されていました。SAP 階層のスイッチオーバーまたはフェールオーバー時に ASCS/SCS インスタンスがバックアップノードで起動され、そのシステムの共有メモリからロックテーブル（レプリケーションテーブル）のバックアップコピーを取得します。エンキューサーバーがロックテーブルを正常に取得すると、ERS インスタンスに自分自身を終了するよう通知する信号を送信します。
ERS リソースがバックアップノードの LifeKeeper でアクティブ（ISP）になると、ERS インスタンスは、利用可能になったときに元のプライマリノードで起動します。その時点で、レプリケーションサーバーはエンキューサーバーに再接続し、ロックテーブルのレプリケーションを再開します。ERS インスタンスがバックアップノードで実行されている場合、LifeKeeper GUI のステータスは「StandBy」から「ERS Running」に変わります。




## SPS-L 9.4.0 以降の ERS リソース

SPS-L 9.4.0以降では、ERS リソースは独自の独立した階層で動作するように再設計されました。

独立階層の ERSv1 リソース




仮想 IP と高可用性依存ファイルシステムを備えた独立階層の ERSv2 リソース

この新しい ERS リソース設計は、2ノードクラスターの ERSv1 インスタンスと、任意の数のノードを持つクラスターの ERSv2 インスタンスをサポートします。このリソースタイプを使用すると、ERS インスタンスは、LifeKeeper が現在 Active（ISP）になっているノードと同じノードで開始されます。ERSv2 階層（依存する仮想 IP およびファイルシステムリソースを含む）が対応するセントラルサービスリソース階層とは独立してフェールオーバーできるようにするために、設計を変更しました。

対応するセントラルサービスリソースと同じノードで ERS リソースが Active（ISP）にならないようにす

るために、この新しい ERS リソースタイプは、各quickCheck 間隔（デフォルト：2分）で次のことを確認します。

1. ERS リソースが、対応するセントラルサービスリソースと同じノードで Active になっている
2. ロックテーブルのレプリケーションは、エンキューサーバーとレプリケーションサーバーの間で同期している
3. ERS 階層内のすべてのリソースを正常に再配置できるクラスター内に、使用可能な別のノードが存在する

これら3つの条件がすべて満たされている場合、クラスターノード間でエンキューサーバーのロックテーブルデータの冗長性を提供するために、LifeKeeper は ERS 階層を別のクラスターノードに自動的に再配置します。LifeKeeper でフラグ「sap_no_ers_relocation_<ERSタグ>」を設定すると、この自動再配置動作を無効にできます。これは、次のコマンドで実行できます（SAP-EXM_ERS12はERSリソースのタグの例です）。

```
/opt/LifeKeeper/bin/flg_create -f "sap_no_ers_relocation_SAP-EXM_ERS12"
```

このフラグを作成しても、ERSインスタンスのローカルリカバリーの失敗によるフェールオーバーは無効になりません。

**NFS を使用する場合の注意点：** この設計により、sapmnt ファイルシステム上の ERS リソースの LifeKeeper 依存関係が削除されるため、調整可能な値SAP_NFS_CHECK_DIRS を適切に使用して、sapmnt NFS 共有にアクセスできないことに起因するアクションスクリプトのハングを防止する必要があります。詳細については、NFS の考慮事項へのリンク を参照してください。

## 階層に存在する ERS リソースタイプの確認方法

既存の ERS リソースが LifeKeeper のどのバージョンで作成されたかわからない場合は、次のコマンドを実行します（<ERSタグ>をERSリソースのタグ名に置き換えてください）。

```
/opt/LifeKeeper/bin/ins_list -f: -t <ERS Tag> | cut -d: -f6
```

このコマンドの出力は次のようになります。

EXMERS12sap2/sapmntFULLFULL5

- 出力の最後の桁が2の場合、このリソースは SPS-L 9.3.2 以前で作成されています。このタイプのリソースは、2ノードクラスターの ERSv1 インスタンスを表すためにのみ使用でき、SAP階層の最上位に位置し、その下の対応する中央サービス（ASCS/SCS）インスタンスと依存関係を持つ必要があります。独立した階層で動作する新しい設計に切り替える方法については、以下の **ERS リソースタイプのアップグレード** セクションを参照してください。

- 出力の最後の桁が5の場合、このリソースは SPS-L 9.4.0 以降で作成されています。このタイプのリソースを使用して、2ノードクラスターの ERSv1 インスタンスまたは任意の数のノードを持つクラスターの ERSv2 インスタンスを表すことができます。これは、対応するセントラルサービスリソースから独立した階層で動作する必要があります。

## ERS リソースタイプのアップグレード

SPS-L 9.4.0 以降で新しい ERS リソースタイプにアップグレードするには、次の手順を実行してください。

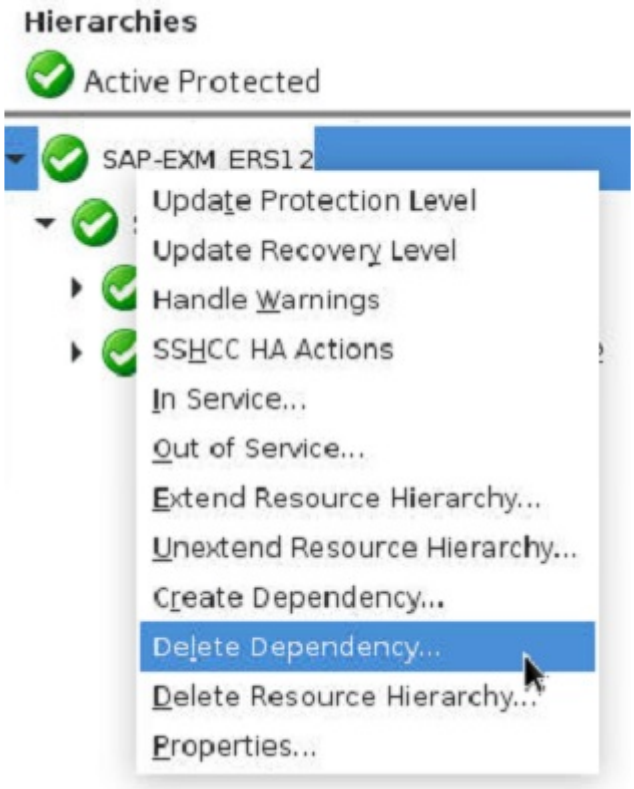
1. アップグレードプロセスを試行する前に、次のコマンドを実行して、すべてのクラスターノードで LifeKeeper 階層のバックアップを作成します。

```
/opt/LifeKeeper/bin/lkbackup -c --cluster
```

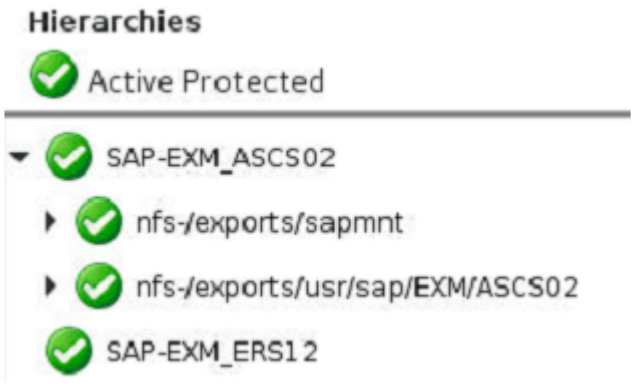
- 作成中に何か間違えた場合は、すべてのノードで LifeKeeper を停止して次のコマンドを実行することにより、保存された階層構成を復元できます。

```
/opt/LifeKeeper/bin/lkbackup -x --cluster
```

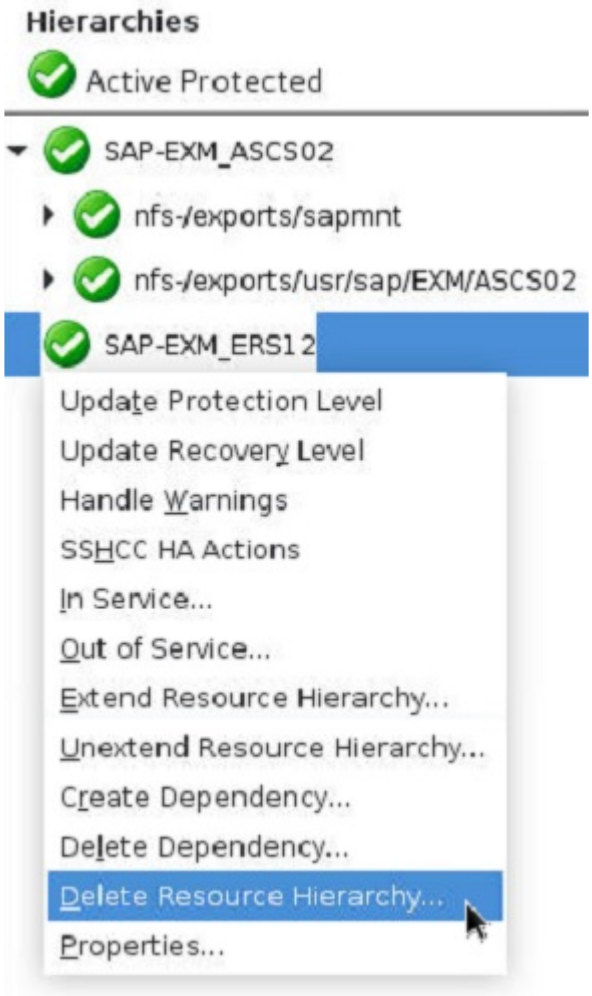
2. LifeKeeper 階層パネルで ERS リソースを右クリックし、 **[Delete Dependency...]** を選択します。




3. ERS リソースのすべての依存関係を削除します。すべての依存関係を削除するには、 **[Delete Dependency...]** を何度か選択する必要がある場合があります。依存関係を削除すると、次の図に示すように、ERS リソースは子の依存関係やそれに依存する他のリソースを持たず、単独の階層に存在するようになります。




4. LifeKeeperの階層パネルで ERS リソースを右クリックし、 **[Delete Resource Hierarchy...]** を選択します。 **ターゲットサーバー** として任意のサーバーを選択し、 **[Next]** をクリックします。 **[Delete]** をクリックし、すべてのノードの ERS リソースを削除します。 **警告:** 前のステップで ERS リソースと SAP 階層内の他のリソース間のすべての依存関係が正常に削除できなかった場合、 このステップによって **SAP 階層全体が削除される可能性があります。**




5. ERS リソースをすべてのノードで正常に削除したら、「SAP のインストール → SAP 階層の作成 」の「ERS リソースの作成」セクションの指示に従って、新しい ERS リソースを作成し、目的のクラスターノードに拡張します。

# 6.17.8.11. ENSAv1 から ENSAv2 へのアップグレード

スタンドアロンエンキューサーバー バージョン1からスタンドアロンエンキューサーバー バージョンバージョン2にアップグレードするには、まず SAP カーネルバージョンが ENSAv2 をサポートしていることを確認してから、次の手順を実行します。

1. デフォルトプロファイル（通常は /usr/sap/<SID>/SYS/profile/DEFAULT.PFL にあります）で以下のパラメーターを設定します。これらのパラメーターは、すべてのインスタンスで同じにする必要があります。

   - enq/enable=TRUE
     enq/serverhost=<
   ASCS instance host>
   enq/serverinst=<ASCS instance number>
   enque/process_location=REMOTESA

2. ASCSインスタンスプロファイル（通常 /usr/sap/<SID>/SYS/profile/<SID>_ASCS<No>_<VIP> にあります）で、次のパラメーターを設定します。

   - _ENQ = enq.sap$(
   SAPSYSTEMNAME)_$(INSTANCE_NAME)
   Execute_01 = local rm -f $(_ENQ)
   Execute_02 = local ln -s -f $(DIR_EXECUTABLE)/enq_server$(FT_EXE)
   Start_Program_01 = local $(_ENQ) pf=$(_PF)

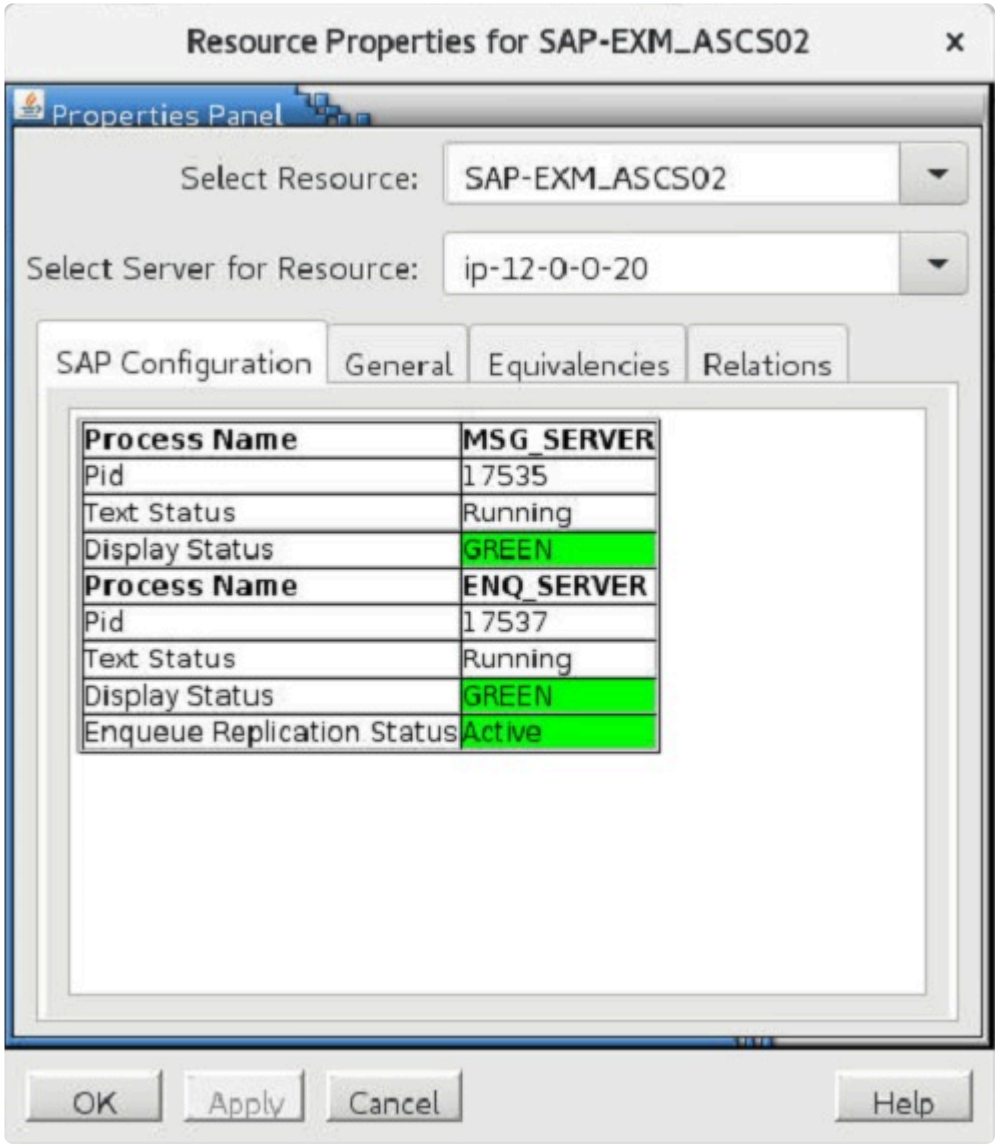
   **注記:** Execute_* および Start_Program_* パラメーターに使用される番号は、このプロファイルでそのパラメーターにまだ使用されていない最初の番号である必要があります。

3. デフォルトおよびASCSインスタンスプロファイルでパラメーターを設定した後、次のコマンドを実行してASCSインスタンスのSAP Start Serviceを再起動します（<sid>を小文字のSAP SIDに、<SID>を大文字のSAP SIDに置き換えます）。

   - su - <sid>adm -c "sapcontrol -nr <
   ASCS Inst#> -function RestartService <SID>"

4. **ASCS**リソースが拡張されているすべてのクラスターノードで、 ファイル/opt/LifeKeeper/subsys/appsuite/resources/sap/INFO_<ASCS Tag> を編集し、SAPENQ_VERSION=2であることを確認します。INFOファイルにSAPENQ_VERSIONの値が含まれていない場合、行「SAPENQ_VERSION=2」を追加する必要があります。

5. SAPシステムを再起動します。ASCSインスタンスが再起動すると、enserver（ENSAv1）プロセスではなくenq_server（ENSAv2）プロセスが使用されます。これは、LifeKeeperでASCSリソースを右クリックし、 **[Properties...]** を選択することで確認できます。




注記： 対応するERSリソースのERSv1からERSv2へのアップグレードが完了するまで、システムのエンキューレプリケーションステータスにはActiveと表示されません。詳細については、ERSv1からERSv2へのアップグレード を参照してください。

# 6.17.8.12. ERSv1 から ERSv2 へのアップグレード

エンキューレプリケーションサーバー バージョン1からエンキューレプリケーションサーバー バージョン2にアップグレードするには、まず SAP カーネルバージョンが ERSv2 をサポートしていることを確認してから、次の手順を実行します。

1. ENSAv1からENSAv2へのアップグレード の指示に従って、ENSAv2を使用するようにASCSインスタンスをアップグレードします。SAPでは混合バージョン構成はサポートしていないため、同じバージョンのスタンドアロンエンキューサーバーおよびエンキューレプリケーションサーバーを使用する必要があります。詳細については、ENSAv2/ERSv2のSAPドキュメントを参照してください。

2. ERSインスタンスの仮想IPと共有ファイルシステムをセットアップします。またLifeKeeperで、対応する仮想IPおよびファイルシステムのリソース階層を作成します。これらは、ERSリソースの再作成中にステップ9で使用します。 **注記：** これらのLifeKeeperリソースは、対応するASCSインスタンスが現在Standby（OSU）であるノード上でActive（ISP）である必要があります。

3. デフォルトプロファイル（通常 /usr/sap/<SID>/SYS/profile/DEFAULT.PFL にあります）で以下のパラメーターを設定します。これらのパラメーターは、すべてのインスタンスで同じにする必要があります。

    - enq/enable=TRUE
      enq/serverhost=<
    ASCS instance host>
    enq/serverinst=<ASCS instance number>
    enq/replicatorhost=<ERS instance host>
    enq/replicatorinst=<ERS instance number>
    enque/process_location=REMOTESA

4. ASCSインスタンスプロファイル（通常 /usr/sap/<SID>/SYS/profile/<SID>_ASCS<No>_<VIP> にあります）で、次のパラメーターを設定します。

    - enq/server/replication/enable = true
      _ENQ = enq.sap$(
    SAPSYSTEMNAME)_$(INSTANCE_NAME)
    Execute_01 = local rm -f $(_ENQ)
    Execute_02 = local ln -s -f $(DIR_EXECUTABLE)/enq_server$(FT_EXE) $(_ENQ)
    Start_Program_01 = local $(_ENQ) pf=$(_PF)

**注記：** Execute_* および Start_Program_* パラメーターに使用される番号は、このプロファイルでそのパラメーターにまだ使用されていない最初の番号である必要があります。

5. ASCSインスタンスプロファイル（通常 /usr/sap/<SID>/SYS/profile/<SID>_ERS<No>_<VIP> にあります）で、次のパラメーターを設定します。

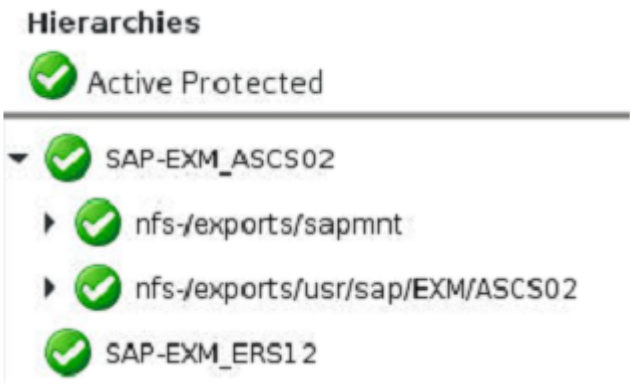
- _ENQR = enqr.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
  Execute_01 = local rm -f $(_ENQR)
  Execute_02 = local ln -s -f $(DIR_EXECUTABLE)/enq_replicator$(FT_EXE) $(_ENQR)
  Start_Program_00 = local $(_ENQR) pf=$(_PF)

注記： Execute_* および Start_Program_* パラメーターに使用される番号は、このプロファイルでそのパラメーターにまだ使用されていない最初の番号である必要があります。

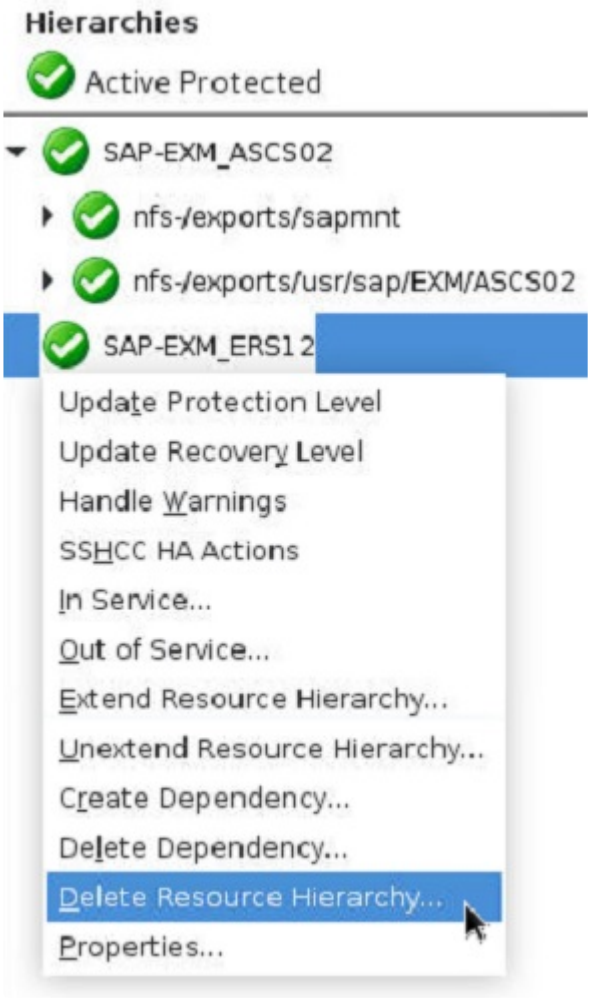
6. デフォルト、ASCS、およびERSインスタンスプロファイルのパラメーターを設定した後、次のコマンドを実行して、ASCSおよびERSインスタンスのSAP Start Serviceを再起動します（<sid>を小文字のSAP SIDに、<SID>を大文字のSAP SIDに置き換えます）。

- su – <sid >adm -c "sapcontrol -host <ASCS VIP> -nr <ASCS Inst#> -function RestartService <SID>"
  su – <sid >adm -c "sapcontrol -host <ERS VIP> -nr <ERS Inst#> -function RestartService <SID>"

7. アップグレードする必要のあるLifeKeeperに既存のERSリソースがない場合は、手順9に進んでください。それ以外の場合は、リソースを右クリックして **[Delete Dependency....]** を選択し、LifeKeeperのERSリソースのすべての依存関係を削除します。すべての依存関係を削除するには、**[Delete Dependency...]** を何度か選択する必要がある場合があります。依存関係を削除すると、次の図に示すように、ERSリソースは子の依存関係やそれに依存する他のリソースを持たず、単独の階層に存在するようになります。

- 

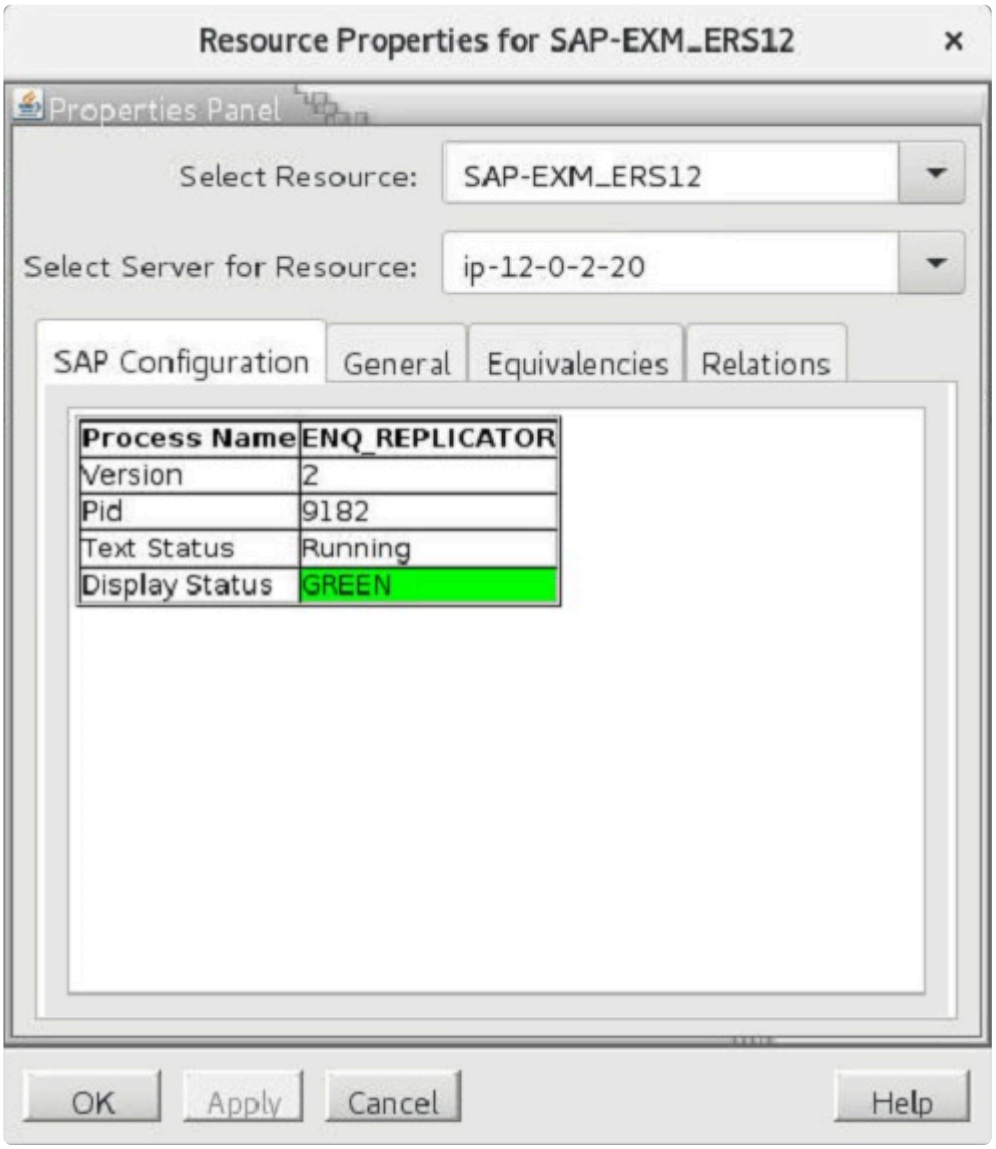

8. LifeKeeperの階層パネルでERSリソースを右クリックし、 **[Delete Resource Hierarchy...]** を選択します。 **ターゲットサーバー** として任意のサーバーを選択し、 **[Next]** をクリックします。 **[Delete]** をクリックして、すべてのノードのERSリソースを削除します。 **警告：** 前のステップでERSリソースとSAP階層内の他のリソース間のすべての依存関係が正常に削除できなかった場合、このステップによって**SAP階層全体が削除される可能性があります。**




9. すべてのノードでERSリソースを正常に削除したら、「 SAPインストール→SAP階層の作成 」の「ERSリソースの作成」セクションの指示に従って、新しいERSリソースを作成し、目的のクラスターノードに拡張します。ERSリソースの作成中に、手順2で作成した仮想IPおよび依存ファイルシステムリソースを選択します（該当する場合）。選択すると、これらのリソースはERS階層に子の依存関係として自動的に追加されます。

10. **ERS**リソースが拡張されたすべてのクラスターノードで、 ファイル/opt/LifeKeeper/subsys/appsuite/resources/sap/INFO_<ERS Tag> を編集し、SAPENQREP_VERSION=2であることを確認します。INFOファイルにSAPENQREP_VERSIONの値が含まれていない場合、行「SAPENQREP_VERSION=2」を追加する必要があります。

11. SAPシステムを再起動します。ERSインスタンスが再起動すると、enrepserver（ERSv1）プロセスではなく、enq_replicator（ERSv2）プロセスが使用されます。これは、LifeKeeperでERSリソースを右クリックし、 **[Properties…]** を選択することで確認できます。



# 6.17.9. SAP トラブルシューティング

本セクションでは、LifeKeeper SAP リソース階層の作成 / 拡張中、およびリソースの削除 / リストア中に表示される可能性のあるメッセージの一覧を示します。また、必要に応じて、エラーの原因や、エラー状態を解決するために必要な処置について説明を付け加えます。

他の LifeKeeper コンポーネントからメッセージが出されることもあります。そのような場合は、メッセージカタログ を参照してください。メッセージカタログには、操作、管理、GUI など、LifeKeeper for Linux を使用しているときに遭遇する可能性がある、すべてのエラーコードが列挙されています。また、エラーコードの原因に関する追加の説明や、問題解決のために必要な処置についても、必要に応じて記載されています。この完全なリストを検索すると、受信したエラーコードを見つけることができます。また、関連する LifeKeeper コンポーネントの個別のメッセージカタログに直接アクセスすることもできます。

このセクションのメッセージは、次のトピックに分類されています。

- ERS インスタンスの変更
- ASCS + ERS Restart_Program パラメーター
- 階層削除時のエラー
- フェイルオーバ中または In-Service 処理中の SAP エラーメッセージ
- SAP インストールエラー
- sapinit のトラブルシューティング
- LifeKeeper ログファイルに出現する「tset」エラー

# 6.17.9.1. ERSプロファイルでAutostartを無効にする

## 症状:

ERS インスタンスのステータスチェックを行うと、選択したインスタンスで sapstart が発生する。

ERS インスタンスは、両方のシステムで常に実行されている。

## 原因:

ERS インスタンスのプロファイルに *Autostart=1* が設定されている場合、特定の `sapcontrol` 呼び出しにより、実行中のコマンドの一部としてインスタンスが起動されます。

## 処置:

クラスタ内で実行中の ERS インスタンスを停止し、ERS インスタンスのプロファイルを変更して *Autostart=0* に設定します。

このプロファイルの変更を有効にするには、sapstartsrvの再起動が必要です。これを行うには、次のコマンドを実行します（を小文字のSAP SIDに置き換え、を大文字のSAP SIDに置き換えてください）。

```
su - <sid>adm -c "sapcontrol -nr <ERS Inst#> -function RestartService <SID>"
```

autostart の値を変更した場合は、変更を反映させるため再起動が必要になります。

# 6.17.9.2. ASCS + ERS Restart_Program パラメーター

## 症状：

エンキューサーバーまたはエンキューレプリケータープロセスが、プライマリーサーバーとバックアップサーバーの両方で同時に実行されている。

## 原因：

ERSインスタンスプロファイルがStart_Programではなく Restart_Programでインスタンスを開始するように設定されている場合、ERSインスタンスが何らかの理由で終了した際、sapstartプロセスはERSインスタンスを自動的に再起動します。
これにより予期しない動作が発生し、フェールオーバーまたはスイッチオーバー中にエンキューサーバーのロックテーブルが失われる可能性があります。

## 処置：

エンキューサーバーおよびエンキューレプリケータープロセスの起動時に、Restart_Programではなく Start_Programを使用するよう、ASCSおよびERSインスタンスプロファイルのパラメーターを変更してください。詳細については、「 ASCSおよびERSインスタンスプロファイル設定の変更 」を参照してください。

ASCSおよびERSプロファイルが更新され、Restart_Programの代わりにStart_Programを使用し、SAP Start Serviceが再起動されたら、システムを再起動し、更新されたプロファイルが読み込まれ、キャッシュが有効になっていないことを確認することを推奨します。

成功した場合、ERSインスタンスは、ASCS/SCSインスタンスが現在実行されていないバックアップシステムでのみ実行する必要があります。

詳細については、 SAPのインストール を参照してください。

# 6.17.9.3. SAP 階層削除時のエラー

## 症状:

使用中のファイルシステムの削除に失敗する。

## 原因:

1. 作成または拡張後に、 **[Resource Protection Level]** が **[Basic]** または **[Minimum]** に設定された場合。リソース保護レベルが Basic または Minimum に設定されている場合、削除処理時に SAP リソース階層が停止されません。これにより、削除が呼び出されても、そのインスタンスのプロセスは実行中のままになります。また、プロセスが保護されたファイルシステムにアクセス中の場合、LifeKeeper はファイルシステムをアンマウントできないことがあります。

2. レプリケートされていない enqueue リソースの **[Resource Protection Level]** が **[Standard]** に設定された場合。リソース保護レベルが Standard に設定されている場合、削除処理時に SAP リソース階層が停止されません。これにより、削除が呼び出されても、そのインスタンスのプロセスは実行中のままになります。また、プロセスが保護されたファイルシステムにアクセス中の場合、LifeKeeper はファイルシステムをアンマウントできないことがあります。

## 処置:

1. **[Basic]** と **[Minimum]** の設定は、リソースを一時的なメンテナンスモードにするときに使用します。継続的な保護レベルとして使用すべきではありません。問題のリソースが Basic または Minimum を継続的な保護レベルとして必要とする場合、そのインスタンスはローカルストレージを使用するように設定するか、リソース階層全体をローカル NFS マウント向けの LifeKeeper NAS Recovery Kit を使用せずに設定する必要があります。

2. **[Standard]** 設定は、レプリケートされた enqueue リソースでのみ使用します。

**注記:** [Standard] は、SPS-L 9.3.2以前で作成され、対応するセントラルサービスリソースに依存するSAP階層の最上位に存在するERSリソースに使用されます。SPS-L 9.4.0以降で作成され、セントラルサービスリソースから独立した階層に存在するERSインスタンスは、保護レベルを [Full] に設定する必要があります。

# 6.17.9.4. SAP 階層のリストアエラー

## 現象:

依存するファイルシステムリソースを含まない SAP リソース階層を以前のホストノードで In-Service に戻そうとすると、SAP リソースの In-Service（リストア）が失敗します。
SAP インスタンスのトレースログに、インスタンスプロセスの1つに必要なポートが既に使用されていることを示す以下のようなメッセージが表示されます。 (この例は、 /usr/sap/<SID>/ASCS<##>/work/dev_ms.new から取得したものです）。

```
[Thr 140225937864512] ***LOG Q0I=> NiIBindSocket: bind (98: Address already in use) [/bas/781_REL/src/base/ni/nixxi.cpp 3946
[Thr 140225937864512] *** ERROR => NiIBindSocket: SiBind failed for hdl 1/sock 6 (SI_EPORT_INUSE/98; I4; ST; 0.0.0.0:3610) [nixxi.cpp 3946]
[Thr 140225937864512] *** ERROR => MsSCommInit: NiBuf2Listen(sapmsSHC) (rc=NIESERV_USED) [msxxserv.c 12838]
[Thr 140225937864512] *** ERROR => MsSInit: MsSCommInit [msxxserv.c 2732]
[Thr 140225937864512] *** ERROR => MsSInit failed, see dev_ms.new for details [msxxserv.c 7363]
```

## 原因:

**リソースの保護レベル** が、作成または拡張後に、 **Basic** 、 **Minimum** 、または **Standard** に設定される。

リソースの保護レベルが Basic、Minimum、または Standard に設定されている場合、SAP リソースを Out-of-Service にする際に、保護されている SAP インスタンスは停止されません。これにより、remove スクリプトが呼び出された後、そのインスタンスのプロセスは実行されたままになります。SAP リソース階層には依存するファイルシステムリソースが含まれていないため、SAP インスタンスが依存するファイルシステムは、SAP 階層がサービスを停止した後もマウントされたままになります。これは、SAP インスタンスプロセスが引き続き実行されることを意味しますが、もし SAP リソース階層に依存するファイルシステムリソースが含まれていた場合は、LifeKeeper が基盤となるファイルシステムをアンマウントするときに `fuser -k` コールで強制終了させます。

SAP インスタンスプロセスが実行されたままになっているサーバー上で LifeKeeper がリソースを In-Service に戻そうとすると、必要なポートが既に実行中のプロセスによって使用されているため、保護されたインスタンスを起動できません。

## 対応:

1. **[Basic]** および **[Minimum]** 設定は、リソースを一時的にメンテナンスモードにする場合にのみ使用してください。これらの設定は、継続的な保護レベルの設定として使用しないでください。

2. **[Standard]** 設定は、 レプリケートされた enqueue リソースにのみ使用できます。特に、SAP リソースに LifeKeeper の保護下にある依存ファイルシステムがない場合は使用しないでください。

> 注意: **[Standard]** は、SPS-L 9.3.2 以前で作成され、対応するセントラルサービスリソースに依存する SAP 階層の最上位に存在する ERS リソースにも使用されます。SPS-L 9.4.0以降で作成され、セントラルサービスリソースから独立した階層に存在する ERS インスタンスは、保護レベルを **[Full]** に設定する必要があります。

3. この問題を解決するには、以下の手順に従ってください。
   a. SAP リソースが拡張されたクラスター内のすべてのサーバーを再起動します。再起動後、保護された SAP インスタンスのすべてのプロセスが停止し、必要なポートが利用可能になります。
   b. サーバーの1つで SAP リソース階層を In-Service にします。LifeKeeper GUI でリソースを右クリックし、[Update Protection Level] を選択して、SAP リソースの保護レベルを **[Full]** に変更します。

1. SAP リソースが拡張されたクラスター内の各サーバーに対して、ステップ（b）を繰り返します。

# 6.17.9.5. フェイルオーバ中または In Service の SAP エラーメッセージ

SAP のフェイルオーバ後、SAP ログにエラーメッセージが記録されます。エラーメッセージの多くは正常であり、無視できます。

## DB 障害発生時

**BVx: Work Process is in reconnect status** – このエラーメッセージは、単に、作業プロセスがデータベースへの接続を失ったため、再接続を試みていることを示しています。

**BVx: Work Process has left reconnect status** – これは実際にはエラーではありませんが、データベースが復帰し、プロセスがデータベースに再接続したことを示しています。

**その他のエラー** – データベースがダウンしている間、いくつものエラーがログに記録される可能性があります。

## CI の起動時

**E15: Buffer SCSA Already Exists** – このエラーメッセージは、実際にはエラーではありません。単に、以前に作成された共有メモリ領域が SAP が使用するシステムに見つかったことを示しています。

**E07: Error 00000 : 3No such process in Module rslgsmcc (071)** – *SAP ノート 7316 を参照* – 前回のシャットダウン中に、ロックが正しく解放されませんでした。このエラーメッセージは無視できます。

## LifeKeeper In Service

In Service の動作時に、LifeKeeper の [In Service] ダイアログに以下のメッセージが表示されることがあります。

```
error: permission denied on key 'net.unix.max_dgram_qlen'

error: permission denied on key 'kernel.cap-bound'
```

これらのエラーは、saposcol が起動されたときに発生するもので、無視できます *(SAP ノート 201144 を参照)*。

# 6.17.9.6. SAP インストールエラー

## tnsnames.ora または listener.ora ファイルの名前の誤り

### 原因:

Oracle データベースを使用する場合で、PAS バックアップサーバのインストールを実行する際に、SAP インストールプログラムにより *tnsnames.ora* または *listener.ora* ファイルのサーバ名が誤っているというエラーが出る場合、Oracle バイナリがローカルファイルシステムにインストールされていない可能性があります。

### 処置:

設定を正しく実行するためには、 */oracle/<SID>/920_<32 または 64>* の Oracle バイナリが各サーバのローカルファイルシステム上にインストールされている必要があります。

# 6.17.9.7. sapinit のトラブルシューティング

## 症状:

init スクリプトにより開始された sapstartsrv プロセスまたは追加の SAP インスタンスプロセスが正しく動作しない、または LifeKeeper スタンバイノードでプロセスを実行しようとする。

## 原因:

SAP は、SAP インスタンスをローカルノードで自動的に起動するために init スクリプトを提供しています。リソースが LifeKeeper の保護に追加されているとき、init スクリプト (`sapinit`) が、現在のノードで実行すべきでない SAP インスタンスプロセスを起動しようとすることがあります。

## 処置:

`sapinit` スクリプトを無効化するか、`sapinit` を変更して LifeKeeper 保護下のインスタンスをスキップするように設定します。この動作を無効化するには、ユーザは `sapinit` を停止する必要があります *(例: `/etc/init.d/sapinit stop`)*。sapinit スクリプトも、chkconfig または同様のツール *(例: `chkconfig sapinit off`)* を使用して無効化する必要があります。

# 6.17.10. SAP メンテナンスモード

## SAP ソフトウェアのアップデート中に LifeKeeper の保護下にあるリソースをメンテナンスモードにする

LifeKeeper には、SAP 階層内のリソースを メンテナンスモード にする機能があります。これにより、ユーザーはバックアップサーバ上で階層を In-service にしなくても、SAP ソフトウェアをアップグレードできます。

### SAP 階層でメンテナンスモードが有効になっている間

- 階層内のすべてのリソースに対して、リソースのヘルスモニタリング、ローカルリカバリ、およびフェイルオーバーが無効になります。

  **注記：** アップグレードプロセス中のデータ破損の可能性を最小限に抑えるために、LifeKeeper は SCSI リザベーションエラーを検出して対処します。これにより、リソース階層のフェイルオーバーが発生したり、SCSI リザベーションが失われた場合にシステムが停止することがあります。SCSI デバイスにアクセスできない場合の LifeKeeper の動作を変更するには、Coreパラメータ一覧 の SCSIERROR パラメータを参照してください。

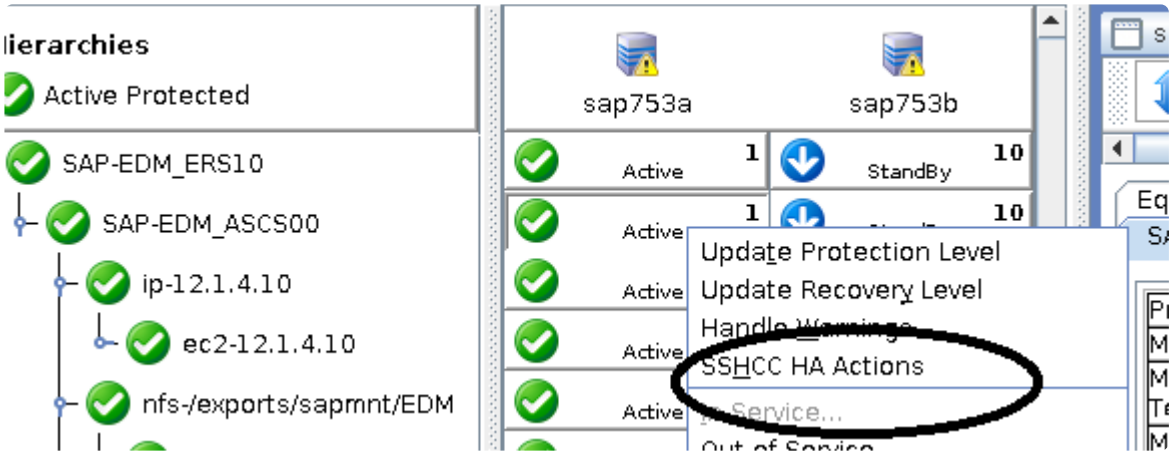
  **注記：** Quorum/Witness 機能を使用している場合、LifeKeeper はデフォルトでは、Quorum を失うとノードを停止します。これは、Quorum の喪失に対処する必要がある重要なアプリケーション階層がある場合に役立ちますが、ネットワーク通信エラーが発生した場合に保守作業を実行している間に予期しない停止を引き起こす可能性があります。手動でQuorum/Witness 機能を無効/有効にする手順は、マニュアルの Quorum / Witness のセクションを参照してください。

- メンテナンスモードが無効になるか、ユーザーが手動でミラーを再開するまで、階層内の DataKeeper ミラーはすべて一時停止されます。

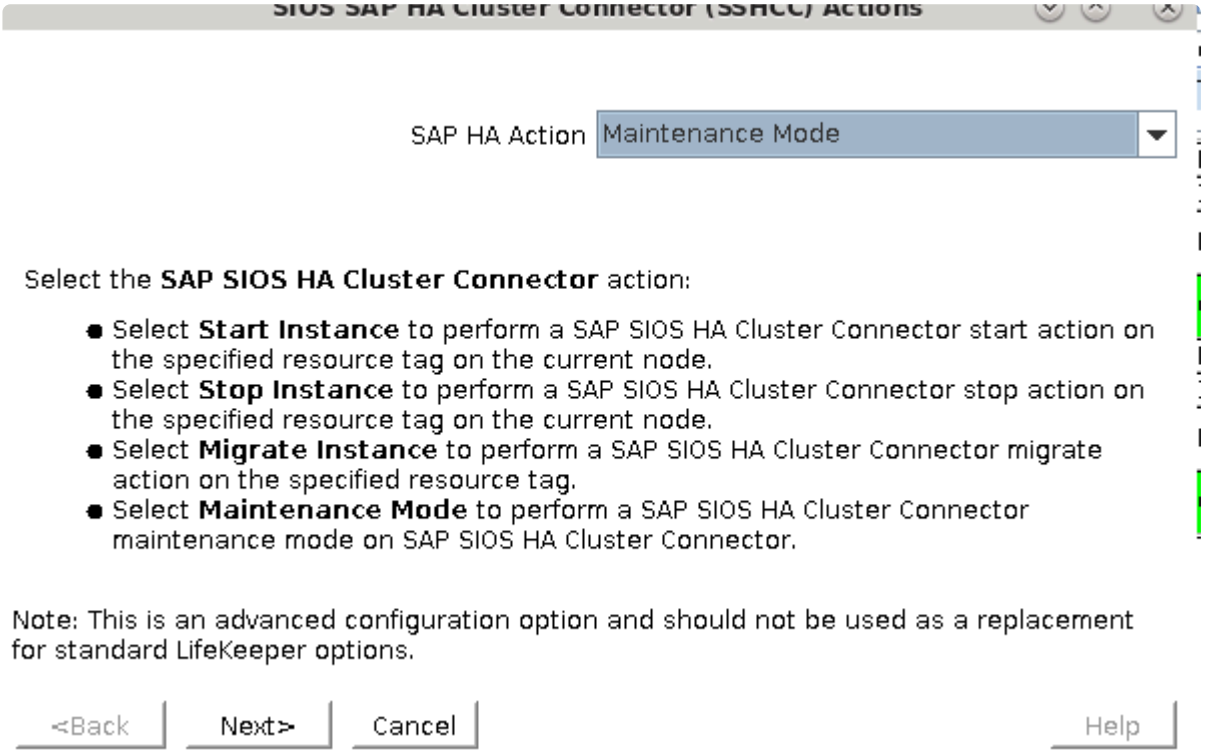
  **注記：** メンテナンスモードが階層に対して有効になっている間にミラーを手動で再開することは推奨しません。これは、ヘルスモニタリング、ローカルリカバリ、およびフェイルオーバーが無効になっているため、LifeKeeper はミラーの障害に対処できないからです。

SAP 階層をメンテナンスモードにするには、以下の手順を実行します。

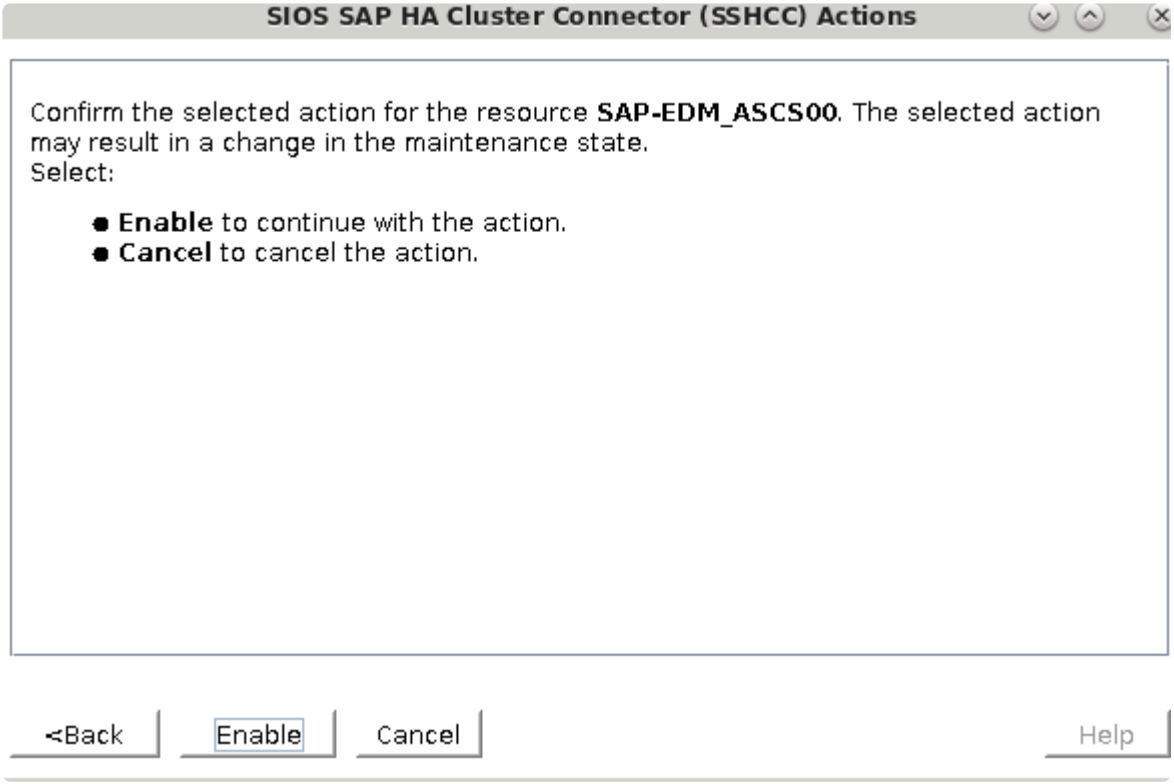
1. **LifeKeeper GUI** を 開きます。
2. 現在In-serviceのノードの各SAPリソースを 右クリック し、 **[SSHCC HA Actions]** をクリックします。




3. ドロップダウンボックスで **[Maintenance Mode]** を選択します。




4. 選択した階層のすべてのリソースに対して、メンテナンスモード を **有効/無効** にするかどうかを選択します。この操作は、クラスタ内のすべてのノードで実行します。




**注意:** メンテナンスモードは階層レベルで有効もしくは無効にされるので、各独立した SAP リソース階層 (例えば A/SCS および ERS 階層)に対して個別に有効もしくは無効にする必要があります。

**注意:** SAPカーネル7.49 PL 200以降では、SAP Update Managerを使用してSAPソフトウェアをメンテナンスモードにすると、対応するLifeKeeper SAP階層は、SAP SIOS HA Cluster ConnectorがSAPインスタンスに対してアクティブである場合、HAコネクタ関数HASetMaintenanceModeを介して自動的にメンテナンスモードになります。LifeKeeperのリソースをLifeKeeper GUI を使用して手動でメンテナンスモードにする場合、アップグレード前にSAP Update Manager のマニュアルを参照して、SAPソフトウェア自体をメンテナンスモードにする方法を決定する必要があります。

## コマンドラインから SAP メンテナンスモードを有効または無効にする

クラスター内の全てのサーバー上の全リソース階層に対して SAP メンテナンスモードを有効するには、LifeKeeper の保護下にある SAP リソースをホストしているサーバーのうち一つで以下のコマンドを実行してください。

```
sudo /opt/LifeKeeper/lkadm/subsys/appsuite/sap/bin/lk_maintenance_mode --mode=enable --cluster
```

オプションの `--tag` パラメーターはメンテナンスモードに設定する特定のリソース階層のみ指定するために使用されます。 例えば (A)SCS および ERS リソース階層のみメンテナンスモードにする場合は、以下のコマンドを LifeKeeper の保護下にある SAP リソースをホストするサーバーのうち一つで実行してくださ

い。

```
sudo /opt/LifeKeeper/lkadm/subsys/appsuite/sap/bin/lk_maintenance_mode --mode=enable --tag=<A/SCS Resource Tag>,<ERS Resource Tag> --cluster
```

クラスター内の全てのサーバー上の全リソース階層に対して SAP メンテナンスモードを無効にするには、以下のコマンドを LifeKeeper の保護下にある SAP リソースをホストするサーバーの一つで実行してください。

```
sudo /opt/LifeKeeper/lkadm/subsys/appsuite/sap/bin/lk_maintenance_mode --mode=disable --cluster
```

上に記載されているように、オプションの —tag パラメーターは、特定のリソース階層のメンテナンスモードを無効にするために使用されます。

```
sudo /opt/LifeKeeper/lkadm/subsys/appsuite/sap/bin/lk_maintenance_mode --mode=disable --tag=<A/SCS Resource Tag>,<ERS Resource Tag< --cluster
```

## LifeKeeper 保護下にあるリソースのメンテナンスモードステータスの確認

LifeKeeper GUI は現在、各リソース階層のメンテナンスモードステータスを表示しません。 LifeKeeper で SAP 階層のメンテナンスモードステータスを確認するには、コマンドラインから次のコマンドを実行します。

```
sudo /opt/LifeKeeper/lkadm/subsys/appsuite/sap/bin/lk_maintenance_mode --mode=check --tag=<Resource Tag> --cluster
```

ここで <Resource Tag> は、SAP リソースの LifeKeeper タグです（例： SAP-SID_ASCS00 または SAP-SID_ERS10）。出力には、指定されたリソースを含む階層のメンテナンスモードが完全に有効になっているか、部分的に有効になっているか、または完全に無効になっているかが表示されます。

# 6.17.11. リソースポリシー管理

注記: LifeKeeper for Linux バージョン 9.3.2 以降では、LifeKeeper UI の SSHCC アクションメニューを介して、SAP 階層の HA メンテナンスモードを有効にできます。詳細については、SAP メンテナンスモード を参照してください。

## 概要

LifeKeeper for Linux のリソースポリシー管理では、リソースのローカルリカバリとフェイルオーバの動作管理機能が提供されます。リソースポリシーは、 **lkpolicy** コマンドラインツール (CLI) を使用して管理できます。

## LifeKeeper

LifeKeeper for Linux には、個々のアプリケーションおよび関連し合うアプリケーションのグループを監視する機能があり、定期的にローカルリカバリを実行したり、保護下のアプリケーションに障害が発生したときに通知することができます。関連し合うアプリケーションの例としては、主アプリケーションが下位のストレージまたはネットワークリソースに依存する階層などがあります。アプリケーションまたはリソースに障害が発生した場合のデフォルトの動作は以下の通りです。

1. **ローカルリカバリ**: 最初に、リソースまたはアプリケーションの **ローカル** でリカバリを試みます。このときは、外部の介入なしにローカルサーバ上でリソースまたはアプリケーションをリストアしようとします。ローカルリカバリが成功した場合、LifeKeeper for Linuxは追加のアクションを実行しません。

2. **フェイルオーバ**: 次に、ローカルリカバリでリソースまたはアプリケーションのリストアに *失敗した* (またはリソースを監視する *リカバリキット* がローカルリカバリをサポートしていない) 場合、 *フェイルオーバ* は開始されます。フェイルオーバ処理では、クラスタ内の別のサーバ上で該当アプリケーション (および依存するすべてのリソース) を起動しようと試みます。

リカバリ動作の詳細については、LifeKeeper for Linux 障害検出とリカバリのシナリオ を参照してください。

## ポリシーによるカスタム動作およびメンテナンスモード動作

LifeKeeper for Linux Version 7.5 以降では、デフォルトのリカバリ動作を変更する追加ポリシーを設定する機能をサポートします。リソース単位 *または* サーバ単位で、4 つのポリシーが設定可能です (リソース単位のポリシーに関する注意については下のセクションを参照してください)。 ***サーバレベルでポリシーを変更する方法を推奨します。***

利用可能なポリシーは以下の通りです。

### 標準ポリシー

- **Failover** - このポリシー設定を使用すると、リソースフェイルオーバを有効 / 無効にできます。(**S**:

リザベーションが適切に処理されるには、 **フェイルオーバ** は、個々の SCSI リソースで無効にすることはできません。)

- **LocalRecovery** - LifeKeeper for Linux は、デフォルトでは、フェイルオーバを実行する前に、個々のリソースまたは保護対象アプリケーション全体を再起動することにより、保護対象リソースのリカバリを試みます。このポリシー設定を使用すると、ローカルリカバリを有効 / 無効にできます。

- **TemporalRecovery** - 通常、LifeKeeper for Linux は、障害リソースのローカルリカバリを実行します。ローカルリカバリに失敗すると、LifeKeeper for Linux は、リソース階層を別ノードにフェイルオーバします。ローカルリカバリに成功した場合は、フェイルオーバは実行されません。

  ローカルリカバリに成功した場合でも、サーバの何らかの異常によって短時間の間にローカルリカバリが再試行される場合があり、結果として何度も連続してローカルリカバリが試行されることになります。これが発生すると、問題のアプリケーションは可用性が悪化します。

  この反復的なローカルリカバリ / 障害サイクルを回避するために、時間的リカバリポリシーを設定できます。時間的リカバリポリシーを使用すると、管理者は指定した時間内に試行するローカルリカバリの回数を (成功かどうかにかかわらず) 制限することができます。

  *例:* リソースが試行するローカルリカバリの回数を 30 分間で 3 回に限定するポリシー定義をユーザが設定した場合、30 分 *以内に* 3 回目のローカルリカバリが試行されると、LifeKeeper for Linux はフェイルオーバを実行します。

  定義した時間的リカバリポリシーは *有効* または *無効* にできます。時間的リカバリポリシーが *無効* の場合、時間的リカバリ処理は継続して実行され、ポリシーが *適用されるはず* の時間に通知がログに表示されますが、実際のアクションは実行されません。

  **注記:** 時間的リカバリポリシーを設定した状態で、フェイルオーバとローカルリカバリの一方または両方を無効にすることは可能です。フェイルオーバまたはローカルリカバリを無効にした場合に、時間的リカバリポリシーは実行されることが **ない** ため、この状態は非論理的です。

### メタポリシー

「メタ」ポリシーは、他の複数のポリシーに影響を与える可能性があるポリシーです。通常、これらのポリシーは、標準ポリシーであれば複数個の設定が必要になるような特定のシステム動作を実現するためのショートカットとして使用します。

- **NotificationOnly** - このモードでは、管理者は LifeKeeper for Linux を「監視専用」状態にすることができます。 ***1 つのリソース (または、サーバ単位のポリシーの場合はすべてのリソース) のローカルリカバリおよびフェイルオーバの両方が影響を受けます。*** 障害が検知されると、ユーザインターフェースには **Failure** 状態が表示されます。 *ただし、リカバリもフェイルオーバも実行されません。* **注記:** 管理者は、障害の原因となった問題を手動で修正し、障害が起きたリソースを復帰させて通常の LifeKeeper for Linux の運用を継続する必要があります。

## リソースレベルのポリシーに関する重要な考慮事項

リソースレベルのポリシーとは、リソース階層全体またはサーバレベルのポリシーとは異なり、特定のリ

ソースにのみ適用されるポリシーです。

*例*:

アプリケーション
- IP
- file system

上記のリソース階層では、アプリケーションは IP とファイルシステムの両方に依存しています。ポリシーは、特定のリソースのローカルリカバリまたはフェイルオーバを無効にするように設定できます。これは、例えば、IP リソースのローカルリカバリが失敗し、IP リソースのフェイルオーバが *無効* に設定されていた場合、IP リソースはフェイルオーバを実行せず、他のリソースのフェイルオーバも発生させないことを意味します。ただし、ファイルシステムリソースのローカルリカバリが失敗し、ファイルシステムリソースのポリシーのフェイルオーバが無効化されていない場合、階層全体がフェイルオーバを実行します。

**注記:** 重要事項として、リソースレベルのポリシーは設定対象の特定のリソースに *のみ* 適用されることに注意してください。

上記は単純な例です。複雑な階層を構成することもできるため、リソースレベルのポリシーを設定するときは注意してください。

# lkpolicy ツール

**`lkpolicy`** ツールは、LifeKeeper for Linux が稼働するサーバのポリシーを管理 (参照、設定、削除) するためのコマンドラインツールです。`lkpolicy` は、ポリシーの設定および修正、ポリシーの削除、利用可能なポリシーと現在の設定値の表示をサポートします。さらに、設定したポリシーは、有効または無効に設定できるため、リカバリ動作に影響を与えながらリソース / サーバ設定を保持できます。

全体的な使用方法は次の通りです。

```
lkpolicy [--list-policies | --get-policies | --set-policy | --remove-policy]
<name value pair data…>
```

`<name value pair data…>`は、運用方法 *および* 対象のポリシーによって異なります (特にポリシーを設定する場合)。 *例*: 有効 / 無効タイプのポリシーのほとんどでは、必要なのは `--on` または `--off` のスイッチのみですが、時間的ポリシーの場合は、しきい値を設定するための値も必要です。

## lkpolicy の使用方法の例

### ローカルおよびリモートサーバとの認証

**`lkpolicy`** ツールは、サーバが公開する API を通じて LifeKeeper for Linux サーバーと通信します。この API は、lkpolicy ツールなどのクライアントに対して認証を要求します。`lkpolicy` ツールで LifeKeeper for Linux サーバーに最初にアクセスしようとしたときに、そのサーバーに対する認証情報がまだ保存されていない場合、ユーザーは認証情報を求められます。認証情報はユーザー名とパスワードの形式であり、

さらに以下の条件があります。

1. クライアントには LifeKeeper for Linux の管理者権限が必要です。したがって、そのユーザ名は、( *PAM* による) オペレーティングシステムの認証設定によって *lkadmin* グループに属する必要があります。必ずしも ***root*** で実行する ***必要はありません*** が、root ユーザはデフォルトで適切なグループに属しているため、root を使用することもできます。

2. 認証情報は *認証情報ストア* に保存されるため、ツールを使用してこのサーバにアクセスするたびに手動で認証情報を入力する必要はありません。

認証情報ストアと credstore ユーティリティによる管理の詳細については、LifeKeeper for Linux の認証情報の設定 を参照してください。

`lkpolicy` によるセッションの例は以下のようになります。

```
[root@thor49 ~]# lkpolicy -l -d v6test4
Please enter your credentials for the system 'v6test4'.
Username: root
Password:
Confirm password:
Failover
LocalRecovery
TemporalRecovery
NotificationOnly
[root@thor49 ~]# lkpolicy -l -d v6test4
Failover
LocalRecovery
TemporalRecovery
NotificationOnly
[root@thor49 ~]#
```

### ポリシーのリスト表示

lkpolicy --list-policy-types

### 現在のポリシーの表示

lkpolicy --get-policies

lkpolicy --get-policies tag=*

lkpolicy --get-policies --verbose tag=mysql* # all resources starting with mysql

lkpolicy --get-policies tag=mytagonly

### ポリシーの設定

```
lkpolicy --set-policy Failover --off

lkpolicy --set-policy Failover --on tag=myresource

lkpolicy --set-policy Failover --on tag=\*

lkpolicy --set-policy LocalRecovery --off tag=myresource

lkpolicy --set-policy NotificationOnly --on

lkpolicy --set-policy TemporalRecovery --on recoverylimit=5 period=15

lkpolicy --set-policy TemporalRecovery --on --force recoverylimit=5 period=10
```

### ポリシーの削除

```
lkpolicy --remove-policy Failover tag=steve
```

*注記: NotificationOnly はポリシーのエイリアスです。NotificationOnly を有効にすることは、対応する LocalRecovery および Failover ポリシーを無効にすることと等価です。*

# 6.17.12. LifeKeeper ログファイルに出現する「tset」エラー

## 原因:

SAP と Database Recovery Kit によって使用された su コマンドにより、LK ログに次のような「tset」エラーメッセージが出力されます。

```
tset: standard error: Invalid argument
```

このエラーは、SAP 管理者およびデータベースユーザのホームディレクトリにあるプロファイルファイルの 1 つによって出力されるもので、非インタラクティブシェルのみに存在します。

## 処置:

データベースユーザと SAP 管理者に C シェルを使用している場合、これらのユーザのホームディレクトリの *.sapenv_<hostname>.csh* に次の行を追加してください。このコードは、「tset」を実行するかどうかを決定する次のコードを囲むように追加する必要があります。

```
if ( $?prompt ) then
    tty -s
    if ( $status == 0) then
        .
        .
        .
    endif
endif
```

注記:「tty -s」から内側の「endif」までのコードは、既にファイル内に存在しています。

データベースユーザと SAP 管理者に Bash シェルを使用している場合、これらのユーザのホームディレクトリの *.sapenv_<hostname>.sh* に次の行を追加してください。

「tset」を実行するかどうかを決定するコードの前に、以下を追加します。

```
case $- in
```

***i*) INTERACTIVE ="yes";;***

```
        *) INTERACTIVE =“no”;;
    esac
```

「tset」を実行するかどうかを決定するコードを囲むように、以下を追加します。

```
            if [ $INTERACTIVE == “yes” ]; then
            tty -s
            if [ $? -eq 0 ]; then
                    .
                    .
                    .
              fi
        fi
```

注記:「tty -s」から内側の「endif」までのコードは、既にファイル内に存在しています。

# 6.18. SAP HANA Recovery Kit 管理ガイド

v9.5.0より、SIOS は新しい SAP HANA Application Recovery Kitをリリースしました。SIOS は **2022年3月31日** まで、9.4.xリリースでのSAP HANA gen/appベースのRecovery Kitのサポートを継続します。LifeKeeper for Linux v9.5以降をご利用の場合は、新しい（ビルトインの）SAP HANA Application Recovery Kitを使用する必要があります。

既存の SAP HANA gen/app ベースの Recovery Kit は v9.5.0 では サポートされていません 。LifeKeeper for Linux v9.5.0へのアップグレードを希望するユーザーは、既存のSAP HANA gen/appベースのRecovery Kitから新しいSAP HANA Recovery Kitに 移行する必要があります 。

SAP HANA は、幅広い用途に展開できるエンタープライズクラスのインメモリデータベースシステムで、SAP S/4HANA ERP プラットフォームの基盤です。
SAP HANA Recovery Kitは、LifeKeeper for Linux における SAP HANA データベースの耐障害性保護を提供します。

**注意：** SAP HANA を 9.7.0 未満からアップグレードする方法については こちら をクリックしてください。

## 本ガイドの構成

このガイドには、SAP HANA 階層の作成および管理に役立つ次のトピックが含まれています。

- SAP HANA サポート構成 ： SAP HANA がサポートする構成をリストします。
- SAP HANA Recovery Kit ハードウェアおよびソフトウェアの要件 SAP HANA Recovery Kitの適切なセットアップ、インストール、操作に必要なハードウェアとソフトウェアをリストしています。
- SAP HANA Recovery Kit 概要 SAP HANA Recovery Kitの機能について説明しています。
- LifeKeeper でのSAP HANA の設定 SAP HANAソフトウェアをインストールおよび構成する手順を提供します。
- SAP HANA リソースの設定タスク LifeKeeper GUIを使用して階層で実行できるさまざまな機能（作成、拡張、削除、および拡張解除）について説明しています。
- SAP HANAリソース階層の管理 SAP HANA階層の継続的な管理に関する重要な推奨事項を提供します。
- SAP HANAトラブルシューティング SAP HANA Recovery Kitに関連するエラーメッセージをリストして説明しています。

## LifeKeeperドキュメント

次の LifeKeeper for Linux 製品ドキュメントは、SIOS Technology Corp. のウェブサイトでご覧いただけます。

- LifeKeeper for Linux リリースノート
- LifeKeeper for Linux テクニカルドキュメンテーション

- LifeKeeper for Linux インストレーションガイド
- アプリケーションリカバリーキット

本ドキュメントは、オプションのLifeKeeper Recovery Kitに関連するドキュメントとともに、SIOS Technology Corp.のウェブサイト http://jpdocs.us.sios.com/ またはLifeKeeper GUIのヘルプメニューで提供されています。

# SAP HANAドキュメント

Documentation for SAP HANAに関するドキュメントは、以下より入手できます。

https://help.sap.com/viewer/ec06fadc50b64b6184f835e4f0e1f52f/4.2.9/ja-JP

# 6.18.1. v9.7.0 未満からのアップグレード

HANA を v9.7.0 より前のバージョンからアップデートすると、すべてのシステムで **インフォフィールド** が更新され、リソースを複数のターゲットに拡張できるようになります。

次のコマンドを使用してインフォフィールドを調べます。
```
/opt/LifeKeeper/bin/ins_list -d <HostName> -t <Tag> -f: | cut -d: -f6 | tr '\002' ':'
```

v9.7.0 より前の LifeKeeper を実行しているシステムでは、出力は次のようになります。
```
SPS:HDB00:sync:SiteA:logreplay
```

アップデートされた LifeKeeper バージョンでは、出力は次のようになります。
```
SPS:HDB00:node2/sync:SiteA:node2/logreplay
```

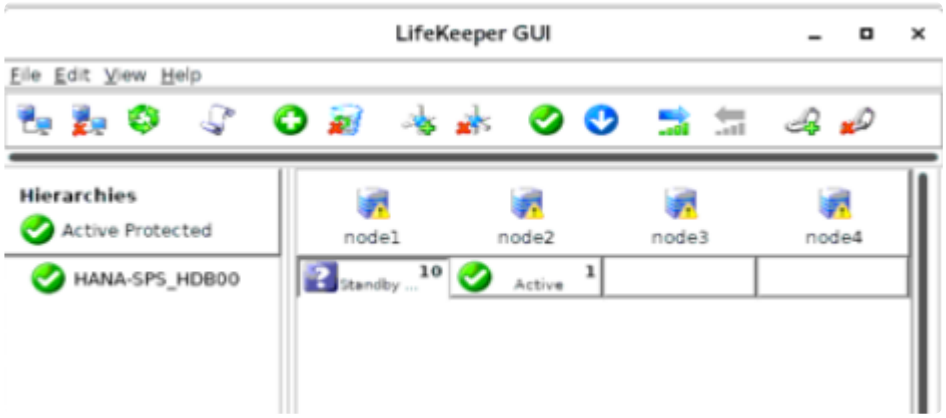
3 ノード クラスタの場合、出力は次のようになります。
```
SPS:HDB00:node2/sync node3/async:SiteA:node2/logreplay node3/logreplay
```

### LifeKeeper は、アップデート後にインフォフィールド構造をどのように自動的に更新しますか?

インフォフィールド構造は、次の場合に各ノードで更新されます。

- LifeKeeper v9.7.0 以降を実行しているシステムで HANA リソースが in-service になると、最初に成功した quickCheck によって、必要に応じて同等の各システムのインフォフィールドが更新されます。
- LifeKeeper v9.7.0 以降を実行しているシステムに HANA リソースを拡張する場合、HANA 拡張スクリプトは、必要に応じて同等の各システムのインフォフィールドを更新します。
- LifeKeeper v9.7.0 以降を実行しているシステムで HANA リソースを out of service にする場合、HANA remove スクリプトは、必要に応じて同等の各システムのインフォフィールドを更新します。
- ノードから HANA リソースの拡張を解除すると、HANA remove スクリプトは、そのノードに関する情報を各同等のリソースインフォフィールドから削除します。

アップグレードされたサーバーで実行されている **LifeKeeper GUI** は、**quickCheck** がスタンバイサーバーのインフォフィールドを更新するまで、スタンバイサーバーの **SAP HANA** リソースを不明なステータスで表示します。



ローリング アップグレード中に障害が発生した場合は、v9.7.0 未満のクラスターのローリングアップグレード中の障害 を参照してください。

# 6.18.2. SAP HANA サポート構成

✻ **注意:** SAP の構築において使用するオペレーティングシステムおよび構成についてはSAP, SIOS およびインフラの提供者 (パブリック/プライベート クラウドまたはオンプレミス)がサポートしている必要があります。SAP HANA の各バージョンにおけるサポート対象のオペレーティングシステムについては、SAP のProduct Availability Matrix (PAM) および SAPノート2235581 (SAP HANA: Supported Operating Systems) を確認してください。

✻ LifeKeeper for Linux v9.7.0 から、SAP HANA Recovery Kit は、最大 4 つのクラスターノード (1 つのプライマリーデータベースサーバーが最大 3 つのセカンダリーサーバーにレプリケート) を持つマルチターゲット HSR 構成をサポートし、DR ロケーションの追加ノードをダウンタイム/災害から保護します。 複雑なスクリプトや面倒な管理タスクは必要ありません。

✻ LifeKeeper v9.7.0 で SAP HANA リソースの構造が変更されたため、v9.7.0 の SAP HANA リソースは以前のバージョンの同等のリソースと互換性がありません。 この互換性の問題は、すべてのクラスターノードが v9.7.0 に正常にアップデートされると、LifeKeeper によって自動的に解決されますが、ローリング アップデート手順の途中でリソース障害が発生すると、SAP HANA リソースが予期しない動作をする可能性があります。このため、v9.7.0 以降にアップデートする前に、SAP HANA リソースをホストする各クラスターノードで lkbackup アーカイブを作成することを**強くお勧めします**。詳細については SAP HANA – 既知の問題と制限 と lkbackup を参照してください。

| パッケージ | バージョン | OS / アプリケーションバージョンのサポート |
|---|---|---|
| LifeKeeper Core (SAP HANA) | 9.7.0 | RHEL 7.6 (HANA 2.0 SPS04 以降, HANA 2.0 SPS05 以下)<br>RHEL 7.7 (HANA 2.0 SPS04 rev 48 以降、HANA 2.0 SPS05 以下)<br>RHEL 7.9 (HANA 2.0 SPS05 のみ、rev 54 以降)<br>RHEL 8.0 (HANA 2.0 SPS04 のみ、rev 40 以降)<br>RHEL 8.1 (HANA 2.0 SPS04 rev 45 以降、HANA 2.0 SPS06 以下)<br>RHEL 8.2 (HANA 2.0 SPS04 rev 48.02 と HANA 2.0 SPS05 rev 52 以降、HANA 2.0 SPS06 以下)<br>RHEL 8.4 (HANA 2.0 SPS05 rev 55 以降)<br>RHEL 8.6 (HANA 2.0 SPS05 rev 59.02、HANA 2.0 SPS06 rev 63 以降)<br>RHEL 9.0 (HANA 2.0 SPS05 rev 59.04 以降、HANA 2.0 SPS06 rev 63 以降)<br>SLES 12 SP4 (HANA 2.0 SPS04 以降、HANA 2.0 SPS05 以下)<br>SLES 12 SP5 (HANA 2.0 SPS04 rev 45 以降、HANA 2.0 SPS06 以下)<br>SLES 15 (GA) (HANA 2.0 SPS04 以降、HANA 2.0 SPS05 以下)<br>SLES 15 SP1 (HANA 2.0 SPS04 rev 44 以降、HANA 2.0 SPS06 以下) |

| | | |
|---|---|---|
| | | SLES 15 SP2 (HANA 2.0 SPS04 rev 48.01 以降、HANA 2.0 SPS06 以下)<br>SLES 15 SP3 (HANA 2.0 SPS05 rev 55 以降)<br>SLES 15 SP4 (HANA 2.0 SPS05 rev 59.02、HANA 2.0 SPS06 rev 63 以降) |
| LifeKeeper SAP HANA Recovery Kit | 9.7.0 | SAP HANA 2.0 SPS04, SPS05, SPS06<br><br>SAP HANA Recovery Kit はSAP HANA 2.0 SPS04 より前のバージョンをサポートしません。 |

**注意:** 拡張SAPサポート用に構築されたオペレーティングシステムのバージョン（Red Hat Enterprise Linux for SAP Business Applications, Red Hat Enterprise Linux for SAP Solutions, Red Hat Enterprise Linux for SAP HANA, SUSE Linux Enterprise Server for SAP Applicationsなど）も、実行中のLinuxカーネルのバージョンが上記のサポート対象OSのバージョンと同じであればサポートされます。

SAP NetWeaver または S/4HANA のアプリケーションサーバーコンポーネントを保護するためにサポートされている構成は LifeKeeper SAP ソリューションページ を参照してください。

# 6.18.3. SAP HANA Recovery Kit ハードウェアおよびソフトウェアの要件

v9.5.0 より、SIOS は新しい SAP HANA Application Recovery Kit をリリースしました。SIOS は **2022年3月31日** まで、9.4.xリリースでの SAP HANA gen/app ベースの Recovery Kit のサポートを継続します。LifeKeeper for Linux v9.5 以降をご利用の場合は、新しい（ビルトインの）SAP HANA Application Recovery Kit を使用する必要があります。

既存の SAP HANA gen/appベースのRecovery Kitはv9.5.0 では サポートされていません。LifeKeeper for Linux v9.5.0 へのアップグレードを希望するユーザーは、既存の SAP HANA gen/appベースのRecovery Kitから新しい SAP HANA Recovery Kit に 移行する必要があります 。

## ハードウェアの要件

- サーバー： サーバーは、SAP HANA マスターガイド、本ガイドで言及している SAP ノート、LifeKeeper for Linuxドキュメント、LifeKeeper for Linuxリリースノートに記載されている要件に従って構成する必要があります。

- ストレージ： SAP HANA システムレプリケーションを使用している SAP HANA データベースの場合、共有ストレージは不要です。特別なストレージ要件は、SAP HANA マスターガイドおよび前述の SAP ノートに記載されています。

## ソフトウェア要件

- **LifeKeeper** ソフトウェア： 必ずクラスター内の各サーバーに同じバージョンの LifeKeeper ソフトウェアをインストールし、同じバージョンの LifeKeeper ソフトウェアパッチを適用してください。

- **SAP HANA** ソフトウェア： SAP HANA Recovery Kit を使用するには、SAP HANA Platform Edition 2.0 SP04 以降が必要です。LifeKeeper および LifeKeeper SAP HANA Recovery Kit を構成する前に、同じバージョンの SAP HANA ソフトウェアを各サーバーにインストールして構成する必要があります。SAP HANA ライセンスはハードウェアに関連付けられているため、各サーバーにはSAPの独自のライセンスが必要です。

  LifeKeeper で SAP HANA リソース階層を作成する前に、SAP HANA システムレプリケーションを有効にして完全に構成し、データベースをすべてのクラスターノードで実行する必要があります。

- **LifeKeeper SAP HANA Recovery Kit**： SAP HANA Recovery Kit は、LifeKeeper for Linux v9.5.0 以降の Core インストールにオプションの Recovery Kit としてバンドルされています。

注意: LifeKeeper for Linux v9.7.0 から、SAP HANA Recovery Kit は、最大 4 つのクラスターノード (1 つのプライマリーデータベースサーバーが最大 3 つのセカンダリーサーバー

にレプリケート) を持つマルチターゲット HSR 構成をサポートしました。

- **Witness サーバー：** Quorum / Witness を使用して、クラスターノード間の通信障害によるスプリットブレインシナリオを防ぐことをお勧めします。 ノード数が偶数のクラスターでは、専用の Witness サーバーとして機能するクラスターノードの追加が必要になる場合があります。

# 6.18.4. SAP HANA Recovery Kit 概要

SAP HANA Recovery Kitは、SAP HANA Platform Edition 2（SP04以降）と互換性があります。

SAP HANAは、可用性を高めるための3つの異なるメカニズムを提供しています。

- ホストの自動フェイルオーバー – SAP HANA システムに追加された少なくとも1つのスタンバイノード。これらのノードは、スタンバイモードで動作するように構成されています。必要なプロセスまたはデータベースがアクティブでない場合、LifeKeeper はそれらの再起動を試みます。
  プライマリノードでプロセスの再起動に失敗した場合、LifeKeeper はバックアップノードでデータベースを In Service にし、そのノードを SAP HANA システムレプリケーションのプライマリマスターとして登録し、以前のプライマリノードをセカンダリレプリケーションサイトとして登録しようとします。以前のプライマリノードをセカンダリ SAP HANA システムレプリケーションサイトとして構成できない場合、問題が修正されて正常に登録できようになるまで、そのノードでリソースは障害（OSF）としてマークされます。以前のプライマリノードがセカンダリレプリケーションサイトとして正常に登録されると、LifeKeeper はノード上の SAP HANA リソースの状態をスタンバイ（OSU）に更新します。

- ストレージレプリケーション – SAP HANA プライマリーノードで使用されるストレージは、すべてのデータを別の SAP HANA ノードに複製します。この複製は、SAP HANA システムからの制御プロセスなしで実行されます。このストレージレプリケーション機能は、ハードウェアパートナーが提供します。

- システムレプリケーション – SAP HANA は、SAP 独自の組み込みレプリケーションフレームワークを使用して、すべてのデータをプライマリー SAP HANA ノードから 1 つまたは複数のバックアップノードにレプリケートします。 データベースに設定されたプリロードオプションによっては、レプリケートされたデータが各セカンダリー SAP HANA ノードのメモリに常にプリロードされる場合があります。

SAP HANA Recovery Kit を使用すると、システムレプリケーションを利用する SAP HANA システムを SIOS LifeKeeper を通じて保護および管理できます。

Recovery Kit は、すべてのノードで SAP HANA システムを起動し、SAP HANA システムレプリケーションのテイクオーバーおよびレプリケーションサイト登録プロセスを実行できます。SAP HANA システムの機能を保証するために、以下のプロセスと状態が継続的に監視されます。

- すべてのノード上の SAP ホストエージェント
- すべてのノード上の HDB インスタンスのSAP Start Service（sapstartsrv）
- すべてのノード上の SAP HANA データベースの状態
- SAP HANA システムレプリケーションモードの状態（アクティブノードでプライマリー、セカンダリーノードで sync|syncmem|async）

必要なプロセスまたはデータベースがアクティブでない場合、LifeKeeper はそれらの再起動を試みます。プライマリーノードでプロセスの再起動に失敗した場合、LifeKeeperはバックアップノードでデータベースを In Service にし、そのノードを SAP HANA システムレプリケーションのプライマリーマスターとして登録して、以前のプライマリーノードをセカンダリーレプリケーションサイトとして登録しようとします。以前のプライマリーノードをセカンダリー SAP HANA システムレプリケーションサイトとして構成で

きない場合、問題が修正されて正常に登録できるまで、そのノードでリソースは障害（OSF）としてマークされます。以前のプライマリノードがセカンダリーレプリケーションサイトとして正常に登録されると、LifeKeeper はノード上の SAP HANA リソースの状態をスタンバイ（OSU）に更新します。

SAP HANA システムレプリケーションの状態が無効の場合、SAP HANA リソースも “Out of Service – Faulty”（OSF）状態になります。テイクオーバーを実行するかどうか、または SAP HANA システムレプリケーションモードの修正方法については、データベース管理者に相談して決定する必要があります。

LifeKeeper で SAP HANA リソースに対して “Out of Service” アクションを実行すると、デフォルトではプライマリーノード上のデータベースのみが停止します。セカンダリーノードのデータベースは引き続きアクティブで、SAP HANA システムレプリケーションモードを保持します。

# 6.18.4.1. SAP HANA GUI ステータス

アクティブ（ISP）リソースには、以下の表に示すステータスがあります。これらの警告および障害のステータスの一部は一時的なものであり、LifeKeeperが必要なプロセスを回復しようとしている間、またはLifeKeeperがHANAリソース階層のフェイルオーバーを開始する前に表示される場合があります。

| GUI のテキスト | GUIプロパティのステータス | 説明 | アイコン |
|---|---|---|---|
| Active | アクティブ | SapstartsrvおよびHDBインスタンスが正常に実行されています。（ステータス名：ISP） | |
| Active – sapstartsrv Failure | Sapstartsrvが起動していない、HDBステータスが不明である | Sapstartsrvが起動していません。sapstartsrvはHDBのステータスを確認するために使用されるため、HDBインスタンスのステータスは不明です。（ステータス名：ISPSapStartSrvNotRunning） | |
| Active – HSR Disabled | HSRはレプリケーションモード「None」で起動している | SapstartsrvおよびHDBは正常に実行されています。レプリケーションモードは「None」で、SAP HANAレプリケーションが無効になっていることを示しています。（ステータス名：ISPHSRDisabled） | |
| Active – HDB Stopped | Sapstartsrvは起動しているが、HDBは正常に起動していない | Sapstartsrvは起動していますが、HDBインスタンスは正常に起動していません。（ステータス名：ISPHDBNotRunning） | |
| Active – Secondary | HSRはセカンダリーレプリケーションモードで実行されている | SapstartsrvとHDBインスタンスは正常に実行されていますが、アクティブノードがセカンダリーレプリケーションサイトとして登録されています。通常の操作では、アクティブノードはプライマリーレプリケーションサイトであることが期待されます。（ステータス名：ISPSecondary） | |
| Active – Unknown Repl Mode | レプリケーションモードを特定できない | Sapstartsrv および HDBインスタンスは正常に実行されています。「`hdbnsutil -sr_state`」を使用してレプリケーションモードを特定できないか、サポートされていないモードを返します。（ステータス名：ISPUnknownReplMode） | |
| Active – Suspended | データベースがプライマリーおよび一時停止している。 | データベースはSAP HANAリソースが現在 In Service になっているサーバーで一時停止しています。これは予期しない状態で、通常、ユーザーが LifeKeeper の外部で “Takeover with handshake” を実行した時に発生します。（ステータス名：ISPSuspended） | |

スタンバイ（OSU）リソースのGUIに表示されるステータスは、アクティブ（ISP）リソースのリストアお

よびquickCheckプロセスの影響を受けます。スタンバイリソースは In Service の操作中に最終的なステータスになるまで数分かかる場合があり、その間に一時的なIntermidiateのステータスが表示される場合がありますが、これはSAP HANAリソースの正常な動作です。以下に示す警告および障害のステータスの一部は一時的なものであり、LifeKeeper が必要なプロセスの回復を試みている間、またはスタンバイサーバーをセカンダリーレプリケーションサイトとして再登録しようとしているときに表示される場合があります。SAP HANAデータベースを正常に起動できない場合、またはバックアップサーバー上のセカンダリーレプリケーションサイトとして登録できない場合は、バックアップサーバー上のリソースのステータスは失敗（Failed：OSF）に移行します。スタンバイリソースのステータスを以下の表に示します。

| GUIのテキスト | GUIプロパティのステータス | 説明 | アイコン |
|---|---|---|---|
| Standby | クラスターHANAリソースISPがない。HDBは実行されていない | これは、同等のリソースが別のノードで In Service でない場合に Out of Service になっているリソースの通常のステータスです。（ステータス名：OSUStopped） | |
| Standby – HDB Running | HDBは実行中、HSRモニタリングは非アクティブ | SapstartsrvとHDBインスタンスはスタンバイ（OSU）ノードで実行されています。アクティブ（ISP）ノードが存在しないか、複数のアクティブ（ISP）ノードがあるか、quickCheckがアクティブノードのHSRステータスを判断していないかのいずれかです。（ステータス名：OSUHDBRunning [1つのアクティブノード] またはOSUSplitBrain [複数のアクティブノード]） | |
| Standby – In Sync | HSRがアクティブで同期されている | HSRが構成され、レポートが同期された状態でSapstartsrvとHDBインスタンスが実行されています。（ステータス名：SecondaryActive） | |
| Standby – sapstartsrv Failure | sapstartsrvが実行されておらず、HDBステータスが不明 | Sapstartsrvが実行されていません。アクティブ（ISP）ノードが存在します。sapstartsrvはHDBのステータス確認に使用されるため、HDBインスタンスのステータスは不明です。（ステータス名：OSUSapStartSrvNotRunning） | |
| Standby – Unknown Repl. Mode | レプリケーションモードを判別できない | SapstartsrvとHDBインスタンスが実行されています。アクティブ（ISP）ノードが存在します。`hdbnsutil -sr_state`を使用してレプリケーションモードを決定できません。（ステータス名：OSUUnknownReplMode） | |
| Standby – HSR Disabled | レプリケーションモードが「None」 | SapstartsrvとHDBインスタンスが実行されています。アクティブ（ISP）ノードが存在します。`hdbnsutil -sr_state`に従ってレプリケーションモードが「none」であるか、HANAユーティリティsystemReplicationStatus.pyが「10」を返しました。（ステータス名：OSUHSRDisabled） | |

| | | | |
|---|---|---|---|
| Standby – HSR Error | アクティブノードHSRがエラー（11）を報告 | SapstartsrvとHDBインスタンスが実行されています。アクティブ（ISP）ノードが存在します。HANAユーティリティsystemReplicationStatus.py は「11」を返しました。（ステータス名：OSUHSRError） | |
| Standby – Initializing | HSRの初期化中 | SapstartsrvとHDBインスタンスが実行されています。アクティブ（ISP）ノードが存在します。HANAユーティリティsystemReplicationStatus.pyは「13」を返しました。（ステータス名：SecondaryInitializing） | |
| Standby – Syncing | HSRの同期中 | SapstartsrvとHDBインスタンスが実行されています。アクティブ（ISP）ノードが存在します。HANAユーティリティsystemReplicationStatus.pyは「14」を返しました。（ステータス名：SecondarySyncing） | |
| Standby – HDB Stopped | クラスター内のHANAリソースISP、HDBが実行されていない | Sapstartsrvが実行されているが、HDBインスタンスは実行されていません。アクティブ（ISP）ノードが存在します。（ステータス名：OSUHDBNotRunning） | |
| Standby – Primary | HSRはレプリケーションモード「Primary」で実行されている | SapstartsrvとHDBインスタンスは正常に実行されていますが、スタンバイノードがプライマリーレプリケーションサイトとして登録されています。通常の運用では、スタンバイノードはセカンダリーレプリケーションサイトであることが期待されます。（ステータス名：OSUPrimary） | |
| Standby – Unknown HSR Status | アクティブノードHSRがエラー（12）を報告 | SapstartsrvとHDBインスタンスが実行されています。アクティブ（ISP）ノードが存在します。HANAユーティリティsystemReplicationStatus.pyは「12」を返しました。（ステータス名：SecondaryUnknownHSRStatus） | |
| Standby – Unexpected HSR state | ISPノードのHSRステータスが予想範囲10～15にない | SapstartsrvとHDBインスタンスが実行されています。アクティブ（ISP）ノードが存在します。HANAユーティリティsystemReplicationStatus.pyは、10～15の予想範囲外の値を返しました。（ステータス名：OSUUnexpectedHSRState） | |
| Standby – Suspended | データベースがプライマリーおよび一時停止している。 | データベースはSAP HANAリソースが現在 Out of Service になっているサーバーで一時停止しています。これは通常、一時停止されていたデータベースが停止され、セカンダリーレプリケーションサイトとして再登録されている間の “Takeover with handshake” プロセス実行中に表示される予期された一時的なステータスとなります。（ステータス名：OSUSuspended） | |

# 6.18.4.2. SAP HANAリソース階層

次の例は、一般的なSAP HANAリソース階層を示しています。

この例での子リソース **vip-sps-hana** は、データベースへのクライアント接続に使用される切り替え可能なIPアドレスを保護しています。

フェイルオーバーが発生した場合、SPSはIPアドレスとデータベースリソースをバックアップサーバーでIn Serviceにします。まだコミットされていないデータベーストランザクションは、再度実行する必要があります。

# 6.18.4.3. マルチテナントデータベースのサポート

SAP HANA Recovery Kit は、マルチテナントデータベース環境でも動作することが可能です。ただし、下記の動作と特記事項についてご注意ください。

- SAP HANA システムレプリケーション (HSR) は、 リカバリーキットの HSR モニタリングを正しく機能させるために、システム上の **全ての** テナントデータベースに対するレプリケーションを正常に初期化できる必要があります。 つまり、例えば HSR を有効にする前に、全てのテナントデータベースのバックアップを取得する必要があるということです。HSR が保護されたインスタンス内に含まれるいずれかのテナントデータベースに対して非同期である場合、SAP HANA Recovery Kit は 、スタンバイシステムに !HANA_DATA_OUT_OF_SYNC フラグを残し、SAP HANA リソース階層のスイッチオーバーもしくはフェイルオーバーを回避します。

- SAP HANA Recovery Kit は現在 個々のテナントデータベースを手動で停止した場合 (つまり、HDBSQL の‘alter system stop database DB_NAME’ コマンドを使用して停止すること。)は、それらを再起動することはありません。この場合、SAP HANA ソフトウェアは HDB インスタンスのプロセスのリストから一致する indexserver プロセスを削除し、 SAP HANA Recovery Kit は、テナントデータベースの手動停止を意図的なものとして扱います。

- SAP HANA Recovery Kit は現在テナントデータベースを個別のレベルで監視していません。 SAP HANA Recovery Kit は(起動している各テナントデータベースのindexserver プロセスを含む) 全部の HDB インスタンスを全体として監視します。 HDB インスタンスのいずれかのプロセスが失敗すると、全てのインスタンスの再起動を試みます。つまり、例えば TENANT2 の indexserver プロセスが失敗するとTENANT1 の indexserver プロセスが予期せず再起動されてしまいます。

# 6.18.5. LifeKeeper での SAP HANA の設定

SAP HANA データベースおよび LifeKeeper ソフトウェアのインストールと構成には、次の手順を実行することをお勧めします。これらの各手順は、後続の詳細なタスクにリンクしています。

SAP HANAソフトウェアのインストール

SAP HANAシステムレプリケーションの構成

SAP HANAインスタンスプロファイルの変更

LifeKeeper ソフトウェアのインストール

これらのタスクを完了すると、LifeKeeper リソース階層を作成して SAP HANA データベースを保護する準備が整います。

# 6.18.5.1. SAP HANA ソフトウェアのインストール

同一のパラメーター/設定を使用して、クラスター内のすべてのサーバーに SAP HANA ソフトウェアをインストールします。特に、すべてのシステムで同じ SAP システムID（SID）とインスタンス番号を使用する必要があります。インストールの詳細については、SAP HANA マスターガイドを参照してください。

# 6.18.5.2. SAP HANA システムレプリケーションの構成

SAP HANA システムレプリケーションガイド（http://help.sap.com で入手可能）の指示に従って、SAP HANA システムレプリケーションを構成します。対象のプライマリーレプリケーションサイトで SAP HANA システムレプリケーションが有効になり、バックアップサーバーがセカンダリーレプリケーションサイトとして登録されたら、本ガイドの次の手順に進んでください。

注意：マルチテナントデータベース環境において、SAP HANA システムレプリケーションは、SAP HANA Recovery Kit を正しく機能させることができるよう全てのテナントデータベースに対するレプリケーションを正常に初期化できる必要があります。特に、システムのレプリケーションを有効にする前に保護対象のインスタンスに含まれる全てのテナントデータベースのバックアップを取得する必要があります。

注意：SAP HANA システムのレプリケーションに使用される IP アドレスのペア間に LifeKeeper のコミュニケーションパスを作成してください。専用ネットワークがシステムレプリケーションに使用される場合、これらの IP アドレスは通常、保護すべき SAP HANA データベースの global.ini ファイルの [system_replication_hostname_resolution] セクションにあります。

# 6.18.5.3. SAP HANA インスタンスプロファイルの変更

## HDB インスタンスの Autostart を無効にする

フェイルオーバーまたはシステムの再起動中に LifeKeeper で SAP HANA リソースを管理するには、システム起動時に HDB インスタンスを自動的に起動するAutostart 機能を無効にする必要があります。この機能を無効にするには、 すべてのクラスターノードで HDB インスタンスのインスタンスプロファイル (通常 /usr/sap/<SID>/SYS/profile/<SID>_HDB<##>_<HostName> に存在) を編集し、プロファイル内に

```
Autostart = 0
```

という記述があるようにします。必要に応じてこの行を追加または変更し、変更をプロファイルに保存してください。

**警告:** HDB インスタンスの Autostart が無効になっていない場合、SAP HANA プライマリシステムレプリケーションサイトのマシン障害により、元のプライマリノードがオンラインに戻った際に「システムレプリケーションのスプリットブレイン」シナリオが発生します。このシナリオでは、HDB インスタンスが実行されており、複数のクラスターノードで同時にプライマリマスターとして登録されています。その結果、LifeKeeper は、ユーザーが SAP HANA プライマリシステムレプリケーションサイトとして登録したいサイトを判断できません。SAP HANA リソースは元のプライマリノードで OSF（「Out of Service – 障害」）状態になり、データベース管理者が状況を手動で解決するまで、開いているすべてのコンソールに警告メッセージが表示されます。詳細については、 スプリットブレインシナリオの解決 を参照してください。

**注記:** SAP HANA リソースの正常な管理を継続するために、LifeKeeper はすべてのクラスターノード上の HDB インスタンスの Autostart パラメーターの値を quickCheck の間隔ごと（LKCHECKINTERVAL パラメーター、デフォルト2分で定義）に監視し、必要に応じて Autostart を自動的に無効にします。

# 6.18.5.4. LifeKeeper ソフトウェアのインストール

SAP HANA ソフトウェアをインストールし、SAP HANA システムレプリケーションの構成が完了したら、LifeKeeper Core ソフトウェアと必要なパッチをインストールし、その後 SAP HANA Recovery Kit をインストールできます。

LifeKeeper パッケージのインストールの詳細については、 LifeKeeper for Linux インストレーションガイド を参照してください。

# 6.18.6. SAP HANAリソースの設定タスク

前のセクションで説明したセットアップタスクを完了すると、SAP HANAリソース階層を作成して拡張できるようになります。

SIOS Protection Suite for Linux SAP HANA Recovery Kitを設定するには、次のタスクを実施します。

- **リソース階層の作成** – SAP HANAリソース階層を作成します。
- **リソース階層の削除** – SAP HANAリソース階層を削除します。
- **リソース階層の拡張** – SAP HANAリソース階層をプライマリサーバーからバックアップサーバーに拡張します。
- **リソース階層の拡張解除** – SPSクラスター内の単一サーバーからSAP HANAリソース階層の拡張を解除（削除）します。

LifeKeeperリソース階層を設定するための追加の手順については、 SPS for Linuxテクニカルドキュメンテーション を参照してください。

以下のタスクは、すべてのRecovery Kitで手順が同じ一般的なタスクなため、SPS for Linuxテクニカルドキュメント内の 管理 セクションの説明を参照してください。

- リソース依存関係の作成 既存のリソース階層と別のリソースインスタンスの間に親子の依存関係を作成し、依存関係の変更をクラスター内の該当するすべてのサーバーに通知します。
- リソース依存関係の削除 リソースの依存関係を削除し、依存関係の変更をクラスター内のすべての該当するサーバーに通知します。
- In Service 特定のサーバーでリソース階層をIn Serviceにします。
- Out of Service 特定のサーバーでリソース階層をOut of Serviceにします。
- プロパティの 表示 / 編集 特定のサーバー上のリソース階層のプロパティを表示または編集します。

**注記：** 本セクションの残りの部分では、 **Edit** メニューを使用して設定タスクを実施します。

また、ほとんどのタスクは以下によって実行できます。

- ツールバーから実行する
- ステータス表示の左ペインでグローバルリソースを右クリックする
- ステータス表示の右ペインでリソースを右クリックする

右クリック方式を使用すると、 **Edit** メニューの使用時に必要な特定の情報の入力をせずに済みます。

# 6.18.6.1. SAP HANA リソース階層の作成

> **重要：** SAP HANA リソース階層を作成する前に、SAP HANA システムレプリケーションを有効にして完全に構成し、データベースをクラスター内のすべてのサーバーで実行する必要があります。 詳細については、 SAP HANA システムレプリケーションの構成 をご覧ください。

プライマリサーバーで次の手順を実行します。

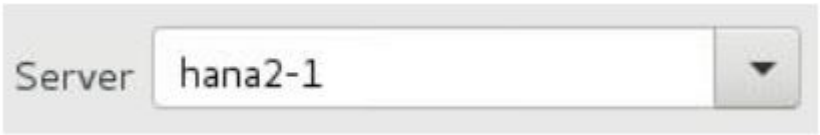
1. **Edit** メニューから、 **[Server]** 、 **[Create Resource Hierarchy]** を選択します。 ***Create Resource Wizard*** のダイアログが表示されます。

2. ドロップダウンリストから **SAP HANA** を選択し、 **[Next]** をクリックします。

3. SAP HANA リソースに使用する **スイッチバックタイプ** として、 *intelligent* を選択します。 **[Next]** をクリックします。

Intelligent スイッチバックとは、バックアップサーバーへのフェイルオーバー後に、管理者が手動で SAP HANA リソースをプライマリーサーバーに戻す必要があることを意味します。 注意: このバージョンの LifeKeeper は、SAP HANA リソースの Automatic スイッチバックをサポートしていません。 さらに、Automatic スイッチバックの制限は、SAP HANA リソースを含む階層内の他の LifeKeeper リソースにも適用されます。 このスイッチバック機能は、SAP HANA リソースを含む階層を切り替えません。

4. SAP HANA リソースを作成する **サーバー** を選択します。最初の SAP HANA リソースを作成する場合、このステップで選択するサーバーは、そのサーバー上でデータベースが現在 SAP HANA システムレプリケーションのプライマリマスターとして登録されているサーバーでなければなりません。 **[Next]** をクリックします。

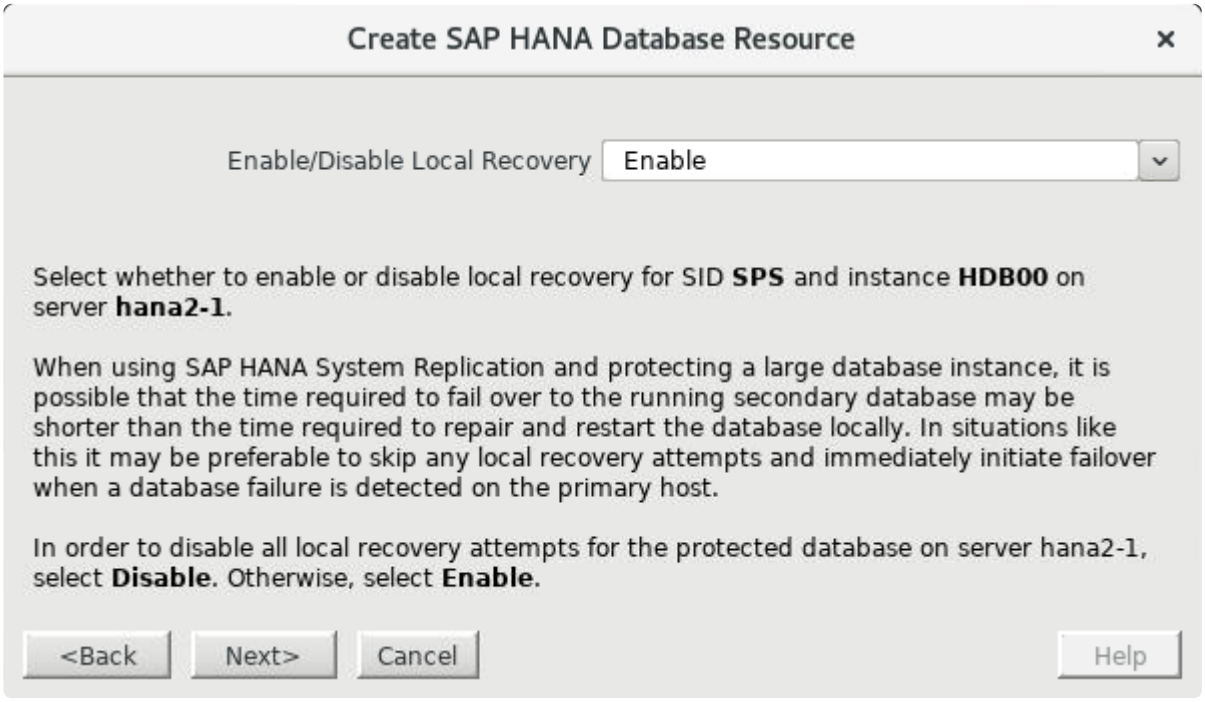



5. SAP HANAデータベースがインストールされている **SAP HANA SID** を選択し、 **[Next]** をクリックします。

6. LifeKeeperで保護する **SAP HANA** インスタンスを選択し、 **[Next]** をクリックします。

7. LifeKeeperで保護する、 **依存する仮想IPリソース** を選択します。 リストに表示するには、IPリソースはすでに存在し、選択したサーバーで In Service になっている必要があります。LifeKeeper IP リソースを使用せずにフェイルオーバーまたはスイッチオーバーで仮想 IP / ホスト名の切り替えを管理する場合は、「*none*」を選択します。 **[Next]** をクリックします。

8. 指定したサーバーで SAP HANA リソースに対するローカルリカバリーを **Enable** または **Disable** に設定します。SAP HANA システムレプリケーションを使用している状況において、特に大規模のデータベースを保護している場合、プライマリーデータベースで障害を検知後、現在のプライマリーホスト上でデータベースを停止し再起動するよりは、起動しているスタンバイデータベースインスタンスへ切り替えたほうが早い場合があります。全てのローカルリカバリーの試行を無効にする場合は **Disable** を選択し、そうでない場合は **Enable** を選択してください。



9. 選択したサーバー上に作成される SAP HANA リソースを識別するために使用される **SAP HANA リソースタグ** を入力します。 デフォルトのタグ名の形式は HANA-<SID>_<HDB インスタンス> ですが、現在別の LifeKeeper リソースの識別に使用されていないものであれば、別のタグ名を使用するように変更できます。タグ名で使用できる文字は、文字、数字、および次の特殊文字「-」、「_」、「.」、「/」です。

10. **[Create]** をクリックします。 ***Create Resource Wizard*** でSAP HANAリソース階層を作成します。LifeKeeper は、入力されたデータと SAP HANA システムレプリケーションフレームワークから取得したデータを検証します。問題が検出されると、情報ボックスにエラーメッセージが表示されます。 **[Next]** をクリックします。

11. SAP HANA リソース階層が正常に作成されたことを示すメッセージが表示され、その階層をクラスター内の別のサーバーに拡張してフェイルオーバー保護を実現する必要があります。 **[Next]** をクリックします。

12. **[Continue]** をクリックします。その後、LifeKeeper は ***Pre-Extend Wizard*** を起動します。 リソース階層を別のサーバーに拡張する方法の詳細については、 SAP HANA リソース階層の拡張 を参照してください。

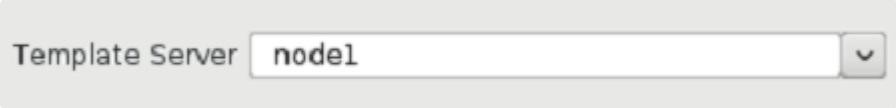
# 6.18.6.2. SAP HANA リソース階層の拡張

> **重要：** SAP HANA リソース階層を拡張する前に、SAP HANA システムレプリケーションを有効にして完全に構成し、データベースをクラスター内のすべてのサーバーで実行する必要があります。詳細については、 SAP HANA システムレプリケーションの構成 をご覧ください。

この操作は、 **[Edit]** メニューから実行するか、 **リソース階層の作成** オプションの完了時に自動的に開始できます。この場合、以下の手順2から開始する必要があります。

1. **[Edit]** メニューから、 **[Resource]** 、 **[Extend Resource Hierarchy]** の順に選択します。 ***Pre-Extend Wizard*** が表示されます。拡張操作に慣れていない場合は、 **[Next]** をクリックします。 LifeKeeper のリソース階層拡張のデフォルト値に精通しており、入力/確認のプロンプトをスキップしたい場合は、 **[Accept Defaults]** をクリックしてください。

2. 既存のSAP HANAリソースを拡張する **テンプレートサーバー** を選択します（ *右ペインのリソースインスタンスを右クリックして **Extend** （タスクの拡張）を選択した場合、このダイアログボックスは表示されません* ）。 **[Next]** をクリックします。




3. 拡張したいSAP HANAリソースの **拡張するタグ** を選択します（ *いずれかのペインでリソースインスタンスを右クリックして **Extend** （タスクの拡張）を選択した場合、このダイアログボックスは表示されません* ）。 **[Next]** をクリックします。

4. SAP HANAリソースに使用する **スイッチバックタイプ** に *intelligent* を選択し、 **[Next]** をクリックします。

インテリジェントスイッチバックとは、バックアップサーバーへのフェイルオーバー後、管理者がSAP HANAリソースをプライマリーサーバーに手動で切り替える必要があることを意味します。 注意： LifeKeeper のこのリリースは、SAP HANAリソースの自動スイッチバックをサポートしていません。さらに、自動スイッチバックの制限事項は、SAP HANAリソースに依存する他のLifeKeeperリソースにも適用されます。

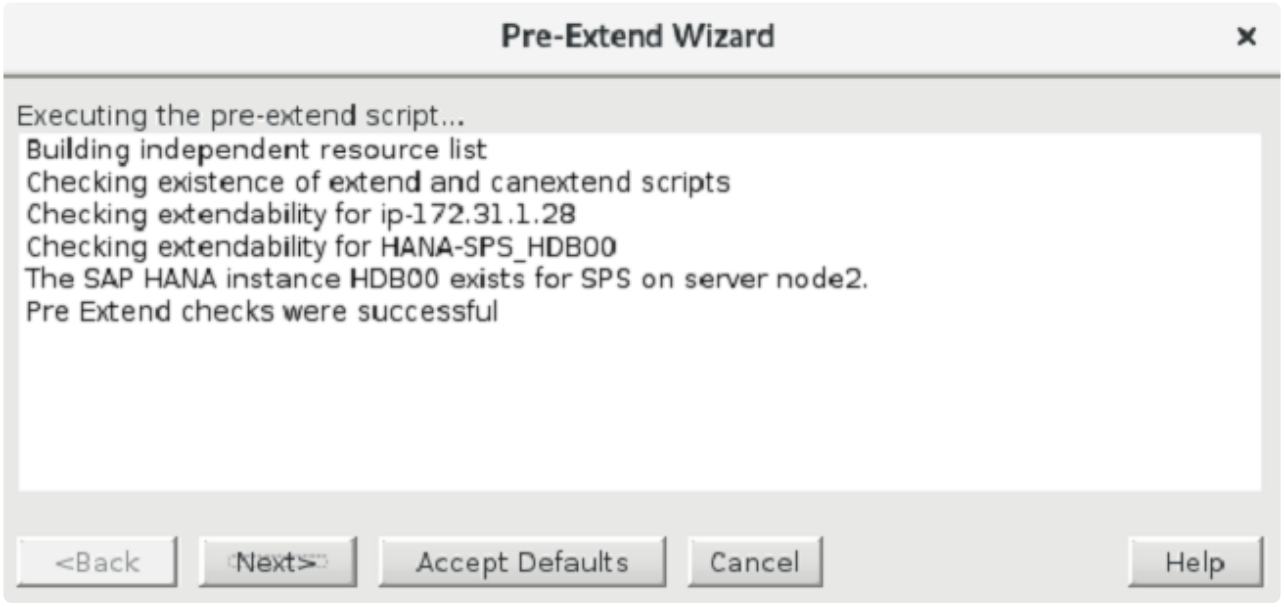
5. **テンプレートの優先順位** を選択または入力します。 **[Next]** をクリックします。

これは、現在稼働中のサーバー上のSAP HANA階層の優先順位です。1～999の未使用の優先順位はすべて有効で、数値が小さいほど優先順位が高くなります（1 =最高）。拡張プロセスは、別のシステムですでに使用されているこの階層の優先順位を拒否します。デフォルト値の使用を推奨します。 **注記：** ここで選択した値は、階層の最初の拡張に対してのみ表示されます。

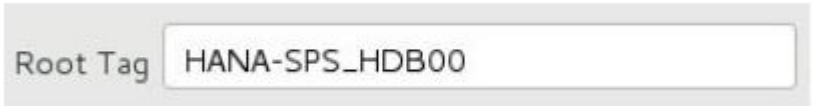
6. **ターゲットの優先順位** を選択または入力し、 **[Next]** をクリックします。

これは、新しく拡張されたSAP HANA階層の、他のサーバー上の同等の階層に対する優先順位です。1～999の未使用の優先順位はすべて有効で、リソースのカスケードフェイルオーバーのシーケンスにおけるサーバーの優先順位を示します。LifeKeeper for Linux は、デフォルトでは階層が作成されるサーバーに優先順位1を割り当てることに注意してください。優先順位は連続している必要はありませんが、2つのサーバーが特定のリソースに対して同じ優先順位を持つことはできません。

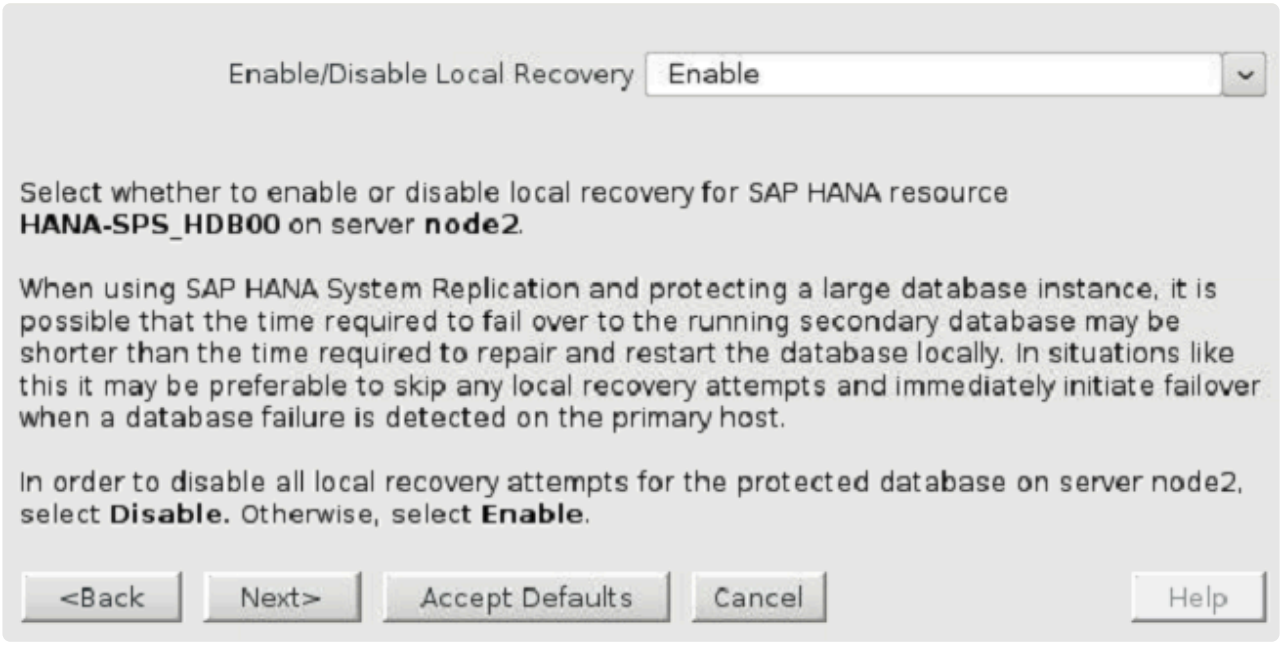
7. 事前拡張チェックが成功したというメッセージが表示されたら、 **[Next]** をクリックします。



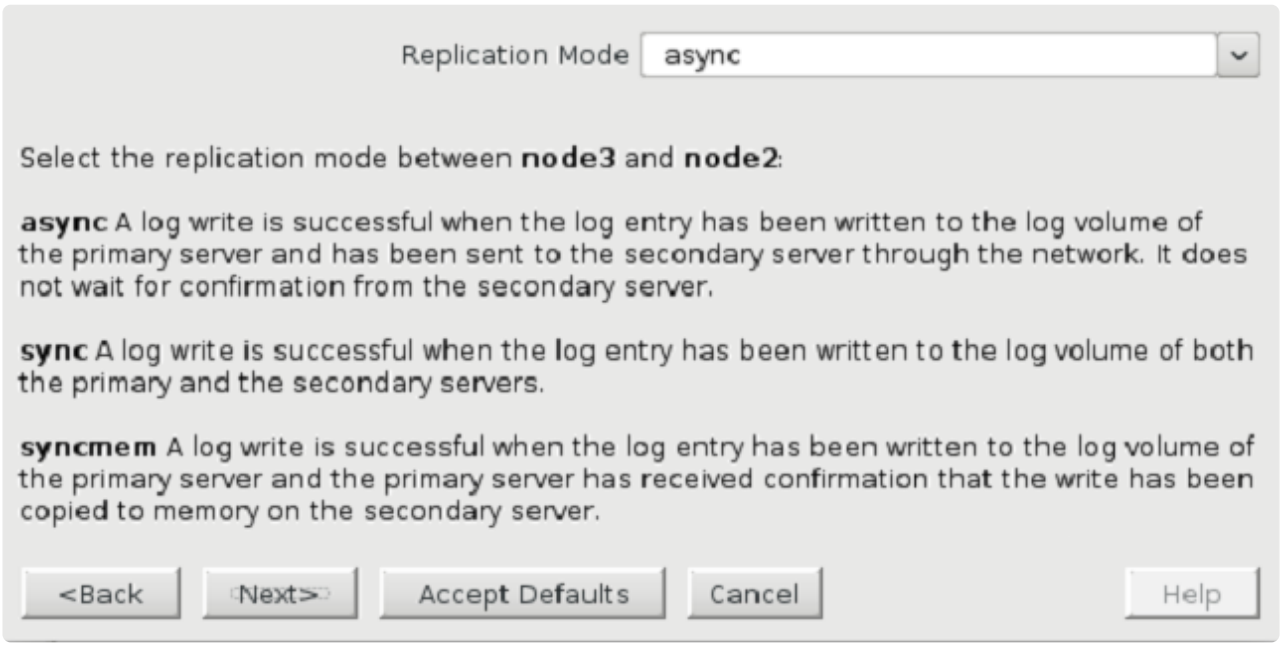

8. ***Extend Wizard*** により、SAP HANA リソースの **ルートタグ** を入力するよう求められます。これは、LifeKeeper がターゲットサーバーで作成される同等のSAP HANAリソースを識別するために使用する一意の名前です。 **注記：** SAP HANA リソースタグ名はすべてのクラスターサーバーで同じである必要があるため、このダイアログボックスでは編集できません。



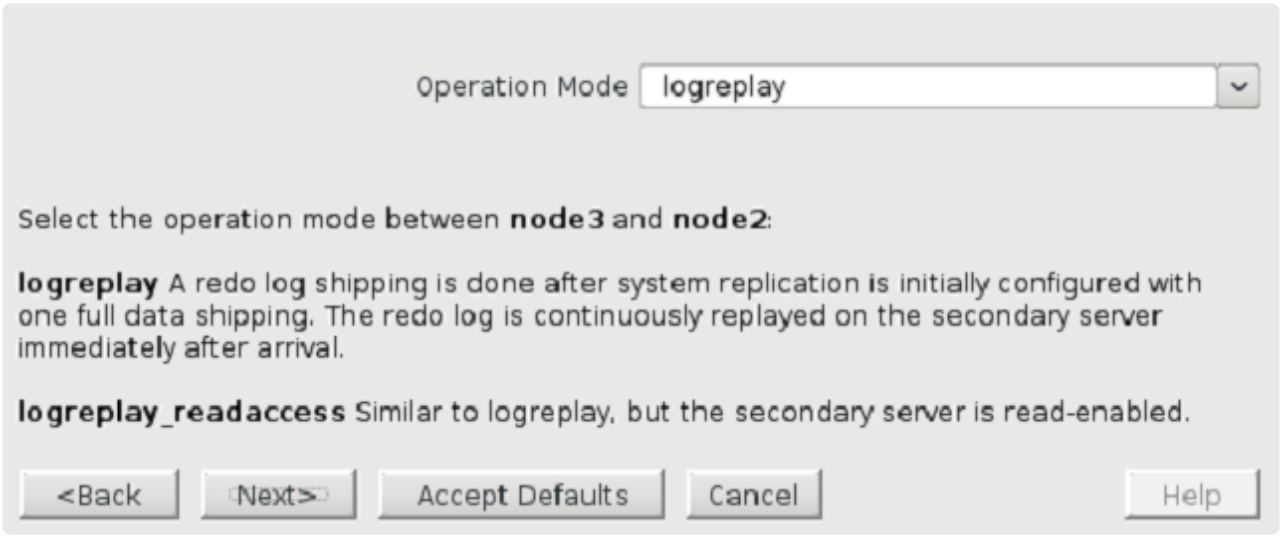
9. 指定したサーバーで SAP HANA リソースに対するローカルリカバリーを **Enable** または **Disable** に設定します。SAP HANA システムレプリケーションを使用している状況において、特に大規模のデータベースを保護している場合、プライマリーデータベースで障害を検知後、現在のプライマリーホスト上でデータベースを停止し再起動するよりは、起動しているスタンバイデータベースインスタンスへ切り替えたほうが早い場合があります。全てのローカルリカバリーの試行を無効にする場合は **Disable** を選択し、そうでない場合は **Enable** を選択してください。



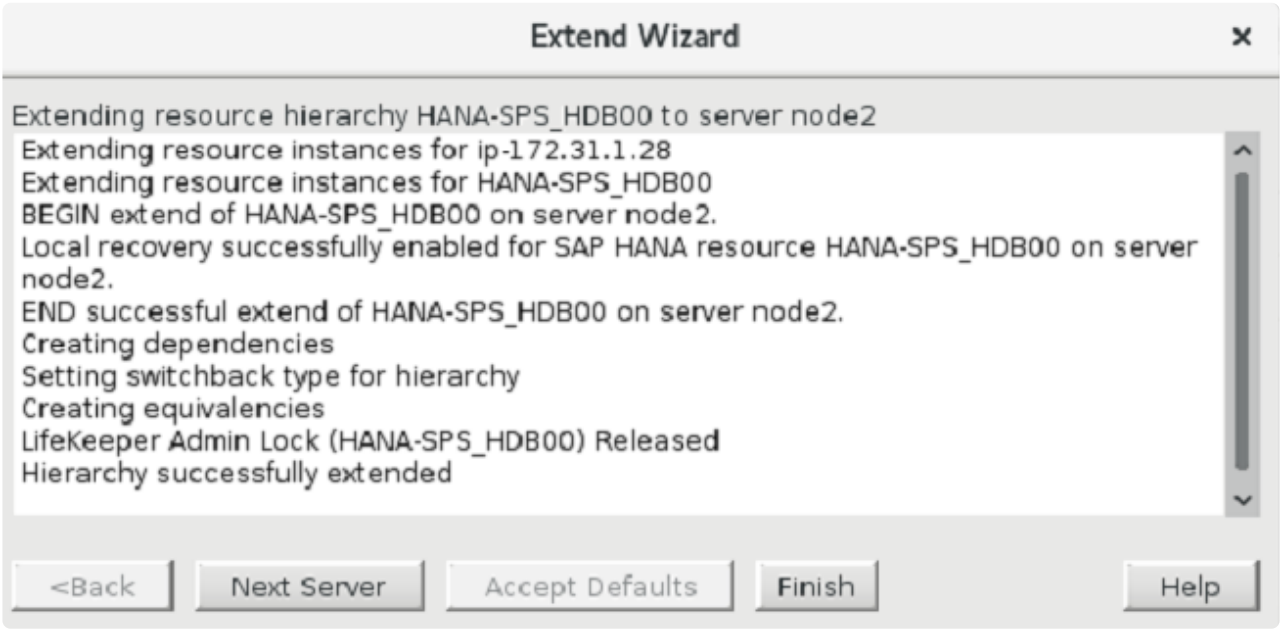

10. (*第3、第4ノードに拡張する場合のみ表示されます。*) 拡張先のノードと前のセカンダリーノードの間のレプリケーションモードを選択します。




11. (*第3、第4ノードに拡張する場合のみ表示されます。*) 拡張先のノードと前のセカンダリーノードの間のオペレーションモードを選択します。

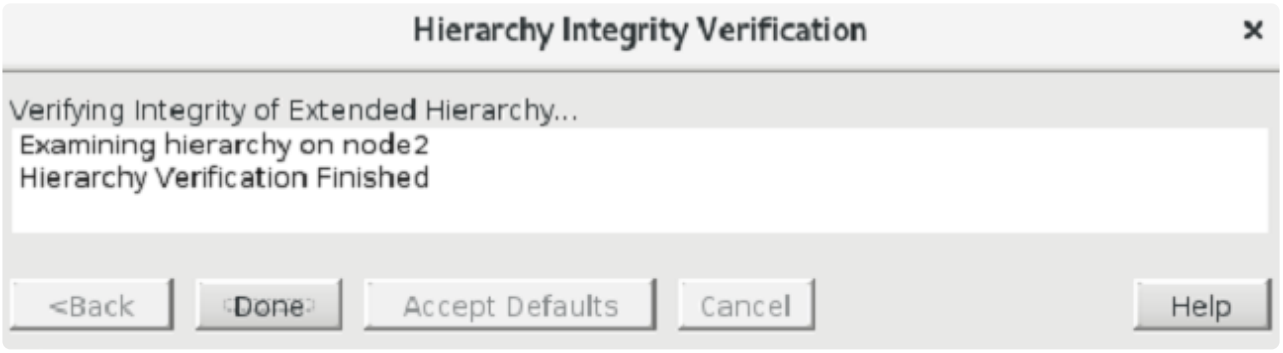



12. **[Extend]** をクリックします。その後 ***Extend Wizard*** は、SAP HANA リソース階層をターゲットサーバーに拡張します。 問題が検出された場合、インフォメーションボックスにエラーメッセージが表示されます。




13. "`Hierarchy successfully extended`"というメッセージを受け取ったら、 **[Next Server]** をクリックして階層を別のサーバーに拡張するか、他に実行する拡張操作がない場合は **[Finish]** をクリックします。

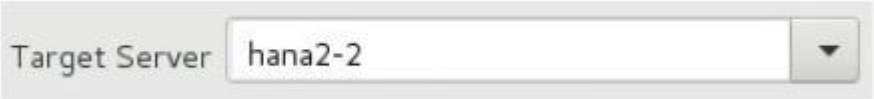
14. "`Hierarchy Verification Finished`"というメッセージが表示されたら、 **[Done]** をクリックします。


Hierarchy Integrity Verification
Verifying Integrity of Extended Hierarchy...
Examining hierarchy on node2
Hierarchy Verification Finished
<Back
Done
Accept Defaults
Cancel
Help

# 6.18.6.3. SAP HANAリソース階層の拡張解除

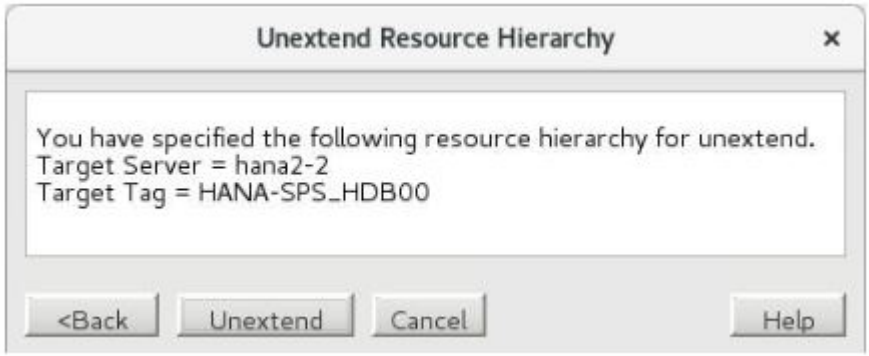
SPSクラスター内の単一サーバーからリソース階層を削除するには、次の手順を実行します。

1. **[Edit]** メニューから、 **[Resource]** 、 **[Unextend Resource Hierarchy]** の順に選択します。
2. SAP HANAリソースの拡張を解除する **ターゲットサーバー** を選択します。リソースが現在 In Service のサーバーは選択できません（ *右ペインでリソースインスタンスを右クリックして **Unextend** タスクを選択した場合、このダイアログボックスは表示されません* ）。 **[Next]** をクリックします。

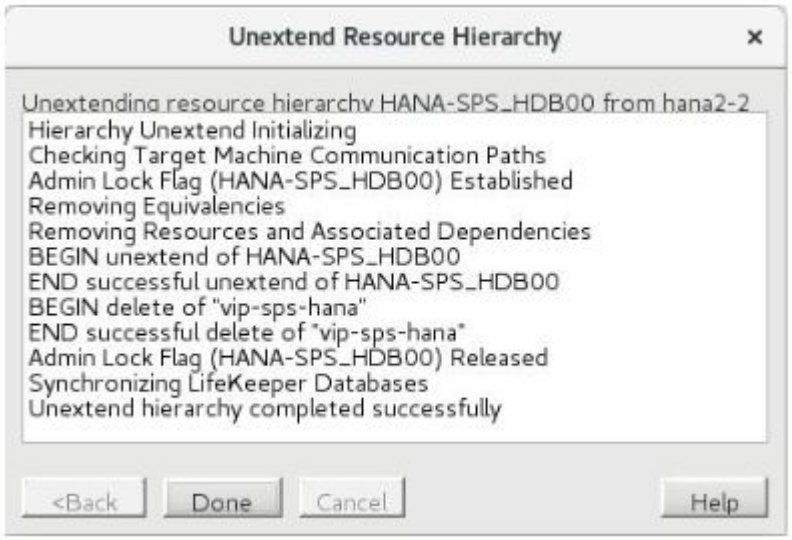



3. 拡張解除するSAP HANA階層を選択し、 **[Next]** をクリックします（ *いずれかのペインでリソースインスタンスを右クリックして **Unextend** タスクを選択した場合、このダイアログボックスは表示されません* ）。

4. 対象のサーバーと、拡張解除を選択したSAP HANAリソース階層を確認する情報ボックスが表示されます。 **[Unextend]** をクリックします。




5. SAP HANAリソースが正常に拡張解除されたことを確認する別の情報ボックスが表示されます。 **[Done]** をクリックして、 **リソース階層の拡張解除** メニューの選択を終了します。


Unextend Resource Hierarchy
Unextending resource hierarchy HANA-SPS_HDB00 from hana2-2
Hierarchy Unextend Initializing
Checking Target Machine Communication Paths
Admin Lock Flag (HANA-SPS_HDB00) Established
Removing Equivalencies
Removing Resources and Associated Dependencies
BEGIN unextend of HANA-SPS_HDB00
END successful unextend of HANA-SPS_HDB00
BEGIN delete of "vip-sps-hana"
END successful delete of "vip-sps-hana"
Admin Lock Flag (HANA-SPS_HDB00) Released
Synchronizing LifeKeeper Databases
Unextend hierarchy completed successfully
<Back
Done
Cancel
Help

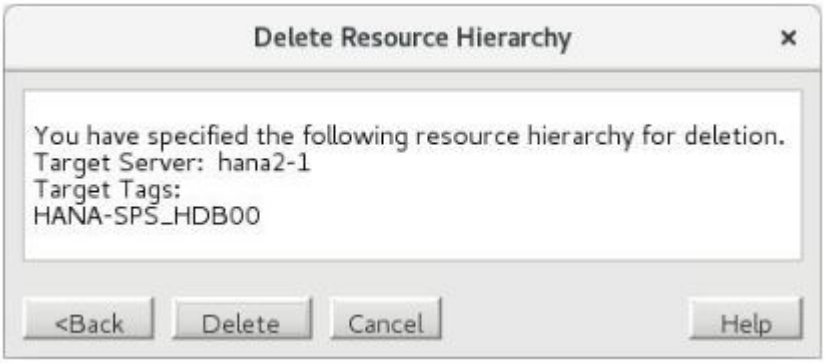
# 6.18.6.4. SAP HANAリソース階層の削除

SPSクラスター環境のすべてのサーバーからSAP HANAリソースを削除するには、次の手順を実行します。

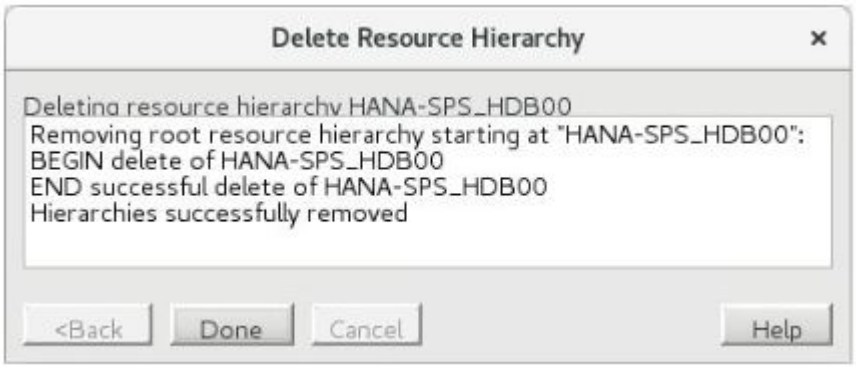
1. **[Edit]** メニューから、 **[Resource]** 、 **[Delete Resource Hierarchy]** の順に選択します。
2. SAP HANAリソース階層を削除する **ターゲットサーバー** の名前を選択します（ *グローバルリソースの左ペインまたは個々のリソースインスタンスの右ペインから右クリックして* ***Delete Resource*** *タスクを選択した場合、このダイアログは表示されません* ）。

3. **削除する階層** を選択します（ *左または右ペインのリソースインスタンスを右クリックして* ***Delete Resource*** *タスクを選択した場合、このダイアログは表示されません* ）。 **[Next]** をクリックします。

4. 削除する対象のサーバーと選択した階層の選択を確認する情報ボックスが表示されます。 **[Next]** をクリックします。




5. SAP HANAリソースが正常に削除されたことを確認する別の情報ボックスが表示されます。




6. **[Done]** をクリックして終了します。

# 6.18.6.5. SAP HANA リソース階層のテスト

## テストシナリオ

SAP HANA Recovery Kitの動作を理解するには、次のテストを実行してください。テストを実行する前に、次の前提条件を満たす必要があります。

- LifeKeeper およびSAP HANA データベースは、SIOS および SAP が提供するインストール手順に従ってインストールおよび設定する必要があります。
- SAP HANA システムレプリケーションは、有効なレプリケーションモード (sync、syncmem、または async) および操作モード (delta_datashipping、logreplay、または logreplay_readaccess) のいずれかを使用して登録された、クラスター内のすべてのサーバーで有効かつアクティブである必要があります。 詳細については、SAP HANA システムレプリケーションの構成 を参照してください。
- SAP HANA データベースに関連付けられた切り替え可能なIPアドレスを LifeKeeper IP リソースで管理する場合、IPリソースに対する SAP HANA リソースの依存関係が存在する必要があります。詳細については、 SAP HANA リソース階層の作成 のステップ 7 を参照してください。

## テストケース

高可用性 SAP HANAクラスターを運用環境に配置する前に、一般的な障害と復旧のシナリオを十分にテストしておくことが非常に重要です。 このページに記載されているテストケースは、高可用性 SAP HANA クラスターデプロイの包括的なテスト計画を開発する際の出発点として使用することを目的としています。以下のサンプル値は、全体を通して使用されます。

| | |
|---|---|
| **Primary Server Host Name** | node1 |
| **Standby Server Host Name** | node2/node3/node4 |
| **SAP SID** | SPS |
| **SAP HANA Database Instance** | HDB00 |
| **SAP HANA LifeKeeper Resource Tag Name** | HANA-SPS_HDB00 |
| **HANA System Replication Primary Site Name** | SiteA |
| **HANA System Replication Secondary Site Name** | SiteB/SiteC/SiteD |
| **HANA System Replication Mode** | sync |
| **HANA System Replication Operation Mode** | logreplay |

テスト時には、これらのサンプル値をテストを実行する環境に適合させる必要があります。

注意：以下のテストケースは、プライマリーノード（node1）で SAP HANA のリソース

階層が In-Service されていることを想定しています。 **完全にカバーするには、node2、node3 などでリソース階層が稼働している間にも同じテストを実行する必要があります。(“node2” はセカンダリーノードを指す言葉として使用されます。“node2” を “node3” などに置き換えてください)** サーバーの役割を逆にしてこれらのシナリオをテストするには、以下に示す各テストケースで、“primary” ↔ “standby”、 “node1” ↔ “node2/node3/node4”、および“SiteA” ↔ “SiteB/SiteC/SiteD”に置換します。

## 手動のスイッチオーバー

このセクションのテストケースでは、手動スイッチオーバーを正常に実行できることを確認します。

| 手動スイッチオーバーテスト |
| --- |
| **説明** |
| SAP HANA リソース階層は、プライマリーノードからスタンバイ サーバーに手動で切り替えることができます。 |
| **前提条件** |
| このテストを実行する前に、次の条件が満たされていることを確認してください。<br>• 全サーバー（ノード 1 とセカンダリーノード）が稼働しており、<br>• SAP HANA リソース階層は、ノード 1 では In-Service (ISP)、ノード 2 では Out-of-Service (OSU) で、<br>• HANA システムのレプリケーションは同期しています。 |
| テストステップ |
| 1. 次のいずれかの方法を使用して、ノード 2でSAP HANAリソース階層を In-Service にします。<br>a. LifeKeeper GUIで、ノード 2の HANA-SPS_HDB 00 リソースを右クリックし、コンテキストメニューから「In Service ...」を選択します。表示される確認ダイアログで、「In Service」をクリックしてスイッチオーバープロセスを開始します。<br>b. ノード 2のターミナルウィンドウから、lkadmin グループ権限を持つユーザー(例えば、root ユーザとして)として次のコマンドを実行します。<br>`[root@node2 ~]# sudo /opt/LifeKeeper/bin/lkcli resource restore --tag HANA-SPS_HDB00` |
| **想定される結果** |
| 1. SAP HANA リソース階層は、ノード 1 で正常に Out-of-Service になりました。 このプロセス中に、データベースはノード 1 で停止され、依存リソース (該当する場合は関連付けられた仮想 IP アドレスなど) もノード 1 で削除されます。<br>2. SAP HANAリソース階層は、ノード 2で正常に In-Service になりました。このプロセス中に、次の操作を行います。 |

a. 依存リソース(関連する仮想IPリソースなど (該当する場合)は、ノード 2 で In-Service になります。
b. ノード 2 で実行中のデータベースインスタンスは、HANA システムレプリケーションのプライマリーレプリケーションロールに昇格されます。
c. ノード 1 のデータベースインスタンスは、適切な HSR パラメーター (サイト名「SiteA」、レプリケーションモード「sync」、操作モード「logreplay」など) を使用して、HANA システムレプリケーションのセカンダリーレプリケーションサイトとして登録されます。
d. データベースインスタンスが ノード 1 で起動されます。

**注記:** ノード 1でのデータベースの登録と再起動のプロセスには、数分かかる場合があります。このプロセスが完了すると、LifeKeeper GUIはノード 1 上のHANA-SPS_HDB 00リソースの状態を「Standby-In Sync」として表示します。

ハンドシェイクテイクオーバーテスト
**注記:** このテストケースは、**LifeKeeper v9.5.2**以降が必要です。

**説明**

SAP HANA のリソース階層は、SAP HANA の「Takeover with Handshake」機能を使用して、プライマリーノードからスタンバイノードに手動で切り替えることができます。

**前提条件**

このテストを実行する前に、次のことを確認してください。

- 両方のサーバー (ノード 1 とノード 2) が稼働しており、
- SAP HANA リソース階層は、ノード 1 では In-Service (ISP)、ノード 2 では Out-of-Service (OSU) で、
- HANA システムのレプリケーションは同期している。

テストステップ

1. 次のいずれかの方法を使用して、ノード 2 で SAP HANA リソース階層を In-Service にします。

   a. LifeKeeper GUI で、ノード 2 の HANA-SPS_HDB00 リソースを右クリックし、コンテキストメニューから [In Service – Takeover with Handshake...] を選択します。 表示される確認ダイアログで、[Perform Takeover] をクリックしてスイッチオーバープロセスを開始します。

   b. ノード 1 または ノード 2 のいずれかのターミナルウィンドウから、lkadmin グループ権限を持つユーザーとして (たとえば、root ユーザーとして) 次のコマンドを実行します。

   ```
   # sudo /opt/LifeKeeper/bin/lkcli resource config hana --tag
   HANA-SPS_HDB00 --takeover_with_handshake node2
   ```

**想定される結果**

1. SAP HANA リソース階層は、ノード 1 で正常に Out-of-Service になります。 このプロセス中、データベースはノード 1 で停止されませんが、依存リソース (該当する場合は関連付けられた仮想 IP

アドレスなど) はノード 1 で削除されます。

2. SAP HANA リソース階層は、ノード 2 正常に In-Service になりました。このプロセス中に、次の操作を行います。
   a. 依存するリソース（関連する仮想IPリソースなど、該当する場合）は、ノード 2 上で In-Service 化されます。
   b. ノード 2 上の稼働中のデータベースインスタンスは、HANA システムレプリケーションのプライマリーレプリケーションの役割に昇格します。HSR のテイクオーバープロセス中、ノード 1 上のデータベースインスタンスは中断されます。
   c. ノード 1 の中断されたデータベースインスタンスが停止されます。
   d. ノード 1 のデータベースインスタンスは、適切な HSR パラメーター (サイト名「SiteA」、レプリケーションモード「sync」、操作モード「logreplay」など) を使用して、HANA システムレプリケーションのセカンダリーレプリケーションサイトとして登録されます。
   e. ノード 1 でデータベースインスタンスを起動します。

**注記:** ノード 1 のデータベースの停止、登録、再起動のプロセスには数分かかる場合があります。このプロセスが完了すると、LifeKeeper の GUI にはノード 1 の HANA-SPS_HDB00 リソースの状態が「Standby – In Sync」と表示されるようになります。

## グレイスフルシャットダウン

このセクションのテストケースでは、各サーバーが正常に再起動されたときの SAP HANA リソース階層の予想される動作を検証します。

✻ 注意：最初のテストで予測される結果は、「シャットダウン時のスイッチオーバー」方式がプライマリーノードに対して有効か無効かによって異なります。詳細については、「サーバーのシャットダウンストラテジーの設定」を参照してください。

| プライマリーノードリブートテスト |
| --- |
| **説明** |
| プライマリーノードが正常に再起動されると、SAP HANA リソース階層は期待通りに動作します。 |
| **前提条件** |
| このテストを実行する前に、次のことを確認してください。<br>• 両方のサーバー (ノード 1 とノード 2) が稼働しており、<br>• SAP HANA リソース階層は、ノード 1 では In-Service (ISP)、ノード 2 では Out-of-Service (OSU) で、<br>• HANA システムのレプリケーションは同期している。 |
| テストステップ |
| 1. グレイスフルリブート ノード 1: |

```
[root@node1 ~]# reboot now
```

**想定される結果**

1. ノード1 がシャットダウンしている間
   a. SAP HANA リソース階層は、ノード1 で正常に Out-of-Service になりました。 このプロセス中に、データベースはノード 1 で停止され、依存リソース (該当する場合は関連付けられた仮想 IP アドレスなど) もノード 1 で削除されます。
   b. **[node1 で「Switchover on Shutdown」が有効になっている場合]** SAP HANAリソース階層は、ノード 2で正常に In-Service になりました。このプロセス中に、次の操作を行います。
      i. 依存するリソース (該当する場合は関連付けられた仮想 IP リソースなど) は、ノード 2 で In-Service になります。
      ii. ノード 2 で実行中のデータベースインスタンスは、HANA システムレプリケーションのプライマリーレプリケーションロールに昇格されます。 ノード 1 がシャットダウンされたため、HANA システムのレプリケーションは現在非アクティブです。

2. ノード 1 がオンラインに戻った後:
   a. **[ノード 1 で「Switchover on Shutdown」が無効になっている場合]** SAP HANAリソース階層は、ノード 1で自動的に In-Service に戻ります。このプロセス中に、次の操作を行います。
      i. 依存するリソース (該当する場合は関連付けられた仮想 IP リソースなど) は、ノード 1 で In-Service になります。
      ii. ノード 1 のデータベースインスタンスが開始され、HANA システムレプリケーションが ノード 2 のセカンダリーレプリケーションサイトに再開されます。
   b. **[ノード 1 で「Switchover on Shutdown」が有効になっている場合]** SAP HANA リソース階層は、ノード 2 で In-Service のままです。
      i. ノード 1 がオンラインに戻った後の ノード 2での最初の quickCheck サイクル中に、SAP HANA Recovery Kit によって、データベースインスタンスが ノード 1で実行されていないことが検出され、“remoteregisterdb“イベントが発生します。
      ii. 「remoteregisterdb」イベントスクリプトは、適切な HSR パラメーター (サイト名「SiteA」、レプリケーションモード「sync」、操作モード「logreplay」など) を使用してノード 1 をセカンダリー HANA システムレプリケーションサイトとして登録し、ノード 1 でデータベースインスタンスを起動します。

      **注意：** ノード 1 でデータベースを登録して再起動するプロセスには、数分かかる場合があります。 このプロセスが完了すると、LifeKeeper GUI はノード 1 の HANA-SPS_HDB00 リソースの状態を「Standby – In Sync」として表示します。

## スタンバイノードリブートテスト

**説明**

スタンバイノードが正常に再起動されると、SAP HANA リソース階層は期待通りに動作します。

**前提条件**

このテストを実行する前に、次のことを確認してください。

- 両方のサーバー (ノード 1 とノード 2) が稼働しており、
- SAP HANA リソース階層は、ノード 1 では In-Service (ISP)、ノード 2 では Out-of-Service (OSU) で、
- HANA システムのレプリケーションは同期している。

テストステップ

1. グレイスフルリブート　ノード 2

```
[root@node2 ~]# reboot now
```

想定される結果

1. ノード 2 がシャットダウンしている間、データベースインスタンスは ノード 2 で停止します。 ノード 2 が再起動し、セカンダリーデータベースインスタンスが再起動されるまで、HANA システムレプリケーションは非アクティブになります。

2. ノード 2 がオンラインに戻った後
   a. ノード 2 がオンラインに戻った後の ノード 1 での最初の quickCheck サイクル中に、SAP HANA Recovery Kit はデータベースインスタンスが ノード 2 で実行されていないことを検出し、「remoteregisterdb」イベントを発生させます。
   b. 「remoteregisterdb」イベント スクリプトは、ノード 2 でデータベースインスタンスを開始します。

**注記:** ノード 2 でデータベースを再起動するプロセスには、数分かかる場合があります。 このプロセスが完了すると、LifeKeeper GUI は ノード 2 の HANA-SPS_HDB00 リソースの状態を「Standby – In Sync」として表示します。

## マシンフェイルオーバー

このセクションのテストケースでは、プライマリーノードが強制的に再起動されたときの SAP HANA リソース階層の予想される動作を検証します。

マシンフェイルオーバーテスト

説明

プライマリーノードが強制的に再起動されると、SAP HANA リソース階層が期待通りに動作します。

前提条件

このテストを実行する前に、次のことを確認してください。

- 両方のサーバー (ノード 1 とノード 2) が稼働しており、
- SAP HANA リソース階層は、ノード 1 では In-Service (ISP)、ノード 2 では Out-of-Service (OSU) で、
- HANA システムのレプリケーションは同期している。

テストステップ

1. ノード 1 を強制的に再起動します。

```
[root@node1 ~]# echo b > /proc/sysrq-trigger
```

**想定される結果**

1. ノード 2のLifeKeeperが ノード 1がダウンしていることを検出すると(正確な時間は「LifeKeeper ハートビートの調整」に使用される値によって異なります。)、SAP HANAリソース階層が ノード 2で正常に In-Service になります。 このプロセス中に、次の操作を行います。
   a. 依存するリソース (該当する場合は関連付けられた仮想 IP リソースなど) は、ノード 2 で In-Service になります。
   b. ノード 2 で実行中のデータベースインスタンスは、HANA システムレプリケーションのプライマリーレプリケーションロールに昇格されます。 ノード 1 がシャットダウンされたため、HANA システムのレプリケーションは現在非アクティブです。

2. ノード 1 がオンラインに戻った後、SAP HANA リソース階層は ノード 2 で In-Service のままになります。
   a. ノード 1 がオンラインに戻った後の ノード 2 での最初の quickCheck サイクル中に、SAP HANA Recovery Kit はデータベースインスタンスが ノード 1 で実行されていないことを検出し、「remoteregisterdb」イベントを発生させます。
   b. 「remoteregisterdb」イベント スクリプトは、適切な HSR パラメーター (サイト名「SiteA」、レプリケーションモード「sync」、操作モード「logreplay」など) を使用してノード 1 をセカンダリー HANA システムレプリケーションサイトとして登録し、ノード 1 でデータベースインスタンスを起動します。

      **注意：**ノード 1 でデータベースを登録して再起動するプロセスには、数分かかる場合があります。 このプロセスが完了すると、LifeKeeper GUI はノード 1 の HANA-SPS_HDB00 リソースの状態を「Standby – In Sync」として表示します。

## SAP ホストエージェントの障害

このセクションのテストケースは、サポートする SAP Host Agent 関連のプロセスが各サーバーで失敗した場合に、SAP HANA リソース階層に期待される動作を検証します。

**プライマリーノード SAP ホスト実行障害テスト**

**説明**

プライマリーノードで saphostexec プロセスが強制終了されると、SAP HANA リソース階層は期待通りに動作します。

**前提条件**

このテストを実行する前に、次のことを確認してください。

- 両方のサーバー (ノード 1 とノード 2) が稼働しており、
- SAP HANA リソース階層は、ノード 1 では In-Service (ISP)、ノード 2 では Out-of-Service (OSU) で、

- HANA システムのレプリケーションは同期している。

テストステップ

ノード 1 で saphostexec プロセスを停止します。

```
[root@node1 ~]# /usr/sap/hostctrl/exe/saphostexec -stop
```

想定される結果

1. 次の quickCheck 間隔で、SAP HANA Recovery Kit は saphostexec プロセスが ノード 1 で実行されていないことを検出し、「recover」イベントを起動して再起動します。
2. 「recover」イベントスクリプトは、ノード 1 で saphostexec プロセスを再起動します。

### スタンバイノード SAP Host Exec 障害テスト

説明

saphostexec プロセスがスタンバイノードで強制終了されると、SAP HANA リソース階層は期待通りに動作します。

前提条件

このテストを実行する前に、次のことを確認してください。

- 両方のサーバー (ノード 1 とノード 2) が稼働しており、
- SAP HANA リソース階層は、ノード 1 では In-Service (ISP)、ノード 2 では Out-of-Service (OSU) で、
- HANA システムのレプリケーションは同期している。

テストステップ

1. ノード 2 で saphostexec プロセスを停止します。

   ```
   [root@node2 ~]# /usr/sap/hostctrl/exe/saphostexec -stop
   ```

想定される結果

1. 次の quickCheck 間隔で、SAP HANA Recovery Kit は saphostexec プロセスが ノード 2 で実行されていないことを検出し、「remoteregisterdb」イベントを起動して再起動します。
2. 「remoteregisterdb」イベントスクリプトは、ノード 2 で saphostexec プロセスを再起動します。

### プライマリーノード SAP OS Collector 障害テスト

説明

saposcol プロセスがプライマリーノードで強制終了された場合、SAP HANA リソース階層は期待通りに動作します。

**前提条件**

このテストを実行する前に、次のことを確認してください。

- 両方のサーバー (ノード 1 とノード 2) が稼働しており、
- SAP HANA リソース階層は、ノード 1 では In-Service (ISP)、ノード 2 では Out-of-Service (OSU) で、
- HANA システムのレプリケーションは同期している。

**テストステップ**

ノード 1 で `saposcol` プロセスを停止します。

```
[root@node1 ~]# /usr/sap/hostctrl/exe/saposcol -k
```

**想定される結果**

1. 次の quickCheck 間隔で、SAP HANA Recovery Kit は `saposcol` プロセスが ノード 1 で実行されていないことを検出し、「recover」イベントを起動して再起動します。
2. 「recover」イベント スクリプトは、ノード 1 で `saposcol` プロセスを再起動します。

## スタンバイノード SAP OS Collector 障害テスト

**説明**

`saposcol` プロセスがスタンバイノードで強制終了されると、SAP HANA リソース階層は期待通りに動作します。

**前提条件**

このテストを実行する前に、次のことを確認してください。

- 両方のサーバー (ノード 1 とノード 2) が稼働しており、
- SAP HANA リソース階層は、ノード 1 では In-Service (ISP)、ノード 2 では Out-of-Service (OSU) で、
- HANA システムのレプリケーションは同期している。

**テストステップ**

ノード 2 で `saposcol` プロセスを停止します。

```
[root@node2 ~]# /usr/sap/hostctrl/exe/saposcol -k
```

**想定される結果**

1. 次の quickCheck 間隔中に、SAP HANA Recovery Kit は `saposcol` プロセスが ノード 2 で実行されていないことを検出し、「remoteregisterdb」イベントを起動して再起動します。
2. 「remoteregisterdb」イベントスクリプトは、ノード 2 で `saposcol` プロセスを再起動します。

## データベース障害

このセクションのテストケースでは、保護された HANA データベースインスタンス内のプロセスが失敗した場合の SAP HANA リソース階層の予想される動作を検証します。

**注意：**

- このセクションで期待される結果の一部は、SAP HANA リソースのローカルリカバリーが有効か無効かによって異なります。 ローカルリカバリーはデフォルトで有効になっています。 詳細については「SAP HANA リソースのローカルリカバリーポリシーおよび TemporalRecoveryPolicy の設定」を参照してください。

- これらのテストの実行中にデータベースインスタンスが正常に停止した場合 (たとえば、`HDB stop` コマンドを使用)、データベースを正常に停止するバックグラウンドプロセスが SAP HANA Recovery Kit のプロセスと競合する可能性があります。 データベースをローカルで再起動しようとすると、SAP HANA リソース階層のフェイルオーバーが発生する可能性があります。 このため、`HDB kill-9` コマンドなどを使用して、オペレーティングシステムレベルでデータベースプロセスを強制的かつ即時に強制終了することにより、HDB インスタンスのクラッシュをシミュレートすることをお勧めします。詳細については「SAP HANA – 既知の問題と制限」を参照してください。

| プライマリーノード データベースインスタンス障害テスト |
|---|
| **説明** |
| プライマリーノードで HDB インスタンスプロセスが強制終了された場合、SAP HANA リソース階層は期待通りに動作します。 |
| **前提条件** |
| このテストを実行する前に、次のことを確認してください。<br><br>• 両方のサーバー (ノード 1 とノード 2) が稼働しており、<br>• SAP HANA リソース階層は、ノード 1 では In-Service (ISP)、ノード 2 では Out-of-Service (OSU) で、<br>• HANA システムのレプリケーションは同期している。 |
| **テストステップ** |
| HANA 管理ユーザーとして (<sid>adm)、ノード 1 の HDB インスタンスプロセスを (オペレーティングシステムレベルで) 強制終了します。<br><br>`[root@node1 ~]# su - spsadm -c "HDB kill-9"` |
| **想定される結果** |
| 1. **[ノード 1 の HANA-SPS_HDB00 リソースに対してローカルリカバリが有効になっている場合]** SAP HANA Recovery Kit が障害を検出し、ノード 1 でデータベースインスタンスを再起動します。<br>a. 次の quickCheck の間隔で、SAP HANA Recovery Kit は ノード 1上でHDB インスタンスプロセスが実行されていないことを検出し、それらを再起動するために「recover」イベントを |

発生させます。
b. 「recover」イベントスクリプトは、ノード 1 で HDB インスタンスプロセスを再起動します。

2. **[ノード 1 の HANA-SPS_HDB00 リソースのローカルリカバリが無効になっている場合]** LifeKeeper は、SAP HANA リソース階層の ノード 2 へのフェイルオーバーをただちに開始します。
   a. SAP HANA リソース階層は、ノード 1 で正常に Out-of-Service になりました。 この処理中に、依存リソース (該当する場合は関連付けられた仮想 IP アドレスなど) が ノード 1 で削除されます。
   b. SAP HANAリソース階層は、ノード 2で正常に In-Service になりました。この処理中に、次の操作を行います。
      i. 依存するリソース (該当する場合は関連付けられた仮想 IP リソースなど) は、ノード 2 で In-Service になります。
      ii. ノード 2 で実行中のデータベースインスタンスは、HANA システムレプリケーションのプライマリーレプリケーションロールに昇格されます。
      iii. ノード 1 のデータベースインスタンスは、適切な HSR パラメーター (サイト名「SiteA」、レプリケーションモード「sync」、操作モード「logreplay」など) を使用して、HANA システムレプリケーションのセカンダリーレプリケーションサイトとして登録されます。
      iv. データベースインスタンスが ノード 1 で開始されます。

      **注意：**ノード 1 でデータベースを登録して再起動する処理には、数分かかる場合があります。 この処理が完了すると、LifeKeeper GUI はノード 1 の HANA-SPS_HDB00 リソースの状態を「Standby – In Sync」として表示します。

## スタンバイノード データベースインスタンス 障害テスト

**説明**

スタンバイノードで HDB インスタンスプロセスが強制終了された場合、SAP HANA リソース階層は期待通りに動作します。

**前提条件**

このテストを実行する前に、次のことを確認してください。

- 両方のサーバー (ノード 1 とノード 2) が稼働しており、
- SAP HANA リソース階層は、ノード 1 では In-Service (ISP)、ノード 2 では Out-of-Service (OSU) で、
- HANA システムのレプリケーションは同期している。

テストステップ

HANA 管理ユーザー (<sid>adm) として、ノード 2 の HDB インスタンスプロセスを (オペレーティングシステムレベルで) 強制終了します。

```
[root@node2 ~]# su - spsadm -c "HDB kill-9"
```

| 想定される結果 |
| --- |
| 1. 次の quickCheck 間隔で、SAP HANA Recovery Kit は HDB インスタンスプロセスが ノード 2 で実行されなくなったことを検出し、「remoteregisterdb」イベントを起動して再起動します。<br>2. 「remoteregisterdb」イベント スクリプトは、ノード 2 で HDB インスタンスプロセスを再起動します。 |

# 付録: 便利な SAP HANA 管理コマンド

SAP HANA 環境のステータスは、SAP が提供するダッシュボード (HANA Studio や HANA Cockpit など) を介して監視できますが、テスト中に次のコマンドも役立つ場合があります。 全体を通して、<sid> は保護された SAP HANA データベース インストールの小文字の SAP SID を示し、<InstNum> は保護された HDB インスタンスのインスタンス番号を示します (たとえば、インスタンス HDB00、<InstNum> は 00)。

| コマンド | 説明 |
| --- | --- |
| su – <sid>adm -c “sapcontrol -nr <InstNum> -function StopService” | HDB インスタンスの sapstartsrv プロセスを停止します。 |
| su – <sid>adm -c “sapcontrol -nr <InstNum> -function StartService <SID>” | HDB インスタンスの sapstartsrv プロセスを開始します。 |
| su – <sid>adm -c “sapcontrol -nr <InstNum> -function GetProcessList” | HDB インスタンスプロセスの現在のステータスを表示します。 |
| su – <sid>adm -c “HDB stop” | HDB インスタンスを正常に停止します。 |
| su – <sid>adm -c “HDB kill-9” | HDB インスタンスプロセスを強制終了します。 |
| su – <sid>adm -c “HDB start” | HDB インスタンスを開始します。 |
| su – <sid>adm -c “hdbnsutil -sr_state” | ローカルサーバーで HANA システムのレプリケーション状態を確認します。 |
| su – <sid>adm -c “python /hana/shared/<SID>/HDB<InstNum>/exe/python_support/systemReplicationStatus.py” | 現在の HANA システムのレプリケーションステータスを確認します。 このコマンドは、プライマリー HANA システムレプリケーションサイトであるサーバーで実行する必要があります。 |
| su – <sid>adm -c “hdbsql -n <HANA virtual host name> -i <InstNum> -u SYSTEM -p <SYSTEM user password> -d <SID> ‘\s’” | 関連する仮想ホスト名を介して <SID> テナントデータベースへの接続をテストします。 |
| /usr/sap/hostctrl/exe/saphostexec -status | saphostexec のステータスを確認してください。 |
| /usr/sap/hostctrl/exe/saphostexec -stop | saphostexec を停止します。 |

| | |
|---|---|
| /usr/sap/hostctrl/exe/saphostexec -restart | saphostexec を再起動します。 |
| /usr/sap/hostctrl/exe/saposcol -s | saposcol のステータスを確認してください。 |
| /usr/sap/hostctrl/exe/saposcol -k | saposcol を停止します。 |
| /usr/sap/hostctrl/exe/saposcol -l | saposcol を開始します。 |
| top -U <sid>adm<br>ps -ef \| grep <sid>adm | <sid>adm ユーザーが所有するプロセスの情報を表示します。 |

# 6.18.7. SAP HANA リソース階層の管理

重要： 本ガイドで特に明記されていない限り、LifeKeeper で保護されたSAP HANA データベースのすべての管理タスクは、LifeKeeper を介して実行する必要があります。LifeKeeper でSAP HANA リソースがアクティブ/ISPな場合にデータベースの停止や LifeKeeper の外部でのSAP HANA システムレプリケーションの無効化などの管理操作を実行すると、LifeKeeper はデータベースを望ましい実行状態に戻すアクションを実行します。

## SAP HANA リソースのスイッチオーバー

プライマリーデータベースインスタンスのスイッチオーバーが開始されると、SAP HANA Recovery Kit は次の手順を実行します。

- データベースインスタンスを以前のプライマリーノードで停止します。
- 新しいプライマリーノード（以前のセカンダリーノード）でSAP HANA システムレプリケーションのテイクオーバーを実行します。
- 各スタンバイノード（以前のプライマリーノードを含む）は、新しいプライマリーサイトに SAP HANA System Replication のセカンダリーサイトとして再登録されます。
- データベースインスタンスを、以前のプライマリーノードで起動します。

```
ERROR:hana:restore:HANA-SPS_HDB00:136266:The contents of SAP HANA database $instance may not be in sync on $me. To protect the data, LifeKeeper will not restore $tag. Please restore the resource on the previous source server to allow the resync to complete. To force the resource in-service use “lkcli hana force --sys $me --tag $tag” or the GUI option “Force In Service”.
```

SAP HANA リソースを in-service にしているときにこのメッセージが表示された場合、プライマリーデータベースインスタンスが停止したときに、SAP HANA システムレプリケーションが同期していなかったことを意味します。 したがって、セカンダリーデータベースサーバーにまだ複製されていないデータがプライマリーデータベースサーバーに存在する可能性があります。このため、LifeKeeperはセカンダリーサーバーがプライマリーレプリケーションの役割を引き継ぐことを許可しません。この場合、SAP HANA リソース階層を以前のプライマリーサーバーで In Service に戻し、再同期を完了することを推奨します。

以前のプライマリーサーバーがダウンして回復できない場合、詳細は SAP HANA 階層の restore 失敗 を参照してください。

## SAP HANA データベースの Takeover with Handshake

“Takeover with handshake” は、SAP HANA 2.0 SPS04 以降に追加された機能で、新しいデータベースホスト上で SAP HANA システムレプリケーションの引き継ぎを行う前にプライマリーデータベースを (完全に停止するのではなく)—時停止することで、スイッチオーバー時のプライマリーデータベースのダウンタ

イムを抑えます。SAP HANA Recovery Kit で “Takeover with handshake” を実施する場合は、 Takeover with Handshake を参照してください。

## SAP HANA データベースの停止

LifeKeeper で SAP HANA リソースが Out of Service になると、プライマリーデータベースインスタンスのみが停止します。セカンダリーデータベースインスタンスは、HANAリソース階層のスイッチオーバーまたはフェイルオーバー中のダウンタイムを最小限に抑えるために引き続き実行されます。

例外的な動作を引き起こす2つの特別なケースがあります。

1. SAP HANA リソースが Out of Service になっているシステム上において、LifeKeeper フラグ `!volatile!hana_leave_db_running_<HANA Tag>` が設定されている場合は、LifeKeeper は Out of Service の操作中にデータベースインスタンスを停止しません。このフラグが意図せず設定されている場合は、下記のコマンドを使用して削除することができます。:

   ```
   /opt/LifeKeeper/bin/flg_remove -f '!volatile!hana_leave_db_running_<HANA Tag>'
   ```

2. SAP HANA リソースが Out of Service になっているシステムにおいてデータベースが一時停止している場合、LifeKeeper は Out of Service の操作中にデータベースインスタンスを停止しません。一時停止されたデータベースインスタンスを起動し続けることで、必要に応じて再開するオプションを残しておくことができます。このシナリオは、LifeKeeper の外部でデータベース管理者による “Takeover with handshake” が手動で行われた際に最も発生しやくなります。必要に応じて一時停止されたデータベースインスタンスは下記のコマンドを使用して手動で停止することが可能です。:

   ```
   su - <sid>adm -c "sapcontrol -nr <InstNum> -function StopWait 600 5"
   ```

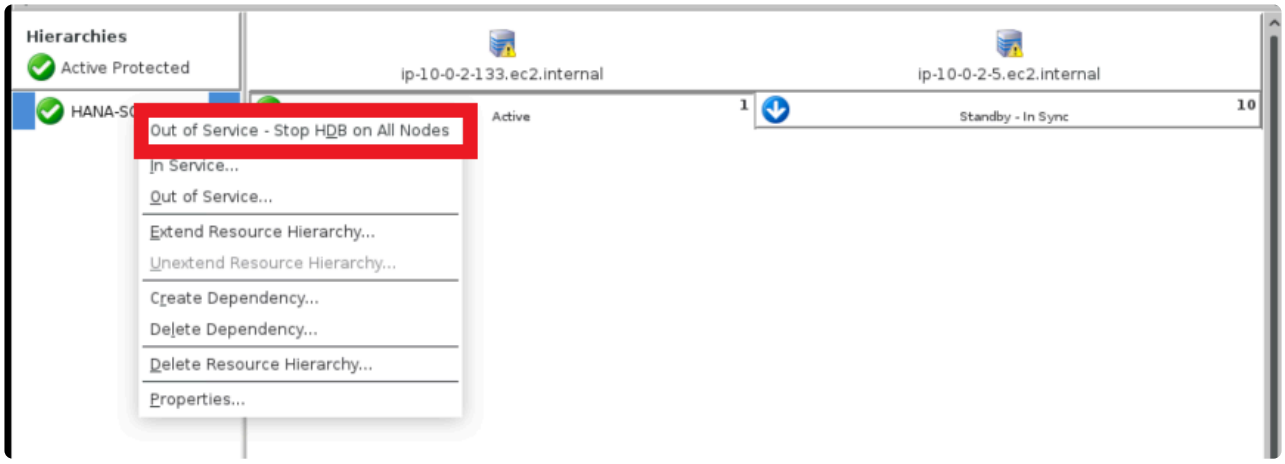
   <sid> はHANAインストール時に使用した小文字のSAPシステムID、<InstNum> はHDBインスタンス番号です。

## すべての SAP HANA データベースの停止（メンテナンスモード）

このオプションを選択すると、プライマリー HANA リソースが Out of Service になり、HANA リソースクラスター内のすべての HANA データベースインスタンスが停止します。このオプションにより、バックアップマシンにすぐにフェイルオーバー/スイッチオーバーされる可能性があるため、細心の注意を払って実行する必要があります。 **注記：** このオプションは、メンテナンス期間中など、セカンダリーデータベースインスタンスも停止する必要がある場合にのみ選択してください。

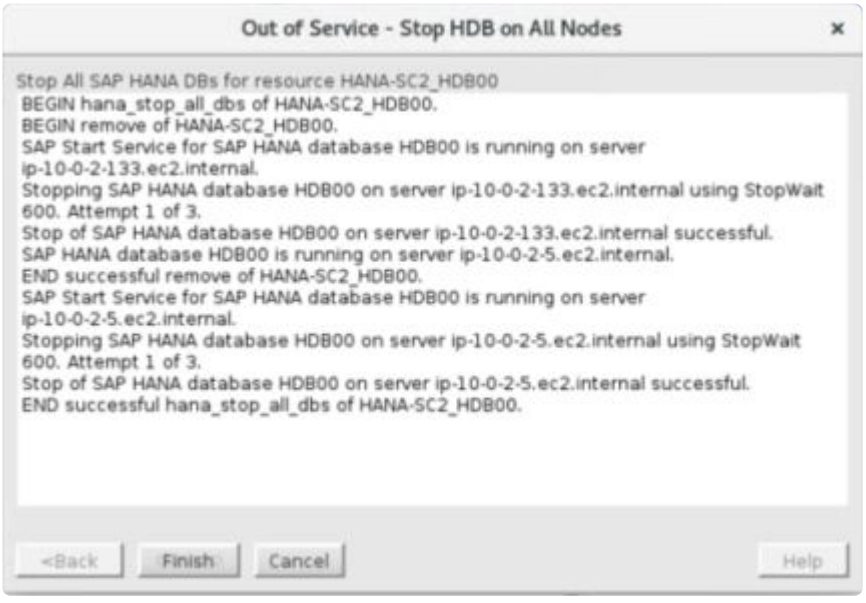
このオプションを使用するには、HANA リソース階層で次の手順を実行します

1. 左パネルの下の HANA リソースまたは In Service のサーバーのいずれかを右クリックして、 **[Out of Service – Stop HDB on All Nodes]** オプションを選択します。




2. HANA リソースを確認し、ダイアログボックスに表示される指示に従います。 **[Stop All SAP HANA DBs]** をクリックしてプロセスを開始します。

3. プロセスが終了したら、 **[Finish]** をクリックしてプロセスを完了します。



4. SAP HANA リソースの最終状態は次のように表示されます。

5. すべてのメンテナンス作業が完了したら、最後のプライマリーシステムで SAP HANA リソース階層を In Service にしてください。

保護対象の SAP HANA データーベースは、SAP HANA リソースが現在 In Service (ISP) になっているサーバー上で次のコマンドを実行することで クラスター内の全てのサーバー上で停止することもできます。

```
/opt/LifeKeeper/bin/hana_stop_all_dbs -t <HANA Resource Tag>
```

# 6.18.7.1. レプリケーションモードおよびオペレーションモードの変更

! セカンダリーレプリケーションサイトのレプリケーションモードとオペレーションモードは変更できますが、LifeKeeperの外部でSAP HANA システムレプリケーションの手動によるテイクオーバーまたはスイッチオーバーは実行しないでください。これを行うとエラー状態になり、SAP HANA システムレプリケーションモードは LifeKeeper がアクティブ / スタンバイサーバーで期待しているモードと一致しなくなります。このエラー状態は、LifeKeeper GUI に次のリソース状態で表示されます。

この状態の間、SAP HANA リソースの監視はすべて、プライマリーレプリケーションサイトとなるサーバー上の LifeKeeper でリソースが In Service になるまで中断されます。

## レプリケーションモードの変更

データベースがセカンダリーレプリケーションサイトで実行されている場合でも、 "hdbnsutil -sr_changemode" コマンドを使用して SAP HANA システムレプリケーションモードを変更できます。

1. セカンダリーレプリケーションサイトで次のコマンドを実行します。

```
su - <sid>adm -c "hdbnsutil -sr_changemode --mode=[sync|syncmem|async]"
```

<sid> は、SAP HANA インストール時に使用した小文字の SAP システム ID で、希望するレプリケーションモード（sync、syncmem、またはasync）を --mode オプションで指定します。このコマンドを実行すると、セカンダリーサイトでデータベースを実行または停止できます。

2. レプリケーションモードが正常に変更されたことを確認するには、セカンダリーレプリケーションサイトで次のコマンドを実行し、"mode" パラメーターが新しいレプリケーションモードに更新されたことを確認します。

```
su - <sid>adm -c "hdbnsutil -sr_state"
```

次の quickCheck 間隔（デフォルトは2分）後、LifeKeeper は変更を自動的に検出し、同等のすべての SAP HANA リソースの情報フィールドを更新して新しいレプリケーションモードを含めます。これは、次のコマンドを使用して HANA リソースの情報フィールドを調べることで確認できます。

```
/opt/LifeKeeper/bin/ins_list -d <HostName> -t <Tag> -f: | cut -d: -f6 | tr '\002' ':'
```

<HostName> は、情報を取得するサーバーのホスト名、 <Tag> は、指定されたサーバー上の LifeKeeper の SAP HANA リソースのタグ名（例：HANA-SPS_HDB00）です。

現在保存されているセカンダリーレプリケーションモードは、このコマンドの出力の3番目のフィールドです。

```
SPS:HDB00:node2/sync:SiteA:node2/logreplay
```

2 ノードを超える構成の場合、出力は次のようになります。

```
SPS:HDB00:node2/sync node3/async:SiteA:node2/logreplay node3/logreplay
```

## オペレーションモードの変更

> ✻ 注意：オペレーションモードを logreplay または logreplay_readaccess から delta_datashipping に変更するには、レプリケーションの再開時にプライマリーサイトからセカンダリーサイトへの完全なデータシッピングが必要になります。これらのオペレーションモードの変更の1つが検出されると、SAP HANA によってこの完全なデータシッピングが自動的に実行されます。

SAP HANA システムレプリケーションでオペレーションモードを変更するには、通常関連するセカンダリーサイトでデータベースを停止し、新しいオペレーションモードでデータベースを再登録し、セカンダリーサイトでデータベースを再起動する必要があります。 このプロセス中に LifeKeeper がセカンダリーサイトのデータベースを自動的に再起動するのを防ぐために、データベースが新しいオペレーションモードで正常に再登録されるまで、HANA リソースの監視を一時停止することをお勧めします。

> ✻ 注意：このプロセスは、オペレーションモードを変更する必要がある、各セカンダリーノードに適用する必要があります。

1. リソースが現在アクティブ（ISP）になっているサーバーで対応する *nomonitor* フラグを作成して、HANAリソースの監視を一時停止します。

   ```
   /opt/LifeKeeper/bin/flg_create -f 'nomonitor_<Tag>'
   ```

   <Tag> は、そのサーバー上のLifeKeeperのHANAリソースタグです（例：HANA-SPS_HDB00）。このプロセスが完了したら、次のコマンドで *nomonitor* フラグを削除して、リソースの監視を再開することを忘れないようにしてください。

   ```
   /opt/LifeKeeper/bin/flg_remove -f 'nomonitor_<Tag>'
   ```

   **警告：** メンテナンス操作の実行後にこのフラグを削除しないと、SAP HANAデータベースの障害がLifeKeeperによって検出されなくなります。

2. 次のコマンドを実行して、バックアップサーバーでSAP HANAデータベースが停止していることを確認します。

```
su - <sid>adm -c "sapcontrol -nr <HDB Inst#> -function StopSystem HDB"
```

<HDB Inst#> は、SAP HANAデータベースインスタンスのインスタンス番号です（例：HDB00という名前のインスタンスの場合、インスタンス番号は00です）。

3. 使用したいレプリケーションモードとオペレーションモードを指定してバックアップサーバーで次のコマンドを実行し、セカンダリーレプリケーションサイトとして再登録します。

```
su - <sid>adm -c "hdbnsutil -sr_register --name=<SecondarySiteName> --
remoteHost=<PrimaryHost> --remoteInstance=<PrimaryInst#> --
replicationMode=[sync|syncmem|async] --
operationMode=[delta_datashipping|logreplay|logreplay_readaccess]"
```

<SecondarySiteName> は、セカンダリーレプリケーションサイトを識別するためにSAP HANAシステムレプリケーションで使用されるエイリアス、 <PrimaryHost> は、プライマリーレプリケーションサイトとして現在登録されているサーバーのホスト名、 <PrimaryInst#> はプライマリーレプリケーションサイトのHDBインスタンスのインスタンス番号です。 **注記：** 新しいレプリケーションモードまたはオペレーションモードで再登録する前に、 "hdbnsutil -sr_unregister" コマンドでセカンダリーサイトを登録解除する必要はありません。

> ✻ 注意：マルチターゲットシステムレプリケーションでは、logreplay および logreplay_readaccess モードのみがサポートされます。

4. バックアップサーバーが新しいオペレーションモードでセカンダリーレプリケーションサイトとして正常に再登録されたことを確認するには、バックアップサーバーで次のコマンドを実行し、"is secondary/consumer system" が「true」であり、パラメーター"operation mode" が新しいオペレーションモードに更新されていることを確認します。

```
su - <sid>adm -c "hdbnsutil -sr_state"
```

5. バックアップサーバーがセカンダリーレプリケーションサイトとして正常に登録されたら、バックアップサーバーで次のコマンドを実行してデータベースを起動します。

```
su - <sid>adm -c "sapcontrol -nr <HDB Inst#> -function StartSystem HDB"
```

<HDB Inst#> は、SAP HANAデータベースインスタンスのインスタンス番号です。

6. プロセスが完了したら、ステップ1で作成した *nomonitor* フラグを削除して、LifeKeeperによるHANAリソースの監視を再開します。

```
/opt/LifeKeeper/bin/flg_remove -f 'nomonitor_<Tag>'
```

**警告：** このフラグを削除しないと、SAP HANAデータベースの障害がLifeKeeperによって検出されなくなります。

次のquickCheck間隔（デフォルトは2分）後、LifeKeeperは変更を自動的に検出し、同等のすべてのSAP

HANAリソースの情報フィールドを更新して新しいオペレーションモードを含めます。これは、次のコマンドを使用してHANAリソースの情報フィールドを調べることで確認できます。

```
/opt/LifeKeeper/bin/ins_list -d <HostName> -t <Tag> -f: | cut -d: -f6 | tr
'\002' ':'
```

<HostName> は情報を取得するサーバーのホスト名で、 <Tag> は指定されたサーバー上のLifeKeeperのSAP HANAリソースのタグ名です。（例：HANA-SPS_HDB00）

現在保存されているセカンダリーレプリケーションモードは、このコマンドの出力の 5 番目のフィールドです。

```
SPS:HDB00:node2/sync:SiteA:node2/logreplay
```

2 ノードを超える構成の場合、出力は次のようになります。

```
SPS:HDB00:node2/sync node3/async:SiteA:node2/logreplay node3/logreplay
```

# 6.18.7.2. Takeover with Handshake

## SAP HANA データベースの Takeover with Handshake

“Takeover with handshake” は、SAP HANA 2.0 SPS04 以降に追加された機能で、新しいデータベースホスト上でSAP HANA システムレプリケーションの引き継ぎを行う前にプライマリーデータベースを (完全に停止するのではなく)一時停止することで、スイッチオーバー時のプライマリーデータベースのダウンタイムを抑えます。

“Takeover with handshake” が開始されると、SAP HANA Recovery Kit は以下の処理を実行します。:

- データベースインスタンスを、旧プライマリーノードで起動したたままにする。
- 新しいプライマリーノード(旧セカンダリーノード)でSAP HANA システムレプリケーションの “Takeover with handshake” を実行する 。新しいプリマリーシステムレプリケーションサイトを登録する前に、旧プライマリーノードのデータベースを一時停止のステータスにする。
- 新しいセカンダリーノード (旧プライマリーノード) の一時停止されたデータベースを停止し、セカンダリーシステムレプリケーションサイトとして再登録する。
- データベースインスタンスを新しいセカンダリーノードで起動する。

## Takeover with Handshake の実行

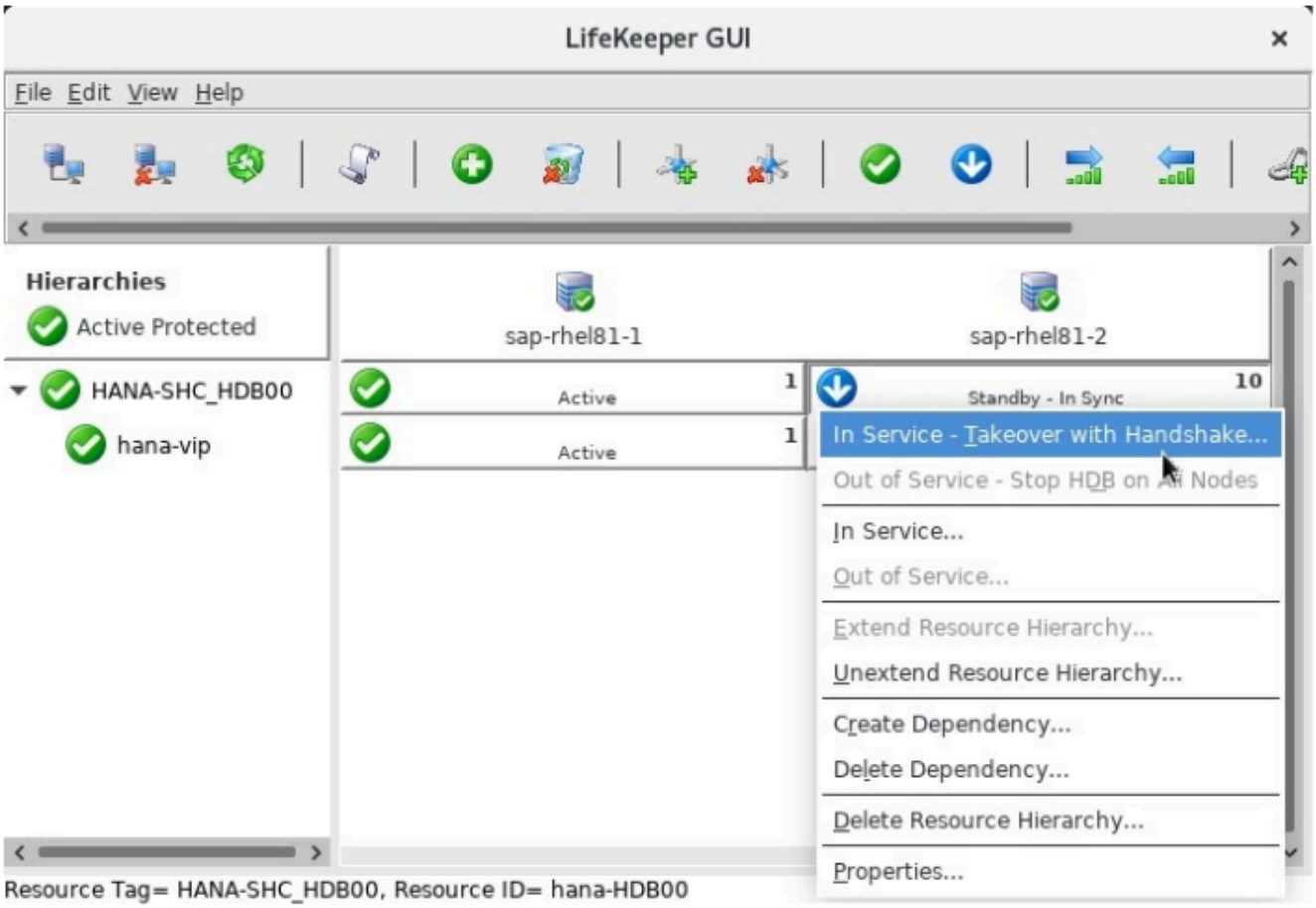
### 要件

Takeover with handshake を実行する前に、以下の要件を満たしている必要があります。:

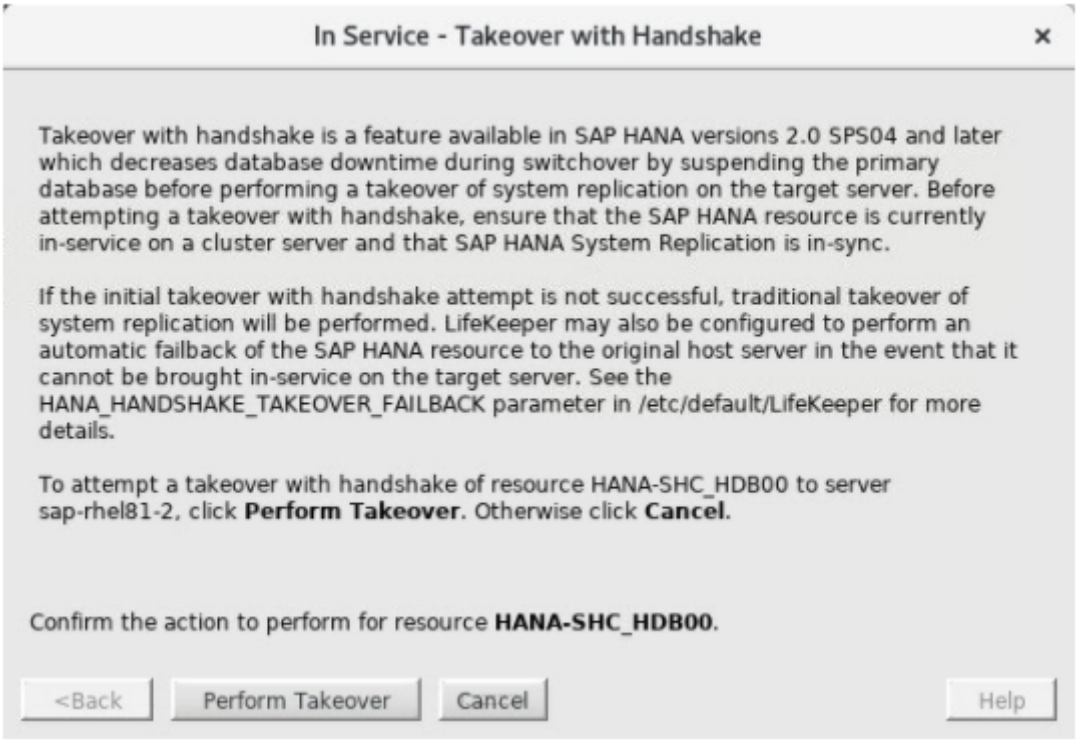
1. 各サーバーに SAP HANA 2.0 SPS04 以降がインストールされている。
2. クラスター内のサーバーで保護対象の SAP HANA データベースが In Service になっている。
3. SAP HANA システムレプリケーションが同期中になっている。

### LifeKeeper GUI でTakeover with Handshake を実行する

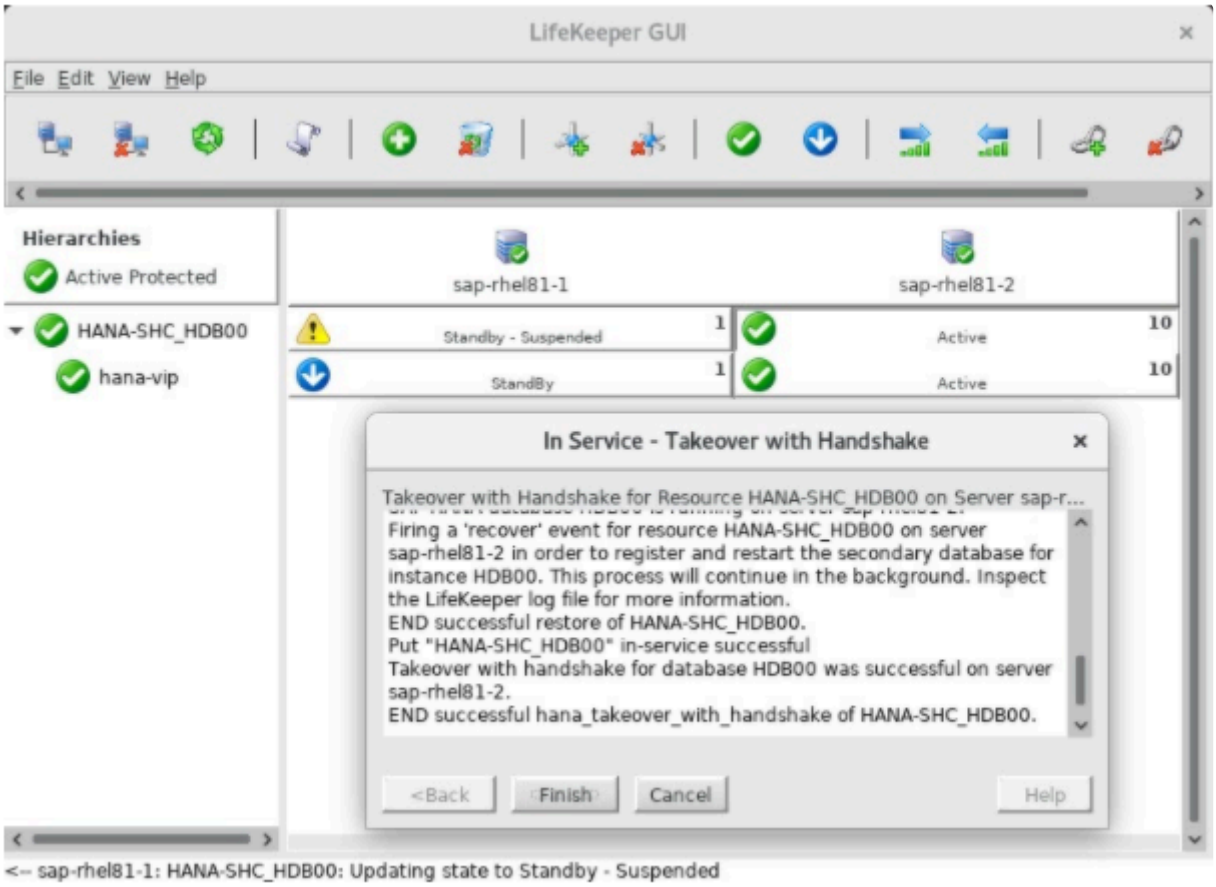
LifeKeeper GUI でTakeover with Handshake を実行するためには, スタンバイサーバーのリソース上で右クリックし、 “In Service – Takeover with Handshake…” を選択します。




Takeover with handshake の機能が説明された確認用ダイアログが表示されます。:




**Perform Takeover** をクリックしてTakeover with handshake を開始します。この処理が完了すると 各ノードの SAP HANA リソースのステータスが **Standby – Suspended** と **Active** へと切り替わります。




SAP HANA リソースがターゲットサーバーで In Service になったら、'remoteregisterdb' のイベントがバックグラウンドで自動的に実行され、セカンダリーデータベースインスタンスを停止、登録、再起動します。SAP HANA リソースを In Service にする間に、このセカンダリーデータベースの再起動のプロセスをフォアグラウンドで行いたい場合は、 /etc/default/LifeKeeper で、`HANA_REGISTER_SECONDARY_DURING_RESTORE=true` と設定してください。

'remoteregisterdb' のイベントが正常にセカンダリーデータベースインスタンスを再起動できたら、リソースステータスが **Standby – HDB Running** へ切り替わります。LifeKeeper がその後発生するquickCheck サイクル中にHRS 監視プロセスを開始し、HSR が In sync であることを判断したあと、リソースステータスは、 **Standby – In Sync** に切り替わります。この時点でSAP HANA データベースは高可用性を提供します。

## コマンドラインで Takeover with Handshake を実行する

以下のどちらかのコマンドを実行することによりTakeover with handshake をコマンドラインで実行することも可能です。:

- `/opt/LifeKeeper/bin/hana_takeover_with_handshake -t <HANA Tag> -s <Target Server> [-b]`
- `lkcli resource config hana –tag <HANA Tag> –takeover_with_handshake <Target Server>`

1番目のコマンドの -b パラメーターは、ターゲットサーバーでどの SAP HANA リソース階層を In Service にするかをコントロールします。-b オプションがない場合は、階層全体 (全ての親リソースと共有された

依存関係をもつリソース)が In Service になります。 -b オプションを付けると、指定された SAP HANA リソースとその依存関係が In Service になります。

### フェイルバックの動作を制御する

デフォルトでは SAP HANA リソースが、Takeover with handshake の実行中に In Service にならない場合はターゲットサーバーで **Out of Service – Failed (OSF)** の状態のままになり手動で In Service に戻すことが要求されます。このシナリオにおいて LifeKeeper は自動フェイルバックを行って、旧ホスト上で SAP HANA リソース階層を In Service に戻す設定をすることもできます。 自動フェイルバックは以下の設定を行うことで有効になります。/etc/default/LifeKeeper で `HANA_HANDSHAKE_TAKEOVER_FAILBACK=true` を設定してください。

## 一時停止したプライマリーデータベースを再開する

保護対象の SAP HANA データベースが、現在リソースが In Service になっているノードで一時停止される (例えば管理者が、LifeKeeper の外部で Takeover with handshake を手動で実施するためなどの理由で)と、 下記のようにリソースのステータスが、 **Active – Suspended** および **Standby – Primary** になります。

このステータスになっている間、すべての SAP HANA リソースの監視は一時停止され、下記のようなメッセージが記録され、問題が解決されるまでSAP HANA リソースが現在 In Service になっているシステムですべてのオープンになっているターミナルに対してブロードキャストされます。:

```
EMERG:hana:quickCheck:HANA-SHC_HDB00:136377:SAP HANA database HDB00
corresponding to resource HANA-SHC_HDB00 is currently suspended on server sap-
rhel81-1 due to actions performed outside of LifeKeeper. Please take the SAP
HANA resource out of service on server sap-rhel81-1 and bring it in-service on
the server where the database should be registered as primary master. Bringing
resource HANA-SHC_HDB00 back in-service on sap-rhel81-1 will resume the
suspended database. Resource monitoring for HANA-SHC_HDB00 will be suspended
until the issue is resolved.
```

必要に応じて以下のコマンドを一時停止されているサーバー上で実行することでデータベースを手動で再開することが可能です。:

```
su - <sid>adm -c "hdbnsutil -sr_resumeSuspendedPrimary"
```

このコマンドを使用してデータベースを再開した後、プライマリーとして登録されるスタンバイサーバー

で以下のコマンドを実行して手動で停止する必要があります。：

```
su – <sid>adm -c “sapcontrol -nr <InstNum> -function StopWait 600 5”
```

スタンバイサーバーを新しいプライマリーサーバーにする場合は、SAP HANA リソースをSAP HANA リソースのステータスが **Standby – Primary** になっているサーバーへ切り替えることで解決できます。

# 6.18.7.3. SAP HANA リソースのローカルリカバリーポリシーおよび TemporalRecoveryPolicy の設定

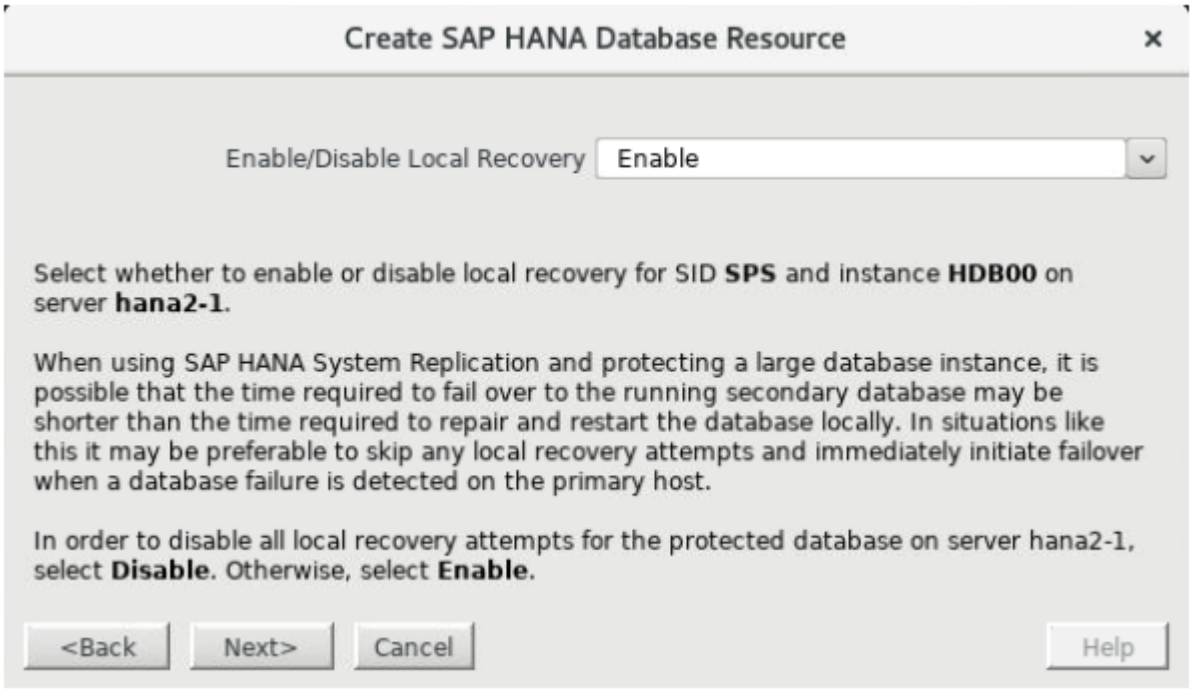
## ローカルリカバリーポリシー

HANA システムレプリケーション（HSR）を利用した高い可用性の SAP HANA デプロイメントでは、アクティブデータベースインスタンスからのデータが、スタンバイノード上で実行中のセカンダリーデータベースインスタンスに継続的にレプリケーションされます。この実行中のセカンダリーインスタンスはウォームスタンバイとして機能し、データベースの In-Service 中に起動するという時間のかかる要件を伴わずに、SAP HANA リソースをスタンバイノードにスイッチオーバー、フェイルオーバーする機能を提供します。
このため、特に大規模なデータベースインスタンスを保護する場合、障害が検出されたときにアクティブノードでデータベースを再起動するのではなく、直ちにスタンバイノードにフェイルオーバーした方が高速に処理できる場合があります。

この動作は、SAP HANA リソースに対して LifeKeeper のローカルリカバリーポリシーを設定することで実行できます。以下で、ローカルリカバリーポリシーの設定方法をいくつか説明します。

### リソース作成中のローカルリカバリーポリシーの設定

SAP HANA リソースの作成中にダイアログが表示され、リソースのローカルリカバリーを有効または無効にすることができます。リソースを作成するノード上でローカルリカバリーの試行を許可する場合は [Enable] を、そのノード上でデータベース障害が検出された場合にローカルリカバリーをすべてスキップし、リソース階層を直ちにスタンバイノードにフェイルオーバーする場合は [Disable] を選択します。

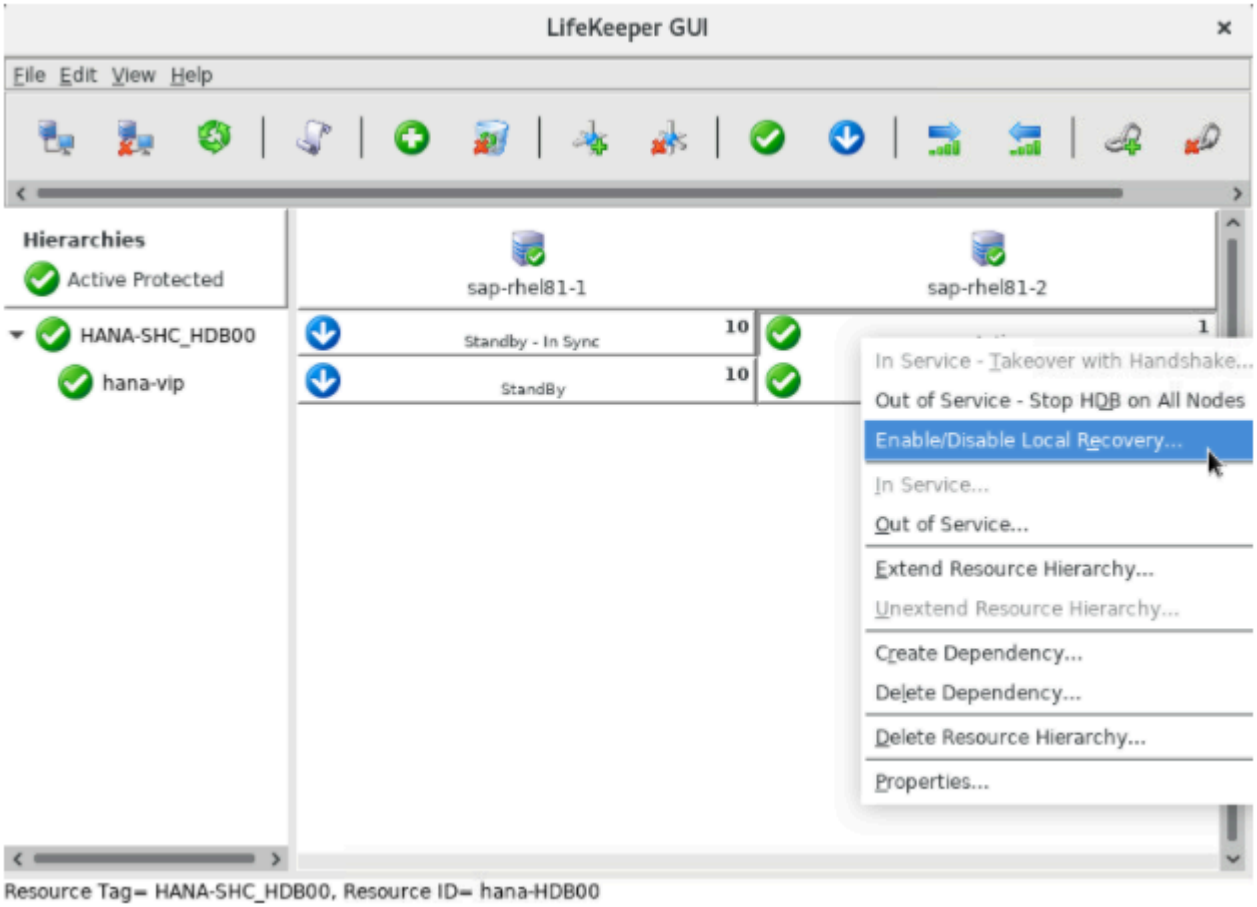


## リソース拡張中のローカルリカバリーポリシーの設定

SAP HANA リソースの拡張中にダイアログが表示され、リソースのローカルリカバリーを有効または無効にすることができます。リソースが拡張されるノード上でローカルリカバリーの試行を許可する場合は **[Enable]** を、そのノード上でデータベース障害が検出された場合にローカルリカバリーをすべてスキップし、リソース階層を直ちにスタンバイノードにフェイルオーバーする場合は **[Disable]** を選択します。

## LifeKeeper GUI を使用したローカルリカバリーポリシーの設定

LifeKeeper GUI で SAP HANA リソースを右クリックし、**[Enable/Disable Local Recovery...]** を選択すると、選択したノード上のリソースのローカルリカバリーを有効または無効にするためのダイアログが表示されます。




## LKCLI を使用したローカルリカバリーポリシーの設定

次の LKCLI コマンドを使用して、コマンドが実行されるノード上の SAP HANA リソースのローカルリカバリーを有効または無効にすることができます。コマンドの <HANA Tag> を、構成する SAP HANA リソースのタグ名に置き換えます。

```
# /opt/LifeKeeper/bin/lkcli resource config hana --tag <HANA Tag> --
set_local_recovery_policy <enable|disable>
```

## LKPolicy を使用したローカルリカバリーポリシーの設定

LKPolicy ユーティリティを使用して、ローカルリカバリーポリシーを作成できます。

次のコマンドを実行して、コマンドが実行されるノード上の特定のリソースのローカルリカバリーを無効にします。

```
# /opt/LifeKeeper/bin/lkpolicy --set-policy LocalRecovery --off tag=<Resource
Tag>
```

次のコマンドを実行して、コマンドが実行されるノード上の特定のリソースのローカルリカバリーを有効にします。

```
# /opt/LifeKeeper/bin/lkpolicy --set-policy LocalRecovery --on tag=<Resource
Tag>
```

> 注意：LKPolicy ユーティリティを初めて使用する場合は、まず LifeKeeper の認証情報ストアを使用してユーザーの認証情報を設定する必要があります。詳細については、 認証情報の設定 を参照してください。

# Temporal Recovery ポリシー

状況によっては、必要なデータベースプロセスがすべて正常に開始できたにもかかわらず、その後すぐにノード側の問題により障害が発生する場合があります。SAP HANA リソースに対してローカルリカバリーが有効になっている場合、このような状況では、quickCheck の失敗とローカルリカバリーの成功が延々と繰り返される可能性があります。そのため、短時間のうちに何度もローカルリカバリーを試みた場合、すべての試行が成功したとしても、ノードの問題が示唆されることがあります。

このように延々と続くローカルリカバリーサイクルを回避するために、 LKPolicy ユーティリティを使用して、各ノードにTemporal Recovery ポリシーを確立することができます。
Temporal Recovery ポリシーを使用するとY 分以内に X 回のローカルリカバリーの試行が発生した場合、LifeKeeper は障害の発生したノードからスタンバイノードにすべてのリソース階層を直ちにフェイルオーバーします（X と Y はポリシーによって設定されたパラメーターです）。

例として、次のコマンドをノードで実行することで、10分以内にそのノードで3回のローカルリカバリーを実行すると、フェイルオーバーを実行するトリガーとなる Temporal Recovery ポリシーを設定できます。

```
# /opt/LifeKeeper/bin/lkpolicy --set-policy TemporalRecovery --on
recoverylimit=3 period=10
```

次のコマンドを実行すると、既存のTemporal Recovery ポリシーを削除できます。

```
# /opt/LifeKeeper/bin/lkpolicy --remove-policy TemporalRecovery
```

# ローカルリカバリー機能の強化

https://fast.wistia.net/embed/iframe/l8v21odnmf

# 6.18.7.4. マルチターゲットレプリケーション環境でのログ保持の管理

## ログ保持設定

SAP HANAのドキュメントによると、プライマリーシステムが複数のセカンダリーシステムに複製されている場合は、 **force log retention** および **log retention propagation** を使用して、最適化された再同期を実現し、引き継ぎまたはその他の切断状況の後に完全なデータシッピングが起こることを回避する必要があります。

**SAP HANA** ヘルプドキュメントから

> If enable_log_retention = force_on_takeover is configured, the log will be retained during replication for all direct secondaries until a takeover is executed. During takeover, the parameter is set to force. This means the log will be retained independently of any secondary system. The configuration must be done manually (for example, by the administrator or using setup scripts) because the SAP HANA system doesn't know when the system landscape has been completely reconfigured.

※以下参考情報として翻訳を掲載します。英語の原文と合わせてご参照ください。

> enable_log_retention = force_on_takeover が構成されている場合、テイクオーバーが実行されるまで、すべてのダイレクトセカンダリーのレプリケーション中にログが保持されます。 テイクオーバーの間、このパラメーターは force に設定されます。 これは、ログがどのセカンダリーシステムからも独立して保持されることを意味します。SAP HANA システムはシステムのランドスケープが完全に再構成されたことを認識しないため、設定は手動で行う必要があります(たとえば、管理者による設定やセットアップスクリプトの使用)。

> **!** LifeKeeper for Linux はこれらのパラメーターを自動的に管理しないため、システムランドスケープ全体の引き継ぎまたは再登録の成功後に、管理者が手動で更新する必要があります。 enable_log_retention および propagate_log_retention のパラメーターの詳細については、SAP HANA システムレプリケーションガイドを参照してください。

# 6.18.8. SAP HANAトラブルシューティング

メッセージカタログ は、LifeKeeper for Linux の使用中に表示される可能性のあるすべてのメッセージのリストを掲載しています。また、エラーの原因の説明や、エラー状態を解決するために必要な処置も提供しています。このリストで、受信したエラーコードを検索することができます。または、SAP HANA Recovery Kitの使用中に表示される可能性のあるすべてのメッセージのリストが記載された SAP HANA Recovery Kit メッセージカタログ を直接確認しても構いません。

SAP HANA Recovery Kitを使用する際の既知の問題や制限事項については、 SAP HANA – 既知の問題と制限 を参照してください。

# 6.18.8.1. SAP HANA 階層の restore 失敗

## SAP HANAリソースの restore が、データの不整合や古いデータのために失敗する

SAP HANA Recovery Kit は、一貫性のないデータまたは古いデータを持つ可能性のあるシステムで SAP HANA リソース (データベース) を in-service にすることを許可しません。 データの一貫性を維持するために、次のフラグが使用されます。

| フラグ名 | 説明 |
|---|---|
| !HANA_DATA_OUT_OF_SYNC_<tag> | このフラグは、システム上のデータが最新でないか、または矛盾している可能性があることを判断するために、使用されます。 このフラグは、リソースがシステム上で in-service になると、すべてのスタンバイシステムで作成されます。 このフラグは、レプリケーションが確立され、そのシステムの HSR ステータスが “ACTIVE” になると削除されます。 |
| !HANA_LAST_OWNER_<tag> | このフラグは、データが in-service (Primary) であり、同期されていない可能性があるデータの場所を示すために、使用されます。 このフラグは、 in-service のリソースを持つシステムで作成されます。 システム上でリソースが停止 (out-of-service) され、すべてのスタンバイシステムが同期している (HSR ステータスが “ACTIVE” ) 場合、フラグは削除されます。 |
| !HANA_DATA_CONSISTENCY_UNKNOWN_<tag> | このフラグは、システム上のデータが最新かどうかを判断できない場合に、使用されます。このフラグは、いずれかのシステムが稼働中か、またはいずれかのシステムが LAST_OWNER フラグを持っている場合、起動時にシステムに作成されます。 LifeKeeper の起動時に、他のすべてのシステムがアクセス可能であり、 in-service となっているシステムがなく、LAST_OWNER フラグが設定されているシステムがない場合、すべてのシステムで DATA_CONSISTENCY_UNKNOWN フラグが削除されます。このフラグは、スタンバイシステムへのレプリケーションが確立されるため、復元中に削除されます。 |
| !HANA_STARTUP_DONE_<tag> | このフラグは、スタートアップルーチンと in-service 操作との間の競合を回避するために使用されます。 このフラグは、起動スクリプトが完了して、 LifeKeeper の初期化が行われている間に in-service を許可するときに作成されます。 in-service 操作が完了すると、このフラグは削除されます。 |

LifeKeeper の起動時に、SAP HANA Recovery Kit はクラスター内の各システムのリソースステータスをチ

ェックします。

- いずれかのシステムで SAP HANA リソースが in-service の場合、起動すると、稼働中のシステムの quickCheck プロセスがデータを同期します。
- クラスター内のいずれかのシステムが起動時に応答しない場合、SAP HANA Recovery Kit は、すべてのシステムが使用可能になるまで待機してから、SAP HANA リソースを in-service にする前にデータの一貫性を判断します。
  - すべてのシステムがクラスターに再参加すると、LifeKeeper は自動的に起動するように設定された（AUTORES_ISP が設定されている）システムでリソースを in-service にします。複数のサーバーが自動的に restore するように設定されている場合、システム管理者はリソースを強制的に in-service にするシステムを決定する必要があります。
  - すべてのシステムが適したタイミングでクラスターに再参加できない場合、管理者は SAP HANA リソースを強制的に in-service にすることを選択できます。 WARNING: これはダウンしたシステムに非同期の変更がないことを管理者が認識している場合、またはそれらの変更を意図的に失う場合にのみ行う必要があります。

> 注記: SAP HANA リソースが in-service になり、データがクラスター内の他のシステムと同期されると、一部のシステムがダウンしていても、スイッチオーバーとフェイルオーバーを続行できます。

- 同期されていないデータを持つシステムが 2 つ以上ある場合、システム管理者は、最新のデータを持つシステムを決定し、リソースを強制的に稼働させる必要があります。 これによりすべてのシステムが、強制的に in-service となったシステム上のデータと同期されます。

# 関連メッセージ

## “last owner” に同期されていないデータがあるときに HANA を開始する

```
ERROR:hana:restore:HANA-SPS_HDB00:136280:The resource HANA-SPS_HDB00 protectin
g SAP HANA database HDB00 has the last owner flag on node1 indicating there ar
e changes on node1 that have not replicated to all systems.  The resource shou
ld be brought in-service on node1 to allow the data to replicate to all system
s.
```

この状況は、すべてのデータがクラスター内のすべてのシステムと同期されていないときに、node1 上の SAP HANA リソースが out-of-service になった場合に発生する可能性があります。 システム node2 は、リソースが out-of-service となった時に起動され、そのデータが最新かどうかを判断できませんでした。 node1 でリソースを in-service にして、node2 とデータを同期する必要があります。

## SAP HANA リソースが in-service となっていないときに複数のサーバーが再起動する

```
ERROR:hana:restore:HANA-SPS_HDB00:136282:The SAP HANA database HDB00 consisten
```

```
cy can not be determined on node2. The resource HANA-SPS_HDB00 can not be brou
ght in-service until all servers are available to determine consistency of th
e data. To protect the data, LifeKeeper will not restore HANA-SPS_HDB00 on nod
e2. To force the resource in-service use "lkcli hana force --sys node2 --tag H
ANA-SPS_HDB00" or the GUI option "Force In Service".
```

この状況は、node2 が起動し、SAP HANA リソースがどのシステムでも in-service の状態ではなく、クラスター内に応答していないシステムが 1 つ以上ある場合に発生する可能性があります。起動サーバーは、応答しないシステムに非同期の内容があるかどうかを判断できません。SAP HANA Recovery Kit は、すべてのシステムが利用可能になるまで、SAP HANA リソースが node2 で in-service となることを許可しません。Recovery Kit は、システム上のデータベースが一貫しているかどうか、または最新のデータを持っているかどうかを判断できない場合、フラグ '!HANA_DATA_CONSISTENCY_UNKNOWN_<Tag>' を使用します。 一貫性が確認されると、フラグは削除されます。

リソースが起動時に自動的に開始するように設定されれいて(リソースインスタンスで AUTORES_ISP が設定されている)、起動時にこのエラーが発生した場合、クラスター内のすべてのシステムがオンラインになると、restore が自動的に行われます。

## SAP HANA リソースを強制的に in-service にする

エラーメッセージに記載された lkcli コマンドを使用するか、GUI オプション “Force In Service” を使用してリソースを強制的にオンラインにすると、次のメッセージが表示されます。

```
ERROR:hana:restore:HANA-SPS_HDB00:136285:The SAP HANA database HDB00 consisten
cy can not be determined on node2.
```

```
INFO:hana:restore:HANA-SPS_HDB00:136286:The resource HANA-SPS_HDB00 protectin
g SAP HANA database HDB00 is being forced in-service.
```

### データが同期されていない複数のサービス

```
ERROR:hana:restore:HANA-SPS_HDB00:136281:The resource HANA-SPS_HDB00 protectin
g SAP HANA database HDB00 has the last owner flag on node1, node2 indicating t
here are changes on each system that were not replicated to all systems.  The
resource should be brought in-service on the system with the most up-to-date d
ata to allow the data to replicate to all systems.
```

この状況は、node1 のリソースが in-service の状態で、障害が発生し再起動しない場合に発生する可能性があります。リソースは node2 にフェイルオーバーし、リソースを in-service にしてしばらく動作しますが、その後失敗します。この時点で、両方のサーバーが SAP HANA リソースの “last owners” (最後の所有者) であり、変更が同期されていない可能性があります。 node1 と node2 が修復されて再起動されると、LifeKeeper は SAP HANA リソースを in-service にすることを許可しません。システム管理者は、最良のデータを使用してシステム上でリソースを強制的に in-service にさせる必要があります。 '!HANA_LAST_OWNER_<Tag>' フラグを使用して、データベースがデータベースのソースとして最後に

使用された場所を特定します。 この場合、両方のシステムにこのフラグがあります。 すべてのサーバーでデータが同期しているときにリソースが停止して out-of-service になると、 “last owners” フラグが削除されます。

## リソースが in-service にならないようにする

```
ERROR:hana:restore:HANA-SPS_HDB00:136284:To protect the data, LifeKeeper will not restore HANA-SPS_HDB00 on node1. To force the resource in-service use "lkcli hana force --sys node1 --tag HANA-SPS_HDB00" or the GUI option "Force In Service".
```

このメッセージは、LifeKeeper がリソースを in-service にすることが安全でないと判断した場合に表示されます。 以前のエラーメッセージは安全でない問題を示します。 メッセージに示されている指示に従って、必要に応じてリソースを強制的に in-service にしてください。

## リソースを強制的に in-service にする

リソースを強制的に in-service にすると、次のメッセージが表示されます。

```
INFO:hana:restore:HANA-SPS_HDB00:136286:The resource HANA-SPS_HDB00 protecting SAP HANA database HDB00 is being forced in-service.
```

# 6.18.8.2. スプリットブレインシナリオの解決

「スプリットブレイン」シナリオは、SAP HANAデータベースが実行され、複数のクラスターノードでSAP HANAのプライマリレプリケーションサイトとして構成されている場合に発生します。スプリットブレインが発生した場合、LifeKeeperは、データベース管理者が問題を手動で解決するまで、すべてのHANAデータベースの監視を一時停止します。

SAP HANAリソース階層で発生する可能性のある一般的なスプリットブレインシナリオは、2種類あります。

- **LifeKeeper HANAリソースのスプリットブレイン**: HANAリソースは、複数のクラスターノード上のLifeKeeperでアクティブ（ISP）です。この状態は通常、クラスターノード間のコミュニケーションパスに影響を与える一時的なネットワーク障害が原因です。

- **SAP HANAシステムレプリケーションのスプリットブレイン**: HANAリソースは、LifeKeeperのプライマリノードでアクティブ（ISP）、バックアップノードでスタンバイ（OSU）ですが、データベースは両方のノードでプライマリレプリケーションサイトとして実行および登録されています。この状態は通常、フェイルオーバー中に以前のプライマリノードでデータベースを停止できなかった、データベースのAutostartが有効になっている、またはLifeKeeperの外部のセカンダリレプリケーションサイトでデータベース管理者が "hdbnsutil -sr_takeover" を手動で実行したことが原因です。

各タイプのスプリットブレインシナリオを解決するための推奨事項を以下に示します。

## LifeKeeper HANAリソースのスプリットブレインの解決

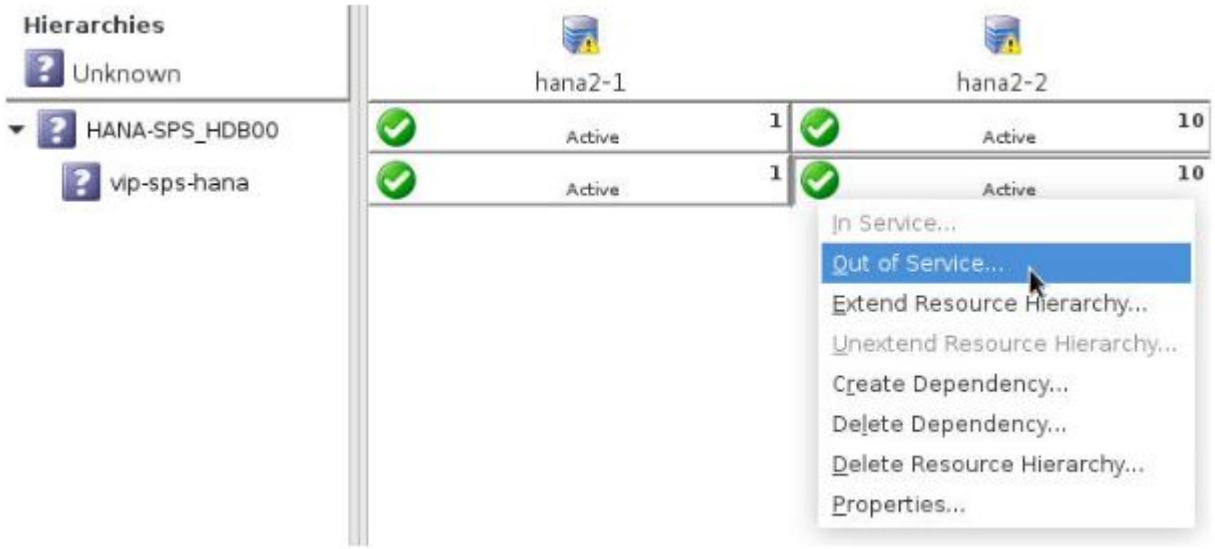
このスプリットブレインシナリオでは以下のようなメッセージがログに出力され、問題が解決されるまでquickCheckの間隔（デフォルトでは2分）ごとに、開いているすべてのコンソールに表示されます。

```
EMERG:hana:quickCheck:HANA-SPS_HDB00:136363:WARNING: A temporary communication
failure has occurred between servers hana2-1 and hana2-2. Manual intervention
is required in order to minimize the risk of data loss. To resolve this
situation, please take one of the following resource hierarchies out of
service: HANA-SPS_HDB00 on hana2-1 or HANA-SPS_HDB00 on hana2-2. The server
that the resource hierarchy is taken out of service on will become the
secondary SAP HANA System Replication site.
```

**推奨される解決方法：**

1. 各クラスターノードでデータベースを調査し、どのインスタンスに最新のデータまたは関連データが含まれているかを確認します。この確認は、データに精通している適任なデータベース管理者が行う必要があります。

2. 保持する必要があるデータを含むノードのHANAリソースはLifeKeeperでアクティブ（ISP）のままであり、セカンダリレプリケーションサイトとして再登録されるノードのHANAリソース階層は、LifeKeeperで完全にOut of Serviceになります。階層のサービスを停止するノード上のHANAリソース階層の各リーフリソースを右クリックし、 **[Out of Service ...]** をクリックします。




> このステップでは、SAP HANAリソース階層全体（ある場合は仮想IPリソースを含む）を、セカンダリレプリケーションサイトとして再登録されるノードでOut of Serviceにすることが重要です。

3. SAP HANAリソース階層のサービスがOut of Serviceになると、LifeKeeperは次のquickCheck間隔（デフォルトでは2分）の間にセカンダリノードをセカンダリレプリケーションサイトとして再登録します。レプリケーションが再開されると、アクティブノードに存在しないスタンバイノードのデータは失われます。スタンバイノードがセカンダリレプリケーションサイトとして再登録されると、SAP HANA階層は高い可用性が保たれた状態に戻ります。

## SAP HANAシステムレプリケーションのスプリットブレインの解決

このスプリットブレインシナリオでは、次のようなメッセージがログに出力され、問題が解決されるま

で、quickCheckの間隔（デフォルトでは2分）ごとに、開いているすべてのコンソールに表示されます。

```
EMERG:hana:quickCheck:HANA-SPS_HDB00:136234:WARNING: SAP HANA database HDB00
is running and registered as primary master on the following servers: hana2-2,
hana2-1. Manual intervention is required in order to minimize the risk of data
loss. To resolve this situation, please stop database HDB00 on the standby
server by running the command 'su - spsadm -c "sapcontrol -nr 00 -function
StopWait 600 5"' on that server, allow LifeKeeper to register the standby
server as a secondary replication site, then use LifeKeeper to bring resource
HANA-SPS_HDB00 in-service on the intended primary replication site.
```

**推奨される解決方法：**

1. 各クラスターノードでデータベースを調査して、スタンバイノードにアクティブノードには存在しない重要なデータが存在するかどうかを確認します。スプリットブレイン状態の間に重要なデータがスタンバイノードのデータベースにコミットされた場合、データをアクティブノードに手動でコピーする必要があります。これは、データに精通している適任なデータベース管理者が行う必要があります。

2. 不足しているデータがスタンバイノード上のデータベースからアクティブノードにコピーされたら、LifeKeeper警告メッセージに示されているコマンドを実行して、スタンバイノード上のデータベースを停止します。

   ```
   su - <sid>adm -c "sapcontrol -nr <Inst#> -function StopWait 600 5"
   ```

   <sid> はHANAインストール時の小文字のSAPシステムIDで、 <Inst#> はHDBインスタンスのインスタンス番号です（例：インスタンスHDB00のインスタンス番号は00）。

3. データベースが正常に停止したら、LifeKeeperは次のquickCheck間隔（デフォルトでは2分）の間にスタンバイノードをセカンダリレプリケーションサイトとして再登録します。レプリケーションが再開されると、アクティブノードに存在しないスタンバイノードのデータは失われます。スタンバイノードがセカンダリレプリケーションサイトとして再登録されると、SAP HANA階層は高い可用性が保たれた状態に戻り、クラスター内の任意のサーバーでIn Service になります。

# SAP HANA マルチターゲットスプリットブレインリカバリー

マルチターゲット構成でスプリットブレインが発生すると、HANA リソースが稼働中の 1 台のサーバーもスタンバイターゲットに接続され、同期される可能性があります。 スタンバイターゲットに接続され、同期しているサーバーがサービスを継続するように選択されている場合、LifeKeeper はサービスを停止したノードに自動的に接続して同期します。 これは推奨されるリカバリー方法です。

ただし、稼働状態を維持するように選択されたノードがスタンバイターゲットに接続されていない場合、LifeKeeper がスタンバイターゲット上のデータベースをリモート登録しようとすると失敗する可能性があります。 その場合、スタンバイターゲットではリモートレプリケーションサイトの登録を手動で解除することが必要になる場合があります。 これにより、LifeKeeper はデータベースを登録できるようになりますが、SAP HANA はターゲットへの全同期を開始する可能性があります。

次のコマンドは、HANA 接続を表示します。

```
 su - spsadm -c "python /usr/sap/SPS/HDB00/exe/python_support/systemReplicatio
nStatus.py --sapcontrol=1"
```

hana2-2 がスタンバイターゲットに接続されていない場合、結果は次のようになります。

```
hana2-2# su - spsadm -c "python /usr/sap/SPS/HDB00/exe/python_support/systemRe
plicationStatus.py --sapcontrol=1"
SAPCONTROL-OK: <begin>
site/2/REPLICATION_MODE=PRIMARY
site/2/SITE_NAME=SiteA
local_site_id=2
SAPCONTROL-OK: <end>
hana2-2#
```

hana2-1 が hana2-3 に接続されている場合

```
hana2-1# su - spsadm -c "python /usr/sap/SPS/HDB00/exe/python_support/systemRe
plicationStatus.py --sapcontrol=1"
SAPCONTROL-OK: <begin>
service/ip-12-0-0-237/30001/DATABASE=SYSTEMDB
service/ip-12-0-0-237/30001/HOST=hana2-1
service/ip-12-0-0-237/30001/PORT=30001
service/ip-12-0-0-237/30001/SERVICE_NAME=nameserver
service/ip-12-0-0-237/30001/VOLUME_ID=1
service/ip-12-0-0-237/30001/SITE_ID=1
 :
 :
site/2/SITE_NAME=SiteD
site/2/SOURCE_SITE_ID=1
site/2/REPLICATION_MODE=ASYNC
site/2/REPLICATION_STATUS=UNKNOWN
overall_replication_status=UNKNOWN
site/1/REPLICATION_MODE=PRIMARY
site/1/SITE_NAME=SiteC
local_site_id=1
site/2/SECONDARY_FULLY_SYNCED=False
overall_in_sync_status=SYSTEM_NOT_IN_SYNC
SAPCONTROL-OK: <end>
hana2-1#
```

```
hana2-3# su - spsadm -c "python /usr/sap/SPS/HDB00/exe/python_support/systemRe
plicationStatus.py --sapcontrol=1"
SAPCONTROL-OK: <begin>
site/2/REPLICATION_MODE=ASYNC
```

```
site/2/SITE_NAME=SiteD
site/2/SOURCE_SITE_ID=1
site/2/PRIMARY_MASTERS=hana2-1
local_site_id=2
SAPCONTROL-OK: <end>
hana2-3#
```

上記のように hana2-2 がサービス停止になった場合、LifeKeeper はデータベースを登録できるようになり、部分的な再同期が発生します。 代わりに hana2-1 がサービス停止になった場合、hana2-2 は hana2-1 を登録できますが、hana2-3 を登録できない可能性があり、hana2-3 の登録を解除するには手動介入が必要になります。

GUI には、hana2-3 のステータスが “Standby – Unknown HSR Status” または “Standby – HSR Error” として表示され、上記の *ReplicationStatus.py* 出力には hana 2-2 の代わりに引き続き “site/2/PRIMARY_MASTERS=hana2-1” が表示されます。hana2-1 上の LifeKeeper が hana2-3 を登録できるようにするには、まずデータベースの登録を解除する必要があります。

```
hana2-3# su - spsadm -c "hdbnsutil -sr_unregister"
unregistering site ...
done.
hana2-3#
```

hana2-3 のリソースの状態は、 “Standy – HSR Disabled”、次に “Standby – Initializing”、 “Standby – Syncing”、最後に “Standby – In Sync”に移行します。 これにより、hana2-3 への全同期が発生する可能性があります。

# 6.18.8.3. v9.7.0 未満のクラスターのローリングアップグレード中の障害

## v9.7.0 未満のクラスターのローリングアップグレードの途中で、障害が発生した場合はどうなりますか?

既存の HANA リソース階層を持つユーザーは、LifeKeeper v9.7.0 未満のバージョンから v9.7.0 以降にアップデートする前に、必要に応じてロールバックできるように lkbackup アーカイブを生成する必要があります。 詳細については lkbackup を参照してください。

考えられるシナリオ

1. ユーザーは、LifeKeeper v9.7.0 未満のバージョンを実行している、既存の 2 ノード の SAP HANA クラスター環境 (ノード 1、ノード 2) を持っています。
2. ノード 2 で LifeKeeper 9.7.0 以降にアップグレードします。
3. ノード 2 で HANA リソース階層を in-service にします。
    a. HANA quickCheck スクリプトは、ノード 1 とノード 2 の両方でインフォフィールドを更新します。
4. ノード 2 の HANA 階層に障害が発生すると、LifeKeeper は階層を ノード 1 にフェイルオーバーします。
5. ノード 1 の HANA リソース (9.7.0 未満のバージョンの LifeKeeper を実行中) は、更新されたインフォフィールド構造からレプリケーション モードを取得できないため、ノード 2 へのレプリケーションを再確立できません。

## LifeKeeper v9.7.0 アップデートによってインフォフィールドが更新された後、v9.7.0 未満の LifeKeeper システムでリソースが in-service になった場合、どのようなエラーが発生しますか?

リソースは v9.7.0 未満のシステムで in-service になりますが、スタンバイシステムへのレプリケーションは再確立されません。 in-service システムの LifeKeeper ログには、次のエラーが記録されます。

- INFO:hana:remoteregisterdb:HANA-SPS_HDB00:136175:failed. trace file nameserver_node2.00000.000.trc may contain more error details.
- ERROR:hana:remoteregisterdb:HANA-SPS_HDB00:136258:Attempt to register server node2 as the secondary SAP HANA System Replication site for database HDB00 failed with exit code 127.
- NOTIFY:hana:remoteregisterdb:HANA-SPS_HDB00:136701:END failed remoteregisterdb of HANA-SPS_HDB00 with return value of 1.

## ユーザーは、インフォフィールドの 9.7.0 より前の形式にどのようにロールバックできますか?

推奨:

- v9.7.0 未満のバージョンの LifeKeeper の実行中に作成された各ノードの最新の lkbackup アーカイブからリソース階層を restore します。

詳細については lkbackup を参照してください。

# 6.19. SAP MaxDB Recovery Kit 管理ガイド

SAP MaxDB は、幅広い用途に使用できる SQL ベースの非常に強力なデータベースシステムです。これは拡張性が高く、プラットフォームに依存しません。また、完全なトランザクションサポートを提供します。データベースシステムは元来 SAP が所有していましたが、現在はオープンソースコミュニティにリリースされています。

SAP MaxDB Recovery Kit は、SPS for Linux 環境における SAP MaxDB データベースの耐障害性保護を提供します。

## ドキュメントの内容

本マニュアルでは以下のトピックを取り上げ、SAP MaxDB の階層の定義および管理をサポートします。

- SAP MaxDB Recovery Kit の要件 。SAP MaxDB Recovery Kit の適切なセットアップ、インストールおよび操作を行うために必要なハードウェアとソフトウェアの一覧です。
- 概要 。SAP MaxDB Recovery Kitの特徴と機能について説明しています。
- 設定上の考慮事項 。SAP MaxDB Recovery Kit のインストールおよび設定前に考慮すべき情報を記載しています。
- SPS で SAP MaxDB を設定する 。SAP MaxDB ソフトウェアおよび SAP ソフトウェアのインストールおよび設定の手順を説明しています。
- リソース設定作業 。LifeKeeper GUI を使用して階層で実行できるさまざまな機能（作成、拡張、削除および拡張解除）について説明しています。
- リソース階層の管理 。SAP MaxDB 階層の継続的な管理に関する重要な推奨事項を記載しています。
- トラブルシューティング 。SAP MaxDB Recovery Kit に関連するエラーメッセージを一覧にして説明しています。
- 付録：Raw/IO によるデータベースデバイスの作成 。SAP MaxDB Recovery Kitで使用するためのRawデバイスの設定に関する要件と手順を説明しています。

## SIOS Protection Suite/LifeKeeper ドキュメンテーション

SIOS Technology Corp. の Web サイトでは、以下の SPS LifeKeeper 製品のドキュメンテーションを提供しています。

- SPS for Linux リリースノート
- SPS for Linux テクニカルドキュメンテーション

- SPS for Linux インストールガイド
- オプションの Recovery Kit ドキュメンテーション

このドキュメンテーション、およびオプションの LifeKeeper Recovery Kit 関連のドキュメンテーションは、次に示す SIOS Technology Corp の Web サイトから入手できます。

http://jpdocs.us.sios.com/

また、LifeKeeper GUI 内の [Help] メニューからも参照できます。

## SAP MaxDB ドキュメンテーション

インストールガイド、ユーザーマニュアル、リファレンスマニュアルなどの SAP MaxDB ドキュメンテーションは、以下のサイトから入手できます。

http://maxdb.sap.com/documentation/

# 6.19.1. SAP MaxDB Recovery Kit の要件

SPS の構成は、SPS for Linux SAP MaxDB Recovery Kit をインストールする前に、以下の要件を満たしている必要があります。SPS ハードウェアとソフトウェアのインストールと設定に関する具体的な手順については、SPS for Linux インストールガイド を参照してください。

## ハードウェア要件

- **サーバー** - サーバーは、SPS for Linux インストールガイド およびSPS for Linux リリースノート に記載されている要件に従って設定してください。

- **IP ネットワークインターフェースカード** - 各サーバーには、イーサネット TCP/IP 対応のネットワークインターフェースカードが少なくとも1つ必要です。また、SPS クラスターには最低2つのコミュニケーションパスが必要です。ハートビートには、独立した2つのサブネットを使用する2つの LAN ベースの通信パスが推奨され、少なくとも1つはプライベートネットワークとして構成する必要があります。TCP と TTY ハートビートの組み合わせもサポートしています。

## ソフトウェア要件

- **TCP/IP** ソフトウェア – SPS 構成の各サーバーにTCP/IP ソフトウェアが必要です。

- **SAP MaxDB** ソフトウェア –「SIOS Protection Suite for Linux サポートマトリックス」に記載されている、サポートされているバージョンの SAP MaxDB ソフトウェアをインストールする必要があります。

  **注記：** クラスター内のすべてのサーバーに同じバージョンの SAP MaxDB ソフトウェアをインストールする必要があります。

- **SPS** ソフトウェア – 同じバージョンの SPS ソフトウェアをインストールし、クラスター内の各サーバーに同じバージョンの SPS ソフトウェアパッチを適用してください。

- **SPS for Linux SAP MaxDB Recovery Kit** – SAP MaxDB Recovery Kit は、SPS for Linux インストールイメージファイル（Steeleye-lkSAPDB）で提供しています。Red Hat Package Manager（RPM）を使用して、パッケージング、インストール、削除を行います。

# 6.19.2. SAP MaxDB Recovery Kit の概要

LifeKeeper for Linux SAP MaxDB Recovery Kit は、LifeKeeper 内の SAP MaxDB インスタンスを保護するための仕組みを提供します。SAP MaxDB ソフトウェア、SPS Core、SAP MaxDB Recovery Kit は、クラスター内の複数のサーバーにインストールされています。SAP MaxDB データベースインスタンスが SPS の保護対象になると、クライアントは LifeKeeper で保護された IP アドレスを使用してデータベースに接続します。また、LifeKeeper で保護された IP アドレスを別途作成し、親 SAP MaxDB リソースインスタンスと子 IP アドレスリソース間に手動で依存関係を作成する必要があります。SAP MaxDBサーバーで障害が発生した場合、LifeKeeper はまずローカルサーバーで復旧を試みます。ローカルでの復旧に失敗した場合、LifeKeeper はバックアップサーバーにフェイルオーバーします。

# 6.19.2.1. SAP MaxDB リソース階層

以下に、一般的な SAP MaxDB リソース階層の例を示します。

## Active/Standby 構成

| リソース | **SAP MaxDB** ソフトウェアコンポーネント |
|---|---|
| /maxdb | |
| > /sapdb/data (1 ~ 4) | SPSL データベースのデータベース データを保護します |
| > /sapdb/programs | SPSL データベースのデータベース プログラムを保護します |

## すべての構成 (Active/Active 構成を含む)

| リソース | **SAP MaxDB** ソフトウェア コンポーネント |
|---|---|
| > /sapdb/data (1 ~ 4) | SPSL データベースのデータベース データを保護します |
| > /sapdb/logs | SPSL データベースのデータベース ログ スペースを保護します |
| IP リソース (推奨) | クライアント接続に使用される切り替え可能な IP アドレスを保護します |

**注記:** リソースが作成されると、データは同じファイルシステム名で複製されます。

フェイルオーバーが発生した場合、LifeKeeper はファイルシステム、IP アドレス、およびデータベースリソース (すべてのリソース依存関係を含む) をバックアップノードで稼働させます。 クライアントは切断され、サーバーに再接続する必要があります。 コミットされていない SQL ステートメントは、再入力する必要があります。

# 6.19.3. SAP MaxDB の構成に関する考慮事項

本セクションには、SAP MaxDB Recovery Kit の設定および管理を開始する前に考慮すべき情報を記載しています。

> ✻ データベース構成ディレクトリは、手動またはその他の方法でレプリケートするか、プライマリーノードとセカンダリーノードの間で同期を保つ必要があります。 データベース構成情報が両方のノードで一致しない場合、プライマリーがダウンした場合にセカンダリーが引き継ぐことができなくなります。

Raw I/O の使用

DataKeeper でのミラーファイルシステムの使用

アクティブ/スタンバイの場合の考慮事項

アクティブ/スタンバイ設定の例

アクティブ/アクティブの場合の考慮事項

アクティブ/アクティブ設定の例

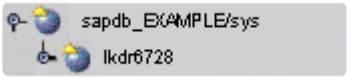
# 6.19.3.1. Raw I/O の使用

Raw デバイスで SAP MaxDB を使用する場合は、SPS のインストールイメージファイルから SPS Raw I/O Recovery Kit をインストールする必要があります。また、使用前に Raw I/O デバイスを適切に設定する必要があります。手順ついては、付録 を参照してください。

# 6.19.3.2. DataKeeper でのミラーファイルシステムの使用

SAP Max DB Recovery Kit は、デバイス領域としての DataKeeper の使用をサポートしています。さらに SAP MaxDB ソフトウェアは、ミラーリングされたファイルシステムにインストールすることができます。

例えば、SAP MaxDB リソースの依存ファイルシステムは次のようになりますが、これはシステムのデバイス領域とその依存関係のファイルシステム、DataKeeper リソースのミラーを示しています。




レプリケーション（SIOS DataKeeper）ファイルシステムのリソースはSAP MaxDBリソースを作成する前に作成する必要があります。

# 6.19.3.3. 内部向けロードバランサーの使用

SAP MaxDB Recovery Kit は Azure がサポートしているような内部向けロードバランサー (ILB) の使用をサポートします。ILB を有効にする場合は、LifeKeeper の設定ファイル (/etc/default/LifeKeeper) の MAXDB_ILB_ENABLED の値を 1 に設定してください。

ILBは、1つのサーバーのみで Xサーバーポートをアクティブにする必要があります。そのため、すべての SAP MaxDB リソースが同じ階層で、また同じノードでアクティブにすることになります。IndepDataPath および IndepProgPath が 共有ファイルシステム上にある場合、すべてのリソースは必然的に同一サーバー上に存在するはずです。IndepDataPath および IndepProgPath が共有ファイルシステムにない場合は、Generic Application リソースをリーフノードおよび/もしくはルートノードとして作成すると、同一サーバーで階層をアクティブにすることが可能です。

ILBの制限により、X サーバーはひとつのサーバー上でアクティブである必要があります。アクティブ/アクティブ構成は ILB でサポートされていません。

複数のSAP MaxDBリソースを単一の階層に配置するには、Generic Application リソースを最下位のリーフノードとして、および/または階層の最上位のルートノードとして別途作成してください。（Generic Application リソース階層の作成 に記述されている restore/remove スクリプトには、/bin/true を指定することができます。）

# 6.19.3.4. アクティブ/スタンバイの場合の考慮事項

アクティブ/スタンバイの構成では、バックアップサーバーは SAP MaxDB をアクティブに実行していませんが、プライマリサーバーの障害に備えて待機します。次のシナリオで、アクティブ/スタンバイ構成で SAP MaxDB リソースのインスタンスを保護する際に満たすべき具体的な要件を説明します。

## アクティブ/スタンバイのシナリオ

標準的なアクティブ/スタンバイ構成をシナリオ1およびシナリオ2で説明します。

## シナリオ1

SAP MaxDB の *IndepDataPath* 、 *IndepProgPath* および *DependPath* が **プライマリサーバー上の複数の共有ファイルシステムにインストールされている。**

- *IndepDataPath* 、 *IndepProgPath* および *DependPath* の3つのパスは、リソースインスタンスを保護するすべてのサーバー間で共有する必要があります。
- レジストリーファイル `/etc/opt/sdb` は、クラスター内の各サーバーに存在する必要があります。このファイルは、リソース拡張操作中にターゲットサーバーからアクセス可能でなければならないため、共有ストレージに置いてはいけません。
- データベースインスタンスのデータデバイス領域（`data devspaces`）、ログのデバイス領域（`log devspaces`）およびシステムのデバイス領域（`sys devspaces`）は、共有ディスク（共有ファイルシステムまたは共有 Raw デバイスのいずれか）上に存在する必要があります。

## シナリオ2

SAP MaxDB の *IndepDataPath* および *IndepProgPath* が両方のサーバーでローカルにインストールされている。SAP MaxDB *DependPath* はローカルまたはプライマリサーバー上の共有ファイルシステムにインストールできる。

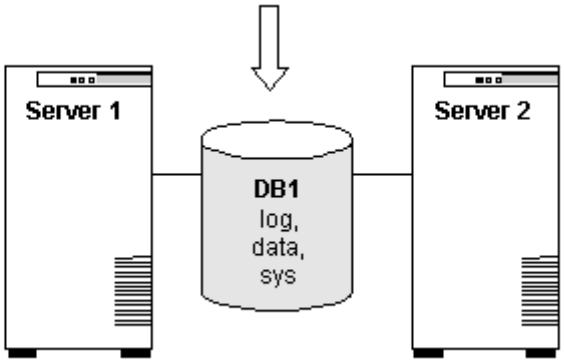
- レジストリーファイル `/etc/opt/sdb` は、クラスター内の各サーバーに存在する必要があります。このファイルは、リソース拡張操作中にターゲットサーバーからアクセス可能でなければならないため、共有ストレージに置いてはいけません。
- データベースインスタンスのデータデバイス領域（`data devspaces`）、ログのデバイス領域（`log devspaces`）およびシステムのデバイス領域（`sys devspaces`）は、共有ディスク（共有ファイルシステムまたは共有 Raw デバイスのいずれか）上に存在する必要があります。
- データベースインスタンス実行ディレクトリー(`RUNDIRECTORY`)は、共有ストレージに配置する必要があります。`RUNDIRECTORY` の値は、`DBMCLI` コマンドの `param_directput` を使用して変更できます。データベースの作成後に `RUNDIRECTORY` の値を変更した場合は、パラメーターの更新を反映するために、データベースインスタンスを停止して再起動する必要があります。

- データベースインスタンスの config (<IndepDataPath>/config) ディレクトリー構造は、データベースインスタンスが保護されるクラスター内のすべてのサーバー上の同じ場所に存在する必要があります。さらに、データベースインスタンスのパラメーターファイルは、テンプレート（またはプライマリ）サーバーからクラスター内のすべてのバックアップサーバーにコピーする必要があります。パラメーターの更新後には、パラメーターファイルをクラスター内のすべてのサーバーに再配布します。必要なファイルは以下の通りです。

```
config/Databases.ini

config/Installations.ini
```

# 6.19.3.4.1. アクティブ/スタンバイ設定の例

IndepData = /shr1/data
IndepPrograms = /shr1/programs
DependPath = /shr1/depend




**設定上の注意事項：**

- 両方のサーバーで共有ストレージ上の *IndepProgPath* 、 *DependPath* 、 *IndepDataPath* を使用します。
- データベースインスタンス DB1 は、共有ストレージにあります。これには、すべてのログデバイス領域、データデバイス領域およびシステムデバイス領域が含まれます。
- *Server 1* がアクティブな時は、 *Server 2* は共有ディスク上のファイルとディレクトリーにはアクセスできません。

# 6.19.3.5. アクティブ/アクティブの場合の考慮事項

アクティブ/アクティブ設定では、各サーバーは1つの SAP MaxDB インスタンスをアクティブに実行しており、障害発生時には他のサーバーのバックアップとして機能します。以下のシナリオでは、アクティブ/アクティブ設定で SAP MaxDB リソースインスタンスを保護する際に、順番に守る必要のある具体的な要件を示します。

内部ロードバランサーは、アクティブ/アクティブ構成ではサポートされていません。

## アクティ/アクティブのシナリオ

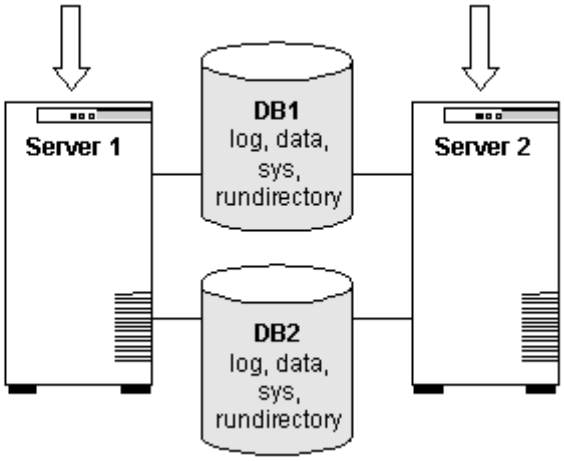
SAP MaxDB の *IndepDataPath* 、 *IndepProgPath* 、 *DependPath* が両方のサーバーにローカルにインストールされている。

- レジストリーファイル `/etc/opt/sdb` は、クラスター内の各サーバーに存在する必要があります。このファイルは、リソース拡張操作中にターゲットサーバーからアクセス可能でなければならないため、共有ストレージに置いてはいけません。
- データベースインスタンスのデータデバイス領域 (`data devspaces`)、ログのデバイス領域 (`log devspaces`) およびシステムのデバイス領域 (`sys devspaces`)は、共有ディスク（共有ファイルシステムまたは共有 Raw デバイスのいずれか）上に存在する必要があります。
- データベースインスタンス実行ディレクトリー（`RUNDIRECTORY`）は、共有ストレージに配置する必要があります。`RUNDIRECTORY` の値は、`DBMCLI` コマンドの `param_directput` を使用して変更できます。データベースの作成後に `RUNDIRECTORY` の値を変更した場合は、パラメーターの更新を反映するために、データベースインスタンスを停止して再起動する必要があります。
- データベースインスタンスの`config(<IndepDataPath>/config)`ディレクトリー構造は、データベースインスタンスが保護されるクラスター内のすべてのサーバー上の同じ場所に存在する必要があります。さらに、データベースインスタンスのパラメーターファイルは、テンプレート（またはプライマリ）サーバーからクラスター内のすべてのバックアップサーバーにコピーする必要があります。パラメーターの更新後には、パラメーターファイルをクラスター内のすべてのサーバーに再配布します。必要なファイルは以下の通りです。

```
config/Databases.ini

config/Installations.ini
```

# 6.19.3.5.1. アクティブ/アクティブ設定の例

| | |
|---|---|
| IndepData = /usr/sapdb/data<br>IndepPrograms = /usr/sapdb/prog<br>DependPath = /usr/sapdb/dep | IndepData = /usr/sapdb/data<br>IndepPrograms = /usr/sapdb/prog<br>DependPath = /usr/sapdb/dep |




**設定上の注意事項：**

- *IndepDataPath* 、 *IndepProgPath* 、 *DependPath* は両方のサーバーにローカルにインストールします。
- 各データベースは別々の共有ディスクに構成されています。データベースインスタンスには、すべてのログデバイス領域、システムデバイス領域およびデータデバイス領域が含まれます。
- 各データベースの `RUNDIRECTORY` も共有ディスク上にあります。
- `DB1` のデータベース構成ファイルが *Server 2* にコピーされ、`DB2` のデータベース構成ファイルが *Server 1* にコピーされています。構成ファイルは `<IndepDataPath>/config/<db instance>` にあります。
- 最初に、 *Server 1* は `DB1` を実行し、 *Server 2* は `DB2`を実行します。スイッチオーバーの際には、1台のサーバーで両方のデータベースを実行できます。

# 6.19.4. LifeKeeper での SAP MaxDB の設定

SAP MaxDB データベースおよび SPS ソフトウェアのインストールと設定には、次の手順を推奨します。各ステップでは、詳細な作業を説明しています。

- SAP MaxDB ソフトウェアのインストール
- SAP MaxDB データベースの作成
- User_Key の作成
- LifeKeeper Core のインストールと SAP Max DB Recovery Kit

これらのタスクを実行すると、リソース階層を作成して SAP MaxDB データベースを保護する準備が整います。

# 6.19.4.1. SAP MaxDB ソフトウェアのインストール

同じパラメーターと設定を使用して、クラスター内のすべてのサーバーに SAP MaxDB ソフトウェアをインストールします。詳細については、「SAP MaxDBインストールガイド」を参照してください。SAP MaxDB が SPS で確実に動作するようにするための推奨事項を以下に示します。

- root 以外 のシステムユーザー（OS ユーザー）は、次のようにすべてのサーバーに存在する必要があります。
- この OS ユーザーは、SAP MaxDB ソフトウェアのインストールとサブディレクトリの所有者として指定されているか、データベースインスタンスの管理に必要なソフトウェアのインストールパスとサブディレクトリに対して適切な権限を持っている必要があります。
- この OS ユーザーには、DBMCLI および x_server ユーティリティを使用する権限が必要です。また OS ユーザーは、x_server コマンドを使用して vserver の起動および停止を実行できる必要があります。
- OS ユーザー名に使用できるのは英数字のみです。
- この OS ユーザーのユーザー ID とグループ ID は、すべてのサーバーで同一でなければなりません。
- SAP MaxDB クライアントソフトウェアパッケージをインストールする必要があります。またこれらのパッケージには、SAP MaxDB DBMCLI クライアントユーティリティと SAP MaxDB x_server ユーティリティが含まれている必要があります。
- SAP MaxDB のリソース階層を含む各 SPS サーバーは、SAP MaxDB インスタンスの `/etc/services` ファイルに同じサービスエントリを持つ必要があります。

# 6.19.4.2. SAP MaxDB データベースの作成

「SAP MaxDB ユーザーマニュアル」の指示に従ってデータベースを作成してください。また、以下の推奨事項にも留意してください。

- クライアントユーティリティーを使用して、開始、停止、状態取得、データベースパラメーターの取得を実行する権限を持つ DBM オペレーターが必要です。
- データベースインスタンスのデータデバイス領域（`data devspaces`）、ログデバイス領域（`log devspaces`）およびシステムデバイス領域（`sys devspaces`）は、共有ディスク（共有ファイルシステムまたは共有 Raw デバイスのいずれか）上に存在している必要があります。
- SAP MaxDB データベース名には英数字のみ使用できます。
- User_Key は、DBMCLI ユーティリティーでの操作中に SAP MaxDB Recovery Kit で使用するために必要です。必要なパラメーターについては、 User_Key の作成 を参照してください。
- データベースの作成後、SAP MaxDB データベースインスタンスの自動起動を無効にする必要があります。 一旦 SPS の保護下に入ると、SPS がデータベースの開始と停止を処理します。
- `sdb` ファイルは、`/etc/opt` のすべてのサーバーに存在する必要があります。このファイルが存在しない場合、いくつかの SAP MaxDB ユーティリティーが誤った結果を返す可能性があります。これは、リソースの作成および拡張中の SPS の動作にも影響します。アクティブ/スタンバイ設定では、このファイルを手動でバックアップサーバーにコピーする必要があります。
- `databases.ini` ファイルが `IndepDataPath/config` ディレクトリーのすべてのサーバーに存在することを確認してください。
- LifeKeeper で設定した `databases.ini` ファイル内の各データベースエントリーをバックアップサーバにコピーしてください。

# 6.19.4.3. User_Key の作成

SAP MaxDB インスタンスには、ユーザーがデータベースインスタンスに正常にアクセスするためのいくつかのオプションが必要です。これらの必要な情報は、データベースインスタンスへのアクセスに使用されている SAP MaxDB ツールに渡す必要があります。SAP MaxDB ソフトウェアには、SAP MaxDB ツールに対する多くの必須オプションの指定を簡素化するための xuser ツールが含まれています。 xuser ツールを使用すると、ユーザーデータをあらかじめ定義して保存できます。このデータが保存されると、DBMCLI またはユーザーオプションを必要とする他のツールを呼び出すときにこのデータを使用できます。この事前定義されたユーザーデータは、ユーザーキー（User_Key）に保管されます。個々のユーザーは、同じデータベースまたは複数のデータベースの複数のユーザーキーを管理および保持できます。各キーには、ユーザー名/パスワード、データベース名、データベースサーバー名などのオプションの組み合わせが含まれています。

SAP MaxDB Recovery Kit では、保護されている各データベースインスタンスに対して有効なユーザーキーが必要です。この User_Key は、データベースインスタンスを所有する OS ユーザーによる作成およびアクセスが可能である必要があります。各User_Keyに対して指定されたユーザー情報は、次の権限を持つ DBM オペレーターのものでなければなりません。

- DBStart
- DBStop
- DBInfoRead
- ParamRead

User_Key は、<file name> が以下の表に記載する有効な XUSER エントリーを含む場合、次のコマンドで生成できます。

```
xuser -b <file name>
```

| パラメーター | パラメーターの定義 |
|---|---|
| USERKEY | User_Key のユニークな名前 |
| USERID | dbm オペレーターのユーザー名 |
| PASSWORD | ユーザーのパスワード |
| SERVERDB | このキーが参照するデータベースインスタンス名 |
| SERVERNODE | データベースを実行しているサーバー名（これは、DNS または SPS で保護されている IP のホストファイルエントリーである必要があります） |
| SQLMODE | 互換性のある SQL ダイアレクトを決定します |
| CACHELIMIT | 特定のセッションのキャッシュ制限を決定します |
| TIMEOUT | 非アクティブなセッションを終了するまでの時間（単位：秒）（デフォルトは-1） |
| ISOLATION | ユーザーに影響するロックに使用される分離レベルを決定します（デフォルトは-1） |

| | |
|---|---|
| DB_LOCAL | データベースのロケールを指定します |

パラメーターに関する詳細は、「SAP MaxDB ユーザーマニュアル」を参照してください。適切なエントリーが指定されたら、xuser ツールを使用して OS ユーザーのホームディレクトリに .XUSER.62 ファイルを生成します。2つのキーを含むサンプル XUSER ファイルは以下に含まれています（DEFAULT User_Key のエントリーが存在する必要があります）。

**DEFAULT**
**NULLDB**
**NULLDB**
**NULLDB**
**LKIP.example.com**
**INTERNAL**
**-1**
**-1**
**-1**
**my_locale**
**LK_USERKEY**
**LKDBMOPER**
**LKDBMPASSWD**
**DB1**
**LKIP.example.com**
**INTERNAL**
**-1**
**-1**
**-1**
**en_US**

この例の XUSER ファイルでは、 `DEFAULT` と `LK_USERKEY` の2つのユーザーキーを作成するように指定しています。xuserツールを実行してUser_Keyを生成した後は、 **xuserツールで使用するために指定したファイルを削除しないでください** 。

# 6.19.4.4. LifeKeeper ソフトウェアのインストール

SAP MaxDB ソフトウェアをインストールし、データベースを作成して User_Key を作成したら、LifeKeeper Core ソフトウェアと必要なパッチのインストール後に SAP MaxDB Recovery Kit をインストールします。また、Raw デバイスで SAP MaxDB を使用する場合は、LifeKeeper のインストールイメージファイルから Raw I/O Recovery Kit をインストールする必要があります。Raw デバイスの設定に関する要件と手順については、付録 を参照してください。

LifeKeeperパッケージのインストールに関する詳細は、LifeKeeper for Linux インストールガイド を参照してください。

# 6.19.5. SAP MaxDB リソース設定作業

前のセクションで説明した設定作業を完了すると、SAP MaxDB リソース階層の作成および拡張の準備が整います。

LifeKeeper for Linux SAP MaxDB Recovery Kit の設定には、以下のタスクが利用可能です。

- **リソース階層の作成** - SAP MaxDB リソース階層を作成する
- **リソース階層の削除** - SAP MaxDB リソース階層を削除する
- **リソース階層の拡張** - SAP MaxDB リソース階層をプライマリサーバーからバックアップサーバーに拡張する
- **リソース階層の拡張解除** - SAP MaxDB リソース階層を SPS クラスター内の単一のサーバーから拡張を解除（削除）する

LifeKeeper Core のリソース階層の設定手順については、SIOS テクノロジーのウェブサイトに掲載している LifeKeeper for Linux テクニカルドキュメント を参照してください。

以下のタスクはすべての Recovery Kit で共通の手順であるため、LifeKeeper for Linux テクニカルドキュメント の 管理 セクションで説明しています。

- Create a Resource Dependency - 既存のリソース階層と別のリソースインスタンスとの間に親子関係を作成し、クラスター内のすべてのターゲットサーバーに依存関係の変更を伝播します。
- Delete a Resource Dependency - リソースの依存関係を削除し、クラスター内のすべてのターゲットサーバーに依存関係の変更を伝播します。
- In Service - リソース階層を特定のサーバー上でアクティブにします。
- Out of Service - 特定のサーバー上でリソース階層を非アクティブにします。
- View /Edit - 特定のサーバー上のリソース階層のプロパティの表示または編集を行います。

**注記：** このセクションの残りの部分では、設定は **[Edit]** メニューから実行します。また、次の操作によってほとんどのタスクを実行することができます。

- ツールバーから実行する
- **[Status]** 画面の左ペインにある **[global resource]** を右クリックする
- **[Status]** 画面の右ペインにある **[resource]** を右クリックする

右クリックを使用することにより、**[Edit]** メニューを使用して情報を入力しないで済みます。

# 6.19.5.1. SAP MaxDB リソース階層の作成

以下の手順をプライマリサーバーで実行してください。

1. **[Edit]** メニューの **[Server]** から **[Create Resource Hierarchy]** を選択します。

   *Create Resource Wizard* ダイアログボックスが表示されます。

2. ドロップダウンリストから **[SAP MaxDB Database]** を選択し、 **[Enter]** をクリックします。

3. 以下の情報の入力を要求されます。ダイアログボックスで **[Back]** ボタンが有効な場合は、前のダイアログボックスに戻ることができます。これは、エラーが発生した際に、前に入力した情報を修正する必要がある場合に便利な機能です。また、いつでも **[Cancel]** をクリックして、作成処理全体を取り消すことができます。

| フィールド | ヒント |
|---|---|
| **Switchback Type** | インテリジェントまたは自動のいずれかを選択します。これにより、フェイルオーバー後にバックアップサーバー上で in-service（アクティブ）になった後の SAP MaxDB リソースのプライマリサーバーへのスイッチバック方法が決まります。インテリジェントスイッチバックでは、管理者がリソースをプライマリサーバーに戻す必要がありますが、プライマリサーバーがオンラインに戻って SPS のコミュニケーションパスが再確立されると、自動的にスイッチバックが行われます。<br><br>**注：** スイッチバック戦略は、SAP MaxDB リソースが使用する従属リソースのものと一致している必要があります。 |
| **SAP MaxDB Programs Directory** | このフィールドは、対応するサーバーの SAP `/etc/opt/sdb` ファイルにある SAP MaxDB プログラムパスがデフォルトで含まれています。別のディレクトリパスを入力することもできます。パス名に使用できる有効な文字は、文字、数字および特殊文字 - _ . / です。 |
| **SAP MaxDB Instance** | このフィールドには、デフォルトで、SPS 階層が存在しないシステム上にある最初の SAP MaxDB インスタンスの名前が含まれます。ドロップダウンリストには、システムで使用可能な他のデータベースインスタンスが表示されます。 |
| **SAP MaxDB System User** | SAP MaxDB コマンドを所有しているか、SAP MaxDB コマンドを実行する権限のあるシステムユーザーです。 このユーザーは、対応するサーバー上に存在する必要があります。選択画面で有効なユーザー名を入力します。 |
| **User_Key** | このフィールドには、XUSER User_Key のデフォルト値が含まれます。User_Key は、SAP MaxDB ツールで使用するデータベースユーザーのデータを保存するために使用されます。対応するサーバー、OS ユーザーおよびデータベースインスタンスの組み合わせに対して有効な User_Key を入力します。 |
| **SAP MaxDB** | これは、プライマリサーバー上の新しい SAP MaxDB データベースリソースの一意のタグ名です。デフォルトのタグ名は、SAP MaxDB インスタンスで構成されています。別の一意の |

| | |
|---|---|
| **Database Tag** | タグ名を入力することもできます。タグに使用できる有効な文字は、文字、数字および特殊文字 - _ . / です。 |

4. **[Create]** をクリックし、Create Resource Wizard で SAP MaxDB リソース階層を作成します。すると SPSは入力されたデータを検証します。SPS が問題を検出すると、情報ボックスにエラーメッセージが表示されます。

5. SAP MaxDB リソース階層を正常に作成したことを示すメッセージが表示され、フェイルオーバー保護を実施するためにクラスター内の別のサーバーにその階層を拡張する必要があります。 **[Next]** をクリックします。

6. **[Continue]** をクリックすると、SPS は *Pre-Extend Wizard* を表示します。リソース階層を他のサーバーに拡張する方法については、 SAP MaxDB リソース階層の拡張 のステップ2を参照してください。

# 6.19.5.2. SAP MaxDB リソース階層の拡張

この操作は、 **[Edit]** メニューから開始することも、 **[Create Resource Hierarchy]** オプションの完了時に自動的に開始することもできます。その場合は、以下の手順2を参照してください。

1. **[Edit]** メニューの **[Resource]** から **[Extend Resource Hierarchy]** を選択してください。 *Pre-Extend Wizard* が表示されます。拡張操作に慣れていない場合は、 **[Next]** をクリックしてください。SPS の **[Extend Resource Hierarchy]** のデフォルト値が分かっていて、入力と確認を省略する場合は **[Accept Defaults]** をクリックしてください。

2. *Pre-Extend Wizard* に以下の情報を入力します。

   注記： 最初の2つのフィールドは、 **[Edit]** メニューから拡張を開始した場合にのみ表示されます。

| フィールド | ヒント |
|---|---|
| **Template Server** | SAP MaxDB リソースが現在 in service のサーバーを選択します。 |
| **Tag to Extend** | 拡張する SAP MaxDB リソースを選択します。 |
| **Target Server** | 拡張先のサーバーを入力するか、選択してください。 |
| **Switchback Type** | これにより、フェイルオーバー後にバックアップサーバー上で in-service（アクティブ）になった後の SAP MaxDB リソースのプライマリサーバーへのスイッチバック方法が決まります。インテリジェントまたは自動のいずれかを選択できます。スイッチバックタイプは、後で必要に応じて **[Resource Properties]** ダイアログボックスの **[General]** タブで変更できます。<br><br>**注記:** スイッチバック戦略は、SAP MaxDB リソースが使用する従属リソースのものと一致している必要があります。 |
| **Template Priority** | **[Template Priority]** を選択または入力します。これはサーバーで現在 in service の SAP MaxDB 階層の優先順位です。優先順位は、1～999 の範囲で未使用の値が有効で、小さい数字ほど優先順位が高くなります (数字 1 が最高の優先順位に相当)。拡張処理時に、別のシステムですでに使用中の優先順位をこの階層に対して指定することはできません。デフォルト値を推奨します。<br><br>**注記:** このフィールドは階層を最初に拡張するときだけ表示されます。 |
| **Target Priority** | これは、他のサーバーにある同等の階層に対する、新しく拡張する SAP MaxDB 階層の優先順位です。1～999 の範囲で、まだ優先順位として使用されていない値が有効で、リソースのカスケーディングフェイルオーバシーケンスにおけるサーバーの優先順位を示します。SPS のデフォルトでは、階層が作成されたサーバーに「1」が割り当てられることに注意してください。優先順位は連続している必要はありませんが、特定のリソースに対して 2 つのサーバーで同じ優先順位を割り当てることはできません。 |

3. 拡張前のチェックが正常に終了したというメッセージが表示された後、**[Next]** をクリックしてください。

4. 拡張する階層に応じて、拡張されるリソースタグ (一部編集不可) を示す一連の情報ボックスが表示されます。

5. *Extend Wizard* に、次の情報を入力します。

| | |
|---|---|
| SAP MaxDB Programs Directory | このフィールドは、対応するサーバーの SAP `/etc/opt/sdb` ファイルにある SAP MaxDB プログラムパスがデフォルトで含まれています。パス名に使用できる有効な文字は、文字、数字および特殊文字 - _ . / です。 |
| User_Key | このフィールドには、XUSER User_Key のデフォルト値が含まれます。User_Key は、SAP MaxDB ツールで使用するデータベースユーザーのデータを保存するために使用されます。対応するサーバー、OS ユーザーおよびデータベースインスタンスの組み合わせに対して有効な User_Key を入力します。 |
| SAP MaxDB Database Tag | これは、ターゲットサーバー上の新しい SAP MaxDB データベースリソースの一意のタグ名です。デフォルトのタグ名は、SAP MaxDB インスタンスで構成されています。別の一意のタグ名を入力することもできます。タグに使用できる有効な文字は、文字、数字および特殊文字 - _ . / です。 |

6. 「階層の拡張が完了しました」というメッセージが表示されたら、[Next Server] をクリックして階層を別のサーバーに拡張するか、拡張しない場合は **[Finish]** をクリックします。

7. 「`Hierarchy Verification Finished`」というメッセージが表示されたら、**[Done]** をクリックします。

# 6.19.5.3. SAP MaxDB リソース階層の拡張解除

SPS クラスター内の単一サーバーからリソース階層を削除するには、次の手順を実行してください。

1. **[Edit]** メニューの **[Resource]** から **[Unextend Resource Hierarchy]** を選択します。
2. SAP MaxDB リソースの拡張を解除するターゲットサーバーを選択します。リソースが現在 in-service のサーバーを選択することはできません。（このダイアログボックスは、右ペインでリソースインスタンスを右クリックして **[Unextend]** タスクを選択した場合は表示されません）。 **[Next]** をクリックします。
3. 拡張を解除する SAP MaxDB 階層を選択し、 **[Next]** をクリックします。（このダイアログは、いずれかのペインでリソースインスタンスを右クリックして **[Unextend]** タスクを選択した場合は表示されません）。
4. ターゲットサーバーと拡張の解除を選択した SAP MaxDB リソース階層を確認する情報ボックスが表示されます。 **[Unextend]** をクリックします。
5. SAP MaxDB リソースの拡張解除が正常に完了したことを確認する別の情報ボックスが表示されます。 **[Done]** をクリックして、 **[Unextend Resource Hierarchy]** メニューを終了します。

# 6.19.5.4. SAP MaxDB リソース階層の削除

SPS 構成のすべてのサーバーから SAP MaxDB リソースを削除するには、次の手順を実行します。

1. **[Edit]** メニューの **[Resource]** から **[Delete Resource Hierarchy]** を選択します。
2. SAP MaxDB リソース階層を削除するターゲットサーバーを選択します。（グローバルリソースの左ペインまたは個別のリソースインスタンスの右ペインのいずれかを右クリックして **[Delete Resource]** タスクを選択した場合、このダイアログは表示されません。）
3. 削除する階層を選択します。（このダイアログは、左ペインまたは右ペインでリソースインスタンスを右クリックして **[Delete Resource]** タスクを選択した場合は表示されません。） **[Next]** をクリックします。
4. 削除するターゲットサーバーと階層を確認する情報ボックスが表示されます。 **[Next]** をクリックします。
5. SAP MaxDB リソースが正常に削除されたことを確認する別の情報ボックスが表示されます。
6. **[Done]** をクリックして終了します。

# 6.19.5.5. リソース階層のテスト

プライマリサーバーからバックアップサーバーへのリソースインスタンスのフェイルオーバーをシミュレートする手動のスイッチオーバーを開始することによって、SAP MaxDB リソース階層をテストすることができます。

## LifeKeeper GUI から手動スイッチオーバーを実行する

LifeKeeper GUI から手動でスイッチオーバーを開始するには、 **[Edit]** メニューの **[Resource]** から **[InService]** を選択します。例えば、バックアップサーバー上で in-service 要求が実行されると、SAP MaxDB のリソース階層はバックアップサーバー上で in-service 状態になり、プライマリサーバー上のサービスは停止されます。この時点で、元のバックアップサーバーはプライマリサーバーになり、元のプライマリサーバーがバックアップサーバーになります。

Out of Service 要求を実行すると、リソース階層は他のサーバー上でサービスを開始することなく、Out of Serviceとなります。

**重要：** リソース階層をバックアップサーバー上で In Service にした後、特に Raw デバイスをデバイス領域として使用する場合は、データベースへの接続を試みる必要があります。これは、すべてのディスクパーティションがバックアップサーバー上に表示され、Rawバインディングが正しく確立されていることを確認するために必要です。

Raw バインディングがバックアップサーバー上に確立されていない場合は、新しいパーティションがプライマリサーバーで作成されて構成に追加されているにもかかわらず、パーティションテーブルがバックアップサーバー上でまだ更新されていないことが原因である可能性が高いです。

これを解決するには、パーティションテーブルが正しく更新されるようにバックアップサーバーを再起動してください。

# 6.19.6. LifeKeeper リソース階層の管理

リソース階層の作成後、次のタスクが必要になる場合があります。

---

User_Key の変更

OS User の変更

パラメーターの更新

# 6.19.6.1. User_Key の変更

既存の階層の User_Key を変更する必要がある場合は、その階層を削除して再作成する必要があります。

# 6.19.6.2. OS User の変更

データベースインスタンスを所有する OS User を変更する必要がある場合は、その階層を削除して再作成する必要があります。

# 6.19.6.3. パラメーターの更新

SAP MaxDB インスタンスのデータベースパラメーターが更新されたら、更新されたパラメーターファイルがインスタンスを保護しているすべてのサーバーに確実に再配布されるようにする必要があります。*IndepDataPath* が共有ディスク上にある場合、インスタンスを保護しているすべてのサーバーは自動的に更新されます。

# 6.19.7. SAP MaxDB トラブルシューティング

## 一般的なヒント

SAP MaxDB Recovery Kitの使用中に、次のエラーメッセージが表示される場合があります。

| エラーメッセージ | 解決策 |
| --- | --- |
| The tunable MAXDB_ILB_ENABLED is set in the LifeKeeper defaults file for Internal Load Balancer support. Protecting multiple SAP MaxDB resources in this configuration requires all SAP MaxDB resource hierarchies to share a common resource. | 各 SAP MaxDB リソースに対して共通するリソースに依存関係を作成、もしくは内部向けのロードバランサーのサポートを無効にしてください。Azure にもあるような内部向けロードバランサー (ILB) は、 1 つのサーバーのみで X サーバーポートをアクティブにする必要があります。LifeKeeper の SAP MaxDB リソースに対する in-service 操作により、X Server の処理が開始します。 通常の out-of-service の操作では、X サーバーの処理は停止しません。<br><br>LifeKeeperの設定ファイル（ */etc/default/LifeKeeper* ）で MAX_ILB_ENABLED を設定することで、ノード内最後の SAP MaxDB リソースが削除されたときに、X サーバープロセスも停止するようにサービス停止の動作を変更できます。この時、LifeKeeper が X サーバーを1つのサーバーのみでアクティブにしておくために、すべての SAP MaxDB リソースが同じ階層にある必要があります。<br><br>複数の SAP MaxDB リソースを単一の階層に配置するには、以下の手順で階層の最上位にルートノードとして Generic Application リソースを作成します。<br><br>• **[Edit]** メニューで、 **[Server]** 、 **[Create Resource Hierarchy]** の順に選択します。 **[Create Resource Wizard]** ダイアログが表示されます。<br>• ドロップダウンリストから **[Generic Application]** を選択し、 **[Enter]** をクリックします。<br>• Generic Application リソース階層の作成 の指示に従ってください。restore/ remove スクリプトは「/bin/true」を指定しても構いません。<br>• ルートリソースを作成したら、ルートリソースと各 SAP MaxDB リソースの間に依存関係を作成します。<br>　◦ **[Edit]** メニューで、 **[Resource]** 、 **[Create Dependency]** の順に選択します。<br>　◦ リソースが In-Service になっている サーバー を選択します。<br>　◦ **[Parent Resource Tag]** にて、親リソースとして先ほど作成したルートリソースをドロップダウンリストから選択します。 |

| | |
|---|---|
| | ◦ **[Child Resource Tag]** にて、子リソースとして SAP MaxDB リソースを選択します。<br><br>ターミナルリーフノードを使用することもお勧めします。ユーティリティー上に単一のルートリソースを作成したら、create_terminal_leaf を使用して、階層の各ブランチに Generic Application リソースを作成できます。上記で作成したルートリソースのタグが「MaxDB_root」の場合、リーフノードを作成するには、「create_terminal_leaf MaxDB_root」を実行します。<br>ルートリソースとターミナルリーフリソースとの依存関係を作成して、新しい SAP MaxDB リソースを階層に追加する必要があります。 |
| Unable to create pipe `/usr/spool/sql/fifo/<db instance>` | `/usr/spool/sql` ディレクトリには、データベースインスタンスを所有するシステムユーザーのアクセスを許可するための適切な権限が必要です。 |
| Open device space *<dev>* permission denied | バックアップとプライマリ上のデバイス領域では、同じユーザーおよびグループの権限に加え、同じ所有者が必要です。 |
| Unable to set uid on startup | `<DependPath>/pgm/dbmsrv` の setuid ビットを設定する必要があります。ファイルの所有者は SAP MaxDB システムユーザーでなければなりません。 |
| runtime environment error | 考えられる原因に対し、それぞれ解決策が存在します。<br>• **データベースインスタンスのパラメーターと設定ファイルが存在しません。**<br>データベースパラメーターファイルを作成するか、テンプレートサーバーからファイルをコピーします。<br>• **データベースにライブラリの問題が発生しました。**<br>サーバーとソフトウェアインストールの組み合わせの場合、*libpthread-0.8.so* ライブラリを使用する必要があります。 手順については、SAP MaxDB のマニュアルを参照してください。<br>• **データベースインスタンスの環境が壊れています。**<br>プロセスを手動で強制終了する必要があります。その後、リソースを in-service 状態に復元してください。 |
| open Registry: Permission denied | `/usr/spool/sql/ini` は、SAP MaxDB ソフトウェアを所有しているユーザーおよびグループが所有する必要があります。さらに、ユーザーおよびグループはこのディレクトリで読み取り/書き込みのアクセス権限を持っている必要があります。 |
| **ERR_USRREAD**: could not read user data | `<IndepDataPath>/config/<db instance>` の設定ファイルがサーバーに存在しないか、適切なアクセス権限がありません。データベースインスタンスを所有しているシステムユーザーに対して適切なアクセス権限が設定されたファイルが存在することを確認してください。 |

# 6.19.7.1. SAP MaxDB Recovery Kit エラーメッセージ

| エラー番号 | メッセージ |
|---|---|
| 111000 | Usage: %s independent_program_path <validate:value_1:…:value_n:> |
| 111001 | Usage: %s %s %s |
| 111002 | No value specified to script %s for input argument %s. |
| 111003 | User %s with User_Key %s cannot access instance %s.<br><br>Action: Specify a User_Key for the given user with database access rights. |
| 111004 | The user %s does not exist on the server %s. |
| 111005 | The SAP DB instance %s is not running on server %s. |
| 111006 | Database Manager Utilities were not found in the specified path %s. |
| 111007 | An SPS internal error occurred in utility %s.<br><br>Action: Retry operation. |
| 111008 | Unable to obtain %s device space information for SAP DB instance %s for user %s and User_Key %s.<br><br>Action: Verify that the user and User_Key are valid for the corresponding database instance. |
| 111009 | Unable to create raw resource hierarchy for %s.<br><br>**Action: Verify that the underlying device is a shared device.** |
| 111010 | Unable to create filesystem resource hierarchy for %s.<br><br>Action: Verify that the underlying device is a shared device. |
| 111011 | Unable to determine the type of the dev space or install path %s. |

| | |
|---|---|
| | Action: Valid dev space types include file system and/or raw devices. |
| 111012 | The path %s is not on a shared filesystem . |
| 111013 | The SAP DB instance %s is already under SPS protection on server %s. |
| 111014 | The SAP DB instance %s has been successfully started on server %s. |
| 111015 | The SAP DB instance %s has been successfully stopped on server %s. |
| 111016 | Unable to start SAP DB instance %s on server %s. |
| 111017 | Unable to stop SAP DB instance %s on server %s. |
| 111018 | Attempting db_warm for database instance %s after db_start failure. |
| 111019 | The SAP DB x_server has been successfully started on server %s. |
| 111020 | The SAP DB x_server has been successfully stopped on server %s. |
| 111021 | The SAP DB x_server is not running on server %s |
| 111022 | Unable to start SAP DB x_server on server %s.<br><br>Action: A problem has occurred using the x_server utility, check the SAP DB logs and correct the problem. |
| 111023 | Unable to stop SAP DB x_server on server %s.<br><br>Action: A problem has occurred using the x_server utility; check the SAP DB logs and correct the problem. |
| 111024 | The SAB DB file SAP_DBTech.ini was not found on server %s.<br><br>Action: Verify that SAP DB is installed correctly on the specified server. |
| 111025 | The user id for user %s is not the same on server %s and %s. |
| 111026 | The group id for user %s is not the same on server %s and %s. |
| 111027 | The service file entries for are not the same on server %s and %s. |
| 111028 | One or more of the SAP DB service file entries do not exist on server %s. |
| 111029 | No dependents were found for resource %s on server %s. |
| 111064 | The tunable MAXDB_ILB_ENABLED is set in the LifeKeeper defaults file for Internal Load Balancer support. Protecting multiple SAP MaxDB resources in this configuration requires all |

| | |
|---|---|
| | SAP MaxDB resource hierarchies to share a common resource.<br><br>**Action: Please refer to SAP MaxDB Troubleshooting.** |

# 6.19.8. 付録：Raw I/O によるデバイス領域を作成する

## 要件

Raw I/O で SAP MaxDB Recovery Kit を使用するには、次の要件を満たす必要があります。

- Linux OS はRaw I/O デバイスをサポートしている必要があります。多くのディストリビューションでこのサポートはカーネル2.4に含まれていましたが、カーネル2.2で Raw I/Oをサポートするディストリビューションもあります。
- すべての Raw I/O デバイスは、共有ディスクパーティションにバインドされている必要があります。Raw I/O デバイスに配置されるデバイス領域 *（devspace）* の数によって、必要な Raw デバイスと共有ディスクパーティションの正確な数が決まります。Raw デバイス上で *devspaces* を作成する方法については、SAP MaxDB マニュアルを参照してください。
- SAP MaxDB ソフトウェアのバージョンは、Raw I/O デバイスの使用をサポートしている必要があります。

# 6.19.8.1. Raw I/O セットアップ手順

1. SAP MaxDB のデバイス領域に適したサイズの共有ディスクパーティションを選択します。

2. 未使用の Raw デバイスノードをこのパーティションにバインドします。これはマシンがリブートされてルートアクセスが必要な時に毎回実行する必要があるため、システム初期化ファイル（rc.local または boot.local）に Raw バインディングを追加しても構いません。階層が SPS 保護下に置かれたら、これらのバインディングはファイルから削除する必要があります。SPSは、SPS保護下にある Raw I/Oデバイスの Raw バインディングを再確立します。どの Raw デバイスノードがすでに使用されているかを確認するには、以下の例のように raw -qa コマンドを実行します。
```
raw -qa
raw /dev/raw/raw1 /dev/sda1
```

3. Rawデバイスコントローラ（/dev/raw/rawctl または /dev/raw）およびデータベースインスタンスを保護するすべてのサーバー上のディスクパーティションの両方に、グローバル読み取りアクセス権を設定します。
```
chmod a+r /dev/raw/rawctl
```

4. データベースインスタンスを保護するすべてのサーバー上の Raw デバイスに対するグループおよびユーザーの読取り/書込み権限を設定します。
```
chmod 664 /dev/raw/raw1
```

5. データベースインスタンスを保護するすべてのサーバー上で、指定されたデータベースインスタンスの Raw デバイスの所有者を SAP MaxDB OS Userに変更します。
```
chown -R sapdb:sapdb /dev/raw/raw1
```

6. param_adddevspace または db_adddevspace を使用して、データベースにデバイス領域を追加します。SAP MaxDB ユーザーマニュアル、データベースマネージャー CLI マニュアルを参照してください。

# 6.19.8.2. 階層作成後のデバイス領域の追加

SPS で SAP MaxDB 階層を作成した後にRaw I/O デバイスまたは共有ファイルシステムにテーブルスペースを追加する場合は、LifeKeeper GUI を使用して Raw I/O デバイスまたはファイルシステムのリソース階層を手動で作成する必要があります。新たに作成されたリソース階層は、SAP MaxDB リソース階層に従属している（子になっている）必要があります。必要に応じて、更新されたパラメーターファイルを、データベースインスタンスを保護するすべてのサーバーに再配布する必要があります（*IndepDataPath* が共有ディスク上にある場合は必要ありません）。

# 6.20. Sybase Recovery Kit 管理ガイド

## Sybase ASE Recovery Kit ドキュメンテーション

Sybase Adaptive Server Enterprise (以下 Sybase ASE) は、高性能ビジネスアプリケーション向けの強力なデータ管理プラットフォームです。Sybase ASE は、OLTP 負荷の処理を特に得意とする汎用的なエンタープライズクラスの RDBMS です。金融、オンラインビジネス、その他の技術領域で幅広く使用されています。Sybase ASE プラットフォームは、Adaptive Server、Monitor Server、Backup Server、その他のプラグインコンポーネントを含む多くの標準コンポーネントで構成されています。Adaptive Server コンポーネントは、リレーショナルデータベースサーバです。Monitor Server は、Adaptive Server データベースサーバとは独立したサーバであり、Adaptive Server を監視します。Monitor Server は、リアルタイムデータまたはヒストリカルデータをクライアントアプリケーションに提供できます。Backup Server は、Open Server ベースのアプリケーションで、Adaptive Server でのデータベースのバックアップ (ダンプ) とリストア (ロード) をすべて管理します。

SIOS Protection Suite for Linux Sybase ASE Recovery Kit は、Sybase ASE コンポーネントの Adaptive Server、Monitor Server、Backup Server に対して SPS によるリソース保護を提供します。

## SIOS Protection Suite/LifeKeeper ドキュメンテーション

以下は、SIOS Technology Corp. のドキュメンテーション Web サイトで提供されている SIOS Protection Suite for Linux 関連ドキュメントの一覧です。

- SPS for Linux リリースノート
- SPS for Linux テクニカルドキュメンテーション
- Optional Recovery Kit ドキュメンテーション

## Sybase ASE のドキュメンテーション

*Installation Guide Adaptive Server for Linux*、*User Manual* 、 *Monitor Server User Manual* 、*Troubleshooting Guide* 、Reference Manual を含む Sybase ASE ドキュメンテーションは、Web サイトの次の場所にあります。

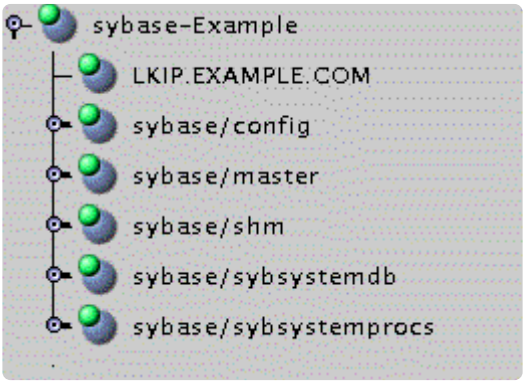
Sybase 製品のドキュメンテーション

# 6.20.1. Sybase ASE Recovery Kit の概要

LifeKeeper for Linux Sybase ASE Recovery Kit は、LifeKeeper 内の Sybase ASE Server インスタンスを保護する機能を提供します。Sybase ASE ソフトウェア、LifeKeeper Core、および Sybase ASE Recovery Kit は、クラスタ内の 2 台以上のサーバにインストールされます。Sybase ASE サーバインスタンスが LifeKeeper の保護下に入ったら、クライアントは LifeKeeper が保護する IP アドレスを使用してデータベースに接続します。LifeKeeper が保護する IP アドレスは、Sybase ASE リソース階層の作成前に別に作成する必要があります。Sybase ASE リソース階層の作成では、親 Sybase ASE リソースインスタンスと子 IP アドレスリソース間に依存関係を作成します。Sybase ASE サーバインスタンスに障害が発生した場合、LifeKeeper はまずローカルサーバ上でのリカバリを試みます。ローカルリカバリに失敗した場合、バックアップサーバにフェイルオーバします。

## Sybase ASE リソース階層

以下に示すのは、Sybase ASE リソース階層の一例です。




上記の例の依存関係は、以下の保護されるリソースに対応しています。

| リソース | **Sybase ASE** ソフトウェアコンポーネント |
|---|---|
| LKIP.EXAMPLE.COM | クライアント接続に使用される切り替え可能な IP アドレスを保護します。 |
| sybase/config | Sybase Adaptive Server、Monitor Server、Backup Serverの設定ファイルが格納されたファイルシステムを保護します。 |
| sybase/master | Sybase ASE マスターデバイスを保護します。 |
| sybase/shm | Sybase Adaptive Server と Monitor Server 共有メモリパスを保護します。 |
| sybase/sybsystemdb | Sybase ASE sybsystemdb デバイスを保護します。 |
| sybase/ sybsystemprocs | Sybase ASE sybsystemprocs デバイスを保護します。 |

フェイルオーバが発生した場合、LifeKeeper はバックアップサーバでファイルシステム、IP アドレス、およびデータベースリソース (すべてのリソース依存関係を含む) を In Service の状態にします。クライアントは切断され、サーバに再接続する必要があります。まだコミットされていない SQL ステートメントはす

べて再入力する必要があります

# 6.20.2. Sybase ASE Recovery Kit ハードウェア要件とソフトウェア要件

## Sybase ASE Recovery Kit の要件

以下に、Sybase ASE Recovery Kit を適切に設定、インストール、運用する上で必要なハードウェアとソフトウェアを挙げます。

LifeKeeper for Linux Sybase Recovery Kit をインストールする前に、LifeKeeper の構成が次の要件を満たしている必要があります。LifeKeeper のハードウェアとソフトウェアのインストールと設定に関する具体的な手順については、 LifeKeeper for Linux Installation Guide を参照してください。

## ハードウェア要件

- サーバ - サーバは、要件にしたがって設定する必要があります。要件は、 LifeKeeper for Linux テクニカルドキュメンテーション 、および SPS for Linux リリースノート に記載されています。

- **IP** ネットワークインターフェースカード - 各サーバは、イーサネット TCP/IP をサポートするネットワークインターフェースカードを 1 つ以上必要とします。ただし、LifeKeeper クラスタは、少なくとも 2 つのコミュニケーションパスを必要とすることに注意してください。LAN を使用した 2 つのコミュニケーションパスは、ハートビート用に別々のサブネットに接続することをお勧めします。また、少なくともどちらか 1 つのサブネットは、プライベートネットワークとして設定します。TCP と TTY のハートビートの組み合わせもサポートされています。

- ストレージ – サーバは、LifeKeeper でサポートされている共有ストレージ、もしくは LifeKeeper for Linux Data Replication ストレージを使用するように設定する必要があります。

## ソフトウェア要件

- **TCP/IP** ソフトウェア – LifeKeeper 環境の各サーバには TCP/IP ソフトウェアが必要です。

- **Sybase ASE** ソフトウェア – LifeKeeper は、 Sybase ASE ソフトウェアのバージョン 15.5 およびそれ以降をサポートしています。このバージョンは、Sybase Inc. の http://www.sap.com/japan/index.html から入手できます。 **注記:** クラスタ内の全サーバに同じバージョンの Sybase ASE ソフトウェアをインストールする必要があります。また、LifeKeeper の保護下にあるサーバ上にインストール可能な Sybase ASE ソフトウェアバージョンは 1 つのみです。

- **LifeKeeper** ソフトウェア – クラスタ内の各サーバにインストールする LifeKeeper ソフトウェアのバージョン、および、適用する LifeKeeper ソフトウェアパッチのバージョンが同一であることが絶対条件です。

- **LifeKeeper for Linux IP Recovery Kit** – LifeKeeper for Linux IP Recovery Kit は、LifeKeeper for Linux Sybase ASE Recovery Kit により必要とされます。LifeKeeper for Linux IP Recovery Kit は、FTP ダウンロードにより LifeKeeper for Linux Core イメージファイルで提供されます。

- **LifeKeeper for Linux Sybase ASE Recovery Kit** – Sybase ASE Recovery Kit は、FTP ダウンロードによりイメージファイルで提供されます。Red Hat Package Manager (rpm) でパッケージされており、rpm を使用してインストール / アンインストールを行います。LifeKeeper for Linux Sybase ASE Recovery Kit の FTP ダウンロードによるイメージファイルには、次の rpm ファイルが格納されています。

  **steeleye-lkSYBASE**

# 6.20.3. Sybase ASE Recovery Kit 設定上の考慮事項

以下は、Sybase ASE Recovery Kit のインストールと設定を行う前に考慮すべき情報です。

---

raw I/O の使用

DK4L でのミラーファイルシステムの使用

interfaces ファイルの考慮事項

Sybase Software Asset Manager (SySAM)

アクティブ / スタンバイの考慮事項

アクティブ / アクティブの考慮事項

Sybase Monitor Server と Backup Server の考慮事項

Network Attached Storage (NAS) を使用した LifeKeeper 保護下の Sybase

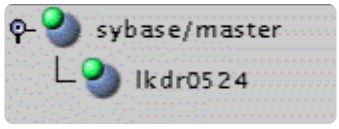
# 6.20.3.1. raw I/O の使用

raw デバイスで Sybase ASE を使用することを予定している場合は、LifeKeeper Raw I/O Recovery Kit を LifeKeeper Core イメージファイルからインストールする必要があります。また、使用前に raw I/O デバイスを適切に設定する必要もあります。手順については、raw I/O によるデータベースデバイスの作成 を参照してください。

# 6.20.3.2. DataKeeper でのレプリケーションファイルシステムの使用

Sybase ASE Recovery Kit では、SIOS DataKeeper をデバイス空間として使用できます。また、Sybase ASE ソフトウェアをレプリケーションファイルシステムにインストールできます。

例えば、Sybase ASE リソースの依存ファイルシステムは次のようになります。下図は、DataKeeper リソースミラーで、システムデバイス空間のファイルシステムとその依存関係を示しています。




レプリケーション（SIOS DataKeeper）ファイルシステムのリソースはSybase ASEリソースを作成する前に作成する必要があります。

# 6.20.3.3. interfaces ファイルの考慮事項

Sybase ASE Recovery Kit は、Sybase ASE interfaces ファイルを使用して、クライアントの IP アドレスとポートを検出します。このファイルは $SYBASE 配下にあり、通常、interfaces と呼ばれます。このファイルは、srvbuild または同様の設定ユーティリティを使用して Adaptive Server、Monitor Server、または Backup Server インスタンスが作成されるたびに更新されます。LifeKeeper for Linux Sybase ASE Recovery Kit では、このファイルに保護対象の各 Sybase ASE コンポーネントに対するエントリが含まれることを必要とします。コメント行は使用できません。interfaces ファイルに記述するすべてのサーバ名は、有効な仮想 IP アドレスに解決可能である必要があります。Sybase ASE リソース階層を保護するすべてのサーバは、interfaces ファイルに記述されているサーバ名を解決できる必要があります。また、仮想 IP アドレスをサーバ名の代わりに使用することを推奨します。

例

- master tcp ether example.com 4100
- query tcp ether example.com 4100

**Example_back**

- master tcp ether example.com 4200
- query tcp ether example.com 4200

**Example_mon**

- master tcp ether example.com 4200
- query tcp ether example.com 4200

*interfaces ファイルのサンプル*

# 6.20.3.4. Sybase Software Asset Manager (SySAM)

Sybase Software Asset Management (SySAM) を使用すると、ライセンスされた Sybase 製品を管理できます。Sybase ASE サーバの起動時に、各 ASE サーバコンポーネントは、特定の機能を実行するための権限があるかどうか、その環境のライセンスファイルを確認します。ASE サーバがこの作業を実行するためには、ライセンスマネージャとベンダモジュールが実行中である必要があります。LifeKeeper for Linux Sybase ASE Recovery Kit は、SySAM ライセンスマネージャに対する保護を提供していません。ライセンスマネージャは、冗長化されたサーバシステム内に設定することを推奨します。冗長化されたサーバシステムでは、冗長化されたライセンスにより、ローカルサーバをキューの 1 番目のライセンスサーバとして指定し、リモートサーバをバックアップライセンスサーバに指定することができます。SySAM アプリケーションは、まず 1 番目のサーバのライセンスファイルからライセンスをチェックアウトしようとします。何らかの理由でそのサーバに障害が発生した場合、リストの 2 番目にあるサーバにアクセスし、以下同様に実行されます。冗長化されたライセンスサーバ環境のユーザプロファイルで、LM_LICENSE_FILE 変数を正しく設定する必要があります。

# 6.20.3.5. アクティブ / スタンバイの考慮事項

アクティブ / スタンバイ設定では、バックアップサーバは Sybase ASE をアクティブに実行せず、プライマリサーバに障害が発生した場合に備えてスタンバイします。以下に、アクティブ / スタンバイ設定で Sybase ASE リソースインスタンスを保護する場合に従わなければならない具体的な要件を示します。

## シナリオ 1

Sybase ASE 製品は、クラスタ内のすべてのサーバにローカルでインストールされている。

- すべての Sybase Adaptive Server、Monitor Server、Backup Server デバイスは、共有ストレージ上に設定されている。
- Sybase Adaptive Server、Monitor Server、Backup Server の設定ファイルは、共有ファイルシステム上に格納されている。
- Sybase Adaptive Server と Monitor Server の共有メモリディレクトリは、共有ファイルシステム上に置かれている。
- interfaces ファイルは、保護対象の各インスタンスに対する共通のエントリを含むよう、すべてのサーバ上で手作業で更新しなければならない。
- すべての interfaces ファイルエントリは、リソースが保護されるすべてのサーバ上で解決可能でなければならない。
- 各 Adaptive Server、Monitor Server、Backup Server 用の RUN ファイルは、クラスタ内のすべてのサーバ上に存在しなければならない。
- 各 Adaptive Server、Monitor Server、Backup Server 用の RUN ファイルは、クラスタ内のすべてのサーバ上で実行可能でなければならない。
- 各 Adaptive Server、Monitor Server、Backup Server 用の RUN ファイルは、クラスタ内のすべてのサーバ上で同一のオプションを含んでいなければならない。

## シナリオ 2

Sybase ASE 製品は、プライマリサーバ上の 1 つまたは複数の共有ファイルシステムにインストールされている。

- すべての Sybase Adaptive Server、Monitor Server、Backup Server デバイスは、共有ストレージ上に設定されている。
- Sybase Adaptive Server、Monitor Server、Backup Server の設定ファイルは、共有ファイルシステム上に格納されている。
- Sybase Adaptive Server と Monitor Server の共有メモリディレクトリは、共有ファイルシステム上

に置かれている。

- interfaces ファイルは、ターゲットサーバで更新する必要はない。
- すべての interfaces ファイルエントリは、リソースが保護されるすべてのサーバ上で解決可能でなければならない。
- LifeKeeper Backup Server において、/etc/ld.so.conf にSybase 製品ライブラリのエントリを追加して更新する必要がある。
    - º $SYBASE/ASE/lib のエントリを追加する。
    - º $SYBASE/OCS/lib のエントリを追加する。
    - º Sybase

ASE インストール済み製品を含む共有ファイルシステムをマウントして、ldconfig を実行する。

# 6.20.3.6. アクティブ / アクティブの考慮事項

アクティブ / アクティブ設定では、障害時、他の LifeKeeper サーバに対するバックアップとして各サーバがアクティブに 1 つまたは複数の Sybase ASE サーバを実行します。以下に、アクティブ / アクティブ設定で Sybase ASE リソースインスタンスを保護する場合に、順を追って従わなければならない具体的な要件を示します。

## シナリオ 1

Sybase ASE 製品は、クラスタ内のすべてのサーバにローカルでインストールされている。

- すべての Sybase Adaptive Server、Monitor Server、Backup Server デバイスは、共有ストレージ上に設定されている。
- Sybase Adaptive Server、Monitor Server、Backup Server の設定ファイルは、共有ファイルシステム上に格納されている。
- Sybase Adaptive Server と Monitor Server の共有メモリディレクトリは、共有ファイルシステム上に置かれている。
- interfaces ファイルは、保護対象の各インスタンスに対する共通のエントリを含むよう、すべてのサーバ上で手動で更新しなければならない。
- すべての interfaces ファイルエントリは、リソースが保護されるすべてのサーバ上で解決可能でなければならない。
- 各 Adaptive Server、Monitor Server、Backup Server 用の RUN ファイルは、クラスタ内のすべてのサーバ上に存在しなければならない。
- 各 Adaptive Server、Monitor Server、Backup Server 用の RUN ファイルは、クラスタ内のすべてのサーバ上で実行可能でなければならない。
- 各 Adaptive Server、Monitor Server、Backup Server 用の RUN ファイルは、クラスタ内のすべてのサーバ上で同一のオプションを含んでいなければならない。

## シナリオ 2

Sybase ASE 製品は、プライマリサーバ上の 1 つまたは複数の共有ファイルシステムにインストールされている。

- すべての Sybase Adaptive Server、Monitor Server、Backup Server デバイスは、共有ストレージ上に設定されている。
- Sybase Adaptive Server、Monitor Server、Backup Server の設定ファイルは、共有ファイルシステム上に格納されている。

- Sybase Adaptive Server と Monitor Server の共有メモリディレクトリは、共有ファイルシステム上に置かれている。
- interfaces ファイルは、ターゲットサーバで更新する必要はない。
- すべての interfaces ファイルエントリは、リソースが保護されるすべてのサーバ上で解決可能でなければならない。
- LifeKeeper Backup Server において、/etc/ld.so.conf にSybase 製品ライブラリのエントリを追加して更新する必要がある。
    - º $SYBASE/ASE/lib のエントリを追加する。
    - º $SYBASE/OCS/lib のエントリを追加する。
    - º Sybase

ASE インストール済み製品を含む共有ファイルシステムをマウントして、ldconfig を実行する。

# 6.20.3.7. Sybase Monitor Server と Backup Server

注記 ：Sybase Monitor Server は Sybase ASE ARK v9.0.2 およびそれ以降のバージョンではサポートされません。

LifeKeeper for Linux Sybase ASE Recovery Kit は、Adaptive Server、Backup Server、Monitor Server コンポーネントにリソース保護を提供します。ただし、Backup Server コンポーネントと Monitor Server コンポーネントは、リソース階層の必須コンポーネントではありません。Sybase Backup Server と Sybase Monitor Server は、リソース保護から除外できます。Sybase Monitor Server または Sybase Backup Server の保護が必要ない場合、リソース階層の作成中に該当するコンポーネントの選択画面で「none」を選択できます。GUI によるリソース作成中に「none」を選択すると、選択されたコンポーネントがリソース階層の保護から除外されます。 **注記:**「none」は、Sybase ASE Recovery Kit の予約語です。そのため、Sybase Backup Server と Sybase Monitor Server のいずれも、「none」という名前を付けることはできません。

これらのコンポーネントを保護するかどうかを選択する際は、Adaptive Server の設定ファイル、デバイスパス、ログパス、または共有メモリディレクトリとファイルシステムを共有する設定ファイルが、LifeKeeper によって保護されることに注意することが重要です。1 つまたは複数のコンポーネントを LifeKeeper の保護から除外する場合は、保護するコンポーネントと保護しないコンポーネントが共有されないようにファイル配置を考慮する必要があります。

# 6.20.3.8. Network Attached Storage の使用

クラスタストレージとして NFS ファイルサーバ (Network Attached Storage) を使用する場合、LifeKeeper の設定においていくつかの特別な考慮事項があります。

## NAS Recovery Kit の使用

LifeKeeper for Linux で NFS サーバを共有ストレージアレイとして使用する場合、オプションの Network Attached Storage (NAS) Recovery Kit が必要になります。各クラスタノードに NAS Recovery Kit (およびライセンス) をインストールしてください。詳細については、NAS Recovery Kit のドキュメンテーションを参照してください。

### エラーメッセージ

Sybase ASE で Network Attached Storage (NAS) を使用する際、システムクラッシュによるフェイルオーバ後、Sybase が再開できないといった事象に遭遇する可能性があります。Sybase エラーログを確認すると、エラーの原因が分かります。

#### Sybase ASE 15.x

```
00:00:00000:00000:2011/05/09 16:08:51.66 kernel Adaptive Server
Enterprise(Developer Edition)
00:00:00000:00000:2011/05/09 16:08:51.66 kernel basis_dlock: file
'/s10/sybase-data155/data/master.dat' already in use by an ASE
00:00:00000:00000:2011/05/09 16:08:51.66 kernel kdconfig: unable to
read primary master device
00:00:00000:00000:2011/05/09 16:08:51.66 server kiconfig: read of
config block failed
```

これは、LifeKeeper によって制御されている NFS ファイルシステム上のファイル「*master.dat*」に、Sybase データサーバによって NFS ロックが設定されたことを示しています。システムクラッシュによりロックがクリアされなかったので、LifeKeeper はデータサーバを In Service の状態に戻すことができません。Sybase は何か別のプロセスが *master.dat* ファイルを使用しているとみなします。

### 解決方法

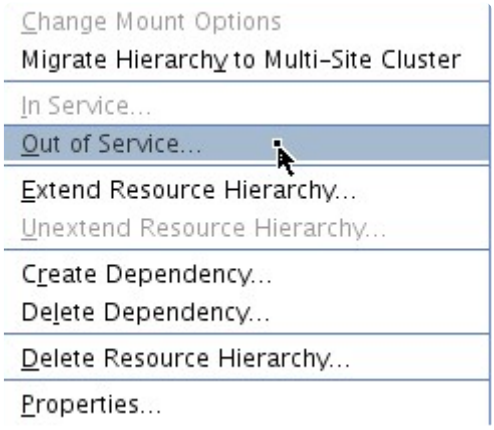
この事象を解決するためには、ファイルシステムリソースを作成する前に、NFS のオプション「nolock」を付けて *master.dat* を格納する予定の NFS ファイルシステムをマウントしてください。デフォルトでは、NFS はファイルのロックを有効にします。「nolock」オプションがリソース作成前に使用されると、LifeKeeper はこのオプションを認識し、ファイルシステムリソースを In Service の状態にするたびにこのオプションを使用します。LifeKeeper は (クラスタノードから) *master.dat* を含んだファイルシステムへのアクセスを制御するので、通常、ロックは重要ではありません。テスト中に使用される NFS マウントオプションは「`rw,sync,tcp,nfsvers=3,noac,nolock`」です。

Sybase ASE のバイナリが格納されているファイルシステムなど、Sybase リソース階層によって使用さ

れるその他のファイルシステム上では、「nolock」を使用する必要はありません。

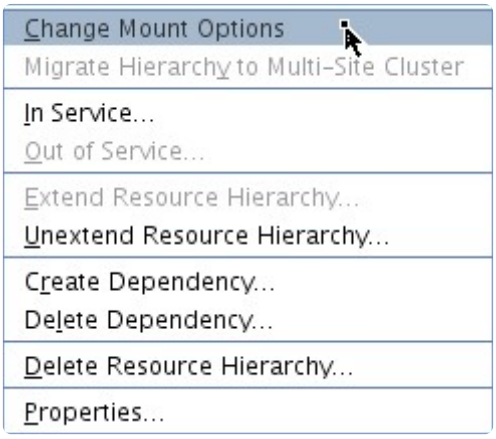
NAS ファイルシステムリソースがすでに「nolock」オプション設定なしに作成されている場合は、以下の手順に従ってマウントオプションを変更してください。

1. LifeKeeper GUI を使用して変更する必要があるファイルシステムリソースに対して Out of Service 操作を行ってください。この操作は LifeKeeper GUI でファイルシステムリソース上にポインタを置き、右クリックしてドロップダウンメニューから [Out of Service] を選択することで可能です。この操作は親リソースも同じく Out of Service 状態にします。

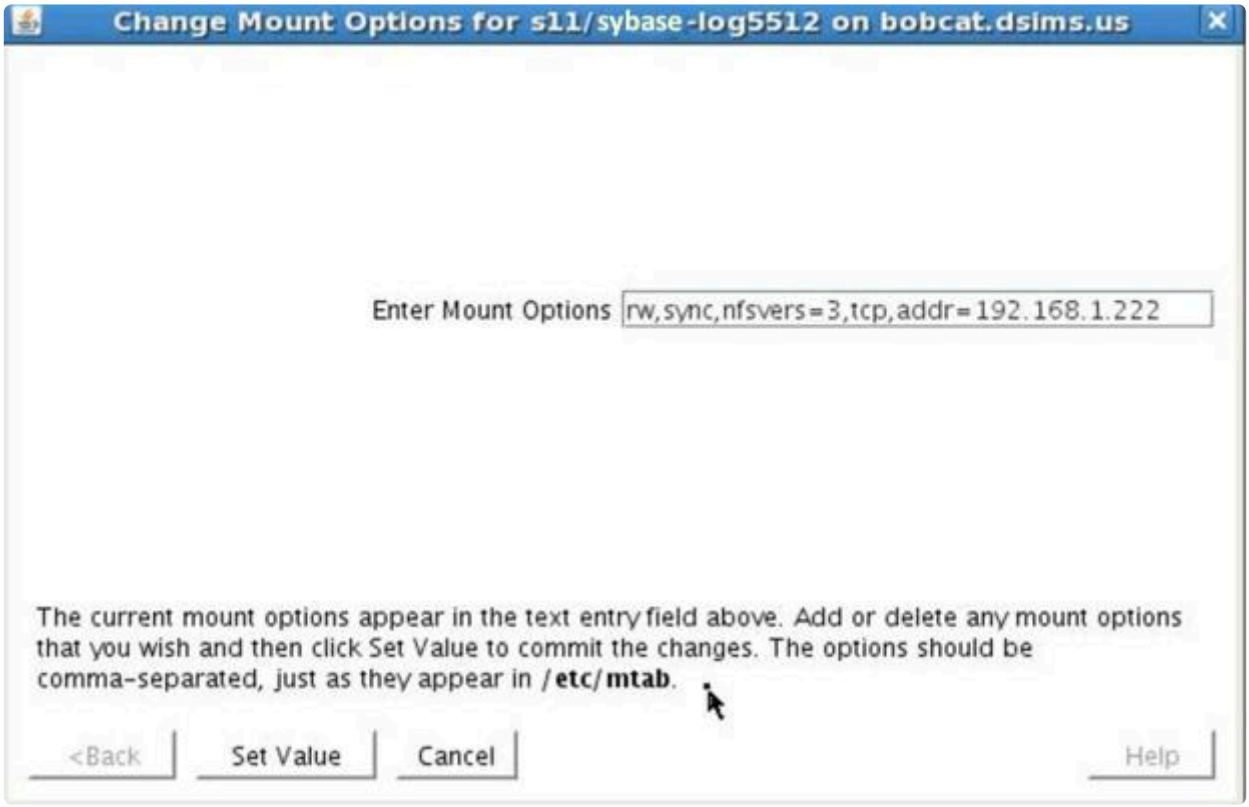



2. [Out of Service] 操作を実行し、プロセスを完了させてください。

3. 一旦ファイルシステムリソースが Out of Service 状態になったら、ポインタをリソース上に置き右クリックし、ドロップダウンメニューから [Change Mount Options] を選択してください。

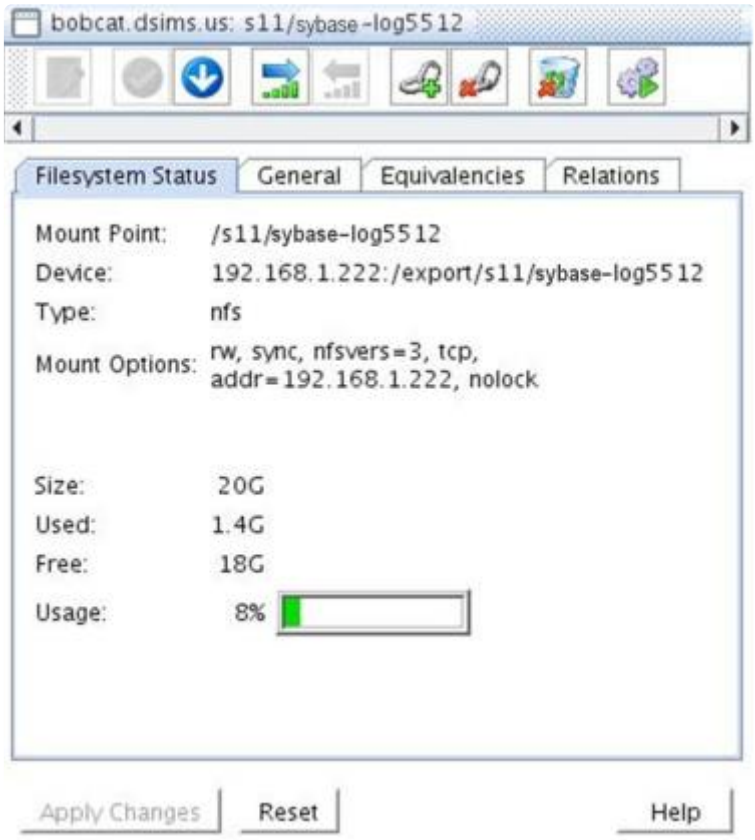



4. ポップアップウィンドウで、オプションの行に「nolock」を追加し、[Set Value] をクリックしてください。クラスタの各ノードで手順 3 および 4 を繰り返す必要があります。




5. マウスを右クリックし、[In Service] を選択し、NAS ファイルシステムリソースを In Service の状態に戻してください。

6. ファイルシステムリソースのプロパティパネルが、「nolock」が現在のマウントオプションであることを表示します。



# 6.20.4. LifeKeeper を使用した Sybase ASE のインストールと設定

## Sybase ASE と LifeKeeper のインストール / 設定

本セクションでは、Sybase ASE ソフトウェアと LifeKeeper ソフトウェアのインストールおよび設定方法を説明します。

Sybase ASE 製品と LifeKeeper ソフトウェアのインストールと設定には以下の手順をお勧めします。各ステップは、詳細作業にリンクしています。

Sybase ASE ソフトウェアのインストール

Sybase ASE サーバの作成

LifeKeeper ソフトウェアのインストール

これらの作業を実行すると、LifeKeeper リソース階層を作成して Sybase ASE サーバを保護できるようになります。

## リソース設定作業

階層の作成、拡張、削除、拡張解除など、LifeKeeper GUI で実行可能なさまざまな機能を説明します。

前のセクションで説明したセットアップ作業を終了したら、Sybase ASE リソース階層を作成して拡張できるようになります。

LifeKeeper for Linux Sybase ASE Recovery Kit の設定に関して、以下の作業を行うことができます。

- Create Resource Hierarchy - Sybase ASE リソース階層を作成します。
- Delete Resource Hierarchy - Sybase ASE リソース階層を削除します。
- Extend Resource Hierarchy - Sybase ASE リソース階層をプライマリサーバからバックアップサーバに拡張します。
- Unextend Resource Hierarchy - LifeKeeper クラスタ内の 1 つのサーバから Sybase ASE リソース階層を拡張解除 (削除) します。
- Testing Your Resource Hierarchy - Sybase ASE リソース階層をテストします。

LifeKeeper Core のリソース階層 (ファイルシステムや IP リソースなど) の設定方法については、SPS for Linux テクニカルドキュメンテーション の GUI Administrative Tasks セクションを参照してください。

以下の作業については、すべての Recovery Kit で同じ手順を使用する共通の作業であるため、 SPS for Linux テクニカルドキュメンテーション の 管理 セクションを参照してください。

- Create a Resource Dependency 。既存のリソース階層と別のリソースインスタンスとの間に親子の依存関係を作成し、クラスタ内のすべての対象サーバに依存関係の変更を反映します。
- Delete a Resource Dependency 。リソースの依存関係を削除して、クラスタ内のすべての対象サーバに依存関係の変更を反映します。
- In Service 。リソース階層を特定のサーバで In Service の状態にします。
- Out of Service 。リソース階層を特定のサーバで Out of Service の状態にします。
- View / Edit 特定のサーバでリソース階層のプロパティを表示または編集します。

**注記:** このセクションの残りの部分では、設定作業はすべて **[Edit]** メニューから実行されます。この設定作業の大半は次の方法でも実行できます。

- ツールバーから
- 状況表示の左側ペインにあるグローバルリソースを右クリックする
- 状況表示の右側ペインにあるリソースを右クリックする

右クリックの方法を使用すると、 **[Edit]** メニューを使用するときに要求される情報を入力せずにすみます。

# 6.20.4.1. Sybase ASE ソフトウェアのインストール

Sybase ASE ソフトウェアは、クラスタ内の全サーバに同一のパラメータ / 設定を使用してインストールする必要があります。詳細については、Installation Guide Adaptive Server for Linux を参照してください。以下は、LifeKeeper を Sybase ASE と連携させるための追加の推奨事項です。

- 非 root のシステムユーザ (Sybase OS ユーザ) がすべてのサーバ上に存在する必要があります。そのユーザは、リソースが保護されるすべてのサーバ上で同じユーザ ID、グループ ID、ホームディレクトリを持たなければなりません。
- Sybase ASE 共通ソフトウェアパッケージがインストールされている必要があります。このパッケージは、Sybase `srvbuild` と Sybase `isql` ユーティリティの両方を提供します。
- Sybase ASE リソース階層を含む各 LifeKeeper サーバは、 *$SYBASE/interfaces* ファイル内に Sybase ASE サーバについて同一のサービスエントリを持つ必要があります。
- *$SYBASE/ASE-<version>* と *$SYBASE/ASE* 間にリンクが存在することを確認してください。リンクが存在しない場合、手動で作成する必要があります。補足情報については、ASE と OCS のリンクの作成 を参照してください。
- $SYBASE/OCS-<version> と $SYBASE/OCS 間にリンクが存在することを確認してください。リンクが存在しない場合、手動で作成する必要があります。補足情報については、ASE と OCS のリンクの作成 を参照してください。
- Adaptive Server、Monitor Server、Backup Server に共有メモリパラメータを設定する方法の詳細については、 *Installation Guide Adaptive Server for Linux* を参照してください。
- データベースデバイスは LifeKeeper にて共有ストレージとして保護されている必要があります。また設定ファイルなども LifeKeeper により保護されている共有ストレージ上に置く必要があります。詳細は、Sybase ASE サーバの作成 も参照してください。
- Sybase ASE 共通ソフトウェアパッケージは、LifeKeeper により保護されている共有ストレージ上にインストールするか、クラスタ内の全サーバにてローカル領域の同一パスにインストールしてください。

# 6.20.4.2. Sybase ASE サーバの作成

注記 ：Sybase Monitor Server は Sybase ASE ARK v9.0.2 およびそれ以降のバージョンではサポートされません。

Sybase の Adaptive Server、Monitor Server、Backup Server を設定するには、お手元の Installation Guide Adaptive Server for Linux を参照してください。次の考慮事項に従う必要があります。

- srvbuild ユーティリティもしくはその他の Sybase ASE ユーティリティを使用して Sybase Adaptive Server インスタンスを作成する
    - すべてのシステムデバイスを共有ストレージ上で設定する
    - Adaptive Server 設定ファイルを共有ストレージ上で設定する
    - Adaptive Server 共有メモリディレクトリを共有ストレージ上で設定する
    - LifeKeeper の切り替え可能 IP アドレスを使用するためのインターフェースを設定する
    - オプションで、共有ストレージ上にログを設定する
- 必要に応じて、Sybase Monitor Server インスタンスを作成する
    - すべてのシステムデバイスを共有ストレージ上で設定する
    - Monitor Server 設定ファイルを共有ストレージ上で設定する
    - Monitor Server 共有メモリディレクトリを共有ストレージ上で設定する
    - LifeKeeper の切り替え可能 IP アドレスを使用するためのインターフェースを設定する
    - オプションで、共有ストレージ上にログを設定する
- 必要に応じて、Sybase Backup Server インスタンスを作成する
    - すべてのシステムデバイスを共有ストレージ上で設定する

- º Monitor Server 設定ファイルを共有ストレージ上で設定する
- º Monitor Server 共有メモリディレクトリを共有ストレージ上で設定する
- º LifeKeeper の切り替え可能 IP アドレスを使用するためのインターフェースを設定する
- º オプションで、共有ストレージ上にログを設定する

# 6.20.4.3. LifeKeeper ソフトウェアのインストール

Sybase ASE ソフトウェアをインストールし、データベースサーバを作成したら、LifeKeeper Core ソフトウェアと LifeKeeper for Linux IP Recovery Kit および必要なパッチをインストールし、続いて Sybase ASE Recovery Kit をインストールできます。raw デバイスで Sybase ASE を使用する予定がある場合は、LifeKeeper Raw I/O Recovery Kit を LifeKeeper Core イメージファイルからインストールする必要があります。raw デバイスの設定の要件および方法については、「raw I/O によるデータベースデバイスの作成」を参照してください。

LifeKeeper パッケージのインストール方法については、SPS for Linux インストールガイド を参照してください。

# 6.20.4.4. Sybase ASE リソース階層の作成

注記: プライマリサーバ上でSybase ASE が起動していることを確認してください。

プライマリサーバで次の手順を実行します。

1. [Edit] メニューの [Server] から [Create Resource Hierarchy] を選択します。

   [Create Resource Wizard] ダイアログボックスが表示されます。

2. ドロップダウンリストから [Sybase ASE Database] を選択し、[Next] をクリックしてください。

3. 以下の情報を入力するように要求されます。ダイアログボックスで [Back] ボタンが有効な場合は、前のダイアログボックスに戻ることができます。これは、エラーが発生して、前に入力した情報を修正する必要がある場合に便利な機能です。いつでも [Cancel] をクリックして、作成処理全体を取り消すことができます。

| フィールド | ヒント |
|---|---|
| **Server** | Sybase ASE リソースを作成する LifeKeeper サーバを選択します。 |
| **Switchback Type** | intelligent または automatic を選択します。フェイルオーバ後、バックアップサーバで Sybase ASE リソースが In Service (アクティブ) になるとき、プライマリサーバにスイッチバックされる方法が決まります。インテリジェントスイッチバック (intelligent) の場合、リソースをプライマリサーバにスイッチバックするときに管理者の介入が必要ですが、自動スイッチバック (automatic) の場合は、プライマリサーバがオンラインに戻り、LifeKeeper のコミュニケーションパスが再確立されるとすぐにスイッチバックが実行されます。<br><br>注記: スイッチバック方法は、Sybase ASE リソースが使用する依存リソースのスイッチバック方法と一致する必要があります。 |
| **Sybase Install Directory** | Sybase ASE 製品のインストール場所を指定します。他のディレクトリパスを入力することもできます。パス名に対する有効な文字は、アルファベット、数字および以下の特別な記号になります。- _ ./ |
| **Sybase Instance** | このフィールドはSybaseデータディレクトリを含むディレクトリパスの特定に使われます。データディレクトリは通常、ASE-<version>/RUN_* ファイルをインスタンスとして含みま |

| | |
|---|---|
| **Directory** | す。 |
| **Sybase Instance** | システムで最初に検出された、LifeKeeper 階層が存在しない Sybase インスタンスがデフォルトで入ります。ドロップダウンリストには、LifeKeeper サーバ上で利用可能なその他の Sybase インスタンスが表示されます。このフィールドは LifeKeeper の保護下に置かれる Sybase ASE Database インスタンスの指定に使われます。指定対象となるインスタンスは存在し稼働していなくてはなりません。 |
| **Sybase Username** | このフィールドは Sybase System Administrator のユーザー名の入力に使われます。デーフォールトのユーザー名は sa です。この System Administrator はログインしており、保護される Sybase Adaptive Server 上のすべてのデータベースに対する特権を付与されている必要があります。 |
| **Sybase Login Password** | このフィールドは Sybase System Administrator 用のパスワードの指定に使われます。 |
| **Sybase Backup Server** | 指定された Adaptive Server インスタンスに対する Sybase Backup Server を指定します。この Sybase Backup Server は、LifeKeeper の保護下に置かれます。Sybase Backup Server を LifeKeeper の保護下に置く必要がない場合、「none」を選択できます。 |
| **Sybase ASE Database Tag** | プライマリサーバ上の新しい Sybase ASE データベースリソースに対する固有のタグ名です。デフォルトのタグ名は語「sybase」の後にAdaptive Server インスタンスの名前を付けたものです。別の固有のタグ名を入力することもできます。タグ名に対する有効な文字は、アルファベット、数字および以下の特別な記号になります。- _ ./ |

4. **[Next]** をクリックしてください。 **[Create Resource Wizard]** が表示され、Sybase ASE リソース階層が作成されます。LifeKeeper によって入力データが検査されます。問題が検出された場合、情報ボックスにエラーメッセージが表示されます。

5. Sybase ASE リソース階層が正常に作成されたこと、フェイルオーバによる保護を実現するにはその階層をクラスタ内の別のサーバに拡張する必要があることを示すメッセージが表示されます。 **[Next]** をクリックします。

6. **[Continue]** をクリックします。 **[Pre-extend Wizard]** が起動されます。リソース階層を別のサーバに拡張する方法の詳細については、「Sybase ASE リソース階層の拡張」の **手順 2** を参照してください。

# 6.20.4.5. Sybase ASE リソース階層の拡張

この操作は [Edit] メニューから開始できます。または、[Create Resource Hierarchy] オプションが完了すると自動的に開始されます。その場合は下の手順 2 を参照してください。

1. [Edit] メニューの [Resource] から [Extend Resource Hierarchy] を選択します。[Pre-Extend Wizard] が表示されます。拡張操作に慣れていない場合は、[Next] をクリックします。LifeKeeper のリソース階層の拡張作業のデフォルト値が分かっていて、入力と確認を省略する場合は、[Accept Defaults] をクリックします。

2. [Pre-Extend Wizard] で以下の情報を入力します。

   **注記:** 最初の 2 つのフィールドは [Edit] メニューの [Extend] から操作を開始した場合にだけ表示されます。

| フィールド | ヒント |
|---|---|
| **Template Server** | Sybase ASE リソースが現在 In Service のサーバを選択します。 |
| **Tag to Extend** | 拡張する Sybase ASE リソースを選択します。 |
| **Target Server** | 拡張先のサーバを入力または選択します。 |
| **Switchback Type** | フェイルオーバ後、バックアップサーバで Sybase ASE リソースが In Service (アクティブ) になるとき、プライマリサーバにスイッチバックされる方法が決まります。intelligent または automatic を選択できます。スイッチバックタイプは、必要な場合 [Resource Properties] ダイアログボックスの [General] タブで後から変更できます。<br><br>**注記**: スイッチバック方法は、Sybase ASE リソースが使用する依存リソースのスイッチバック方法と一致する必要があることに注意してください。 |
| **Template Priority** | **テンプレートの優先順位** を選択または入力します。これはサーバで現在 In Service の Sybase ASE 階層の優先順位です。優先順位は、1 ~ 999 の範囲で未使用の値が有効で、小さい数字ほど優先順位が高くなります (数字 1 が最高の優先順位に相当)。拡張処理時に、別のシステムですでに使用中の優先順位をこの階層に対して指定することはできません。デフォルト値を推奨します。<br><br>**注記**: このフィールドは階層を最初に拡張するときだけ表示されます。 |
| **Target Priority** | これは、新しく拡張する Sybase ASE 階層が、他のサーバに存在する同等の階層に対して持つ相対的な優先順位です。1 ~ 999 の範囲で、まだ優先順位として使用されていない値が有効で、リソースのカスケーディングフェイルオーバシーケンスにおけるサーバの優先順位を示します。LifeKeeper のデフォルトでは、階層が作成されたサーバに「1」が割り当てられる |

| | |
|---|---|
| | ことに注意してください。優先順位は連続している必要はありませんが、特定のリソースについて 2 つのサーバに同じ優先順位を割り当てることはできません。 |

3. 拡張前のチェックが正常に終了したというメッセージが表示されたら、Next をクリックします。

4. 拡張する階層に応じて、拡張されるリソースタグ (一部編集不可) を示す一連の情報ボックスが表示されます。

5. [Extend Wizard] で以下の情報を入力します。

| | |
|---|---|
| **Sybase ASE Install Directory** | デフォルトでは、テンプレートリソースの Sybase ASE インストールパスが入ります。有効な Sybase ASE インストールパスを指定する必要があります。パス名に対する有効な文字は、アルファベット、数字および以下の特別な記号になります。- _ ./ |
| **Sybase ASE Database Tag** | プライマリサーバ上の新しい Sybase ASE データベースリソースに対する固有のタグ名です。デフォルトのタグ名は語「sybase」の後にAdaptive Server インスタンスの名前を付けたものです。別の固有のタグ名を入力することもできます。タグ名に対する有効な文字は、アルファベット、数字および以下の特別な記号になります。- _ ./ |

6. 「Hierarchy extend operations completed (階層の拡張操作が完了しました)」というメッセージが表示されたら、階層を別のサーバに拡張する場合は、[Next Server] をクリックし、拡張作業が必要ない場合は、[Finish] をクリックします。

7. 「Hierarchy Verification Finished (階層の検査が完了しました)」というメッセージが表示されたら、[Done]をクリックします。

# 6.20.4.6. Sybase ASE リソース階層の拡張解除

LifeKeeper クラスタ内にある 1 つのサーバからリソース階層を削除するには、次の手順を実行します。

1. [Edit] メニューの [Resource] から [Unextend Resource Hierarchy] を選択します。
2. Sybase ASE リソースを拡張解除するターゲットサーバを選択します。リソースが現在 In Service のサーバは選択できません。(このダイアログボックスは、右側のペインでリソースインスタンスを右クリックして [Unextend] 作業を選択した場合には、表示されません。)[Next] をクリックします。
3. 拡張解除する Sybase ASE 階層を選択して、[Next] をクリックします。(このダイアログボックスは、いずれかのペインでリソースインスタンスを右クリックして [Unextend] 作業を選択した場合には、表示されません。)
4. 拡張解除のために選択したターゲットサーバと Sybase ASE リソース階層を確認する情報ボックスが表示されます。[Unextend] をクリックします。
5. Sybase ASE リソースが正常に拡張解除されたことを示す別の情報ボックスが表示されます。[Done] をクリックして、[Unextend Resource Hierarchy] メニューを終了します。

# 6.20.4.7. Sybase ASE リソース階層の削除

LifeKeeper 設定内のすべてのサーバから Sybase ASE リソースを削除するには、次の手順を実行します。

1. [Edit] メニューの [Resource] から [Delete Resource Hierarchy] を選択します。
2. Sybase ASE リソース階層を削除するターゲットサーバの名前を選択します。

   **注記:** 左側のペインからグローバルリソースを右クリックするか、右側のペインから個々のリソースインスタンスを右クリックして [Delete Resource] 作業を選択した場合、このダイアログボックスは表示されません。
3. [Hierarchy to Delete] を選択します。(このダイアログボックスは、左側または右側のペインでリソースインスタンスを右クリックして、[Delete Resource] 作業を選択した場合には、表示されません。)[Next] をクリックします。
4. 選択したターゲットサーバと、削除の対象として選択した階層を確認する情報ボックスが表示されます。[Next] をクリックします。
5. Sybase ASE リソースが正常に削除されたことを示す別の情報ボックスが表示されます。
6. [Done] をクリックして終了します。

# 6.20.4.8. Sybase ASE のリソース階層のテスト

Sybase ASE リソース階層のテストは、手動でスイッチオーバーすることで、プライマリーサーバーからバックアップサーバーへのリソースサーバーインスタンスのフェイルオーバーをシミュレートして実行できます。

## LifeKeeper GUI からの手動スイッチオーバーの実行

LifeKeeper GUI で **[Edit] > [Resource] > [In Service]** を選択すると、スイッチオーバを開始できます。例えば、バックアップサーバーで [In Service] 要求 が実行されると、Sybase ASE リソース階層がバックアップサーバー側で In Service になり、プライマリサーバー側では Out of Service になります。この時点で、元のバックアップサーバがプライマリサーバーに、元のプライマリサーバーがバックアップサーバーに変わります。

**[Out of Service]** 要求を実行した場合、リソース階層は Out of Service になりますが、別のサーバで In Service にはなりません。

**重要:** バックアップサーバーでリソース階層を In Service した後は、データベースに接続してみる必要があります。特に、raw デバイスをデバイス空間として使用している場合は接続を試みる必要があります。この作業は、バックアップサーバーですべてのディスクパーティションが認識されており、raw バインディングが正しく確立されていることを確認するために必要です。

バックアップサーバーで raw バインディングが確立されていない場合、その原因として最も可能性が高いのは、新しいパーティションをプライマリーサーバに作成して設定に追加したにも関わらず、バックアップサーバー側のパーティションテーブルが更新されていないためです。

解決策は、パーティションテーブルが正しくアップデートされるように、バックアップサーバーを再起動することです。

# 6.20.5. Sybase ASE Recovery Kit の管理

## リソース階層の管理

以下は、Sybase ASE 階層の継続的な管理に対する重要な推奨事項です。

リソース階層を作成後、以下の作業が必要になることがあります。

- Sybase Backup Server の保護の変更
- Sybase Monitor Server の保護の変更
- パラメータの更新

# 6.20.5.1. Sybase Backup Server の保護の変更

Sybase Backup Server は、Open Server ベースのアプリケーションで、Adaptive Server でのデータベースのバックアップ (ダンプ) とリストア (ロード) をすべて管理します。Sybase Backup Server は、LifeKeeper Sybase ASE リソース階層の作成中に保護下に置くことも、リソース階層を作成した後に LifeKeeper の保護下に加えることもできます。また、リソース階層を作成した後に、Sybase Backup Server を LifeKeeper の保護下から除外することもできます。

## Sybase Backup Server の追加

Sybase Backup Server を既存の Sybase ASE リソース階層に追加するには、Sybase の srvbuild またはその他の設定ユーティリティによって Backup Server を作成済みである必要があります。

1. [Edit] メニューから [Resource] を選択し、次に [Properties] を選択してください。[Resource Properties] ウィザードが開きます。

2. [Select Resource] プルダウンメニューからリソースタグを選択してください。これは、LifeKeeper 保護下にある変更対象の Sybase ASE リソースのリソースタグです。

3. [Select Server for Resource] プルダウンメニューで [LifeKeeper Server] を選択してください。これは、Sybase ASE リソースインスタンスを更新するサーバです。複数の LifeKeeper サーバ上で変更が必要な場合、この手順をクラスタ内の各サーバについて繰り返す必要があります。

4. [Resource Properties] ページで [Resource Configuration] ボタンをクリックしてください。これで、手順 3 で選択した保護リソースに対する [Reconfiguration] ウィザードが開きます。ウィザードの最初の画面には、この LifeKeeper 保護リソースの現在の設定が表示されます。[Next] をクリックします。

5. 有効な Sybase Backup Server が指定されたサーバ上に存在する場合、追加または削除する Sybase Backup Server が次の画面のプルダウンリストに表示されます。追加する Sybase Backup Server をリストから選択します。[Next] をクリックします。 **注記:** Sybase ASE インストールにおいて、共有ストレージに Sybase ソフトウェアがインストールされている場合、そのインストールを含むファイルシステムが、再設定を実行するサーバ上で In-Service される必要があります。

6. 有効な Sybase Monitor Server が存在する場合、次の画面で設定できます。Monitor Server の保護を変更する際の考慮事項については、Sybase Monitor Server の保護の変更 を参照してください。

7. [Reconfigure] を選択してください。何らかのエラーが表示された場合、それを修正してから先に進んでください。エラーがない場合、[Done] をクリックしてください。

8. Sybase Backup Server 設定ファイルパスまたは関連するデータベースデバイスは、LifeKeeper ファイルシステムリソースで手動で保護し、親リソース階層の依存リソースにする必要があります。

9. Sybase Backup Server に関連する仮想 IP アドレスは、親リソース階層の依存リソースにする必要があります。関連する IP アドレスを特定するには、interfaces ファイル内で、Sybase Backup Server 名の後のマスターおよびクエリ行を参照してください。

# Sybase Backup Server の削除

以下に、Sybase Backup Server を既存の Sybase ASE リソース階層から削除する手順を説明します。

1. [Edit] メニューから [Resource] を選択し、次に [Properties] を選択してください。[Resource Properties] ウィザードが開きます。

2. [Select Resource] プルダウンメニューからリソースタグを選択してください。これは、LifeKeeper 保護下にある変更対象の Sybase ASE リソースのリソースタグです。

3. [Select Server for Resource] プルダウンメニューで [LifeKeeper Server] を選択してください。これは、Sybase ASE リソースインスタンスを更新するサーバです。複数の LifeKeeper サーバ上で変更が必要な場合、この手順をクラスタ内の各サーバについて繰り返す必要があります。

4. [Resource Properties] ページで [Resource Configuration] ボタンをクリックしてください。これで、手順 3 で選択された保護リソースに対する [Reconfiguration] ウィザードが開きます。ウィザードの最初の画面には、この LifeKeeper 保護リソースの現在の設定が表示されます。[Next] をクリックします。

5. 有効な Sybase Backup Server が指定されたサーバ上に存在する場合、追加または削除する Sybase Backup Server が次の画面のプルダウンリストに表示されます。Sybase Backup Server の保護を解除するには、リストで [none] を選択します。[Next] をクリックします。

6. 有効な Sybase Monitor Server が存在する場合、次の画面で設定できます。Monitor Server の保護を変更する際の考慮事項については、Sybase Monitor Server の保護の変更 を参照してください。

7. [Reconfigure] を選択してください。何らかのエラーが表示された場合、それを修正してから先に進んでください。エラーがない場合、[Done] をクリックしてください。

8. 不要になった Sybase Backup Server 設定ファイルパスまたは関連するデータベースデバイスは、Sybase ASE リソース依存関係を解除し、LifeKeeper から削除する必要があります。

9. 不要になった Sybase Backup Server 仮想 IP リソースは、Sybase ASE リソース依存関係を解除し、LifeKeeper から削除する必要があります。

# 6.20.5.2. Sybase Monitor Server の保護の変更

注記 ：Sybase Monitor Server は Sybase ASE ARK v9.0.2 およびそれ以降のバージョンではサポートされません。

Monitor Server は、Adaptive Server データベースサーバとは独立したサーバであり、Adaptive Server を監視します。Monitor Server は、リアルタイムデータまたはヒストリカルデータをクライアントアプリケーションに提供できます。Sybase Monitor Server は、LifeKeeper Sybase ASE リソース階層の作成中に保護下に置くことも、リソース階層を作成した後に LifeKeeper の保護下に加えることもできます。また、リソース階層を作成した後に、Sybase Monitor Server を LifeKeeper の保護下から除外することもできます。

## Sybase Monitor Server の追加

Sybase Monitor Server を既存の Sybase ASE リソース階層に追加するには、Sybase の srvbuild またはその他の設定ユーティリティによって Monitor Server を作成済みである必要があります。

1. [Edit] メニューから [Resource] を選択し、次に [Properties] を選択してください。[Resource Properties] ウィザードが開きます。

2. [Select Resource] プルダウンメニューからリソースタグを選択してください。これは、LifeKeeper 保護下にある変更対象の Sybase ASE リソースのリソースタグです。

3. [Select Server for Resource] プルダウンメニューで [LifeKeeper Server] を選択してください。これは、Sybase ASE リソースインスタンスを更新するサーバです。複数の LifeKeeper サーバ上で変更が必要な場合、この手順をクラスタ内の各サーバについて繰り返す必要があります。

4. [Resource Properties] ページで [Resource Configuration] ボタンをクリックしてください。これで、手順 3 で選択された保護リソースに対する [Reconfiguration] ウィザードが開きます。ウィザードの最初の画面には、この LifeKeeper 保護リソースの現在の設定が表示されます。[Next] をクリックします。

5. 有効な Sybase Backup Server が存在する場合、次の画面で設定できます。Backup Server の保護を変更する際の考慮事項については、Sybase Backup Server の保護の変更 を参照してください。

6. 有効な Sybase Monitor Server が指定されたサーバ上に存在する場合、追加または削除する Sybase Monitor Server が次の画面のプルダウンリストに表示されます。追加する Sybase Monitor Server をリストから選択します。[Next] をクリックします。 **注記:** Sybase ASE インストールにおいて、共有ストレージに Sybase ソフトウェアがインストールされている場合、再設定を実行するサーバ上で、そのインストールを含むファイルシステムが In-Service される必要があります。

7. [Reconfigure] を選択してください。何らかのエラーが表示された場合、それを修正してから先に進んでください。エラーがない場合、[Done] をクリックしてください。

8. Sybase Monitor Server 設定ファイルパスまたは関連するデータベースデバイスは、LifeKeeper フ

ファイルシステムリソースで手動で保護し、親 Sybase ASE リソース階層の依存リソースにする必要があります。

9. Sybase Monitor Server に関連する仮想 IP アドレスは、親 Sybase ASE リソース階層の依存リソースにする必要があります。関連する IP アドレスを特定するには、 interfaces ファイル内で、Sybase Monitor Server 名の後のマスターおよびクエリ行を参照してください。

# Sybase Monitor Server の削除

以下に、Sybase Monitor Server を既存の Sybase ASE リソース階層から削除する手順を説明します。

1. [Edit] メニューから [Resource] を選択し、次に [Properties] を選択してください。[Resource Properties] ウィザードが開きます。

2. [Select Resource] プルダウンメニューからリソースタグを選択してください。これは、LifeKeeper 保護下にある変更対象の Sybase ASE リソースのリソースタグです。

3. [Select Server for Resource] プルダウンメニューで [LifeKeeper Server] を選択してください。これは、Sybase ASE リソースインスタンスを更新するサーバです。複数の LifeKeeper サーバ上で変更が必要な場合、この手順をクラスタ内の各サーバについて繰り返す必要があります。

4. [Resource Properties] ページで [Resource Configuration] ボタンをクリックしてください。これで、手順 3 で選択された保護リソースに対する [Reconfiguration] ウィザードが開きます。ウィザードの最初の画面には、この LifeKeeper 保護リソースの現在の設定が表示されます。[Next] をクリックします。

5. 有効な Sybase Backup Server が存在する場合、次の画面で設定できます。Backup Server の保護を変更する際の考慮事項については、Sybase Backup Server の保護の変更 を参照してください。

6. 有効な Sybase Monitor Server が指定されたサーバ上に存在する場合、追加または削除する Sybase Monitor Server が次の画面のプルダウンリストに表示されます。Sybase Monitor Server の保護を解除するには、リストで [none] を選択します。[Next] をクリックします。

7. [Reconfigure] を選択してください。何らかのエラーが表示された場合、それを修正してから先に進んでください。エラーがない場合、[Done] をクリックしてください。

8. 不要になった Sybase Monitor Server 設定ファイルパスまたは関連するデータベースデバイスは、Sybase ASE リソース依存関係を解除し、LifeKeeper から削除する必要があります。

9. 不要になった Sybase Monitor Server 仮想 IP リソースは、Sybase ASE リソース依存関係を解除し、LifeKeeper から削除する必要があります。

# 6.20.5.3. Sybase ASE Recovery Kit のパラメータの更新

Sybase ASE インスタンスのデータベースパラメータを更新するときは、すべての変更を行っても、クラスタ内の全 LifeKeeper サーバ上でそのインスタンスが正しく機能することを確認する必要があります。変更を行うことにより、LifeKeeper リソース (ファイルシステム、raw デバイス、仮想 IP アドレスなど) の追加や削除が必要になる場合、そのリソースを手動で追加し、親 Sybase ASE リソース階層の依存リソースにする必要があります。

# 6.20.6. Sybase ASE Recovery Kit トラブルシューティング

## Sybase ASE Recovery Kit のエラーメッセージ

Sybase ASE Recovery Kit に関連するエラーメッセージの一覧と説明です。

| | |
|---|---|
| 114000 | Usage: %s (使用法: %s) |
| 114001 | The Sybase Install Directory cannot be empty. (Sybase インストールディレクトリを空にすることができません)<br><br>処置: このフィールドに値を指定してください。 |
| 114002 | The path ”%s” is not a valid directory (パス「%s」は有効なディレクトリではありません) |
| 114003 | The Sybase Product was not found in the directory %s on server %s. (サーバ %s のディレクトリ %s に Sybase 製品が見つかりませんでした。)処置:指定の場所にサポート対象バージョンの Sybase がインストールされていることを確認してください。 |
| 114004 | The specified instance %s is not a valid Sybase ASE Server on %s. (指定されたインスタンス %s は、%s 上の有効な Sybase ASE サーバではありません。 |
| 114005 | Unable to verify that the Sybase ASE Server %s is running. (Sybase ASE サーバ %s が実行中であることを確認できません。) |
| 114006 | The Sybase Monitor Server %s will be protected. (Sybase Monitor Server %s を保護します。) |
| 114007 | The Sybase Backup Server %s will be protected. (Sybase Backup Server %s を保護します。) |
| 114008 | The Sybase ASE Server %s is already under LifeKeeper protection on %s. (Sybase ASE Server %s は、すでに %s 上で LifeKeeper に保護されています。) |
| 114009 | An unknown error has occurred in utility %s on server %s. (サーバ %s 上のユーティリティ %s で未知のエラーが発生しました。)<br><br>処置:LifeKeeper ログで詳細を確認し、操作をやり直してください。 |
| 114010 | Unable to get the version for the Sybase Server %s installed under %s on %s. (%s 上の %s にインストールされている Sybase サーバ %s のバージョンを取得できません。) |
| 114011 | The device %s for Sybase ASE Server %s is not a valid device. (Sybase ASE Server %s のデバイス %s は、有効なデバイスではありません。) |
| 114012 | Sybase ASE サーバ %s のデバイスを取得中にエラーが発生しました。 |
| 114013 | Unable to create raw resource hierarchy for %s. (%s の raw リソース階層を作成できません。) |

| | |
|---|---|
| 114014 | Unable to create file system resource hierarchy for %s. (%s のファイルシステムリソース階層を作成できません。) |
| 114015 | The path %s is not on a shared file system.(パス %s が共有ファイルシステムにありません。) |
| 114016 | Unable to create resource dependency for parent %s and child %s. (親 %s と子 %s にリソース依存関係を作成できません。) |
| 114017 | Information: LifeKeeper will not protect the path %s because it is not located on a shared file system. (情報: パス %s は共有ファイルシステム上にないため、LifeKeeper によって保護されません。) |
| 114018 | Unable to get the owner for the Sybase ASE Server%s installed under%s on %s. (%s 上の %s にインストールされている Sybase ASE サーバ %s の所有者を取得できません。) |
| 114019 | Unable to open file%s on server%s due to error %s. (エラー「%s」により、サーバ %s 上でファイル %s を開けません。) |
| 114020 | There are no hosts defined for the Sybase ASE Server %s in the file%s. (ファイル %s 内に Sybase ASE サーバ %s のホストが定義されていません。) |
| 114021 | There are no ports defined for the Sybase ASE Server %s in the file %s. (ファイル %s 内に Sybase ASE サーバ %s のポートが定義されていません。) |
| 114022 | The specified host name %s defined for the Sybase ASE Server %s in the file %s cannot be resolved. (ファイル %s 内に Sybase ASE サーバ %s に対して指定されているホスト名 %s を解決できません。) |
| 114023 | Unable to detect the host and ports for the Sybase ASE Server %s. (Sybase ASE サーバ %s のホストとポートを検出できません。) |
| 114024 | サーバ % 上の IP アドレス %s に LifeKeeper リソース階層が存在しません。<br><br>処置:指定された IP アドレスに LifeKeeper リソース階層を作成してください。 |
| 114025 | The values specified for the target and the template servers are the same.(ターゲットサーバとテンプレートサーバに指定された値が同一です。)<br><br>処置:ターゲットサーバとテンプレートサーバに正しい値を指定してください。 |
| 114026 | The system user %s does not exist on the server %s. (サーバ %s 上にシステムユーザ %s が存在しません。) |
| 114027 | The group id for user %s is not the same on template server %s and target server %s. (ユーザ %s のグループ ID がテンプレートサーバ %s とターゲットサーバ %s で同一ではありません。) |
| 114028 | The user id for user %s is not the same on template server %s and target server %s. (ユーザ %s のユーザ ID がテンプレートサーバ %s とターゲットサーバ %s で同一ではありません。) |

| | |
|---|---|
| 114029 | There are no IP dependent resources defined for the Sybase resource %s on %s. (%s 上に Sybase リソース %s に対して定義された IP 依存リソースがありません。)<br><br>処置:必要な依存 IP リソース階層を作成します。 |
| 114030 | The interfaces defined for Sybase ASE Server %s differ on template server %s and target server %s (Sybase ASE サーバ %s に定義されたインターフェースが、テンプレートサーバ %s とターゲットサーバ %s で異なっています) |
| 114031 | The ports defined for Sybase ASE Server %s differ on template server %s and target server %s (Sybase ASE サーバ %s に定義されたポートが、テンプレートサーバ %s とターゲットサーバ %s で異なっています) |
| 114032 | The port %s used by the Sybase resource hierarchy %s on the server %s is in use by another application on server%s. (サーバ %s 上で Sybase リソース階層 %s によって使用されるポート %s が、サーバ %s 上の別のアプリケーションによって使用されています。) |
| 114033 | The startup of the Sybase ASE Server(s) on %s failed for the following Sybase ASE Server(s): %s. (%s 上の Sybase ASE サーバの起動が、次の Sybase ASE サーバにおいて失敗しました: %s) |
| 114034 | Unable to stop the Sybase ASE Server(s) %s on %s. (%s 上で Sybase ASE サーバ %s を停止できません。) |
| 114035 | The Sybase ASE resource hierarchy %s does not contain any valid gen/filesys or scsi/raw resource dependents on server %s.(Sybase ASE リソース階層 %s に、サーバ %s 上で有効な gen/filesys または scsi/raw リソース依存関係がありません。)<br><br>処置:階層に有効な依存関係がないため、階層を削除して再作成する必要があります。 |
| 114036 | There are no Sybase ASE Servers available for protection with LifeKeeper.(LifeKeeper が保護できる Sybase ASE サーバが存在しません。) |
| 114037 | Unable to obtain the pid of the backupserver process corresponding to instance %s.(インスタンス %s に対応するBackup Server プロセスの pid を取得できません。) |
| 114038 | The pid detected for Sybase Backup Server %s in the LifeKeeper pidfile %s.LK on server %s exists in another LifeKeeper pidfile on this server.(サーバ %s 上の LifeKeeper pidfile %s.LK 内で 検出された Sybase Backup Server %s の pid が、このサーバ上の別の LifeKeeper pidfile に存在します。)<br><br>処置:pid ファイル内の重複する pid エントリを解決する必要があります。実行中でないインスタンスの pid ファイルを削除する必要があります。 |
| 114039 | Unable to update the resource instance %s on server %s. (サーバ %s 上でリソースインスタンス %s を更新できません。) |
| 114040 | The update of the resource instance %s failed on server %s. All attempts to rollback the instance |

| | |
|---|---|
| | information field have failed. (サーバ %s 上でリソースインスタンス %s の更新に失敗しました。インスタンス情報フィールドをロールバックする試みがすべて失敗しました。)処置:手動による介入が必要です。 |
| 114041 | The interfaces file %s on %s contains an invalid comment line. (%s 上のインターフェースファイル %s に無効なコメント行があります。)処置:インターフェースファイルを修正し、コメント行を削除してください。 |
| 114042 | One or more of the Sybase ASE Servers is missing from the file %s.(1 台または複数台の Sybase ASE サーバがファイル %s に欠落しています。) |
| 114043 | The file %s does not exist on server %s.(サーバ %s 上にファイル %s が存在しません。) |
| 114044 | The reconfiguration of the Sybase ASE resource hierarchy %s on server %s was successful (サーバ %s 上の Sybase ASE リソース階層 %s の再設定に成功しました。) |
| 114045 | The update of the resource instance %s failed on server %s. The instance information field has not been modified.(サーバ %s 上でリソースインスタンス %s の更新に失敗しました。インスタンス情報フィールドが変更されていません。)<br><br>処置:再設定の操作をもう一度行ってください。 |
| 114046 | The home directory for user %s is not the same on template server %s and target server %s (ユーザ %s のホームディレクトリがテンプレートサーバ %s とターゲットサーバ %s で同一ではありません。) |
| 114047 | The file %s on server %s is a link that does not resolve to a dependent shared resource on the template server %s.(サーバ %s 上のファイル %s はリンクですが、テンプレートサーバ %s 上の依存共有リソースに解決できません。) |
| 114048 | The link %s and its resolved path %s are not on a protected shared filesystem. (リンク %s とそのリンクを解決したパス %s は、保護された共有ファイルシステム上にありません。) |

# 6.20.7. Appendix – Sybase ASE で Raw I/O を使用してデバイス空間を作成する

## raw I/O によるデータベースデバイスの作成

---

raw I/O で Sybase ASE を使用するための要件

命名規則

raw I/O の設定手順

階層作成後のデータベースデバイスの追加

ASE と OCS のリンクの作成

# 6.20.7.1. raw I/O で Sybase ASE を使用するための要件

raw I/O で Sybase ASE Recovery Kit を使用する場合、次の要件を満たす必要があります。

- Linux OS が raw I/O デバイスをサポートしている必要があります。ほとんどのディストリビューションでは、カーネル 2.4 でサポートされましたが、カーネル 2.2 からサポートしているディストリビューションもあります。

- すべての raw I/O デバイスを共有ディスクパーティションにバインドする必要があります。raw I/O デバイス上に配置されるデータベースデバイスの数 (devspaces) により、必要な raw デバイスと共有ディスクパーティションの正確な数が決まります。raw デバイス上でデータベースデバイスを作成するための指針については、Installation Guide Adaptive Server for Linux を参照してください。

- Sybase ASE ソフトウェアのバージョンが raw I/Oデバイスの使用をサポートしている必要があります。

# 6.20.7.2. 命名規則

raw デバイスとコントローラの命名方法は、Linux ディストリビューションごとに異なります。

- Red Hat の場合、デバイス名は /dev/raw/raw<number> でコントローラ名は /dev/rawctl です。
- SuSE SLES 7 バージョンの場合、デバイス名は /dev/raw<number> で、コントローラ名は /dev/raw です。
- SuSE SLES 8 バージョンの場合、デバイス名は /dev/raw/raw<number> で、コントローラ名は /dev/rawctl です。

# 6.20.7.3. raw I/O の設定手順

1. Sybase ASE データベースデバイスに適したサイズの共有ディスクパーティションを選択します。

2. このパーティションに、未使用の raw デバイスノードをバインドします。これはマシンを再起動するたびに実行する必要があり、root アクセスを必要とするため、システム初期化ファイル (rc.local または boot.local) に raw バインディングを追加することもできます。このバインディングは、階層を LifeKeeper の保護下においた後、ファイルから削除する必要があります。LifeKeeper は、LifeKeeper の保護下にある raw I/O デバイスの raw バインディングを再確立します。コマンド raw -qa を使用して、どの raw デバイスノードが使用中か確認します。たとえば、以下のようになります。

   ```
   # raw -qa

   # raw /dev/raw/raw1 /dev/sda1
   ```

3. データベースインスタンスを保護するすべてのサーバ上で、raw デバイスコントローラ (/dev/rawctl または /dev/raw) とディスクパーティションの両方に、グローバル読み取り権限を設定してください。

   ```
   # chmod a+r /dev/rawctl (or chmod a+r /dev/raw )
   ```

4. データベースインスタンスを保護するすべてのサーバ上で、raw デバイスにグループおよびユーザの読み取り / 書き込み権限を設定してください。

   ```
   # chmod 664 /dev/raw/raw1
   ```

5. データベースインスタンスを保護するすべてのサーバ上で、raw デバイスの所有者を、指定されたデータベースインスタンスの Sybase ASE 所有者に変更してください。

   ```
   # chown -R sybase:sybase /dev/raw/raw1
   ```

6. raw デバイスのデータベースサーバへの追加については、Installation Guide Adaptive Server for Linux を参照してください。

# 6.20.7.4. 階層作成後のデータベースデバイスの追加

LifeKeeper で Sybase ASE 階層を作成後に、データベースデバイスを raw I/O デバイスまたは共有ファイルシステムに追加する場合、その raw デバイスまたはファイルシステムに対するリソース階層を LifeKeeper GUI を使って手動で作成する必要があります。そして、新しく作成したリソース階層を、Sybase ASE リソース階層の依存リソース階層 (子) にする必要があります。

# 6.20.7.5. ASE と OCS のリンクの作成

LifeKeeper for Linux Sybase ASE Recovery Kit では、パス $SYBASE/ASE-<version> が $SYBASE/ASE にシンボリックリンクされていることが要求されます。また、パス $SYBASE/OCS-<version> は $SYBASE/OCS にシンボリックリンクされている必要があります。LifeKeeper for Linux Sybase ASE Recovery Kit は、これらのリンクを使用してさまざまな Sybase ユーティリティやファイルにアクセスします。リンクを作成するには、次の手順を実行します。

1. コマンドライン上で $SYBASE ディレクトリに移動してください。

   例:

   ```
   server1 # cd $SYBASE

   server1 # pwd

        /opt/sybase-12.5
   ```

2. ASE-<version> ディレクトリを検索してください。

   例:

   ```
   server1 # ls -ld ASE*

   drwxrwxr-x 16 sybase sybase 4096 Nov 18 09:08 ASE-12_5

   lrwxrwxrwx 1 sybase sybase 8 Nov 17 11:35 ASEP -> ASEP-1_0

   drwxrwxr-x 4 sybase sybase 4096 Nov 17 11:35 ASEP-1_0
   ```

   注記: ASE-12_5 と ASE 間にリンクが既に存在する場合、手順 5 に進んでください。

3. ASE-<version> ディレクトリの中に bin/srvbuild ユーティリティがあることを確認してください。

   例:

   ```
   server1 # ls ASE-12_5/bin/srvbuild

        srvbuild
   ```

   注記:「No such file or directory」エラーが発生する場合、選択したパスが間違っています。

4. コマンドライン上で、特定された ASE-<version> ディレクトリと ASE 間にリンクを作成してください。

   例:

```
server1 # pwd

     /opt/sybase-12.5

server1 # ln -s ASE-12_5 ASE
```

5. リンクが正しく作成されたことを確認してください。

   例:

```
server1 # ls -ld ASE*

     lrwxrwxrwx 1 sybase sybase 8 Nov 17 11:20 ASE -> ASE-12_5

     drwxrwxr-x 16 sybase sybase 4096 Nov 18 09:08 ASE-12_5

     lrwxrwxrwx 1 sybase sybase 8 Nov 17 11:35 ASEP -> ASEP-1_0

     drwxrwxr-x 4 sybase sybase 4096 Nov 17 11:35 ASEP-1_0

server1 # ls ASE/bin/srvbuild

     srvbuild
```

6. コマンドライン上で $SYBASE ディレクトリに移動してください。

   例:

```
server1 # cd $SYBASE

server1 # pwd

     /opt/sybase-12.5
```

7. OCS-<version> ディレクトリを検索してください。

   例:

```
server1 # ls -ld OCS*

     drwxrwxr-x 16 sybase sybase 4096 Nov 18 09:08 OCS-12_5
```

   **注記:** OSC-12_5 と OSC 間にリンクが既に存在する場合、手順 5 に進んでください。

8. OCS-<version> ディレクトリの中に bin/isql ユーティリティがあることを確認してください。

   例:

```
server1 # ls OCS-12_5/bin/isql

     isql
```

**注記:**「No such file or directory」エラーが発生する場合、選択したパスが間違っています。

9. コマンドライン上で、特定された OCS-<version> ディレクトリと OCS 間にリンクを作成してください。

   例:

```
server1 # pwd

     /opt/sybase-12.5

server1 # ln -s OCS-12_5 OCS
```

10. リンクが正しく作成されたことを確認してください。

    例:

```
server1 # ls -ld OCS*

     lrwxrwxrwx 1 sybase sybase 8 Nov 17 11:20 OCS -> OCS-12_5

     drwxrwxr-x 16 sybase sybase 4096 Nov 18 09:08 OCS-12_5

server1 # ls OCS/bin/isql

     isql
```

# 6.21. VMDK as Shared Storage Recovery Kit 管理ガイド

## VMDK as Shared Storage Recovery Kit テクニカルドキュメンテーション

LifeKeeper for Linux VMDK as Shared Storage Recovery Kit（以下、VMDK Recovery Kit と呼ぶ）は、LifeKeeper 環境においてVMwareの仮想ハードディスクを共有ストレージとして使用するものです。VMDK Recovery Kit は、LifeKeeper ユーザに対し、 仮想ハードディスクを LifeKeeper 階層の基本ストレージとして使用する機会を提供します。

## ドキュメントの内容

本書は以下のトピックについて説明しています。

- ドキュメンテーションとリファレンス 。LifeKeeper for Linux のドキュメントと所在の一覧です。
- 要件 。VMDK Recovery Kit を適切に設定、導入、運用するうえで必要なハードウェアとソフトウェアについて説明しています。LifeKeeper for Linux ソフトウェアの具体的なインストール / アンインストール方法については、SIOS Protection Suite インストールガイド を参照してください。
- 概要 。VMDK Recovery Kit の特徴と機能について説明しています。
- LifeKeeper for Linux VMDK Recovery Kit の設定 。VMDK Recovery Kit を適切に設定するうえで必要な手順について説明しています。
- LifeKeeper 設定作業 。LifeKeeper GUI を使用して VMDK リソース階層を作成し管理する作業について説明しています。
- トラブルシューティング 。LifeKeeper for Linux のエラー･メッセージの一覧とその説明です。

# 6.21.1. VMDK のドキュメンテーションとリファレンス

以下に、SIOS Technology Corp が発行している SPS 関連情報の一覧を示します。

- LifeKeeper for Linux リリースノート
- LifeKeeper for Linux テクニカルドキュメンテーション
- LifeKeeper for Linux インストールガイド
- Optional Recovery Kit ドキュメンテーション

このドキュメンテーション、および他の LifeKeeper Recovery Kit に関連するドキュメンテーションは、以下の Web サイトで入手できます。

http://jpdocs.us.sios.com

# 6.21.2. VMDK のハードウェア要件とソフトウェア要件

LifeKeeper VMDK Recovery Kit をインストールする前に、LifeKeeper の構成が次の要件を満たしている必要があります。LifeKeeper for Linux ハードウェアおよびソフトウェアの設定に関する具体的な手順については、LifeKeeper for Linux インストールガイド を参照してください。

## ハードウェア要件

- サーバ ：製品媒体に付属する LifeKeeper for Linux リリースノート および LifeKeeper for Linux インストールガイド (製品メディアに収録) に記載の要件に従って設定されたLifeKeeper for Linux 対応の VMware ゲスト。

- **IP ネットワークインターフェースカード** ：各サーバは、イーサネット TCP/IP をサポートするネットワークインタフェースを 1 つ以上必要とします。ただし、LifeKeeper クラスタには 2 つの通信パスが必要で、独立した 2 つのサブネットを使用する 2 つの分離した LAN ベース通信パスが推奨され、これらの 1 つ以上をプライベートネットワークとして構成する必要があることを忘れないでください。TCP と TTY のハートビートの組み合わせもサポートされています。

## ソフトウェア要件

- **TCP/IP** ソフトウェア ：LifeKeeper 環境の各サーバには TCP/IP ソフトウェアが必須です。

- **LifeKeeper** ソフトウェア ：クラスタの各サーバにインストールした LifeKeeper for Linux ソフトウェアのバージョン、および適用した LifeKeeper for Linux ソフトウェアパッチのバージョンが同一であることが絶対条件です。

- **LifeKeeper for Linux VMDK Recovery Kit** - VMDK Recovery Kit は LifeKeeper インストールイメージに含まれています。セットアップスクリプトのリカバリーキット選択画面で選択するとインストールされます。

- **Linux** ソフトウェア ：VMDK Recovery Kit の実行に必要な追加ソフトウエアは LifeKeeper インストールイメージに含まれています。セットアップスクリプトは VMDK Recovery Kit と同時にインストールします。

  LifeKeeper for Linux ソフトウェアの具体的なインストール / アンインストール方法については、LifeKeeper for Linux インストールガイド を参照してください。

# 6.21.3. VMDK Recovery Kit の概要

LifeKeeper for Linux VMDK Recovery Kit の第 1 の目的は、LifeKeeper ユーザに対し、共有ストレージとデータレプリケーションに代わる代替ストレージ方式を提供することです。

VMDK Recovery Kit を使用すると、LifeKeeper 保護下のサーバまたはクライアント上に LifeKeeper リソース階層を作成できます。このリソース階層には、VMwareハイパーバイザーの提供する仮想ハードディスクが接続され、そのディスク上に作成されたファイルシステムがマウントされます。

仮想ハードディスクを接続したノードで障害が検出されると、VMDK Recovery Kit は、事前判別されたバックアップノードへのフェイルオーバーを開始し、バックアップノードに同じ仮想ハードディスクを再接続します。そのため、仮想ディスク上に構成されたファイルシステムが LifeKeeper サーバーまたはクライアントにマウントされたら、それを LifeKeeper 階層のストレージの基礎として完全に使用できます。

VMDK Recovery Kit のリソース階層は、LifeKeeper Core 製品（steeleye-lk パッケージ）から 入手可能な現存の File System Recovery Kit を使用して作成されます。

## VMDK Recovery Kit の制限事項

- このバージョンの VMDK Recovery Kit では、仮想ハードディスクへのアクセスが失敗したときのローカルリカバリーをサポートしていません。障害が検出されると、デフォルトアクションによって、バックアップサーバーへの階層の転送が開始されます。 リソース階層の構成に応じて、このアクションはハング･プロセスにつながることがあります。ハング･プロセスを回避するには、サーバを停止してバックアップ･ サーバへのフェイルオーバを強制するように、デフォルト･アクションを変更します。デフォルトの切り替え動作を変更するには、LifeKeeper デフォルトファイルで VMDK_ERROR の設定を変更します。VMDK_ERROR の詳細については、このドキュメントのセクション「 LifeKeeper for VMDK Recovery Kit の設定 」を参照してください。

- クラスタに参加するすべてのゲストは SCSI コントローラの構成を同一にする必要があります。VMDK Recovery Kit はリソースの作成時に仮想ハードディスクが接続されていた SCSI コントローラに対して仮想ハードディスクの再接続を行います。

- リソース起動時にはスナップショットの作成･復元が出来ません。 仮想ハードディスクがノード間を移動して変更された場合にスナップショットとの整合性が保証できなくなるため、明示的に禁止されます。

# 6.21.4. LifeKeeper for Linux VMDK Recovery Kitの設定

本セクションでは、LifeKeeper for Linux VMDK Recovery Kitの設定の詳細について説明します。VMDK Recovery Kitの設定と管理を開始する前に考慮すべき事項についても記載しています。LifeKeeper Coreリソース階層の設定方法については、LifeKeeper for Linuxテクニカルドキュメンテーション を参照してください。

# 6.21.4.1. VMDK設定上の考慮事項

VMDK Recovery Kitには、ホットプラグ対応ハードディスクをサポートするSCSI コントローラーが必要です。“LSI Logic SAS” または “VMware paravirtual“を選択してください。

LifeKeeper for Linux VMDK Recovery Kit を操作する前に、次の事項を検討する必要があります。

1. 仮想ハードディスクを共有し、かつVMDKリソース階層を拡張するクラスタ構成内のサーバへ、VMDK Recovery Kitをインストールします。仮想ハードディスクは VMDK形式で作成します。

2. 仮想ハードディスクは、クラスターを構成するゲスト間で共有されているデータストア上に作成する必要があります。

3. 仮想ハードディスクの排他制御はハイパーバイザーに依存しているため、接続するSCSIコントローラの共有設定を「なし」に設定する必要があります。

4. このキットは VMwareの提供するAPIを使用して仮想ハードディスクを操作するため、クラスタに参加するゲストを実行しているすべてのVMware ESXiホスト、もしくは管理しているvCenter Serverへhttpsでアクセスできる必要があります。

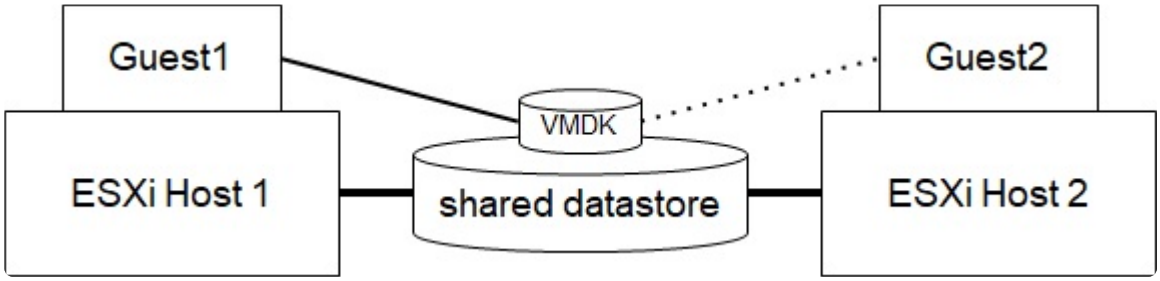
5. VMDK階層をビルドするために使用するビルトインファイルシステムのリカバリーキットでは、フェイルオーバー条件でマウントされたファイルシステムを使用 している LifeKeeper保護下に入っていないプロセスを検出したり、除去したりすることはできません。そのため、LifeKeeper保護下に入れるプロセスのみが、VMDK 保護下のファイルシステムを使用するように設定することを強くお勧めします。

6. 調整可能パラメータVMDK_ERRORは、 仮想ハードディスクへのアクセス権を失った際にVMDK Recovery Kitが実行するアクションを制御します。調整可能パラメータには、2つの 値 ( **halt** 及び **event** ) があり、 **halt** がデフォルトです。値が **halt** に設定され、アクセス権を失ったことを検出すると、VMDK Recovery Kit は即座にシステムを停止し、バックアップ・サーバへのフェイルオーバを発生させます。 **event** に設定された場合、アクセス権を失うとLifeKeeper coreにディスクが異常であることを通知し、通知を受けたLifeKeeper coreは対応する処理を起動します。この時行われる処理は上位リソースに依存して決定されます。

# 6.21.4.2. VMDKの構成例

フェイルオーバーが発生したときに、 **LifeKeeper for Linux VMDK Recovery Kit** を使用しているとどのような処理が行われるか、その例をいくつか示します。

## 構成 1：アクティブ／スタンバイ設定の例

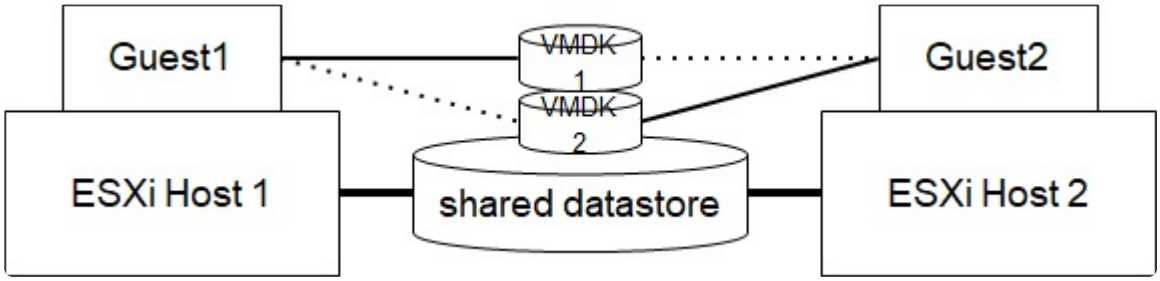



この構成では、ESXi Host1にある Guest1がVMDK Recovery Kitソフトウェアにより、仮想ハードディスクへアクセスできるため、アクティブとみなされます。Guest2は別の処理を実行します。 Guest1に障害が発生した場合、Guest 2 が VMDKへのアクセス権を獲得しファイルシステムへアクセスできるようにします。

**設定の注意点：**

- VMDK Recovery Kitを両方のゲストOSにインストールしておく必要があります。
- ファイルシステムは共有したVMDK仮想ハードディスク上に作成します。
- Guest 1 がアクティブである間、Guest 2 は 共有仮想ディスク上のファイルおよびディレクトリにアクセスできません。

## 構成 2：アクティブ／アクティブ設定の例




アクティブ／アクティブ設定は、VMDK Recovery Kit ソフトウェアをアクティブに実行し、 かつ個別の仮想ハードディスク接続した複数のシステムで構成されます。

**設定の注意点：**

- VMDK Recovery Kitを両方のゲストOSにインストールしておく必要があります。
- 最初に、Guest 1 が一つの仮想ハードディスクへ接続して、Guest 2 が別の仮想ハードディスクに接

続します。切り替え時には、1 つのシステムで両方の仮想ハードディスクへ接続します。

# 6.21.5. LifeKeeper VMDK Recovery Kit の設定作業

LifeKeeper for Linux VMDK Recovery Kit の管理作業はすべて、LifeKeeper グラフィカルユーザーインタフェース（GUI）で実行できます。LifeKeeper GUI では、VMDK リソースの設定、管理、監視作業をガイド付きで行えます。

LifeKeeper for Linux VMDK Recovery Kit の設定に関しては、次の作業が可能です。

- ESXiホストの登録 ：仮想ハードディスクを管理している ESXi ホストの情報を登録します。
- VMオプションの変更 ：VMDK Recovery Kit で必要なオプションを設定します。
- VMDK リソース階層の作成 ：VMDK リソース階層を作成します。
- VMDK リソース階層の削除 ：VMDK リソース階層を削除します。
- VMDK リソース階層の拡張 ：VMDK リソース階層をプライマリ･サーバからバックアップサーバに拡張します。
- VMDK リソース階層の拡張解除 ：LifeKeeper クラスタの 1 つのサーバの VMDK リソース階層を拡張解除（削除）します。
- 従属関係の作成 ：既存のリソース階層と別のリソースインスタンスとの間に子従属関係を作成し、クラスタ内のすべての対象サーバに従属関係の変化を通知します。
- 従属関係の削除 ：リソースの従属関係を削除して、クラスタ内のすべての対象サーバ に従属関係の変化を通知します。
- In Service ：リソース階層をアクティブにします。
- Out of Service ：リソース階層を非アクティブにします。
- 表示 / 編集 ： リソース階層のプロパティを表示または編集します。

**注記：** このセクションの残りの部分では、設定作業はすべて **[Edit]** メニューから実行することになっています。この設定作業の大半は次の方法でも実行できます。

1. ツールバーから状態表示の左ペインにあるグローバルリソースを右クリック。
2. 状態表示の右ペインにあるリソースインスタンスを右クリック。

※右クリックの方法を使用すると、 **[Edit]** メニューを使用したときに必要な情報を入力しないですみます。

# 6.21.5.1. ESXiホストの登録

VMDK リソースを作成する前に、ESXi ホストの情報を登録する必要があります。以下の手順で登録してください。

1. コンソール画面で以下のコマンドを実行してください。

   ```
   # /opt/LifeKeeper/lkadm/subsys/scsi/vmdk/bin/esxi_register -a <ESXi ホスト名>
   ```

   コマンドを実行すると、対話的にユーザ名およびパスワードを問い合わせてくるので、ESXi ホストにログインすることのできるユーザ・パスワードを入力してください。
   正しくログインできなかった場合はエラーになり登録できません。

2. クラスタを構成する各 ESXi ホストを同じように登録して下さい。

3. すべての ESXi ホストが登録できたら、以下のコマンドで正しく登録できていることを確認してください。

   ```
   # /opt/LifeKeeper/lkadm/subsys/scsi/vmdk/bin/esxi_register -l
   ```

   登録済みのホスト一覧が出力されます。

4. クラスタ内のすべてのノードで同様に ESXi ホストを登録して下さい。

`esxi_register` コマンドの詳細は「 メンテナンス 」を参照して下さい。

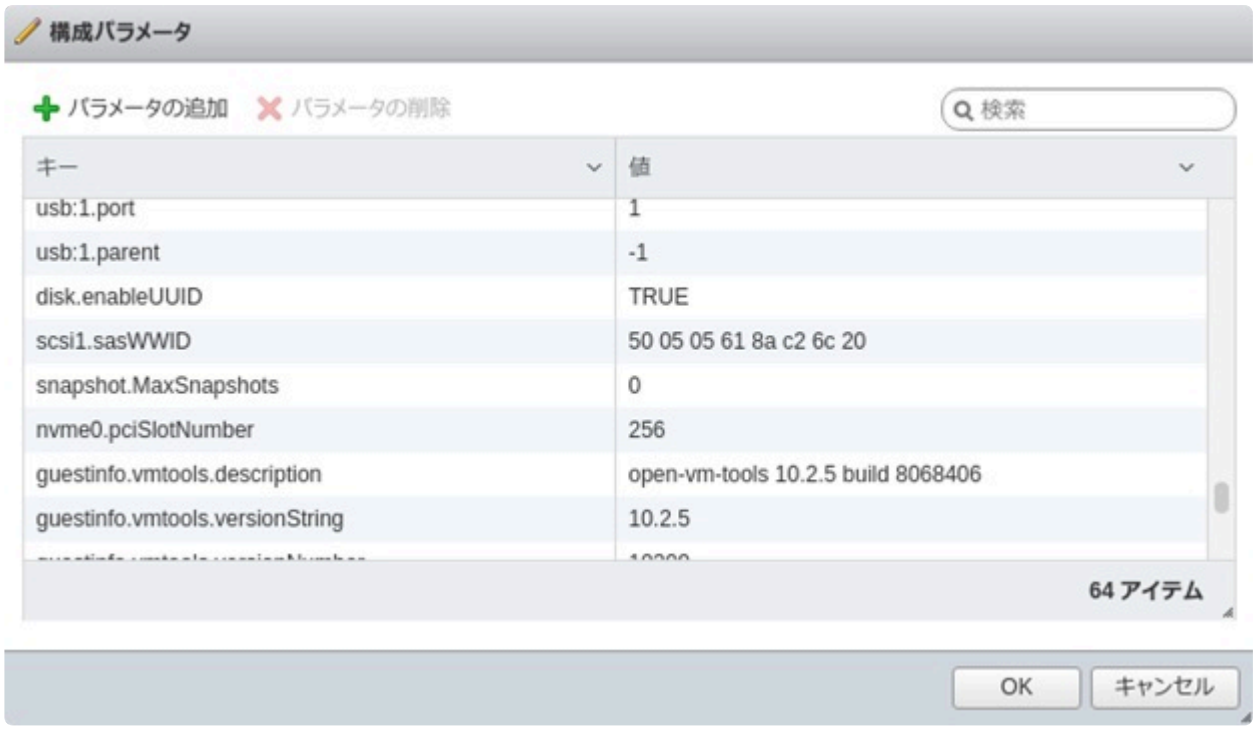
# 6.21.5.2. 仮想マシンオプションの設定変更

VMDK Recovery Kitでは、以下のオプションを設定する必要があります。

| キー | 値 |
|---|---|
| disk.enableUUID | TRUE |

VMの「設定の編集」ー「仮想マシン　オプション」ー「詳細」ー「構成パラメータ」ー「構成の編集」で、編集ダイアログを開いて上記パラメータを追加して下さい。

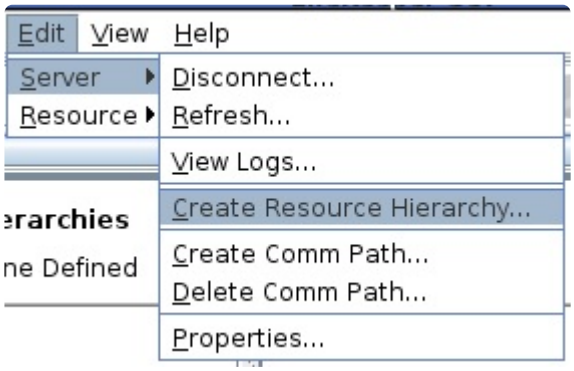
設定の編集: RHEL7.6 (ESXi 6.7 仮想マシン)
詳細
設定
アクセラレーション機能の無効化
ログの有効化
デバッグと統計
正常に実行
スワップ ファイルの場所
デフォルト
仮想マシンを含むクラスタまたはホストの設定を使用します。
仮想マシン ディレクトリ
仮想マシンと同じディレクトリにスワップ ファイルを格納します。
ホストによって指定されたデータストア
構成パラメータ
構成の編集...
遅延感度
正常


構成パラメータ
パラメータの追加
パラメータの削除
検索
キー
値
usb:1.port
1
usb:1.parent
-1
disk.enableUUID
TRUE
scsi1.sasWWID
50 05 05 61 8a c2 6c 20
snapshot.MaxSnapshots
0
nvme0.pciSlotNumber
256
guestinfo.vmtools.description
open-vm-tools 10.2.5 build 8068406
guestinfo.vmtools.versionString
10.2.5
64 アイテム
OK
キャンセル

# 6.21.5.3. VMDK リソース階層の作成

プライマリサーバーで以下を実行して、 **リソース作成ウィザード** を開始します。

1. **[Edit] > [Server] > [Create Resource Hierarchy]** を選択します。




2. **[リカバリーキットの選択]** ダイアログが表示されます。ドロップダウンリストから **[ファイルシステム]** オプションを選択します。VMDK Resource Hierarchy を、仮想ハードディスク上に作成したファイルシステム階層にします。

**[Next]** をクリックして次に進みます。

**注記：** 階層作成手順のどの段階でも、 **[Cancel]** ボタンをクリックすると、作成プロセス全体がキャンセルされます。

3. **[スイッチバックタイプ]** ダイアログが表示されます。スイッチバックタイプによって、フェイルオーバ後、バックアップサーバーで VMDK リソースがサービス中（アクティブ）になったとき、VMDK リソースをプライマリーサーバーにスイッチバックする方法が決まります。スイッチバックタイプは intelligent または automatic です。インテリジェントスイッチバックの場合、リソースをプライマリーサーバーに戻すために管理者の操作が必要ですが、自動スイッチバックの場合は、プライマリーサーバーがオンラインに戻り、LifeKeeper との通信パスが再確立されるとすぐにスイッチバックが実行されます。

**[Next]** をクリックして次に進みます。

4. **[server]** ダイアログが表示されます。VMDK リソースを作成するサーバの名前を選択します（通常は、これがプライマリーサーバーです）。クラスター内のすべてのサーバがドロップダウンリストボックスに表示されます。

**[Next]** をクリックして次に進みます。

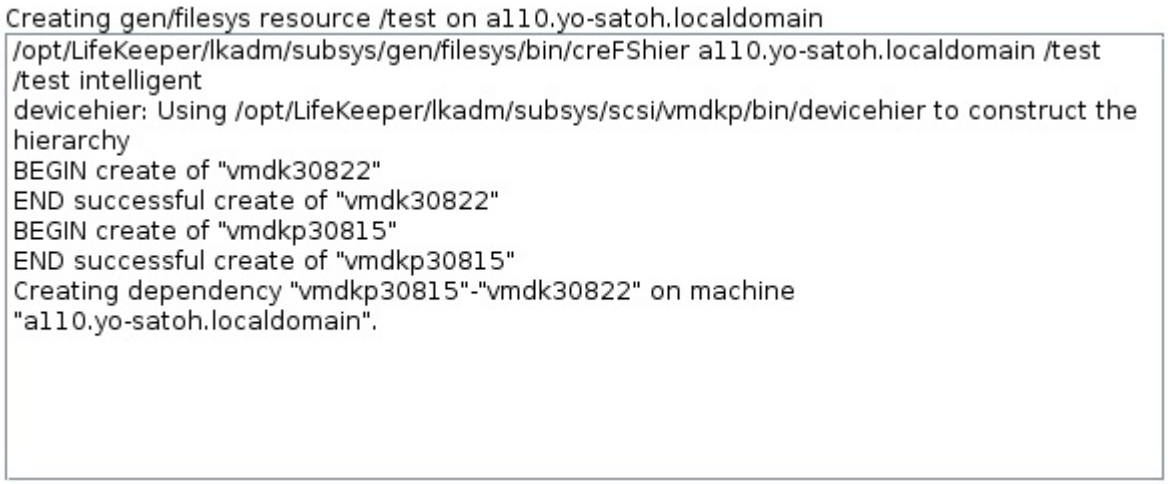
5. VMDK (File System) Resource Hierarchy で保護する **[マウントポイント]** パスを選択します。すべての「ローカル」ファイルシステム（つまり、共有ストレージを使用するファイルシステム） および VMDK Recovery Kit で扱える仮想ハードディスクのマウントポイントがリストされます。ドロップダウンリストから目的のマウントポイントを選択します。

**[Next]** をクリックして次に進みます。

6. **[ルートタグ]** ダイアログに、ターゲットサーバー（つまり、上記で選択したサーバー）上のリソースインスタンスの固有名が自動的に表示されます。デフォルトを受け入れることもできますし、文字、数字、および特殊文字-、_、.、または/で構成 された固有のタグ名を入力することもできます。

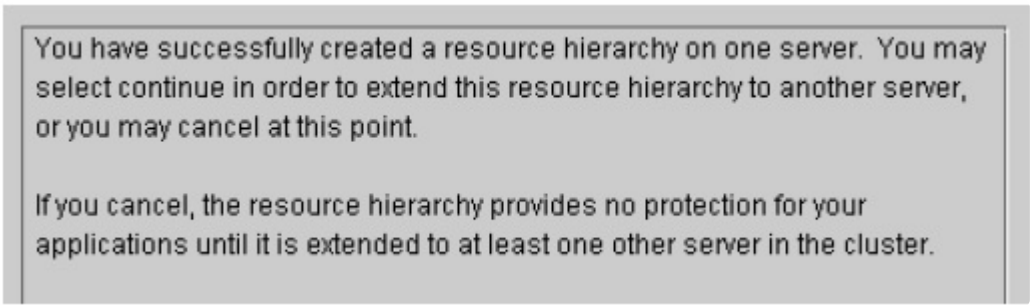
**[Create Instance]** をクリックします。

7. 階層の作成を開始することを示す情報ボックスが表示されます。

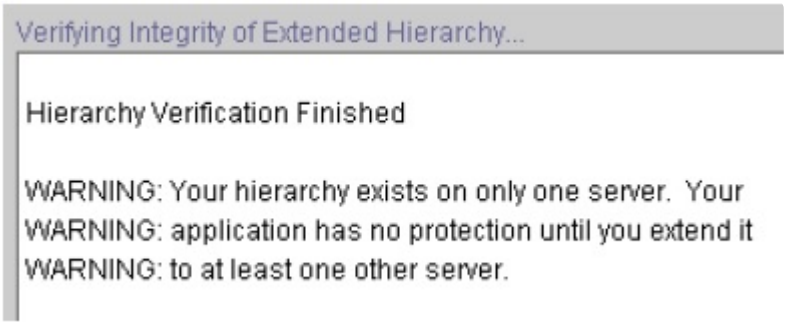



**[Next]** をクリックして次に進みます。

8. VMDK リソース階層の作成に成功したことを示す情報ボックスが表示されます。リソース階層を LifeKeeper で保護するには、クラスター内の別のサーバーにリソース階層を 拡張する必要があります。




リソースを拡張する場合は、 **[Continue]** をクリックします。 後でリソースを拡張する場合は、 **[Cancel]** をクリックします。




9. **[Done]** をクリックして、[リソース階層の作成]メニューを終了します。

# 6.21.5.4. VMDK リソース階層の削除

プライマリーサーバーで次の作業を行います。LifeKeeper 構成内のすべてのサーバから VMDK リソースを削除するには、次の手順を実行します。

1. **[LifeKeeper GUI]** メニューから **[Edit]** を選択し、次に **[Resource]** を選択します。ドロップダウンメニューから、 **[Delete Resource Hierarchy]** を選択します。

2. VMDK リソース階層を削除する **ターゲットサーバー** の名前を選択します。

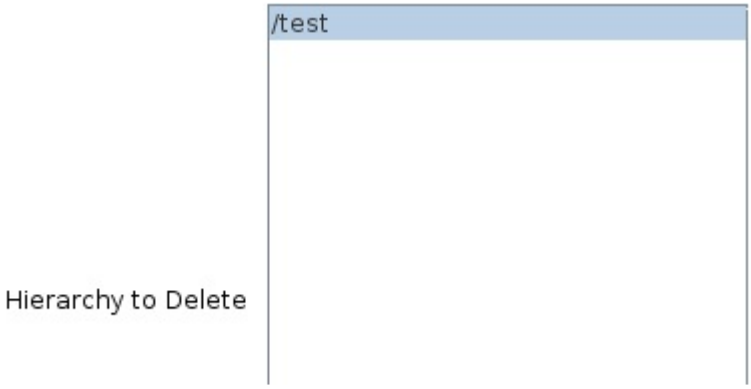
   **注記：** 左側のペインからグローバルリソースを右クリックするか、右側のペインから個々のリソースインスタンスを右クリックして **[リソース階層の削除]** を選択した場合、このダイアログ・ボックスは表示されません。

   **[Next]** をクリックして次に進みます。

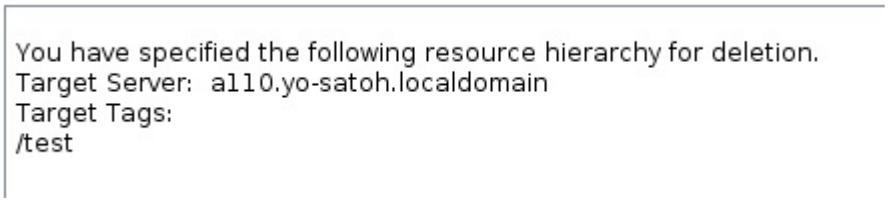
3. **[削除する階層]** を選択します。削除するリソース階層を特定して、強調表示にします。

   **注記：** 左側のペインからグローバル・リソースを右クリックするか、右側のペインから個々のリソース・インスタンスを右クリックして **[リソース階層の削除]** を選択した場合、このダイアログ・ボックスは表示されません。

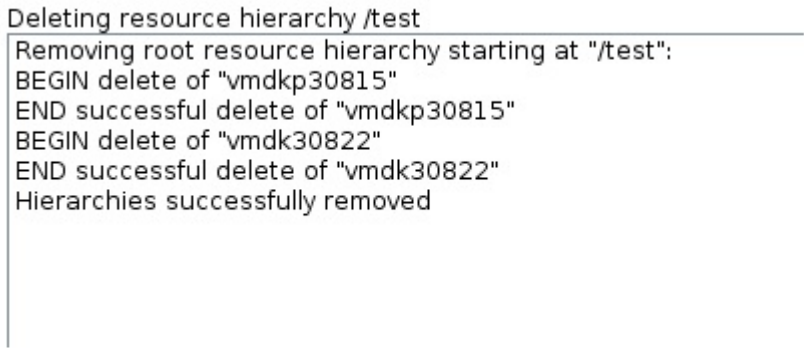
   **[Next]** をクリックして次に進みます。

4. 選択したターゲットサーバーと、削除の対象として選択した階層を確認する情報ボックスが表示されます。

**[Delete]** をクリックして次に進みます。

5. File System VMDK リソースインスタンスの削除に成功したことを示す情報ボックス が表示されます。




6. **[Done]** をクリックして、 **[リソース階層の削除]** メニューを終了します。

# 6.21.5.5. VMDK 階層の拡張

階層は、作成後に、クラスター内の別のサーバに拡張する必要があります。リソースインスタンスを拡張するには、次の 3 つの方法が可能です。

1. VMDK リソース階層の作成に成功したときに、**[Continue]** を選択して、バックアップサーバーにリソース階層を拡張する処理に進むことができます。

2. LifeKeeper GUI の左ペインまたは右ペインで拡張されていない階層を右クリックします。

3. LifeKeeper GUI で、ドロップダウンメニューから **[Edit] > [Resource] > [Extend Resource Hierarchy]** を選択します。選択すると、**[リソース階層の拡張]** ウィザードが起動されます。**[Extend Resource Hierarchy]** オプションの **[Accept Defaults]** ボタンは、LifeKeeperの **[リソース階層の拡張]** の既定値をよく知っていて、値の入力や確認をしないで素早く LifeKeeper リソース階層を拡張したいユーザ向けです。GUI ダイアログを使用して対話的に段階を追って LifeKeeper リソース階層を拡張する場合は、**[Next]** ボタンを選択します。

   a. 最初に表示されるダイアログボックスでは、VMDK リソース階層が現在サービス中であるテンプレートサーバーを選択します。**[テンプレートサーバー]** で選択するテンプレートサーバーと、次のダイアログボックスの **[拡張するタグ]** で選択する拡張するタグが、サービス中（アクティブ）のリソース階層を表すことを忘れないでください。選択したテンプレートサーバでサービス中になっていない リソースタグを選択すると、エラーメッセージが表示されます。このダイアログのドロップダウンボックスには、クラスタ内の全サーバの名前が示されます。

   **注記：** VMDK リソース階層作成後、引き続きリソース階層の拡張作業に入った場合は、作成段階でテンプレートサーバーが特定されているため、このダイアログボックスは表示されません。これは、GUI ウィンドウの左ペインの VMDK リソースアイコンまたは右ペインの VMDK (File System) リソース･ボックスを右クリックして、**[リソース階層の拡張]** を選択した場合も同様です。

   **注記：** 階層拡張の途中で **[Cancel]** ボタンをクリックすると、拡張処理が取り消されます。ただし、すでにリソースを別のサーバーに拡張している場合は、明示的に拡張解除するまで、そのインスタンスの拡張は有効です。

   **[Next]** をクリックして次に進みます。

   b. **[拡張するタグ]** を選択します。これは、テンプレートサーバーからターゲットサーバーに拡張する VMDK インスタンスの名前です。ウィザードのドロップダウンリストボックスには、テンプレートサーバー上に作成されているすべてのリソースが表示されます。

   **注記：** ここでも、VMDK リソース階層の作成後すぐにリソース階層の拡張作業に進んだ場合は、作成段階ですでにリソースのタグ名が特定されているため、このダイアログボックスは表示されません。これは、GUI ウィンドウの左ペインの VMDK (File System) リソースアイコンまたは右ペインの VMDK (File System) リソースボックスを右クリックして、[リソース階層

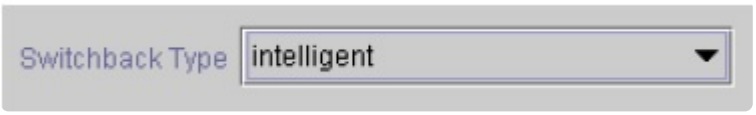
の拡張]を選択した場合も同様です。

**[Next]** をクリックして次に進みます。

c. VMDK リソース階層を拡張するターゲットサーバーを選択します。

**[Next]** をクリックして次に進みます。

d. **[スイッチバックタイプ]** ダイアログが表示されます。スイッチバックタイプによって、フェイルオーバー後、バックアップサーバーで VMDK リソースがサービス中（アクティブ）になったとき、VMDK リソースをプライマリーサーバーにスイッチバックする方法が決まります。スイッチバックタイプは intelligent または automatic です。インテリジェントスイッチバックの場合、リソースをプライマリーサーバーに戻すために管理者の操作が必要ですが、自動スイッチバックの場合 は、プライマリーサーバーがオンラインに戻り、LifeKeeper との通信パスが再確立されるとすぐにスイッチバックが実行されます。



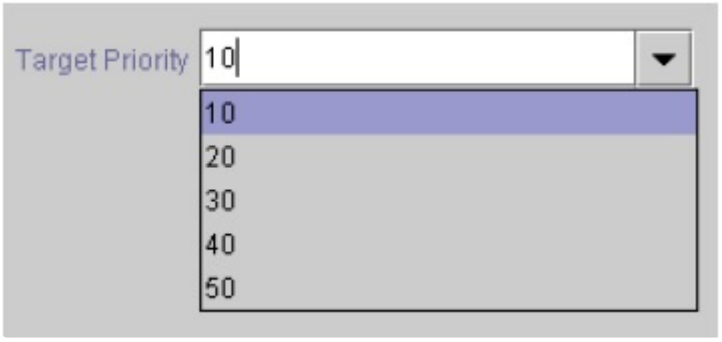

**[Next]** をクリックして次に進みます。

e. テンプレートの優先順位を選択または入力します。これはサーバで現在サービス中の VMDK 階層の優先順位です。優先順位は、1～999 の範囲で未使用の値が有効で、小さい数字ほど優先順位が高くなります（数字 1 が最高の優先順位に相当します）。拡張操作時には、別のシステムで既に使用中の優先順位をこの階層に対して指定することはできません。デフォルト値を推奨します。

**注記：** このフィールドは階層を最初に拡張するときだけ表示されます。

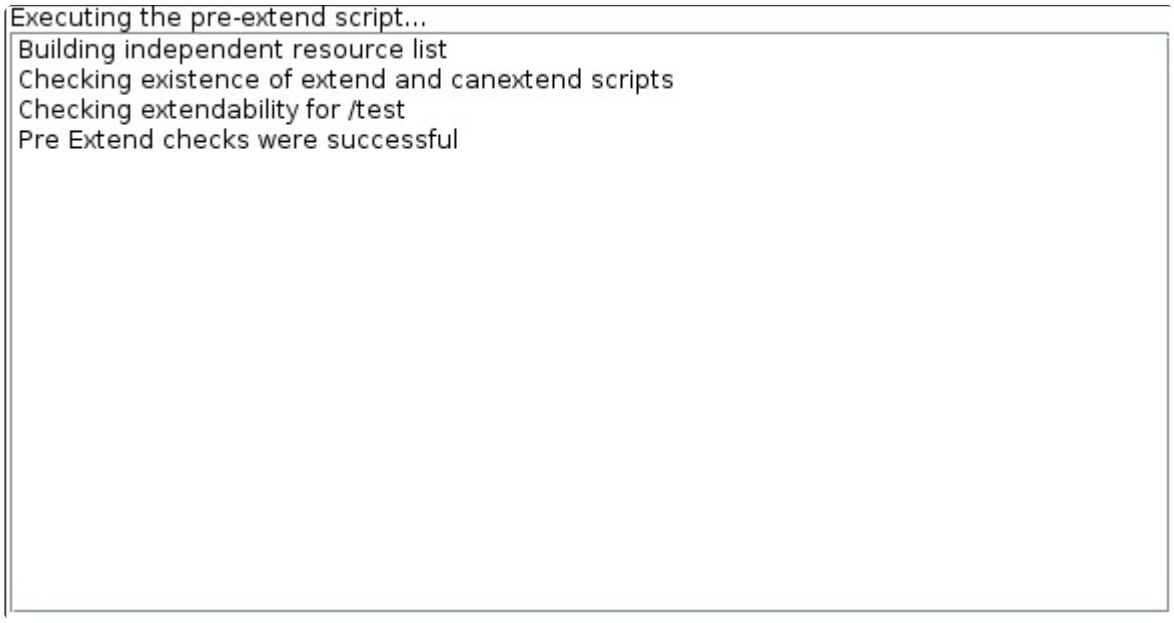
**[Next]** をクリックして次に進みます。

f. ターゲットの優先順位を選択または入力します。これは、別サーバーにある同等の階層に対する新しく拡張する VMDK 階層の優先順位です。1～999 の範囲の未使用の値が有効で、リソースの連鎖フェイルオーバ・シーケンスにおけるサーバの優先順位を示します。数値が小さいほど優先順位は高くなります（1 は最高の優先順位を表わします）。LifeKeeper のデフォルトでは、階層が作成されたサーバに「1」が割り当てられることに注意してください。優先順位は連続している必要はありませんが、特定のリソースについて 2 つのサーバに同じ優先順位を割り当てることはできません。




**[Next]** をクリックして次に進みます。

g. 環境がチェックされ、このリソースを拡張するのに必要な要件がすべて満たされていることを示す情報ボックスが表示されます。満たされていない要件がある場合、 **[Next]** ボタンが使用できなくなり、 **[Back]** ボタンが有効になります。




リソースの拡張の設定を変更する場合は、 **[Back]** ボタンをクリックします。 後でリソースを拡張する場合は、[キャンセル]をクリックします。リソース階層の拡張を実行する場合は、 **[Next]** をクリックします。

**[Finish]** をクリックして、VMDK リソース・インスタンスの拡張に成功したことを確認します。

4. **[Done]** をクリックして、 **[リソース階層の拡張]** メニューを終了します。

**注記：** 必ず両方のサーバで新しいインスタンスの機能をテストしてください。

# 6.21.5.6. VMDK 階層の拡張解除

1. LifeKeeper GUI メニューで、 **[Edit] > [Resource] > [Unextending Resource Hierarchy]** を選択します。

2. VMDK リソースを拡張解除する **[ターゲットサーバー]** を選択します。リソースが現在サービス中（アクティブ）のサーバは選択できません。

   **注記：** 左側のペインからグローバルリソースを右クリックするか、右側のペインから個々のリソースインスタンスを右クリックして拡張解除作業を選択した場合、 このダイアログボックスは表示されません。

   **[Next]** をクリックして次に進みます。

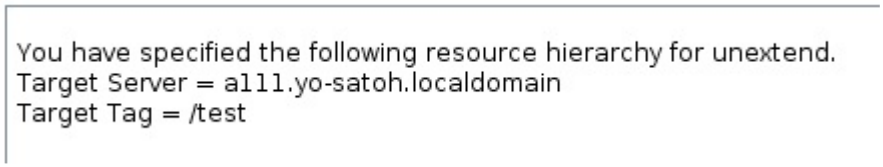
3. **[拡張解除する階層]** を選択します。

   **注記：** 左側のペインからグローバルリソースを右クリックするか、右側のペインから個々のリソースインスタンスを右クリックして拡張解除作業を選択した場合、 このダイアログボックスは表示されません。

   **[Next]** をクリックして次に進みます。

4. 拡張解除するために選択したターゲットサーバと VMDK リソース階層を確認する情報ボックスが表示されます。

   **[Unextend]** をクリックします。

5. VMDK リソースの拡張解除に成功したことを示す別の情報ボックスが表示されます。

6. **[Done]** をクリックして、 **[リソース階層の拡張解除]** メニューを終了します。

# 6.21.5.7. VMDK リソース階層のテスト

VMDK リソース階層のテストは、手動でサーバを切り替えて、プライマリーサーバーからバックアップサーバーへのリソースインスタンスのフェイルオーバをシミュレートすることによって行えます。

## LifeKeeper GUI からの手動切り替えの実行

LifeKeeper GUI で **[Edit] > [Resource] > [In Service]** を選択すると、手動切り替えを開始できます。たとえば、バックアップサーバーでサービス中リクエストが実行されると、VMDK リソース階層がバックアップサーバー側でサービス中になり、プライマリサーバー側ではサービス休止になります。この時点で、元のバックアップサーバーがプライマリサーバーになり、元のプライマリーサーバーがバックアップサーバーになります。

**[Our of Service]** を実行した場合、リソース階層はサービス休止になりますが、別のサ ーバーでサービス中にはなりません。

# 6.21.5.8. VMDK のメンテナンス

## ノード障害によるフェイルオーバーからの復旧

ESXi ホストの障害によりフェイルオーバーが発生した場合、仮想ハードディスクの切り離しが行えないため、停止したゲスト側にも仮想ハードディスクが接続されたままになります。
このため、復帰時に複数のゲストが仮想ハードディスクへアクセスを試みますが、その操作はハイパーバイザにより制限されるため、ゲストは起動できない状態になります。
このような状態が発生した場合には、 v Sphere client を使用して、ゲストから仮想ハードディスクを切り離すことで復帰できます。

## ESXi ログイン情報の変更

ESXi ホストのユーザー名･パスワードを変更する場合は以下の手順を実行してください。

1. LifeKeeper もしくはすべての VMDK リソースを停止します。
2. コマンドラインから以下のコマンドを実行して下さい。
   # /opt/LifeKeeper/lkadm/subsys/scsi/vmdk/bin/esxi_register -u <ESXi ホスト名>
   対話的に新しいユーザー名およびパスワードを入力してください。正常にログインできた場合には、新しい情報に更新されます。
   VMDK リソースが起動している場合やログインできない場合はエラーになり、情報は更新されません。
3. 停止した LifeKeeper もしくは VMDK リソースを起動します。
4. すべてのノードに対して同様の操作をおこない、ログイン情報を更新します。

## ESXi ホスト情報の削除

登録している ESXi ホストの情報を削除する場合には、以下の手順を実行して下さい。

1. LifeKeeper もしくはすべての VMDK リソースを停止します。
2. コマンドラインから以下のコマンドを実行して下さい。
   # /opt/LifeKeeper/lkadm/subsys/scsi/vmdk/bin/esxi_register -d <ESXi ホスト名>
3. 停止した LifeKeeper もしくは VMDK リソースを起動します。
4. すべてのノードに対して同様の操作をおこない、ホスト情報を削除してください。

## esxi_register詳細

| | |
|---|---|
| ホストの登録 | esxi_register -a <ESXi ホスト名> |
| ホストの削除 | esxi_register -d <ESXi ホスト名> |
| ログイン情報の更新 | esxi_register -u <ESXi ホスト名> |
| 登録済みホストの一覧 | esxi_register -l |

# 6.21.6. VMDK トラブルシューティング

| 症状 | 考えられる要因 |
|---|---|
| VMDKリソースをスイッチオーバーすると、元アクティブノードのシステムが停止する。 | 原因 : VMDKリソースのquickCheckデーモンのPIDファイル */tmp/LK-vmdk-** が削除されるとrecover処理により新しいquickCheckデーモンが起動します。その際古いデーモンは停止されません。リソースのremove時にはPIDファイルに対応するquickCheckデーモンのみを停止するため、古いデーモンがVMDKのデタッチを検知し、システムを停止させます。<br><br>対応 : PIDファイルを削除しないでください。<br>また、一般的に */tmp* 以下にある一定期間更新されていないファイルは自動的に削除されます。これはtmpwatchのcron実行やsystemd-tmpfilesなどによって実現されます。PIDファイルが削除されないように各仕組みに応じた除外設定を行って下さい。<br><br>設定例: systemd-tmpfiles の場合<br><br>`echo "x /tmp/LK-vmdk-*" > /etc/tmpfiles.d/lifekeeper.conf`<br><br>quickCheckデーモンが複数起動しているかどうかの確認方法と起動していた場合の対応方法<br><br>`for tag in `ins_list -f, -a scsi -r vmdk \| cut -d, -f4`; do echo -n "$tag: "; pgrep -f "vmdk_quickCheck.ps1.*$tag$" \| wc -l; done`<br><br>を実行して結果が “<vmdkタグ名>: 1” 以外の場合、<br><br>`pkill -INT -f "^/opt/LifeKeeper/bin/pwsh /opt/LifeKeeper/lkadm/subsys/scsi/vmdk/bin/vmdk_quickCheck.ps1"`<br>`pkill lkcheck`<br><br>を実行してください。 |
| ESXiホストの登録に失敗する。 | RHEL9系OSでESXiホストの登録時「No usable version of libssl was found」のエラーで失敗する場合があります。<br><br>原因 : PowerShellが依存するライブラリがシステムにインストールされていないため。<br><br>対応 : OSの提供するパッケージcompat-openssl11をインストールする。 |
| リソース作成時にマウントポイントが選択肢にでない。 | 以下の要因が考えられます。<br>• PowerShell / PowerCLIがインストールされていない<br>• ESXiホストが登録されていない |

| | |
|---|---|
| | • disk.enableUUIDパラメータが設定されていない<br>• 仮想ハードディスクが共有されていないデータストア上にある<br>• SCSIコントローラの共有設定が「仮想」もしくは「物理」になっている<br><br>/var/log/lifekeeper.logに詳細なエラーが記録されているので、内容を確認して設定を見直して下さい。 |
| VMDK リソースの In Service にかかる時間が長くなる。 | **原因** : In Service 時に行われる処理の中に ESXi ホスト上の稼働中の仮想マシン数に比例して時間のかかる処理が含まれているため。<br><br>**対応** : 根本修正は将来のリリースで検討中です。この問題の当面の回避策としては以下をご検討ください。<br>• VMDK リソースの数を減らす。<br>各 VMDK リソース毎に当該の処理を行うため、 VMDK リソースの数に比例してかかる時間が長くなります。 **1 つのリソース階層で複数のパーティションやファイルシステムが必要な場合** には、複数のVMDKリソースを使用するのではなく、単一のVMDKリソース上に複数のパーティションやファイルシステムを作成してください。<br>• クラスターノードに関係しない ESXi ホストが VMDK リソースに登録されていれば、それらのホストの登録を解除する。<br>ESXi ホスト情報の管理方法<br>• 稼働中の仮想マシンの数を減らす。 |

# 6.21.6.1. VMDK エラーメッセージ

本セクションでは、LifeKeeper VMDK リソース階層の作成、拡張時、またはリソースの削除、復元時に表示される可能性のあるメッセージの一覧を示します。必要に応じて、エラーの原因およびエラー状態を解消するために必要な処置についても説明しています。

その他の LifeKeeper コンポーネントからもメッセージが表示される可能性があります。 そのような場合は、適切な LifeKeeper コンポーネントのマニュアルを参照してください。

このセクションのメッセージは、次のトピックに分類されています。

- 共通エラー・メッセージ
- 階層の作成
- 階層の拡張
- 階層の削除、復元、リカバリー

## 共通エラーメッセージ

| エラー番号 | エラーメッセージ |
|---|---|
| 000002 | Usage error （使用法のエラー） |
| 000010 | Error getting resource information （リソース情報取得時のエラー） |
| 000011 | Both Tag and ID name not specified （タグ名および ID 名が指定されていません） |
| 000019 | Resource not found on local server （ローカル・サーバにリソースが見つかりません） |
| 000022 | END failed hierarchy <tag name> in service on server <server name> （サーバ <サーバ名> でサービス中の階層 <タグ名> の END が失敗しました） |
| 000026 | END failed ACTION for <tag name> on server <server name> due to <signal> signal （<シグナル> シグナルのためにサーバ <サーバ名> の <タ グ名> ACTION の END が失敗しました） |

## 階層の作成

| エラー番号 | エラーメッセージ |
|---|---|
| 000012 | Usage error（使用法のエラー） |
| 000013 | Error getting resource information（リソース情報取得時のエラー） |
| 000014 | Resource with either matching tag <tag name> or ID exists （タグ <タグ名> または ID が一致するリソースが存在します） |
| 000015 | ins_create failed on server <server name> （サーバ <サーバ名> で ins_create に失敗しました） |

| | |
|---|---|
| 000018 | Error creating resource <tag name> on server <server name> （サーバ <サーバ名> でリソース <タグ名> の作成エラー） |
| 000021 | Removing resource instance < tag name> from server <server name> due to an error during creation （作成時のエラーにより、リソース・インスタンス <タグ 名> をサーバ <サーバ名> から削除しています） |
| 000023 | Error bringing resource < tag name> in service on server <server name> （サーバ <サーバ名> でリソース <タグ名> をサービス中 にするときにエラー） |
| 000024 | Failed resource creation of resource < tag name> on server <server name> （サーバ <サーバ名> でリソース <タグ名> の作成に失敗しました） |
| 000027 | Removing file system dependency from <parent tag> to <child tag> on server <server name> due to an error during creation （作成時のエラーにより、サーバ <サーバ名> の <親タグ> から <子タグ> へのファイル・システム従属関係を削除し ています） |
| 000028 | Removing file system hierarchy <filesys tag> created by <parent tag> on server <server name> due to an error during creation （作成時のエラーにより、サーバ <サーバ名> の <親タグ> によって作成されたファイル・システム階層 <ファイルシステム・タグ> を削除しています） |
| 000029 | Switchback type mismatch between parent <parent tag> and child <child tag> on server <server name> （サーバ <サーバ名> の親 <親タグ> と子 <子タグ> のスイ ッチバック・タイプが一致していません）<br><br>**処置：** スイッチバック・タイプの不一致は予想外の動作を 引き起こすことがあります。ins_setas コマンドを使用して スイッチバック・タイプを手動で変更することによって、 この不一致をなくすことができます。 |
| 000030 | create: tag name not specified （作成時：タグ名が指定されていません）<br><br>または<br><br>extend: tag name not specified （拡張時：タグ名が指定されていません） |

## 階層の拡張

| エラー番号 | エラーメッセージ |
|---|---|
| 000003 | Template resource < tag name> on server <server name> does not exist （サーバ <サーバ名> にテンプレート・リソース <タグ名> が存在しません） |
| 000004 | Template resource < tag name> cannot be extended to server <server name> because it already exists there （テンプレート・リソース <タグ名> がサーバ <サーバ名> に すでに存在するため拡張できません） |
| 000005 | Cannot access canextend script on server <server name> （サーバ <サーバ名> の canextend スクリプトにアクセスできません） |
| 000006 | Cannot access extend script <path to extend> on server <server name> （サーバ <サーバ名> の extend スクリプト <extend へのパス> にアクセスできません） |

| | |
|---|---|
| 000007 | Cannot access depstoextend script <path to depstoexend> on server <server name> （サーバ <サーバ名> の depstoextend スクリプト <depstoextend へのパス> にアクセスできません） |
| 000008 | Cannot extend resource < tag name> to server <server name> （サーバ <サーバ名> にリソース <タグ名> を拡張できません） |
| 000009 | Either <templatesys> or <templatetag> argument missing （<テンプレート・システム> または <テンプレート・タグ> 引数が指定されていません） |
| 000014 | Resource with either matching tag <tag name> or ID exists （タグ <タグ名> または ID が一致するリソースが存在します） |
| 000015 | ins_create failed on server <server name> （サーバ <サーバ名> で ins_create に失敗しました） |
| 000018 | Error creating resource <tag name> on server <server name> （サーバ <サーバ名> でリソース <タグ名> の作成エラー） |
| 000025 | END failed resource extension of < tag name> on server <server name> due to a "<signal>" signal - backing out changes made to server （"<シグナル>" シグナルのためにサーバ <サーバ名> の <タグ名> リソース拡張の END が失敗しました - サーバに加えた変更を元に戻しています） |
| 000030 | create: tag name not specified （作成時：タグ名が指定されていません）<br><br>または<br><br>extend: tag name not specified （拡張時：タグ名が指定されていません） |

### 復元

| エラー番号 | エラーメッセージ |
|---|---|
| 000023 | Error bringing resource < tag name> in service on server <server name> （サーバ <サーバ名> でリソース <タグ名> をサービス中にするときにエラー） |

### リソース監視

| エラー番号 | エラーメッセージ |
|---|---|
| 000001 | Calling sendevent for resource < tag name> on server <server name> （サーバ <サーバ名> でリソース <タグ名> の sendevent を呼び出 しています） |

## VMDK Recovery Kit のエラーメッセージ

| エラー番号 | エラーメッセージ |
|---|---|

| | |
|---|---|
| 137000 | PowerShell is not installed. (PowerShell がインストールされていません) |
| 137001 | PowerCLI is not installed. (PowerCLI がインストールされていません) |
| 137002 | A valid network interface was not found. (ネットワークインタフェースが見つかりません) |
| 137003 | Attaching VMDK $vmdkfile. ($vmdkfile をアタッチします) |
| 137004 | VMDK already attached. (VMDKファイルがすでにアタッチ済みです) |
| 137005 | Failed to attach VMDK. (VMDKファイルのアタッチに失敗しました) |
| 137006 | Attach success. (VMDK をアタッチしました) |
| 137008 | Detaching VMDK $vmdkfile. ($vmdkfile をデタッチします) |
| 137009 | VMDK Already detached. (VMDKファイルがすでにデタッチ済みです) |
| 137010 | Failed to detach VMDK. (VMDKファイルのデタッチに失敗しました) |
| 137011 | Detach success. (VMDK をデタッチしました) |
| 137012 | Restarting VMDK status checker daemon. (監視デーモンを再起動します) |
| 137016 | Stopping VMDK status checker daemon. (監視デーモンを停止します) |
| 137020 | Failed to execute VMDK status checker daemon. (監視デーモンの起動に失敗しました) |
| 137026 | Flushing $dev ($dev のバッファをフラッシュします) |
| 137027 | Skipping flush for $dev. ($dev のバッファはフラッシュしません) |
| 137030 | Disk not specified. (ディスクが指定されていません) |
| 137031 | Cannot get disk uuid for $Disk. Please check your ESXi settings. (ディスクのIDが取得できないので、ESXiの設定を確認してください) |
| 137032 | PowerCLI failed. %s (PowerCLI のエラーです) |
| 137034 | Cannot bring VMDK resource \"%s\" in service on server \"%s\". (VMDK リソースの起動に失敗しました) |
| 137035 | Error detected conflict in expected tag name \"%s\" on target machine \"%s\". (タグ名が重複しています) |
| 137036 | Template resource \"%s\" on server \"%s\" does not exist. (テンプレートのタグ名が存在しません) |
| 137037 | This system is not a VMware guest. (VMwareのゲストではありません) |
| 137038 | Unable to find shared device on \"%s\" for \"%s\". (共有可能な VMDK がありません) |
| 137039 | Unable to find SCSI controller on \"%s\" for \"%s\". (SCSI コントローラが見つかりません) |
| 137050 | Failed to connect to ESXi server $addr. (ESXi ホストの接続に失敗しました) |
| 137051 | There is no ESXi server connected. (ESXi ホストに接続していません) |
| 137055 | Cannot determine ESXi VM ID because multiple network interfaces were found with the MAC address $MAC_ADDR. ($MAC_ADDR のネットワークインタフェースが複数見つかったため、自分のIDを取得できません) |

| | |
|---|---|
| 137056 | Failed to $action VMDK $VMDK_FILENAME. Retrying $action in $wait_sec milliseconds. ($VMDK_FILENAME の操作に失敗したため、$wait_sec 秒間待機します) |
| 137057 | Usable SCSI controller not found. (使用可能な SCSI コントローラが見つかりません) |
| 137058 | Cannot find VMDK with ID $UUID. ($UUID の VMDK が見つかりません) |
| 137059 | This guest has snapshots present. (スナップショットがあります) |
| 137060 | The VMDK with ID $UUID cannot be attached to this guest. ($UUID の VMDK はこのゲストにアタッチできません) |
| 137061 | The virtual storage controller has an incompatible sharing mode configured. (SCSI コントローラの共有モードが正しく設定されていません) |
| 137062 | VMDK_TIMER too short. Using default value. (VMDK_TIMER の設定値が短すぎるため、初期値を使用します) |
| 137063 | Calling sendevent for resource `”$Tag`” on server `”$me`”“ (リソース $Tag でイベントが発生しました) |
| 137064 | skip quickcheck for `”$Tag`” on server `”$me`”, sendevent pending. (イベント発生中のため q監視処理を中止します) |
| 137065 | sendevent issued for tag `”$Tag`” has not finished, halt server `”$me`” (イベントの通知に失敗しました) |
| 137066 | skip quickcheck for `”$Tag`” on server `”$me`”, sendevent pending. (イベント発生中のため q監視処理を中止します) |
| 137068 | The VMDK detection failed. Retry count exceeded. (規定時間内にディスクが認識されません) |
| 137070 | Connect failed. (ESXi に接続できません) |
| 137071 | Get-LocalVM failed. (自VMの情報が取得できません) |
| 137072 | The VMDK is remote detached. This server has lost ownership. (リモートでデタッチ要求が発行されたため、VMDK のアクセス権を失いました) |
| 137073 | The VMDK quickCheck daemon has been stopped. (監視デーモンが停止しています) |
| 137074 | VMDK_RETRY too small. Using default value. (VMDK_RETRY が小さすぎるため、初期値を使用します) |
| 137075 | Cannot find virtual SCSI controller $CONTROLLER. (SCSIコントローラーがありません) |
| 137076 | The virtual storage controller has an incompatible sharing mode configured. (SCSIコントローラーの設定が正しくありません) |
| 137077 | Cannot find VM with MAC address $MAC_ADDR. ($MAC_ADDR のネットワークを使用するVMがありません) |
| 137078 | Cannot find VM with MAC address $MAC_ADDR. ($MAC_ADDR のネットワークを使用するVMがありません) |
| 137100 | Re-reading partition table on %s. (パーティション情報を再読込します) |

| | |
|---|---|
| 137101 | Partition information not defined for %s on %s. Retry. (パーティション情報を読み込めないので再試行します) |
| 137102 | Partition information not defined for %s on %s. (パーティションがありません) |
| 137103 | Resource %s is OSF, Skip flushing buffers. (リソースが OSF 状態なので、バッファのフラッシュを行いません) |
| 137104 | Flushing buffers on %s. (バッファをフラッシュします) |
| 137105 | Device not specified. (デバイス名の指定がありません) |
| 137106 | Cannot get device uuid for $Device. Please check your ESXi settings. (デバイスのUUIDが取得できないため、ESXiの設定を確認してください) |
| 137107 | %s is not shareable with any machine. (VMDK が共有されていません) |
| 137109 | Creating dependency \”%s\”-\”%s\” on machine \”%s\”. (依存関係を作成します) |
| 137110 | Dependency \”%s\”-\”%s\” on machine \”%s\” exists. (すでに依存関係が存在します) |
| 137111 | Failed to create dependency \”%s\”-\”%s\” on machine \”%s\”. (依存関係の作成に失敗しました) |
| 137112 | Cannot bring VMDKP resource \”%s\” in service on server \”%s\”. (VMDKP リソースの起動に失敗しました) |
| 137113 | detected conflict in expected tag name \”%s\” on target machine \”%s\”. (タグ名が重複しています) |

# 7. パラメーター一覧

> ! LCMHBEATTIME は、デフォルトから変更しないでください。ネットワークやシステムに大きな負荷がかかると、LCMBEATTIME が 5 秒未満の場合、予期せぬハートビートの欠落が発生することがあります。各サーバーが「生きている」と判断するためのチューニングは、LCMNUMHBEATS を変更して行うことを強く推奨します。

## Coreパラメーターの一覧

下記の表は、Core パラメーター名とその意味を説明しています。これらの値は */etc/default/LifeKeeper* 設定ファイルを編集することにより設定可能です。

| パラメーター名 | パラメーターの意味 | 設定値 | デフォルト値 | パラメーター適用タイミング | 備考 |
|---|---|---|---|---|---|
| FS_KERNEL_RETRIES | SIGTERMやSIGKILLシグナルによりファイルシステムにアクセスしているプロセスが停止できなかった場合に、forceumountスクリプトがファイルシステムのアンマウントを試行する最大回数。アンマウントは最大回数を超えるまで3秒毎に試行されます。 | 整数値 | 60 | 適宜 (すぐに反映されます) | |
| FS_UMOUNT_RETRIES | アンマウントしたいファイルシステムにアクセスしている全プロセスを停止するためにforceumountスクリプトがSIGTERMシグナルを発行する最大回数。試行回数が最大回数を超えた場合、forceumountスクリプトはSIGKILLシグナルによるプロセス停止を最大3回試行します。 | 整数値 | 1 | 適宜 (すぐに反映されます) | |
| REMOTETIMEOUT | 他のマシンから予測される応答がくる前に”lcdsendremote”機能により | 整数値 | 900 | LifeKeeper 起動時 (**LifeKeeper再起動時** | |

| | | | | | |
|---|---|---|---|---|---|
| | 送信要求処理をするための時間です。この時間を経過しても応答がない場合は、代替の経路(利用可能な場合)での処理を試みます。この値によってタイムアウトと判定された場合には、リソース切り替えに失敗します。常にタイムアウトが発生するような環境である場合には、この値を伸ばすことによって対応する場合があります。 | | | に反映されます) | |
| CONFIRMSODEF | フェイルオーバー確認を設定している時、マシンのフェイルオーバー処理で行われるデフォルト動作を指定します。デフォルト動作は、タイムアウトの間 （CONFIRMSOTOを参照してください）に管理者から手動で応答がこない場合のみ、行われます。 | 0: フェイルオーバーを続行します。<br>1: フェイルオーバーをブロックします。 | 0 | 適宜 (すぐに反映されます) | ［Confirm Failover］と［[Block Resource Failover］の設定 を参照してください。 |
| CONFIRMSOTO | フェイルオーバー確認を設定している時、管理者からの指示が行われるまでの待機時間を秒単位で指定します。タイムアウト時間が経過すると、CONFIMRSODEF で指定したデフォルト動作が行われます。そうでなければ、管理者が指示した動作を実行します。 | 整数値 | 600 | 適宜 (すぐに反映されます) | マシンのフェイルオーバーの処理は管理者からの確認を受け取るか、タイムアウト時間が終了するまで遅延します。<br>［Confirm Failover］と［Block Resource Failover］の設定 を参照してください。 |
| FAILFASTTIMER | ローカルシステムによってSCSI RESERVEコマンドによりデバイス | 整数値 | 5 | LifeKeeper 起動時 (**LifeKeeper再起動時** | |

| | | | | | |
|---|---|---|---|---|---|
| | にリザーブをかける間隔です。リザーブを失った場合は、システムは停止、再起動します。 | | | に反映されます) | |
| SCSIERROR | SCSIデバイスをオープンできない、接続できない、あるいは他のSCSIエラー(タイムアウト等)が発生したときの動作を決定します。 | event: スイッチオーバー<br>halt: データの破損を避けるため即時強制停止、強制再起動 | event | LifeKeeper 起動時 (**LifeKeeper再起動時**に反映されます) | SCSIリザベーションが失われた場合は、本パラメーターの設定にかかわらず常に停止になります。 |
| LKCHECKINTERVAL | アプリケーションの監視をする間隔を秒単位で指定します。値を0にした場合、監視が無効になります。 | 0または1以上の整数値 | 120 | LifeKeeper 起動時 (**LifeKeeper再起動時**に反映されます) | LifeKeeperをアップグレードすると、デフォルト値に上書きされます。設定を変更している場合は、再設定してください。 |
| FILESYSFULLWARN | LifeKeeperログに警告メッセージを出力し始める、ファイルシステム使用量の閾値を指定します。 0 を設定した場合、監視が無効になります。 | 整数値 | 90 | 適宜 (すぐに反映されます) | |
| FILESYSFULLERROR | LifeKeeperログにエラーメッセージを出力し始める、ファイルシステム使用量の閾値を指定します。<br>さらに、この閾値に達した時にはLKROOT/events/filesys/diskfull/notify スクリプトが呼び出されます。<br>0 を設定した場合、監視が無効になります。 | 整数値 | 95 | 適宜 (すぐに反映されます) | |

| | | | | | |
|---|---|---|---|---|---|
| LK_TRAP_MGR | SNMPトラップを受けるネットワークを１つまたは複数（カンマ区切り）指定します。この値を指定していない場合は、SNMPトラップの送信は行いません。 | 文字列 | （未設定） | 適宜 (すぐに反映されます) | /opt/LifeKeeper/bin/lk_configsnmp を使用して、この値を設定できます。設定値にスペースを含めないでください。特に複数指定する場合、カンマの前後にスペースを含めないでください。 |
| LK_NOTIFY_ALIAS | LifeKeeperクラスターが特定のイベントが発生したときに通知するメッセージを受けとるために使用するEmailアドレスまたはアドレスリストです。値を指定していない場合は、通知を行いません。使用例は次の通りです。<br>LK_NOTIFY_ALIAS=<br>– 通知を行いません。<br>LK_NOTIFY_ALIAS=user1@domain1<br>– domain1のuser1に通知を行います。<br>LK_NOTIFY_ALIAS=user1@domain1,user2@domain1<br>– domain1のuser1、user2に通知を行います。 | 文字列 | （未設定） | 適宜 (すぐに反映されます) | /opt/LifeKeeper/bin/ lk_confignotifyalias を使用して、この値を設定できます。 |
| LKSYSLOGTAG | syslog出力時に使用するタグ名です。 | 文字列 | LifeKeeper | LifeKeeper 起動時 (**LifeKeeper再起動時**に反映されます) | |
| LKSYSLOGSELECTOR | syslogのファシリティです。 | user, daemon, local0, local1, …または local7 | local6 | LifeKeeper 起動時 (**LifeKeeper再起動時**に反映されます) | |
| LCMHBEATTI | ハートビート信号を送 | 整数値 | 5 | LifeKeeper 起 | 0を設定する |

| | | | | | |
|---|---|---|---|---|---|
| ME | 信する間隔を秒単位で指定します。またこの時間内に対向サーバーからハートビート信号を含むLCM信号を受信できなかった場合、ハートビート停止と判断されます。 | | | 動時 (**LifeKeeper再起動時**に反映されます) | と、デフォルト値が設定されます。 詳細はテクニカルドキュメンテーション「LifeKeeper ハートビートの調整」を参照してください。 |
| LCMNUMHBEATS | コミュニケーションパスが切断していると判断するハートビートの連続失敗回数を指定します。<br>実装上は回数ではなく、LCMHBEATTIME x LCMNUMHBEATS秒間LCM通信がない場合、コミュニケーションパス切断と判断されます。 | 整数値 | 3 | LifeKeeper 起動時 (**LifeKeeper再起動時**に反映されます) | 0を設定すると、デフォルト値が設定されます。 詳細はテクニカルドキュメンテーション「LifeKeeper ハートビートの調整」を参照してください。 |
| LC_MESSAGES | 言語環境を変更します。 | 文字列 | C | LifeKeeper 起動時 (**LifeKeeper再起動時**に反映されます) | 値を変更する場合、LifeKeeperの動作に悪影響を及ぼす可能性があることに注意してください。悪影響は、メッセージカタログがさまざまな言語とユーティリティに対応してインストールされているかどうか、および LifeKeeperが予期していないテキスト出力をそれらが生成するかどうかによって異なります。 |
| GUI_WEB_PORT | LifeKeeper Management Web servers (lkGUI) | 整数値 | 81 | steeleye-lighttpd再起動 | steeleye-lighttpdの再起動方法： |

| | | | | | |
|---|---|---|---|---|---|
| | に使用されるポートを指定します。 | | | | /opt/LifeKeeper/sbin/sv restart /opt/LifeKeeper/etc/service/steeleye-lighttpd |
| API_SSL_PORT | LifeKeeper APIで使用するポートを指定します。 | 整数値 | 778 | steeleye-lighttpd再起動 | steeleye-lighttpdの再起動方法：/opt/LifeKeeper/sbin/sv restart /opt/LifeKeeper/etc/service/steeleye-lighttpd |
| LOGMGR_LOGLEVEL | Generic Applicationのログレベルを指定します。 | LK_INFO または LK_ERROR | LK_ERROR | LK 起動時もしくはlk_logmgrプロセスの再起動(**LifeKeeper再起動時に反映されます**) | |
| RESERVATIONS | RESERVATIONS パラメーターは、SCSI リザベーションコマンドを発行するために使用される方法を決定します。 SCSI リザベーションは、１つのシステムだけがクラスター内のデータにアクセスできるように、I/O フェンシングを実行するために使用されます。<br>有効な設定は次の通りです。<br>ioctl : 中間層の SCSI ドライバーで ioctl コマンドを使用して、リザーブとリリースのコマンドを発行します。<br>none : lkscsid デーモンは、共有 SCSI デバイスの監視のみを実行します。 それは、SCSI リザベーションを維持しません。 この設定をするときは、クラスタ | ioctl<br>none | ioctl | LifeKeeper 起動時 (**LifeKeeper 再起動時に反映されます**) | |

| | | | | | |
|---|---|---|---|---|---|
| | 一内のデータの信頼性を保証するために、別の方法 で I/O フェンシングを構成する必要があります。この設定は、制限された構成とデバイスのセットでのみ、LifeKeeper によってサポートされます。 | | | | |
| STANDBY_NODE_WRITE_PROTECTION | リザベーションを無効 ( RESERVATIONS=none ) に設定した時、スタンバイノードから共有ストレージへの書き込みを禁止にする機能です。<br>この機能を有効にする場合は、 SNHC と SNHC_DISKCHECK も有効にする必要があります。詳細は Standby Node Health Check パラメータ一覧 を参照ください。 | enable<br>disable | enable | 適宜 (すぐに反映されます) | 詳細は、STANDBY NODE WRITE PROTECTION 機能 を参照してください。 |

# 7.1. Recovery Kit for EC2パラメーター一覧

下記の表は、Recovery Kit for EC2パラメーター名とその意味を説明しています。これらの値は */etc/default/LifeKeeper* 設定ファイルを編集することにより設定可能です。
Recovery Kit for EC2 のコンポーネントはメモリに常駐していないため、 */etc/default/LifeKeeper* の値を変更すると即時に反映されます。LifeKeeper を再起動する必要はありません。

| パラメーター名 | パラメーターの意味 | 設定値 | デフォルト値 | 備考 |
|---|---|---|---|---|
| EC2_RESTORE_TIMEOUT | リソースのリストアタイムアウト時間を秒単位で指定します。 | 整数値 | 300 | |
| EC2_REMOVE_TIMEOUT | リソースを停止するタイムアウト時間を秒単位で指定します。 | 整数値 | 300 | |
| EC2_RECOVER_TIMEOUT | ローカルリカバリのタイムアウト時間を秒単位で指定します。 | 整数値 | 300 | |
| EC2_QUICKCHECK_TIMEOUT | quickCheck のタイムアウト時間を秒単位で指定します。 | 整数値 | 100 | |
| EC2_AWS_REGION | EC2 リソースが動作するリージョンを指定します。 | 文字列 | (未設定) | |
| IP_NOLINKCHECK | 保護対象のネットワークインターフェースのリンクチェックを無効に設定します。 | 0: 有効<br>1: 無効 | 0 | 本パラメーターは Elastic IP を保護しているときにのみ適用されます。 |
| IP_WAIT_LINKDOWN | 保護対象のネットワークインターフェースをダウン状態にしてからアップ状態に戻すまで待機する秒数を指定します。一部の環境では、この 2 つの動作の間に待機時間が必要です。 | 整数値 | 5 | 本パラメーターは Elastic IP を保護しているときにのみ適用されます。 |
| IP_MAX_LINKCHK | リンクが修復されてからアップ状態に戻るまで待機する最大時間を秒単位 | 整数値 | 5 | 本パラメーターは Elastic IP を保 |

| | | | | |
|---|---|---|---|---|
| | で指定します。一部の環境では、この値を増加する必要があります。 | | | 護しているときにのみ適用されます。 |
| AWSCLI_CONNECT_TIMEOUT | awsコマンドのオプション—cli–connect–timeoutの値を秒単位で設定します。 | 整数値 | 10 | 本パラメーターは Route53 と共通のパラメーターです。 |
| AWSCLI_READ_TIMEOUT | awsコマンドのオプション—cli–read–timeout の値を秒単位で設定します。 | 整数値 | 5 | 本パラメーターは Route53 と共通のパラメーターです。 |
| HTTP_PROXY<br>HTTPS_PROXY<br>NO_PROXY | サービスエンドポイントへのアクセスで HTTP プロキシを使用する場合に設定してください。ここで設定した値がそのまま AWS CLI へ渡されます。詳しくはAWSのドキュメントを参照してください。<br>docs.aws.amazon.com/ja_jp/cli/latest/userguide/cli-http-proxy.html | 文字列 | （未設定） | 本パラメーターは Route53 や Quorum と共通のパラメーターです。 |
| EC2_NO_SRC_DEST_CHECK_QUICKCHECK | quickCheck及びrestore時に、送信元/送信先チェックが無効になっているか確認します。restore時はEC2_NO_SRC_DEST_CHECKの設定に従った処理を行います。 | 0:確認しない<br>1:確認する | 1（新規インストール時はこの値で環境変数定義が追加されます。未定義の場合は0として扱われます） | 本パラメーターは Route Table を保護しているときにのみ適用されます。 |
| EC2_NO_SRC_DEST_CHECK | 送信元/送信先チェックの設定を無効化します。 | 0:無効化しない<br>1:無効化する | 1 | 本パラメーターは Route Table を保護しているときにのみ適用され |

| | | | | |
|---|---|---|---|---|
| | | | | ます。 |

# 7.2. IPパラメーター一覧

下記の表は、IPパラメーター名とその意味を説明しています。これらの値は */etc/default/LifeKeeper* 設定ファイルを編集することにより設定可能です。

| パラメーター名 | パラメーターの意味 | 設定値 | デフォルト値 | パラメーター適用タイミング | 備考 |
|---|---|---|---|---|---|
| IP_PINGTRIES | IPヘルスチェック実行時のpingのリトライ数を指定します。 | 整数値 | 3 | 適宜(すぐに反映されます) | |
| IP_PINGTIME | LifeKeeperがIPヘルスチェック時に、pingの パケット１個の 応答を待つ時間を秒単位で指定します。 | 整数値 | 1 | 適宜(すぐに反映されます) | ブロードキャスト ping の替わりに手動で設定した Ping List を使用する場合、この値を 3 秒以上に設定しても効果は得られません。Linux の TCP/IP の実装では、Ping List コマンドにタイムアウト値を設定したとしても、3 秒以上応答を受け取れない場合は常に、「Destination Host Unreachable」エラーを返すためです。 |
| IP_QUICKCHECK_TIMEOUT | クイックチェックが誤判定することがあれば/etc/default/LifeKeeperの設定値を延ばしてください。 | 整数値 | 12 | 適宜(すぐ | この値を設定するために LifeKeeper のプロセスの停止もしくは再起動を行う必要はありません |

| | | | | | |
|---|---|---|---|---|---|
| | IPクイックチェックが完了するまでのデフォルトの時間は12秒です。 | | | に反映されます) | |
| NOIPUNIQUE | IPリソースをIn Serviceにする時、同じIPアドレスがネットワーク上に存在していないかチェックする機能を無効に設定します。デフォルトでは、LifeKeeperはIPアドレスが他のネットワーク上で使用されていないかどうか確認します。 | 0: 有効 1: 無効 | 0 | 適宜(すぐに反映されます) | |
| NOBCASTPING | IPリソースのヘルスチェック時のブロードキャストpingメカニズムを無効に設定します。 | 0: 有効 1: 無効 | 0 | 適宜(すぐに反映されます) | |
| IP_NOLINKCHECK | IPヘルスチェック実行時にリンク状態を確認する設定を無効に設定します。 | 0: 有効 1: 無効 | 0 | 適宜(すぐに反映されます) | この設定は、特に「仮想IPのリンクチェックに失敗しました」というメッセージがログに出力された場合、仮想環境で無効にする必要がある場合があります。 |
| IP_MAX_LINKCHK | リンクが修復されてからアップ状態に戻るまで待機する最大時間を秒単位で指定します。一部の環境では、この値を増加す | 整数値 | 5 | 適宜(すぐに | |

| | | | | | |
|---|---|---|---|---|---|
| | る必要があります。 | | | 反映されます) | |
| IP_WAIT_LINKDOWN | 保護対象のネットワークインターフェースをダウン状態にしてからアップ状態に戻すまで待機する時間を秒単位で指定します。一部の環境では、この 2 つの動作の間に待機時間が必要です。 | 整数値 | 5 | 適宜(すぐに反映されます) | |

# 7.3. LB Health Check パラメーター一覧

設定可能なパラメーター一覧を下記に記載します。

| パラメーター | 説明 |
| --- | --- |
| HC_TIMEOUT | ロードバランサーからの接続タイムアウト値として設定します。指定された秒数のあいだ接続がない場合は障害として扱われ、フェイルオーバーを実行します。デフォルト値は 60 秒です。このパラメータは、/etc/default/LifeKeeper で設定します。<br><br>**設定例:**<br>HC_TIMEOUT=30 |
| HC_READ_TIMEOUT | ロードバランサーからのデータ到達までのタイムアウト値として設定します。指定された秒数だけデータがない場合、ロードバランサーに応答文字列を送信し、その後TCPコネクションを閉じます。デフォルト値は 10 秒です。このパラメータは、/etc/default/LifeKeeper で設定します。<br><br>**設定例:**<br>HC_READ_TIMEOUT=20 |

# 7.4. MQパラメーター一覧

下記の表は、MQパラメーター名とその意味を説明しています。これらの値は */etc/default/LifeKeeper* 設定ファイルを編集することにより設定可能です。

| パラメーター名 | パラメーターの意味 | 設定値 | デフォルト値 | パラメーター適用タイミング | 備考 |
|---|---|---|---|---|---|
| MQS_QUICKCHECK_TIMEOUT_SC | サーバ接続チェックのタイムアウト時間を秒単位で指定します。 | 整数値 | 10 | 適宜(すぐに反映されます) | |
| MQS_QUICKCHECK_TIMEOUT_CC | クライアント接続チェックのタイムアウト時間を秒単位で指定します。 | 整数値 | 10 | 適宜(すぐに反映されます) | |
| MQS_QUICKCHECK_TIMEOUT_PUTGET | PUT/GETチェックのタイムアウト時間を秒単位で指定します。 | 整数値 | 10 | 適宜(すぐ | |

| | | | | | |
|---|---|---|---|---|---|
| | | | | に反映されます) | |
| MQS_QUICKCHECK_TIMEOUT_PS | publish/subscribe の使用有無チェックのタイムアウト時間を秒単位で指定します。 | 整数値 | 5 | 適宜(すぐに反映されます) | |
| MQS_QUICKCHECK_TIMEOUT_CLUSTER | キューマネージャが WebSphere MQ クラスタの一部であるかどうかチェックするタイムアウト時間を秒単位で指定します。 | 整数値 | 5 | 適宜(すぐに反映されます) | |
| MQS_QUICKCHECK_TIMEOUT | quickCheck スクリプトのタイムアウト時間を秒単位で指定します。 | 整数値 | 40 | 適宜(すぐに反映されます) | 10以下を設定した場合はデフォルト値が設定されます。 |
| MQS_QMGR_START_TIMEOUT | キューマネージャの起動コマンド完了のタイムアウト時間を秒単位で指定します。 | 整数値 | 60 | 適宜(すぐに | |

| | | | | | |
|---|---|---|---|---|---|
| | | | | 反映されます) | |
| MQS_CMDS_START_TIMEOUT | コマンドサーバの起動コマンド完了のタイムアウト時間を秒単位で指定します。 | 整数値 | 30 | 適宜(すぐに反映されます) | |
| MQS_LISTENER_START_TIMEOUT | リスナーの起動 コマンド完了のタイムアウト時間を秒単位で指定します。 | 整数値 | 30 | 適宜(すぐに反映されます) | |
| MQS_LISTENER_LIST_TIMEOUT | リスナーの list コマンド完了のタイムアウト時間を秒単位で指定します。 | 整数値 | 10 | 適宜(すぐに反映されます) | |
| MQS_CHECK_TIMEOUT_ACTION | サーバ接続チェック、またはクライアント接続チェックがタイムアウトになった場合のアクションを指定し | ignore: ログに記録するが、リカバリは行 | ignore | 適宜(すぐに反 | |

| | | | | | |
|---|---|---|---|---|---|
| | ます。 | わない sendevent: ローカル リカバリ を行う | | 映されます) | |
| MQS_LISTENER_CHECK_DELAY | リスナーの起動から、リスナー正常起動チェックを行うまでの時間を秒単位で指定します。デフォルト値で、使用状態のポートを検出するのに十分です。 | 整数値 | 2 | 適宜(すぐに反映されます) | |
| NO_AUTO_STORAGE_DEPS | MQ リソース階層作成の間、キューマネージャとログストレージディレクトリに、共有ストレージの確認とファイルシステムリソースの作成を行うかどうか決定します。<br>本パラメーターを 0 に設定すると、これらのタスクが実行されます。1 に設定すると、これらのタスクを回避します。 | 0 または 1 | 0 | 適宜(すぐに反映されます) | |
| MQS_DSPMQVER_TIMEOUT | dspmqver コマンド（WebSphere MQ のバージョンを知るために必要）のタイムアウト時間を秒単位で指定します。 | 整数値 | 5 | 適宜(すぐに反映されます) | 2以下を設定した場合はデフォルト値が設定されます。 |

| | | | | | |
|---|---|---|---|---|---|
| MQS_SKIP_CRT_MISSING_Q | 欠落したテストキューを自動で作成するかどうかを指定します。<br>本パラメーターを 0 に設定すると、欠落したテストキューは自動で作成されます。1 に設定すると、このプロセスはスキップされます。 | 0 または 1 | 0 | 適宜(すぐに反映されます) | |
| MQS_ALT_USER_NAME | すべての WebSphere MQコマンドに別のユーザーネームを使用。デフォールトでは、ユーザー”mqm”が使用される。もし設定されている場合には、ユーザーはそのプライマリーグループをグループ”mqm”に設定するか、そのグループにセカンダリーメンバーシップを持たなくてはならない。 | 文字列 | 設定されていない、もしくはユーザーが”mqm”グループにメンバーシップを持っていない場合、mqm | 適宜(すぐに反映されます) | もし、WebSphere MQのアドオンパッケージが”mqm”以外のユーザーを必要とする場合にのみ設定 |

# 7.5. NFSパラメーター一覧

下記の表は、NFSパラメーター名とその意味を説明しています。これらの値は */etc/default/LifeKeeper* 設定ファイルを編集することにより設定可能です。

| パラメーター名 | パラメーターの意味 | 設定値 | デフォルト値 | パラメーター適用タイミング | 備考 |
|---|---|---|---|---|---|
| RESTARTMOUNTD | NFSリストア時にrpc.mountの停止と再起動を行います。 | true: 有効 false: 無効 | true | 適宜(すぐに反映されます) | |
| NFSEXPORTFSTIMER | exportfs コマンドがハングアップしたと判断する待機時間を秒単位で指定します。exportfs コマンドがこの時間内に終了しなかった場合、ハングアップしたと判断します。そして、exportfs コマンドを呼び出しているプロセスとその子プロセスに対して TERM シグナルを送信し、それでも終了しなかったプロセスには KILL シグナルを送信します。 | 整数値 | 30 | 適宜(すぐに反映されます) | |

# 7.6. Recovery Kit for Oracle Cloud Infrastructure パラメーター一覧

下記の表は、Recovery Kit for Oracle Cloud Infrastructure (RK for OCI) が使用するパラメーター名とその意味を説明しています。これらの値は */etc/default/LifeKeeper* 設定ファイルを編集することにより設定可能です。
RK for OCI のパラメーターは */etc/default/LifeKeeper* の値を変更すると即時に反映されます。LifeKeeper を再起動する必要はありません。

| パラメーター名 | パラメーターの意味 | 設定値 | デフォルト値 |
|---|---|---|---|
| OCIVIP_RESTORE_TIMEOUT | リソースのリストアタイムアウト時間を秒単位で指定します。 | 整数値 | 60 |
| OCIVIP_REMOVE_TIMEOUT | リソースを停止するタイムアウト時間を秒単位で指定します。 | 整数値 | 60 |
| OCIVIP_RECOVER_TIMEOUT | ローカルリカバリーのタイムアウト時間を秒単位で指定します。 | 整数値 | 60 |
| OCIVIP_QUICKCHECK_TIMEOUT | quickCheck のタイムアウト時間を秒単位で指定します。 | 整数値 | 60 |
| OCI_CLI_PROFILE | OCI CLI が使用するプロファイル名を指定します。<br><br>詳しくは OCI のドキュメントを参照してください。<br>https://docs.oracle.com/ja-jp/iaas/Content/API/SDKDocs/cliconfigure.htm | 文字列 | なし |

# 7.7. Oracleパラメーター一覧

下記の表は、Oracleパラメーター名とその意味を説明しています。これらの値は */etc/default/LifeKeeper* 設定ファイルを編集することにより設定可能です。

| パラメーター名 | パラメーターの意味 | 設定値 | デフォルト値 | パラメーター適用タイミング | 備考 |
|---|---|---|---|---|---|
| ORACLE_ORATABLOC | デフォルトでは、/etc/oratab が使用されます。<br>代替oratabファイルのあるディレクトリを指定します。<br>通常は /etc/oratab を参照しますが、そのファイルが存在しない場合には、このパラメーターで指定されたディレクトリにある oratab を読み込みます。 | 文字列 | /var/opt/oracle | 適宜(すぐに反映されます) | |
| LK_ORA_NICE | 最大許容接続に達した時に、原因となるデータベース接続障害に対してリカバリを試みるかどうかを指定します。<br>最大許容接続に達した時のリカバリの実行は、スタンバイノードへのフェイルーバの原因となることがあります。 | 0:リカバリ実行<br>1:リカバリ回避 | 0 | 適宜(すぐに反映されます) | |
| ORACLE_RESTORE_TIMEOUT | Oracleリソース起動時のタイムアウト時間を秒単位で指定します。 | 整数値 | 300 | 適宜(すぐに反映されます) | |
| ORACLE_REMOVE_TIMEOUT | Oracleリソース停止時のタイムアウト時間を秒単位で指定します。 | 整数値 | 300 | 適宜(すぐに反映されます) | |

# 7.8. PostgreSQLパラメーター一覧

下記の表は、PostgresSQLパラメーター名とその意味を説明しています。これらの値は */etc/default/LifeKeeper* 設定ファイルを編集することにより設定可能です。

| パラメーター名 | パラメーターの意味 | 設定値 | デフォルト値 | パラメーター適用タイミング | 備考 |
|---|---|---|---|---|---|
| LKPGSQL_KILLPID_TIME | プロセス ID が停止した後、そのプロセスに対する再チェックを行うまでの時間を秒単位で指定します。 | 整数値 | 3 | 適宜 (すぐに反映されます) | デフォルト値より小さい場合は、デフォルト値が設定されます。 |
| LKPGSQL_CONN_RETRIES | 旧 LKPGSQLMAXCOUNT – 操作 （開始もしくは停止） を行った後、クライアント接続を試みる回数を指定します。 | 整数値 | 12 | 適宜 (すぐに反映されます) | デフォルト値より小さい場合は、デフォルト値が設定されます。 |
| LKPGSQL_ACTION_RETRIES | アクションコマンドに失敗するまで、開始と停止を試行する回数を指定します。 | 整数値 | 4 | 適宜 (すぐに反映されます) | デフォルト値より小さい場合は、デフォルト値が設定されます。 |
| LKPGSQL_STATUS_TIME | status コマンドのタイムアウト時間を秒単位で指定します。 | 整数値 | 17 + (3 * LKPGSQL_KILLPID_TIME) | 適宜 (すぐに反映されます) | デフォルト値より小さい場合は、デフォルト値が設定されます。 |
| LKPGSQL_QCKHANG_MAX | データベースインスタンスがフェイルオーバ / sendevent を発生させるまでに許容される、quickCheckがハングする回数を指定します。 | 整数値 | 2 | リソース作成時 | 1より小さい場合は、デフォルト値が設定されます。 |
| LKPGSQL_CUSTOM_DAEMON | postgres デーモン（postgres.bin,postmaster,postgres,edb-postgres） に対する追加の別名の | 文字列 | （未設定） | リソース作成時 | |

| | | | | | |
|---|---|---|---|---|---|
| | 指定を許可します。 | | | | |
| LKPGSQL_IDIRS | 旧 LKPGSQL_IPORTS – immediate オプションのみを使用してシャットダウンするインスタンスに対する datadir のエントリを含みます。 | 文字列 | （未設定） | 適宜 (すぐに反映されます) | |
| LKPGSQL_SDIRS | Smart オプションを使用してシャットダウンするインスタンスに対する datadir のエントリを含みます。 | 文字列 | （未設定） | 適宜 (すぐに反映されます) | |
| LKPGSQL_DISCONNECT_CLIENT | データベースに障害が発生している間の PostgreSQL リソース階層の振る舞いを制御します。本パラメーターを有効にすると、クライアントプロセスに SIGTERM シグナルが送信され、データベースから強制的に切断されます。この処置は、postmasterプロセスがローカルリカバリ中に稼働していない場合のみに取ることができます。 | 0: 無効<br>1: 有効 | 1 | 適宜 (すぐに反映されます) | PostgreSQL 8.2 以降では、本パラメーターをご利用いただけません。 |
| LKPGSQL_DISCONNECT_CLIENT_BYTAG | LKPGSQL_DISCONNECT_CLIENT と類似していますが、この設定は、処置をこの設定項目で指定したカンマで区切られたタグのリストに限定します。 | 文字列 | （未設定） | 適宜 (すぐに反映されます) | PostgreSQL 8.2 以降では、本パラメーターをご利用いただけません。 |
| LKPGSQL_RESUME_PROC | プロセスの停止状態の検出時 （プロセス状態が T ）、再開するか無視するかを決定します。 | 0: 無視<br>1: 再開 | 1 | 適宜 (すぐに反映されます) | |
| LKPGSQLDEBUG | PostgreSQL database kit および postgres データベースのデバッグを有効にします。0 ～ 5 の値が有効です。<br>この設定項目は、オプション -d <LKPGSQLDEBUG> を使用して postmaster データベースへと渡されます。 | 0～5 の整数値 | 0 | 適宜 (すぐに反映されます) | 設定可能な値でない場合は、デフォルト値が設定されます。 |

# 7.9. Quorumパラメーター一覧

下記の表は、Quorumパラメーター名とその意味を説明しています。これらの値は */etc/default/LifeKeeper* 設定ファイルを編集することにより設定可能です。

| パラメーター名 | パラメーターの意味 | 設定値 | デフォルト値 | パラメーター適用タイミング | 備考 |
|---|---|---|---|---|---|
| QUORUM_MODE | Quorum モードを指定します。 | majority tcp_remote storage none または off | majority | 適宜 (すぐに反映されます) | 詳細はテクニカルドキュメンテーション「Quorum/Witness」を参照。 |
| QUORUM_HOSTS | Quorum チェックで接続確認を行うホストを “host:port” の形式で指定します。複数のホストを指定する場合、カンマ区切りで指定してください（スペースは含めないこと）。<br><br>（設定例）<br><br>QUORUM_HOSTS=myhost:80,router1:443,router2:22 | 文字列 | （未設定） | 適宜 (すぐに反映されます) | このパラメーターは、QUORUM_MODE が *tcp_remote* の場合のみ、適用されます。 |
| WITNESS_MODE | Witness モードを指定します。 | remote_verify storage none または off | remote_verify | 適宜 (すぐに反映されます) | 詳細はテクニカルドキュメンテーション「Quorum/Witness」を参照。 |
| QUORUM_TIMEOUT_SECS | Quorum チェックで tcp/ip 接続を完了するための タイムアウト時間を秒単位で指定します。この時間内に接続が完了しない場合は、失敗、または使用不可となります | 整数値 | 20 | 適宜 (すぐに反映されます) | このパラメーターは、QUORUM_MODE が *tcp_remote* の場合のみ、適用されます。 |
| QUORUM_LOSS_ACTION | Quorum が失われた場合の動作を指定します。 | fastkill fastboot osu | fastkill | 適宜 (すぐに反映されます) | 詳細はテクニカルドキュメンテーション「Quorum/Witness」を参照。 |

| | | | | | |
|---|---|---|---|---|---|
| QWK_STORAGE_TYPE | 共有ストレージのタイプを指定します。<br>storage モードで使用する場合は、必ず指定してください。 | block<br>file<br>aws_s3 | (未設定) | qwk_storage_init 実行時 | このパラメーターは、QUORUM_MODE および WITNESS_MODE に storage を指定した場合のみ適用されます。<br>詳細はテクニカルドキュメンテーション「storage モード」を参照。 |
| QWK_STORAGE_HBEATTIME | QWK オブジェクトを読み書きする間隔を秒単位で指定します。この設定はLCMHBEATTIME の設定より大きいまたは同じである必要があります。 | 5以上10以下の整数値 | 6 | qwk_storage_init 実行時 | このパラメーターは、QUORUM_MODE および WITNESS_MODE に storage を指定した場合のみ適用されます。特に QWK_STORAGE_TYPE=aws_s3 の場合は、この値を7 以上の値に増やすことを推奨します。<br>詳細はテクニカルドキュメンテーション「storage モード」を参照。 |
| QWK_STORAGE_NUMHBEATS | Witness チェックで対象ノードに障害が発生していると判断するための値を指定します。QWK オブジェクトの読み込みにおいて、このパラメーターに指定した回数以上更新が停止していると、対象ノードに障害が発生していると判断します。この設定はLCMNUMHBEATS の設定より大きいまたは同じである必要があります。 | 3以上の整数値 | 4 | qwk_storage_init 実行時 | このパラメーターは、QUORUM_MODE および WITNESS_MODE に storage を指定した場合のみ適用されます。<br>詳細はテクニカルドキュメンテーション「storage モード」を参照。 |
| QWK_STORAGE_OBJECT_<ホスト名><br>**注意:** ホスト名に”-”ま | QWK オブジェクトのパスを指定します。<br>*storage* モードで使用する場 | 文字列(最大長は256文 | (未設定) | qwk_storage_init 実行 | このパラメーターは、QUORUM_MODE および |

| | | | | | |
|---|---|---|---|---|---|
| たは”.”を含む場合、アンダースコア“_”に置き換えてください。（例：lksios-1 → lksios_1）<br>**注意:** LifeKeeperで使用するホスト名は lcd uname コマンドで調べることができます。 | 合は、必ずクラスターを構成するすべてのノード分のパラメーターを指定してください。<br><br>**[QWK_STORAGE_TYPEが block の場合]**<br>デバイスファイルのパスを指定してください。<br>**注意:** WWID (/dev/disk/by-id/) を使用して永続的なパスを指定してください。<br>（設定例）<br>QWK_STORAGE_OBJECT_nodeA=/dev/disk/by-id/xxxxx<br>QWK_STORAGE_OBJECT_nodeB=/dev/disk/by-id/yyyyy<br><br>**[QWK_STORAGE_TYPEが file の場合]**<br>レギュラーファイルのパスを指定してください。<br>（設定例）<br>QWK_STORAGE_OBJECT_nodeA=/quorum/nodeA<br>QWK_STORAGE_OBJECT_nodeB=/quorum/nodeB<br><br>**注意**: nodeA および nodeB は、ディレクトリーではなく通常のファイルである必要があります。<br><br>**[QWK_STORAGE_TYPEが aws_s3 の場合]**<br>Amazon S3 オブジェクトの s3uri を指定してください。LifeKeeper が動作しているリージョンと異なったリージョンの S3 を使用してください。１リージョンで十分ですが、２リージョン指定する場合は、カンマ区切り（スペースは含まない）で列挙してください。<br>（設定例1）<br>QWK_STORAGE_OBJECT_n | 字です） | | 時 | WITNESS_MODE に storage を指定した場合のみ適用されます。<br>詳細はテクニカルドキュメンテーション「storage モード」を参照。 |

| | | | | | |
|---|---|---|---|---|---|
| | odeA=s3://bucket1/nodeA,s3://bucket2/nodeA<br>QWK_STORAGE_OBJECT_nodeB=s3://bucket1/nodeB,s3://bucket2/nodeB<br>（設定例2）<br>QWK_STORAGE_OBJECT_nodeA=s3://bucket/quorum/nodeA<br>QWK_STORAGE_OBJECT_nodeB=s3://bucket/quorum/nodeB | | | | |
| HTTP_PROXY<br>HTTPS_PROXY<br>NO_PROXY | サービスエンドポイントへのアクセスでHTTPプロキシを使用する場合に設定してください。<br>ここで設定した値がそのままAWS CLI へ渡されます。<br>詳しくは AWS のドキュメント を参照してください。 | 文字列 | （未設定） | `qwk_storage_init` 実行時 | このパラメーターは EC2 や Route53 と共通のパラメーターです。 |

# 7.10. Route53 パラメーター一覧

下記の表に Route53 Recovery Kit の動作のチューニングパラメーターとその説明を示します。チューニングを行う場合には、 両ノードの */etc/default/LifeKeeper* 設定ファイルに追加することで設定されます。 Route53 Recovery kit のコンポーネントはメモリに常駐していないため、 */etc/default/LifeKeeper* の値を変更すると即時に反映されます。LifeKeeperやOSの再起動は必要ありません。

| パラメーター名 | 意味 | 設定値 | デフォルト値 | パラメーター適用タイミング | 注記 |
|---|---|---|---|---|---|
| ROUTE53_TTL | TTL (Time To Live) の設定値を秒単位で指定します。 | 整数値 | 10 | スイッチオーバー後 | |
| ROUTE53_CHANGEID_INTERVAL | Route 53 API通信のステータス確認時のインターバルを秒単位で指定します。 | 整数値 | 20 | 適宜(すぐに反映されます) | |
| ROUTE53_CHANGEID_TRY_COUNT | Route 53 API通信のステータス確認時のトライ回数を指定します。 | 整数値 | 5 | 適宜(すぐ | |

| | | | | | |
|---|---|---|---|---|---|
| | | | | に反映されます) | |
| ROUTE53_QUICKCHECK_TIMEOUT | quickCheck のタイムアウト値を秒単位で指定します。 | 整数値 | 25 | 適宜(すぐに反映されます) | クイックチェックのタイムアウトを防ぐには、ROUTE53_QUICKCHECK_TIMEOUT を調整します。<br><br>quickcheck タイマーをデフォルトの25からAWS CLIの実行時間より長いものにするのがよいでしょう。 75 (ROUTE53_QUICKCHECK_TIMEOUT=75) に設定することをお勧めします。 |
| AWSCLI_CONNECT_TIMEOUT | awsコマンドのオプション—cli–connect–timeout に与える値を秒単位で指定します。 | 整数値 | 10 | 適宜(すぐに反映されます) | 本パラメーターは EC2 と共通のパラメーターです。 |
| AWSCLI_READ_TIMEOUT | awsコマンドのオプション—cli–read–timeout に与える値を秒単位で指定します。 | 整数値 | 5 | 適宜(すぐに反映されます) | 本パラメーターは EC2 と共通のパラメーターです。 |
| HTTP_PROXY<br>HTTPS_PROXY<br>NO_PROXY | サービスエンドポイントへのアクセスで HTTP プロキシを使用する場合に設定してください。ここで設定した値がそのまま | 文字列 | (未設定) | 適宜(すぐに | 本パラメーターは EC2 や Quorum と共通のパラメーターです。 |

| | | | | | |
|---|---|---|---|---|---|
| | AWS CLI へ渡されます。詳しくはAWSのドキュメントを参照してください。<br>docs.aws.amazon.com/ja_jp/cli/latest/userguide/cli-http-proxy.html | | | 反映されます) | |

# 7.11. SAPパラメーター一覧

下記の表は、SAPパラメーター名とその意味を説明しています。これらの値は */etc/default/LifeKeeper* 設定ファイルを編集することにより設定可能です。

| パラメーター名 | パラメーターの意味 | 設定値 | デフォルト値 | パラメーター適用タイミング | 備考 |
|---|---|---|---|---|---|
| SAP_START_WAIT | リストアおよびリカバリープロセスによって発行された`sapcontrol -function StartWait`コマンドが保護されたSAPインスタンスを開始するまでのタイムアウト (秒単位) 。 | 整数値 | 228 | 適宜 (すぐに反映されます) | |
| SAP_STOP_WAIT | リムーブプロセスによって発行された`sapcontrol -function StopWait`コマンドが保護されたSAPインスタンスを停止するためのタイムアウト (秒単位) 。 | 整数値 | 228 | 適宜 (すぐに反映されます) | |
| SAP_EXPERTMODE | LifeKeeper SAP リソースの作成中に、SAP Recovery Kit はいくつかのチェックを実行して、どの SAP 関連ファイルシステムがノード間で共有されているかを判断し、対応する共有ファイルシステム リソースを自動的に作成しようとします。 これらのチェックを無効にするには (たとえば、複製されたファイルシステムを使用する際の柔軟性を高めるため)、SAP_EXPERTMODE=1 を設定します。 | 0: ファイルシステムチェックを実行し、SAP 関連の共有ファイルシステムのリソースを自動的に作成します。<br><br>1: ファイルシステムのチェックをスキップし、SAP 関連の共有ファイル システムのリソースを自動的に作成しません。 | 0 | SAP リソースの作成時に必要な場合 | SAP リソースの作成時に、SAP Recovery Kit が共有ファイルシステムの依存リソースを自動的に作成しないようにするには、このパラメータ値を 1 に設定します。 これは、例えば Data Keeper を使用して SAP 関連のファイルシステムをレプリケートする場合に便利です。 |
| SAP_DB_CHEC | 保護された SAP インスタンスを起動しようとする | 0:データベースチェックを | 0 | 適宜 (すぐに反映 | データベースが SIOS LifeKeeper |

| | | | | | |
|---|---|---|---|---|---|
| K_IGNORE | と、SAP Recovery Kit は、saphostctrl GetDatabaseStatus Call を使用して、関連付けられているデータベース (その情報は SAPDBHOST/dbtype/dbname/dbhost SAP プロファイル パラメータから取得されます) のステータスをチェックします。<br>このパラメータを 1 に設定すると、データベースステータス チェックが無効になります。 | 実行する<br><br>1: データベースチェックを無視する | | されます) | によって保護されていない場合、または SAP Recovery Kit によって実行されたデータベースチェックが予期せず失敗する場合は、このパラメータ値を 1 に設定します。 |
| SAP_CONFIG_REFRESH | Configuration Propertiesページの更新頻度を秒単位で指定します。 | 整数値 | LKCHECKINTERVAL/2 | 適宜 (すぐに反映されます) | 5より小さい場合は、デフォルト値が設定されます。 |
| SAP_CREATE_NAS | NAS マウントされたファイルシステムの NAS リソースを自動的に生成するかどうかを指定します。 | 0: 無効<br>1: 有効 | 1 | 適宜 (すぐに反映されます) | |
| SAP_NFS_CHECK_DIRS | チェックするNFSマウントポイントのリストをカンマ区切りで指定します。 | 文字列 | 空欄 | NFS共有ファイルシステムを使用する際に必要な時 | Amazon EFSマウントポイントには使用しないでください。 |
| SAP_QUICKCHECK_TIMEOUT | quickCheck のタイムアウト時間を秒単位で指定します。 | 整数値 | 60 | 適宜 (すぐに反映されます) | |
| SAP_RESTORE_TIMEOUT | restore のタイムアウト時間を秒単位で指定します。 | 整数値 | 516 | 適宜 (すぐに反映されます) | |
| SAP_REMOVE_TIMEOUT | remove のタイムアウト時間を秒単位で指定します。 | 整数値 | 804 | 適宜 (すぐに反映されます) | |
| SAP_RECOVER_TIMEOUT | recover のタイムアウト時間を秒単位で指定します。 | 整数値 | 1320 | 適宜 (すぐに反映されま | デフォルト値より小さい場合は、デフォルト |

| | | | | | |
|---|---|---|---|---|---|
| | | | | す) | 値が設定されます。 |

# 7.12. DataKeeperパラメーター一覧

下記の表は、DataKeeperパラメーター名とその意味を説明しています。これらの値は */etc/default/LifeKeeper* 設定ファイルを編集することにより設定可能です。

| パラメーター名 | パラメーターの意味 | 設定値 | デフォルト値 | パラメーター適用タイミング | 備考 |
|---|---|---|---|---|---|
| LKDR_CHUNK_SIZE | bitmapのチャンクサイズをKB単位で指定します。 | 整数値 | 256 | リソース作成時 | このパラメーターには、0からミラーサイズまでの任意の値を設定できます。値は2の累乗である必要があります。この値が2の累乗でない場合、あるいはミラーのサイズより大きい場合、ミラーの作成に失敗します。0に設定すると、デフォルトで256kになります。値が大きすぎると、1つのダーティブロックに対しても巨大なデータのチャンクが再同期されるため、ビットマップにはあまりメリットがありません。値が小さすぎる場合、ビットマップのオーバーヘッドが顕著になり、ミラーのスループットが制限されます。 |
| LKDR_CONNECT_NBD_DURING_RESTORE | ミラーの復元時（In | True または False | True | 複数の | デフォルト設定（「true」）では、ミラーが復元されると利用 |

| | | | | | |
|---|---|---|---|---|---|
| | | | | ミラーと複数のターゲットを持つクラスターでスイッチオーバーとフェールオーバーのパフォーマンスを | 可能なすべてのターゲットにデータがすぐに複製されるため、データの整合性が最も高くなります。In Serviceのパフォーマンス（「false」設定）を改善するために、最初のquickCheck 間隔（LKCHECKINTERVALで定義、デフォルトでは120秒）の間に NBD 接続を確立できます。これにより、ミラーを In Service にするのに必要な時間が短縮されますが、データがターゲットに複製されない時間も生じます。複数のミラーと複数のターゲットを使用している場合に、最もパフォーマンスが改善されます。 |
| | | | | | |
| | Service状態になった時）にNBD接続を確立する必要があるかどうかを指定します。 | | | | |

| | | | | | |
|---|---|---|---|---|---|
| | | | | 向上させたい時 | |
| LKDR_SPEED_LIMIT | 同期に使用する最大帯域幅を指定します。可能な最大速度で再同期が実行されるように、この値を十分高い値に設定する必要があります。 | 整数値 | 500000 | リソース再同期 | |
| LKDR_SPEED_LIMIT_MIN | 同時に他の I/O が実行されているときに許可する再同期の速度を指定します。同期の実行時に通常の I/O 動作が妨げられ | 整数値 | 20000 | リソース再同期 | このパラメーターは、LKDR_SPEED_LIMIT の設定よりも小さくする必要があります。 |

| | | | | | |
|---|---|---|---|---|---|
| | ないようにするには、経験則として、この値をドライブの最大書き込みスループットの半分以下に設定する必要があります。 | | | | |
| LKDR_ASYNC_LIMIT | ターゲットへの同期を遅らせることのできる書き込み要求の数を指定します。非同期ミラーを使用してパフォーマンスを向上させるには、この値を増やします。 | 整数値 | 4096 | リソース作成時 | このパラメーターの有効範囲は、0～16383です。16383を超える値に設定すると、ミラーの作成に失敗します。0に設定された場合、非同期制限はデフォルトで4096になります。1に設定された場合、デフォルトで256になります。2～16383の間の値を指定すると、非同期制限はその値に設定されます。注：4k以下の値は推奨されません。 |
| LKDR_NO_FULL_SYNC | 新しく追加されたタ | 0: 抑制しない | 0 | 適宜(す | 詳細はテクニカルドキュメンテーション「全同期の回避」を参照。 |

| | | | | | |
|---|---|---|---|---|---|
| | ーゲットの全同期を抑制します。 | 1: 抑制する | | ぐに反映されます) | |
| LKDR_WAIT_TO_RESYNC | ミラーの再同期前に、どの親リソースを In Service にする必要があるかを指定します。 | False, hierarchy, <resource type> | filesys | 複製の再開前に複製されたデーターの整合性を検証する時 | False は、9.5.1 以前のバージョンでの通常の動作となります。“Hierarchy” は全階層が In Service になるまで待ちます。‘filesys’ のような “<resource type>” は、そのタイプの全ての親リソースがIn Service になるまで待ちます。 |
| LKDR_WAIT_FOR_PREVIOUS_SOURCE_TIMEOUT | ビットマップをマージできるように、前のソースがクラスターに参加するのを待 | -1： 前のソースがクラスターに再参加するまで待機する<br>0： 待機しない<br>> 0： 待機する時 | -1 | 適宜(すぐに反映されます) | 詳細はテクニカルドキュメンテーション「SIOS DataKeeper の仕組み 」を参照。 |

| | | | | | |
|---|---|---|---|---|---|
| | つ時間を指定します。ターゲットをクラスターに追加する際の完全再同期を回避するには、前のソースのビットマップを最初にマージする必要があります。この設定はすべてのnetraidデバイスに適用され、個々のデバイスに異なる値を設定することはできません。 | 間（秒） | | | |
| NBD_NR_REQUESTS | 所定の時間処理を保留できるI/O要 | 整数値 | 2048 | 適宜(すぐに | |

| | | | | | |
|---|---|---|---|---|---|
| | 求の数を指定します。ミラーリングのパフォーマンスを高めるには、この値を増やします。 | | | 反映されます) | |

# 7.13. Standby Node Health Check パラメータ一覧

Standby Node Health Check では、各機能の有効/無効や値を設定する必要があります。
これらの設定は、/etc/default/LifeKeeper 設定ファイルでカスタマイズすることが可能です。

| パラメータ名 | 説明 | 設定値 | デフォルト値 | 適応タイミング | 備考 |
|---|---|---|---|---|---|
| SNHC | Standby Node Health Check の全体の機能を有効にします。有効にする場合は 1 に設定してください。デフォルトでは無効となっています。さらに個々の機能の SNHC_XX を有効にする必要があります。 | 0 : 無効<br>1 : 有効 | 0 | lkcheck 起動時 | 設定後は lkcheck の再起動をしてください。詳細は「Standby Node Health Check 」を参照してください。 |
| SNHC_CPUCHECK | Node 監視 機能における、CPU 監視を有効にします。有効にする場合は 1 に設定してください。デフォルトでは無効となっています。 | 0 : 無効<br>1 : 有効 | 0 | 適宜 | |
| SNHC_CPUCHECK_THRESHOLD | CPU 監視において、異常と判断するCPU使用率の閾値を指定します。指定されていない場合、99 で実行されます。 | 10 以上 99 以下の整数値 | 99 | 適宜 | 設定がされていない場合 ERROR ログにて設定を促します。 |
| SNHC_CPUCHECK_TIME | CPU 監視において SNHC_CPUCHECK_THRESHOLD で設定した使用率を連続で超えた場合に異常と判断する回数を指定します。設定されていない場合、1 で実行されます。 | 1 以上 100 以下の整数値 | 1 | 適宜 | 設定がされていない場合 ERROR ログにて設定を促します。 |

| | | | | | |
|---|---|---|---|---|---|
| SNHC_MEMCHECK | Node監視 機能における、Memory 監視を有効にします。有効にする場合は 1 に設定してください。デフォルトでは無効となっています。 | 0：無効<br>1：有効 | 0 | 適宜 | |
| SNHC_MEMCHECK_THRESHOLD | Memory 監視において、異常と判断する Memory 使用率の閾値を指定します。設定されていない場合、99 で実行されます。 | 10 以上 99 以下の整数値 | 99 | 適宜 | 設定がされていない場合 ERROR ログにて設定を促します。 |
| SNHC_MEMCHECK_TIME | Memory 監視において SNHC_MEMCHECK_THRESHOLD で設定した使用率を連続で超えた場合に異常と判断する回数を指定します。設定されていない場合、1 で実行されます。 | 1 以上 100 以下の整数値 | 1 | 適宜 | 設定がされていない場合 ERROR ログにて設定を促します。 |
| SNHC_IPCHECK | IP リソースの OSU リソース監視 を有効にします。デフォルトでは無効となっています。 | 0：無効<br>1：有効 | 0 | 適宜 | |
| SNHC_IPCHECK_SLEEPTIME | IP リソースの OSU リソース監視で対象となる NIC が down している場合、NIC を起動してからリンクの確認を実行するまでの待機時間を設定します。NIC が down していた場合、監視処理後に再度 NIC を down します。 | 待機時間（秒） | 3 | 適宜 | |
| SNHC_DISKCHECK | Disk または DMMP リソースの OSU リソース監視 を有効にします。デフォルトでは無効となっています。 | 0：無効<br>1：有効 | 0 | 適宜 | |
| SNHC_DISKCHECK_<Tag名> | 指定した Tag 名の Disk または DMMP リソースのみの OSU リソース監視 を有効にします。デフォルトでは無効となっています。 | 0：無効<br>1：有効 | 0 | 適宜 | |
| SNHC_NASCHECK | NAS リソースの OSU リソース監視 を有効にします。 | 0：無効<br>1：有効 | 0 | 適宜 | |

| | | | | | |
|---|---|---|---|---|---|
| SNHC_NASCHECK_<Tag名> | 指定された Tag 名の NAS リソースのみ OSU リソース監視を有効にします。 | 0 : 無効<br>1 : 有効 | 0 | 適宜 | Tag 名にハイフンが入っている場合はアンダーバーに置き換えて設定してください。 |

# 7.14. SAP HANA パラメーターリスト

次の表には、SAP HANA パラメーターの名称と意味を記載しています。これらの値は、 /etc/default/LifeKeeper 構成ファイルを編集することによってチューニングできます。

| パラメーター名 | 意味 | 設定値 | デフォルト値 | パラメーター適用タイミング | 注記 |
|---|---|---|---|---|---|
| HANA_HANDSHAKE_TAKEOVER_FAILBACK | Takeover with Handshakeに失敗した時に SAP HANA リソースの自動フェイルバックを実行するかどうかを指定します。‘true’が設定されるとLifeKeeper は Takeover with Handshake に失敗した場合に旧ホストで SAP HANA リソースを In Service にもどします。‘false’ (デフォルト)が設定されると、Takeover with Handshake が失敗した後Out of Service Failed (OSF) のステータスのままになり、手動でIn Service にする必要があります。 | true もしくは false | false | 適宜 (すぐに反映されます) | |
| HANA_HSR_POLL_INTERVAL | LifeKeeper が SAP HANA システムレプリケーションのステータスをポーリングする頻度（秒）を指定します。 | 正の整数 | 10 | 適宜 (すぐに反映されます) | |
| HANA_QUICKCHECK_TIMEOUT | SAP HANA quickCheck スクリプトのタイムアウト値（秒）を指定します。 | 正の整数、最低値 30 | LKCHECKINTERVAL – 10 | 適宜 | 30 未満に設定すると、自動的にデフォルトで 30 になります。 |
| HANA_REGISTER_SECONDARY_DURING_RESTORE | プライマリーデータベースインスタンスに対する In Service 処理が完了する前に SAP HANA Recovery Kit がセカンダリーデータベースを登録し、再起動するかどうかを指定します。 ‘true’が設定されるとSAP HANA リソースは、SAP HANA Recovery Kit がセカンダリーデータベースを登録し、再起動するまで In Service になりません。 ‘false’ (デフォルト)が設定されるとSAP HANA リソースは、プライマリーデータベースイン | true または false | false | 適宜 (すぐに反映されます) | |

| | | | | | |
|---|---|---|---|---|---|
| | スタンスが起動するとすぐに In Service になり、セカンダリーデータベースの登録、再起動のため、ローカルリカバリーイベントがその後すぐに開始されます。 | | | | |
| HANA_RECOVER_TIMEOUT | SAP HANA リカバリースクリプトのタイムアウト値（秒）を指定します。 | 正の整数 | 1800 | 適宜 (すぐに反映されます) | |
| HANA_START_WAIT | SAP HANA データベースインスタンスを開始するときに待機する時間（秒）を指定します。 | 正の整数 | 2700 | 適宜 (すぐに反映されます) | |
| HANA_STOP_WAIT | SAP HANA データベースインスタンスを停止するときに待機する時間（秒）を指定します。 | 正の整数 | 600 | 適宜 (すぐに反映されます) | |
| HANA_STOP_COUNT | SAP HANA データベースインスタンスを停止するときに sapcontrol StopWait コマンドを発行する試行回数を指定します。 | 正の整数 | 3 | 適宜 (すぐに反映されます) | |
| HANA_STOP_FINAL_WAIT | HANA_STOP_COUNT が SAP HANA データベースの停止に失敗した後、データベースが完全に停止するまで待機する追加の時間（秒）を指定します。 | 正の整数 | 60 | 適宜 (すぐに反映されます) | |
| HANA_RECOVER_AUTORES_ISP | LifeKeeper が起動すると、AUTORES_ISP が設定されたリソースを持つすべての階層が自動的に in-service になります。 他のシステムがダウンしている場合、HANA は起動時にリソースを自動的に in-service にすることはできません。 このフラグを true に設定すると、クラスター内の最後のシステムが起動したときに、階層が AUTORES_ISP システムで in-service になります。 例えば、2 ノード クラスターでは、node1 が起動すると、node2 がダウンしている場合に AUTORES_ISP が設定されていても、HANA リソースは自動的に in-service になりません。 node2 が起動すると、node1 の自動 restore が開始されます。 | true または false | true | 適宜 (すぐに反映されます) | |
| HANA_STARTUP_TIMEOUT | restore がシステム上の LifeKeeper の起動完了を待つ最大時間（秒）で | 正の整数 | 900 | 適宜 (すぐに反映さ | |

| | | | | | |
|---|---|---|---|---|---|
| | す。 | | | れます) | |

# 7.15. SAP MaxDB パラメーター一覧

以下の表は、SAP MaxDB リカバリーキットのパラメーターの名前と意味を説明しています。これらの値は、 */etc/default/LifeKeeper* 設定ファイルを編集することにより設定可能です。

| パラメーター名 | パラメーターの意味 | 設定値 | デフォルト値 | パラメーター適用タイミング | 備考 |
|---|---|---|---|---|---|
| MAXDB_START_TIMEOUT | ハングしているIn Serviceのアクションを中止するまで待機する秒数 | 整数 | 300 | 適宜 (すぐに反映されます) | |
| MAXDB_PID_CLEANUP | ハングしたIn Serviceのアクションの子プロセスをクリーンアップ（強制終了）するかどうかを指定します | y：クリーンアップする<br>n：クリーンアップしない | n | 適宜 (すぐに反映されます) | |
| MAXDB_STOP_COUNT | LifeKeeperがデータベースのオフラインを試行する回数 | 整数 | 5 | 適宜 (すぐに反映されます) | |
| MAXDB_WAIT | データベースのオフライン試行間の待機秒数 | 整数 | 5 | 適宜 (すぐに反映されます) | |
| MAXDB_DEBUG | デバッグロギング | 0: | 0 | 適宜 (すぐに反映されます) | |

| | | | | | |
|---|---|---|---|---|---|
| | を有効または無効にします | 無効化 1: 有効化 | | | |
| MAXDB_OFFLINE_ENABLED | LifeKeeper MaxDB リソースをOut of Serviceにしたときにデータベースをオフラインにするかどうかを指定します | 0: オフラインにしない 1: オフラインにする | 1 | データベースを停止せずにLifeKeeperを一時的に停止する必要がある場合（例：アップグレードやメンテナンスの実施など）に、注意して使用してください | メンテナンスが完了したら、データベースをオフラインに戻してください |

# 8. エラーコード一覧

総合メッセージカタログ をご参照ください。

# 8.1. 総合メッセージカタログ

**Control F** を使用して、各カタログの特定のエラーコードを検索します。エラーコードの検索には、画面右上の検索ボックスをご利用ください。

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 000200 | ERROR | pam_start() failed | |
| 000201 | ERROR | pam_authenticate failed (user %s, retval %d) | |
| 000202 | ERROR | pam_end() failed?!?! | |
| 000203 | ERROR | Did not find expected group ‘lkguest’ | |
| 000204 | ERROR | Did not find expected group ‘lkoper’ | |
| 000205 | ERROR | Did not find expected group ‘lkadmin’ | |
| 000208 | ERROR | pam_setcred establish credentials failed (user %s, retval %d) | **原因**: ユーザー {user} の有効なログイン資格情報を確立できません。pam_setcred 呼び出しで返された値は次の通りです：{retval}<br><br>**処置**: 詳細については、/var/log/security および /var/log/messages を確認して下さい。 |
| 000209 | ERROR | pam_setcred delete credentials failed (user %s, retval %d) | **原因**: ユーザー {user} の有効なログイン資格情報を確立できません。pam_setcred 呼び出しで返された値は次の通りです：{retval}<br><br>**処置**: 詳細については、/var/log/security および /var/log/messages を確認して下さい。 |
| 000902 | ERROR | Error removing system name from loopback address line in /etc/hosts file. You must do this manually before starting the GUI server. | **原因**: システム名が /etc/hosts ファイルから削除されていません。<br><br>**処置**: システム名を手動で削除し、GUI サーバーを再起動して、次のコマンドを入力して下さい。: run <action name> |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 000918 | ERROR | LifeKeeper GUI Server error during Startup | 原因: 異常な状態により、GUI サーバーが終了しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 001052 | FATAL | Template resource "%s" on server "%s" does not exist | 原因: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。 |
| 001053 | ERROR | Cannot access canextend script "%s" on server "%s" | 原因: {server} でスクリプト CANEXTEND が見つからなかったので、LifeKeeper は拡張前のチェックを実行できませんでした。<br><br>処置: LifeKeeper の設定を確認してください。 |
| 001054 | ERROR | Cannot extend resource "%s" to server "%s" | 原因: LifeKeeper は、{server} でリソース {resource} を拡張できませんでした。 |
| 001055 | ERROR | Cannot access extend script "%s" on server "%s" | 原因: LifeKeeper は、{server} でスクリプト EXTEND が見つからなかったので、リソース階層を拡張できませんでした。<br><br>処置: LifeKeeper の設定を確認してください。 |
| 001057 | ERROR | Cannot extend resource "%s" to server "%s" | 原因: LifeKeeper は、{server} でリソース {resource} を拡張できませんでした。 |
| 001059 | ERROR | Resource with tag "%s" already exists | 原因: リソースで指定した名前はすでに使用中です。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: リソースに別の名前を選択するか、既存のリソースを使用してください。 |
| 001060 | ERROR | Resource with either matching tag "%s" or id "%s" already exists on server "%s" for App "%s" and Type "%s" | **原因**: リソースで指定した名前または ID はすでに使用中です。<br><br>**処置**: リソースに別の名前または ID を選択するか、既存のリソースを使用してください。 |
| 001061 | ERROR | Error creating resource "%s" on server "%s" | **原因**: リソースの作成中に予期しない障害が発生しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 001081 | WARN | IP address \"$ip\" is neither v4 nor v6 | **原因**: 指定した IP アドレスは IPv4 アドレスでも IPv6 アドレスでもありません。<br><br>**処置**: 指定した名前またはアドレスを確認し、再試行してください。名前を指定している場合は、名前の解決が有効な IP アドレスを返していることを確認してください。 |
| 004024 | ERROR | | **原因**: LCD はリソースのリカバリー中にリソース ID {id} のリソース情報のフェッチに失敗しました。<br><br>**処置**: 入力リソース ID を確認し、リカバリー操作を再試行してください。 |
| 004028 | ERROR | %s occurred to resource \"%s\" | **原因**: リソース {resource} のローカルリカバリーが失敗しました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004055 | ERROR | attempt to remote-remove resource \"%s\" that can’t be found | **原因**: タグ名 {tag} によるリソースの検索中に、リモートによるサービスからのリソース削除に失敗しました。<br><br>**処置**: 入力タグ名を確認し、リカバリー操作を再試行してください。 |
| 004056 | ERROR | attempt to remote-remove resource \"%s\" that is not a shared resource | **原因**: タグ名 {tag} が共有リソースでない場合に、リモートによるサービスからのリソース削除に失敗しました。<br><br>**処置**: 入力タグ名を確認し、リカバリー操作を再試行してください。 |
| 004060 | ERROR | attempt to transfer-restore resource \"%s\" that can’t be found | **原因**: タグ名 {tag} を指定したサービスリソースのリモート転送に失敗しました。<br><br>**処置**: 入力タグ名を確認し、リカバリー操作を再試行してください。 |
| 004061 | ERROR | attempt to transfer-restore resource \"%s\" that is not a shared resource with machine \"%s\" | **原因**: LifeKeeper は、リモート {machine} からサービスのリソースをリモート転送する際に、{tag} 名による共有リソースの検出に失敗しました。<br><br>**処置**: 入力タグ名を確認し、リカバリー操作を再試行してください。 |
| 004089 | ERROR | ERROR: Parallel recovery initialization failed.\n | **原因**: パラレルリカバリーは、階層内にあるリソースのリストの初期化に失敗しました。<br><br>**処置**: 関連するエラーをログで確認 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | し、レポートされている問題を解決してください。 |
| 004091 | ERROR | ERROR: fork failed. continuing to next resource\n | **原因**: パラレルリカバリーで、単一のリソースをリストアしようとする新しいプロセスを分岐できませんでした。 |
| 004093 | ERROR | ERROR: reserve failed. continuing to next resource\n | **原因**:パラレルリカバリーは、集合階層から単一のリソースのリザーブに失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004096 | ERROR | ERROR: clone %d is hung, attempting to kill it\n | **原因**: リソースリカバリーの単一のサブプロセスが、リソース階層全体のパラレルリカバリーの際にハングしました。<br><br>**処置**: ハングしているサブプロセスが自動的に停止してください |
| 004097 | ERROR | ERROR: Could not kill clone %d\n | **原因**: ハングしているサブプロセスの停止に失敗しました。 |
| 004116 | ERROR | %s | **原因**: メモリ内データオブジェクトのオンディスクバージョンを書き込む際に、中間フォルダを作成しようとして失敗しました。これはシステムエラーです。<br><br>**処置**: ログで詳細なエラー情報を確認し、中間フォルダが作成されていない理由を判別してください。 |
| 004117 | ERROR | open(%s) | **原因**: メモリ内データオブジェクトのオ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ンディスクバージョンを書き込む際に、一時ファイルを開こうとして失敗しました。これはシステムエラーです。<br><br>**処置**: ログで詳細なエラー情報を確認し、ファイルを開くことができない理由を判別してください。 |
| 004118 | ERROR | write(%s) | **原因**: メモリ内データオブジェクトのオンディスクバージョンを書き込む際に、一時ファイルに書き込もうして失敗しました。これはシステムエラーです。<br><br>**処置**: ログで詳細なエラー情報を確認し、ファイルの書き込みに失敗した理由を判別してください。 |
| 004119 | ERROR | fsync(%s) | **原因**: メモリ内データオブジェクトのオンディスクバージョンを書き込む際に、一時ファイルに fsync を実行しようとして失敗しました。これはシステムエラーです。<br><br>**処置**: ログで詳細なエラー情報を確認し、「fsync」が失敗した理由を特定してください。 |
| 004120 | ERROR | close(%s) | **原因**: メモリ内データオブジェクトのオンディスクバージョンを書き込む際に、一時ファイルを閉じようとして失敗しました。これはシステムエラーです。<br><br>**処置**: ログで詳細なエラー情報を確認し、ファイルのクローズに失敗した理由を判別してください。 |
| 004121 | ERROR | rename(%s, %s) | **原因**: メモリ内データオブジェクトのオンディスクバージョンを書き込む際に、一時ファイル {file} の名前を元のフ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ファイル {file} に変更しようとして失敗しました。これはシステムエラーです。<br><br>**処置**: ログで詳細なエラー情報を確認し、ファイルの名前変更に失敗した理由を判別してください。 |
| 004122 | ERROR | open(%s) | **原因**: メモリ内データオブジェクトのオンディスクバージョンを書き込む際に、中間ディレクトリ {directory} を開こうとして失敗しました。これはシステムエラーです。<br><br>**処置**: ログで詳細なエラー情報を確認し、ディレクトリのオープンに失敗した理由を判別してください。 |
| 004123 | ERROR | fsync(%s) | **原因**: メモリ内データオブジェクトのオンディスクバージョンを書き込む際に、中間ディレクトリ {directory} に fsync を実行しようとして失敗しました。これはシステムエラーです。<br><br>**処置**: ログで詳細なエラー情報を確認し、ディレクトリの「fsync」に失敗した理由を特定してください。 |
| 004124 | ERROR | close(%s) | **原因**: メモリ内データオブジェクトのオンディスクバージョンを書き込む際に、中間ディレクトリ {directory} を閉じようとして失敗しました。これはシステムエラーです。<br><br>**処置**:ログで詳細なエラー情報を確認し、ディレクトリのクローズに失敗した理由を判別してください。 |
| 004125 | ERROR | wrote only %d bytes of requested %d\n | **原因**: メモリ内データオブジェクトのオンディスクバージョンを書き込む際 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | に、書き込まれるデータの最終的なサイズ {size} バイトが要求バイト数 {number} を下回っているため、書き込み操作が失敗しました。<br><br>**処置**: ログで関連エラー情報を確認し、データの書き込みに失敗した理由を判別してください。 |
| 004126 | ERROR | open(%s) | **原因**: データオブジェクトのオンディスクバージョンをバッファに読み取っている間に、データファイルを開こうとして失敗しました。これはシステムエラーです。<br><br>**処置**: ログで詳細なエラー情報を確認し、ファイルのオープンに失敗した理由を判別してください。 |
| 004127 | ERROR | open(%s) | **原因**: データオブジェクトのオンディスクバージョンをバッファに読み取っている間に、一時データファイルを開こうとして失敗しました。これはシステムエラーです。<br><br>**処置**: ログで詳細なエラー情報を確認し、ファイルのオープンに失敗した理由を判別してください。 |
| 004128 | ERROR | read(%s) | **原因**: データオブジェクトのオンディスクバージョンをバッファにロードしている間に、データファイルを読み取ろうとして失敗しました。これはシステムエラーです。<br><br>**処置**:ログで詳細なエラー情報を確認し、ファイルの読み取りに失敗した理由を判別してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 004129 | ERROR | read buffer overflow (MAX=%d)\n | **原因**: データオブジェクトのオンディスクバージョンをバッファに読み取ろうとしている間に、読み取りバッファの制限 {max} に達しました。<br><br>**処置**: LifeKeeper 設定を確認し、LifeKeeper を再起動してください。 |
| 004130 | ERROR | close(%s) | **原因**: データオブジェクトのオンディスクバージョンをバッファに読み取っている間に、データファイルのクローズに失敗しました。これはシステムエラーです。<br><br>**処置**: ログで詳細なエラー情報を確認し、ファイルのクローズに失敗した理由を判別してください。 |
| 004131 | ERROR | rename(%s, %s) | **原因**: データオブジェクトのオンディスクバージョンをバッファに読み取っている間に、一時データファイルの名前変更に失敗しました。これはシステムエラーです。<br><br>**処置**: ログで詳細なエラー情報を確認し、ファイルの名前変更に失敗した理由を判別してください。 |
| 004132 | ERROR | Can’t open %s : %s | **原因**: アプリケーションとリソースタイプ情報のオンディスクバージョンをバッファに読み取っている間にエラー {error} でディレクトリ {directory} のオープンに失敗しました。これはシステムエラーです。<br><br>**処置**: ログで詳細なエラー情報を確認し、ディレクトリのオープンに失敗した理由を判別してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 004133 | ERROR | path argument may not be NULL | **原因**: 入力ソースパスが見つからないので、ファイルコピー中にコマンド「lcdrcp」が失敗しました。<br><br>**処置**: 入力ソースパスを確認し、「lcdrcp」を再試行してください。 |
| 004134 | ERROR | destination path argument may not be NULL | **原因**:入力ディスティネーションパスが見つからないので、ファイルコピー中にコマンド「lcdrcp」が失敗しました。<br><br>**処置**: 入力ディスティネーションパスを確認し、「lcdrcp」を再試行してください。 |
| 004135 | ERROR | destination path can’t be zero length string | **原因**:「lcdrcp」によるファイルコピー中に、入力ディスティネーションパスが空でした。<br><br>**処置**: 入力ディスティネーションパスを確認し、「lcdrcp」を再試行してください。 |
| 004136 | ERROR | open(%s) | **原因**:「lcdrcp」によるファイルコピー中に、ソースファイルパスのオープンに失敗しました。これはシステムエラーです。<br><br>**処置**: 入力ソースパスがあるかどうか、および使用可能かどうかを確認し、「lcdrcp」を再試行します。また、関連ログで詳細なエラー情報を確認してください。 |
| 004137 | ERROR | fstat(%s) | **原因**:「lcdrcp」によるファイルコピー中に、「fstat」を使用したファイル属性のフェッチに失敗しました。これはシステムエラーです。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**:ログで詳細なエラー情報を確認してください。 |
| 004138 | ERROR | file \"%s\" is not an ordinary file (mode=0%o) | **原因**:「lcdrcp」によるファイルコピー中に、通常のものではないファイルとしてソースファイルが検出されました。<br><br>**処置**: 入力ソースファイルパスを確認し、「lcdrcp」を再試行してください。 |
| 004151 | FATAL | lcdMalloc failure | **原因**: 共有メモリに、要求されたサイズのメモリを割り当てることができませんでした。<br><br>**処置**: 致命的なエラーが生成されます。 |
| 004152 | ERROR | having \"%s\" depend on \"%s\" would produce a loop | **原因**: 要求された依存関係を追加すると、依存関係がループします。<br><br>**処置**: 要求された依存関係を修正し、依存関係の作成を再試行してください。 |
| 004164 | ERROR | Priority mismatch between resources %s and %s. Dependency creation failed. | **原因**: {resource1} と {resource2} のプロパティが一致しません。<br><br>**処置**: リソースプロパティは一致している必要があります。どちらか一方または両方のプロパティを同じ値に変更し、依存関係の作成を再試行してください。 |
| 004176 | ERROR | %s | **原因**: コマンド「doabort」がコアファイルを書き込むための {directory} の作成に失敗しました。これはシステムエラーです。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 004182 | ERROR | received signal %d\n | 原因: 信号 {signum} を受信しました。 |
| 004186 | ERROR | %s: ::receive(%d) protocol error on incoming_mailbox %s | 原因: {function} で、タイムアウト時間（{timeout}秒）以内にメッセージを受信しようとしましたが、受信メールボックス {mailbox} が非アイドル状態のため受信できませんでした。<br><br>処置: クラスター内の接続の状態を確認し、プロセスを再試行してください。 |
| 004190 | ERROR | %s: ::receive(%d) did not receive message within %d seconds on incoming_mailbox %s | 原因: 関数 {function} で、受信メールボックス {mailbox} でタイムアウト {timeout} 秒以内にメッセージを受信しようとしたところ、失敗しました。<br><br>処置: クラスター内の接続の状態を確認し、プロセスを再試行してください。 |
| 004204 | ERROR | attempt to send illegal message | 原因: 不正なメッセージを送信しようとしたため、メッセージの送信に失敗しました。 |
| 004205 | ERROR | destination system \"%s\" is unknown | 原因: 不明なディスティネーションシステム名 {system} のため、メッセージの送信に失敗しました。<br><br>処置: システムの設定と状態を確認し、関連エラーをログで確認してください。システムが完全に初期化されたら、同じプロセスを再試行してください。 |
| 004206 | ERROR | destination mailbox \"%s\" at system \"%s\" is unknown | 原因: 不明なディスティネーションシステム名 {system} の不明なメールボックス {mailbox} のため、メッセージの送信に失敗しました。このエラーは、LCD が完全に初期化される前にメッセージを |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 送信することで発生します。<br><br>**処置**:システムの設定と状態を確認し、関連エラーをログで確認してください。システムが完全に初期化されたら、同じプロセスを再試行してください。 |
| 004208 | ERROR | destination system \"%s\" is alive but the \"%s\" mailbox process is not listening. | **原因**: メッセージの送信に失敗しました。ディスティネーションシステム {system} へのネットワーク接続は稼働中ですが、ディスティネーションメールボックスとの通信は失われました。<br><br>**処置**: システムの設定と状態を確認し、関連エラーをログで確認してください。システムが完全に初期化されたら、同じプロセスを再試行してください。 |
| 004209 | ERROR | destination system \"%s\" is dead. | **原因**: ディスティネーションシステム {system} との接続が失われたため、メッセージの送信に失敗しました。<br><br>**処置**: システムの設定と状態を確認し、関連エラーをログで確認してください。システムが完全に初期化されたら、同じプロセスを再試行してください。 |
| 004211 | ERROR | can’t send to destination \"%s\" error=%d | **原因**: 内部エラー {error} のため、ディスティネーションシステム {system} へのメッセージ送信に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004217 | ERROR | destination system \"%s\" is out of service. | **原因**: ディスティネーションシステム {sy |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | stem} との接続が失われたため、メッセージの送信に失敗しました。<br><br>**処置**: システムの設定と状態を確認し、関連エラーをログで確認してください。システムが完全に初期化されたら、同じプロセスを再試行してください。 |
| 004221 | ERROR | destination system \"%s\" went out of service. | **原因**: ディスティネーションシステム {system} との接続が失われたため、メッセージの送信に失敗しました。<br><br>**処置**: システムの設定と状態を確認し、関連エラーをログで確認してください。システムが完全に初期化されたら、同じプロセスを再試行してください。 |
| 004228 | ERROR | Can’t get host name from getaddrinfo() | **原因**:「getaddrinfo()」を使用してホスト名を取得しているときに障害が発生したため、ネットワークオブジェクトの作成に失敗しました。<br><br>**処置**: システムの設定と状態を確認し、同じプロセスを再実行してください。 |
| 004234 | ERROR | IP address pair %s already in use | **原因**: TCP コミュニケーションパスで IP アドレスペア {pair} がすでに使用中のため、ネットワークオブジェクトの作成に失敗しました。<br><br>**処置**: 入力 IP アドレスペアを確認し、ネットワークの作成を再実行してください。 |
| 004258 | WARN | Communication to %s by %s FAILED | **原因**: コミュニケーションパス {path} によるシステム {system} への通信に失敗 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | しました。<br><br>**処置**: システム設定とネットワーク接続を確認してください。 |
| 004261 | WARN | COMMUNICATIONS failover from system \"%s\" will be started. | **原因**: すべてのコミュニケーションパスがダウンしたため、システム {system} からのフェイルオーバーが開始されます。<br><br>**処置**: システム設定とネットワーク接続の状態を確認してください。フェイルオーバーが完了したら、システムの状態を確認してください。 |
| 004292 | ERROR | resource \"%s\" %s | **原因**: 現在の状態が不明なため、リソースを in service にできませんでした。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004293 | ERROR | resource \"%s\" %s | **原因**: 現在の状態により許可されていないため、リソースを in service にできませんでした。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004294 | ERROR | resource \"%s\" requires a license (for Kit %s/%s) but none is installed | **原因**: リソースの関連リカバリーキットにはライセンスが必要です。<br><br>**処置**: リソースを in service にするサーバーに、リカバリーキットのライセンスをインストールしてください。 |
| 004297 | ERROR | secondary remote resource \"%s\" on machine \"%s\" is already in-service, so reso | **原因**:セカンダリリモートリソース {reso |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | urce \"%s\" on machine \"%s\" can’t be br ought in-service. | urce} がマシン {machine} ですでに in se rvice になっているため、リソース {reso urce} をマシン {machine} で in service にできませんでした。<br><br>**処置**: リモートリソースを手動で out of s ervice に変更し、ローカルリソースで再び in service にしてください |
| 004298 | ERROR | remote resource \"%s\" on machine \"%s\" is still in-service, restore of resource \"%s\" will not be attempted!\n | |
| 004300 | ERROR | restore of resource \"%s\" has failed | **原因**: リソースを in service にできませんでした。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004311 | ERROR | can’t perform \"remove\" action on resour ces in state \"%s\" | **原因**: 現在の状態が {state} のため、リソースを out of service にできませんでした。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004313 | ERROR | remove of resource \"%s\" has failed | **原因**: リソース {resource} をサービス休止にできませんでした。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004318 | ERROR | %s,priv_globact(%d,%s): script %s FAILE D returning %d | **原因**: 指定したエラーコードでグローバルアクションスクリプトが失敗しました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004332 | ERROR | action \"%s\" has failed on resource \"%s\" | **原因**: リソースアクションが失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004351 | ERROR | a \"%s\" equivalency must have one remote resource | **原因**:同じシステムに 2 つの入力タグ名が存在しているため、イクイバレンシ {eqvtype} の作成に失敗しました。<br><br>**処置**: 入力リソースタグ名を修正し、同じプロセスを再実行してください。 |
| 004356 | WARN | Use unsupported option %s for remove. This option may be removed in future upgrades. | |
| 004376 | FATAL | wait period of %u seconds for LCM to become available has been exceeded (lock file \"%s\" not removed) | **原因**: 妥当な時間内に LCM デーモンが使用可能になりませんでした。LCD は LCM なしでは操作できません。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004386 | ERROR | initlcdMalloc;shmget | **原因**: 共有メモリセグメントを初期化できませんでした。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。製品ドキュメントを参照し、サーバーが最低要件を満たしていること、およびオペレーティングシステムが適切に設定されていることを確認して |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ください。 |
| 004439 | WARN | intermachine recovery skipped for %s. Failed to obtain resource_state_change lock.\n | |
| 004444 | WARN | License key (for Kit %s/%s) has EXPIRED | **原因**: ライセンスが期限切れです。<br><br>**処置**: サポートに問い合わせ、新しいライセンスを入手してください。 |
| 004445 | WARN | License key (for Kit %s/%s) will expire at midnight in %ld days | **原因**: ライセンスが間もなく期限切れです。<br><br>**処置**: サポートに問い合わせ、新しいライセンスを入手してください。 |
| 004466 | ERROR | system \"%s\" not defined on machine \"%s\". | **原因**: 指定したシステム名が不明です。<br><br>**処置**: システム名を確認し、操作を再試行してください。 |
| 004467 | ERROR | system \"%s\" unknown on machine \"%s\" | **原因**: 指定したシステム名が認識されません。<br><br>**処置**: システム名を確認し、操作を再試行してください。 |
| 004494 | ERROR | COMMAND OUTPUT: %s | **原因**: アクションまたはイベントスクリプトが予期しない結果になりました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004495 | ERROR | COMMAND OUTPUT: %s | **原因**: アクションまたはイベントスクリプトが予期しない結果になりました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004496 | ERROR | COMMAND OUTPUT: %s | **原因**: アクションまたはイベントスクリプトが予期しない結果になりました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004497 | ERROR | COMMAND OUTPUT: %s | **原因**: アクションまたはイベントスクリプトが予期しない結果になりました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004498 | ERROR | COMMAND OUTPUT: %s | **原因**: アクションまたはイベントスクリプトが予期しない結果になりました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004499 | ERROR | COMMAND OUTPUT: %s | **原因**: アクションまたはイベントスクリプトが予期しない結果になりました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004500 | ERROR | COMMAND OUTPUT: %s | **原因**: アクションまたはイベントスクリプトが予期しない結果になりました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 004501 | ERROR | COMMAND OUTPUT: %s | **原因**: アクションまたはイベントスクリプトが予期しない結果になりました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004502 | ERROR | COMMAND OUTPUT: %s | **原因**: アクションまたはイベントスクリプトが予期しない結果になりました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004503 | ERROR | COMMAND OUTPUT: %s | **原因**: アクションまたはイベントスクリプトが予期しない結果になりました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004504 | ERROR | COMMAND OUTPUT: %s | **原因**: アクションまたはイベントスクリプトが予期しない結果になりました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004505 | ERROR | COMMAND OUTPUT: %s | **原因**: アクションまたはイベントスクリプトが予期しない結果になりました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004506 | ERROR | COMMAND OUTPUT: %s | **原因**: アクションまたはイベントスクリプトが予期しない結果になりました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004507 | ERROR | COMMAND OUTPUT: %s | **原因**: アクションまたはイベントスクリプトが予期しない結果になりました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004508 | ERROR | COMMAND OUTPUT: %s | **原因**: アクションまたはイベントスクリプトが予期しない結果になりました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004509 | ERROR | COMMAND OUTPUT: %s | **原因**: アクションまたはイベントスクリプトが予期しない結果になりました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004510 | ERROR | COMMAND OUTPUT: %s | **原因**: アクションまたはイベントスクリプトが予期しない結果になりました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004511 | ERROR | COMMAND OUTPUT: %s | **原因**: アクションまたはイベントスクリプトが予期しない結果になりました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 004512 | ERROR | %s | **原因**: リモートマシンでエラーが発生しました。<br><br>**処置**: リモートマシンのログを確認し、詳細を確認してください。 |
| 004565 | ERROR | can’t set resource state type to ILLSTATE | **原因**: リソースを不正な状態に配置しようとしました。<br><br>**処置**: リソースを不正な状態で配置しないでください。 |
| 004567 | ERROR | split brain detected while setting resource \"%s\" to \"%s\" state (SHARED equivalency to resource \"%s\" on machine \"%s\" which is in state \"%s\"). Setting local resource ISP but aborting the operation. | **原因**: マシン {machine} の SHARED イクイバレンシの プライマリリソース {resource} が {state} という状態にあるため、リソース {resource} を {state} という状態に変更できませんでした。<br><br>**処置**: スプリットブレインが発生しました。フェールオーバー操作は中止されました。スプリットブレインリソース（複数のシステム上の ISP）を手動で適切な状態にする必要があります。 |
| 004575 | ERROR | COMMAND OUTPUT: %s | |
| 004607 | ERROR | no resource instance has tag \"%s\" | **原因**: 指定したタグのリソースが存在しません。<br><br>**処置**: 有効なタグを指定するか、関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004608 | ERROR | no resource instance has identifier \"%s\" | **原因**: 指定した識別子のリソースが存在しません。<br><br>**処置**: 有効な識別子を指定するか、関連するエラーをログで確認し、レポートされている問題を解決してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 004619 | ERROR | resource with tag \"%s\" already exists with identifier \"%s\" | **原因**: 指定したタグ名はすでに存在します。<br><br>**処置**: 別のタグ名を選択してください。 |
| 004620 | ERROR | resource with identifier \"%s\" already exists with tag \"%s\" | **原因**: 指定した識別子はすでに存在します。<br><br>**処置**: このリソースで使用する別の識別子を選択してください。 |
| 004643 | ERROR | Instance tag name is too long. It must be shorter than %d characters. | **原因**: タグ名が長すぎます。<br><br>**処置**: 256 文字未満のタグ名を指定してください。 |
| 004646 | ERROR | Tag name contains illegal characters | **原因**: タグ名に不正な文字が含まれています。<br><br>**処置**: こちら を参照してください。 |
| 004691 | ERROR | can’t set both tag and identifier at same time | **原因**: タグと識別子が両方指定されました。<br><br>**処置**: タグまたは識別子の一方のみを指定してください。 |
| 004745 | ERROR | failed to access lkexterrlog path=%s | **原因**: システム情報を収集するユーティリティ「lkexterrlog」にアクセスできません。<br><br>**処置**: パッケージ「steeleye-lk」がインストールされていること、およびユーティリティ「lkexterrlog」がアクセス可能であることを確認してください。 |
| 004746 | ERROR | lkexterrlog failed runret=%d cmdline=%s | **原因**: システム情報の収集時に、ユーテ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ィリティ「lkexterrlog」が失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004782 | ERROR | Resource \"%s\" was in state \"%s\" before event occurred – recovery will not be attempted | **原因**: リソースはすでに in service です。リカバリーは実行されません。 |
| 004783 | ERROR | Resource \"%s\" was already in state \"%s\" before event occurred | **原因**: リソースはリカバリーを許可するために適切な状態にありませんでした。<br><br>**処置**: リカバリーが必要な場合は、リソースを ISP 状態にしてください。 |
| 004786 | ERROR | %s on failing resource \"%s\" | **原因**: リソースを復旧しようとしているときにエラーが発生しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004788 | EMERG | failed to remove resource ‘%s’. SYSTEM HALTED. | **原因**: エラーが発生し、リカバリー中にリソースをサービス休止にできませんでした。リソースが 2 つのシステムでアクティブにならないように、システムが再起動されました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004793 | ERROR | lcdsendremote transfer resource \"%s\" to \"%s\" on machine \"%s\" failed (rt=%d) | **原因**: リソースとその依存関係を別のシステムへ転送中に障害が発生しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。関連エラーがないかどう |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | か、もう一方のシステムのログを確認してください。 |
| 004797 | ERROR | Restore of SHARED resource \"%s\" has failed | **原因**: リソースのリストア中にエラーが発生しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 004806 | ERROR | Restore in parallel of resource \"%s\" has failed; will re-try serially | **原因**: パラレルリカバリーが失敗しました。関連するエラーをログで確認し、レポートされている問題を解決してください。<br><br>**処置**: 処置は必要ありません。システムはシリアル形式でリカバリーを続行します。リカバリーが失敗した場合は、復旧できなかったリソースに関連するエラーメッセージを確認し、対応策を探してください |
| 004819 | ERROR | read_temporal_recovery_log(): failed to fopen file: %s. fopen() %s. | **原因**: メモリにロードするために一時リカバリーログファイル {file} を開こうとしたところ、エラー {error} で失敗しました。<br><br>**処置**: システムログファイルを確認し、レポートされているエラーを修正してから、操作を再試行してください。 |
| 004820 | ERROR | read_temporal_recovery_log(): failed to malloc initial buf for temporal_recovery_stamp. | **原因**: ログ情報を格納するためのメモリを取得しようとしている際に、一時リカバリーログ情報のメモリへのロードに失敗しました。<br><br>**処置**: システムログファイルを確認し、レポートされているエラーを修正してから、操作を再試行してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 004821 | ERROR | read_temporal_recovery_log(): failed to reallocate buffer for temporal_recovery_stamp. | **原因**: ログ情報の格納に必要なメモリサイズを増やそうとしている際に、一時リカバリーログ情報のメモリへのロードに失敗しました。<br><br>**処置**: システムログファイルを確認し、レポートされているエラーを修正してから、操作を再試行してください。 |
| 004822 | ERROR | write_temporal_recovery_log(): failed to open file: %s. | **原因**: 一時ファイル {temporary name} のオープンに失敗したため、一時リカバリーログファイルの更新が終了しました。<br><br>**処置**: システムログファイルを確認し、レポートされているエラーを修正してから、操作を再試行してください。 |
| 004823 | ERROR | rename(%s, %s) failed. | **原因**: 一時ファイル {temporary name} を実際のログファイル {real name} に名称変更できなかったため、一時リカバリーログファイルの更新が終了しました。<br><br>**処置**: システムログファイルを確認し、レポートされているエラーを修正してから、操作を再試行してください。 |
| 004829 | FATAL | err=%s line=%d Semid=%d numops=%zd perror=%s | **原因**: セマフォ ID {semaphore} の変更がエラー {err} およびエラーメッセージの説明 {perror} で失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。また、システムログファイルを確認し、レポートされているエラーを修正してから、操作を再試行してください。 |
| 004860 | ERROR | restore ftok failed for resource %s with path %s | **原因**: パス {path} を使用するリソース {tag} のセマフォ操作で使用する IPC キー |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | の生成に失敗しました。これはシステムエラーです。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。また、システムログファイルを確認し、レポートされているエラーを修正してから、操作を再試行してください。 |
| 004861 | ERROR | semget failed with error %d | **原因**: インスタンスファイルに関連付けられたセマフォ識別の取得に失敗しました。これはシステムエラーです。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。また、システムログファイルを確認し、レポートされているエラーを修正してから、操作を再試行してください。 |
| 004862 | ERROR | semctl SEMSET failed with error %d | **原因**: リカバリープロセス中に使用されるセマフォの作成と初期化がエラー {error number} で失敗しました。これはシステムエラーです。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。また、システムログファイルを確認し、レポートされているエラーを修正してから、操作を再試行してください。 |
| 004863 | ERROR | semop failed with error %d | **原因**: リカバリープロセス中に使用されるセマフォの設定がエラー {error number} で失敗しました。これはシステムエラーです。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。また、システムログファイルを確認し、レポートされているエラーを修正してから、操作を再試行してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 004864 | ERROR | semctl SEMSET failed with error %d | **原因**: リカバリープロセス中に使用されるセマフォの解放がエラー {error number} で失敗しました。これはシステムエラーです。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。また、システムログファイルを確認し、レポートされているエラーを修正してから、操作を再試行してください。 |
| 004865 | ERROR | restore action failed for resource %s (exit: %d) | **原因**: リソース {tag} を In Service にする操作が失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。レポートされているエラーを修正し、操作を再試行してください。 |
| 004872 | ERROR | Remote remove of resource \"%s\" on machine \"%s\" failed (rt=%d) | **原因**: ローカルシステムに転送するために {server} でリソース {tag} をサービス休止にする要求が失敗しました。<br><br>**処置**: ローカルシステムで、隣接するログメッセージから詳細を確認してください。また、リソースを削除するには、{server} のログメッセージで障害に関する詳細を確認してください。 |
| 004875 | ERROR | remote remove of resource \"%s\" on machine \"%s\" failed | **原因**: ローカルシステムに転送するために {server} でリソース {tag} をサービス休止にする要求が失敗しました。<br><br>**処置**: ローカルシステムで、隣接するログメッセージから詳細を確認してください。また、リソースを削除するには、{server} のログメッセージで障害に関する詳細を確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 004876 | ERROR | remote remove of resource \"%s\" on machine \"%s\" failed | **原因**: ローカルシステムに転送するために {server} でリソース {tag} をサービス休止にする要求が失敗しました。<br><br>**処置**: ローカルシステムで、隣接するログメッセージから詳細を確認してください。また、リソースを削除するには、{server} のログメッセージで障害に関する詳細を確認してください。 |
| 005045 | ERROR | tli_fdget_i::execute unable to establish a listener port | **原因**: ネットワーク接続を適切に設定できませんでした。<br><br>**処置**: すべてのネットワークハードウェアおよびドライバが適切に設定されていることを確認してください。それでもこのメッセージが表示され、リソースを in service にできない場合は、サポートに問い合わせてください。 |
| 005055 | ERROR | tli_fdget_o::execute – async connect failure | **原因**: ネットワーク接続を適切に設定できませんでした。<br><br>**処置**: すべてのネットワークハードウェアおよびドライバが適切に設定されていることを確認してください。それでもこのメッセージが表示され、リソースを in service にできない場合は、サポートに問い合わせてください。 |
| 005061 | ERROR | tli_fdget_o::execute – bind socket | **原因**: ネットワーク接続を適切に設定できませんでした。<br><br>**処置**: すべてのネットワークハードウェアおよびドライバが適切に設定されていることを確認してください。それでもこのメッセージが表示され、リソースを in service にできない場合は、サポートに問い合わせてください。 |
| 005090 | WARN | system_driver::add_driver: cmd=%s\n | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 005108 | WARN | system_driver::rm_driver: cmd=%s\n | |
| 005145 | ERROR | opening the file | **原因**: パイプを開く、または作成することができませんでした。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 005164 | ERROR | tli_handler::handle-error:sending/receiving data message | **原因**: メッセージの送信または受信に失敗しました。<br><br>**処置**: 詳細については、隣接するログメッセージを確認してください。これは一時的なエラーの可能性がありますが、このエラーが引き続き発生し、サーバーが通信できない場合は、サーバーのネットワーク構成を確認してください。 |
| 005165 | WARN | errno %d\n | **原因**: メッセージの送信または受信に失敗しました。<br><br>**処置**: 詳細については、隣接するログメッセージを確認してください。これは一時的なエラーの可能性がありますが、このエラーが引き続き発生し、サーバーが通信できない場合は、サーバーのネットワーク構成を確認してください。 |
| 005166 | WARN | poll 0x%hx\n | **原因**: メッセージの送信または受信に失敗しました。<br><br>**処置**: 詳細については、隣接するログメッセージを確認してください。これは一時的なエラーの可能性がありますが、このエラーが引き続き発生し、サーバーが通信できない場合は、サーバーのネットワーク構成を確認してください。 |
| 005167 | WARN | handler for sys %s\n | **原因**: メッセージの送信または受信に失 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 敗しました。<br><br>**処置**: 詳細については、隣接するログメッセージを確認してください。これは一時的なエラーの可能性がありますが、このエラーが引き続き発生し、サーバーが通信できない場合は、サーバーのネットワーク構成を確認してください。 |
| 005225 | WARN | so_driver::handle_error: sending/receiving data message errno %d: %s | **原因**: メッセージの送受信に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。これは一時的なエラーの可能性もありますが、このエラーが続き、サーバーが通信できない場合は、サーバーのネットワーク設定を確認してください。 |
| 005235 | WARN | found tcp connection to iwstp\n | |
| 005236 | WARN | didn’t find tcp connection to iwstp\n | |
| 005237 | WARN | lcm_handler retry send from %s:%s to %s:%s (%d)\n | |
| 005238 | WARN | detected duplicate request from %s:%s to %s:%s\n | |
| 005239 | WARN | lcm_handler retry timer set to %d based on lcd remote timeout of %d s (%d s) | |
| 005240 | WARN | lcm_handler retry count/time (%d/%zu) has exceeded the maximum. Giving up…\n | |
| 005241 | WARN | dup_list: %s %s %s %s %d %d (%d) | |
| 005242 | WARN | clean up stale dup_list entry (%d s): %s:%s to %s:%s last: %s | |
| 005243 | WARN | add new dup_list entry %s:%s to %s:%s %d %d (%d | |
| 005244 | WARN | closing fd %d\n | |
| 005245 | WARN | openpoll fd %d\n | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 006012 | ERROR | quickCheck script '%s' (%d) failed to exit after %u seconds. Forcibly terminated. Please examine the script or adjust the LKCHECKINTERVAL parameter in %s. | **原因**: quickCheck スクリプトの処理に時間がかかりすぎているか、スクリプトがハングしています。<br><br>**処置**: メッセージテキストに記載されている手順を実行してください。 |
| 006014 | ERROR | LKCHECKINTERVAL parameter is too short. It is currently set to %ld seconds. It should be at least %ld seconds. Please adjust this parameter in %s and execute 'kill %d' to restart the lkcheck daemon. | **原因**: lkcheck は、LKCHECKINTERVAL 内にすべてのリソースをチェックできませんでした。<br><br>**処置**: 予想よりも長く実行されている quickCheck がないかどうかを確認します。 これは、リソースに問題があることを示している可能性があります。 必要に応じて、LKCHECKINTERVAL パラメーターを増やしてください。 |
| 006102 | ERROR | COMMAND OUTPUT: $LKROOT/bin/sendevent | **原因**: これは、「sendevent」（イベント生成）コマンドの結果です。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 006103 | ERROR | COMMAND OUTPUT: $LKROOT/bin/sendevent | **原因**: これは、「sendevent」（イベント生成）コマンドの結果です。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 006104 | ERROR | COMMAND OUTPUT: $LKROOT/bin/sendevent | **原因**: これは、「sendevent」（イベント生成）コマンドの結果です。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 006502 | ERROR | CPU usage has exceeded the threshold ($threshold%) for $count check cycles. | |
| 006504 | ERROR | Could not open /proc/meminfo | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 006505 | ERROR | Memory usage has exceeded the threshold ($threshold%) for $count check cycles. | |
| 006508 | ERROR | [$SUBJECT event] mail returned $err | |
| 006509 | ERROR | COMMAND OUTPUT: cat /tmp/err$$ | |
| 006511 | ERROR | snmptrap returned $err for Trap 190 | |
| 006512 | ERROR | COMMAND OUTPUT: cat /tmp/err$$ | |
| 006514 | ERROR | [$SUBJECT event] mail returned $err | |
| 006515 | ERROR | COMMAND OUTPUT: cat /tmp/err$$ | |
| 006517 | ERROR | snmptrap returned $err for Trap 200 | |
| 006518 | ERROR | COMMAND OUTPUT: cat /tmp/err$$ | |
| 006520 | ERROR | Failed to update error count in $cpu_file: $! | |
| 006521 | ERROR | Failed to update error count in $mem_file: $! | |
| 006523 | ERROR | The SNHC_CPUCHECK_THRESHOLD setting is not valid. Please set SNHC_CPUCHECK_THRESHOLD to a value between 10 and 99 in /etc/default/LifeKeeper. If not set, the value will default to 99. | |
| 006524 | ERROR | The SNHC_CPUCHECK_TIME setting is not valid. Please set SNHC_CPUCHECK_TIME to a value between 1 and 100 in /etc/default/LifeKeeper. If not set, the value will default to 1. | |
| 006525 | ERROR | The SNHC_MEMCHECK_THRESHOLD setting is not valid. Please set SNHC_MEMCHECK_THRESHOLD to a value between 10 and 99 in /etc/default/LifeKeeper. If not set, the value will default to 99. | |
| 006526 | ERROR | The SNHC_MEMCHECK_TIME setting is not valid. Please set SNHC_MEMCHECK_TIME to a value between 1 and 100 in /etc/default/LifeKeeper. If not set, the value will default to 1. | |
| 006528 | ERROR | Could not open /proc/stat | |
| 006529 | ERROR | Could not open /proc/stat | |
| 006530 | ERROR | Could not use $tmp_path | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 007053 | ERROR | malloc failed. Assume that it is a monitoring target device. | |
| 007058 | ERROR | %s: %s failed on '%s', result:%d, Sense Key = %d. | 原因: SCSI デバイスをリザーブできませんでした。またはステータスを確認できませんでした。これは、ストレージが誤動作しているか、ディスクが別のサーバーによってリザーブされているためです。<br><br>処置: 隣接するログメッセージで詳細を確認し、リソースが適切に処理されていることを確認してください。 |
| 007059 | ERROR | %s: %s failed on '%s', result:%d. | 原因: SCSI デバイスをリザーブできませんでした。またはステータスを確認できませんでした。これは、ストレージが誤動作しているか、ディスクが別のサーバーによってリザーブされているためです。<br><br>処置: 隣接するログメッセージで詳細を確認し、リソースが適切に処理されていることを確認してください。 |
| 007060 | EMERG | %s: failure on device '%s'. SYSTEM HALTED. | 原因: SCSI デバイスをリザーブできませんでした。またはステータスを確認できませんでした。これは、ストレージが誤動作しているか、ディスクが別のサーバーによってリザーブされているためです。サーバーはリブート/停止されます。<br><br>処置: ストレージが適切に動作していることを確認し、適切に動作している場合は、リソースが適切に処理されており、別のサーバーで in service になっていることを確認してください。 |
| 007072 | ERROR | %s: failed to open SCSI device '%s', initiate recovery. errno=0x%x, retry count=%d. | 原因: 保護された SCSI デバイスを開くことができませんでした。デバイスに障 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 害が発生しているか、システムからデバイスが削除されている可能性があります。<br><br>**処置**: システムが停止されるか、バックアップノードへのフェイルオーバーが開始されます。この場合のデフォルトアクションはフェイルオーバーですが、SCSIERROR の設定で変更可能です。 |
| 007073 | ERROR | %s: failed to open SCSI device '%s', RETRY. errno=%d, retry count=%d. | **原因**: 保護された SCSI デバイスを開くことができませんでした。デバイスに障害が発生しているか、システムからデバイスが削除されている可能性があります。<br><br>**処置**: このエラーは致命的なものではありません。操作は 5 秒以内で再試行されます。問題が続行する場合、システムは停止またはリソースのフェイルオーバーを実行してください |
| 007075 | ERROR | %s: RESERVATION CONFLICT on SCSI device '%s'. ret=%d, errno=0x%x, retry count=%d. | **原因**: 別のサーバーと競合しているため、SCSI デバイスをリザーブできませんでした。これは、ストレージが誤動作しているか、ディスクが別のサーバーによってリザーブされているためです。<br><br>**処置**: 隣接するログメッセージで詳細を確認し、リソースが適切に処理されていることを確認してください。 |
| 007077 | ERROR | %s: DEVICE FAILURE on SCSI device '%s', initiate recovery. ret=%d, errno=0x%x, retry count=%d. | **原因**: SCSI デバイスをリザーブできませんでした。またはステータスを確認できませんでした。これは、ストレージが誤動作しているか、ディスクが別のサーバーによってリザーブされているためです。<br><br>**処置**: 隣接するログメッセージで詳細を |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 確認し、リソースが適切に処理されていることを確認してください。 |
| 007078 | ERROR | %s: DEVICE FAILURE on SCSI device '%s', RETRY. ret=%d, errno=0x%x, retry count=%d. | **原因**: SCSI デバイスをリザーブできませんでした。またはステータスを確認できませんでした。これは、ストレージが誤動作しているか、ディスクが別のサーバーによってリザーブされているためです。<br><br>**処置**: 隣接するログメッセージで詳細を確認し、リソースが適切に処理されていることを確認してください。 |
| 010002 | WARN | flag $flag not present, send message again. | **原因**: このメッセージは、再試行される未完のプロセスを示します。<br><br>**処置**: 隣接するログメッセージで、繰り返されているエラーを確認してください。 |
| 010003 | ERROR | COMMAND OUTPUT: $LKBIN/ins_remove | **原因**: このメッセージは「ins_remove」コマンドの結果の一部です。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。これは実際にはエラーではない可能性もあります。 |
| 010006 | WARN | flg_list -d $i took more than $pswait seconds to complete… | **原因**: サーバーのフラグリスト処理の時間が予想以上にかかりました。別のサーバーとの接続に問題が発生している可能性もあります。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 010007 | ERROR | flag $flag not present, switchovers may occur. | **原因**: クラスター内のいずれかのサーバーに、現在のサーバーからのフェイルオ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ーバー操作の不許可を指示できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細を確認し、クラスターで予期しない動作を監視してください。 |
| 010008 | WARN | flag $flag not present, send message again. | **原因**: プロセスは不完全ですが、再試行されます。<br><br>**処置**: 隣接するログメッセージで詳細を確認し、繰り返されている警告/エラーをチェックしてください。 |
| 010023 | FATAL | LifeKeeper failed to initialize properly. | **原因**: LifeKeeper を起動しようとしている間に、致命的なエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 010025 | ERROR | Unable to get a unique tag name on server “%s” for template resource “%s” | **原因**: ストレージリソースの作成プロセスで適合するタグを自動的に生成できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。他のエラーがない場合は、操作を再試行してください。 |
| 010034 | FATAL | Unable to start lcm. | **原因**: ソフトウェアのコアコンポーネントを開始できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細を確認し、レポートされている問題を解決してください。 |
| 010038 | WARN | Waiting for LifeKeeper core components to initialize has exceeded 10 seconds. Continuing anyway, check logs for further de | **原因**: ソフトウェアの一部が、予想よりも起動に時間がかかっています。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | tails. | **処置**: メッセージテキストに記載されている手順を実行してください。 |
| 010039 | WARN | Waiting for LifeKeeper core components to initialize has exceeded 10 seconds. Continuing anyway, check logs for further details. | **原因**: ソフトウェアの一部が、予想よりも起動に時間がかかっています。<br><br>**処置**: メッセージテキストに記載されている手順を実行してください。 |
| 010046 | ERROR | The dependency creation failed on server $SERVER:" `cat $TEMP_FILE` | **原因**: 所定のサーバーで依存関係を作成できませんでした。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 010063 | ERROR | $REMSH error | **原因**: 別のサーバーからデータをバックアップする要求が失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 010085 | ERROR | lkswitchback($MACH): Automatic switchback of \"$loctag\" failed | **原因**: リソースが予想どおりにスイッチバックされませんでした。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 010102 | ERROR | admin machine not specified | **原因**:「getlocks」操作で無効なパラメーターが指定されました。<br><br>**処置**: パラメーターを確認し、操作を再試行してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 010107 | WARN | Lock for $m is ignored because system is OOS | **原因**: ロックが作成されたシステムがアクティブではないため、ロックは無視されました。<br><br>**処置**: 関連するエラーをログで確認してください。これは無害なエラーの可能性もあります。 |
| 010108 | ERROR | lock acquisition timeout | **原因**: ロックの取得に予想よりも時間がかかりました。 または許容時間を超えました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 010109 | ERROR | could not get admin locks." `cat /tmp/ER$$` | **原因**: ソフトウェアは、リソースの管理に必要なロックの取得に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 010112 | ERROR | lcdrcp failed with error no: $LCDRCPRES | **原因**: ファイルを別のサーバーにコピーできませんでした。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 010116 | ERROR | unable to set !lkstop flag | **原因**: サーバーがユーザの要求によって停止されていることを示すためのフラグを設定できませんでした。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 010121 | ERROR | Extended logs aborted due to a failure in opening $destination. ($syserrmsg) | **原因**: 拡張ログファイルを開くときに、ユーティリティ「lkexterrlog」が失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 010124 | ERROR | lkswitchback($MACH): Automatic switchback of \"$loctag\" to $UNAME is not allowed${delay_msg}. | **原因**: $loctag を持つ階層内の 1 つ以上のリソースがスイッチバックできません。<br><br>**処置**: すべてのリソースが使用可能になったら、手動で階層を元に戻してください。 |
| 010125 | WARN | COMMAND OUTPUT: $LKROOT/bin/lkswitchback | **原因**: 現時点ではスイッチバックを完了できません。メッセージで指定された時間内に再試行されます。<br><br>**処置**: 再試行する必要がある場合はなし。遅延リトライを停止するには、 "at" ジョブをキャンセルしてください。 |
| 010132 | ERROR | Unable to retrieve reservation id from "%s". Error: "%s". Attempting to regenerate. | **原因**: SCSI 3 Persistent Reservations に使用する一意の ID の取得元のファイル /opt/LifeKeeper/config/.reservation_id を開くことができません。<br><br>**処置**: ありません。ID の再生成とファイルの更新が試行されます。 |
| 010134 | ERROR | LifeKeeper does not appear to be running. Unable to determine the uniqueness of the reservation ID within the cluster. | **原因**: LifeKeeperが実行されていないため、リソースIDの一意性を確認できません。<br><br>**処置**: LifeKeeper を起動し、”/opt/LifeKeeper/bin/genresid -v” で一意性チェックを再実行してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 010135 | ERROR | The current reservation ID of "%s" is not unique within the cluster. A new ID must be generated by running "%s/bin/genresid -g" on "%s". | **原因**: システムに定義されたリザベーション ID は、クラスター内で一意でないため、使用できません。<br><br>**処置**: このノードのすべてのリソースを out of service にしてから、「/opt/LifeKeeper/bin/genresid -g」を実行して一意のリザベーション ID を生成してください。 |
| 010136 | ERROR | Unable to store reservation id in "%s". Error: "%s" | **原因**: SCSI 3 Persistent Reservations に使用する一意の ID の保存先のファイル /opt/LifeKeeper/config.reservation_id を開くことができません。<br><br>**処置**: ファイルを開くことに失敗した理由としてリストされているエラーを修正し、このノードのすべてのリソースを out of service にしてから、「/opt/LifeKeeper/bin/genresid -g」を実行して新しい一意のリザベーション ID を生成してください。 |
| 010137 | ERROR | Failed to generate a reservation ID that is unique within the cluster. | **原因**: 生成したリザベーション ID はすでにクラスター内の別のノードに定義されています。リザベーション ID はクラスター内で一意である必要があります。<br><br>**処置**: このノードのすべてのリソースを out of service にしてから、「/opt/LifeKeeper/bin/genresid -g」を実行して新しい一意のリザベーション ID を生成してください。 |
| 010138 | ERROR | %s | |
| 010139 | WARN | %s | |
| 010140 | ERROR | $COMMAND_SNMPTRAP returned $exit code for Trap $oid:$result | |
| 010141 | ERROR | LK_TRAP_MGR is specified in /etc/default/LifeKeeper, but $COMMAND_SNMPTR | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | AP command is not in PATH. | |
| 010142 | ERROR | $COMMAND_EMAIL returned $exitcode:$result | |
| 010143 | ERROR | LK_NOTIFY_ALIAS is specified in /etc/default/LifeKeeper, but $COMMAND_EMAIL command is not in PATH. | |
| 010144 | ERROR | can’t opendir $LICENSE_DIR: $! | |
| 010145 | ERROR | lktest failed | |
| 010146 | ERROR | lkcheck failed | |
| 010147 | ERROR | ins_list failed: exit code = $exit_code | |
| 010159 | ERROR | Maintenance mode disable currently in progress, can’t enable maintenance mode. If this problem persists, consider using the —force option. | |
| 010160 | ERROR | Maintenance mode enable currently in progress, can’t disable maintenance mode. If this problem persists, consider using the —force option. | |
| 010161 | ERROR | $tag is not a valid resource tag on the local machine, aborting. Please check the spelling and try again. | |
| 010163 | ERROR | $cmd script not found or not executable on system $sys. | |
| 010165 | ERROR | An error occurred while running \’$LKROOT/lkadm/subsys/appsuite/sap/bin/$cmd -m=${opt_mode}${tag_cmd}${force_cmd}\’ on system $sys (exit code: $remexec_ret). Please inspect the logs on that system for more information. | |
| 010168 | WARN | Maintenance mode was not fully ${opt_mode}d for at least one resource on system $sys. | |
| 010172 | ERROR | Maintenance mode was not fully ${opt_mode}d for the requested resources on at least one system in the cluster. | |
| 010173 | ERROR | LifeKeeper is not running on system $me. Unable to ${opt_mode} maintenance mode. Aborting. | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 010179 | ERROR | Maintenance mode action \'${opt_mode}\' did not complete successfully for resource $tag on system $me. | |
| 010181 | ERROR | An error occurred while attempting maintenance mode action \'${opt_mode}\' on system $me for resources: @{local_hier_tags}. | |
| 010187 | ERROR | Resource $tag has not been extended to system $sys. | |
| 010222 | ERROR | scsifree(%s): LKSCSI_Release(%s) unsuccessful | **原因**: リザーブされていると思われる SCSI デバイスは予想どおりに解放されませんでした。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。システムが正常に機能している場合、このエラーは無害な可能性があります。 |
| 010231 | ERROR | scsiplock(%s): reserve failed. | **原因**: SCSI デバイスでリザーブを取得できませんでした<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 010250 | ERROR | Failed to exec command '%s' | **原因**:「lklogmsg」ツールがサブコマンド {command} の実行に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。サブコマンドが存在しており、有効なコマンドまたはプログラムであることを確認してください。通常の操作でこのメッセージが表示された場合は、サポートに問い合わせてください。 |
| 010256 | ERROR | scsi_tur(%s): open failed. | |
| 010260 | ERROR | scsi_tur(%s): test unit ready failed. | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 010402 | EMERG | local recovery failure on resource $opt s{'N'}, trigger VMware HA… | 原因: LifeKeeper Single Server Protection の動作中に、リソースを復旧できませんでした。(VMware-HA が有効になっている場合は) 障害を処理するために VMware-HA が間もなくトリガされます。<br><br>処置: 処置は必要ありません。VMware が障害を処理してください。 |
| 010413 | ERROR | COMMAND OUTPUT: cat /tmp/err$$ | 原因: これは、失敗した可能性がある「snmptrap」コマンドの結果です。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 010420 | EMERG | local recovery failure on resource $opt s{'N'}, trigger reboot… | 原因: LifeKeeper Single Server Protection の動作中に、リソースを復旧できませんでした。障害を処理するために再起動が間もなくトリガされます。<br><br>処置: 処置は必要ありません。 |
| 010440 | ERROR | [$SUBJECT event] mail returned $err | 原因: これは、通知電子メールを「mail」コマンドで送信できなかったことを示してください。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 010443 | ERROR | COMMAND OUTPUT: cat /tmp/err$$ | 原因: これは、失敗した可能性がある「mail」コマンドの結果です。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 010445 | ERROR | COMMAND OUTPUT: cat /tmp/err$$ | 原因: これは、失敗した可能性がある「mail」コマンドの結果です。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 010463 | ERROR | LifeKeeper: name of machine is not specified, ARGS=$ARGS | 原因:「comm_down」イベントで無効な引数が指定されました。<br><br>処置: LifeKeeper 設定を確認し、操作を再試行してください。 |
| 010471 | ERROR | COMM_DOWN: Attempt to obtain local comm_down lock flag failed | 原因: 別のノードとの通信障害の処理中に、ローカルロックを取得できませんでした。これにより、フェイルオーバーの処理が中断されます。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。フェイルオーバーが正常に実行されない場合は、サポートに問い合わせてください。 |
| 010482 | ERROR | LifeKeeper: name of machine is not specified, ARGS=$ARGS | 原因:「comm_up」イベントで無効な引数が指定されました。<br><br>処置: LifeKeeper 設定を確認し、操作を再試行してください。 |
| 010484 | WARN | flg_list -d $MACH check timed-out ($delay seconds). | 原因:「flg_list」コマンドがタイムアウト値 {delay} 秒に達しました。 |
| 010487 | WARN | flg_list -d $MACH check timed-out, unintended switchovers may occur. | 原因:「flg_list」コマンドがタイムアウト値に達しました。<br><br>処置: 意図しないスイッチオーバーが発生した場合は、リソースツリーをスイッ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | チバックしてください。 |
| 010492 | WARN | $m | **原因**: 他のサーバーのいずれかが、このサーバー {server} には稼働していないように見えますが、Witness サーバーが承諾していません。<br><br>**処置**: 他のサーバーが稼働していないことを確認し、リソースを手動でスイッチオーバーしてください。 |
| 010494 | ERROR | LifeKeeper: COMM_UP to machine $MACH completed with errors. | **原因**:「COMM_UP」イベントの間に、予期しない障害が発生しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 010503 | ERROR | lcdrecover hung or returned error, attempting kill of process $FPID | **原因**:「lcdrecover」の処理に時間がかかりすぎているか、エラーアウトしています。 |
| 010506 | ERROR | Intelligent Switchback Check Failed | **原因**:「lcdrecover」を 5 回実行し、いずれも失敗しました。<br><br>**処置**: リソースツリーを手動でスイッチオーバーしてください。 |
| 010535 | ERROR | LifeKeeper: name of machine is not specified, ARGS=$ARGS | |
| 010600 | ERROR | removing hierarchy remnants | |
| 010627 | WARN | Equivalency Trim: does not have a full complement of equivalencies. Hierarchy will be unextended from | **原因**: 拡張中にエラーが発生しました。<br><br>**処置**: LifeKeeper のログを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 010629 | WARN | Your hierarchy exists on only one server. Your application has no protection until yo u extend it to at least one other server. | **原因**: 階層は 1 番目のサーバー上にあります。<br><br>**処置**: 階層を 2 番目のサーバーに拡張してください。 |
| 010712 | ERROR | Unextend hierarchy failed | **原因**: サーバーからリソース階層を拡張できませんでした。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 010746 | ERROR | $ERRMSG Target machine \"$TARGE T_MACH\" does not have an active LifeK eeper communication path to machine \"$aMach\" in the hierarchy." >&2 | **原因**: ターゲットサーバーがクラスター内の他のサーバーと適切に通信していないため、階層を拡張できません。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。すべてのサーバーが相互にコミュニケーションパスを持っていることを確認してください。 |
| 010763 | ERROR | lock failed | |
| 011000 | ERROR | appremote: unknown command type %d(‘%c’)\n | **原因**: 内部エラー。<br><br>**処置**: 製品を再起動してください。 |
| 011001 | ERROR | depremote: unknown command type %d(‘%c’)\n | **原因**: 内部エラー。<br><br>**処置**: 製品を再起動してください。 |
| 011002 | ERROR | eqvremote: unknown command type %d(‘%c’)\n | **原因**: 内部エラー。<br><br>**処置**: 製品を再起動してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 011003 | ERROR | flgremote: unknown command type %d(‘%c’)\n | 原因: 内部エラー。<br><br>処置: 製品を再起動してください。 |
| 011004 | WARN | Illegal creation of resource | 原因: これは、通常の状態では発生しません。 |
| 011009 | ERROR | insremote: unknown change field command type %d(‘%c’)\n | 原因: 内部エラー。<br><br>処置: 製品を再起動してください。 |
| 011010 | ERROR | insremote: unknown command type %d(‘%c’)\n | 原因: 内部エラー。<br><br>処置: 製品を再起動してください。 |
| 011011 | FATAL | %s | 原因: LifeKeeper は IPC キーを取得できませんでした。<br><br>処置: 隣接するログメッセージで詳細を確認してください。 |
| 011012 | FATAL | semget(%s,%c) | 原因: LifeKeeper はセマフォ設定 ID を取得できませんでした。<br><br>処置: 隣接するログメッセージで詳細を確認してください。 |
| 011013 | FATAL | shmget(%s,%c) | 原因: システムは共有メモリセグメントを割り当てることができませんでした。<br><br>処置: 隣接するログメッセージで詳細を確認してください。 |
| 011014 | FATAL | prefix_lkroot("out")); | 原因: /opt/LifeKeeper/out にアクセス中に、システムエラーが発生しました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: /opt/LifeKeeper/out にアクセスできない理由を特定してください。 |
| 011015 | ERROR | Please contact the original supplier of your evaluation software, or visit http://www.us.sios.com for information about purchasing a permanent version of the LifeKeeper product. | **原因**: デモライセンスを実行しています。<br><br>**処置**: サポートに問い合わせ、新しいライセンスを入手してください。 |
| 011016 | ERROR | The product license installed on this system indicates single node operation but the LifeKeeper Single Server Protection package does not appear to be installed. Please install the LifeKeeper SSP package to ensure proper operation | **原因**: LifeKeeper Single Server Protection のライセンスがありますが、LifeKeeper Single Server Protection がインストールされていません。<br><br>**処置**: LifeKeeper Single Server Protection をインストールするか、実行している製品に適合するライセンスを入手してください。 |
| 011018 | ERROR | License manager initialization failed. | **原因**: ライセンスマネージャの初期化に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 011020 | EMERG | License manager initialization failed. | **原因**: ライセンスマネージャの初期化に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 011021 | EMERG | Error in obtaining LifeKeeper license key. | **原因**: ライセンスに問題があります。<br><br>**処置**: サポートに問い合わせ、新しいライセンスを入手してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 011022 | EMERG | Error in obtaining LifeKeeper license key. | **原因**: ライセンスに問題があります。<br><br>**処置**: サポートに問い合わせ、新しいライセンスを入手してください。 |
| 011023 | EMERG | A recovery kit Protection Suite WAS NOT FOUND with the restricted LifeKeeper license. | **原因**: ライセンスに問題があります。<br><br>**処置**: サポートに問い合わせ、新しいライセンスを入手してください。 |
| 011024 | EMERG | Error in obtaining LifeKeeper license key. | **原因**: ライセンスに問題があります。<br><br>**処置**: サポートに問い合わせ、新しいライセンスを入手してください。 |
| 011025 | EMERG | A recovery kit Protection Suite WAS NOT FOUND with the restricted LifeKeeper license. | **原因**: LifeKeeper は有効なライセンスキーを見つけられませんでした。<br><br>**処置**: ライセンスキーがサーバーに対して有効であることを確認し、操作を再試行してください。 |
| 011026 | EMERG | Error in obtaining LifeKeeper license key. | **原因**: ライセンスに不明な問題があります。<br><br>**処置**: サポートに問い合わせ、新しいライセンスを入手してください。 |
| 011027 | EMERG | A recovery kit Protection Suite WAS NOT FOUND with the restricted LifeKeeper license. | **原因**: ライセンスに問題があります。<br><br>**処置**: サポートに問い合わせ、新しいライセンスを入手してください。 |
| 011028 | ERROR | Error, This version of LifeKeeper must be running in the %c%c locale. The environment variable %s is not set correctly. Please consult your documentation for how to set this. | **原因**: 設定されているロケールに問題があります。<br><br>**処置**: マニュアルなどを参照の上、適切 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | なロケールを設定してください。 |
| 011029 | FATAL | can’t set reply system | **原因**: メッセージの送信に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。これは一時的なエラーの可能性もあります。 |
| 011030 | FATAL | can’t set reply mailbox | **原因**: メッセージの送信に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。これは一時的なエラーの可能性もあります。 |
| 011031 | ERROR | Failure reading output of ‘%s’ on behalf of %s | **原因**: 一時ファイル /tmp/OUT.{pid} にアクセスしている際にシステムエラーが発生しました。<br><br>**処置**: /tmp/OUT.{pid} にアクセスできない理由を特定してください。 |
| 011032 | ERROR | Failure reading output of ‘%s’ | **原因**: 一時ファイル /tmp/ERR.{pid} にアクセスしている際にシステムエラーが発生しました。<br><br>**処置**: /tmp/ERR.{pid} にアクセスできない理由を特定してください。 |
| 011033 | ERROR | event \"%s,%s\" already posted for resource with id \"%s\" | **原因**: このメッセージは情報提供の目的でのみ表示されます。 |
| 011034 | ERROR | no resource has id of \"%s\" | **原因**: LifeKeeper は {id} リソースを見つけられませんでした。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: パラメーターを確認し、「sendevent」操作を再試行してください。 |
| 011044 | ERROR | flagcleanup:fopen(%s) | **原因**: /opt/LifeKeeper/config/flg を読み取っている際にシステムエラーが発生しました。<br><br>**処置**: /opt/LifeKeeper/config/flg を読み取ることができない理由を特定してください。 |
| 011045 | ERROR | flagcleanup:fopen(%s) | **原因**: /opt/LifeKeeper/config/flg を書き込んでいる際にシステムエラーが発生しました。<br><br>**処置**: /opt/LifeKeeper/config/flg を書き込むことができない理由を特定してください。 |
| 011046 | ERROR | flagcleanup:fputs(%s) | **原因**: /opt/LifeKeeper/config/flg を書き込んでいる際にシステムエラーが発生しました。<br><br>**処置**: /opt/LifeKeeper/config/flg を書き込むことができない理由を特定してください。 |
| 011047 | ERROR | flagcleanup:rename(%s,%s) | **原因**: /opt/LifeKeeper/config/.flg to /opt/LifeKeeper/config/flg の名前を変更している際にシステムエラーが発生しました。<br><br>**処置**: /opt/LifeKeeper/config/flg の名前を変更できない理由を特定してください。 |
| 011048 | ERROR | flagcleanup:chmod(%s) | **原因**: /opt/LifeKeeper/config/flg の権限を変更している際にシステムエラーが発 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 生しました。<br><br>**処置**: LifeKeeper が /opt/LifeKeeper/config/flg の権限を変更できない理由を特定してください。 |
| 011049 | ERROR | License check failed with error code %d | **原因**: ライセンスに問題があります。<br><br>**処置**: サポートに問い合わせ、新しいライセンスを入手してください。 |
| 011051 | ERROR | lcdinit: clearing Disk Reserve file failed | **原因**: /opt/LifeKeeper/subsys/scsi/resources/disk/disk.reserve を書き込んでいる際にシステムエラーが発生しました<br><br>**処置**: /opt/LifeKeeper/subsys/scsi/resources/disk/disk.reserve を書き込むことができない理由を特定してください。 |
| 011052 | FATAL | malloc() failed | **原因**: システムは LifeKeeper 用にメモリを割り当てられませんでした。<br><br>**処置**: データセグメントのプロセスの上限を増やしてください。 |
| 011053 | FATAL | /tmp/LK_IS_UNAVAIL | **原因**: /tmp/LK_IS_UNAVAIL を書き込んでいる際にシステムエラーが発生しました。<br><br>**処置**: /tmp/LK_IS_UNAVAIL を書き込むことができない理由を特定してください。 |
| 011054 | FATAL | /tmp/LK_IS_UNAVAIL | **原因**: /tmp/LK_IS_UNAVAIL を書き込んでいる際にシステムエラーが発生しました。<br><br>**処置**: /tmp/LK_IS_UNAVAIL を書き込む |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ことができない理由を特定してください。 |
| 011055 | FATAL | config/LK_IS_ON | 原因: config/LK_IS_ON を書き込んでいる際にシステムエラーが発生しました。<br><br>処置: config/LK_IS_ON を書き込むことができない理由を特定してください。 |
| 011056 | ERROR | License check failed with error code %d | 原因: ライセンスに問題があります。<br><br>処置: サポートに問い合わせ、新しいライセンスを入手してください。 |
| 011057 | ERROR | lcdremote: unknown command type %d(‘%c’)\n | 原因: メッセージの受信に失敗しました。<br><br>処置: 隣接するログメッセージで詳細を確認してください。これは一時的なエラーの可能性もありますが、このエラーが続き、サーバーが通信できない場合は、サーバーのネットワーク設定を確認してください。 |
| 011059 | FATAL | Could not write to: %s | 原因: /opt/LifeKeeper/config/LK_START_TIME にアクセスしている際にシステムエラーが発生しました。<br><br>処置: /opt/LifeKeeper/config/LK_START_TIME にアクセスできない理由を特定してください |
| 011060 | FATAL | received NULL message | 原因: メッセージの受信に失敗しました。<br><br>処置: 隣接するログメッセージで詳細を確認してください。これは一時的なエラーの可能性もありますが、このエラーが |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 続き、サーバーが通信できない場合は、サーバーのネットワーク設定を確認してください。 |
| 011061 | ERROR | unknown data type %d(‘%c’) on machine \"%s\"\n | **原因**: メッセージの受信に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。これは一時的なエラーの可能性もありますが、このエラーが続き、サーバーが通信できない場合は、サーバーのネットワーク設定を確認してください。 |
| 011062 | WARN | LifeKeeper shutdown in progress. Unable to perform failover recovery processing for %s\n | **原因**: LifeKeeper はシャットダウン中に所定のリソースをフェイルオーバーできませんでした。<br><br>**処置**: リソースツリーを別のサーバーに手動でスイッチオーバーしてください。 |
| 011063 | WARN | LifeKeeper resource initialization in progress. Unable to perform failover recovery processing for %s\n | **原因**: LifeKeeper は起動中に所定のリソースをフェイルオーバーできませんでした。<br><br>**処置**: LifeKeeper が起動したら、リソースツリーを手動でスイッチオーバーしてください。 |
| 011068 | ERROR | ERROR on command %s | **原因**:「rlslocks」コマンドの実行中にエラーが発生しました。<br><br>**処置**: 隣接するメッセージで詳細を確認してください。 |
| 011070 | ERROR | ERROR on command %s | **原因**:「getlocks」コマンドの実行中にエラーが発生しました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 隣接するログメッセージで詳細を確認してください。 |
| 011080 | FATAL | out of memory | **原因**: 内部エラー。<br><br>**処置**: データセグメントのプロセスの上限値を増やしてください。 |
| 011081 | FATAL | Failed to ask ksh to run: %s | **原因**: ksh の呼び出し中にシステムエラーが発生しました。<br><br>**処置**: pdksh (v8.0 以前) または steeley e-pdksh (v81 以降) のパッケージがインストールされていることを確認してください。 |
| 011082 | ERROR | Failed to remove: %s | **原因**: /tmp/LCM_IS_UNAVAIL を削除している際にシステムエラーが発生しました。<br><br>**処置**: /tmp/LCM_IS_UNAVAIL を削除できない理由を特定してください。 |
| 011083 | ERROR | Failed to remove: %s | **原因**: /tmp/LK_IS_UNAVAIL をリンク解除している際にシステムエラーが発生しました。<br><br>**処置**: /tmp/LK_IS_UNAVAIL をリンク解除できない理由を特定してください。 |
| 011084 | FATAL | Failed to generate an IPC key based on: %s | **原因**: /opt/LifeKeeper にアクセス中に、システムエラーが発生しました。<br><br>**処置**: /opt/LifeKeeper にアクセスできない理由を特定してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 011085 | ERROR | semget(%s,%c) failed | **原因**: セマフォを削除している際にシステムエラーが発生しました。<br><br>**処置**: セマフォを手動で削除してください。 |
| 011086 | ERROR | shmget(%s,%c) failed | **原因**: 共有メモリセグメントを削除している際にシステムエラーが発生しました。<br><br>**処置**: 共有メモリセグメントを手動で削除してください。 |
| 011087 | ERROR | semctl(IPC_RMID) failed | **原因**: セマフォを削除している際にシステムエラーが発生しました。<br><br>**処置**: セマフォを手動で削除してください。 |
| 011088 | ERROR | shmctl(IPC_RMID) failed | **原因**: 共有メモリセグメントを削除している際にシステムエラーが発生しました。<br><br>**処置**: 共有メモリセグメントを手動で削除してください。 |
| 011089 | FATAL | Execution of lcdstatus on remote system <%s> failed\n | **原因**: リモート {node} がダウンし、ネットワークからアクセスできません。または、リモートノードで別のシステムに問題が発生しています。<br><br>**処置**: リモートノードをオンラインにするか、隣接するメッセージで詳細を確認するか、リモートノードのログで詳細を確認してください。 |
| 011090 | FATAL | | **原因**: 内部エラー。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: メッセージ本文に記載されている手順を実行してください。 |
| 011091 | WARN | | **原因**: これは通常の状況では発生しません。 |
| 011092 | FATAL | | **原因**: ライセンスに問題があります。<br><br>**対応**: メッセージ本文に記載されている手順を実行してください。 |
| 011093 | FATAL | | **原因**: ライセンスに問題があります。<br><br>**対応**: メッセージ本文に記載されている手順を実行してください。 |
| 011094 | FATAL | | **原因**:ライセンスに問題があります。<br><br>**対応**: メッセージ本文に記載されている手順を実行してください。 |
| 011095 | FATAL | | **原因**: ライセンスに問題があります。<br><br>**対応**:メッセージ本文に記載されている手順を実行してください。 |
| 011096 | FATAL | | **原因**: ライセンスに問題があります。<br><br>**対応**: メッセージ本文に記載されている手順を実行してください。 |
| 011097 | FATAL | | **原因**: ライセンスに問題があります。<br><br>**対応**: 新しいライセンスを取得するには、サポートに連絡してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 011098 | FATAL | | **原因**: ライセンスに問題があります。<br><br>**対応**: 新しいライセンスを取得するには、サポートに連絡してください。 |
| 011099 | FATAL | | **原因**: ライセンスに問題があります。<br><br>**対応**: 新しいライセンスを取得するには、サポートに連絡してください。 |
| 011100 | FATAL | | **原因**: ライセンスに問題があります。<br><br>**対応**: 新しいライセンスを取得するには、サポートに連絡してください。 |
| 011101 | FATAL | | **原因**: ライセンスに問題があります。<br><br>**対応**: 新しいライセンスを取得するには、サポートに連絡してください。 |
| 011102 | FATAL | | **原因**: ライセンスに問題があります。<br><br>**対応**: 新しいライセンスを取得するには、サポートに連絡してください。 |
| 011103 | FATAL | | **原因**: ライセンスに問題があります。<br><br>**対応**: 新しいライセンスを取得するには、サポートに連絡してください。 |
| 011104 | FATAL | | **原因**: ライセンスに問題があります。<br><br>**対応**: 新しいライセンスを取得するには、サポートに連絡してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 011105 | FATAL | | **原因**: ライセンスに問題があります。<br><br>**対応**: メッセージ本文に記載されている手順を実行してください。 |
| 011111 | ERROR | action \"%s\" on resource with tag \"%s\" has failed | **原因**: リソース {tag} のアクション {action} に失敗しました。<br><br>**処置**: 隣接するメッセージで詳細を確認してください。 |
| 011112 | ERROR | netremote: network “%s” with device “%s” not found on “%s” | **原因**: LifeKeeper はネットワークデバイスを見つけられませんでした。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 011113 | ERROR | netremote: unknown subcommand type %d(‘%c’)\n | **原因**: 内部エラー。<br><br>**処置**: 製品を再起動してください。 |
| 011114 | ERROR | netremote: unknown command type %d(‘%c’)\n | **原因**: 内部エラー。<br><br>**処置**: 製品を再起動してください。 |
| 011117 | ERROR | sysremote: system \"%s\" not found on \"%s\" | **原因**: 無効なシステム名が指定されました。<br><br>**処置**: システム名を再確認し、コマンドを再実行してください。 |
| 011118 | ERROR | sysremote: unknown subcommand type %d(‘%c’)\n | **原因**: 内部エラー。<br><br>**処置**: 製品を再起動してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 011119 | ERROR | sysremote: unknown command type %d(‘%c’)\n | **原因**: 内部エラー。<br><br>**処置**: 製品を再起動してください。 |
| 011120 | ERROR | typremote: unknown command type %d(‘%c’)\n | **原因**: 内部エラー。<br><br>**処置**: 製品を再起動してください。 |
| 011129 | ERROR | Failure during run of ‘%s’ on behalf of %s | **原因**: コマンドの実行に失敗しました。<br><br>**処置**: メッセージで詳細を確認し、問題を特定してください。 |
| 011130 | ERROR | %s | **原因**: コマンド {command} が予期しない結果になりました。<br><br>**処置**: 隣接するエラーメッセージの内容に従って処置を実行してください。 |
| 011131 | EMERG | \nERROR!\nThe update of the expiration timestamp for the evaluation version of the LifeKeeper product on this system has failed. The evaluation product will no longer execute.\n\nPlease contact the original supplier of your evaluation software for assistance.\n\n | **原因**: デモライセンスに問題があります。<br><br>**処置**: サポートに問い合わせ、新しいライセンスを入手してください。 |
| 011132 | EMERG | \nERROR!\nThe expiration timestamp associated with the evaluation version of the LifeKeeper product on this system has been improperly altered. The evaluation software expires immediately.\n\nPlease contact the original supplier of your evaluation software, or visit http://www.us.sios.com for information about purchasing a permanent version of the LifeKeeper product.\n\n | **原因**: デモライセンスで、時間の改ざんが検出されました。<br><br>**処置**: サポートに問い合わせ、新しいライセンスを入手してください。 |
| 011133 | EMERG | \nERROR!\nThe expiration timestamp associated with the evaluation version of the LifeKeeper product on this system has been improperly altered. The evaluation software expires immediately.\n\nPlease cont | **原因**: デモライセンスで、時間の改ざんが検出されました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | act the original supplier of your evaluation software, or visit http://www.us.sios.com for information about purchasing a permanent version of the LifeKeeper product.\n\n | **処置**: サポートに問い合わせ、新しいライセンスを入手してください。 |
| 011134 | EMERG | \nERROR!\nThe evaluation version of the LifeKeeper product on this system has expired.\n\nPlease contact the original supplier of your evaluation software, or visit http://www.us.sios.com for information about purchasing a permanent version of the LifeKeeper product.\n\n | **原因**: この製品のデモライセンスが期限切れです。<br><br>**処置**: サポートに問い合わせ、新しいライセンスを入手してください。 |
| 011135 | EMERG | \nERROR!\nThe expiration timestamp associated with the evaluation version of the LifeKeeper product on this system has been improperly altered. The evaluation software expires immediately.\n\nPlease contact the original supplier of your evaluation software, or visit http://www.us.sios.com for information about purchasing a permanent version of the LifeKeeper product.\n\n | **原因**: デモライセンスで、時間の改ざんが検出されました。<br><br>**処置**: サポートに問い合わせ、新しいライセンスを入手してください。 |
| 011136 | EMERG | \nIMPORTANT NOTICE!\nThe LifeKeeper product running on this system is an evaluation version which will expire on %s\nTo continue functioning beyond that time, the software must be upgraded to the permanent version.\n\nPlease contact the original supplier of your evaluation software, or visit http://www.us.sios.com for information about purchasing a permanent version of the LifeKeeper product.\n\n | **原因**: デモライセンスを実行しています。<br><br>**処置**: サポートに問い合わせ、新しいライセンスを入手してください。 |
| 011138 | EMERG | The LifeKeeper product on this system is using an evaluation license key which will expire at midnight on %s. To continue functioning beyond that time, a permanent license key must be obtained. | **原因**: デモライセンスを実行しています。<br><br>**処置**: サポートに問い合わせ、新しいライセンスを入手してください。 |
| 011142 | WARN | LifeKeeper Recovery Kit %s license key NOT FOUND | **原因**: {kit} の Application Recovery Kit のライセンスが見つかりませんでした。<br><br>**処置**: サポートに問い合わせ、新しいラ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | イセンスを入手してください。 |
| 011150 | ERROR | COMMAND OUTPUT: %s | **原因**: コマンド「eventslcm」が予期しない結果になりました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 011151 | EMERG | %s | **原因**: このバージョンの LifeKeeper Core パッケージは、中華人民共和国内または日本国内での使用に限られています。 |
| 011152 | EMERG | Localized license failure | **原因**: お客様のロケールと製品ライセンスが作成されたロケールが一致していません。<br><br>**処置**: サポートに問い合わせ、お客様のロケールに合った新しいライセンスを入手してください。 |
| 011154 | EMERG | Single Node flag check failed. | **原因**: LifeKeeper Single Server Protection のライセンスがありますが、LifeKeeper Single Server Protection がインストールされていません。<br><br>**処置**:LifeKeeper Single Server Protection をインストールするか、実行している製品に適合するライセンスを入手してください。 |
| 011155 | EMERG | %s | **原因**: この製品のライセンスキーが期限切れです。<br><br>**処置**: サポートに問い合わせ、新しいライセンスを入手してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 011162 | EMERG | %s | 原因: この製品のライセンスキーが期限切れです。<br><br>処置: サポートに問い合わせ、新しいライセンスを入手してください。 |
| 011163 | EMERG | Single Node license check failed | 原因: LifeKeeper Single Server Protection のライセンスがありますが、LifeKeeper Single Server Protection がインストールされていません。<br><br>処置: LifeKeeper Single Server Protection をインストールするか、実行している製品に適合するライセンスを入手してください。 |
| 011164 | EMERG | %s | 原因: この製品のライセンスキーが期限切れです。<br><br>処置: サポートに問い合わせ、製品のパーマネントライセンスキーを入手してください。 |
| 011165 | ERROR | LifeKeeper initialize timed out in tag \"%s\" | |
| 015000 | ERROR | COMMAND OUTPUT: /opt/LifeKeeper/sbin/chpst | 原因:「steeleye-lighttpd」プロセスでエラーが発生しました。 エラーの詳細は実際のログメッセージに記載されています。<br><br>処置: 設定を修正すると、「steeleye-lighttpd」が自動的に再起動します。 |
| 103001 | ERROR | LifeKeeper has detected an error while trying to determine the node number(s) of the DB partition server(s) for the instance | 原因: db2nodes.cfg にサーバー名が含まれていません。<br><br>処置: db2nodes.cfg が有効であることを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 103002 | ERROR | LifeKeeper was unable to get the version for the requested instance "%s" | 原因:「db2level」コマンドが DB2 のバージョンを返しませんでした。<br><br>処置: DB2 の設定を確認してください。 |
| 103003 | ERROR | LifeKeeper has detected an error while trying to determine the node number(s) of the DB partition server(s) for the instance | 原因: DB2 Application Recovery Kit は、DB2 インスタンスのノードを見つけることができませんでした。<br><br>処置: DB2 の設定を確認してください。 |
| 103004 | ERROR | Unable to get the information for resource "%s" | 原因: リソース情報の取得に失敗しました。<br><br>処置: LifeKeeper の設定を確認してください。 |
| 103005 | ERROR | Unable to get the information for resource "%s" | 原因: リソース情報の取得に失敗しました。<br><br>処置: LifeKeeper の設定を確認してください。 |
| 103006 | ERROR | Unable to get the instance information for resource "%s" | 原因: インスタンス情報の取得に失敗しました。<br><br>処置: LifeKeeper の設定を確認してください。 |
| 103007 | ERROR | Unable to get the instance home directory information for resource "%s" | 原因: インスタンスのホームディレクトリのパスの取得に失敗しました。<br><br>処置: LifeKeeper の設定を確認してください。 |
| 103008 | ERROR | Unable to get the instance type information for resource "%s" | 原因: DB2 Application Recovery Kit は、無効なインスタンスタイプを見つけ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ました。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 103009 | ERROR | LifeKeeper has encountered an error while trying to get the database configuration parameters for database "$DB" | **原因**:「db2 get db cfg for $DB」コマンドの実行中に予期しないエラーが発生しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 103012 | ERROR | LifeKeeper was unable to start the database server for instance "%s" | **原因**: 要求された DB2 インスタンスの起動が失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。レポートされているエラーを修正してから、「restore」操作を再試行してください。 |
| 103013 | ERROR | LifeKeeper was unable to start the database server for instance "%s" | **原因**: 要求された DB2 インスタンスの起動が失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。レポートされているエラーを修正してから、「restore」操作を再試行してください。 |
| 103015 | ERROR | An entry for the home directory "%s" of instance "%s" does not exist in "/etc/fstab" | **原因**: マルチパーティションのデータベースのインスタンスのホームディレクトリは、「/etc/fstab」に存在する必要があります。<br><br>**処置**: ホームディレクトリが「/etc/fstab」に存在することを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 103016 | ERROR | LifeKeeper was unable to mount the home directory for the DB2 instance "%s" | **原因**: マルチパーティションのデータベースのインスタンスのホームディレクトリをマウントできませんでした。<br><br>**処置**: ホームディレクトリがマウントされていることを確認し、操作を再試行してください。 |
| 103017 | ERROR | Unable to get the instance nodes information for resource "%s" | **原因**: インスタンスノードの取得に失敗しました。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 103018 | ERROR | LifeKeeper was unable to start database partition server "%s" for instance "%s" | **原因**: 要求された DB2 インスタンスの起動が失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。レポートされているエラーを修正してから、「restore」操作を再試行してください。 |
| 103020 | ERROR | LifeKeeper was unable to stop the database server for instance "%s" | **原因**: 要求された DB2 インスタンスのシャットダウンが失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。レポートされているエラーを修正してから、「remove」操作を再試行してください。 |
| 103021 | ERROR | LifeKeeper was unable to stop the database server for instance "%s" | **原因**: 要求された DB2 インスタンスのシャットダウンが失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。レポートされているエラーを修正してから、「remove」操作を再試行 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | してください。 |
| 103023 | ERROR | Unable to get the instance nodes information for resource "%s" | **原因**: インスタンスノードの取得に失敗しました。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 103024 | ERROR | LifeKeeper was unable to stop database partition server "%s" for instance "%s" | **原因**: 要求された DB2 インスタンスのシャットダウンが失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。レポートされているエラーを修正してから、「remove」操作を再試行してください。 |
| 103026 | ERROR | Unable to get the instance nodes information for resource "%s" | **原因**: インスタンスノードの取得に失敗しました。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 103027 | FATAL | The argument for the DB2 instance is empty | **原因**: 作成操作で無効なパラメーターが指定されました。<br><br>**処置**: パラメーターを確認し、操作を再試行してください。 |
| 103028 | FATAL | Unable to determine the DB2 instance home directory | **原因**: DB2 Application Recovery Kit は、DB2 インスタンスのホームディレクトリを特定できませんでした。<br><br>**処置**: インスタンスの所有者の名前がインスタンス名と同じであることを確認し、操作を再試行してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 103029 | FATAL | Unable to determine the DB2 instance type | 原因: DB2 Application Recovery Kit は、DB2 インスタンスのタイプを特定できませんでした。<br><br>処置: DB2 の設定を確認してください。 |
| 103030 | FATAL | LifeKeeper has detected an error while trying to determine the node number(s) of the DB partition server(s) for the instance | 原因: DB2 Application Recovery Kit は、DB2 インスタンスのノードを見つけることができませんでした。<br><br>処置: DB2 の設定を確認してください。 |
| 103031 | ERROR | The path "%s" is not on a shared filesystem | 原因: インスタンスのホームディレクトリは、共有ファイルシステム上になければなりません。<br><br>処置: パスが共有ファイルシステム上にあることを確認し、作成操作を再試行してください。 |
| 103032 | ERROR | LifeKeeper was unable to get the DB tablespace containers for instance "%s" or the log path for one of its databases | 原因: LifeKeeper はデータベースのテーブルスペースコンテナの場所を特定できなかったか、コンテナがマウント済みのファイルシステム上のパスにあることを確認できませんでした。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。レポートされているエラーを修正してから、「create」操作を再試行してください。 |
| 103033 | ERROR | The path "%s" is not on a shared filesystem | 原因: データベースのテーブルスペースコンテナのパスは、共有ファイルシステム上になければなりません。<br><br>処置: データベースのテーブルスペースコンテナが共有ファイルシステム上にあることを確認し、操作を再試行してくだ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | さい。 |
| 103034 | ERROR | A DB2 Hierarchy already exists for instance "%s" | 原因: LifeKeeper によってすでに保護されている DB2 インスタンスを保護しようとしました。<br><br>処置: 別の DB2 インスタンスを LifeKeeper の保護用に選択する必要があります。 |
| 103035 | ERROR | The file system resource "%s" is not in-service | 原因: DB2 リソースが依存するファイルシステムは in service である必要があります。<br><br>処置: ファイルシステムリソースが in service であることを確認して、「create」操作を再試行してください。 |
| 103036 | ERROR | Unable to create the hierarchy for raw device "%s" | 原因: LifeKeeper は、リソース {raw device} を作成できませんでした。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから「create」操作を再試行する必要があります。 |
| 103037 | ERROR | A RAW hierarchy does not exist for the tag "%s" | 原因: LifeKeeper は、raw リソース {tag} を見つけられませんでした。<br><br>処置: LifeKeeper の設定を確認してください。 |
| 103038 | ERROR | LifeKeeper was unable to create a dependency between the DB2 hierarchy "%s" and the Raw hierarchy "%s" | 原因: 要求された親 DB2 リソースと子 Raw リソースとの間の依存関係の作成に失敗しました。<br><br>処置: 隣接するログメッセージで詳細お |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | よび関連メッセージを確認してください。レポートされているエラーを修正してから、「create」操作を再試行してください。 |
| 103039 | ERROR | LifeKeeper could not disable the automatic startup feature of DB2 instance "%s" | 原因: DB2 の設定を更新しようとして、予期しないエラーが発生しました。<br><br>処置: システム起動時にインスタンスが自動で起動されないようにするために、DB2AUTOSTART を手動で更新する必要があります。 |
| 103040 | ERROR | DB2 version "%s" is not installed on server "%s" | 原因: LifeKeeper は DB2 のインストール場所を見つけられませんでした。<br><br>処置: DB2 の設定を確認してください。 |
| 103041 | ERROR | The instance owner "%s" does not exist on target server "%s" | 原因:「canextend」操作または「extend」操作の間に、テンプレートサーバーから DB2 インスタンスの所有者を取得できませんでした。<br><br>処置: 指定したサーバーに DB2 インスタンスの所有者が存在していることを確認してください。所有者が存在しない場合、クラスター内のすべてのサーバーで、同じ uid および gid を使用して作成する必要があります。 |
| 103042 | ERROR | The instance owner "%s" uids are different on target server "%s" and template server "%s" | 原因: DB2 インスタンスの所有者 {user} のターゲットサーバー {target server} のユーザ ID が、テンプレートサーバー {template server} のユーザ {user} の値と一致しません。<br><br>処置: DB2 インスタンスの所有者 {user} のユーザ ID はクラスター内のすべてのサーバーで一致している必要がありま |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | す。ユーザ ID の不一致をすべてのサーバーで手動で修正し、「canextend」操作を再試行する必要があります。 |
| 103043 | ERROR | The instance owner "%s" gids are different on target server "%s" and template server "%s" | 原因:DB2 インスタンスの所有者 {user} のターゲットサーバー {target server} のグループ ID が、テンプレートサーバー {template server} のユーザ {user} の値と一致しません。<br><br>処置: DB2 インスタンスの所有者 {user} のグループ ID はクラスター内のすべてのサーバーで一致している必要があります。グループ ID の不一致をすべてのサーバーで手動で修正し、「canextend」操作を再試行する必要があります。 |
| 103044 | ERROR | The instance owner "%s" home directories are different on target server "%s" and template server "%s" | 原因: ターゲットサーバー {target server} のユーザ {user} のホームディレクトリが、テンプレートサーバー {template server} の DB2 インスタンスの所有者のホームディレクトリと一致しません。<br><br>処置: DB2 インスタンスの所有者 {user} のホームディレクトリは、クラスター内のすべてのサーバーで一致している必要があります。ディレクトリの不一致をすべてのサーバーで手動で修正し、「canextend」操作を再試行する必要があります。 |
| 103045 | ERROR | LifeKeeper was unable to get the DB2 "SVCENAME" parameter for the DB2 instance | 原因:「db2 get dbm cfg」コマンドの実行中に予期しないエラーが発生しました。<br><br>処置: DB2 の設定を確認してください。 |
| 103046 | ERROR | Unable to get the value of the DB2 "SVCENAME" parameter for the DB2 instance %s. | 原因: DB2 の「SVCENAME」パラメーターが null に設定されています。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: DB2 の設定を確認してください。 |
| 103047 | ERROR | LifeKeeper was unable to get the contents of the "/etc/services" file on the server "%s" | **原因**: テンプレートサーバーの「/etc/services」に、DB2 インスタンスのサービス名が含まれていません。<br><br>**処置**:「/etc/services」内にある DB2 インスタンスのサービス名は、クラスター内のすべてのサーバーで一致している必要があります。サービス名の不一致をすべてのサーバーで手動で修正し、「canextend」操作を再試行する必要があります。 |
| 103048 | ERROR | LifeKeeper was unable to get the contents of the "/etc/services" file on the server "%s" | **原因**: ターゲットサーバーの「/etc/services」に、DB2 インスタンスのサービス名が含まれていません。<br><br>**処置**:「/etc/services」内にある DB2 インスタンスのサービス名は、クラスター内のすべてのサーバーで一致している必要があります。サービス名の不一致をすべてのサーバーで手動で修正し、「canextend」操作を再試行する必要があります。 |
| 103049 | ERROR | The "/etc/services" entries for the instance "%s" are different on target server "%s" and template server "%s" | **原因**: インスタンスの「/etc/services」エントリに不一致があります。<br><br>**処置**:「/etc/services」内にある DB2 インスタンスのサービス名は、クラスター内のすべてのサーバーで一致している必要があります。サービス名の不一致をすべてのサーバーで手動で修正し、「canextend」操作を再試行する必要があります。 |
| 103050 | ERROR | The home directory "%s" for instance "%s" is not mounted on server "%s" | **原因**: LifeKeeper はマルチパーティションインスタンス用の db2nodes.cfg を見つけられませんでした。<br><br>**処置**: ホームディレクトリがマウントさ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | れていることを確認し、操作を再試行してください。 |
| 103051 | ERROR | Error getting resource information for resource "%s" on server "%s" | 原因: テンプレートサーバーからのリソース情報の取得に失敗しました。<br><br>処置: LifeKeeper の設定を確認してください。 |
| 103052 | ERROR | LifeKeeper was unable to add instance "%s" and/or its variables to the DB2 registry | 原因:「db2iset」コマンドの実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 103053 | ERROR | Usage: %s instance | |
| 103054 | ERROR | Unable to determine the DB2 instance type | 原因: DB2 Application Recovery Kit は、DB2 インスタンスのタイプを特定できませんでした。<br><br>処置: DB2 の設定を確認してください。 |
| 103055 | ERROR | LifeKeeper has detected an error while trying to determine the node number(s) of the DB partition server(s) for the instance | 原因: DB2 Application Recovery Kit は、DB2 インスタンスのノードを見つけることができませんでした。<br><br>処置: DB2 の設定を確認してください。 |
| 103056 | ERROR | Usage: %s instance | |
| 103058 | ERROR | Usage: %s instance | |
| 103059 | ERROR | Usage: %s instance | |
| 103060 | ERROR | Unable to determine the DB2 instance home directory | 原因: DB2 Application Recovery Kit は、DB2 インスタンスのホームディレクトリを特定できませんでした。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: インスタンスの所有者の名前がインスタンス名と同じであることを確認し、操作を再試行してください。 |
| 103061 | ERROR | Unable to determine the DB2 instance type | **原因**: DB2 Application Recovery Kit は、DB2 インスタンスのタイプを特定できませんでした。<br><br>**処置**: DB2 の設定を確認してください。 |
| 103062 | ERROR | LifeKeeper has detected an error while trying to determine the node number(s) of the DB partition server(s) for the instance | **原因**: DB2 Application Recovery Kit は、DB2 インスタンスの該当するノードを見つけることができませんでした。<br><br>**処置**: DB2 の設定を確認してください。 |
| 103063 | ERROR | Unable to determine the DB2 install path | **原因**: DB2 Application Recovery Kit は、そのインスタンスの DB2 を見つけることができませんでした。<br><br>**処置**: DB2 の設定を確認してください。 |
| 103064 | ERROR | Usage: nodes -t tag -a add_nodenum \| nodes -t tag -d delete_nodenum \| nodes -t tag -p | |
| 103065 | ERROR | Invalid input provided for "%s" utility operation, characters are not allowed. | **原因**:「nodes」コマンドで無効なパラメーターが指定されました。<br><br>**処置**: パラメーターを確認し、操作を再試行してください。 |
| 103066 | ERROR | Unable to get the information for resource "%s" | **原因**: LifeKeeper は、リソース {tag} を見つけられませんでした。<br><br>**処置**: パラメーターを確認し、操作を再試行してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 103067 | ERROR | The DB2 instance "%s" is not a EEE or Multiple Partition instance | **原因**: リソースタグ {tag} はシングルパーティションインスタンスです。<br><br>**処置**: パラメーターを確認し、操作を再試行してください。 |
| 103069 | ERROR | Node "%s" is already protected by this hierarchy | **原因**:「nodes」コマンドで無効なパラメーターが指定されました。<br><br>**処置**: パラメーターを確認し、操作を再試行してください。 |
| 103070 | ERROR | Node number "%s" is the last remaining node protected by resource "%s". Deleting all nodes is not allowed. | **原因**:「nodes」コマンドで無効なパラメーターが指定されました。<br><br>**処置**: パラメーターを確認し、操作を再試行してください。 |
| 103071 | ERROR | LifeKeeper is unable to get the equivalent instance for resource "%s" | **原因**:「nodes」コマンドの実行中に予期しないエラーが発生しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 103072 | ERROR | Unable to set NodesInfo for resource "%s" on "%s" | **原因**:「nodes」コマンドの実行中に予期しないエラーが発生しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 103073 | ERROR | Unable to set NodesInfo for resource "%s" on "%s" | **原因**:「nodes」コマンドの実行中に予期しないエラーが発生しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 103074 | ERROR | Usage: %s instance | |
| 103075 | ERROR | Usage: %s instance | |
| 103076 | ERROR | Unable to determine the DB2 instance type | 原因: DB2 Application Recovery Kit は、DB2 インスタンスのタイプを特定できませんでした。<br><br>処置: DB2 の設定を確認してください。 |
| 103077 | ERROR | Unable to determine the DB2 instance home directory | 原因: DB2 Application Recovery Kit は、DB2 インスタンスのホームディレクトリを特定できませんでした。<br><br>処置: インスタンスの所有者の名前がインスタンス名と同じであることを確認し、操作を再試行してください。 |
| 103078 | ERROR | The database server is not running for instance "%s" | 原因: DB2 インスタンスのプロセスチェックで、実行中のプロセスが見つかりませんでした。<br><br>処置: DB2 インスタンスを開始してください。 |
| 103079 | ERROR | LifeKeeper has detected an error while trying to determine the node number(s) of the DB partition server(s) for the instance | 原因: DB2 Application Recovery Kit は、DB2 インスタンスのノードを見つけることができませんでした。<br><br>処置: DB2 の設定を確認してください。 |
| 103080 | ERROR | One or more of the database partition servers for instance "%s" is down | 原因: すべてのデータベースパーティションサーバーが実行されている必要があります。<br><br>処置: すべてのデータベースパーティションサーバーが実行されていることを確認し、操作を再試行してください。 |
| 103081 | ERROR | DB2 local recovery detected another reco | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | very process in progress for "%s" and will exit. | |
| 103082 | ERROR | Failed to create flag "%s" | **原因**: DB2 ローカルリカバリー処理を制御するためのフラグを作成しようとして、予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 103083 | ERROR | Failed to remove flag "%s" | **原因**: DB2 ローカルリカバリー処理を制御するためのフラグを削除しようとして、予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 103084 | ERROR | Unable to determine the DB2 instance \"$Instance\" home directory | **原因**: DB2 Application Recovery Kit は、DB2 インスタンスのホームディレクトリを特定できませんでした。<br><br>**処置**: インスタンスの所有者の名前がインスタンス名と同じであることを確認し、操作を再試行してください。 |
| 104002 | FATAL | $msg | **原因**: このメッセージは内部ソフトウェアエラーを示します。<br><br>**処置**: スタックトレースにエラーの原因が示されます。 |
| 104003 | FATAL | $self->Val('Tag') . " is not an SDR resource"); | **原因**: 非データレプリケーションリソースに対してデータレプリケーションアクションが実行されました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決して |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ください。 |
| 104010 | ERROR | $self->{'md'}: bitmap merge failed, $action | **原因**: ビットマップのマージ操作に失敗しました。<br><br>**処置**: ターゲットサーバーにミラーや保護されたファイルシステムがマウントされている可能性があります。または、ビットマップファイルがターゲットに見つからない可能性があります。ターゲットサーバーを確認してください。 |
| 104022 | ERROR | $argv[1]: mdadm failed ($ret) | **原因**:「mdadm」コマンドによるミラーへのデバイスの追加に失敗しました。<br><br>**処置**: これは通常、一時的な状態です。 |
| 104023 | ERROR | $_); | **原因**: メッセージに「mdadm」コマンドの結果が含まれています。 |
| 104025 | ERROR | failed to spawn monitor | **原因**: システムは、「mdadm -F」監視プロセスの開始に失敗しました。これは、通常の状態では発生しません。<br><br>**処置**: システムを再起動し、競合の可能性があるものが解決されていることを確認してください。 |
| 104026 | ERROR | cannot create $md | **原因**: ミラーデバイスを作成できませんした。<br><br>**処置**: デバイスがまだ使用中ではないこと、およびミラー作成用の他の全パラメーターが正しいことを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 104027 | ERROR | $_ | **原因**: このメッセージに「mdadm」コマンドの結果が含まれています。 |
| 104035 | ERROR | Too many failures. Aborting resync of $md | **原因**: 正常範囲とは思われないほど長い時間、デバイスがビジー状態です。<br><br>**処置**: システムを再起動し、デバイスがビジーではなくなっていることを確認してください。 |
| 104036 | ERROR | Failed to start nbd-server on $target (error $port) | **原因**: ターゲットサーバーで nbd-server プロセスを開始できませんでした。<br><br>**処置**: ターゲットディスクデバイスが使用可能であること、およびデバイス ID が変更されていないことを確認してください。 |
| 104037 | ERROR | Failed to start compression (error $port) | **原因**: システムは、「バランス」トンネルプロセスを開始できませんでした。または、ネットワークに問題が発生しました。<br><br>**処置**: ネットワークが正常に実行されていること、および 10000 ~ 10512 の範囲内にある TCP ポートがオープンで未使用の状態にあることを確認してください。ソフトウェアがすべてのシステムに正常にインストールされていることを確認してください。 |
| 104038 | ERROR | Failed to start nbd-client on $source (error $ret) | **原因**: nbd-client プロセスをソースサーバーで開始できませんでした。<br><br>**処置**: レポートされている errno 値を調べ、レポートされている問題を解決してください。たとえば、errno 値 110 は「Connection timed out」（接続のタイムアウト）を示します。これは、ネットワー |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | クかファイアーウォールに問題がある可能性があります。 |
| 104039 | ERROR | Failed to add $nbd to $md on $source | **原因**: これは通常、一時的な状態です。<br><br>**処置**: このエラーが続く場合、システムを再起動し、競合の可能性があるものを解決してください。 |
| 104045 | ERROR | failed to stop $self->{'md'} | **原因**: ミラーデバイスを停止できませんした。<br><br>**処置**: デバイスがビジーでないこと、またはマウントされていないことを確認してください。「mdadm —stop」を手動で実行し、デバイスを停止してください。 |
| 104048 | WARN | failed to kill $proc, pid $pid | **原因**: プロセスをシグナル化できませんでした。これは、プロセスが終了していることを示している可能性があります。<br><br>**処置**: 問題のプロセスが実行されていないことを確認し、実行されている場合は、システムを再起動して、終了できないプロセスを消去してください。 |
| 104050 | ERROR | Setting $name on $dest failed: $ret. Please try again. | **原因**: システムは「mirrorinfo」ファイル設定を設定できませんでした。<br><br>**処置**: ネットワークとシステムを確認し、ミラー設定操作を再試行してください。 |
| 104052 | FATAL | Specified existing mount point "%s" is not mounted | **原因**: マウントポイントがアンマウント状態になりました。<br><br>**処置**: マウントポイントがマウントされていることを確認し、操作を再試行して |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ください。 |
| 104055 | ERROR | Failed to set up temporary $type access to data for $self->{'tag'}. Error: $ret | **原因**: ターゲットサーバーでファイルシステムまたはデバイスを使用できません。ミラーされたデータは、ミラーが一時停止してから再開されるまでターゲットサーバーでは使用できません。<br><br>**処置**: ターゲットサーバーを再起動し、競合の可能性があるものを解決してください。 |
| 104057 | ERROR | Failed to undo temporary access for $self->{'tag'} on $self->{'sys'}. Error: $ret. Please verify that $fsid is not mounted on server $self->{'sys'}. | **原因**: ターゲットサーバーでファイルシステムをアンマウントできませんでした。<br><br>**処置**: ファイルシステムおよびデバイスがターゲットサーバーでビジーでないことを確認してください。ターゲットサーバーを再起動し、競合の可能性があるものを解決してください。 |
| 104062 | FATAL | Cannot find a device with unique ID "%s" | **原因**: ターゲットディスクを識別できませんでした。<br><br>**処置**: 適切なストレージリカバリーキットがターゲットサーバーにインストールされていることを確認してください。ターゲットディスクのデバイス ID が変更されていないことを確認してください。 |
| 104066 | FATAL | Cannot get the hardware ID of device "%s" | **原因**: ターゲットディスクデバイスの一意の ID が見つかりませんでした。<br><br>**処置**: 適切なストレージリカバリーキットがターゲットサーバーにインストールされていることを確認してください。ターゲットディスクのデバイス ID が変更されていないことを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 104067 | FATAL | Asynchronous writes cannot be enabled without a bitmap file | **原因**: 無効なパラメーターでミラーを作成しようとしました。<br><br>**処置**: ビットマップファイルパラメーターを指定するか、同期書き込みを指定する必要があります。 |
| 104068 | FATAL | Failed to extend dependent resource %s to system %s. Error %s | **原因**: 階層の拡張操作に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 104070 | FATAL | Unable to extend the mirror "%s" to system "%s" | **原因**: 階層の拡張操作に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 104071 | ERROR | Failed to restore target device resources on $target->{'sys'} : $err | **原因**: ターゲットサーバーで in service 操作に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 104074 | FATAL | Cannot get the hardware ID of device "%s" | **原因**: ミラーに使用しようとしている基本のディスクデバイスを認識するストレージリカバリーキットがありません。<br><br>**処置**: 適切なストレージリカバリーキットがインストールされていることを確認してください。必要に応じ、/opt/LifeKeeper/subsys/scsi/resources/DEVNAME/device_pattern ファイルにデバイス名を入れてください。 |
| 104081 | FATAL | Cannot make the %s filesystem on "%s" (%d): | **原因**:「mkfs」コマンドが失敗しました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**:ディスクデバイスが書き込み可能で、エラーがないこと、および選択したファイルシステムのファイルシステムツールがインストールされていることを確認してください。 |
| 104082 | FATAL | %s | **原因**: このメッセージに「mkfs」コマンドの結果が含まれています。 |
| 104083 | FATAL | Cannot create filesys hierarchy "%s" | **原因**: リソースの作成に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 104086 | ERROR | The "%s_data_corrupt" flag is set in "%s/subsys/scsi/resources/netraid/" on system "%s". To avoid data corruption, LifeKeeper will not restore the resource. | **原因**: 予期しないデータ破損を防ぐための警告データとして data corrupt フラグファイルが設定されています。ファイルが削除されるまで、このサーバー上のミラーをリストアできません。<br><br>**処置**: 問題のサーバーでデータが有効な場合、以下のいずれかを実行できます。1) ファイルを削除してミラーをリストアしてください。2) LifeKeeper GUI または「mirror_action force」コマンドでミラーを強制的にオンラインにしてください。 |
| 104092 | ERROR | Mirror target resource movement to system %s : status %s | **原因**: 階層のスイッチオーバー操作に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 104099 | ERROR | Unable to unextend the mirror for resource "%s" from system "%s" | **原因**: 階層の拡張解除に失敗しました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: ターゲットサーバーを再起動し、競合の可能性があるものを解決して、操作を再試行してください。 |
| 104106 | ERROR | remote 'bitmap -m' command failed on $target->{'sys'}: $ranges | **原因**: ターゲットサーバーでビットマップマージコマンドが失敗しました。これには、次の 2 つの原因が考えられます。1) ビットマップファイルが見つからないか、破損している。2) ミラー（md）デバイスがターゲットでアクティブになっている。<br><br>**処置**: ミラーおよび保護されているファイルシステムがターゲットでアクティブになっていることを確認してください。ターゲットのビットマップファイルが見つからない場合、ミラーを一時停止して再開し、ビットマップファイルを再作成してください。 |
| 104107 | ERROR | Asynchronous writes cannot be enabled without a bitmap file | **原因**: ミラー作成操作で無効なパラメーターが指定されました。 |
| 104108 | ERROR | Local Partition not available | **原因**: ミラー作成操作で無効なパラメーターが指定されました。 |
| 104109 | ERROR | Cannot get the hardware ID of device "%s" | **原因**: ディスクデバイスの一意の ID を特定できませんでした。<br><br>**処置**: 適切なストレージリカバリーキットがサーバーにインストールされていることを確認してください。ディスクのデバイス ID が変更されていないことを確認してください。 |
| 104111 | FATAL | Insufficient input parameters for "%s" creation | **原因**: ミラー作成操作で無効なパラメーターが指定されました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 104112 | FATAL | Insufficient input parameters for "%s" creation | **原因**: ミラー作成操作で無効なパラメーターが指定されました。 |
| 104113 | FATAL | Insufficient input parameters for "%s" creation | **原因**: ミラー作成操作で無効なパラメーターが指定されました。 |
| 104114 | FATAL | Insufficient input parameters for "%s" creation | **原因**: ミラー作成操作で無効なパラメーターが指定されました。 |
| 104115 | FATAL | Insufficient input parameters for "%s" creation | **原因**: ミラー作成操作で無効なパラメーターが指定されました。 |
| 104117 | FATAL | Insufficient input parameters for "%s" creation | **原因**: ミラー作成操作で無効なパラメーターが指定されました。 |
| 104118 | FATAL | Cannot unmount existing Mount Point "%s" | **原因**: マウントポイントがビジーです。<br><br>**処置**: ファイルシステムがビジーではないことを確認してください。ファイルシステムにアクセスする可能性があるプロセスまたはアプリケーションを停止してください。 |
| 104119 | FATAL | Invalid data replication resource type requested ("%s") | **原因**: ミラー作成操作で無効なパラメーターが指定されました。 |
| 104124 | EMERG | WARNING: A temporary communication failure has occurred between systems %s and %s. In order to avoid data corruption, data resynchronization will not occur. MANUAL INTERVENTION IS REQUIRED. In order to initiate data resynchronization, you should take one of the following resources out of service: %s on %s or %s on %s. The resource that is taken out of service will become the mirror target. | **原因**: ソースサーバーとターゲットサーバーの間で、一時的な通信障害 (スプリットブレインシナリオ) が発生しました。<br><br>**処置**: メッセージテキストに記載されている手順を実行してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 104125 | ERROR | failed to start '$cmd $_[2] $user_args' on '$_[3]': | **原因**: 指定したコマンドが失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 104126 | ERROR | $_ | **原因**: このメッセージに、メッセージ 104125 で失敗したとレポートされているコマンドの結果が含まれています。 |
| 104128 | FATAL | comm path/server not specified | **原因**: netraid.down スクリプトが、通信パスまたはサーバー名を指定せずに呼び出されました。このスクリプトは内部で呼び出されるため、常に適切なパラメーターを使用する必要があります。<br><br>**処置**: このエラーを SIOS サポートに報告してください。 |
| 104129 | WARN | The replication connection for mirror “%s” (resource: “%s”) is down (reason: %s) | **原因**: ミラーへのレプリケーションの接続がダウン状態です。<br><br>**処置**: ネットワークを確認してください。 |
| 104130 | ERROR | Mirror resize failed on %s (%s). You must successfully complete this operation before using the mirror. Please try again. | **原因**: ミラーのサイズ変更操作で、リストされたシステム上にあるミラーのメタデータの更新に失敗しました。<br><br>**処置**: ミラーを使用するには、サイズヘンコウを正常に完了する必要があります。「mirror_resize」を再実行してください (必要に応じて、強制実行する -f を使用)。 |
| 104132 | ERROR | The partition "%s" has an odd number of sectors and system "%s&qquot; is running kernel >= 4.12. Mirrors with this configur | **原因**: ミラーの作成用に選択されたパーティションまたはディスクには奇数のデ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | ation will not work correctly with DataKeeper. Please see the SIOS product documentation for information on how to resize the mirror. | ィスクセクターがあるため、DataKeeperで使用するにはサイズを変更する必要があります。<br><br>**処置**:「parted」コマンドを使用してパーティションのサイズを変更するか、可能な場合はプラットフォーム（VMware、AWS）ツールを使用してディスクのサイズを変更してください。注記：データが失われる可能性があるため、この操作は慎重に行ってください。 |
| 104136 | ERROR | Extend failed. | **原因**: 階層の拡張操作に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 104143 | ERROR | Mirror resume was unsuccessful ($ret) | **原因**:ミラーを確立できませんした。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 104144 | ERROR | Unable to stop the mirror access for $self->{'md'} on system $self->{'sys'}. Error: $ret. Use the "mdadm —stop $self->{'md'}" command to manually stop the mirror. | **原因**: ミラーの一時停止時にターゲットノード上に作成されたミラーデバイスを停止できませんでした。<br><br>**処置**: デバイスがビジーでないこと、またはマウントされていないことを確認してください。「mdadm —stop」を手動で実行し、デバイスを停止してください。 |
| 104145 | WARN | Unable to dirty full bitmap.  Setting fullsync flag. | **原因**: ビットマップ全体をダーティにすることによる完全な再同期を実行できませんでした。代わりに fullsync フラグが使用されます。完全な同期は実行されるので、これは致命的なエラーではありません。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: ありません |
| 104146 | EMERG | WARNING: The target system %s for mirror %s has the target mirror %s currently active. In order to avoid data corruption, data resynchronization will not occur. MANUAL INTERVENTION IS REQUIRED. In order to initiate data resynchronization, you should reboot system %s. | **原因**: ミラーがターゲットシステムで構成されています。<br><br>**処置**: ターゲットシステムを再起動する必要があります。 その後、DataKeeperはミラーを再同期できるようになります。 |
| 104147 | EMERG | WARNING: The target system %s for mirror %s has the target disk %s currently mounted. In order to avoid data corruption, data resynchronization will not occur. MANUAL INTERVENTION IS REQUIRED. In order to initiate data resynchronization, you should unmount %s on %s. A full resync will occur. | **原因**: ミラーディスクがターゲットシステムにマウントされています。<br><br>**処置**: 完全なミラー再同期を開始するには、ミラーディスクをターゲットシステムでアンマウントする必要があります。ディスクで追跡されていない変更が発生したため、完全な再同期が必要です。 |
| 104148 | EMERG | The storage configuration for mirror "%s (%s)" does not have a unique identifier and may have potential for data corruption in some environments in certain circumstances. Please refer to the SIOS Product Documentation for details on DataKeeper storage configuration options. | **原因**: ミラーリング用に選択されたディスクが、オペレーティングシステムにUUIDを提供していません。DataKeeperはUUIDがないとディスクをミラーできません。<br><br>**処置**: ディスク上にGPTパーティションテーブルを作成して、ディスクパーティションにUUIDを提供できます。 DataKeeper for Linux トラブルシューティング を参照してください。 |
| 104155 | EMERG | The mirror %s cannot be forced online at this time. The underlying disk %s is mounted, indicating possible data corruption. MANUAL INTERVENTION IS REQUIRED. You must unmount %s on %s and restore the mirror to the last known mirror source system. A full resync will need to be performed from the source system to %s. | **原因**: ターゲットシステムでミラーディスクがマウントされた。<br><br>**処置**: データ破損を防ぐため、すぐにディスクをアンマウントしてください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 104156 | WARN | Resynchronization of "%s" is in PENDING state. Current sync_action is: "%s" | 原因: md デバイスの再同期がペンディング状態であることが検出されました。<br><br>処置: LifeKeeper は再同期を強制実行して問題の修正を試行してください。関連するエラーをログで確認してください。エラーを解決した後、/proc/mdstat でペンディング状態がクリアされたこと、および datarep リソースの再同期が進行中であるか、完了したことを確認してください。 |
| 104157 | WARN | /etc/sysconfig/raid-check update failed. Please %s "md%d" to SKIP_DEVS. | 原因: /etc/sysconfig/raid-check を変更して、スキップする MD デバイスのリスト（SKIP_DEVS）にエントリを追加または削除することができません。<br><br>処置: raid-check または SKIP_DEVS に関連するエラーがないか、システムログを確認してください。リストされたmd を手動で追加または削除してください。 |
| 104158 | EMERG | WARNING: The local disk partition $self->{ part } for data replication device\n$self->{ md } has failed. MANUAL INTERVENTION IS REQUIRED. | 原因: ミラーのローカルデバイスで問題が発生しました。LKDR_FAILURE でリカバリーアクションが設定されていないため、復旧するには手動による介入が必要です。<br><br>処置: システムログと LifeKeeper ログで、ローカルディスクに関連するエラーを確認してください。 |
| 104163 | WARN | The "%s_data_corrupt" flag is set in "%s/subsys/scsi/resources/netraid/" on system "%s". The mirror is being forced online. | 原因: ミラーは強制的にオンラインになり、data_corrupt フラグを上書きします。指定されたシステム上のデータは、最新のデータとして扱われます。これが正しくない場合、データ破損またはデータ損失につながる可能性があります。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: ありません。 |
| 104164 | ERROR | The "%s_data_corrupt" flag for related mirror resource "%s" is set in "%s/subsys/scsi/resources/netraid/" on system "%s". To avoid data corruption, LifeKeeper will not restore this mirror or any related mirrors in the hierarchy. | **原因**: data_corrupt フラグは、階層内の1つ以上のミラーに存在します。追加データの破損を防ぐため、すべての data_corrupt フラグが解決されるまで、どのミラーも In Service になりません。<br><br>**処置**: LifeKeeperのログを確認して、各ミラーが最後に In Service になっていた場所、つまり各ミラーの最新データが存在する場所を特定してください。すべての階層が In Service であった以前のソースでミラーを In Service にし、ミラーがすべてのターゲットと同期できるようにする必要があります。 |
| 104165 | ERROR | The "%s_data_corrupt" flag for related mirror resource "%s" is set in "%s/subsys/scsi/resources/netraid/" on system "%s". The mirror resource "%s" is being forced online. | **原因**: ミラーは強制的にオンラインになっており、data_corrupt フラグを上書きします。指定されたシステム上のデータは、すべてのターゲットと同期される正しいデータとして扱われます。これが最新のデータでない場合、データ破損またはデータ損失につながる可能性があります。<br><br>**処置**: ありません。 |
| 104170 | ERROR | Failed to create "source" flag file on shared source %s to track mirror source. This may result in a full resync. | **原因**:「ソース」フラグファイルが、ミラーソースを追跡するために指定されたシステムで作成されませんでした。<br><br>**処置**: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104171 | ERROR | Failed to create "source" flag file on %s to track mirror source. Target %s will not be added to mirror. | **原因**:「ソース」フラグファイルが、ミラーソースを追跡するために指定されたシ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ステムで作成されませんでした。<br><br>**処置**: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104172 | ERROR | The "source" flag file on %s does not contain a valid target (%s). Full resync to remaining targets is required. | **原因**:「ソース」フラグファイルには以前のソースのシステム名が含まれている必要がありますが、リストされている名前は構成されたシステムのリストにありませんでした。<br><br>**処置**: この問題を SIOS サポートに報告してください。 |
| 104173 | ERROR | Failed to create "source" flag file on %s to track mirror source. | **原因**:「ソース」フラグファイルが、ミラーソースを追跡するために指定されたシステムで作成されませんでした。<br><br>**処置**: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104174 | ERROR | Failed to create “previous_source” flag file to track time waiting on source. Will not be able to timeout. | **原因**:「previous_source」フラグファイルは、ミラーの以前のソースを追跡するためのミラーソース上に作成されませんでした。<br><br>**処置**: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104175 | ERROR | Failed to create "data_corrupt" flag file on target "%s". | **原因**: リストされたターゲットに「data_corrupt」フラグファイルが作成されませんでした。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104176 | ERROR | The "source" flag file on %s to track mirror source does not exist. Full resync is required. | **原因**:「ソース」フラグファイルはシステム上に存在する必要があり、これがないとミラーの整合性を検証できません。データの信頼性を確保するには、完全な再同期が必要です。<br><br>**処置**: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104177 | ERROR | Failed to determine amount of time waiting on %s. | **原因**: 以前のソースを待機する時間を決定できませんでした。以前のソースが見つからない場合、ターゲットは完全に再同期されて追加されます。<br><br>**処置**: ありません |
| 104178 | ERROR | Failed to update "source" flag file on target "%s", previous source must be merged first. | **原因**: ターゲットのソースフラグファイルは、同期されて停止されると更新されるため、次のIn Service時には以前のソースは必要ありません。<br><br>**処置**: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104180 | ERROR | Internal Error: "previous_source" has the local system name (%s). | **原因**: ローカルシステム名は、previous_source フラグファイルに含めることはできません。<br><br>**処置**: このエラーを SIOS サポートに報告してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 104181 | ERROR | Internal Error: There are no targets waiting on %s to be merged. | **原因**: 以前のソースがマージされるのを待機しているターゲットがありません。<br><br>**処置**: このエラーを SIOS サポートに報告してください。 |
| 104182 | ERROR | Failed to create \"source\" flag file on %s to track mirror source. This may result in a full resync. | **原因**: リストされたターゲットに「ソース」フラグファイルが作成されませんでした。<br><br>**処置**: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104186 | ERROR | Failed to create "last_owner" flag file on %s to track mirror source. This may allow in-service of mirror on old data. | **原因**:「last_owner」フラグファイルがソースに作成されませんでした。<br><br>**処置**: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104187 | WARN | $REM_MACH has ${REM_TAG}_last_owner file, create flag ${FLAGTAG}_data_corrupt. | **原因**: リストされているシステムでは、ミラーが最後にIn Serviceになりました。<br><br>**処置**: リストされているシステムには last_owner ファイルがあります。これは、このシステムには最新のデータがあり、データの損失を防ぐためにミラーをIn Serviceにするのに最適なシステムである可能性が高いことを示しています。 |
| 104188 | WARN | $REM_MACH is not alive, create flag ${FLAGTAG}_data_corrupt. | **原因**: リストされているシステムは稼働していません。<br><br>**処置**: リストされているシステムは稼働していないため、そのシステムがローカ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ルシステムよりも新しいミラーソースであったかどうかを判断できません。このため、ローカルシステムがミラーをIn Serviceにすることを自動的に許可しないでください。 |
| 104190 | ERROR | [$SUBJECT event] mail returned $err | 原因: DataKeeper の再同期が完了したときに通知するメールが失敗しました。このメッセージに続いて、エラーの詳細が表示されます。 |
| 104191 | ERROR | COMMAND OUTPUT: cat /tmp/err$$ | 原因: DataKeeper の再同期が完了したときに通知するメールが失敗しました。このメッセージに続いて、エラーの詳細が表示されます。 |
| 104193 | ERROR | snmptrap returned $err for Trap 143 | 原因: DataKeeper の再同期が完了したときに通知する SNMP トラップ 143 が失敗しました。 このメッセージに続いて、エラーの詳細が表示されます。 |
| 104194 | ERROR | COMMAND OUTPUT: cat /tmp/err$$ | 原因: DataKeeper の再同期が完了したときに通知する SNMP トラップ 143 が失敗しました。 このメッセージに続いて、エラーの詳細が表示されます。 |
| 104196 | ERROR | $msg | 原因: カーネルは非同期レプリケーションをサポートしていません。 メッセージには、失敗の詳細が記載されています。<br><br>処置: メッセージの詳細に従って、このカーネルを使用している間は非同期レプリケーションを構成しないでください。 |
| 104197 | WARN | $msg | 原因: is_async_supported は、非同期でレプリケートするときに注意すべきいく |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | つかの条件を指定する警告メッセージを返しました。<br><br>**処置**: 警告の条件がこの構成に適用されないことを確認してください。 同期レプリケーションが必要な場合があります。 |
| 104200 | EMERG | Continue to wait for %s to merge bitmap and do partial resyncs to all targets, no timeout set. | **原因**: マルチターゲット構成では、すべてのターゲットが部分的に再同期できるように、以前のソースがビットマップをマージできるようになるまでターゲットは構成されません。/etc/defaults/LifeKeeper の LKDR_WAIT_ON_PREVIOUS_SOURCE_TIMEOUT エントリは、無期限に待機するために「-1」に設定されます。<br><br>**処置**: メッセージにリストされている以前のソースのステータスを確認し、クラスターへの再参加を妨げている問題を解決してください。 |
| 104201 | EMERG | To stop waiting for the previous source (forcing a full resync to remaining waiting targets) run: "%s/bin/mirror_action %s fullresync %s %s" on %s. | **原因**: マルチターゲット構成では、ターゲットは構成されず、以前のソースがクラスターに再参加するのを待機します。<br><br>**処置**: メッセージにリストされているコマンドを実行して、このターゲットおよび再同期を待機している残りのターゲットに対して即時の完全再同期を強制してください。 |
| 104202 | EMERG | Continue to wait for %s to merge bitmap and do partial resyncs to all targets. Continue to wait %d more seconds. | **原因**: マルチターゲット構成では、すべてのターゲットが部分的に再同期できるように、前のソースがビットマップをマージできるようになるまでターゲットは構成されません。 /etc/defaults/LifeKeeper の LKDR_WAIT_ON_PREVIOUS_SOURCE_TIMEOUT エントリは、待機す |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | る秒数に設定されます。以前のソースがその時間内にクラスターに参加しない場合、ターゲットは完全に再同期されて追加されます。<br><br>**処置**: メッセージにリストされている以前のソースのステータスを確認し、クラスターへの再参加を妨げている問題を解決してください。 |
| 104203 | EMERG | To stop waiting for the previous source (forcing a full resync to remaining waiting targets) run: \"%s/bin/mirror_action %s fullresync %s %s\" on %s. | **原因**: マルチターゲット構成では、ターゲットは構成されず、以前のソースがクラスターに再参加するのを待機してください。<br><br>**処置**: メッセージにリストされているコマンドを実行して、このターゲットおよび再同期を待機している残りのターゲットに対して即時の完全再同期を強制してください。<br><br>**注記**: リストされたターゲットが削除されて戻らない場合でも、このコマンドを実行して待機を停止してください。 |
| 104207 | ERROR | Failed to create "data_corrupt" flag file on "%s". | **原因**: リストされたソースに「data_corrupt」フラグファイルが作成されませんでした。<br><br>**処置**: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104208 | ERROR | Failed to create "data_corrupt" flag file on target "%s". | **原因**: リストされたターゲットに「data_corrupt」フラグファイルが作成されませんでした。<br><br>**処置**: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | Full になっていないか確認してください。 |
| 104209 | ERROR | Failed to create "data_corrupt" flag file on "%s". | 原因: リストされたソースに「data_corrupt」フラグファイルが作成されませんでした。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104210 | ERROR | Failed to create "data_corrupt" flag file on target "%s". | 原因: リストされたターゲットに「data_corrupt」フラグファイルが作成されませんでした。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104211 | ERROR | Failed to create "data_corrupt" flag file on target "%s". | 原因: リストされたターゲットに「data_corrupt」フラグファイルが作成されませんでした。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104212 | ERROR | The "source" flag file on %s to track mirror source does not exist. Full resync to remaining targets is required. | 原因:「ソース」フラグファイルはシステム上に存在する必要があり、これがないとミラーの整合性を検証できません。データの信頼性を確保するには、完全な再同期が必要です。まだミラーされていないすべてのターゲットでは、完全な再同期が必要です。<br><br>処置: LKROOT（/opt/LifeKeeper）ファ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | イルシステムにエラーがないか、または Full になっていないか確認してください. |
| 104214 | ERROR | Failed to create "source" flag file on %s to track mirror source. | 原因:「ソース」フラグファイルが、ミラーソースを追跡するために指定されたシステムで作成されませんでした。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104216 | ERROR | Failed to create "data_corrupt" flag file on target "%s". | 原因: リストされたターゲットに「data_corrupt」フラグファイルが作成されませんでした。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104217 | ERROR | Failed to create "source" flag file on shared source %s to track mirror source. This may result in a full resync. | 原因:「ソース」フラグファイルが、ミラーソースを追跡するために指定されたシステムで作成されませんでした。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104218 | ERROR | Failed to create "data_corrupt" flag file on target "%s". | 原因: リストされたターゲットに「data_corrupt」フラグファイルが作成されませんでした。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 104221 | ERROR | Failed to create "last_owner" flag file on %s to track mirror source. This may allow in-service of mirror on old data. | 原因: リストされたターゲットに「last_owner」フラグファイルが作成されませんでした。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104222 | ERROR | Failed to create "last_owner" flag file to track mirror source". This may allow in-service of mirror on old data. | 原因:「last_owner」フラグファイルは、ミラーが最後にIn Serviceになった場所を特定するために使用されています。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104223 | ERROR | Failed to create "last_owner" flag file to track mirror source". This may allow in-service of mirror on old data. | 原因:「last_owner」フラグファイルは、ミラーが最後にIn Serviceになった場所を特定するために使用されています。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104224 | ERROR | Failed to create "previous_source" flag file. | 原因:「previous_source」フラグファイルが作成されませんでした。これは、以前のソースのビットマップをマージして完全な同期を回避するために必要です。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 104227 | ERROR | Failed to set %s to %s. | 原因: このメッセージは、nbdドライバー（/sys/block/nbdX）のsysfsパラメーターの設定に失敗したことを示しています。<br><br>処置: このエラーを回避するには、/etc/default/LifeKeeper で次のうち1つ以上を調整する必要がある場合があります。 NBD_NR_REQUESTS NBD_SCHEDULER LKDR_ASYNC_LIMIT |
| 104232 | ERROR | Mirror resize failed on %s (%s). Could not set size to %d. | 原因: ミラーのサイズ変更操作に失敗しました。 |
| 104233 | ERROR | Mirror resize failed on %s (%s). Could not set bitmap to %s and bitmap-chunk to %d. | 原因: ミラーのサイズ変更操作に失敗しました。 |
| 104234 | ERROR | The mirror %s failed to resize. You must successfully complete this operation before using the mirror. Please try again. | 原因: ミラーのサイズ変更操作に失敗しました。 |
| 104235 | ERROR | mirror_resize of mirror %s failed due to signal "%s". | 原因: ミラーのサイズ変更操作に失敗しました。 |
| 104236 | EMERG | Resource "%s" is "OSF". The mirror "%s" will wait to replicate data until all resources in the hierarchy%s are in-service. This may indicate inconsistent data. Verify the data is correct before replicating data; replication will continue when all resources in the hierarchy%s are in-service. A full resync may be necessary (see "LKDR_WAIT_TO_RESYNC" in /etc/default/LifeKeeper). | |
| 104237 | WARN | Resource "%s" is "OSU". The mirror "%s" will wait to replicate data until all resources in the hierarchy%s are in-service. To replicate data immediately run: "%s/bin/mirror_action %s resume" on "%s" (see "LKDR_WAIT_TO_RESYNC" in /etc/default/Lif | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | eKeeper). | |
| 104238 | ERROR | Unable to read /sys/module/nbd/taint. Assuming that the SIOS nbd module is not loaded. | |
| 104239 | EMERG | $nbd_failure_msg | |
| 104242 | ERROR | Unable to read /proc/modules. Assuming that the SIOS nbd module is not loaded. | |
| 104243 | ERROR | Internal script or routine $caller was called for unsupported kernel module '$module'. Supported kernel modules for this script or routine are: $module_list. | **原因**: 指定されたスクリプトまたはルーチンが、サポートされていないカーネルモジュールに対して呼び出されました。<br><br>**対応**: これは内部エラーです。SIOSカスタマーサポートに連絡してください。 |
| 104244 | ERROR | Output from 'modprobe $module' command: $pretty_modprobe_out | **原因**: modprobeコマンドを使用して指定されたカーネルモジュールをロードできませんでした。<br><br>**対応**:失敗したmodprobeのログメッセージの出力を確認し、見つかった問題を解決してください。 |
| 104251 | ERROR | There is no LifeKeeper protected resource with tag $tag on system $me. | **原因**: 指定されたタグは、指定されたシステム上のLifeKeeperで保護されているリソースに対応していません。<br><br>**処置**: リソースタグとシステム名が正しいことを確認してください。 |
| 104252 | ERROR | Resource $tag is not a $app/$typ resource. Please use the $ins_app/$ins_typ resource-specific canfailover script instead. | **原因**: scsi/netraid 固有の canfailover スクリプトが、非 scsi/netraid リソースに対して呼び出されました。<br><br>**処置**: 指定されたリソースの適切なアプリケーションとタイプに対応する canfailover スクリプトを使用してください（存在する場合）。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 107015 | ERROR | Exported file system $opt_t cannot be accessed on $me. | |
| 112000 | FATAL | Usage: "%s %s". Specify the correct usage for the requested command. | 原因: SAP HANA createスクリプトに無効なパラメーターが指定された。<br><br>処置: 適切なパラメーターをSAP HANA createスクリプトに指定してください。 |
| 112004 | ERROR | Neither the SID/instance pair nor the tag parameter were specified for the internal "%s" routine on %s. If this was a command line operation, specify the correct parameters. Otherwise, consult the troubleshooting documentation. | 原因: 内部SAPオブジェクトの作成において無効なパラメーターが指定された。<br><br>処置: SIDおよびインスタンスまたはLifeKeeper のリソースタグ名のいずれかを指定してください。 |
| 112013 | ERROR | The file "%s" does not exist or was not readable on %s. Verify that the specified file exists and/or is readable. | 原因: sapservices ファイルが存在しないか、読み込み不可。<br><br>処置: sapservices ファイルが存在し、読み込み可能であることを確認してください。 |
| 112017 | ERROR | Error getting resource information for resource "%s" on server "%s" | 原因: 指定したサーバー上で指定したリソースに関連する情報を見つけることができない。<br><br>処置: 指定したサーバーに指定したリソースが存在していて、すべての必要なファイルシステムがマウントされているかを確認してください。 |
| 112019 | ERROR | The attempt to update the resource information field for resource %s has failed on %s. View the resource properties manually using "ins_list -t <tag>" to verify that the resource is functional. | 原因: 指定したサーバーで指定したリソースの LifeKeeper リソース情報フィールドを更新できない。<br><br>処置: 指定したサーバーにリソースが存在しているか、もしくはLifeKeeper が起動しているかを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 112022 | ERROR | An error occurred while trying to find the IP address corresponding to "%s" on %s. Verify the IP address or host name exists in DNS or the hosts file. | **原因**: 指定したサーバーにIPアドレスもしくはホスト名が見つからない。<br><br>**処置**: IPアドレスもしくはDNS名がDNSもしくはローカルホストファイルに存在しているかを確認してください。 |
| 112024 | FATAL | There was an error verifying the NFS connections for SAP related mount points on $me. One or more NFS servers is not operational and needs to be restarted. | **原因**: SAP_NFS_CHECK_DIRS パラメーター に掲載されている少なくとも一つの NFS 共有ファイルシステムを現在使用できない。<br><br>**処置**: NFSサーバーの起動、すべての必要とされるNFS関連のサービスの起動、すべての必要とされるファイルシステムがエクスポートされていることを確認してください。 |
| 112028 | ERROR | Unable to determine the user name for the SAP administrative user for resource %s on %s. | **原因**: 指定したサーバー上に指定したリソース情報を確認することができない。<br><br>**処置**: 指定したサーバーに指定したリソースが存在していて、すべての必要なファイルシステムがマウントされているかを確認してください。 |
| 112029 | ERROR | The canextend script "%s" either does not exist or is not executable on %s. | **原因**: SAP canextend スクリプトが存在していないか、実行できない。<br><br>**処置**: SAP canextend スクリプトが存在していて、実行可能か確認してください。 |
| 112037 | ERROR | Unable to create an internal object for the SAP instance using SID %s, instance %s, and tag %s on server %s. Verify that all necessary SAP file systems are mounted and accessible before reattempting the operation. | **原因**: 指定した SAP インスタンスを表すための内部の SAP オブジェクトを作成することができない。<br><br>**処置**: SAP インスタンスが正常にインストール、設定されていて、すべての必要 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | なファイルシステムがマウントされているかを確認してください。 |
| 112040 | ERROR | The SAP Directory "%s" ("%s") does not exist on %s. Verify that the directory exists and that the SAP software is properly installed. | 原因: 指定したSAP インストールディレクトリが指定したサーバーに存在していない。<br><br>処置: SAP ソフトウェアが正常にインストールされすべての必要となるファイルシステムがマウントされていることを確認してください。 |
| 112041 | ERROR | The required utility "%s" was not found or was not executable on %s. Verify the SAP installation and location of the required utility. | 原因: saphostexec または saposcol が見つからないか実行できない。<br><br>処置: SAP Host Agent package が正しくインストールされ、すべての必要となるファイルシステムがマウントされていることを確認してください。 |
| 112042 | ERROR | One or more SAP or LifeKeeper validation checks has failed on %s. Please update the SAP software on this host to include the SAPHOST and SAPCONTROL packages. | 原因: saphostexec または saposcol が見つからないか実行できない。<br><br>処置: SAP Host Agent package が正しくインストールされ、すべての必要となるファイルシステムがマウントされていることを確認してください。 |
| 112048 | ERROR | The SAP instance %s is already under LifeKeeper protection on server %s. Choose another SAP instance to protect or specify the correct instance. | 原因: 指定した SAP インスタンスはすでに指定したサーバーでLifeKeeper により保護されている。<br><br>処置: まだLifeKeeper により保護されていない SAP インスタンスを選択してください。 |
| 112049 | ERROR | Unable to locate the SAP Mount directory on %s. Verify that all SAP file systems are mounted and accessible before reattempti | 原因: SAP のマウント (sapmnt) ディレクトリの場所を特定できない。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | ng the operation. | **処置**: すべての必要となるファイルシステムがマウントされ、すべての必要となる SAP インスタンスプロファイルがアクセス可能になっていることを確認してください。 |
| 112050 | ERROR | Detected multiple virtual IP addresses/host names for instance %s on %s. Verify that the instance is configured correctly. | **原因**: 指定したサーバー上に指定した SAP インスタンスに対する複数の仮想 IP もしくはホスト名が検出された。<br><br>**処置**: インスタンスに関連する仮想 IP もしくはホスト名が正しく構成されているかを確認してください。 |
| 112051 | ERROR | The "%s" or "%s" value in the default profile "%s" is still set to the physical host name on %s. The value(s) must be set to a virtual host name. | **原因**: 指定したサーバーで指定したホスト名のパラメーターが物理サーバーのホスト名に指定されている。<br><br>**処置**: ホスト名のパラメーターは仮想ホスト名に設定してください。 |
| 112053 | ERROR | Detected multiple instances under SID %s with the same instance number (%s) on %s. Each instance within a particular SID must have a unique instance number. | **原因**: 同じインスタンス番号のSAP インスタンスが、同じ SAP SID 下に検出された。<br><br>**処置**: SAP環境を再構築することで、指定した SAP SID 下の各インスタンスが一意のインスタンス番号をもつことができます。 |
| 112056 | ERROR | The NFS export for the path "%s" required by the instance %s for the "%s" directory does not have an NFS hierarchy protecting it on %s. You must create an NFS hierarchy to protect this NFS export before creating the SAP resource hierarchy. | **原因**: 指定したファイルシステムのNFS エクスポートが LifeKeeper により保護されていません。<br><br>**処置**: エクスポートしたファイルシステムの LifeKeeper NFS 階層を作成し、SAP リソースを作成してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 112057 | ERROR | Unable to create a file system resource hierarchy for the file system “%s” on “%s”. | **原因**: 指定したサーバーで指定したファイルシステムを保護するための LifeKeeper ファイルシステムリソース階層を作成することができない。<br><br>**処置**: 詳細について LifeKeeper とシステムログを確認してください。 |
| 112058 | ERROR | Unable to create a dependency between parent tag “%s” and child tag “%s” on “%s”. | **原因**: 指定したサーバーで指定したリソース間での依存関係を作成できない。<br><br>**処置**: 詳細について LifeKeeper のログを確認してください。 |
| 112060 | ERROR | All attempts at local recovery for the SAP resource “%s” have failed on “%s”. A failover to the backup server will be attempted. | **原因**: 指定したサーバーで指定した SAP リソースを復旧できない。<br><br>**処置**: SAP リソース階層のフェイルオーバーは自動的に行われます。 ユーザーの介入は必要ありません。 |
| 112061 | ERROR | The values specified for the target and the template servers are the same. Please specify the correct values for the target and template servers. | **原因**: SAP リソース階層の拡張中に提供されたテンプレート/ターゲットサーバーが同一。<br><br>**処置**: テンプレート・ターゲットサーバーの正しい名前を提供し、拡張処理を再度実行してください。 |
| 112062 | ERROR | Unable to find the home directory "%s" for the SAP administrative user "%s" on %s. Verify that the SAP software is installed correctly. | **原因**: 指定したサーバーで指定した SAP ユーザーのホームディレクトリが見つからない。<br><br>**処置**: SAP ソフトウェアが正常にインストールされ、指定した SID に対する適切な SAP 管理ユーザーがサーバーに存在していることを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 112063 | ERROR | The SAP administrative user "%s" does not exist on %s. Verify that the SAP software is installed correctly or create the required SAP user on %s. | 原因: 指定した SAP 管理ユーザーが指定したサーバー上に存在していない。<br><br>処置: SAP ソフトウェアが正常にインストールされているか確認し、必要に応じて SAP 管理ユーザーを作成してください。 |
| 112064 | ERROR | The group ID for user "%s" is not the same on template server "%s" and target server "%s". Please correct the group ID for the user so that it is the same on the template and target servers. | 原因: 指定したSAP 管理ユーザーのグループ IDがテンプレート/ターゲットサーバーで一致していない。<br><br>処置: 指定した SAP 管理ユーザーのグループ ID を修正することで、テンプレート/ターゲットサーバーで同一となります。 |
| 112065 | ERROR | The user ID for user "%s" is not the same on template server "%s" and target server "%s". Please correct the user ID so that it is the same on the template and target servers. | 原因: 指定した SAP 管理ユーザーの ID がテンプレート/ターゲットサーバーで一致していない。<br><br>処置: 指定した SAP 管理ユーザーの ID を修正することで、テンプレート/ターゲットサーバーで同一となります。 |
| 112066 | ERROR | Required SAP utilities could not be found in "%s" on %s. Verify that the SAP software is installed correctly. | 原因: 必要な SAP 実行ファイルもしくは SAP インスタンスプロファイルを配置することができない。<br><br>処置: SAP ソフトウェアが正常にインストールされ構成されていて、すべての必要なファイルシステムがマウントされているか確認してください。 |
| 112069 | ERROR | The command “%s” is not found in the “%s” perl module (“%s”) on “%s”. Please check the command specified and retry the operation. | 原因: 指定したコマンドが指定したサーバーの sap perl modedule に見つからない。<br><br>処置: ユーザーから開始したコマンドア |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | クションによりエラーとなった場合は、正しいルーチン名が remoteControl スクリプトに渡されているかを確認してください。通常のLifeKeeper 運用において発生したエラーの場合は、製品サポートへお問い合わせください。 |
| 112071 | ERROR | The file “%s” exists, but was not read and write enabled on “%s”. Enable read and write permissions on the specified file | **原因**: 指定したファイルが指定したサーバーで読み込み/書き込み権限をもっていない。<br><br>**処置**: 指定したサーバーで読み込み/書き込み権限を有効にしてください。 |
| 112073 | ERROR | Unable to create an internal object for the SAP instance using either SID "%s" and instance "%s" or tag "%s" on server "%s". Either the values specified for the object initialization (SID/instance pair or tag, system) were not valid, or an error occurred while attempting to gather information about the SAP instance. If all specified parameters are correct, verify that all necessary SAP file systems are mounted and accessible before reattempting the operation. | **原因**: 指定した SAP インスタンスを示すための内部 SAP オブジェクトを作成することができない。<br><br>**処置**: SAP インスタンスが正常にインストール、設定されていて、すべての必要なファイルシステムがマウントされているかを確認してください。 |
| 112074 | WARN | Warning: The profile file %s for SID %s and Instance %s has Autostart is enabled on %s. Disable Autostart for the specified instance by setting Autostart=0 in the profile file | **原因**: Autostart パラメーターが指定したサーバーの指定したインスタンスプロファイルで有効になっている。<br><br>**処置**: 指定した SAP インスタンスがインスタンスプロファイルの中で ‘Autostart = 0’に設定されており、Autostartが有効になっています。Autostartを無効にしてください。 |
| 112076 | FATAL | Unable to start the sapstartsrv service for SAP SID $sid and SAP instance $Instance on $me. Verify that the sapservices file is correct and the process can be started manually. | **原因**: 指定した SAP インスタンスに対して SAP Start Service (sapstartsrv) のプロセスを開始できない。<br><br>**処置**: sapservices ファイルがsapstartsr |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | v プロセスを開始するための適切なコマンドを含んでいて、そのプロセスを手動で開始できるかを確認してください。 |
| 112077 | ERROR | Unable to stop the sapstartsrv service for SAP SID %s and SAP instance %s on %s. Verify that the sapservices file is correct and the process can be stopped manually. | 原因: 指定した SAP インスタンスに対して SAP Start Service (sapstartsrv) プロセスを停止することができない。<br><br>処置: sapservices ファイルが sapstartsrv プロセスを開始するための適切なコマンドを含んでいて、そのプロセスを手動で開始できるかを確認してください。 |
| 112078 | ERROR | ERSv1 is only supported in two-node clusters. Resource %s is unable to be extended to system %s. Upgrade to ERSv2 in order to extend the hierarchy to three or more nodes. | 原因: ERSv1インスタンスを表す SAP リソースを 3つ以上のノードに拡張することができない。<br><br>処置: ERS インスタンスを表す SAP リソースを3つ以上のノードに拡張する場合は、ERSv2 へアップグレードしてください。 アップグレード手順は、オンラインプロダクトドキュメント内に記載されています。 |
| 112082 | WARN | Instance %s is running a different version of the enqueue server than its corresponding enqueue replication server. This configuration is not supported by SAP and will lead to unexpected resource behavior. See SAP Note 2711036 – "Usage of the Standalone Enqueue Server 2 in an HA Environment" for more details. Please review the online product documentation for instructions on how to modify the instance profiles for the enqueue server and enqueue replication server so that they use the same version. | 原因: エンキューサーバーとエンキューレプリケーションサーバーのバージョンが一致していない。<br><br>処置: オンラインプロダクトドキュメントに記載のインスタンスプロファイルの修正方法を確認の上、エンキューサーバーとエンキューレプリケーションサーバーのバージョンを一致させてください。 |
| 112086 | ERROR | The ERS resource corresponding to resource %s is in-service and maintaining backup locks on a remote system. Bringing resource %s in-service on %s would result i | 原因: ASCS/SCS リソースに対応する ERS インスタンスがクラスター内の異なるサーバーで起動し、バックアップのエ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | n a loss of the backup lock table. Please bring resource %s in-service on the system where the corresponding ERS resource is currently in-service in order to maintain consistency of the lock table. In order to force resource %s in-service on %s, either (i) run the command \'/opt/LifeKeeper/bin/flg_create -f sap_cs_force_restore_%s\' as root on %s and reattempt the in-service operation or (ii) take the corresponding ERS resource out of service on the remote system. Both of these actions will result in a loss of the backup lock table. | ンキューテーブルを保存していたので、ASCS/SCS リソースに対して in-service 処理が試された。<br><br>**処置**: バックアップのエンキューテーブルを取得するためにERSインスタンスが起動しているサーバーで、ASCS/SCS リソースを in-sevice にしてください。ASCS/SCS リソースを強制的に in-service にする場合は、 '/opt/LifeKeeper/bin/flg_create -f sap_cs_force_restore_<ASCS/SCS Tag>' コマンドを実行し in-service 処理を試みるか、またはリモートサーバーで対応する ERS リソースを out of service にしてください。これら両方の操作を実行すると、バックアップのエンキューテーブルが失われます。 |
| 112089 | ERROR | The internal "%s" routine was called for a resource with unsupported instance type %s. This method supports only SAP instance types TYPE_CS, TYPE_ERS, or TYPE_NEW_ERS (1, 2, or 5). | **原因**: GetEnqVersion ルーチンがサポートされていない SAP インスタンスのタイプに対して呼び出された。インスタンスタイプ 1 (TYPE_CS), 2 (TYPE_ERS), 5 (TYPE_NEW_ERS) のみがサポート対象です。<br><br>**処置**: こちらは内部エラーです。SIOS カスタマーセンターへお問い合わせください。 |
| 112092 | ERROR | The profile "%s" either does not exist or cannot be read on %s. Unable to determine whether enqueue replication is enabled for resource %s. Please verify that the file exists and can be read. | **原因**: 指定したサーバー上に指定したインスタンスプロファイルが存在していない、もしくは読み取りができない。<br><br>**処置**: ファイルが存在していて、読み取り可能かどうかを確認してください。 |
| 112095 | ERROR | Error creating resource “%s” on server “%s” | **原因**: 指定したサーバーに指定したリソースを作成することができない。<br><br>**処置**: 指定したサーバーにSAP ソフトウ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ェアが正常にインストールされ、すべての必要なファイルシステムがマウントされアクセス可能になっていることを確認してください。 |
| 112096 | ERROR | Resource %s is not currently in-service on server %s. Manually bring the resource in-service and retry the operation. | **原因**: 依存関係の作成中に、指定したリソースがサーバーで In Service ではありませんでした。<br><br>**処置**: 指定したサーバーでリソースを In Service にし、再度操作を実施してください。 |
| 112101 | ERROR | Error getting resource information for resource “%s” on server “%s” | **原因**: 指定したサーバー上の指定したリソース情報を確認することができない。<br><br>**処置**: 指定したサーバーに指定したリソースが存在していてすべての必要なファイルシステムがマウントされているかを確認してください。 |
| 112102 | ERROR | Cannot extend resource “%s” to server “%s” | **原因**: 指定したサーバーへ拡張したリソースを作成することができない。<br><br>**処置**: SAP ソフトウェアがターゲットサーバーに正常にインストールされ、すべての必要なファイルシステムがターゲットシステムからアクセス可能になっていることを確認してください。 |
| 112103 | ERROR | Error creating resource “%s” on server “%s” | **原因**: 指定したサーバーに指定したリソースを作成することができない。<br><br>**処置**: 指定したサーバーにSAP ソフトウェアが正常にインストールされ、すべての必要なファイルシステムがマウントされアクセス可能になっていることを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 112104 | ERROR | The extend script "%s" either does not exist or is not executable on %s. | **原因**: 指定したサーバーに指定した extend スクリプトが存在しないか実行することができない。<br><br>**処置**: すべての必要なリカバリーキットがインストールされ、extend スクリプトが実行可能か確認してください。 |
| 112106 | ERROR | Unable to create an internal SAP object for resource "%s" on %s. If the tag is correct, verify that all necessary SAP file systems are mounted and accessible. | **原因**: 指定したサーバー上の指定したリソース情報を確認することができませんでした。<br><br>**処置**: 指定したサーバーに指定したリソースが存在していてすべての必要なファイルシステムがマウントされているかを確認してください。 |
| 112112 | ERROR | Error getting resource information for resource “%s” on server “%s” | **原因**: 指定したサーバー上の指定したリソース情報を確認することができませんでした。<br><br>**処置**: 指定したサーバーに指定したリソースが存在していてすべての必要なファイルシステムがマウントされているかを確認してください。 |
| 112115 | FATAL | Error getting resource information for resource \”$tag\” on server \”$sap::me\”. | **原因**: 指定したサーバー上の指定したリソース情報を確認することができませんでした。<br><br>**処置**: 指定したサーバーに指定したリソースが存在していてすべての必要なファイルシステムがマウントされているかを確認してください。 |
| 112119 | ERROR | Error getting resource information for resource “%s” on server “%s” | **原因**: 指定したサーバー上の指定したリソース情報を確認することができませんでした。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 指定したサーバーに指定したリソースが存在していてすべての必要なファイルシステムがマウントされているかを確認してください。 |
| 112123 | ERROR | Error getting resource information for resource “%s” on server “%s” | **原因**: 指定したサーバー上の指定したリソース情報を確認することができませんでした。<br><br>**処置**: 指定したサーバーに指定したリソースが存在していてすべての必要なファイルシステムがマウントされているかを確認してください。 |
| 112124 | ERROR | Resource with either matching tag “%s” or id “%s” already exists on server “%s” for App “%s” and Type “%s” | **原因**: 指定したリソースタグ名もしくはID および同一のアプリケーション、タイプをもつリソースがすでに指定したサーバーに存在しています。<br><br>**処置**: SAP インスタンスが指定したサーバーでまだLifeKeeper に保護されていないことを確認してください。まだ保護されていない場合は、別のリソースタグ名を選択してください。 |
| 112125 | ERROR | Unable to create an SAP object for resource %s on system %s. | **原因**: 指定のサーバーで指定されたリソースを示すための SAP オブジェクトを作成することができない。<br><br>**処置**: 指定したサーバーにSAP ソフトウェアが正常にインストールされ、すべての必要なファイルシステムがマウントされアクセス可能になっていることを確認してください。 |
| 112126 | FATAL | Usage: “%s %s” Specify the correct usage for the requested command. | **原因**: SAP canextend スクリプトの使用に誤りがあります。<br><br>**処置**: canextend スクリプトに対する正 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | しいパラメーターを指定してください。: canextend <template server> <template tag> |
| 112127 | FATAL | Usage: \"$cmd $usage\" Specify the correct usage for the requested command. | 原因: SAP delete スクリプトの使用に誤りがある。<br><br>処置: SAP delete スクリプトに対する正しいパラメーターを指定してください。: delete [-U] -t <tag> -i <id> |
| 112128 | FATAL | Usage: “%s %s” Specify the correct usage for the requested command. | 原因: SAP depstoextend スクリプトの使用に誤りがある。<br><br>処置: depstoextend スクリプトに対する正しいパラメーターを指定してください。: depstoextend <template server> <template tag> |
| 112129 | FATAL | Usage: “%s %s” Specify the correct usage for the requested command. | 原因: SAP extend スクリプトの使用に誤りがある。<br><br>処置: extend スクリプトに対する正しいパラメーターを指定してください。: extend <template server> <template tag> <switchback> <target tag> |
| 112130 | FATAL | Usage: \"$cmd $usage\" Specify the correct usage for the requested command. | 原因: SAP quickCheck スクリプトの使用に誤りがある。<br><br>処置: quickCheck スクリプトに対する正しいパラメーターを指定してください。: quickCheck -t <tag> -i <id> |
| 112131 | FATAL | Usage: “%s %s” Specify the correct usage for the requested command. | 原因: SAP recover スクリプトの使用に誤りがある。<br><br>処置: recover スクリプトに対する正し |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | いパラメーターを指定してください。: recover -d <tag> |
| 112132 | FATAL | Usage: \"$cmd $usage\". Specify the correct usage for the requested command. | 原因: SAP remoteControl スクリプトの使用に誤りがある。<br><br>処置: remoteControl スクリプトに対する正しいパラメーターを指定してください。: remoteControl <tag> <remote instance> <remote cmd> <remote cmd option> <primary system> <primary tag> |
| 112133 | FATAL | Usage: “%s %s”. Specify the correct usage for the requested command. | 原因: SAP remove スクリプトの使用に誤りがある。<br><br>処置: remove スクリプトに対する正しいパラメーターを指定してください。: remove -t <tag> -i <id> |
| 112134 | FATAL | Usage: \"$cmd $usage\". Specify the correct usage for the requested command. | 原因: SAP restore スクリプトの使用に誤りがある。<br><br>処置: restore スクリプトに対する正しいパラメーターを指定してください。: restore -t <tag> -i <id> |
| 112137 | ERROR | The required parameter \"parent\" was either not provided or was invalid in the $func routine on $me. | 原因: CleanUpルーチンの使用に誤りがある。<br><br>処置: こちらは内部エラーです。SIOS カスタマーセンターへお問い合わせください。 |
| 112138 | ERROR | At least one required process for instance %s was not started successfully during "%s" on server %s. Please check the LifeKeeper and system logs for additional information. | 原因: 指定したサーバーで、指定した SAP インスタンスに対する少なくとも1つのプロセスが正常に開始されませんでした。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: LifeKeeper/システムログ, SAP トレースファイルなどから原因を調査し、操作を再度行ってください。 |
| 112140 | FATAL | The tag parameter was not specified for the internal \"$func\" routine on $me. If this was a command line operation, specify the correct parameters. Otherwise, consult the troubleshooting documentation. | **原因**: 指定したサーバーでタグパラメーターが GetLK ルーチンに指定されていません。<br><br>**処置**: こちらは内部エラーです。SIOS カスタマーセンターへお問い合わせください。 |
| 112141 | ERROR | Either the SID ("%s") or instance ("%s") parameter was not specified for the "%s" routine on %s. | **原因**: SID もしくは SID のどちらかが指定したサーバーの StatusSapServer ルーチン に指定されていません。<br><br>**処置**: こちらは内部エラーです。SIOS カスタマーセンターへお問い合わせください。 |
| 112142 | ERROR | Either the SID ("%s"), instance ("%s"), or instance number ("%s") parameter was not specified for the "%s" routine on %s. | **原因**: SID、インスタンス、インスタンス番号のいずれかが指定のサーバーの StartSapServer ルーチンに提供されていない。<br><br>**処置**: こちらは内部エラーです。SIOS カスタマーセンターへお問い合わせください。 |
| 112143 | ERROR | The SID, instance, or instance number parameter was not specified for the "%s" routine on %s. | **原因**: SID、インスタンス、インスタンス番号のいずれかが指定のサーバーの StopSapServer ルーチンで指定されていない。<br><br>**処置**: こちらは内部エラーです。SIOS カスタマーセンターへお問い合わせください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 112173 | ERROR | The file “%s” does not exist or was not readable on “%s”. Verify that the specified file exists and/or is readable. | **原因**: 指定したファイルが指定したサーバーに存在しないか、読み込み不可になっている。<br><br>**処置**: ファイルが存在しているかを確認および/もしくは権限を修正して読み込み可能な状態にしてください。 |
| 112174 | ERROR | The file “%s” does not exist or was not readable on “%s”. Verify that the specified file exists and/or is readable. | **原因**: 指定したファイルが指定したサーバーに存在しないか、読み込み不可になっています。<br><br>**処置**: ファイルが存在しているかを確認および/もしくは権限を修正して読み込み可能な状態にしてください。 |
| 112175 | ERROR | The file “%s” does not exist or was not readable on “%s”. Verify that the specified file exists and/or is readable. | **原因**: 指定したファイルが指定したサーバーに存在しないか、読み込み不可になっている。<br><br>**処置**: ファイルが存在しているかを確認および/もしくは権限を修正して読み込み可能な状態にしてください。 |
| 112194 | ERROR | There was an error verifying the NFS connections for SAP related mount points on “%s”. One or more NFS servers is not operational and needs to be restarted. | **原因**: マウントポイントが /etc/default/LifeKeeper の SAP_NFS_CHECK_DIRS エントリに記述されている少なくとも一つの 重要なNFS 共有システムが現在無効となっている。<br><br>**処置**: すべての必要な NFS 共有ファイルシステムがアクセス可能であることを確認し、現在操作不可となっている NFS サーバーを再起動してください。 |
| 112195 | FATAL | There was an error verifying the NFS connections for SAP related mount points on $me. One or more NFS servers is not operational and needs to be restarted. | **原因**: マウントポイントが /etc/default/LifeKeeper の SAP_NFS_CHECK_DIRS エントリに記述されている少なくとも一つの 重要なNFS 共有システムが現在無 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 効となっている。<br><br>**処置**: すべての必要な NFS 共有ファイルシステムがアクセス可能であることを確認し、現在操作不可となっている NFS サーバーを再起動してください。 |
| 112196 | WARN | There was an error verifying the NFS connections for SAP related mount points on “%s”. One or more NFS servers is not operational and needs to be restarted. | **原因**: マウントポイントが /etc/default/LifeKeeper の SAP_NFS_CHECK_DIRS エントリに記述されている少なくとも一つの 重要なNFS 共有システムが現在無効となっている。<br><br>**処置**: すべての必要な NFS 共有ファイルシステムがアクセス可能であることを確認し、現在操作不可となっている NFS サーバーを再起動してください。 |
| 112201 | ERROR | The internal object value "%s" was empty. Unable to complete "%s" on %s. | **原因**: リソースタグの引数が GetLKEquivルーチンに提供されていません。<br><br>**処置**: こちらは内部エラーです。SIOS カスタマーセンターへお問い合わせください。 |
| 112203 | ERROR | The internal object value "%s" was empty. Unable to complete "%s" on %s. Additional information available in the LifeKeeper and system logs. | **原因**: SAP インスタンス番号が IsInstanceRunning ルーチンに提供されていません。<br><br>**処置**: こちらは内部エラーです。SIOS カスタマーセンターへお問い合わせください。 |
| 112204 | ERROR | The internal object value "%s" was empty. Unable to complete "%s" on %s. | **原因**: saphostexec もしくは saposcol コマンドのどちらを使用するか判断できない。<br><br>**処置**: SAP kernel 7.3より前のSAP Net Weave に対して SAP_SRVHOST_CM |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | D, SAP_HOSTCTL_CMD, SAP_OSCOL_CMD を使用している場合は、チューニングが正しく設定されていることを確認してください。 |
| 112205 | ERROR | The internal object value "%s" was empty. Unable to complete "%s" on %s. | **原因**: SAP インスタンスが 指定したサーバーのSAPRemExec ルーチンに提供されなかった。<br><br>**処置**: こちらは内部エラーです。SIOS カスタマーセンターへお問い合わせください。 |
| 112206 | ERROR | The required action parameter was not provided. Unable to complete "%s" on %s. | **原因**: 指定したサーバーで SAPRemExec ルーチンに対するアクションが提供されませんでした。<br><br>**処置**: こちらは内部エラーです。SIOS カスタマーセンターへお問い合わせください。 |
| 112208 | ERROR | Error getting the value of "%s" or "%s" from the default profile "%s" on %s. Verify that the specified value exists. | **原因**: 指定したサーバーで提供されたプロファイルから仮想IPもしくはホスト名を取得できない。<br><br>**処置**: プロファイルに適切なエントリーが存在していることを確認してください。 |
| 112209 | FATAL | Unable to gather required information from the SAP default profile for SID $sid ($DPFL) on $me. Verify that the default profile exists and is accessible. | **原因**: 指定したサーバーで提供されたプロファイルから SAP インスタンスに関する情報を取得できない。<br><br>**処置**: SAP ソフトウェアが正常にインストールされ提供されたプロファイルが存在および読み込み可能となっていることを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 112214 | ERROR | Unable to determine the status of the path "%s" ("%s") on %s. The path on %s may require the execution of the command: "mount <ip>:<export> %s". Verify that the SAP software is correctly installed and that all SAP file systems are mounted and accessible. | **原因**: 指定したパスのファイルシステムのステータスを判断できません。<br><br>**処置**: SAP ソフトウェアが正常にインストールされすべての必要となるファイルシステムがマウントされていることを確認してください。 |
| 112219 | ERROR | [HACONNECTOR:%s] Unable to write to file "%s" on %s. If the file already exists, manually enable write permissions on it. | **原因**: 指定したファイルが指定したサーバーで読み込み/書き込み権限をもっていない。<br><br>**処置**: 指定したサーバーの読み込み/書き込み権限を有効にしてください。 |
| 112220 | ERROR | Unable to start the sapstartsrv service for SID %s and SAP instance %s on %s. Verify that the sapservices file is correct and the process can be started manually. | **原因**: 指定した SAP インスタンスに対する SAP Start Service (sapstartsrv) プロセスを開始できない。<br><br>**処置**: sapservices ファイルがsapstartsrv プロセスを開始するための適切なコマンドを含んでいて、そのプロセスを手動で開始できるかを確認してください。 |
| 112221 | ERROR | The internal "%s" routine was called for a resource with unsupported instance type %s. This method supports only SAP instance types TYPE_CS, TYPE_ERS, or TYPE_NEW_ERS (1, 2, or 5). | **原因**: サポートされていないインスタンスタイプの内部SAPオブジェクトに対して、指定されたルーチンが呼び出された。<br><br>**処置**: こちらは内部エラーです。SIOS カスタマーセンターへお問い合わせください。 |
| 112222 | ERROR | The internal "%s" routine was called for a resource with unsupported instance type %s. This method supports only SAP instance types TYPE_CS, TYPE_ERS, or TYPE_NEW_ERS (1, 2, or 5). | **原因**: サポートされていないインスタンスタイプの内部SAPオブジェクトに対して、指定されたルーチンが呼び出された。<br><br>**処置**: こちらは内部エラーです。SIOS |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | カスタマーセンターへお問い合わせください。 |
| 112223 | ERROR | The internal "%s" routine was called for a resource with unsupported instance type %s. This method supports only SAP instance type TYPE_CS (1). | **原因**: サポートされていないインスタンスタイプの内部SAPオブジェクトに対して、指定されたルーチンが呼び出された。<br><br>**処置**: こちらは内部エラーです。SIOSカスタマーセンターへお問い合わせください。 |
| 112224 | ERROR | The internal \"%s\" routine was called for a resource with unsupported instance type %s. This method supports only SAP instance type TYPE_CS (1). | **原因**: サポートされていないインスタンスタイプの内部SAPオブジェクトに対して、指定されたルーチンが呼び出された。<br><br>**処置**: こちらは内部エラーです。SIOSカスタマーセンターへお問い合わせください。 |
| 112225 | ERROR | The internal \"%s\" routine was called for a resource with unsupported instance type %s. This method supports only SAP instance type TYPE_NEW_ERS (5). | **原因**: サポートされていないインスタンスタイプの内部SAPオブジェクトに対して、指定されたルーチンが呼び出された。<br><br>**処置**: こちらは内部エラーです。SIOSカスタマーセンターへお問い合わせください。 |
| 112226 | ERROR | The internal "%s" routine was called for a resource with unsupported instance type %s. This method supports only SAP instance types TYPE_CS, TYPE_ERS, and TYPE_NEW_ERS (1, 2, and 5). | **原因**: サポートされていないインスタンスタイプの内部SAPオブジェクトに対して、指定されたルーチンが呼び出された。<br><br>**処置**: こちらは内部エラーです。SIOSカスタマーセンターへお問い合わせください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 112227 | ERROR | The internal "%s" routine was called for a resource with unsupported instance type %s. This routine only supports SAP instance type TYPE_CS (1). | **原因**: サポートされていないインスタンスタイプの内部SAPオブジェクトに対して、指定されたルーチンが呼び出された。<br><br>**処置**: こちらは内部エラーです。SIOSカスタマーセンターへお問い合わせください。 |
| 112228 | ERROR | The internal "%s" routine was called for a resource with unsupported instance type %s. This method supports only SAP instance type TYPE_NEW_ERS (5). | **原因**: サポートされていないインスタンスタイプの内部SAPオブジェクトに対して、指定されたルーチンが呼び出された。<br><br>**処置**: こちらは内部エラーです。SIOSカスタマーセンターへお問い合わせください。 |
| 112229 | ERROR | The profile \"%s\" either does not exist or cannot be read on %s. Unable to determine whether enqueue replication is enabled for resource %s. Please verify that the file exists and can be read. | **原因**: 指定したサーバー上に指定したインスタンスプロファイルが存在していない、もしくは読み取りができない。<br><br>**処置**: ファイルが存在していて、読み取り可能かどうかを確認してください。 |
| 112325 | ERROR | The $SAP_CONTROL utility cannot be located. Verify that all necessary file systems are mounted and that the $SAP_CONTROL utility can be located in the $sapadm in user's PATH. | **原因**: SAP インスタンスの管理に必要な sapcontrol ユーティリティが配置できない。<br><br>**処置**: 全ての必要なファイルシステムがマウントされ、sapcontrol ユーティリティがSAP 管理ユーザーの Path に配置されていることを確認してください。 |
| 112326 | ERROR | The \"which $SAP_CONTROL\" command for user $sapadmin returned $sapcmd as the location of the $SAP_CONTROL utility, but the utility could not be found or was not executable in this location. Verify that all necessary file systems are mounted and that the $SAP_CONTROL utility can | **原因**: SAP インスタンスの管理に必要な sapcontrol ユーティリティが配置できない。<br><br>**処置**: 全ての必要なファイルシステムが |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | be located in the $sapadmin user's PATH. | マウントされ、sapcontrol ユーティリティがSAP 管理ユーザーのパスに配置されていることを確認してください。 |
| 112412 | ERROR | Invalid mount entry detected (‘$entry’). Verify that all mount entries in the following files on server $me have the correct format: $filesToCheckList. | **原因**: 指定されたマウントエントリーの形式が無効です。<br><br>**処置**: エラーメッセージに表示されたファイルを調べて、すべてのマウントエントリーが正しい形式であることを確認してください (fstab スタイルのエントリーを使用)。 |
| 112416 | ERROR | No parent directory provided. Unable to obtain subdirectories. | **原因**: SID の予想される SAP ディレクトリレイアウトが存在しません。<br><br>**処置**: SAP ディレクトリ構造が正しく設定されていること、およびすべてのリンクが正しいことを確認してください。 |
| 112417 | ERROR | Unable to open directory ‘$ParentDir’: $! | **原因**: LifeKeeper は、指定されたディレクトリを開くことができません。 エラーはメッセージに示されています。<br><br>**処置**: 一覧表示されたディレクトリをチェックして、権限と所有者が正しいことを確認してください。 |
| 112433 | ERROR | Unsupported SAPENQ_VERSION ($enqversion) for resource $tag on $me. Unable to obtain enqueue replication status. | **原因**: 指定したサーバーで指定したリソースの SAPENQ_VERSION パラメーターにサポートされていない値が検知されました。<br><br>**処置**: 指定したリソースの info ファイルで、SAPENQ_VERSION に有効な値 (現在使用しているエンキューサーバーのバージョンを表す1 または 2) が設定されていることを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 112437 | ERROR | Profile \"$srvpf\" not found on $me. Unable to obtain enqueue replication status for instance $inst. | **原因**: 指定したサーバー上に指定したインスタンスプロファイルが存在していない、もしくは読み取りができない。<br><br>**処置**: ファイルが存在していて、読み取り可能かどうかを確認してください。 |
| 112438 | ERROR | Unsupported SAPENQ_VERSION ($enqversion) for resource $tag on $me. Unable to obtain enqueue replication status for instance $inst. | **原因**: 指定したサーバーで指定したリソースのSAPENQ_VERSION パラメーターにサポートされていない値が検知された。<br><br>**処置**: 指定したリソースの info ファイルで、SAPENQ_VERSION に有効な値 (現在使用しているエンキューサーバーのバージョンを表す1 または 2) が設定されていることを確認してください。 |
| 112470 | ERROR | The instance profile %s could not be found on %s. Verify that the SAP software is installed correctly and that all necessary file systems are mounted and accessible. | **原因**: 指定したサーバー上に指定したインスタンスプロファイルが存在していない、もしくは読み取りができない。<br><br>**処置**: ファイルが存在していて、読み取り可能かどうかを確認してください。 |
| 112490 | ERROR | [HACONNECTOR] Unable to determine the corresponding tag for resource with ID \"$res\" on $me. | **原因**: 指定したサーバー上に指定したリソースID をもつ LifeKeeper SAP リソースを見つけることができない。<br><br>**処置**: fra コマンドの —res 引数として提供されるソース ID が有効な LifeKeeper SAP リソースに対応しているか確認してください。 |
| 112507 | ERROR | [HACONNECTOR] At least one required process for the instance was not killed successfully during the fra migrate action on $me. Aborting resource migration. | **原因**: fra マイグレーション実行中に SAP インスタンスが指定したサーバーで正常に停止しなかった。<br><br>**処置**: SAP インスタンスに対して起動中 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | のプロセスを手動で停止し、マイグレーションを再度実行してください。 |
| 112539 | ERROR | [HACONNECTOR] Unable to find rpm information for one or more packages on $me. | 原因: HA Connector gvi (“Get Version Information”)ルーチンは、LifeKeeper および/または the SAP Recovery Kit のバージョン番号を判別することができなかった。<br><br>処置: rpm -q コマンドでLifeKeeper Core および SAP Recovery Kit の rpm 情報を取得できるかを確認してください。 |
| 112976 | ERROR | There is no LifeKeeper protected resource with tag $tag on system $me. | 原因: SAP canfailover スクリプトに渡されたリソースタグに対応する LifeKeeper リソースがない。<br><br>処置: リソースタグ名が正しいことを確認し再度コマンドを実行してください。 |
| 112573 | ERROR | Failed to create or write to \"$file\" on server $me. Verify that the filesystem is not full and that the file is write-enabled. | 原因: 指定されたサーバー上の指定されたファイルの作成または書き込みに失敗しました。<br><br>処置: ファイルシステムがいっぱいでないこと、およびファイルが存在する場合は書き込み可能であることを確認してください。 |
| 112575 | ERROR | Failed to delete file \"$file\" on server $me. Please delete the file manually. | 原因: 指定されたサーバー上の指定されたファイルを削除できませんでした。<br><br>処置: ファイルを手動で削除してください。 |
| 112577 | ERROR | Failed to remove empty directory \"$dir\" on server $me. Please remove the directory manually. | 原因: 指定されたサーバー上の指定されたディレクトリーを削除できませんでした。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: ディレクトリーを手動で削除してください。 |
| 112581 | ERROR | Failed to execute routine $routine with arguments \"$arg_list\" on server $sys. | **原因**: 指定されたサーバーで指定された引数を使用して、指定された SAP perl モジュールルーチンを実行できませんでした。<br><br>**処置**: 指定されたサーバーで LifeKeeper が実行され、完全に初期化されていることを確認してください。 追加情報については、LifeKeeper ログを調べてください。 |
| 112582 | WARN | Detected unsupported NFS version \"$foundNFSVers\" for mount point $mountLoc from shared file system $mountDev. The default NFS version specified by the pingnfs utility will be used instead. If necessary, this value can be overridden by setting the NFS_VERSION parameter in /etc/default/LifeKeeper. | **原因**: サポートされていない NFS バージョンが検出されました。<br><br>**処置**: サポートされているプロトコルに更新するか、サポートされていないプロトコルを使用する必要がある場合は、メッセージ内のデフォルトファイルエントリでオーバーライドしてください。 |
| 112583 | WARN | Detected unsupported NFS transport protocol \"$foundProto\" for mount point $mountLoc from shared file system $mountDev. The default NFS protocol specified by the pingnfs utility will be used instead. If necessary, this value can be overridden by setting the NFS_RPC_PROTOCOL parameter in /etc/default/LifeKeeper. | **原因**: サポートされていない NFS トランスポートプロトコルが検出されました。<br><br>**処置**: サポートされているプロトコルに更新するか、サポートされていないプロトコルを使用する必要がある場合は、メッセージ内のデフォルトファイルエントリでオーバーライドしてください。 |
| 112976 | ERROR | There is no LifeKeeper protected resource with tag $tag on system $me. | **原因**: SAP canfailover スクリプトに指定されたリソースタグは、既存の LifeKeeper リソースに対応していません。<br><br>**処置**: リソースタグ名が正しいことを確認し、コマンドを再実行してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 112977 | ERROR | Resource $tag is not a $app/$typ resource. Please use the $ins_app/$ins_typ resource-specific canfailover script instead. | 原因: SAP canfailover スクリプトに渡された LifeKeeper リソースが、SAP（app suite/sap）リソースではない。<br><br>処置: 指定したリソースに対して適切なタイプ固有の canfailover スクリプトを使用してください。 |
| 122005 | ERROR | Unable to "%s" on "%s" | 原因:「getlocks」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122007 | ERROR | Unable to "%s" on "%s" | 原因:「rlslocks」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122009 | ERROR | The path %s is not a valid file. | 原因: listener.ora ファイルがありません。<br><br>処置: ファイルが存在していることを確認し、操作を再試行してください。 |
| 122010 | ERROR | The listener user does not exist on the server %s. | 原因:「Stat」コマンドでユーザ ID を取得できませんでした。<br><br>処置: 操作を再試行してください。 |
| 122011 | ERROR | The listener user does not exist on the server %s. | 原因: UID が passwd ファイルにありません。<br><br>処置: UID が passwd ファイルに存在していることを確認し、操作を再試行して |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ください。 |
| 122012 | ERROR | The listener user does not exist on the server %s. | 原因:ユーザ名が passwd ファイルにありません。<br><br>処置: ユーザ名が passwd ファイルに存在していることを確認し、操作を再試行してください。 |
| 122023 | ERROR | The %s command failed (%d) | 原因: このメッセージに「lsnrctl」コマンドの戻りコードが含まれています。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122024 | ERROR | $line | 原因: メッセージに「lsnrctl」コマンドの結果が含まれています。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122039 | ERROR | Usage error | 原因: リストア操作で無効なパラメーターが指定されました。<br><br>処置: パラメーターを確認し、操作を再試行してください。 |
| 122040 | ERROR | Script $cmd has hung on the restore of \"$opt_t\". Forcibly terminating. | 原因: リスナーの restore スクリプトがタイムアウト値に達しました。<br><br>処置: listener.ora が有効であること、および /etc/default/LifeKeeper の LSNR_START_TIME（デフォルトでは 35 秒）がリスナーの起動に必要な時間以上の値に設定されていることを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122041 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | 原因: LifeKeeper は、{server} でリソース {resource} をリストアできませんでした。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122045 | ERROR | Error getting resource information for resource "%s" on server "%s" | 原因: リソース情報の取得に失敗しました。<br><br>処置: LifeKeeper の設定を確認してください。 |
| 122046 | ERROR | Usage error | 原因: リストア操作で無効なパラメーターが指定されました。<br><br>処置: パラメーターを確認し、操作を再試行してください。 |
| 122049 | ERROR | The script $cmd has hung on remove of \"$opt_t\".  Forcibly terminating. | 原因: リスナーの remove スクリプトがタイムアウト値に達しました。<br><br>処置: listener.ora が有効であること、および /etc/default/LifeKeeper の LSNR_STOP_TIME (デフォルトでは 35 秒) がリスナーの停止に必要な時間以上の値に設定されていることを確認してください。 |
| 122051 | ERROR | Error getting resource information for resource "%s" on server "%s" | 原因: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。<br><br>処置: LifeKeeper の設定を確認してください。 |
| 122055 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | 原因: LifeKeeper は、{server} でリソース {resource} に対して quickCheck を実行できませんでした。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122057 | ERROR | Error getting resource information for resource "%s" on server "%s" | **原因**: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 122064 | WARN | The %s level is set to %s a %s will not occur. | **原因**: リスナーの最低限の保護レベルは起動と監視です。<br><br>**処置**: リスナーを手動で起動してください。 |
| 122066 | ERROR | Script has hung checking \"$tag\". Forcibly terminating. | **原因**: リスナーの quickCheck スクリプトがタイムアウト値に達しました。<br><br>**処置**: listener.ora が有効であること、および /etc/default/LifeKeeper の LSNR_STATUS_TIME (デフォルトでは 15 秒) がリスナーの確認に必要な時間以上の値に設定されていることを確認してください。 |
| 122067 | ERROR | Usage error | **原因**: quickCheck 操作で無効なパラメーターが指定されました。<br><br>**処置**: パラメーターを確認し、操作を再試行してください。 |
| 122069 | ERROR | Usage error | **原因**: 削除操作で無効なパラメーターが指定されました。<br><br>**処置**: パラメーターを確認し、操作を再試行してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122072 | ERROR | %s: resource "%s" not found on local server | **原因**: 復旧操作で無効なパラメーターが指定されました。<br><br>**処置**: パラメーターを確認し、操作を再試行してください。 |
| 122074 | WARN | The local recovery attempt has failed but %s level is set to %s preventing a failover to another node in the cluster. With %s recovery set all local recovery failures will exit successfully to prevent resource failovers. | **原因**: オプションのリスナーリカバリーレベルがローカルリカバリーにのみ設定されています。<br><br>**処置**: リソースツリーを手動でスイッチオーバーしてください。 |
| 122078 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | **原因**:LifeKeeper は、{server} でリソース {resource} を復旧できませんでした。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122082 | ERROR | Error getting resource information for resource "%s" on server "%s" | **原因**: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 122083 | ERROR | $cmd has hung checking \"$tag\". Forcibly terminating | **原因**: recover スクリプトがシグナルによって停止されました。<br><br>**処置**: listener.ora が有効であることを確認してください。 |
| 122084 | ERROR | Cannot extend resource "%s" to server "%s" | **原因**: LifeKeeper は、{server} でリソース {resource} を拡張できませんでした。<br><br>**処置**: パラメーターを確認し、操作を再 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 試行してください。 |
| 122085 | ERROR | Usage: %s %s | 原因: canextend 操作で無効なパラメーターが指定されました。<br><br>処置: パラメーターを確認し、操作を再試行してください。 |
| 122086 | ERROR | The values specified for the target and the template servers are the same. Please specify the correct values for the target and template servers. | 原因: ターゲットサーバーに指定した値とテンプレートサーバーに指定した値が同じです。<br><br>処置: メッセージテキストに記載されている手順を実行してください。 |
| 122087 | ERROR | Error getting resource information for resource "%s" on server "%s" | 原因: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。<br><br>処置: LifeKeeper の設定を確認してください。 |
| 122088 | ERROR | Error getting resource information for resource "%s" on server "%s" | 原因: リソース情報からのリスナーユーザ名の取得に失敗しました。<br><br>処置: リソース情報フィールドが有効であることを確認し、操作を再試行してください。 |
| 122089 | ERROR | The listener user %s does not exist on the server %s. | 原因: ユーザ名が passwd ファイルにありません。<br><br>処置: ユーザ名が passwd ファイルに存在していることを確認し、操作を再試行してください。 |
| 122090 | ERROR | The id for user %s is not the same on template server %s and target server %s. | 原因: ユーザ ID はどちらのサーバーでも |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 同じものにする必要があります。<br><br>**処置**: ユーザ ID を同じものにしてください。 |
| 122091 | ERROR | The group id for user %s is not the same on template server %s and target server %s. | **原因**: グループ ID はどちらのサーバーでも同じものにする必要があります。<br><br>**処置**: グループ ID を同じものにしてください。 |
| 122092 | ERROR | Cannot access canextend script "%s" on server "%s" | **原因**: {server} で「canextend」スクリプトを見つけられなかったので、LifeKeeper は拡張前のチェックを実行できませんでした。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 122097 | ERROR | Usage: %s %s | **原因**:「config処置s」操作で無効な引数が指定されました。<br><br>**処置**: 引数を確認し、操作を再試行してください。 |
| 122098 | ERROR | Error getting resource information for resource "%s" on server "%s" | **原因**: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 122099 | ERROR | Unable to update the resource %s to change the %s to %s on %s. | **原因**: LifeKeeper は情報フィールドに情報を入力できませんでした。<br><br>**処置**: LifeKeeper を再起動し、操作を再試行してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122100 | ERROR | Error getting resource information for resource "%s" on server "%s" | 原因: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。<br><br>処置: LifeKeeper の設定を確認してください。 |
| 122101 | ERROR | Unable to update the resource %s to change the %s to %s on %s. | 原因: LifeKeeper は {server} で情報フィールドに情報を入力できませんでした。<br><br>処置: {server} で LifeKeeper を再起動し、操作を再試行してください。 |
| 122103 | ERROR | Usage: %s %s | 原因: 作成操作で無効なパラメーターが指定されました。<br><br>処置: パラメーターを確認し、操作を再試行してください。 |
| 122124 | ERROR | END failed hierarchy "%s" of resource "%s" on server "%s" with return value of %d | 原因: LifeKeeper は、{server} でリソース {resource} を作成できませんでした。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122126 | ERROR | Unable to "%s" on "%s | 原因:「rlslocks」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122127 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | 原因: LifeKeeper は、{server} でリソース {resource} を作成できませんでした。<br><br>処置: 関連するエラーをログで確認 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | し、レポートされている問題を解決してください。 |
| 122129 | ERROR | Unable to "%s" on "%s | 原因:「getlocks」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122131 | ERROR | Error creating resource "%s" on server "%s" | 原因: LifeKeeper は、{server} でリソース {resource} を作成できませんでした。<br><br>処置: LifeKeeper の設定を確認してください。 |
| 122133 | ERROR | Unable to create a file system resource hierarchy for the file system %s. | 原因:「filesyshier」の実行中に予期しないエラーが発生しました。<br><br>処置: 隣接するログメッセージで詳細を確認してください。 |
| 122135 | ERROR | Unable to create a dependency between parent tag %s and child tag %s. | 原因:「dep_create」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122140 | ERROR | Resource "%s" is not ISP on server "%s" Manually bring the resource in service and retry the operation | 原因: リスナーリソースが依存する IP リソース {tag} は ISP の必要があります。<br><br>処置: メッセージテキストに記載されている手順を実行してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122141 | ERROR | Unable to create a dependency between parent tag %s and child tag %s. | 原因:「dep_create」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122144 | ERROR | Usage: %s %s | 原因:「create_ins」操作で無効なパラメーターが指定されました。<br><br>処置: パラメーターを確認し、操作を再試行してください。 |
| 122145 | ERROR | An error has occurred in utility %s on server %s. View the LifeKeeper logs for details and retry the operation. | 原因:「app_create」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122146 | ERROR | An error has occurred in utility %s on server %s. View the LifeKeeper logs for details and retry the operation. | 原因:「typ_create」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122147 | ERROR | An error has occurred in utility %s on server %s. View the LifeKeeper logs for details and retry the operation. | 原因:「newtag」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122148 | ERROR | Error creating resource "%s" on server "%s" | 原因: LifeKeeper は、{server} でリソース {resource} を作成できませんでした。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: LifeKeeper の設定を確認してください。 |
| 122149 | ERROR | An error has occurred in utility %s on server %s. View the LifeKeeper logs for details and retry the operation. | **原因**:「ins_setstate」の実行中に予期しないエラーが発生しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122150 | ERROR | Error creating resource "%s" on server "%s" | **原因**: LifeKeeper は、{server} でリソース {resource} を作成できませんでした。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 122151 | ERROR | Usage: %s %s | **原因**:「depstoextend」操作で無効な引数が指定されました。<br><br>**処置**: 引数を確認し、操作を再試行してください。 |
| 122152 | ERROR | Usage: %s %s | **原因**:「extend」操作で無効なパラメーターが指定されました。<br><br>**処置**: パラメーターを確認し、操作を再試行してください。 |
| 122153 | ERROR | Error getting resource information for resource "%s" on server "%s" | **原因**: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 122154 | ERROR | Cannot extend resource "%s" to server "%s" | **原因**: LifeKeeper は、{server} でリソー |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ス {resource} を拡張できませんでした。 |
| 122155 | ERROR | Resource with either matching tag "%s" or id "%s" already exists on server "%s" for App "%s" and Type "%s" | **原因**: リスナーリソースの拡張中に、同じ {tag} または {id} で異なるリソースアプリケーションおよびタイプを使用しているリソースインスタンスが見つかりました。<br><br>**処置**: リソース ID は一意のものにする必要があります。Oracle Listener リソースインスタンスに合致する ID のリソースインスタンスを削除してください。 |
| 122156 | ERROR | Cannot access extend script "%s" on server "%s" | **原因**: LifeKeeper は、{server} でスクリプト EXTEND が見つからなかったので、リソース階層を拡張できませんでした。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 122157 | ERROR | Usage: %s %s | **原因**:「getConfigIps」操作で無効な引数が指定されました。<br><br>**処置**: 引数を確認し、操作を再試行してください。 |
| 122158 | ERROR | The file %s is not a valid listener file. The file does not contain any listener definitions. | **原因**: リスナー定義が見つかりませんでした。<br><br>**処置**: リスナー定義が listener.ora にあることを確認し、操作を再試行してください。 |
| 122159 | ERROR | Usage: %s %s | **原因**:「getSidListeners」操作で無効な引数が指定されました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 引数を確認し、操作を再試行してください。 |
| 122160 | ERROR | The file %s is not a valid listener file. The file does not contain any listener definitions. | **原因**: リスナー定義が見つかりませんでした。<br><br>**処置**: リスナー定義が listener.ora にあることを確認し、操作を再試行してください。 |
| 122161 | ERROR | Usage: %s %s | **原因**:「lsn-display」操作で無効な引数が指定されました。<br><br>**処置**: 引数を確認し、操作を再試行してください。 |
| 122162 | ERROR | Error getting resource information for resource "%s" on server "%s" | **原因**: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 122163 | ERROR | Usage: %s %s | **原因**: updateHelper 操作で無効なパラメーターが指定されました。<br><br>**処置**: パラメーターを確認し、操作を再試行してください。 |
| 122164 | ERROR | END failed hierarchy "%s" of resource "%s" on server "%s" with return value of %d | **原因**: LifeKeeper は、{server} でリソース {resource} を更新できませんでした。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122166 | ERROR | Usage: %s %s | 原因:「updateHelper」操作で無効なパラメーターが指定されました。<br><br>処置: パラメーターを確認し、操作を再試行してください。 |
| 122170 | ERROR | Unable to create a dependency between parent tag %s and child tag %s. | 原因:「dep_create」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122171 | ERROR | Unable to create a dependency between parent tag %s and child tag %s. | 原因:「dep_create」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122172 | ERROR | Usage: %s %s | 原因:「updIPDeps」操作で無効な引数が指定されました。<br><br>処置: 引数を確認し、操作を再試行してください。 |
| 122173 | ERROR | END failed hierarchy "%s" of resource "%s" on server "%s" with return value of %d | 原因: LifeKeeper は、{server} でリソース {resource} を更新できませんでした。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122175 | ERROR | Unable to "%s" on "%s | 原因:「rlslocks」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | し、レポートされている問題を解決してください。 |
| 122177 | ERROR | Unable to "%s" on "%s | 原因:「getlocks」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122180 | ERROR | Unable to create a dependency between parent tag %s and child tag %s. | 原因:「dep_create」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122181 | ERROR | Unable to create a dependency between parent tag %s and child tag %s. | 原因:「dep_create」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122183 | ERROR | The path %s is not a valid file. | 原因: listener.ora ファイルがありません。<br><br>処置: ファイルが存在していることを確認し、操作を再試行してください。 |
| 122185 | ERROR | The file %s is not a valid listener file. The file does not contain any listener definitions. | 原因: LifeKeeper は有効なリスナー定義の検索に失敗しました。<br><br>処置: 有効なリスナー定義が listener.ora にあることを確認し、操作を再試行してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122186 | ERROR | The value specified for %s cannot be empty. Please specify a value for this field. | 原因:設定ファイルまたは実行可能ファイルの {path} フィールドが空です。<br><br>処置: 空でない値を {path} に入力し、操作を再試行してください。 |
| 122187 | ERROR | The path %s is not a valid file or directory. | 原因: 定義済み {path} が無効です。<br><br>処置: {path} が存在していることを確認し、操作を再試行してください。 |
| 122188 | ERROR | The path %s is not a valid file or directory. | 原因: {path} がありません。<br><br>処置: {path} が存在していることを確認し、操作を再試行してください。 |
| 122189 | ERROR | The value specified for %s cannot be empty. Please specify a value for this field. | 原因: 設定および実行パスフィールドが空です。<br><br>処置: フィールドのパスを入力してください。 |
| 122190 | ERROR | Usage: %s %s | 原因:「valid_rpath」操作で無効な引数が指定されました。<br><br>処置: 引数を確認し、操作を再試行してください。 |
| 122191 | ERROR | The values specified for the target and the template servers are the same. | 原因: valid_rpath の無効な引数。<br><br>処置: 引数を確認し、操作を再試行してください。 |
| 122192 | ERROR | Unable to find the configuration file "oratab" in its default locations, /etc/oratab or %s on "%s" | 原因: /etc/oratab または {path} に oratab ファイルがありません。<br><br>処置: oratab ファイルが {path} にあるこ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | と、または /etc/default/Lifekeeper の ORACLE_ORATABLOC が有効なパスに設定されていることを確認してください。 |
| 122193 | ERROR | Unable to find the configuration file "oratab" in its default locations, /etc/oratab or %s on "%s" | 原因: /etc/oratab または {path} に oratab ファイルがありません。<br><br>処置: oratab ファイルが {path} にあること、または /etc/default/Lifekeeper の ORACLE_ORATABLOC が有効なパスに設定されていることを確認してください。 |
| 122194 | ERROR | Unable to find the configuration file "oratab" in its default locations, /etc/oratab or %s on "%s" | 原因: /etc/oratab または {path} に oratab ファイルがありません。<br><br>処置: oratab ファイルが {path} にあること、または /etc/default/Lifekeeper の ORACLE_ORATABLOC が有効なパスに設定されていることを確認してください。 |
| 122195 | ERROR | Unable to find the configuration file "oratab" in its default locations, /etc/oratab or %s on "%s" | 原因: /etc/oratab または {path} に oratab ファイルがありません。<br><br>処置: oratab ファイルが {path} にあること、または /etc/default/Lifekeeper の ORACLE_ORATABLOC が有効なパスに設定されていることを確認してください。 |
| 122196 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | 原因:LifeKeeper は、{server} でリソース {resource} を削除できませんでした。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122197 | ERROR | Unable to find the configuration file \"oratab\" in its default locations, /etc/oratab or $listener::oraTab on \"$me\" | 原因: /etc/oratab または {path} に oratab ファイルがありません。<br><br>処置: oratab ファイルが {path} にあること、または /etc/default/Lifekeeper の ORACLE_ORATABLOC が有効なパスに設定されていることを確認してください。 |
| 122198 | ERROR | remove for $okListener failed. | |
| 122251 | ERROR | Update of pluggable database info field for "%s" on "%s" failed (%s). | |
| 122252 | ERROR | Initial connect with query buffer to database "%s" on "%s" failed, testing output. | |
| 122253 | ERROR | The Oracle database "%s" is not running or no open connections are available on "%s". | |
| 122261 | ERROR | The Oracle resource (%s) and dependency are not set on %s. | |
| 122262 | ERROR | Usage: %s %s | |
| 122263 | ERROR | The restore of %s has timed out on server %s. The default TIMEOUT is 300 seconds. To increase the TIMEOUT, set ORACLE_RESTORE_TIMEOUT in /etc/default/LifeKeeper. | |
| 122264 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | |
| 122268 | ERROR | Failed to create object instance for Oracle on "%s". | |
| 122269 | ERROR | no dependency for Oracle on "%s". | |
| 122270 | ERROR | Unable to find the Oracle executable "%s" on "%s". | |
| 122271 | ERROR | Usage: %s %s | |
| 122272 | ERROR | The remove of %s has timed out on server %s. The default TIMEOUT is 300 seconds. To increase the TIMEOUT, set ORACLE_REMOVE_TIMEOUT in /etc/default/LifeKeeper. | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122273 | ERROR | END failed %s of "%s" on server "%s" du e to a "%s" signal | |
| 122277 | ERROR | Usage: %s %s | |
| 122278 | ERROR | Failed to create object instance for Oracle on "%s". | |
| 122279 | ERROR | no dependency for Oracle on "%s". | |
| 122280 | ERROR | Unable to find the Oracle executable "%s" on "%s". | |
| 122281 | ERROR | The quickCheck of %s has timed out on s erver %s. The default TIMEOUT is 45 sec onds. To increase the TIMEOUT, set OR ACLE_QUICKCHECK_TIMEOUT in /etc/ default/LifeKeeper. | |
| 122282 | ERROR | END failed %s of "%s" on server "%s" du e to a "%s" signal | |
| 122284 | ERROR | Failed to create object instance for Oracle on "%s". | |
| 122285 | ERROR | no dependency for Oracle on "%s". | |
| 122287 | ERROR | Unable to find the Oracle executable "%s" on "%s". | |
| 122288 | ERROR | Usage: %s %s | |
| 122291 | ERROR | Cannot extend resource "%s" to server "%s" | |
| 122292 | ERROR | The values specified for the target and th e template servers are the same: "%s". | |
| 122294 | ERROR | Cannot access canextend script "%s" on server "%s" | |
| 122295 | ERROR | Usage: %s %s | |
| 122296 | ERROR | DB instance "%s" is not protected on "%s". | |
| 122297 | ERROR | Failed to create object instance for Oracle PDB on "%s". | |
| 122298 | ERROR | Unable to locate the oratab file "%s" on "%s". | |
| 122299 | ERROR | END failed hierarchy "%s" of resource "%s" on server "%s" with return value of | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | %d | |
| 122301 | ERROR | Unable to "%s" on "%s" during resource create. | |
| 122302 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | |
| 122304 | ERROR | Unable to "%s" on "%s" during resource create. | |
| 122305 | ERROR | Unable to determine Oracle user for "%s" on "%s". | |
| 122306 | ERROR | Error creating resource "%s" on server "%s" | |
| 122308 | ERROR | Dependency creation between Oracle pluggable database "%s (%s)" and the dependent resource "%s" on "%s" failed. Reason | |
| 122309 | ERROR | %s | |
| 122311 | ERROR | In-service attempted failed for tag "%s" on "%s". | |
| 122312 | ERROR | Usage: %s %s | |
| 122313 | ERROR | Create of app "%s" on "%s" failed with return code of "%d". | |
| 122314 | ERROR | Create of typ "%s" for app "%s" on "%s" failed with return code of "%d". | |
| 122316 | ERROR | Create of resource tag via "newtag" on "%s" failed. | |
| 122318 | ERROR | Error creating resource "%s" on server "%s" | |
| 122320 | ERROR | Setting "resstate" for resource "%s" on "%s" failed with return code of "%d". | |
| 122321 | ERROR | Error creating resource "%s" on server "%s" | |
| 122322 | ERROR | Usage: %s %s | |
| 122323 | ERROR | Usage: %s %s | |
| 122324 | ERROR | Usage: %s %s | |
| 122325 | ERROR | Cannot extend resource "%s" to server | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | "%s" | |
| 122326 | ERROR | Resource with either matching tag "%s" or id "%s" already exists on server "%s" for App "%s" and Type "%s" | |
| 122327 | ERROR | Error creating resource "%s" on server "%s" | |
| 122328 | ERROR | Cannot access extend script "%s" on server "%s" | |
| 122329 | ERROR | Cannot extend resource "%s" to server "%s" | |
| 122330 | ERROR | Usage: %s %s | |
| 122331 | ERROR | Failed to create object instance for Oracle PDB on "%s". | |
| 122332 | ERROR | Usage: %s %s | |
| 122334 | ERROR | Backup node %s is unreachable; abort protection PDB changes. | |
| 122336 | ERROR | Update of protection PDB failed for "%s" on "%s". | |
| 122339 | ERROR | Usage: %s %s | |
| 122340 | ERROR | Usage: %s %s | |
| 122341 | ERROR | Usage: %s %s | |
| 122342 | ERROR | Failed to create object instance for Oracle PDB on "%s". | |
| 122343 | ERROR | Usage: %s %s | |
| 122344 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | |
| 122348 | ERROR | Failed to create flag "%s" on "%s". | |
| 122350 | ERROR | Failed to create object instance for Oracle on "%s". | |
| 122351 | ERROR | no dependency for Oracle on "%s". | |
| 122353 | ERROR | Unable to find the Oracle executable "%s" on "%s". | |
| 122356 | ERROR | Usage: %s %s | |
| 122357 | ERROR | Failed to create object instance for Oracle | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | PDB on "%s". | |
| 122358 | ERROR | The selected oracle SID "%s" is not a CDB. | |
| 122359 | ERROR | No protectable PDB found for the selected SID "%s". | |
| 122360 | ERROR | No protected Oracle database found on "%s". | |
| 122500 | ERROR | Usage: %s %s | **原因**: 作成操作で無効なパラメーターが指定されました。<br><br>**処置**: パラメーターが正しいことを確認し、操作を再実行してください。 |
| 122501 | ERROR | DB instance "%s" is already protected on "%s". | **原因**: {server} で LifeKeeper によってすでに保護されている Oracle データベースインスタンス {sid} を保護しようとしました。<br><br>**処置**: 別のデータベースインスタンス {sid} を LifeKeeper の保護用に選択する必要があります。 |
| 122502 | ERROR | Failed to create object instance for Oracle on "%s". | **原因**: 保護されている Oracle インスタンスの内部表現を作成中に予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122503 | ERROR | Unable to locate the oratab file "%s" on "%s". | **原因**: {server} のデフォルトまたは代替ロケーションで oratab ファイルが見つかりませんでした。<br><br>**処置**: oratab ファイルが存在しており、Oracle ユーザの適切な権限を持っていることを確認してください。「creat |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | e」操作を完了するには有効な oratab ファイルが必要です。 |
| 122504 | ERROR | Unable to determine Oracle user for "%s" on "%s". | 原因: Oracle Application Recovery Kit は Oracle データベースインストールバイナリの所有権を特定できませんでした。<br><br>処置: Oracle バイナリの所有者は {server} の有効な非 root ユーザである必要があります。Oracle データベース所有権の権限と所有権を修正し、操作を再試行してください。 |
| 122505 | ERROR | The Oracle database "%s" is not running or no open connections are available on "%s". | 原因: データベースインスタンス {sid} が実行されていません。またはデータベースへの接続が、指定された認証情報では使用できませんでした。<br><br>処置:「create」操作を完了するには、データベースインスタンス {sid} を {server} で起動し、正しい認証情報を指定する必要があります。 |
| 122506 | ERROR | Unable to determine Oracle dbspaces and logfiles for "%s" on "%s". | 原因: 必要なテーブルスペース、ログファイル、関連データベースファイルの場所を特定するクエリーが失敗しました。これは、内部データベースエラーが原因の可能性もあります。<br><br>処置: 隣接するログメッセージおよび関連エラーで詳細を確認してください。Oracle ログ (alert.log) および関連トレースログ (*.trc) で詳細を確認し、レポートされている問題を修正してください。 |
| 122507 | ERROR | Unknown chunk type found for "%s" on "%s". | 原因: 指定したテーブルスペース、ログファイル、または必要なデータベースファイルが LifeKeeper でサポートされて |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | いるファイルまたはキャラクタデバイスタイプではありません。<br><br>**処置**: 指定したファイル {database_file} は既存のキャラクタデバイスまたはファイルを参照する必要があります。Oracle インストールドキュメントを参照し、指定したファイル {database_file} を、サポートされているファイルまたはキャラクタデバイスタイプとして再作成してください。 |
| 122508 | ERROR | DB Chunk "%s" for "%s" on "%s" does not reside on a shared file system. | **原因**: 指定したテーブルスペース、ログファイル、またはその他の必要なデータベースファイル {database_file} は、クラスター内の他のシステムと共有されているファイルシステムに存在します。<br><br>**処置**: コミュニケーションパスが正常に作成されたことを確認するには、LifeKeeper UI または「lcdstatus (1M)」を使用してください。ストレージ保護に必要なアプリケーションリカバリーキットがインストールされたことを確認するには、「rpm」を使用してください。ファイルが実際には共有ストレージにないことを確認し、ない場合は、共有ストレージデバイスに移動してください。 |
| 122510 | ERROR | File system create failed for "%s" on "%s". Reason: | **原因**: LifeKeeper は、指定したサーバー {server} でリソース {filesystem} を作成できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから「create」操作を再試行する必要があります。 |
| 122511 | ERROR | %s | **原因**: メッセージに「filesyshier」コマンド |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | の結果が含まれています。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 122513 | ERROR | Dependency creation between Oracle database "%s (%s)" and the dependent resource "%s" on "%s" failed.  Reason: | **原因**: LifeKeeper は、データベースリソース {tag} と必要な子リソース {childtag} との間に依存関係を作成できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。問題が修正されている場合は、{tag} と {childtag} との間の依存関係を手動で作成できる可能性があります。 |
| 122514 | ERROR | Unable to "%s" on "%s" during resource create. | **原因**: Oracle Application Recovery Kit は、「rlslocks」コマンドで管理ロックを解除できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |
| 122516 | ERROR | Raw device resource created failed for "%s" on "%s".  Reason: | **原因**: LifeKeeper は、指定したサーバー {server} でリソース {raw device} を作成できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから「create」操作を再試行する必要があります。 |
| 122519 | ERROR | In-service attempted failed for tag "%s" on "%s". | **原因**: {server} の {tag} に対する「perform_action」コマンドがデータベース {sid} の開始に失敗しました。in service 操作に失敗しました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから「create」操作を再試行する必要があります。 |
| 122521 | ERROR | Create of app "%s" on "%s" failed with return code of "%d". | **原因**: 内部アプリケーションタイプを作成する「app_create」コマンドの実行中にエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122522 | ERROR | Create of typ "%s" for app "%s" on "%s" failed with return code of "%d". | **原因**: 内部リソースタイプを作成する「typ_create」コマンドの実行中にエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122524 | ERROR | Setting "resstate" for resource "%s" on "%s" failed with return code of "%d". | **原因**: リソースの状態を {state} に設定する「ins_setstate」コマンドの実行中にエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122525 | ERROR | The values specified for the target and the template servers are the same: "%s". | **原因**:「extend」操作でターゲットサーバーとテンプレートサーバーで指定した値が同じでした。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: {target server} と {template server} に対して正しいパラメーターを指定する必要があります。{target server} は、{tag} が拡張されるサーバーです。 |
| 122526 | ERROR | Unable to locate the oratab file in "/etc" or in "%s" on "%s". | **原因**: {server} のデフォルトまたは代替ロケーションで oratab ファイルが見つかりませんでした。<br><br>**処置**: oratab ファイルが存在しており、Oracle ユーザの適切な権限を持っていることを確認してください。「extend」操作を完了するには有効な oratab ファイルが必要です。 |
| 122527 | ERROR | Unable to retrieve the Oracle user on "%s". | **原因**:「canextend」操作または「extend」操作の間に {template server} から Oracle ユーザを取得できませんでした。<br><br>**処置**: Oracle バイナリの所有者は、{target server} と {template server} の有効なユーザである必要があります。Oracle データベース所有権の権限と所有権を修正し、操作を再試行してください。 |
| 122528 | ERROR | The Oracle user and/or group information for user "%s" does not exist on the server "%s". | **原因**: LifeKeeper は、サーバー {server} の Oracle ユーザ {user} に対して Oracle ユーザおよびグループ情報を検索できません。<br><br>**処置**: 指定した {server} に Oracle ユーザ {user} が存在していることを確認してください。ユーザ {user} が存在しない場合、クラスター内のすべてのサーバーで、同じ uid および gid を使用して作成する必要があります。 |
| 122529 | ERROR | The id for user "%s" is not the same on template server "%s" and target server "%s". | **原因**: ターゲットサーバー {target server} にある Oracle ユーザ {user} のユーザ ID |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | が、テンプレートサーバー {template server} にあるユーザ {user} の値と一致しません。<br><br>**処置**:Oracle ユーザ {user} のユーザ ID はクラスター内のすべてのサーバーで一致している必要があります。ユーザ ID の不一致をすべてのサーバーで手動で修正し、「extend」操作を再試行する必要があります。 |
| 122530 | ERROR | The group id for user "%s" is not the same on template server "%s" and target server "%s". | **原因**: ターゲットサーバー {target server} にある Oracle ユーザ {user} のグループ ID が、テンプレートサーバー {template server} にあるユーザ {user} の値と一致しません。<br><br>**処置**: Oracle ユーザ {user} のグループ ID はクラスター内のすべてのサーバーで一致している必要があります。グループ ID の不一致をすべてのサーバーで手動で修正し、「extend」操作を再試行する必要があります。 |
| 122532 | ERROR | No file system or raw devices found to extend for "%s" on "%s". | **原因**: サーバー {template server} で Oracle リソース {tag} の依存ファイルシステムまたは Raw デバイスリソースが見つかりません。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |
| 122533 | WARN | A RAMDISK (%s) was detected in the ORACLE Database configuration for "%s" on "%s". LifeKeeper cannot protect RAMDISK. This RAMDISK resource will not be protected by LifeKeeper! ORACLE hierarchy creation will continue. | **原因**: 指定したテーブルスペース、ログファイル、またはその他のデータベースファイル {database_file} が ramdisk として検出されました。現在の LifeKeeper 製品では、このタイプのリソースに保護を適用できません。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: ramdisk は保護されません。すべての Oracle データベース操作で必要なデータベースファイル {database_file} が使用可能になるように手動で確認する必要があります。 |
| 122534 | ERROR | Failed to initialize object instance for Oracle sid "%s" on "%s". | **原因**: 保護されている Oracle インスタンスの内部表現を作成中に予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122537 | ERROR | Update of instance info field for "%s" on "%s" failed (%s). | **原因**: 内部リソース情報フィールドを更新する「ins_setinfo」コマンドの実行中にエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122538 | ERROR | Initial connect with query buffer to database "%s" on "%s" failed, testing output. | **原因**: データベースの状態を判別するために Oracle データベース {sid} に接続しようとしたところ、失敗しました。<br><br>**処置**: 指定した認証情報で接続が失敗しました。隣接するログメッセージおよび関連エラーで詳細を確認してください。Oracle ログ (alert.log) および関連トレースログ (*.trc) で詳細を確認し、レポートされている問題を修正してください。 |
| 122542 | ERROR | The "%s [ %s ]" attempt of the database "%s" appears to have failed on "%s". | **原因**: データベースインスタンス {sid} に |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 対して メソッド {action_method} を使用した Oracle アクション {action} がサーバー {server} で失敗しました。<br><br>**処置**: 隣接するログメッセージおよび関連エラーで詳細を確認してください。Oracle ログ (alert.log) および関連トレースログ (*.trc) で詳細を確認し、レポートされている問題を修正してください。 |
| 122543 | ERROR | All attempts to "%s" database "%s" on "%s" failed | **原因**:サーバー {server} の Oracle データベース {sid} に対するアクション {action} を実行するためのあらゆる取り組みが失敗しました。<br><br>**処置**: 隣接するログメッセージおよび関連エラーで詳細を確認してください。Oracle ログ (alert.log) および関連トレースログ (*.trc) で詳細を確認し、レポートされている問題を修正してください。 |
| 122544 | ERROR | Update of "%s" sid "%s" on "%s" failed.  Reason: "%s" "%s" failed: "%s". | **原因**: データベース {sid} の oratab エントリを更新しようとしている際に予期しないエラーが発生しました。このエラーは oratab ファイルを開こうとしている間に発生しました。<br><br>**処置**: システム起動時にデータベースが自動で起動されないようにするために、{sid} の oratab ファイルエントリを手動で更新する必要があります。 |
| 122545 | ERROR | Unable to locate the oratab file in "/etc" or in "%s" on "%s". | **原因**: {server} のデフォルトまたは代替ロケーションで oratab ファイルが見つかりませんでした。<br><br>**処置**: oratab ファイルが存在しており、Oracle ユーザの適切な権限を持っていることを確認してください。「extend」操作を完了するには有効な oratab フ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ァイルが必要です。 |
| 122546 | ERROR | Unable to open file "%s" on "%s" (%s). | **原因**: エラー {error} のため、サーバー {server} では、指定したファイル {file} を開くことも、ファイルにアクセスすることもできませんでした。<br><br>**処置**: 指定したファイル {file} の有無や権限を確認してください。隣接するログメッセージおよび関連エラーで詳細を確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122547 | ERROR | (cleanUpPids):Forcefully killing hung pid(s):pid(s)="%s" | **原因**: プロセス {pid} が正常終了するための要求に応答できませんでした。プロセス {pid} は強制終了します。<br><br>**処置**: コマンドラインで、プロセス {pid} が終了していることを確認してください。隣接するログメッセージで詳細および関連メッセージを確認してください。 |
| 122548 | ERROR | Unable to locate the DB utility (%s/%s) on this host. | **原因**: {path/utility} にある Oracle バイナリおよび必要なデータベースユーティリティ {utility} がこのサーバー {server} では見つかりませんでした。<br><br>**処置**: Oracle バイナリおよび必要なソフトウェアユーティリティがインストールされ、サーバー {server} で適切に設定されていることを確認してください。Oracle バイナリを各ノードにローカルインストールするか、クラスター内の全ノードで使用可能な共有ストレージに配置する必要があります。 |
| 122549 | ERROR | Oracle internal error or non-standard Oracle configuration detected. Oracle User and/or Group set to "root". | **原因**: 検出された Oracle データベースインストールの所有権が、ルートユーザま |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | たはルートグループに解決されています。ルートによる Oracle インストールの所有権は標準的ではない設定です。<br><br>**処置**: Oracle バイナリの所有者は {server} の有効な非 root ユーザである必要があります。Oracle データベース所有権の権限と所有権を修正し、操作を再試行してください。 |
| 122550 | ERROR | Initial inspection of "%s" failed, verifying failure or success of received output. | **原因**: 前の Oracle クエリー {query} またはコマンド {cmd} が成功を返すことができませんでした。<br><br>**処置**: 隣接するログメッセージおよび関連エラーで詳細を確認してください。Oracle ログ (alert.log) および関連トレースログ (*.trc) で詳細を確認し、レポートされている問題を修正してください。 |
| 122551 | ERROR | Logon failed with "%s" for "%s" on "%s". Please check username/password and privileges. | **原因**: 認証情報 {credentials} を使用してサーバー {server} のデータベースインスタンス {sid} へのログオンに失敗しました。無効なユーザ {user} またはパスワードが指定されました。<br><br>**処置**: Oracle データベースユーザ {user} およびパスワード {password} が有効であることを確認してください。また、Oracle データベースユーザ {user} は、試行されるアクションに対して十分な権限を持っている必要があります。 |
| 122552 | ERROR | %s | **原因**: メッセージに「sqlplus」コマンドの結果が含まれています。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122553 | ERROR | Unable to open file "%s" on "%s" (%s). | 原因: エラー {error} のため、サーバー {server} では、指定したファイル {file} を開くことも、ファイルにアクセスすることもできませんでした。<br><br>処置: 指定したファイル {file} の有無や権限を確認してください。隣接するログメッセージおよび関連エラーで詳細を確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122554 | ERROR | The tag "%s" on "%s" is not an Oracle instance or it does not exist. | 原因: サーバー {server} の指定されたタグ {tag} は、既存の有効な Oracle リソースインスタンスを参照していません。<br><br>処置: UI または「lcdstatus (1M)」を使用して、リソースタグ {tag} の有無を確認してください。コマンド「ora-display」を使用するには、リソースタグ {tag} が Oracle リソースインスタンスである必要があります。 |
| 122555 | ERROR | Failed to create object instance for Oracle on "%s". | 原因: Oracle リソースインスタンスの認定ユーザ、パスワード、データベースロールを更新している際に、保護されている Oracle インスタンスの内部表現を作成しようとしたところ、予期しないエラーが発生しました。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122557 | ERROR | Update of user and password failed for "%s" on "%s". | 原因: リソースタグ {tag} のユーザおよびパスワードを更新する要求に失敗しました。指定した認証情報では、サーバー {server} に対する初期検証/接続ができま |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | せんでした。<br><br>**処置**: 目的の操作に対して正しい認証情報 {user/password} が指定されていることを確認してください。隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122559 | ERROR | Update of user and password failed for "%s" on "%s". | **原因**: サーバー {server} のリソースタグ {tag} のユーザおよびパスワード情報の更新に失敗しました。<br><br>**処置**: 目的の操作に対して正しい認証情報 {user/password} が指定されていることを確認してください。隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122562 | ERROR | Unable to find the Oracle executable "%s" on "%s". | **原因**: 必要な Oracle 実行ファイル {exe} がこのサーバー {server} では見つかりませんでした。<br><br>**処置**: Oracle バイナリおよび必要なソフトウェアユーティリティがインストールされ、サーバー {server} で適切に設定されていることを確認してください。Oracle バイナリを各ノードにローカルインストールするか、クラスター内の全ノードで使用可能な共有ストレージに配置する必要があります。 |
| 122566 | ERROR | Unable to find Oracle home for "%s" on "%s". | **原因**: Oracle ホームディレクトリ {Oracle home} には、Oracle インスタンス {sid} の正常な動作に必要なファイルが含まれていないようです。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: Oracle ホームディレクトリ {Oracle home} に Oracle バイナリ、有効な spfile{sid}.ora、または init{sid}.ora ファイルが含まれていることをコマンドラインで確認してください。 |
| 122567 | ERROR | Oracle SID mismatch. The instance SID "%s" does not match the SID "%s" specified for the command. | **原因**: 保護されている Oracle インスタンスの内部表現を作成中に予期しないエラーが発生しました。指定した内部 ID {id} が、予想した SID {sid} と一致しません。<br><br>**処置**: パラメーターが正しいことを確認してください。隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122568 | ERROR | DB Processes are not running on "%s". | **原因**: Oracle インスタンスのプロセスチェックで、サーバー {server} で実行されているプロセスは見つかりませんでした。<br><br>**処置**: ローカルリカバリーが有効になっている場合、Oracle インスタンスはローカルで再起動されます。隣接するログメッセージで詳細および関連メッセージを確認してください。 |
| 122572 | ERROR | Failed to create flag "%s" on "%s". | **原因**: スタンバイノードへのフェイルオーバーを促す Oracle ローカルリカバリー処理を制御するためのフラグを作成しようとして、予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122574 | ERROR | all attempts to shutdown the database %s failed on "%s". | **原因**: ローカルリカバリープロセス中に、おそらくデータベース接続数が上限に達したため、Oracle データベースのシャットダウンに失敗しました。<br><br>**処置**: Oracle ログで、許容接続数の上限に達したことによって起こった接続の障害の有無を確認してください。その障害が見つかった場合は、接続数を増やしてください。また、ローカルリカバリーの実行による quickCheck の障害発生が、接続の失敗を引き起こさないように、LK_ORA_NICE を 1 に設定してください。 |
| 122597 | ERROR | Failed to create object instance for Oracle on "%s". | **原因**: 保護されている Oracle インスタンスの内部表現を作成中、拡張前のチェックの際に予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから、pre-extend を再試行してください。 |
| 122598 | ERROR | Failed to create object instance for Oracle on "%s". | **原因**: 作成中の Oracle インスタンスの内部表現を作成しているときに、Oracle ホームディレクトリの有効性を確認しようとして、予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから、「create」を再試行してください。 |
| 122599 | ERROR | Failed to create object instance for Oracle on "%s". | **原因**: 保護中の Oracle インスタンスの内部表現を作成しているときに、テンプレ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ートシステムで Oracle ユーザをルックアップしようとして、予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから、extend を再試行してください。 |
| 122600 | ERROR | Failed to create object instance for Oracle on "%s". | **原因**: 保護中の Oracle インスタンスの内部表現を作成しているときに、リソースプロパティを表示しようとして、予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから、リソースプロパティの表示を再試行してください。 |
| 122601 | ERROR | Failed to create object instance for Oracle on "%s". | **原因**: 保護中の Oracle インスタンスの内部表現を作成しているときに、有効なデータベース認証を確認しようとして、予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから、コマンドを再試行してください。 |
| 122603 | ERROR | Failed to create object instance for Oracle on "%s". | **原因**: 保護中の Oracle インスタンスの内部表現を作成しているときに、Oracle リソースインスタンスに対してヘルスチェックを実行しようとして、予期しないエラーが発生しました。<br><br>**処置**: 正しい引数が quickCheck コマンドに渡されていることを確認し、隣接す |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | るログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから、restore を再試行してください。 |
| 122604 | ERROR | Failed to create object instance for Oracle on "%s". | **原因**: 保護中の Oracle インスタンスの内部表現を作成しているときに、Oracle リソースインスタンスに対してローカルリカバリーを実行しようとして、予期しないエラーが発生しました。<br><br>**処置**: 正しい引数が「recover」コマンドに渡されていることを確認し、隣接するログメッセージと関連メッセージで詳細を確認してください。レポートされているエラーを修正してから、recover を再試行してください。 |
| 122606 | ERROR | The Oracle database "%s" is not running or no open connections are available on "%s". | **原因**: データベースインスタンス {sid} が実行されていません。またはデータベースへの接続が、指定された認証情報では使用できません。<br><br>**処置**: 選択した操作を完了するには、データベースインスタンス {sid} を {server} で起動し、正しい認証情報を指定する必要があります。 |
| 122607 | ERROR | The Oracle database "%s" is not running or no open connections are available on "%s". | **原因**: データベースインスタンス {sid} が実行されていません。またはデータベースへの接続が、指定された認証情報では使用できません。<br><br>**処置**: 選択した操作を完了するには、データベースインスタンス {sid} を {server} で起動し、正しい認証情報を指定する必要があります。 |
| 122608 | ERROR | Failed to create object instance for Oracle on "%s". | **原因**:「remove」操作で、Oracle リソー |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | スを Out of Service にするために必要なリソースオブジェクトインスタンスを作成できませんでした。<br><br>**処置**: 正しい引数が「remove」コマンドに渡されていることを確認し、隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから、restore を再試行してください。 |
| 122609 | ERROR | Failed to create object instance for Oracle on "%s". | **原因**:「restore」操作で、Oracle リソースを In Service にするために必要なリソースオブジェクトインスタンスを作成できませんでした。<br><br>**処置**: 正しい引数が「restore」コマンドに渡されていることを確認し、隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから、「restore」を再試行してください。 |
| 122610 | ERROR | Unable to "%s" on "%s" during resource create. | **原因**: Oracle Application Recovery Kit は、リソース作成中に「getlocks」コマンドで管理ロックを作成できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してから、create を再試行してください。 |
| 122611 | ERROR | %s | **原因**: 要求された親 Oracle リソースと子 File System リソースとの間の依存関係の作成に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してから、creat |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | e 操作を再試行してください。 |
| 122612 | ERROR | %s | **原因**: 要求された親 Oracle リソースと子 Raw リソースとの間の依存関係の作成に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してから、create 操作を再試行してください。 |
| 122613 | ERROR | %s | **原因**: 要求された親 Oracle リソースと子 Raw リソースとの間の依存関係の作成に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してから、create 操作を再試行してください。 |
| 122614 | ERROR | %s | **原因**: 要求された親 Oracle リソースと子 Listener リソースとの間の依存関係の作成に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してから、create 操作を再試行してください。 |
| 122616 | ERROR | %s | **原因**: Oracle データベースの要求された起動またはシャットダウンに失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してから、「restore」または「remove」操作を再試行してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122618 | ERROR | Dependency creation between Oracle database "%s (%s)" and the dependent resource "%s" on "%s" failed.  Reason: | 原因: LifeKeeper は、データベースリソース {tag} と必要な子リソース {childtag} との間に依存関係を作成できませんでした。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。問題が修正されている場合は、{tag} と {childtag} との間の依存関係を手動で作成できる可能性があります。 |
| 122619 | ERROR | Dependency creation between Oracle database "%s (%s)" and the dependent resource "%s" on "%s" failed.  Reason: | 原因: LifeKeeper は、データベースリソース {tag} と必要な子リソース {childtag} との間に依存関係を作成できませんでした。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。問題が修正されている場合は、{tag} と {childtag} との間の依存関係を手動で作成できる可能性があります。 |
| 122625 | ERROR | Unable to find the Oracle executable "%s" on "%s". | 原因: quickCheck プロセスが、Oracle の実行可能ファイル「sqlplus」を見つけることができませんでした。<br><br>処置: Oracle 設定を確認し、隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされている問題を修正してください。 |
| 122626 | ERROR | Unable to find the Oracle executable "%s" on "%s". | 原因: 削除プロセスが、Oracle の実行可能ファイル「sqlplus」を見つけることができませんでした。<br><br>処置: Oracle 設定を確認し、隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされている問題を修正してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122627 | ERROR | Unable to find the Oracle executable "%s" on "%s". | **原因**: リストアプロセスが、Oracle の実行可能ファイル「sqlplus」を見つけることができませんでした。<br><br>**処置**: Oracle 設定を確認し、隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされている問題を修正してください。 |
| 122628 | ERROR | Unable to find the Oracle executable "%s" on "%s". | **原因**: 復旧プロセスが、Oracle の実行可能ファイル「sqlplus」を見つけることができませんでした。<br><br>**処置**: Oracle 設定を確認し、隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされている問題を修正してください。 |
| 122632 | ERROR | Oracle SID mismatch. The instance SID "%s" does not match the SID "%s" specified for the command. | **原因**: 削除の際に、削除プロセスに渡されるリソースインスタンス {sid} が、{sid} の内部リソースインスタンス情報と一致しません。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 122633 | ERROR | Oracle SID mismatch. The instance SID "%s" does not match the SID "%s" specified for the command. | **原因**: リストアの際に、restore に渡されるリソースインスタンス {sid} が、{sid} の内部リソースインスタンス情報と一致しません。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 122634 | ERROR | Oracle SID mismatch. The instance SID "%s" does not match the SID "%s" specified for the command. | **原因**: リソースリカバリーの際に、recovery に渡されるリソースインスタンス {sid} が、{sid} の内部リソースインスタン |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ス情報と一致しません。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 122636 | ERROR | END failed hierarchy "%s" of resource "%s" on server "%s" with return value of %d | **原因**:{server} で Oracle リソース階層 {tag} の作成に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 122638 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | **原因**:サーバー {server} の Oracle データベースリソース {tag} で作成アクションに失敗しました。シグナル {sig} が作成プロセスによって受信されました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122640 | ERROR | Error creating resource "%s" on server "%s" | **原因**: Oracle リソースインスタンス {tag} を {server} に作成しようとして、予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 122641 | ERROR | Error creating resource "%s" on server "%s" | **原因**: Oracle リソースインスタンス {tag} を {server} に作成しようとして、予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122642 | ERROR | Error creating resource "%s" on server "%s" | 原因: Oracle リソースインスタンス {tag} を {server} に作成しようとして、予期しないエラーが発生しました。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 122643 | ERROR | Cannot extend resource "%s" to server "%s" | 原因: LifeKeeper は、{server} でリソース {resource} を拡張できませんでした。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 122644 | ERROR | Error getting resource information for resource "%s" on server "%s" | 原因: リソースインスタンス情報 {tag} を {server} で取得しようとして、予期しないエラーが発生しました。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正し、extend を再試行してください。 |
| 122645 | ERROR | Cannot access canextend script "%s" on server "%s" | 原因: {server} で依存関係の子リソース用の「canextend」スクリプトが見つからなかったので、LifeKeeper は拡張前のチェックを実行できませんでした。<br><br>処置: LifeKeeper の設定を確認してください。 |
| 122646 | ERROR | Error getting resource information for resource "%s" on server "%s" | 原因: リソースインスタンス情報 {tag} を {server} で取得しようとして、予期しないエラーが発生しました。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認し、レポート |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | されているエラーを修正し、extend を再試行してください。 |
| 122647 | ERROR | Cannot extend resource "%s" to server "%s" | **原因**: LifeKeeper は、{server} でリソース {resource} を拡張できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 122648 | ERROR | Resource with either matching tag "%s" or id "%s" already exists on server "%s" for App "%s" and Type "%s" | **原因**: データベースリソースの拡張中に、同じ {tag} または {id} で異なるリソースアプリケーションおよびタイプを使用しているリソースインスタンスが見つかりました。<br><br>**処置**: リソース ID は一意のものにする必要があります。Oracle リソースインスタンスに合致する ID のリソースインスタンスを削除してください。 |
| 122649 | ERROR | Error creating resource "%s" on server "%s" | **原因**: Oracle リソースインスタンス {tag} を {server} に作成しようとして、予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 122650 | ERROR | Cannot access extend script "%s" on server "%s" | **原因**: 依存関係にある子リソースのスクリプト {extend} がサーバー {server} で見つけられなかったので、サーバー {server} にデータベースリソース {resource} を拡張する要求が失敗しました。<br><br>**処置**: LifeKeeper の設定を確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122651 | ERROR | Cannot extend resource "%s" to server "%s" | **原因**: 依存関係にある子リソースの拡張でエラーが発生したため、サーバー {server} にデータベースリソース {resource} を拡張する要求が失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 122654 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | **原因**: quickCheck プロセスがシグナルを受信したため、データベース {sid} のヘルスチェックが終了しました。これはおそらく、quickCheck プロセスの処理に、割り当てられているよりも長い時間が必要となっているためです。<br><br>**処置**: Oracle リソースのヘルスチェック時間は ORACLE_QUICKCHECK_TIMEOUT の設定で制御してください。ヘルスチェックプロセスの完了に今より長い時間を割り当てる場合は、45 秒よりも大きな値に設定してください。 |
| 122655 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | **原因**: 削除プロセスがシグナルを受信したため、データベース {sid} を「Out of Service」にする要求が終了しました。これはおそらく、削除プロセスの処理に、割り当てられているよりも長い時間が必要となっているためです。<br><br>**処置**: Oracle リソースの削除時間は ORACLE_REMOVE_TIMEOUT の設定で制御してください。削除プロセスの完了に今より長い時間を割り当てる場合は、240 秒よりも大きな値に設定してください。 |
| 122659 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | **原因**: リストアプロセスがシグナルを受信したため、データベース {sid} を「In Service」にする要求が終了しました。こ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | れはおそらく、リストアプロセスの処理に、割り当てられているよりも長い時間が必要となっているためです。<br><br>**処置**: Oracle リソースのリストア時間は ORACLE_RESTORE_TIMEOUT の設定で制御してください。リストアプロセスの完了に今より長い時間を割り当てる場合は、240 秒よりも大きな値に設定してください。 |
| 122663 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | **原因**: リカバリープロセスがシグナルを受信したため、失敗したデータベースのリカバリーが終了しました。これはおそらく、リカバリープロセスの処理に、割り当てられているよりも長い時間が必要となっているためです。<br><br>**処置**: Oracle リソースのリカバリー時間は ORACLE_RESTORE_TIMEOUT と ORACLE_REMOVE_TIMEOUT の設定で制御してください。リカバリーの完了に今より長い時間を割り当てる場合は、一方または両方を 240 秒よりも大きな値に設定してください。 |
| 122670 | ERROR | Update of "%s" sid "%s" on "%s" failed. Reason: "%s" "%s" failed: "%s". | **原因**: データベース {sid} の oratab エントリを更新しようとしている際に予期しないエラーが発生しました。更新プロセスで使用する一時ファイルを開こうとして、エラーが発生しました。<br><br>**処置**: システム起動時にデータベースが自動で起動されないようにするために、{sid} の oratab ファイルエントリを手動で更新する必要があります。 |
| 122671 | ERROR | Update of "%s" sid "%s" on "%s" failed. Reason: "%s" "%s" failed: "%s". | **原因**: データベース {sid} の oratab エントリを更新しようとしている際に予期しないエラーが発生しました。更新プロセ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | スで使用する一時ファイルを閉じようとして、エラーが発生しました。<br><br>**処置**: システム起動時にデータベースが自動で起動されないようにするために、{sid} の oratab ファイルエントリを手動で更新する必要があります。 |
| 122672 | ERROR | Update of "%s" sid "%s" on "%s" failed.  Reason: "%s" "%s" failed: "%s". | **原因**: データベース {sid} の oratab エントリを更新しようとしている際に予期しないエラーが発生しました。一時ファイルの名前を oratab に変更しようとして、エラーが発生しました。<br><br>**処置**: システム起動時にデータベースが自動で起動されないようにするために、{sid} の oratab ファイルエントリを手動で更新する必要があります。 |
| 122673 | ERROR | Unable to log messages queued while running as oracle user %s on %s. Reason: $! | **原因**: メッセージをログファイルに追加しようとして、予期しないエラー {reason} が発生しました。これらのメッセージは、Oracle ユーザとして実行している間に生成されました。<br><br>**処置**: 障害に関する理由を参照し、修正処置を実行してください。 |
| 122674 | ERROR | Unable to open %s Reason: %s. | **原因**: Oracle データベースへの接続を開き、データベース {cmd} を実行しようとしたときに、予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。また、Oracle ログ (alert.log) および関連トレースログ (*.trc) で詳細を確認し、レポートされている問題を修正してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122680 | ERROR | Unable to find Oracle home for “%s” on “%s”. | 原因: Oracle ホームディレクトリ {Oracle home} には、Oracle インスタンス {sid} の正常な動作に必要なファイルが含まれていないようです。<br><br>処置: Oracle ホームディレクトリ {Oracle home} に Oracle バイナリ、有効な spfile{sid}.ora、または init{sid}.ora ファイルが含まれていることをコマンドラインで確認してください。 |
| 122681 | ERROR | Failed to create object instance for Oracle on "%s". | 原因: 保護されている Oracle インスタンスの内部表現を作成中に予期しないエラーが発生しました。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122682 | ERROR | Unable to find the Oracle executable "%s" on "%s". | 原因: 必要な Oracle 実行ファイル {exe} がこのサーバー {server} では見つかりませんでした。<br><br>処置: Oracle バイナリおよび必要なソフトウェアユーティリティがインストールされ、サーバー {server} で適切に設定されていることを確認してください。Oracle バイナリを各ノードにローカルインストールするか、クラスター内の全ノードで使用可能な共有ストレージに配置する必要があります。 |
| 122683 | ERROR | Backup node %s is unreachable; abort username/password changes. | 原因: バックアップノード {node} に接続できません。保留中のユーザー名/パスワードに対する変更は中断されました。<br><br>処置: 指定したバックアップノード {node} 上でSIOS LifeKeeper が起動中で、す |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | べてのコミュニケーションパスが確立されていることが確認できたら、再度操作を行ってください。 |
| 122684 | ERROR | The restore of %s has timed out on server %s. The default TIMEOUT is 300 seconds. To increase the TIMEOUT, set ORACLE_RESTORE_TIMEOUT in /etc/default/LifeKeeper. | **原因**: サーバー {server} 上で Oracle リソース {resource} の restore 処理がタイムアウトしました。これは、In Service 処理が設定された時間を経過すると発生します。<br><br>**処置**: Oracle リソースのリストア時間は ORACLE_RESTORE_TIMEOUT の設定で制御してください。リストアプロセスの完了に今より長い時間を割り当てる場合は、240 秒よりも大きな値に設定してください。 |
| 122685 | ERROR | The remove of %s has timed out on server %s. The default TIMEOUT is 300 seconds. To increase the TIMEOUT, set ORACLE_REMOVE_TIMEOUT in /etc/default/LifeKeeper. | **原因**: サーバー {server} 上で Oracle リソース {resource} の remove 処理がタイムアウトしました。 これは、Out of Service 処理が設定された時間を経過すると発生します。<br><br>**処置**: Oracle リソースの削除時間は ORACLE_REMOVE_TIMEOUT の設定で制御してください。削除プロセスの完了に今より長い時間を割り当てる場合は、240 秒よりも大きな値に設定してください。 |
| 122686 | ERROR | The quickCheck of %s has timed out on server %s. The default TIMEOUT is 45 seconds. To increase the TIMEOUT, set ORACLE_QUICKCHECK_TIMEOUT in /etc/default/LifeKeeper. | **原因**: サーバー {server} 上で Oracle リソース {resource} に対する quickCheck がタイムアウトしました。 これは、health check 処理が設定された時間を経過すると発生します。<br><br>**処置**: Oracle リソースのヘルスチェック時間は ORACLE_QUICKCHECK_TIMEOUT の設定で制御してください。ヘルスチェックプロセスの完了に今より長い |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 時間を割り当てる場合は、45 秒よりも大きな値に設定してください。 |
| 122687 | ERROR | Usage: %s %s | 原因: getalertlog スクリプトに対して無効なパラメーターが指定されました。<br><br>処置: Oracle リソースのタグ名が、最初のコマンドラインのパラメーターとして提供される必要があります。パラメーターが正しいかどうかを確認し、再度操作を行ってください。 |
| 122702 | ERROR | Failed start OHAS. | |
| 122704 | ERROR | Failed start ASM. | |
| 122706 | ERROR | Failed start Diskgroup. | |
| 122708 | ERROR | Failed stop Diskgroup. | |
| 122710 | ERROR | Failed stop ASM. | |
| 122712 | ERROR | Failed stop OHAS. | |
| 122713 | ERROR | OHAS is not running. | |
| 122714 | ERROR | ASM is not running. | |
| 122715 | ERROR | Diskgroup is not running. | |
| 122716 | ERROR | $usage | |
| 122717 | ERROR | Template resource \"$template_tag\" on s erver \"$template_sys\" does note exist | |
| 122718 | ERROR | Cannot extend resource \"$template_tag\" to server \"$me\" | |
| 122719 | ERROR | $usage | |
| 122720 | ERROR | END failed create of \"$Tag\" due to a \"$sig\" signal | |
| 122722 | ERROR | Unable to getlocks on $me during resouc e create. | |
| 122725 | ERROR | Error creating resource $Tag. Error ($rc | |
| 122726 | ERROR | END failed hierarchy create of resource $Tag with return value of $ecode. | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122728 | ERROR | Unable to rlslocks on $me during resource create. | |
| 122729 | ERROR | $usage | |
| 122730 | ERROR | END failed extend of \"$Tag\" due to a \"$sig\" signal | |
| 122732 | ERROR | Template resource \"$TemplateTag\" on server \"$TemplateSys\" does not exist | |
| 122733 | ERROR | Error creating resouce \"$Tag\"on server \"$me\" | |
| 122735 | ERROR | END failed hierarchy extend of resource $Tag with return value of $ecode. | |
| 122736 | ERROR | $usage | |
| 122737 | ERROR | Failed to create object instance for Oracle ASM on $oracle::me | |
| 122738 | ERROR | Usage : $cmd $usage | |
| 122739 | ERROR | Failed to create object instance for Oracle ASM on \"$me\". | |
| 122740 | ERROR | $usage | |
| 122742 | ERROR | backup node $back_dead_name is unreachable; abort protection Diskgroup changes. | |
| 122744 | ERROR | Update of protection Diskgroup failed for \"$tag\" on \"$oarcleasm::me\" | |
| 122746 | ERROR | Update of protection Diskgroup failed foor \"$tag\" on \"$remote\" | |
| 122750 | ERROR | Failed to init the object. | |
| 122751 | ERROR | Usage: %s %s | |
| 123006 | FATAL | Unknown version %s of IP address | 原因: IP アドレスが IPv4 でも IPv6 でも有効ではないようです。<br><br>処置: 有効な IP アドレスを指定してください。 |
| 123008 | ERROR | No pinglist found for %s. | 原因: この IP アドレスの pinglist を開こ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | うとして問題が発生しました。<br><br>**処置**: この IP アドレスの pinglist を指定していることを確認してください。 |
| 123009 | ERROR | List ping test failed for virtual IP %s | **原因**: ping リストのどのアドレスからも応答を受信しませんでした。<br><br>**処置**: このノードのネットワーク接続および ping リストの IP が存在するシステムを確認してください。 |
| 123013 | ERROR | Link check failed for virtual IP %s on interface %s. | **原因**: 要求したインターフェースが「NO-CARRIER」となっています。これは、物理レイヤー接続にリンクが存在しないことを示します。<br><br>**処置**: インターフェースの物理接続を確認し、物理レイヤーリンクを有効にしてください。 |
| 123015 | ERROR | Link check failed for virtual IP %s on interface %s. | **原因**: 要求したインターフェースは bonding インターフェースで、スレーブのいずれか 1 つが「NO-CARRIER」です。これは、物理レイヤー接続にリンクが存在しないことを示します。<br><br>**処置**: スレーブインターフェースの物理接続を確認し、物理レイヤーリンクを有効にしてください。 |
| 123024 | ERROR | IP address seems to still exist somewhere else. | **原因**: IP アドレスはネットワークの別の場所で使用中のようです。<br><br>**処置**: 別の IP アドレスを選択して使用するか、この IP アドレスを特定して、現在の使用を無効にしてください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 123037 | ERROR | must specify machine name containing primary hierarchy | **原因**: creIPhier に指定した引数が不十分です。<br><br>**処置**: 必要な引数をすべて creIPhier に指定してください。 |
| 123038 | ERROR | must specify IP resource name | **原因**: creIPhier に渡された引数が不十分です。<br><br>**処置**: 必要な引数をすべて creIPhier に指定してください。 |
| 123039 | ERROR | must specify primary IP Resource tag | **原因**:「creIPhier」コマンドにプライマリ IP リソースタグの引数が指定されていません。<br><br>**処置**: 必要な引数をすべて指定してください。 |
| 123042 | ERROR | An unknown error has occurred in utility validmask on machine %s. | **原因**:「validmask」ユーティリティの実行中に予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 123045 | ERROR | An unknown error has occurred in utility getlocks. | **原因**:「getlocks」ユーティリティの実行中に予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 123053 | ERROR | Cannot resolve hostname %s | **原因**: IP アドレスに対してホスト名が指定されましたが、システムはホスト名を IP アドレスに解決できませんでした。<br><br>**処置**: ホスト名が正確かどうかを確認してから、名前の解決（DIS または /etc/hosts）が正常に機能しており、ホスト名 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | の IP を返すことを確認してください。 |
| 123055 | ERROR | An unknown error has occurred in utility %s on machine %s. | **原因**: IP リソースの作成中に障害が発生しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 123056 | ERROR | create ip hierarchy failure: perform_action failed | **原因**:作成中に IP アドレスをリストアしようとして、予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 123059 | ERROR | Resource already exists on machine %s | **原因**: すでに存在する IP アドレスを作成しようとしました。<br><br>**処置**: 既存のリソースを再利用するか、すでに存在している IP アドレスを手動で削除するか、別の IP アドレスを使用してください。 |
| 123060 | ERROR | ins_create failed on machine %s | **原因**: IP リソースの作成中に予期しない障害が発生しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 123064 | ERROR | An unknown error has occurred in utility %s on machine %s. | **原因**: IP リソースの依存関係を作成中に障害が発生しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 123066 | ERROR | An error occurred during creation of LifeKeeper application=comm on %s. | 原因:「app_create」の呼び出し中に障害が発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 123068 | ERROR | An error occurred during creation of LifeKeeper resource type=ip on %s. | 原因:「typ_create」の呼び出し中に障害が発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 123089 | ERROR | Link check failed for virtual IP %s on interface %s. | 原因: コマンド "nmcli connection show —active" インターフェイスがアクティブであることを返しませんでした。<br><br>処置: エラーログを確認してください。 |
| 123091 | ERROR | the link for interface %s is down | 原因: 要求したインターフェースが「NO-CARRIER」となっています。これは、物理レイヤー接続にリンクが存在しないことを示します。<br><br>処置: インターフェースの物理接続を確認し、物理レイヤーリンクを有効にしてください。 |
| 123093 | ERROR | the ping list check failed | 原因: ping リストのどのアドレスからも応答を受信しませんでした。<br><br>処置: このノードのネットワーク接続および ping リストの IP が存在するシステムを確認してください。 |
| 123094 | ERROR | IP address is not assigned to interface %s. | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 123095 | ERROR | broadcast ping failed | **原因**: ブロードキャスト ping から応答を受信しませんでした。<br><br>**処置**: サブネット上の 1 つ以上のホストがブロードキャスト ping に応答することを確認してください。仮想 IP が正しいネットワークインターフェースにあることを確認してください。ブロードキャスト ping ではなく、pinglist の使用を検討してください。 |
| 123096 | ERROR | $msg | **原因**: 仮想 IP の実行可能性の判定に使用するブロードキャスト ping に失敗しました。<br><br>**処置**: このリソースの ping リストがプロパティパネルで正常に設定されていることを確認するか、/etc/default/LifeKeeper 設定ファイルに NOBCASTPING=1 を追加してブロードキャスト ping のチェックを確実に無効にしてください。 |
| 123097 | ERROR | exec_list_ping(): broadcast ping failed. | **原因**: 仮想 IP の実行可能性の判定に使用するブロードキャスト ping に失敗しました。<br><br>**処置**: このリソースの ping リストがプロパティパネルで正常に設定されていることを確認するか、/etc/default/LifeKeeper 設定ファイルに NOBCASTPING=1 を追加してブロードキャスト ping のチェックを確実に無効にしてください。 |
| 123102 | WARN | IP is already configured with wrong netmask: expected %s, found %s. | **原因**: IP が間違ったネットマスクで構成されています。<br><br>**処置**: 間違ったネットマスクを使用して、LifeKeeper の外部で IP が追加されました。 コマンドラインから手動で IP を開始しないでください。また、起動時 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | に IP を設定する ifcfg ファイルが sysconfig にないことを確認してください。 |
| 123299 | ERROR | Unable to open %s. Reason %s | |
| 123410 | ERROR | Usage error OSUquickCheck | **原因**: タグまたは ID なしで OSUquickCheck が呼び出されました。<br><br>**処置**: 有効なタグまたは ID を指定して OSUquickCheck を呼び出します。 |
| 123411 | ERROR | OSUquickCheck: both tag and id name not specified | **原因**: OSUquickCheck には、タグまたはリソース ID が必要です。 |
| 123412 | ERROR | resource $Tag not found on local server | **原因**: OSUquickCheck は、タグまたは ID のリソースを見つけることができませんでした。 |
| 123414 | ERROR | The link for network interface $IPObj->{'device'} is down | **原因**: リンクダウンしており、リンクアップしようとして失敗しました。 |
| 123415 | ERROR | No pinglist found for $IPObj->{'ipaddr'} | |
| 123416 | ERROR | List ping test failed for virtual IP $IPObj->{'ipaddr'} | |
| 124004 | FATAL | resource tag name not specified | **原因**:「quickCheck」操作で無効な引数が指定されました。<br><br>**処置**: 正しい引数が渡されたことを確認してください。 |
| 124005 | FATAL | resource id not specified | **原因**:「quickCheck」操作で無効な引数が指定されました。<br><br>**処置**: 正しい引数が渡されたことを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 124007 | FATAL | Failed to get resource information | **原因**: ファイルシステムリソースの情報フィールドに正しい情報が含まれていません。<br><br>**処置**: リソースの情報フィールドに正しい情報を入力するか、最近の「lkbackup」からシステムをリストアして元の情報フィールドをリストアします。 |
| 124008 | ERROR | getId failed | **原因**: ファイルシステムリソースは基本ディスクデバイスを見つけられませんでした。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。リソース階層が有効であること、および必要なストレージキットがすべてインストールされていることを確認してください。 |
| 124009 | ERROR | LifeKeeper protected filesystem is in service but quickCheck detects the following error: | **原因**: ファイルシステムキットはリソースに問題を発見しました。<br><br>**処置**: この直後に表示されるメッセージで詳細を確認してください。 |
| 124010 | ERROR | "$id" is not mounted | **原因**: ファイルシステムリソースがマウントされていません。<br><br>**処置**: 処置は必要ありません。ローカルリカバリーでリソースの再マウントを許可してください。 |
| 124011 | ERROR | "$id" is mounted but with the incorrect mount options (current mount option list: $mntopts, expected mount option list: $infoopts) | **原因**: ファイルシステムリソースが不正にマウントされています。<br><br>**処置**: 処置は必要ありません。ローカルリカバリーでリソースの再マウントを許可してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 124012 | ERROR | "$id" is mounted but on the wrong device (current mount device: $tmpdev, expected mount device: $dev) | **原因**: ファイルシステムリソースに誤ったデバイスがマウントされています。<br><br>**処置**: 処置は必要ありません。ローカルリカバリーでリソースの再マウントを許可してください。 |
| 124015 | ERROR | LifeKeeper protected filesystem "$tag" ($id) is $percent% full ($blocksfree free blocks). | **原因**: ディスク使用量が FILESYSFULLERROR の値（%）を超過しました。<br><br>**処置**: ファイルシステムからデータを削除または移行してください。 |
| 124016 | WARN | LifeKeeper protected filesystem "$tag" ($id) is $percent% full ($blocksfree free blocks). | **原因**: ディスク使用量が FILESYSFULL WARNの値（%）を超過しました。<br><br>**処置**: ファイルシステムからデータを削除または移行してください。 |
| 124020 | FATAL | cannot find device information for filesystem $id | **原因**: ファイルシステムリソースは基本ディスクデバイスを見つけられませんでした。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。リソース階層が有効であること、および必要なストレージキットがすべてインストールされていることを確認してください。 |
| 124029 | ERROR | Failed to find child resource. | **原因**: ファイルシステムリソースは基本ディスクリソースを特定できませんでした。<br><br>**処置**: リソース階層が正しいことを確認してください。 |
| 124032 | FATAL | Script has hung. Exiting. | **原因**: アンマウントする必要があるものの、マウントされているファイルシステ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ムに開かれているファイルが複数のプロセスにありました。それらのプロセスの終了に時間がかかりすぎています。<br><br>**処置**: このエラーが続く場合は、マウントポイントを使用している可能性のあるすべてのソフトウェアを一時的に停止し、アンマウントできるようにしてください。ファイルシステムをそれでもアンマウントできない場合は、サポートに問い合わせてください。 |
| 124042 | ERROR | file system $fs failed unmount; will try again | **原因**: アンマウントする必要があるものの、マウントされているファイルシステムに開かれているファイルが複数のプロセスにありました。場合によっては、それらのプロセスを消去するために、何度か操作を実行する必要があります。<br><br>**処置**: 処置は必要ありません。プロセスをそのまま続行してください。 |
| 124046 | ERROR | file system $fsname failed unmount | **原因**: ファイルシステムをアンマウントできませんでした。<br><br>**処置**: このエラーが続く場合は、マウントポイントを使用している可能性のあるすべてのソフトウェアを一時的に停止し、アンマウントできるようにしてください。ファイルシステムをそれでもアンマウントできない場合は、サポートに問い合わせてください。 |
| 124049 | ERROR | Local recovery of resource has failed (err=$err) | **原因**: ファイルシステムリソースに、ローカルでは修正できない問題があります。<br><br>**処置**: 処置は必要ありません。リソースを別のシステムにフェイルオーバーできるようにしてください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 124051 | WARN | getId failed, try count : $cnt/$try | **原因**: ファイルシステムの quickCheck は、デバイスリソースの基になるデバイス名を取得できませんでした。<br><br>**処置**: 正しいデバイスがマウントされていることを確認してください。 |
| 124052 | ERROR | \"$id\" is mounted but filesystem is shutdown state. | **原因**: xfs_info コマンドが失敗しました。<br><br>**処置**: ファイルシステムがマウントされていることを確認してください。 システムエラーログでエラーを確認してください。 |
| 124054 | ERROR | Failed to change mount option from \"$old_opts\" to \"$new_opts\" for migration | **原因**: マウントオプションは移行されませんでした。<br><br>**処置**: エラーログを確認してください。 |
| 124103 | ERROR | $ERRMSG Script was terminated for unknown reason | **原因**: このメッセージは、通常の状態では表示されません。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124104 | ERROR | $ERRMSG Required template machine name is null | **原因**: canextend 操作で無効な引数が指定されました。<br><br>**処置**: 引数が正しいことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124105 | ERROR | $ERRMSG Required template resource tag name is null | **原因**: canextend 操作で無効な引数が指定されました。<br><br>**処置**: 引数が正しいことを確認してくだ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | さい。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124106 | ERROR | $ERRMSG Unable to access template resource "$TemplateTagName" | **原因**: リソースの基本ディスク情報を特定できませんでした。<br><br>**処置**: テンプレートシステムで階層が正しいことを確認してから拡張してください。 |
| 124107 | ERROR | $ERRMSG Resource "$TemplateTagName" must have one and only one device resource dependency | **原因**: リソースの階層にある基本デバイスが多すぎます。<br><br>**処置**: テンプレートシステムで階層が正しいことを確認してから拡張してください。 |
| 124108 | ERROR | $ERRMSG Unable to access template resource "$TemplateTagName" | **原因**: テンプレートシステムでリソースを見つけられません。<br><br>**処置**: テンプレートシステムで階層が正しいことを確認してから拡張してください。 |
| 124109 | ERROR | $ERRMSG Can not access canextend for scsi/$DeviceResType resources on machine "$TargetSysName" | **原因**: ターゲットシステムに、必須のコンポーネントがいくつか見つかりません。<br><br>**処置**: ターゲットシステムに正しいキットがすべてインストールされ、ライセンスされていることを確認してください。 |
| 124110 | ERROR | $ERRMSG Either filesystem "$TemplateLKId" is not mounted on "$TemplateSysName" or filesystem is not shareable with "$TargetSysName" | **原因**: ファイルシステムがテンプレートシステムのサービス内にないか、ターゲットシステムに拡張するための要件を満たしていません。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: リソースがテンプレートシステムのサービス内にあることを確認し、ファイルシステムを拡張するための要件に関する製品ドキュメントを参照してください。 |
| 124111 | ERROR | $ERRMSG File system type "${FSType}" is not supported by the kernel currently running on "${TargetSysName}" | **原因**: カーネルのサポートがないため、ファイルシステムのタイプをターゲットシステムにマウントできません。<br><br>**処置**: ターゲットシステムにそのカーネルモジュールがすべてインストールされており、正常に設定されていることを確認してから、リソースを拡張してください。 |
| 124112 | ERROR | must specify machine name containing primary hierarchy | **原因**: creFShier 操作で無効な引数が指定されました。<br><br>**処置**: 通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124113 | ERROR | must specify primary ROOT tag | **原因**: creFShier 操作で無効な引数が指定されました。<br><br>**処置**: 通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124114 | ERROR | must specify primary mount point | **原因**: creFShier 操作で無効な引数が指定されました。<br><br>**処置**: 通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124115 | ERROR | must specify primary switchback type | **原因**: creFShier 操作で無効な引数が指 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 定されました。<br><br>**処置**: 通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124118 | ERROR | dep_remove failure on machine \""$PRIMACH"" for parent "$PRITAG" and child "$DEVTAG." | **原因**: 依存関係の作成後のクリーンアップに失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124119 | ERROR | ins_remove failure on machine ""$PRIMACH"" for \"$PRITAG." | **原因**: インスタンスの作成後のクリーンアップに失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124121 | ERROR | ins_remove failure on machine ""$PRIMACH" | **原因**: リソースの作成後のクリーンアップに失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124122 | ERROR | $ERRMSG Script was terminated for unknown reason | **原因**: このメッセージは、通常の状態では表示されません。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124123 | ERROR | $ERRMSG Required template machine name is null | **原因**: depstoextend 操作で無効な引数が指定されました。<br><br>**処置**: スクリプトが正常に読み出されたことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 124124 | ERROR | $ERRMSG Required template resource tag name is null | **原因**: depstoextend 操作で無効な引数が指定されました。<br><br>**処置**: スクリプトが正常に読み出されたことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124125 | ERROR | $ERRMSG Unable to access template resource "$TemplateTagName" | **原因**: リソースが基本ディスクリソースを特定できませんでした。<br><br>**処置**: 階層とすべての依存関係が正しいことを確認してから拡張してください。 |
| 124126 | ERROR | unextmgr failure on machine ""$PRIMACH" | **原因**: 失敗したリソース拡張操作の後のクリーンアップに失敗しました。<br><br>**処置**: 残りのリソースを手動でクリーンアップし、隣接するログメッセージで詳細を確認してください。 |
| 124128 | ERROR | unextmgr failure on machine ""$PRIMACH"" for "$PRITAG." | **原因**: 失敗したリソース拡張操作の後のクリーンアップに失敗しました。<br><br>**処置**: 残りのリソースを手動でクリーンアップし、隣接するログメッセージで詳細を確認してください。 |
| 124129 | ERROR | $ERRMSG Script was terminated for unknown reason | **原因**: このメッセージは、通常の状態では表示されません。<br><br>**処置**: 追加のログメッセージを探して、詳細を確認してください。 |
| 124130 | ERROR | $ERRMSG Required template machine name is null | **原因**: 拡張操作で無効な引数が指定されました。<br><br>**処置**: スクリプトが正常に読み出された |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124131 | ERROR | $ERRMSG Required template resource tag name is null | **原因**: 拡張操作で無効な引数が指定されました。<br><br>**処置**: スクリプトが正常に読み出されたことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124132 | ERROR | $ERRMSG Required target mount point is null | **原因**: 拡張操作で無効な引数が指定されました。<br><br>**処置**: スクリプトが正常に読み出されたことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124133 | ERROR | $ERRMSG Unable to access template resource "$TemplateTagName" | **原因**: 拡張中のタグがテンプレートシステムに存在しません。<br><br>**処置**: テンプレートシステムで階層が正しいことを確認してから拡張してください。 |
| 124134 | ERROR | $ERRMSG Detected conflict in expected tag name "$TargetTagName" on target machine. | **原因**: 拡張するリソースと同じタグのリソースがすでにターゲットシステムにあります。<br><br>**処置**: 競合しているリソースのどちらか一方を別のタグで再作成してください。 |
| 124135 | ERROR | $ERRMSG Resource "$TemplateTagName" does not have required device resource dependency or unable to access this resource on template machine. | **原因**: リソースまたは基本ディスクリソースがテンプレートシステムで見つかりません。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: テンプレートシステムで階層が正しいことを確認してから拡張してください。 |
| 124136 | ERROR | $ERRMSG Resource "$TemplateTagName" must have one and only one device resource dependency | **原因**: リソースは、テンプレートシステムの階層内に複数の基本デバイスを持っています。<br><br>**処置**: 拡張する前に階層が正しいことを確認し、ファイルシステムリソースが単一のディスクリソースにのみ依存していることを確認してください。 |
| 124137 | ERROR | $ERRMSG Can not access extend for scsi/$DeviceResType resources on machine "$TargetSysName" | **原因**: 所定のストレージタイプのサポートに必要なファイルがターゲットシステムにありません。<br><br>**処置**: 必要なキットがターゲットシステムにインストールされ、ライセンスされていることを確認してください。 |
| 124138 | ERROR | $ERRMSG Unable to access target device resource "$DeviceTagName" on machine "$TargetSysName&quot | **原因**: 必要な基本ディスクリソースがターゲットシステムに存在していません。<br><br>**処置**: 隣接するログメッセージで詳細を確認し、ターゲットシステムが拡張対象リソースをホストするために正常に設定されていることを確認してください。 |
| 124141 | ERROR | $ERRMSG Unable to find mount point "$TemplateLKId" mode on template machine | **原因**: テンプレートシステムのマウントポイント詳細を特定できません。<br><br>**処置**: リソースが in service で、テンプレートシステムでアクセス可能であることを確認してから拡張してください。 |
| 124142 | ERROR | $ERRMSG Unable to create or access mount point "$TargetLKId" on target machine | **原因**: ターゲットシステムでマウントポイントを作成できませんでした。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: マウントポイントの親ディレクトリが存在し、ターゲットシステムでアクセス可能であることを確認してください。 |
| 124143 | ERROR | $ERRMSG Two or more conflicting entries found in /etc/fstab on "$TargetSysName&quot | **原因**: デバイスまたはマウントポイントはターゲットシステムで複数回マウントされているようです。<br><br>**処置**: マウントポイントがターゲットシステムにマウントされていないことを確認してから拡張してください。 |
| 124144 | ERROR | $ERRMSG Failed to create resource instance on "$TargetSysName" | **原因**: ターゲットシステムでのリソースの作成に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。ターゲットサーバーでログを必ず確認してください。 |
| 124145 | ERROR | $ERRMSG Failed to set resource instance state for "$TargetTagName" on \"$TargetSysName" | **原因**: ターゲットシステムでソースの状態を OSU に変更できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124146 | ERROR | must specify machine name containing primary hierarchy | **原因**: filesyshier 操作で無効な引数が指定されました。<br><br>**処置**: スクリプトが正常に読み出されたことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124147 | ERROR | must specify primary mount point | **原因**: filesyshier 操作で無効な引数が指定されました。<br><br>**処置**: スクリプトが正常に読み出された |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124149 | ERROR | create file system hierarchy failure | **原因**: リソースインスタンスを検索するプロセスが失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124150 | ERROR | create file system hierarchy failure | **原因**: システムが /etc/mtab ファイルの読み取りに失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124152 | ERROR | create file system hierarchy failure | **原因**: 基本ディスクリソースを見つけられませんでした。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124153 | ERROR | create file system hierarchy failure | **原因**: ファイルシステムリソースの作成に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124154 | ERROR | create file system hierarchy failure | **原因**: リソースの情報フィールドを更新できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124155 | ERROR | create file system hierarchy failure | **原因**: スイッチバック戦略をリソースで設定できませんでした。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124157 | ERROR | create file system hierarchy failure (conflicting entries in /etc/fstab\) | **原因**: マウントポイントを /etc/fstab ファイルから削除できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124160 | ERROR | Unknown error in script filesysins, err=$err | **原因**: このメッセージは、通常の状態では表示されません。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124161 | ERROR | create filesys instance – existid – failure | **原因**: このメッセージは、通常の状態では表示されません。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124163 | ERROR | create filesys instance – ins_list – failure | **原因**: 既存のリソースの確認に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124164 | ERROR | create filesys instance – newtag – failure | **原因**: システムはリソースの推奨タグの生成に失敗しました。<br><br>**処置**: 通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124168 | ERROR | create filesys instance - ins_create - failure | **原因**: ファイルシステムリソースを作成できませんした。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124169 | ERROR | filesys instance - ins_setstate - failure | **原因**:新しいファイルシステムリソースの状態を初期化できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124173 | ERROR | create filesys instance - dep_create - failure | **原因**: 基本ディスクに対するリソースの依存関係を作成できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124174 | ERROR | machine not specified | **原因**: rmenu_mp 操作で無効な引数が指定されました。<br><br>**処置**: スクリプトが正常に読み出されたことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124175 | ERROR | mount point not specified | **原因**: rmenu_mp 操作で無効な引数が指定されました。<br><br>**処置**: スクリプトが正常に読み出されたことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124177 | ERROR | unexpected multiple matches found | **原因**: 1 つまたは複数のシステムが、複数回使用されたファイルシステムまたはマウントポイントを示しています。<br><br>**処置**: ファイルシステムデバイスとマウントポイントを確認し、ファイルシステムが 1 回だけマウントされていることを |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 確認してください。追加のログメッセージを探して、詳細を確認してください。 |
| 124178 | ERROR | machine name not specified | **原因**: rmenump 操作で無効な引数が指定されました。<br><br>**処置**: スクリプトが正常に読み出されたことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124180 | ERROR | must specify filesystem type | **原因**: validfstype 操作で無効な引数が指定されました。<br><br>**処置**: スクリプトが正常に読み出されたことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124181 | ERROR | mount point not specified | **原因**: validmp 操作で無効な引数が指定されました。<br><br>**処置**: スクリプトが正常に読み出されたことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124182 | ERROR | The mount point $MP is not an absolute path | **原因**: 絶対パスではない（「/」で始まらない）マウントポイントが指定されました。<br><br>**処置**:「/」で始まる絶対パスでマウントポイントを指定してください。 |
| 124183 | ERROR | $MP is already mounted on $MACH | **原因**: 要求されたマウントポイントはシステムですでに使用中です。<br><br>**処置**: 使用中ではないマウントポイント |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | を指定するか、操作を再試行する前にアンマウントしてください。 |
| 124184 | ERROR | The mount point $MP is already protected by LifeKeeper on $MACH | **原因**: システムは、指定したマウントポイントをすでに保護しています。<br><br>**処置**: まだ保護されていない別のマウントポイントを選択してください。 |
| 124185 | ERROR | The mount point $MP is not a directory on $MACH | **原因**: マウントポイントが通常のファイルなどの非ディレクトリを参照しています。<br><br>**処置**: ディレクトリを参照するマウントポイントを選択してください。 |
| 124186 | ERROR | The mount point directory $MP is not empty on $MACH | **原因**: 指定したマウントポイントは空でないディレクトリを参照します。<br><br>**処置**: 空のマウントポイントを選択するか、指定したディレクトリの内容を削除してから操作を再試行してください。 |
| 124187 | ERROR | server name not specified | **原因**: valuepmp 操作で無効な引数が指定されました。<br><br>**処置**: スクリプトが正常に読み出されたことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124188 | ERROR | There are no mount points on server $MACH | **原因**: サーバーにファイルシステムリソースのマウントポイントがありません。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 124194 | WARN | Please correct conflicting "/etc/fstab" entries on server $UNAME for: $FSDEV, $FSNAME | **原因**: ファイルシステムリソースを削除したら、/etc/fstab の一部のエントリを手動でクリーンアップする必要があります。<br><br>**処置**: /etc/fstab ファイルを手動でクリーンアップしてください。 |
| 124195 | ERROR | getchildinfo found no $OKAPP child for $PTAG | **原因**: システムは子リソースを階層で見つけられませんでした。<br><br>**処置**: 隣接するログメッセージで詳細を確認し、階層が正しいことを確認してから、操作を再試行してください。 |
| 124196 | ERROR | enablequotas - quotacheck may have failed for $FS_NAME | **原因**: 割り当て操作に失敗しました。<br><br>**処置**: lifekeeper ログおよび /var/log/messages の隣接するログメッセージで詳細を確認してください。 |
| 124198 | ERROR | enablequotas - quotaon failed to turn on quotas for $FS_NAME, reason | **原因**: 割り当て操作に失敗しました。<br><br>**処置**: lifekeeper ログおよび /var/log/messages の隣接するログメッセージで詳細を確認してください。 |
| 124200 | ERROR | The device node $dev was not found or did not appear in the udev create time limit of $delay seconds | **原因**: デバイスノード（/dev/...）が udev によって作成されませんでした。これは、ストレージまたはストレージへのサーバー接続の問題を示している可能性があります。<br><br>**処置**: lifekeeper ログおよび /var/log/messages の隣接するログメッセージで詳細を確認してください。 |
| 124201 | WARN | Device $device not found. Will retry wait to see if it appears. | **原因**: これは、udev がストレージのデバ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | イスノードエントリを作成している間、通常の条件下で発生する可能性もあります。このメッセージは繰り返し表示されないはずです。<br><br>**処置**: lifekeeper ログおよび /var/log/messages の隣接するログメッセージで詳細を確認してください。 |
| 124202 | ERROR | Command "$commandwithargs" failed. Retrying …. | **原因**: 指定されたコマンドは失敗しましたが、一時的なものである可能性もあります。この障害は、通常の操作で発生しますが、継続はしないはずです。<br><br>**処置**: このメッセージが何度も表示される場合は、隣接するログメッセージで詳細を確認してください。 |
| 124204 | WARN | cannot make file system $FSNAME mount point | **原因**: マウントポイントディレクトリを作成できませんでした。<br><br>**処置**: マウントポイントを作成できることを確認してください。これは、ファイルシステム権限やマウントオプションなどによる可能性があります。 |
| 124207 | ERROR | "fsck"ing file system $FSNAME failed, trying alternative superblock | **原因**: このメッセージは、通常のファイルシステムチェックに失敗したことを示します。このメッセージは、ext2 ファイルシステムまたは代替スーパーブロックロケーションが使用されているファイルシステムの場合は問題ないと思われます。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124209 | ERROR | "fsck"ing file system $FSNAME with alternative superblock failed | **原因**: これは、代替スーパーブロックロケーションで ext2 ファイルシステム (ま |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | たは代替スーパーブロックロケーションが使用されている他のファイルシステム) の確認が失敗したことを示します。<br><br>**処置**: 隣接するログメッセージで詳細を確認し、実行方法の手順を確かめてください。 |
| 124210 | WARN | POSSIBLE FILESYSTEM CORRUPTION ON $FSNAME ($FPNAME) | **原因**: ミラーソースと同期していないファイルシステムが in service になったか、フェイルオーバーされました。<br><br>**処置**: 隣接するログメッセージで詳細を確認し、ファイルシステムを安全に in service にする方法を製品ドキュメントで調べてください。 |
| 124211 | ERROR | Reason for fsck failure ($retval): $ret | **原因**: このログメッセージは一連のメッセージの一部で、fsck プロセスからの実際の終了コードを提供します。<br><br>**処置**: 隣接するログメッセージで詳細を確認し、実行方法の手順を確かめてください。 |
| 124212 | ERROR | "fsck" of file system $FSNAME failed | **原因**: ファイルシステムの確認に失敗しました。これは通常、ファイルシステムが破損したことによるものです。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。製品ドキュメントを参照し、可能性のあるファイルシステムの破損を処理する方法を調べてください。 |
| 124213 | WARN | POSSIBLE FILESYSTEM CORRUPTION ON $FSNAME ($FPNAME) | **原因**: システムまたはユーザが、破損している可能性のあるファイルシステムを in service にしようとしました。これは、ミラーサイトと同期していないとき |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | にファイルシステムが切り替えられたか、フェイルオーバーされた場合に起こることがあります。<br><br>**処置**: 隣接するログメッセージで詳細を確認し、リソースを安全に in service にする方法を製品ドキュメントで調べてください。 |
| 124214 | ERROR | Reason for fsck failure ($retval) | **原因**: このメッセージは、ファイルシステムチェックの障害に関する前のログメッセージに続いて表示されるもので、fsck プロセスのプロセス終了コードを提供します。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124218 | ERROR | File system $FSNAME was found to be already | **原因**: このメッセージは、一連のメッセージの一部です。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124219 | ERROR | mounted after initial mount attempt failed. | **原因**: このメッセージは、一連のメッセージの一部です。通常の状態では表示されないはずですが、リソースが in service の場合は致命的なエラーの可能性もあります。<br><br>**処置**: lifekeeper ログおよび /var/log/messages の隣接するログメッセージで詳細を確認してください。 |
| 124220 | ERROR | File system $FSNAME failed to mount. | **原因**: ファイルシステムをマウントできませんでした。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 124221 | WARN | Protected Filesystem $ID is full | **原因**: ファイルシステムの空き容量がありません。<br><br>**処置**: 使用していないデータをファイルシステムから削除するか、より大きなファイルシステムに移行してください。 |
| 124222 | WARN | Dependent Applications may be affected <> | **原因**: これは、あるリソースに対する操作が、リソースの階層に基づいて別のリソースにも実行される可能性が高いことを示します。<br><br>**処置**: 示されているリソースが影響を受けても許容できることを確認してから操作を続行してください。 |
| 124223 | ERROR | Put "$t" Out-Of-Service Failed By Signal | **原因**: このメッセージは、通常の状態では表示されません。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124227 | ERROR | Put "$i" Out-Of-Service Failed | **原因**: 操作に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124230 | ERROR | Put "$t" In-Service Failed By Signal | **原因**: このメッセージは、通常の状態では表示されません。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124231 | ERROR | Put "$t" In-Service Failed | **原因**: 操作に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 124234 | ERROR | Put "$t" In-Service Failed | **原因**: 操作に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124239 | WARN | Invalid subdirectory cache of $id still exists. This may prevent $id to unmount, causing the resource remove to fail. This is most likely caused by a known issue when mounting a subdirectory of an NFSv3 export. Please see the “LifeKeeper Core – Known Issues / Restrictions” section in the LifeKeeper for Linux Technical Documentation. | **原因**: 無効なサブディレクトリーキャッシュは、NFSv3 エクスポートのサブディレクトリーのマウントが原因である可能性が最も高いです。<br><br>**処置**: NFSv4 を使用し、サブディレクトリーをマウントしないでください。 |
| 125102 | ERROR | `printf ‘Template resource "%s" on server "%s" does not exist’ $TemplateTagName $TemplateSysName` | **原因**: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。 |
| 125103 | ERROR | `printf ‘%s is not shareable with any machine.’ $DEV` | **原因**: デバイスは他のシステムと共有されていないようです。<br><br>**処置**: デバイスがクラスター内のすべてのサーバーからアクセス可能であることを確認してください。すべての関連ストレージドライバおよびソフトウェアが正常にインストールおよび設定されていることを確認してください。 |
| 125104 | ERROR | `printf ‘Failed to create disk hierarchy for "%s" on "%s"’ $PRIMACH $DEV` | **原因**: 物理ディスクを保護するリソースの作成に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認し、レポートされている問題を解決してください。 |
| 125107 | ERROR | `printf ‘Template resource "%s" on server "%s" does not exist’ $TemplateTagName $TemplateSysName` | **原因**: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 125114 | ERROR | `printf 'Template resource "%s" on server "%s" does not exist' $TemplateTagName $TemplateSysName` | 原因: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。 |
| 125120 | ERROR | `printf 'Template resource "%s" on server "%s" does not exist' $TemplateTagName $TemplateSysName` | 原因: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。 |
| 125123 | ERROR | `printf 'Cannot access depstoextend script "%s" on server "%s"' $depstoextend $TargetSysName` | 原因: {server} でスクリプト{DEPSTOEXTEND}を見つけられなかったので、LifeKeeper はリソース階層に対して拡張前のチェックを実行できませんでした。<br><br>処置: LifeKeeper の設定を確認してください。 |
| 125126 | ERROR | `printf 'Template resource "%s" on server "%s" does not exist' $ChildTag $TemplateSysName` | 原因: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。 |
| 125129 | ERROR | `printf 'Template resource "%s" on server "%s" does not exist' $TemplateTagName $TemplateSysName` | 原因: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。 |
| 125155 | ERROR | SCSI $DEV failed to lock. | 原因: SCSI デバイスをロックする際に問題が発生しました。<br><br>処置: 隣接するログメッセージで詳細を確認し、レポートされている問題を解決してください。 |
| 125164 | ERROR | SCSI $INFO failed to unlock. | 原因: SCSI デバイスをロック解除する際に問題が発生しました。<br><br>処置: 隣接するログメッセージで詳細を確認し、レポートされている問題を解決してください。 |
| 125181 | ERROR | `printf 'Template resource "%s" on server "%s" does not exist' $TemplateTag  $TemplateSysName` | 原因: LifeKeeper は、{server} でリソー |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ス {tag} を見つけられませんでした。 |
| 125194 | ERROR | Failed to check disk.(tag=\"$opt_t\" | **原因**: 指定されたタグのデバイスが正しく応答しませんでした。<br><br>**処置**: ディスクが正しく応答していることを確認します。 |
| 125197 | ERROR | usage error: $cmd [-q\|-v] [-r\|-w] [-a\|-d <disk device>] | **原因**: $cmd に誤ったパラメーターが渡されました。<br><br>**処置**: コマンドラインに正しいパラメーターがリストにあることを確認してください。ほとんどの場合、ディスクデバイスが見つからないため、コマンドから除外されます。 |
| 125200 | ERROR | getId command does not exist. id=$id, app=$app, typ=$typ | **原因**: getID コマンドが /opt/LifeKeeper/lkadm/subsys/<app>/<typ>/bin に見つかりませんでした。 <app> and <typ> はエラー メッセージに含まれています。<br><br>**処置**: getId が正しくインストールされ、実行権限があることを確認してください。 |
| 125201 | ERROR | The block device given by the -d option is empty. | **原因**: -d 引数には、ブロックデバイスノードが必要です。<br><br>**処置**: 有効なブロックデバイスノードを使用してコマンドを再試行してください。 |
| 125204 | ERROR | Could not set to RO/RW: $err_disks | **原因**: blockdev(8) コマンドは、読み取り専用または読み取り書き込みの設定に失敗しました。<br><br>**処置**: 追加情報がシステムエラーログに |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 記録される場合があります。 |
| 126105 | ERROR | script not specified - $PTH is a directory | **原因**: 指定したスクリプトパスはディレクトリです。<br><br>**処置**: スクリプトのパスを修正してください。 |
| 126110 | ERROR | script $PTH does not exist | **原因**: 指定したスクリプトパスは存在しません。<br><br>**処置**: スクリプトのパスを修正してください。 |
| 126115 | ERROR | script $PTH is a zero length file | **原因**: 指定したスクリプトは空のファイルです。<br><br>**処置**: スクリプトのファイルパスを修正し、スクリプトの内容を確認してください。 |
| 126117 | ERROR | script $PTH is not executable | **原因**: 指定したスクリプトは実行可能ではありません。<br><br>**処置**: スクリプトのファイルパスを修正し、スクリプトファイルの内容を確認したら、正常な実行権限があることを確認してください。 |
| 126125 | ERROR | required template machine name is null | **原因**: 入力テンプレートマシン名が null です。<br><br>**処置**: 入力テンプレートマシン名を修正してください。 |
| 126130 | ERROR | required template resource tag name is null | **原因**: 入力テンプレートリソース {tag} が |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | null です。<br><br>**処置**: 入力テンプレートリソースタグ名を修正してください。 |
| 126135 | ERROR | Unable to generate a new tag | **原因**:拡張の際に、「newtag」スクリプトを使用してターゲットノードでテンプレートタグ名と同じ新規タグを生成できませんでした。タグ名はすでに存在しています。<br><br>**処置**: 同じノードで重複するタグ名を避け、ログで詳細を確認してください。 |
| 126140 | ERROR | Unable to generate a new tag | **原因**: 拡張の際に、「newtag」スクリプトを使用してターゲットノードで入力ターゲットタグ名として新規タグを生成できませんでした。タグ名はすでに存在しています。<br><br>**処置**: 同じノードで重複するタグ名を避け、ログで詳細を確認してください。 |
| 126150 | ERROR | unable to remote copy template \"$_lscript\" script file | **原因**: テンプレートスクリプトファイルのリモートコピーに失敗しました。原因としては、テンプレートノードにテンプレートスクリプトファイルがない、またはスクリプトファイルを使用できない、あるいは「lcdrcp」プロセスで何らかのトランザクション障害が発生したことなどが考えられます。<br><br>**処置**: テンプレートスクリプトの有無/使用可能性を確認し、テンプレートノードへの接続を確かめてください。また、関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126155 | ERROR | failed to create resource instance on \"$TargetSysName\ | **原因**:「ins_create」によるリソースイン |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | スタンスの作成に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126160 | ERROR | failed to set resource instance state for \"$TargetTagName\" on \"$TargetSysName\ | **原因**: GenApp リソース拡張の際に、「ins_setstate」によるリソースインスタンスの状態の設定に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126170 | ERROR | getlocks failure | **原因**:リソース階層の作成時に管理ロックの取得に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126175 | ERROR | instance create failed | **原因**:「appins」による GenApp インスタンスの作成に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126180 | ERROR | unable to set state to OSU | **原因**: GenApp リソース作成の際に、「ins_setstate」によるリソースインスタンスの状態の設定に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126190 | ERROR | resource restore has failed | **原因**: GenApp リソースのリストアに失敗しました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126200 | ERROR | create application hierarchy rlslocks failure | **原因**: GenApp リソースの作成後にロックの解除に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126210 | ERROR | copy $ltype script $lscript failure | **原因**: リソース作成時に、ユーザ提供のスクリプトを適切な GenApp ディレクトリにコピーできませんでした。<br><br>**処置**: ユーザ提供のスクリプトの有無/使用可能性に加え、GenApp ディレクトリを確認してください。また、関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126215 | ERROR | no $ltype script specified | **原因**: GenApp リソース作成時にユーザ定義スクリプトが見つかりません。<br><br>**処置**: 入力アクションスクリプトを確認し、リソース作成を再実行してください。 |
| 126220 | ERROR | no machine name specified | **原因**:GenApp リソース作成時に指定のマシン名が見つかりません。マシン名の入力内容が見つからないため、指定したユーザスクリプトのコピーに失敗しました。<br><br>**処置**: マシン名の入力内容を確認し、リソース作成を再実行してください。 |
| 126225 | ERROR | no resource tag specified | **原因**:リソース作成時に指定のタグ名が |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 見つかりません。<br><br>**処置**: ソースタグ名の入力内容を確認し、リソース作成を再実行してください。 |
| 126230 | ERROR | $ERRMSG Script was terminated for unknown reason | **原因**: 不明な理由で GenApp リソースの拡張に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126235 | ERROR | $ERRMSG Required template machine name is null | **原因**: GenApp リソース拡張時にテンプレートマシン名の入力が見つかりません。<br><br>**処置**: テンプレートマシン名の入力内容を確認し、リソース拡張を再実行してください。 |
| 126240 | ERROR | $ERRMSG Required template resource tag name is null | **原因**: GenApp リソース拡張時にテンプレートリソースタグ名の入力が見つかりません。<br><br>**処置**: テンプレートリソースタグ名の入力内容を確認し、リソース拡張を再実行してください。 |
| 126245 | ERROR | $ERRMSG Can not access extend for $AppType/$ResType resources on machine \"$TargetSysName\ | **原因**: ターゲットノードでの GenApp リソース拡張の際に、「extend」スクリプトを見つけられませんでした。<br><br>**処置**:「extend」スクリプトの有無/使用可能性を確認し、GenApp リソース拡張を再実行してください。 |
| 126250 | ERROR | create application failure - ins_list failed | **原因**: GenApp リソース作成で「ins_lis |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | t」を呼び出している際に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126255 | ERROR | create application failure - unable to generate a new tag | **原因**: GenApp リソース作成で新規タグの生成に失敗しました。<br><br>**処置**: 同じノードで重複するタグ名の使用を避けてください。また、関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126270 | ERROR | create application failure - ins_create failed | **原因**:「ins_create」による GenApp インスタンスの作成に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126275 | ERROR | create application failure - copy_actions failed | **原因**:「copy_actions」によるユーザ指定テンプレートスクリプトファイルのコピーに失敗しました。<br><br>**処置**: テンプレートスクリプトの有無/使用可能性を確認してください。また、関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126290 | ERROR | Unable to obtain tag for resource with id $ID | **原因**:リカバリー時に入力 ID による GenApp リソースタグ名のフェッチに失敗しました。<br><br>**処置**: 入力 ID が正しいこと、および LCD の GenApp リソースの有無/使用可能性を確認してください。また、関連するエラーをログで確認し、レポートされている問題を解決してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 126300 | ERROR | generic application recover script for $TAG was not found or was not executable | **原因**: リカバリー時に GenApp リソースのユーザ定義スクリプトの特定に失敗しました。<br><br>**処置**: ユーザ定義スクリプトの有無/使用可能性を確認し、GenApp リカバリープロセスを再実行してください。 |
| 126310 | ERROR | -t flag not specified | **原因**: GenApp リソースリストア時にリソースタグ名の入力が見つかりません。<br><br>**処置**: リソースタグ名の入力内容を確認し、リソースリストアを再実行してください。 |
| 126315 | ERROR | -i flag not specified | **原因**: GenApp リソースリストア時にリソース内部 ID の入力が見つかりません。<br><br>**処置**: リソース内部 ID の入力内容を確認し、リソースリストアを再実行してください。 |
| 126327 | ERROR | END timeout restore of \"$TAG\" (forcibly terminating | **原因**: 指定されたタイムアウト値内に restore が完了しませんでした。<br><br>**処置**: タイムアウト値を大きくするか、restore がタイムアウトを超えた理由を解決してください。 |
| 126335 | ERROR | restore script \"$LCDAS/$APP_RESTOREDIR/$TAG\" was not found or is not executable | **原因**: リストア時に GenApp リソースのユーザ定義スクリプトの特定に失敗しました。<br><br>**処置**: ユーザ定義スクリプトの有無/使用可能性を確認し、GenApp リストアプロセスを再実行してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 126340 | ERROR | -t flag not specified | 原因: GenApp リソース削除時にリソースタグ名の入力が見つかりません。<br><br>処置: リソースタグ名の入力内容を確認し、リソース削除を再実行してください。 |
| 126345 | ERROR | -i flag not specified | 原因: GenApp リソース削除時にリソース内部 ID の入力が見つかりません。<br><br>処置: リソース内部 ID の入力内容を確認し、リソース削除を再実行してください。 |
| 126357 | ERROR | END timeout remove of \"$TAG\" (forcibly terminating | 原因: 指定されたタイムアウト値内に remove が完了しませんでした。<br><br>処置: タイムアウト値を大きくするか、remove がタイムアウトを超えた理由を解決してください。 |
| 126365 | ERROR | remove script \"$LCDAS/$APP_REMOVEDIR/$TAG\" was not found or was not executable | 原因: 削除時に GenApp リソースのユーザ定義スクリプトの特定に失敗しました。<br><br>処置: ユーザ定義スクリプトの有無/使用可能性を確認し、GenApp 削除プロセスを再実行してください。 |
| 126375 | ERROR | Script has hung checking \"$tag\". Forcibly terminating. | 原因: 待機時間がユーザ定義のタイムアウトを超えたので、タグ名 {tag} の GenApp リソースの「quickCheck」スクリプトが強制的に終了しました。<br><br>処置: GenApp リソースのパフォーマンスを確認し、quickChecking を再起動してください。また、関連するエラーをログで確認し、レポートされている問題を解決してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 126380 | ERROR | Usage error: no tag specified | 原因: GenApp リソース quickCheck 時にリソースタグ名の入力が見つかりません。<br><br>処置: リソースタグ名の入力内容を確認し、リソース quickCheck を再試行してください。 |
| 126385 | ERROR | Internal error: ins_list failed on $tag. | 原因: quickCheck プロセスで、「ins_list」による入力タグ名別の GenApp リソース情報のフェッチに失敗しました。<br><br>処置: 入力タグ名を修正し、quickCheck プロセスを再実行してください。また、関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126390 | FATAL | Failed to fork process to execute $userscript: $! | 原因: GenApp リソースの「quickCheck」プロセスで、ユーザ定義の「quickCheck」スクリプトを実行するためにプロセスを分岐させられませんでした。<br><br>処置: ユーザ定義の「quickCheck」スクリプトの有無/使用可能性を確認し、「quickCheck」プロセスを再実行してください。 |
| 126391 | ERROR | quickCheck has failed for \"$tag\". Starting recovery. | 原因: タグ名 {tag} の GenApp リソースが失敗することが、ユーザ定義の健全性監視スクリプトにより判定されました。「quickCheck」およびリカバリープロセスが開始されます。<br><br>処置: ローカルリカバリーが完了したら、GenApp リソースのパフォーマンスを確認してください。また、関連するエラーをログで確認し、レポートされている問題を解決してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 126392 | WARN | ${convtag}_TIMEOUT: This parameter is old. This parameter will not be supported soon. | 原因: <tag>_TIMEOUT を使用する gen/app リソースのタイムアウトが使用されています。<br><br>処置: gen/app のクイックチェックスクリプトのタイムアウトを指定するには、<tag>_QUICKCHECK_TIMEOUT の形式で指定してください。 |
| 126400 | ERROR | -t flag not specified | 原因:GenApp リソース削除プロセスでリソースタグ名の入力が見つかりません。<br><br>処置: リソースタグ名の入力内容を確認し、リソース削除プロセスを再実行してください。 |
| 126405 | ERROR | -i flag not specified | 原因: GenApp リソース削除プロセスでリソース内部 ID の入力が見つかりません。<br><br>処置: リソース内部 ID の入力内容を確認し、リソース削除プロセスを再実行してください。 |
| 126478 | ERROR | Failed to create tag \'$new_leaf\'. | 原因:「creapphier」ユーティリティは、指定されたタグ {new_leaf} を作成できませんでした。<br><br>処置: /var/log/lifekeeper.log で「creapphier」からの追加メッセージを確認してください。 |
| 126479 | ERROR | Failed to extend tag \'$new_leaf\'. | 処置: /var/log/lifekeeper.log で拡張マネージャーからの追加エラーを確認してください。 |
| 126481 | ERROR | Failed to create dependency on \'$sys\' for \'$new_leaf\' to \'$hier{$leaf}{$sys}{‘Ta | 処置: /var/log/lifekeeper.log で「dep_cre |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | g'}\'. | ate」関数のエラーを確認してください。 |
| 126484 | ERROR | Tag \'$root_tag\' is not in-service. | **原因**: 指定されたタグ{root_tag} は、クラスター内のどのノードでも In Service になっていません。<br><br>**処置**: 指定されたタグ{root_tag} をクラスター内の任意のノードで In Service にし、「create_terminal_leaf」を再実行してください。 |
| 126485 | ERROR | Tag \'$root_tag_1\' was not found, select the root tag for a hierarchy to add a terminal leaf resource. | **原因**:「create_terminal_leaf」に渡された最初のタグ {root_tag_1} が、ユーティリティが実行されたシステムで見つかりませんでした。<br><br>**処置**: リソースがクラスター内のノードで In-service であり、すべてのノードへリソースが完全に拡張されていることを確認してください。 |
| 126486 | ERROR | Unable to create leaf tag from \'$tag\'. | **原因**: 一意の端末リーフタグを作成できませんでした。100回試行した後、一意の端末リーフタグを作成できませんでした。<br><br>**処置**: 複数のリーフタグと、問題を暗示している可能性のある /var/log/lifekeeper.log のエラーを確認してください。 |
| 126487 | ERROR | Tag \'$root_tag_2\' was not found, select the root tag for a hierarchy to add a terminal leaf resource. | **原因**:「create_terminal_leaf」に渡された2番目のタグ {root_tag_2} が、ユーティリティが実行されたシステムで見つかりませんでした。<br><br>**処置**: リソースがクラスター内のノードで In-service であり、すべてのノードへリソースが完全に拡張されていることを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 126488 | ERROR | Tag \'$root_tag_1' is not extended to 3 or more systems. | **原因**: 指定されたタグ {root_tag_1} は、3つ以上のノードに拡張されていません。<br><br>**処置**: 指定したタグ {root_tag_1} を最低でも3つのノードに拡張し、「create_terminal_leaf」を再度実行してください。 |
| 126489 | ERROR | $cmd does not support SDRS resources. | **原因**: マルチサイト構成が検知されました。<br><br>**処置**: ありません |
| 126492 | ERROR | Remove resource $tag failed. | **原因**: 障害発生後の create_terminal_leaf ユーティリティによるクリーンアップで、リソースが正しく削除されませんでした。<br><br>**処置**: リストされているタグがまだ存在する場合は削除してください。 |
| 126494 | ERROR | Delete dependency failed on \'$sys\' for \'$tag\' to \'$parent\'. | **原因**: 障害発生後の create_terminal_leaf ユーティリティのクリーンアップで、依存関係が適切に削除されませんでした。<br><br>**処置**: 依存関係がまだ存在する場合は削除してください。 |
| 126495 | ERROR | New tag \'$new_tag\' was modified during create, expected \'$new_leaf\'. | **原因**: タグ名に矛盾がありました。<br><br>**処置**: 競合のないタグ名を選択してください。 |
| 126496 | ERROR | Tag \'$root_tag_1\' is not in-service. | **原因**: リストされているルートタグは、ターミナルリーフを作成するために in-service である必要があります。<br><br>**処置**: ルートタグを in-service にし |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | て、操作を再試行してください。 |
| 126497 | ERROR | Tag \'$root_tag_2\' is not in-service. | **原因**: リストされているルートタグは、ターミナルリーフを作成するために in-service である必要があります。<br><br>**処置**: ルートタグを in-service にして、操作を再試行してください。 |
| 126498 | ERROR | Tag \'$root_tag_1\' and \'$root_tag_2' are not extended to same systems. | **原因**: ターミナルリーフリソースを作成するために指定された両方の階層が同じシステム上に存在する必要があります。<br><br>**処置**: 同じシステム上で階層を作成および拡張してください。 |
| 128005 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | **原因**: オペレーティングシステムの信号 {signal} によって、{server} 上の {resource} の quickCheck が失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128008 | ERROR | Usage: quickCheck -t <tag name> -i <id> | **原因**: dmmp デバイスの quickCheck コマンドに不正な引数が指定されたため、コマンドが実行できませんでした。<br><br>**処置**: すべてのソフトウェアコンポーネントの正しいバージョンが適切にインストールされていることを確認してください。次の正しい引数を指定してコマンドを再実行してください。-t <Resource Tag> および -i <Resource ID> (quickCheck 対象の dmmp デバイスリソースを特定する ID) |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 128010 | ERROR | quickCheck for "%s" failed checks of underlying paths, initiate recovery. retry count=%s. | 原因: dmmp キットが {count} 回の試行後、デバイスの quickCheck に失敗しました。保護されている dmmp リソースのリカバリーが実行されます。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |
| 128021 | ERROR | unable to find device for uuid "%s". | 原因: restore 操作時に一意の ID を使用してデバイスを見つけられませんでした。<br><br>処置: リソースが正しく設定されていることを確認してください。リストア対象の dmmp デバイスリソースを特定する正しいデバイス ID でコマンドを再実行してください。 |
| 128025 | ERROR | Device "%s" failed to unlock. | 原因: restore 操作時に正常に動作していない {device} が検出され、ロック解除できませんでした。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128026 | ERROR | Device "%s" failed to lock. | 原因: リストア時に {device} をロックできませんでした。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128031 | ERROR | unable to find device for uuid "%s". | 原因: remove 操作時に一意の ID を使用 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | してデバイスを見つけられませんでした。<br><br>**処置**: リソースが正しく設定されていることを確認してください。削除対象の dmmp デバイスリソースを特定する正しいデバイス ID でコマンドを再実行してください。 |
| 128034 | ERROR | Device "%s" failed to unlock. | **原因**: 削除時に {device} をロック解除できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128036 | ERROR | unable to load existing information for device with uuid "%s". | **原因**: 一意の ID を使用してデバイス情報を読み込めませんでした。<br><br>**処置**: リソースが正しく設定されていることを確認してください。dmmp デバイスリソースを特定する正しいデバイス ID でコマンドを再実行してください。 |
| 128037 | ERROR | unable to load existing information for device "%s". | **原因**: 名前を使用してデバイス情報を読み込めませんでした。<br><br>**処置**: リソースが正しく設定されていることを確認してください。dmmp デバイスリソースを特定する正しいデバイス名でコマンドを再実行してください。 |
| 128038 | ERROR | unable to load existing information for device, no dev or uuid defined. | **原因**: 一意のデバイス ID もデバイス名も指定されていないため、デバイス情報を読み込めませんでした。<br><br>**処置**: リソースが正しく設定されている |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ことを確認してください。dmmp デバイスリソースを特定する正しいデバイス ID またはデバイス名でコマンドを再実行してください。 |
| 128041 | ERROR | unable to load existing information for device with uuid "%s". | **原因**: 一意の ID を使用してデバイス情報を読み込めませんでした。<br><br>**処置**: リソースが正しく設定されていることを確認してください。dmmp デバイスリソースを特定する正しいデバイス ID でコマンドを再実行してください。 |
| 128057 | ERROR | All paths are failed on "%s". | **原因**: LifeKeeper は、保護対象の dmmp デバイスへのすべてのパスが障害状態にあることを検出しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |
| 128058 | ERROR | could not determine registrations for "%s"! All paths failed. | **原因**: LifeKeeper は、保護対象の dmmp {device} の登録を特定できませんでした。dmmp {device} へのすべてのパスが障害状態にあります。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |
| 128059 | WARN | path "%s" no longer configured for "%s", remove from path list. | **原因**:LifeKeeper が検出した保護対象の {device} への {path} は、現在は有効ではなく、パスリストから削除されます。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 128060 | WARN | registration failed on path "%s" for "%s". | 原因: LifeKeeper は、保護対象の dmmp {device} の {path} の登録を特定できませんでした。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |
| 128062 | ERROR | all paths failed for "%s". | 原因: LifeKeeper は、保護対象の dmmp {device} の有効なパスを確認できませんでした。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |
| 128072 | ERROR | The daemon "%s" does not appear to be running and could not be restarted. Path failures may not be correctly handled without this daemon. | 原因: LifeKeeper は dmmp デーモンが実行されていることを確認できなかったため、デーモンを再起動できませんでした。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |
| 128078 | ERROR | "%s" resource type is not installed on "%s". | 原因: dmmp デバイスをサポートするための Device Mapper Multipath Recovery Kit がシステムにインストールされていません。<br><br>処置: steeleye-lkDMMP Device Mapper Multipath Recovery Kit rpm をシステムにインストールしてください。 |
| 128083 | ERROR | This script must be executed on "%s". | 原因: dmmp デバイスリソースを作成するための devicehier スクリプトの引数に不正なシステム名が指定されました。<br><br>処置: クラスターノードとコミュニケー |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ションパスが適切に設定されていることを確認してください。devicehier スクリプトに正しいシステム名を指定してください。この名前は、コマンドを実行するシステムの名前に一致する必要があります。 |
| 128084 | ERROR | The device %s is not active. | **原因**: リソース作成時に LifeKeeper は、指定された {device} をシステム上の有効なデバイスとして見つけられませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。次の正しい引数を指定してコマンドを再実行してください。-t <Resource Tag> および -i <Resource ID> (作成する dmmp デバイスリソースを特定する ID) |
| 128086 | ERROR | Failed to create "%s" hierarchy. | **原因**: LifeKeeper は、{device} のためのリソース階層を作成できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128088 | ERROR | Error creating resource "%s" on server "%s" | **原因**: LifeKeeper は、{server} で {tagname} を使用してリソースを作成できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128090 | ERROR | Failed to create dependency "%s"-"%s" on system "%s". | **原因**: LifeKeeper は、{system} で依存関 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 係 {resource tag name} – {resource tag name} を作成できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128091 | ERROR | Error creating resource "%s" on server "%s" | **原因**: LifeKeeper は、{system} で {resource} を作成できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128101 | ERROR | "%s" constructor requires a valid argument. | **原因**: LifeKeeper は、dmmp リソースのオブジェクトを作成できませんでした。<br><br>**処置**: 次の正しい引数を指定してコマンドを再実行してください。-t <Resource Tag> および -i <Resource ID> (dmmp デバイスリソースを特定する ID) |
| 128102 | ERROR | Invalid tag "%s". | **原因**: 指定されたタグ名のリソースインスタンスが見つかりませんでした。<br><br>**処置**: リソースが正しく設定されていることを確認してください。次の正しい引数を指定してコマンドを再実行してください。-t <Resource Tag> および -i <Resource ID> (dmmp デバイスリソースを特定する ID) |
| 128111 | ERROR | Failed to get registrations for "%s": %s. Verify the storage supports persistent reservations. | **原因**: LifeKeeper は {device} の登録の取得に失敗し、メッセージ「bad field in Persistent reservation in cdb」を出力しまし |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | た。<br><br>**処置**: このストレージでパーシステントリザベーションがサポートされているか確認してください。隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128112 | ERROR | Failed to get registrations for "%s": %s. Verify the storage supports persistent reservations. | **原因**: LifeKeeper は {device} の登録の取得に失敗し、メッセージ「illegal request」を出力しました。<br><br>**処置**: このストレージでパーシステントリザベーションがサポートされているか確認してください。隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128136 | ERROR | A previous quickCheck with PID "%s" running for device "%s" has been terminated. | **原因**: LifeKeeper は、dmmp リソースの restore 操作時に前の quickCheck がまだ実行中であることを検出しました。LifeKeeper は quickCheck を終了しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |
| 128137 | ERROR | SCSI reservation conflict on %s during LifeKeeper resource initialization. Manual intervention required. | **原因**: LifeKeeper は、dmmp リソースの restore 操作時に {device} で SCSI リザベーションコンフリクトを検出しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。手動による介入および {device} で |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | のリザベーションコンフリクトの解消が必要です。 |
| 128138 | ERROR | unable to clear registrations on %s. | 原因: LifeKeeper は、{device} 上のすべての登録を消去できませんでした。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128140 | WARN | registration failed on path %s for %s. | 原因: LifeKeeper は、{device} の {path} の登録を作成できませんでした。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128143 | ERROR | reserve failed (%d) on %s. | 原因: LifeKeeper は、{device} 上の {resource} の登録を作成できませんでした。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128145 | ERROR | The server ID "%s" returned by "%s" is not valid. | 原因: LifeKeeperは有効なホスト {ID} を生成できませんでした。<br><br>処置: デバイスを登録するための ID は、クラスター内のサーバーを一意に特定する 1 ～ 12 桁の 16 進数で構成されます。隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正し |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | てから操作を再試行する必要があります。 |
| 128146 | ERROR | device failure on %s. SYSTEM HALTED. | 原因:LifeKeeper は {device} で障害を検出しました。サーバーを再起動します。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |
| 128148 | ERROR | device failure on %s. SYSTEM HALTED DISABLED. | 原因: LifeKeeper は {device} で障害を検出しました。LifeKeeper の設定によって、再起動はスキップされました。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。「SCSIHALT」設定を有効にして、いかなるデバイス障害が検出されたときも再起動が行われるようにしてください。 |
| 128149 | ERROR | device failure or SCSI Error on %s. SEND EVENT DISABLED. | 原因: LifeKeeper は {device} で障害を検出しました。LifeKeeper の設定によって、イベント生成はスキップされました。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。「SCSIEVENT」設定を有効にして、いかなるデバイス障害が検出されたときも sendevent を使用できるようにしてください。 |
| 128150 | ERROR | %s does not have EXCLUSIVE access to %s, halt system. | 原因: LifeKeeper は {server} の {device} でリザベーションコンフリクトを検出しました。サーバーを再起動します。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 128151 | ERROR | %s does not have EXCLUSIVE access to %s, halt system DISABLED. | **原因**: LifeKeeper は {server} の {device} でリザベーションコンフリクトを検出しました。LifeKeeper の設定によって、再起動はスキップされました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。「RESERVATIONCONFLICT」設定を有効にして、いかなるリザベーションコンフリクトが検出されたときも再起動が行われるようにしてください。 |
| 128154 | WARN | unable to flush buffers on %s. | **原因**: LifeKeeper は、dmmp リソースの削除時に {device} のバッファを消去できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128157 | WARN | %s utility not found, limited healthcheck for %s. | **原因**: LifeKeeper は、{device} の健全性チェックを行う「dd」ユーティリティを見つけられませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128160 | ERROR | %s failed to read %s. | **原因**: LifeKeeper は、{utility} の使用中に {device} のディスク I/O テストに失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要がありま |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | す。 |
| 128163 | ERROR | Registration ID "%s" for "%s" is not valid. | 原因: LifeKeeper は {device} の有効な登録 {ID} の生成に失敗しました。<br><br>処置: デバイスを登録するための ID は、デバイスへのパスに由来する 4 桁の 16 進数で構成されます。隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128170 | ERROR | Usage: canextend <Template system name> <Template tag name> | |
| 128500 | ERROR | Usage error | 原因: dmmp デバイスの restore コマンドに不正な引数が指定されたため、コマンドが実行できませんでした。<br><br>処置: 次の正しい引数を指定してコマンドを再実行してください。-t <Resource Tag> および -i <Resource ID> (リストアする dmmp デバイスリソースを特定する ID) |
| 128504 | ERROR | "%s" resource type is not installed on "%s". | 原因: dmmp デバイスをサポートするための Device Mapper Multipath Recovery Kit がシステムにインストールされていません。<br><br>処置: steeleye-lkDMMP Device Mapper Multipath Recovery Kit rpm をシステムにインストールしてください。 |
| 128506 | ERROR | Usage error | 原因: dmmp デバイスの devShared コマンドに不正な引数が指定されたため、コマンドが実行できませんでした。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 次の正しい引数を指定してコマンドを再実行してください。<Template Resource System Name> および <Template Resource Tag> (作成する dmmp デバイスリソースを特定するタグ) |
| 128507 | FATAL | This script must be executed on "%s". | **原因**: dmmp デバイスリソースを作成するための devicehier スクリプトの引数に不正なシステム名が指定されました。<br><br>**処置**: devicehier スクリプトに正しいシステム名を指定してください。この名前は、コマンドを実行するシステムの名前に一致する必要があります。 |
| 128511 | ERROR | Failed to get the ID for the device "%s". Hierarchy create failed. | **原因**: dmmp デバイスリソースを作成するための devicehier スクリプトは、指定されたデバイスの SCSI ID を特定できませんでした。<br><br>**処置**: 指定されたデバイスパスが存在すること、およびそのパスがサポート対象の SCSI ストレージアレイのパスであることを確認してください。 |
| 128512 | ERROR | Failed to get the disk ID for the device "%s". Hierarchy create failed. | **原因**: dmmp ディスクリソースを作成するための devicehier スクリプトは、指定されたディスクの SCSI ID を特定できませんでした。<br><br>**処置**: 指定されたデバイスパスが存在すること、およびそのパスがサポート対象の SCSI ストレージアレイのパスであることを確認してください。 |
| 128513 | ERROR | Failed to create the underlying resource for device "%s". Hierarchy create failed. | **原因**: ベースとなる dmmp ディスクリソースの作成に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細お |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | よび関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128515 | ERROR | Error creating resource "%s" on server "%s" | **原因**: dmmp デバイスリソースの作成に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128517 | ERROR | Failed to create dependency "%s"-"%s" on system "%s". | **原因**: dmmp デバイスリソースと dmmp ディスクリソースとの間の親 / 子の依存関係の作成に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128519 | ERROR | Error creating resource "%s" on server "%s" | **原因**: 新しく作成した dmmp デバイスリソースを in service にする操作が失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128521 | ERROR | Either TEMPLATESYS or TEMPLATETAG argument missing | **原因**: dmmp デバイスリソースの extend コマンドに不正な引数が指定されました。<br><br>**処置**:テンプレートシステムおよびタグ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | の正しい名前を指定して dmmp デバイスリソースの extend を再実行してください。 |
| 128540 | ERROR | Usage error | **原因**: dmmp デバイスの SCSI ID を取得する getId コマンドに不正な引数が指定されました。<br><br>**処置**: 次の正しい引数を指定してコマンドを再実行してください。-i <device path> または -b <device ID>。 |
| 128541 | ERROR | Usage error | **原因**: dmmp デバイスリソースを削除するコマンドに不正な引数が指定されました。<br><br>**処置**: 次の正しい引数を指定してコマンドを再実行してください。-t <dmmp device resource tag>。 |
| 128543 | ERROR | device node "$dev" does not exist. | **原因**: dmmp デバイスリソースをリストアするために必要なデバイスノードが存在しません。リストア時に udev デバイスの作成に割り当てられている待機時間を越えました。<br><br>**処置**: udev がデバイスを作成した後に dmmp デバイスリソースの restore を再実行してください。 |
| 128544 | ERROR | Usage error | **原因**: dmmp デバイスリソースを out of service にする remove コマンドに不正な引数が指定されました。<br><br>**処置**: 次の正しい引数を指定してコマンドを再実行してください。-t <dmmp device resource tag>。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 128550 | ERROR | Failed to get path information.(tag=\"$opt_t\" | 原因: OSUquickCheck が、DMMP デバイスと一致しないタグで呼び出されました。<br><br>処置:正しいタグが使用されていることを確認してください。 |
| 128551 | ERROR | Failed to get the status of path.(tag=\"$opt_t\" | 原因:OSUquickCheck は、渡されたタグの DMMP パスの 'mode' を取得できませんでした。<br><br>処置: DMMP デバイスが構成され、応答していることを確認してください。 |
| 128552 | ERROR | Failed to check the status of path.(tag=\"$opt_t\" | 原因: OSUquickCheck は、DMMP デバイスにアクティブなパスを見つけられませんでした。<br><br>処置: DMMP デバイスが正しく構成されていることを確認してください。 |
| 128553 | ERROR | Failed to get the resource information.(tag=\"$opt_t\" | 原因: OSUquickCheck は、指定されたタグの情報を読み込めませんでした。<br><br>処置: タグが DMMP デバイス用であることを確認してください。 |
| 129100 | FATAL | Failed to load instance from LifeKeeper. | 原因: 無効なリソースタグまたは ID が指定されました。<br><br>処置: タグまたは ID が有効であることを確認し、コマンドを再実行してください。 |
| 129103 | FATAL | No resource matches tag "$self->{'tag'}". | 原因: 無効なリソースタグが指定されました。<br><br>処置: タグが有効であることを確認 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | し、コマンドを再実行してください。 |
| 129104 | FATAL | An error occurred setting LifeKeeper resource information | 原因: LifeKeeper で内部エラーが発生しました。 |
| 129110 | ERROR | Could not get the Elastic Network Interface ID for $dev | 原因: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。<br><br>処置: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129111 | ERROR | Failed to get Allocation ID of Elastic IP "$elasticIp". | 原因: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。<br><br>処置: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129113 | ERROR | Failed to get my instance ID. | 原因: EC2 インスタンスのメタデータのアクセスに失敗しました。<br><br>処置: Amazon コンソールを確認し、操作を再試行してください。 |
| 129114 | ERROR | Failed to get ENI ID. | 原因: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。<br><br>処置: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129116 | ERROR | Failed to associate Elastic IP "$self->{'EIP'}" on "$self->{'DEV'}". | 原因: EC2 API の呼び出しに失敗しまし |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | た。ネットワークの問題と考えられます。<br><br>**処置**: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129118 | WARN | $self->{'EIP'} is not associated with any instance. | **原因**: Elastic IP がインスタンスに関連付けられていません。<br><br>**処置**: LifeKeeper は Elastic IP を関連付ける EC2 API を呼び出して、問題の修正を試行してください。隣接するログメッセージで詳細を確認してください。 |
| 129119 | WARN | $self->{'EIP'} is associated with another instance. | **原因**: Elastic IP が別のインスタンスに関連付けられています。<br><br>**処置**: LifeKeeper は Elastic IP を関連付ける EC2 API を呼び出して、問題の修正を試行してください。隣接するログメッセージで詳細を確認してください。 |
| 129120 | ERROR | Failed to recover Elastic IP. | **原因**: EC2 API が Elastic IP の関連付けに失敗しました。<br><br>**処置**: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129121 | ERROR | Recovery process ended but Elastic IP is not associated with this instance. Please check AWS console. | **原因**: EC2 API が Elastic IP の関連付けに失敗しました。<br><br>**処置**: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129122 | ERROR | Error creating resource \"$target_tag\" with return code of \"$err\". | **原因**: LifeKeeper がサーバー上にリソー |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | スインスタンスを作成できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 129123 | ERROR | Failed to get ENI ID. | **原因**: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。<br><br>**処置**: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129124 | WARN | $self->{'EIP'} is associated with another network interface. | **原因**: Elastic IP は正しいインスタンスに関連付けられていますが、ENI が正しくありません。<br><br>**処置**: LifeKeeper は Elastic IP を関連付ける EC2 API を呼び出して、問題の修正を試行してください。隣接するログメッセージで詳細を確認してください。 |
| 129125 | ERROR | Link check failed for interface '$dev'. | **原因**: 要求したインターフェースが「NO-CARRIER」となっています。これは、リンクが存在しないことを示します。<br><br>**処置**: ネットワークインターフェースを確認し、リンクをアップ状態にしてください。 |
| 129126 | ERROR | Link check failed for interface '$dev'. Reason: down slave. | **原因**: 要求したインターフェースが「NO-CARRIER」となっています。これは、リンクが存在しないことを示します。<br><br>**処置**: ネットワークインターフェースを |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 確認し、リンクをアップ状態にしてください。 |
| 129127 | ERROR | Link check failed for interface \'$dev\'. | 原因: "nmcli connection show --active" の呼び出しで、$dev がアクティブであることが返されませんでした。 |
| 129129 | WARN | The link for network interface '$self->{'DEV'}' is down. Attempting to bring the link up. | 原因: 要求したインターフェースが「NO-CARRIER」となっています。これは、リンクが存在しないことを示します。<br><br>処置: LifeKeeper は、リンクをアップ状態にして Elastic IP とインターフェースを関連付けることにより、問題の修正を試行してください。隣接するログメッセージで詳細を確認してください。 |
| 129130 | ERROR | Failed to modify "$opt_t" to end pint URL "$endpoint". | |
| 129137 | ERROR | The link for network interface '$self->{'DEV'}' is still down. | 原因: LifeKeeper がリンクをアップ状態にできませんでした。<br><br>処置: インターフェースが有効であり、アップ状態であることを確認してください。隣接するログメッセージで詳細を確認してください。 |
| 129139 | WARN | The link for network interface '$self->{'DEV'}' is down. | 原因: 要求したインターフェースが「NO-CARRIER」となっています。これは、リンクが存在しないことを示します。<br><br>処置: ネットワークインターフェースを確認し、リンクをアップ状態にしてください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 129140 | ERROR | Could not get ENI ID for $self->{IP}. | **原因**: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。<br><br>**処置**: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129142 | ERROR | Failed to update route table | **原因**: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。<br><br>**処置**: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129143 | ERROR | You must have exactly one IP address resource as the parent of the RouteTable EC2 resource. Please reconfigure your resource hierarchy. | **原因**: ルートテーブルの EC2 リソースは親として IP リソースを 1 つのみ持つ必要があります。<br><br>**処置**: 必要に応じてリソース階層を修復してください。 |
| 129144 | ERROR | $func called with invalid timeout: $timeout | **原因**: /etc/default/LifeKeeper ファイルに無効なタイムアウト値が指定されました。<br><br>**処置**: /etc/default/LifeKeeper 内にあるすべての EC2_*_TIMEOUT 設定が正しいことを確認してください。 |
| 129145 | ERROR | $func action timed out after $timeout seconds | **原因**: タイムアウト期間中に動作が完了しませんでした。<br><br>**処置**: その動作の EC2_*_TIMEOUT 値 (/etc/default/LifeKeeper 内) を増加することを検討してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 129146 | ERROR | failed to run $func with timeout: $@ | 原因: これは内部エラーです。 |
| 129148 | ERROR | Amazon describe-route-tables call failed (err=%s)(output=%s) | 原因: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。<br><br>処置: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129150 | ERROR | Elastic IP \"$elasticIp\" is associated with another instance. | 原因: Elastic IP が正しいインスタンスに関連付けられていません。<br><br>処置: LifeKeeper は Elastic IP を関連付ける EC2 API を呼び出して、問題の修正を試行してください。隣接するログメッセージで詳細を確認してください。 |
| 129151 | ERROR | Could not get the Association ID for Elastic IP "$elasticIp". | 原因: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。<br><br>処置: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129152 | ERROR | Failed to disassociate Elastic IP "$self->{'EIP'}" on "$self->{'DEV'}". | 原因: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。<br><br>処置: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129153 | ERROR | Failed to disassociate Elastic IP "$elasticIp", (err=%s)(output=%s) | 原因: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129154 | ERROR | Amazon describe-addresses call failed (err=%s)(output=%s) | **原因**: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。<br><br>**処置**: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129155 | ERROR | Amazon describe-address call failed (err=%s)(output=%s) | **原因**: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。<br><br>**処置**: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129157 | ERROR | curl call failed (err=%s)(output=%s) | **原因**: EC2 インスタンスのメタデータのアクセスに失敗しました。<br><br>**処置**: Amazon コンソールを確認し、操作を再試行してください。 |
| 129159 | ERROR | Amazon associate-address call failed (err=%s)(output=%s) | **原因**: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。<br><br>**処置**: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129160 | ERROR | Amazon describe-addresses call failed (err=%s)(output=%s) | **原因**: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129161 | ERROR | Error deleting resource "$otherTag" on "$otherSys" with return code of "$err". | **原因**: LifeKeeper がサーバー上のリソースインスタンスを削除できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 129162 | ERROR | Could not getRouteTablesByIP | |
| 129163 | ERROR | Could not getRouteTablesByIP | |
| 129164 | ERROR | [$SUBJECT event] mail returned $err | |
| 129165 | ERROR | COMMAND OUTPUT: cat /tmp/err$$ | |
| 129167 | ERROR | snmptrap returned $err for Trap 180 | |
| 129168 | ERROR | COMMAND OUTPUT: cat /tmp/err$$ | |
| 129170 | ERROR | This resource is in the old format. Please update. | |
| 129171 | ERROR | Error disabling sourceDest checks with return code of \"$ret\". | |
| 129172 | ERROR | Failed to lookup IP from resource $self->{'tag'}. | |
| 129173 | ERROR | Failed to lookup device GUID for IP resource $iptag. | |
| 129174 | ERROR | Could not find eni_id for device GUID $dev. | |
| 129175 | ERROR | Failed to disable sourceDestChecks. (err=%s)(output=%s | |
| 129176 | ERROR | Failed to disable sourceDestChecks for $eni_id. (err=%s)(output=%s | |
| 129177 | ERROR | Error disabling sourceDest checks with return code of \"$ret\". | |
| 129178 | ERROR | Failed to disable sourceDestChecks for $eni_id. (err=%s)(output=%s | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 129180 | ERROR | Elastic Network Interface $eni sourceDestChecks are enabled. | |
| 129181 | WARN | WARNING Failed to find IP resources since EC2 resource is extended first. Skipping sourceDestChecks. | |
| 129182 | WARN | WARNING SourceDestCheck was skipped because Amazon describe-network-interface-attribute call failed. Please check if ec2:DescribeNetworkInterfaceAttribute and ec2:ModifyNetworkInterfaceAttribute are granted. | |
| 129403 | ERROR | END failed create of $TAG due to a $sig signal | **原因**: 作成プロセスがシグナルによって中断されました。 |
| 129409 | ERROR | The IP resource $IP_RES is not \"ISP\". | **原因**: IP リソースが in service ではありません。<br><br>**処置**: リソースを in service にして、操作をもう一度行ってください。 |
| 129410 | ERROR | Could not find IP resource $IP_RES | **原因**: IP リソースが存在していることを確認し、操作を再試行してください。 |
| 129412 | ERROR | EC2 resource $ID is already protected | **原因**: 指定した ID を持つリソースはすでに存在しています。<br><br>**処置**: 残っている古いリソースをすべてクリーンアップしてから、新しいリソースを再作成してください。 |
| 129416 | ERROR | Error creating resource "$TAG" with return code of "$lcderror". | **原因**: LifeKeeper がリソースインスタンスを作成できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 129418 | ERROR | Dependency creation between "$IP_RES" and "$TAG" failed with return code of "$lcderror". | 原因: LifeKeeper がリソースの依存関係を作成できませんでした。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 129420 | ERROR | In-service failed for tag "$TAG". | 原因: LifeKeeper がリソースインスタンスを in service にできませんでした。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 129423 | ERROR | Could not get ENI ID for $dev. | |
| 129425 | ERROR | Failed to update route table | |
| 129426 | ERROR | In-service (dummy) failed for tag "$TAG". | |
| 129800 | ERROR | canextend checks failed for \"$self->{‘tag’}\" (err=$ret) | 原因: ターゲットサーバーで拡張前のチェックに失敗しました。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 129801 | ERROR | canextend checks failed for "$self->{'tag'}". EC2_HOME "$self->{EC2_HOME}" does not exist on $me. | |
| 133106 | ERROR | You must have exactly one IP address resource as the child of the route53 resource. Please reconfigure your resource hierarchy. | |
| 133111 | ERROR | Failed to init the object. | |
| 133114 | ERROR | Failed start Route53 resource $self->{Tag} | |
| 133115 | ERROR | Failed start Route53 resource $self->{Tag} | |
| 133116 | WARN | Could not Get A Record Value Address : $aAWSresult | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 133117 | WARN | aws call failed : $aAWSresult | |
| 133118 | WARN | aws call failed : $upsertResp | |
| 133119 | ERROR | Could not Update and Create A Record Set : $upsertResp | |
| 133120 | WARN | Could not get Route53 API batch request ID form UPSERT response XML data : $upsertResp | |
| 133121 | WARN | Could not Get A Record Value Address : $aAWSresult | |
| 133122 | WARN | aws call failed : $statusResult | |
| 133123 | ERROR | Faild check change batch request status : $statusResult | |
| 133126 | WARN | Failed to Route53 API response : $rc | |
| 133129 | ERROR | Route53 resource $self->{Tag} is stopped | |
| 133130 | ERROR | $self->{Tag} is in the old format. Please update. | |
| 133601 | ERROR | $usage | |
| 133602 | ERROR | Template resource "$template_tag" on server \"$template_sys" does not exist | |
| 133608 | ERROR | Template resource "$template_tag" on server "$template_sys" does not exist | |
| 133614 | ERROR | Error creating resource "$Tag" on server "$me" | |
| 133617 | ERROR | END failed hierarchy extend of resource  $Tag with return value of $ecode. | |
| 133619 | ERROR | $usage | |
| 133620 | ERROR | END failed extend of "$Tag" due to a "$sig" signal | |
| 133700 | ERROR | $usage | |
| 133701 | ERROR | END failed create of "$Route53Tag" due to a "$sig" signal | |
| 133703 | ERROR | Unable to getlocks on $me during resource create. | |
| 133704 | ERROR | The route53 on $Route53HostName is already protected by LifeKeeper. | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 133706 | ERROR | Error creating resource $Route53Tag. Error ($rc) | |
| 133708 | ERROR | Dependency create failed between $Route53Tag and $Route53IPResTag. Error ($rc). | |
| 133710 | ERROR | In-service attempted failed for tag $Route53Tag. | |
| 133813 | ERROR | Usage: $usage | |
| 133815 | ERROR | The host name "$hostname" is too long. | |
| 133816 | ERROR | The host name "$hostname" is too short. | |
| 133817 | ERROR | The host name "$hostname" contains invalid character. | |
| 133818 | ERROR | The first character must be an alpha character. | |
| 133819 | ERROR | The last character must not be a minus sign or period. | |
| 133826 | ERROR | Host Name cannot be blank. | |
| 134003 | ERROR | catch a \"$sig\" signal | **原因**: create処理がシグナルによって中断されました。<br><br>**処置**: 前後のログメッセージを確認してください。 |
| 134004 | ERROR | Unable to getlocks on $server during resource create. Error ($rc) | **原因**: リソース階層の作成時に管理ロックの取得に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 134005 | ERROR | The service \"$serviceName\" is not supported on $server. Error ($rc) | **原因**: サービスが存在しないか保護できないサービスです。<br><br>**処置**: 適切なサービス名を入力してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 134006 | ERROR | The service \"$serviceName\" is already protected on $server. | **原因**: すでに保護されているサービスです。<br><br>**処置**: このサービスを保護するためのリソースを作成することはできません。 |
| 134007 | ERROR | Error creating resource $tag. Error ($rc) | **原因**: QSPリソースの作成に失敗しました。<br><br>**処置**: 前後のログメッセージを確認して、エラーの原因を取り除いてください。 |
| 134011 | ERROR | In-service attempted failed for tag $tag. | **原因**: QSPリソースの起動に失敗しました。<br><br>**処置**: 保護したいサービスに関連したログを確認して、問題を解決してください。 |
| 134015 | ERROR | Unable to rlslocks on $server during resource create. Error ($rc) | **原因**: QSPリソースの作成後にロックの解除に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 134103 | ERROR | Template resource \"$template_tag\" on server \"$template_sys\" does not exist | **原因**: テンプレートシステムでリソースを見つけられません。<br><br>**処置**: テンプレートシステムで階層が正しいことを確認してから拡張してください。 |
| 134104 | ERROR | Template resource \”$template_tag\” on server \”$template_sys\” is not QSP resource (app=$ins[1], res=$ins[2]) | **原因**: テンプレートリソースはQSPリソースではありません。<br><br>**処置**: テンプレートリソースと同じタイ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | プに拡張してください。 |
| 134105 | ERROR | The service \"$service\" is not supported on $me. Error ($check) | **原因**: 拡張先のサーバーにサービスが存在しません。<br><br>**処置**: 拡張先サーバーにサービスをインストールしてから拡張してください。 |
| 134106 | ERROR | The service \"$service\" is already protected on $me. | **原因**: 拡張先サーバーに同じIDのリソースが既にあります。<br><br>**処置**: 同じサービスのリソースを作成することはできません。 |
| 134203 | ERROR | catch a \"$sig\" signal | **原因**: リソース拡張処理がシグナルによって中断されました。<br><br>**処置**: 前後のログメッセージを確認してください。 |
| 134204 | ERROR | Template resource \"$template_tag\" on server \"$template_sys\" does not exist | **原因**: テンプレートシステムでリソースを見つけられません。<br><br>**処置**: テンプレートシステムで階層が正しいことを確認してから拡張してください。 |
| 134208 | ERROR | Error creating resource \"$tag\" on server \"$me\" | **原因**: LifeKeeperはターゲットサーバーにリソースインスタンスを作成できませんでした。<br><br>**処置**: 前後のログメッセージを確認して、エラーの原因を取り除いてください。 |
| 134401 | ERROR | timeout $cmd for \"$tag\". Forcibly terminating. | **原因**: サービスの起動処理が規定時間を |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 超えても終了しません。<br><br>**処置**: 保護しているサービスについて確認し、restore操作を再試行してください。また、関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 134405 | FATAL | Failed to fork process to execute systemctl command: $! | **原因**: fork が失敗しました。これはシステムエラーです。<br><br>**処置**: fork が失敗する理由を特定してください。 |
| 134407 | ERROR | systemctl command has failed for \"$tag\" | **原因**: systemctl コマンドの実行に失敗しました。<br><br>**処置**: start オプションで systemctl コマンドを手動実行するとエラーになります。エラーメッセージを参考にしてエラーの原因を取り除いてください。 |
| 134501 | ERROR | timeout $cmd for \"$tag\". Forcibly terminating. | **原因**: サービスの停止処理が規定時間を超えても終了しません。<br><br>**処置**: 保護しているサービスについて確認し、remove操作を再試行してください。また、関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 134505 | FATAL | Failed to fork process to execute service command: $! | **原因**: forkが失敗しました。これはシステムエラーです。<br><br>**処置**: forkが失敗する理由を特定してください。 |
| 134507 | ERROR | service command has failed for \"$tag\" | **原因**: service コマンドの実行に失敗しま |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | した。<br><br>**処置**: stop オプションで service コマンドを手動実行するとエラーになります。エラーメッセージを参考にしてエラーの原因を取り除いてください。 |
| 134601 | ERROR | timeout $cmd for \"$tag\". Forcibly terminating. | **原因**: サービスの監視処理が規定時間を超えても終了しません。<br><br>**処置**: 保護しているサービスについて確認してください。また、関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 134605 | FATAL | Failed to fork process to execute service command: $! | **原因**: fork が失敗しました。これはシステムエラーです。<br><br>**処置**: fork が失敗する理由を特定してください。 |
| 134607 | ERROR | service command has failed for \"$tag\" | **原因**: service コマンドの実行に失敗しました。<br><br>**処置**: status オプションで service コマンドを手動実行するとエラーになります。エラーメッセージを参考にしてエラーの原因を取り除いてください。 |
| 134701 | ERROR | timeout $cmd for \"$tag\". Forcibly terminating. | **原因**: サービスの回復処理が規定時間を超えても終了しません。<br><br>**処置**: 保護しているサービスについて確認してください。また、関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 134706 | FATAL | Failed to fork process to execute service command: $! | **原因**: fork が失敗しました。これはシス |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | テムエラーです。<br><br>**処置**: fork が失敗する理由を特定してください。 |
| 134708 | ERROR | service command has failed for \"$tag\" | **原因**: service コマンドの実行に失敗しました。<br><br>**処置**: start オプションで service コマンドを手動実行するとエラーになります。エラーメッセージを参考にしてエラーの原因を取り除いてください。 |
| 134803 | ERROR | tag \"$tag\" does not exist on server \"$me\" | **原因**: 指定されたタグが存在しません。これは内部エラーです。 |
| 134804 | ERROR | app type \”$ins[1]\” is not $app | **原因**: 指定されたタグはQSPリソースではありません。これは内部エラーです。 |
| 134805 | ERROR | res type \”$ins[2]\” is not $res | **原因**: 指定されたタグはQSPリソースではありません。これは内部エラーです。 |
| 134823 | ERROR | tag \"$tag\" does not exist on server \"$me\" | **原因**: 指定されたタグが存在しません。これは内部エラーです。 |
| 134824 | ERROR | app type \”$ins[1]\” is not $app | **原因**: 指定されたタグはQSPリソースではありません。これは内部エラーです。 |
| 134825 | ERROR | res type \”$ins[2]\” is not $res | **原因**: 指定されたタグはQSPリソースではありません。これは内部エラーです。 |
| 134843 | ERROR | tag \"$tag\" does not exist | **原因**: 指定されたタグが存在しません。これは内部エラーです。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 134844 | ERROR | app type \"$ins[1]\" is not $app | 原因: 指定されたタグはQSPリソースではありません。これは内部エラーです。 |
| 134845 | ERROR | res type \"$ins[2]\" is not $res | 原因: 指定されたタグはQSPリソースではありません。これは内部エラーです。 |
| 135802 | ERROR | object[%d] not iterated for %lld ms. | |
| 135806 | ERROR | qwk_config() failed. | |
| 135807 | ERROR | thread_initialize() failed. | |
| 135808 | ERROR | state_monitor_initialize() failed. | |
| 135809 | ERROR | state_monitor() failed. | |
| 135810 | ERROR | start_server() failed. | |
| 135812 | ERROR | ‘fopen’ for %s failed: %m | |
| 135813 | ERROR | ‘qwk_object_path=’ line cannot be found for %s. | |
| 135814 | ERROR | ‘%s’ is unknown config. | |
| 135815 | ERROR | configuration of hbeattime ‘%d’ is incorrect. | |
| 135816 | ERROR | configuration of numhbeats ‘%d’ is incorrect. | |
| 135817 | ERROR | configuration of timeout_multiplier ‘%d’ is incorrect. | |
| 135818 | ERROR | configuration of lcmhbeattime ‘%d’ is incorrect. | |
| 135819 | ERROR | configuration of lcmnumhbeats ‘%d’ is incorrect. | |
| 135820 | ERROR | configuration of qwk_object_type is incorrect. | |
| 135821 | ERROR | configuration of my_node is incorrect. | |
| 135822 | ERROR | configuration of number_of_node ‘%d’ is incorrect. | |
| 135823 | ERROR | configuration of number_of_object ‘%d’ is incorrect. | |
| 135824 | ERROR | configuration of object node is incorrect. | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 135825 | ERROR | configuration of object path is incorrect. | |
| 135826 | ERROR | my_node is not include in qwk_objects. | |
| 135827 | ERROR | ‘open’ for %s failed: %m | |
| 135828 | ERROR | ‘read’ for %s failed: %m | |
| 135829 | ERROR | ‘popen’ for %s failed: %m | |
| 135830 | ERROR | ‘open’ for %s failed: %m | |
| 135831 | ERROR | ‘write’ for %s failed: %m | |
| 135832 | ERROR | ‘popen’ for %s failed: %m | |
| 135833 | ERROR | (bug) buffer overflow | |
| 135834 | ERROR | (bug) data is corrupted | |
| 135835 | ERROR | ‘signature=’ line cannot be found. | |
| 135836 | ERROR | signature ‘%s’ does not match. | |
| 135837 | ERROR | ‘local_node=’ line cannot be found. | |
| 135838 | ERROR | local_node ‘%s’ does not match. | |
| 135839 | ERROR | ‘time=’ line cannot be found. | |
| 135840 | ERROR | ‘sequence=’ line cannot be found. | |
| 135841 | ERROR | sequence ‘%s’ scan failed. | |
| 135842 | WARN | ‘node=’ line cannot be found. index=%d | |
| 135843 | ERROR | ‘commstat=’ line cannot be found. inde x=%d | |
| 135844 | ERROR | ‘checksum=’ line cannot be found. | |
| 135845 | ERROR | checksum ‘%s’ scan failed. | |
| 135846 | ERROR | checksum does not match. | |
| 135849 | ERROR | qwk object was not found. | |
| 135851 | ERROR | failed to read qwk object. | |
| 135852 | ERROR | failed to decode node_info. | |
| 135854 | WARN | sequence backed down from %llu to %llu. | |
| 135855 | ERROR | ‘malloc’ for %zu failed: %m | |
| 135856 | ERROR | thread_create() failed. index=%d | |
| 135870 | ERROR | (bug) data is corrupted | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 135871 | ERROR | format error in request. | |
| 135872 | ERROR | format error in quorum_verify request. lkevent cannot be found. | |
| 135873 | ERROR | format error in witness_verify request. lkevent cannot be found. | |
| 135874 | ERROR | format error in witness_verify request. target_node cannot be found. | |
| 135875 | ERROR | ‘%s’ is unknown command. | |
| 135877 | ERROR | qwk_receive() did not receive full header. Close socket. | |
| 135878 | ERROR | Request is too long. Close socket. | |
| 135879 | ERROR | qwk_receive() did not receive full request. Close socket. | |
| 135880 | ERROR | do_request() failed. | |
| 135881 | ERROR | qwk_send() did not send full header. Close socket. | |
| 135882 | ERROR | qwk_send() did not send full response. Close socket. | |
| 135884 | ERROR | qwk_accept() failed. | |
| 135885 | ERROR | thread_create() failed. | |
| 135886 | ERROR | cannot create socket. | |
| 135887 | ERROR | ‘bind’ failed: %m | |
| 135888 | ERROR | ‘listen’ failed: %m | |
| 135889 | ERROR | create_sockets() failed. | |
| 135890 | ERROR | thread_create() failed. | |
| 135891 | ERROR | ‘pthread_attr_init’ failed: %m | |
| 135892 | ERROR | ‘pthread_attr_setstacksize’ failed: %m | |
| 135893 | ERROR | ‘pthread_create’ failed: %m | |
| 135894 | ERROR | requst is too long. header.size=%zu | |
| 135895 | ERROR | qwk_send() failed for header. | |
| 135896 | ERROR | qwk_send() failed for request. | |
| 135897 | ERROR | qwk_send() failed for termination. | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 135898 | ERROR | qwk_receive() did not receive full header. | |
| 135899 | ERROR | Response buffer is not enough large. Ser ver sent %zu bytes. | |
| 135900 | ERROR | qwk_receive() did not receive full respons e. | |
| 135901 | ERROR | ‘%s’ is unknown command. | |
| 135903 | ERROR | Cannot create socket. | |
| 135904 | ERROR | ‘connect’ failed: %m | |
| 135905 | ERROR | request_send() failed. | |
| 135906 | ERROR | request_receive() failed. | |
| 135907 | ERROR | ‘%s’ is unknown lkevent. | |
| 135908 | ERROR | ‘%s’ is unknown qwktype. | |
| 135909 | ERROR | ‘%s’ is unknown node state. | |
| 135910 | ERROR | ‘%s’ is unknown quorum state. | |
| 135911 | ERROR | ‘accept’ failed: %m | |
| 135912 | ERROR | You must install the LifeKeeper license ke y for Storage Quorum Witness Kit. | |
| 135913 | EMERG | WARNING: Cannot find the configuration for node ‘%s’ in the configuration file. If th e configuration of the cluster is changed a fter the initialization, it needs to be initializ ed again. Please see the SIOS product d ocumentation for information on how to re initialize. | |
| 135914 | WARN | The first cycle did not complete before tim eout (%lld ms). | 原因: StorageモードでのQuorumオブジェクトへの初回書き込みが表示の時間(単位はミリ秒)でタイムアウトしました。<br><br>処置: QWK_STORAGE_HBEATTIME、QWK_STORAGE_NUMHBEATSの設定値を大きくするか、本メッセージ出力後に135802のメッセージが頻繁に出力される場合はQuorum/WitnessのstoragemodeページにあるトラブルシューティングのTroubleshooting項を参照ください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 135916 | WARN | The first cycle did not complete before timeout (%lld ms). | 原因: StorageモードでのQuorumオブジェクトへの初回書き込みが表示の時間(単位はミリ秒)でタイムアウトしました。<br><br>処置: QWK_STORAGE_HBEATTIME、QWK_STORAGE_NUMHBEATSの設定値を大きくするか、本メッセージ出力後に135802のメッセージが頻繁に出力される場合はQuorum/Witnessのstorageモードページにあるトラブルシューティングの項を参照ください。 |
| 135917 | ERROR | params is NULL. return 0 as timeout msec. | 原因: 内部の処理で問題が発生しました。<br><br>処置: 調査を依頼してください。 |
| 135998 | EMERG | An unrecoverable error was detected in qwk_storage_server. Stop restarting qwk_storege_server. | |
| 135999 | ERROR | COMMAND OUTPUT: $LKROOT/lkadm/bin/qwk_storage_server $LKROOT/config/qwk-storage.cfg | |
| 136002 | ERROR | Usage: $usage | 原因: SAP HANA createスクリプトに無効な引数が指定された。<br><br>処置: 次の形式で適切な引数を指定してください：<br><リソースタグ> <SAP SID> <HDBインスタンス> [スイッチバックタイプ] [仮想IPリソースタグ] |
| 136003 | ERROR | END failed create of resource $tag on server $me with return value of $errcode. | 原因: SAP HANAリソースの作成中に障害が発生した。<br><br>処置: SAP HANAシステムレプリケーションがプライマリとセカンダリの両方のレプリケーションサイトで完全に構成お |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | よび有効化されていることを確認し、リソース作成操作を再試行してください。 |
| 136005 | ERROR | An unknown error has occurred in utility rlslocks on server $me. View the LifeKeeper logs for details and retry the operation. | 原因: SAP HANAリソースの作成中にLifeKeeper rlslocksユーティリティで問題が発生する。<br><br>処置: LifeKeeperログファイルで見つかった問題を解決し、リソース作成操作を再試行してください。 |
| 136006 | ERROR | END failed create of resource $tag on server $me with signal $sig. | 原因: SAP HANAリソース作成中のシグナルによる割込み。<br><br>処置: 詳細については、LifeKeeperログファイルを確認してください。 |
| 136008 | ERROR | An unknown error has occurred in utility getlocks on server $me. View the LifeKeeper logs for details and retry the operation. | 原因: SAP HANAリソースの作成中にLifeKeeper getlocksユーティリティで問題が発生する。<br><br>処置: LifeKeeperログファイルで見つかった問題を解決し、リソース作成操作を再試行してください。 |
| 136009 | ERROR | The SAP HANA product was not found in the directory $obj->{'hana_util_path'} on server $me. Verify that SAP HANA is correctly installed and that all necessary file systems are mounted. | 原因: リソースの作成中に、必須のSAP HANAバイナリが見つからなかった。<br><br>処置: クラスター内のすべてのサーバーでSAP HANAが正しくインストールされ、構成されていることを確認してください。 |
| 136010 | ERROR | Failed to create resource as id $id already exists on system $me. | 原因: 指定されたIDのLifeKeeperリソースがすでにシステムに存在する。<br><br>処置: SAP HANAデータベースがLifeKeeperによってすでに保護されているかど |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | うかを確認してください。 |
| 136011 | ERROR | Failed to create new tag $tag for SAP HANA resource on $me. | 原因: 入力されたSAP HANAリソースタグは別のLifeKeeperリソースによってすでに使用されており、LifeKeeper newtagユーティリティは新しいタグを作成できなかった。<br><br>処置: SAP HANAリソースに対し、別のタグ名を選択してください。 |
| 136012 | ERROR | Failed creation of resource with $tag on system $me. | 原因: LifeKeeperで指定したSAP HANAリソースを作成できなかった。<br><br>処置: LifeKeeperログファイルで見つかった問題を解決し、リソース作成操作を再試行してください。 |
| 136014 | ERROR | Failed to create resource dependency for parent $tag and child $virtual_ip_tag. | 原因: SAP HANAリソースとそれが依存する仮想IPリソース間の依存関係の作成に失敗した。<br><br>処置: LifeKeeperログファイルで見つかった問題を解決し、リソース作成操作を再試行してください。 |
| 136015 | ERROR | The info field for resource $tag could not be successfully generated using values [SID: $info_sid, Instance: $info_instance, Replication Mode: $info_repl_mode, Site Name: $info_site_name, Operation Mode: $info_oper_mode]. Please verify that SAP HANA System Replication is fully configured and enabled on both the primary and secondary systems before creating the SAP HANA resource. | 原因: SAP HANAの「リソース情報」フィールドの作成時に無効な値が見つかった。<br><br>処置: SAP HANAが正しくインストールおよび構成されていること、SAP HANAシステムレプリケーションがクラスター内のすべてのサーバーで完全に構成され有効になっていることを確認してください。 |
| 136016 | ERROR | The selected server $me is not the primary/source system for SAP HANA System R | 原因: リソースの作成が、HANAシステ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | eplication for the selected SID $sid and HDB instance $instance. Please select 'Cancel' and start this action on the primary/source HANA System Replication system. | ムレプリケーションのセカンダリシステムで開始されている。<br><br>**処置**: HANAシステムレプリケーションのプライマリシステムで作成操作を開始してください。 |
| 136031 | ERROR | END failed extend of resource $target_tag on server $me with return value of $err_code. | **原因**: SAP HANAリソースの拡張中に問題が発生した。<br><br>**処置**: LifeKeeperログファイルで見つかった問題を解決し、リソース拡張操作を再試行してください。 |
| 136033 | ERROR | Usage: $usage | **原因**: SAP HANA extendスクリプトに無効な引数が指定された。<br><br>**処置**: 次の形式で適切な引数を指定してください：<br><テンプレートシステム> <テンプレートタグ> <スイッチバックタイプ> <ターゲットタグ> |
| 136035 | ERROR | Template resource $template_tag on server $template_sys does not exist. | **原因**: 拡張するテンプレートSAP HANAリソースがテンプレートサーバー上に存在しない。<br><br>**処置**: テンプレートサーバー上に、拡張されているSAP HANAリソースが存在することを確認してください。 |
| 136036 | ERROR | Resource with matching id $target_id already exists on server $me for App $app_type and Type $res_type. | **原因**: 同じLifeKeeper IDのSAP HANAリソースがターゲットシステムにすでに存在する。<br><br>**処置**: SAP HANAデータベースが、ターゲットサーバー上のLifeKeeperによってすでに保護されているかどうかを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 136037 | ERROR | Resource with matching tag $target_tag already exists on server $me for App $app_type and Type $res_type | **原因**: 同じLifeKeeperリソースタグを持つSAPリソースがターゲットサーバーに既に存在する。<br><br>**処置**: SAP HANAデータベースが、ターゲットサーバー上のLifeKeeperによってすでに保護されているかどうかを確認してください。 |
| 136039 | ERROR | Error creating resource $target_tag on system $me. | **原因**: ターゲットサーバーで同等のSAP HANAリソースを作成できなかった。<br><br>**処置**: LifeKeeperログファイルで見つかった問題を解決し、リソース拡張操作を再試行してください。 |
| 136040 | ERROR | The target tag ($target_tag) and template tag ($template_tag) must be the same. | **原因**: プライマリーシステムで使用されるリソースタグ名が、セカンダリーシステムで使用されるリソースタグ名と異なる。 両方の名前は同じでなければならない。<br><br>**処置**: リソースの作成では、プライマリーシステムタグおよびセカンダリーシステムタグと同じ名前を使用してください。 |
| 136041 | ERROR | Error getting resource information for $template_tag on server $template_sys. | **原因**: 指定されたサーバー上の指定された SAP HANA リソースに関する情報を取得できませんでした。<br><br>**処置**: 指定されたタグを持つ SAP HANA リソースが指定されたサーバーに存在することを確認してください。 |
| 136042 | ERROR | Failed to update info fields for equivalent resources while extending resource $target_tag on $me. | **原因**: 指定されたサーバー上の指定された SAP HANA リソースに関する情報を取得できませんでした。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 指定されたタグを持つ SAP HANA リソースが指定されたサーバーに存在することを確認してください。 |
| 136045 | ERROR | Cannot extend resource $template_tag to server $me. | **原因**: SAP HANA canextendスクリプトは、リソースを特定のターゲットシステムに拡張できないことを示している。<br><br>**処置**: LifeKeeperログファイルで見つかった問題を解決し、リソース拡張操作を再試行してください。 |
| 136047 | ERROR | Usage: $usage | **原因**: SAP HANA canextendスクリプトに無効な引数が指定された。<br><br>**処置**: 次の形式で適切な引数を指定してください：<br><テンプレートシステム> <テンプレートタグ> |
| 136048 | ERROR | Resource $template_tag does not exist on server $template_sys. | **原因**: 拡張するテンプレートSAP HANAリソースがテンプレートサーバー上に存在しない。<br><br>**処置**: テンプレートサーバー上に、拡張しているSAP HANAリソースが存在することを確認してください。 |
| 136049 | ERROR | The system user $hana_user does not exist on server $me. | **原因**: SAP HANAデータベースのSAP管理ユーザーが、指定されたサーバーに存在しない。<br><br>**処置**: SAP HANAが正しくインストールおよび構成されていることを確認してください。 |
| 136050 | ERROR | The user id for user $hana_user ($template_uid) on template server $template_sys is not the same as user id ($uid) on target | **原因**: SAP管理ユーザーのユーザーIDが、テンプレートサーバーとターゲット |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | server $me. | サーバーで異なっている。<br><br>処置: SAPのHANAがクラスター内のすべてのサーバーに適切にインストールおよび構成されていることを確認してください。 |
| 136051 | ERROR | The group id for user $hana_user ($template_gid) on template server $template_sys is not the same as group id ($gid) on target server $me. | 原因: SAP管理ユーザーのグループIDが、テンプレートサーバーとターゲットサーバーで異なっている。<br><br>処置: SAPのHANAがクラスター内のすべてのサーバーに適切にインストールおよび構成されていることを確認してください。 |
| 136052 | ERROR | The home directory for user $hana_user ($template_home) on template server $template_sys is not the same as home directory ($user_home) on target server $me. | 原因: SAP管理ユーザーのホームディレクトリが、テンプレートサーバーとターゲットサーバーで異なっている。<br><br>処置: SAPのHANAがクラスター内のすべてのサーバーに適切にインストールおよび構成されていることを確認してください。 |
| 136053 | ERROR | The SAP HANA instance $instance does not exist for $sid on server $me. | 原因: 指定されたSAP HANAデータベースインスタンスのインストールディレクトリが見つからなかった。<br><br>処置: SAPのHANAがクラスター内のすべてのサーバーに適切にインストールおよび構成されていることを確認してください。 |
| 136055 | ERROR | The SAP HANA site name $target_obj->{‘site_name’} on server $me must be different from site name $template_obj->{‘site_name’} on $template_obj->{‘sys’}. | 原因: SAP HANAシステムレプリケーションのサイト名が、プライマリサーバーとセカンダリサーバーの両方で同じになっている。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: セカンダリサーバー上のSAP HANAデータベースを停止し、hdbnsutilユーティリティを使用して、別のサイト名でセカンダリレプリケーションサイトを再登録してください。 |
| 136056 | ERROR | Unable to obtain SAP HANA System Replication parameters for database $instance on server $me. Please verify that SAP HANA System Replication is enabled and properly configured and that the database instance is running on all servers in the cluster. | **原因**: 指定されたシステム上のデータベースのSAP HANAシステムレプリケーションパラメーターを決定できなかった。<br><br>**処置**: SAP HANAシステムレプリケーションが有効で適切に構成されていること、およびデータベースがクラスター内のすべてのサーバーで実行されていることを確認してください。 |
| 136057 | ERROR | Unable to create a HANA object for database $instance on server $me. Please verify that database instance $instance is properly installed. | **原因**: 指定したサーバー上の指定したインスタンスを表す内部HANAオブジェクトを作成できない。<br><br>**処置**: データベースインスタンスが正しくインストールされており、必要なすべてのファイルシステムがマウントされていることを確認してください。 |
| 136062 | ERROR | Invalid operation mode: $oper_mode. Acceptable values: $valid_oper. | **原因**: マルチターゲット構成で SAP HANA リソースを拡張するときに、無効な操作モードが使用されました。<br><br>**処置**: 許容される操作モードは、メッセージにリストされています。 |
| 136065 | ERROR | Failed to update the instance info field. | **原因**: SAP HANA リソースの情報フィールドは、マルチターゲットサポートを許可するように更新されていません。<br><br>**処置**: クラスター内すべてのシステムで LifeKeeper が実行されており、リソースが in-service であることを確認してくだ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | さい。 |
| 136066 | ERROR | The SAP HANA resource $template_tag is configured on the maximum (%d) supported number of servers. | 原因: LifeKeeper HANA リカバリーキットがサポートする最大数のサーバーに SAP HANA リソースが構成されています。<br><br>処置: LifeKeeper HANA Recovery Kit でサポートされるサーバーの最大数は変更できません。 |
| 136067 | ERROR | The SAP HANA resource $template_tag with \"delta_datashipping\" is configured on the maximum (2) supported number of servers. | 原因: SAPは、マルチターゲット構成でのdelta_datashippingをサポートしていません。<br><br>処置: サポートされているオペレーションモードを使用してください。 |
| 136083 | ERROR | Usage: $usage | 原因: SAP HANA deleteスクリプトの引数が無効。<br><br>処置: 有効なHANA LifeKeeperリソースタグ名とリソースIDの両方を入力してください。<br>入力方法：delete -t <tag> -i <id> [-U] |
| 136096 | ERROR | One of the required parameters (server, tag, or action) was missing. Unable to set the local recovery policy. | 原因: 必要なパラメーター（サーバー、タグ、アクション）のいずれかが指定されていません。<br><br>処置: 必要なパラメーターを指定して、操作を再試行してください。 |
| 136097 | ERROR | Failed to $flg_action local recovery for SAP HANA resource $tag on server $node. | 原因: 指定されたサーバー上の指定されたSAP HANAリソースのローカルリカバリーの有効化または無効化に失敗しました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: LifeKeeperが実行中で、完全に初期化されていることを確認してから、操作を再試行してください。 |
| 136099 | ERROR | Failed start of SAP Host Agent on server $sys. | **原因**: 指定されたサーバーでSAP Host Agentプロセス（saphostexec, saposcol）を開始できませんでした。<br><br>**処置**: SAP Host Agentプロセスのトレースログを確認し、見つかった問題を修正してから、操作を再試行してください。 |
| 136100 | ERROR | Failed start of SAP Host Agent on server $sys. | **原因**: 指定されたサーバーでSAP Host Agentプロセス（saphostexec, saposcol）を開始できませんでした。<br><br>**処置**: SAP Host Agentプロセスのトレースログを確認し、見つかった問題を修正してから、操作を再試行してください。 |
| 136101 | ERROR | Failed to register server $sys as a secondary SAP HANA System Replication site. | **原因**: 指定されたサーバーを指定されたデータベースのセカンダリーサイトとして SAP HANA システムレプリケーションに登録できませんでした。<br><br>**処置**: 詳細については、SAP HANA トレースファイル (nameserver_<hostname>.xxxxx.xxx.trc など) を調べてください。 |
| 136119 | WARN | Resource $eqv_tag not found on server $sys. | **原因**: サイトの情報を生成している間、指定されたサーバー上のリソースを更新できませんでした。<br><br>**処置**: 指定されたタグを持つ SAP HANA リソースが、指定されたサーバーに存在することを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 136120 | ERROR | Failed to update site name for $sys, site name not defined. | **原因**: サイト名が定義されていないため、SAP HANA サイト名の検証に失敗しました。<br><br>**処置**: 検証中の SAP HANA リソースに有効なサイト名があることを確認してください。 |
| 136121 | ERROR | Failed to update replication mode for $sys, invalid replication mode \"$repl_mode\" not permitted. | **原因**: レプリケーションモードが定義されているが許可されていないため、SAP HANA レプリケーションモードの検証に失敗しました。<br><br>**処置**: 検証中の SAP HANA リソースに有効なレプリケーションモードがあることを確認してください。 |
| 136122 | ERROR | Failed to update operation mode for $sys, invalid operation mode \"$oper_mode\" not permitted. | **原因**: オペレーションモードが定義されているが許可されていないため、SAP HANA オペレーションモードの検証に失敗しました。<br><br>**処置**: 検証中の SAP HANA リソースの操作モードが有効であることを確認してください。 |
| 136124 | ERROR | Failed to update info field for resource $eqv_tag on server $sys from [$orig_info] to [$new_info]. | **原因**: LifeKeeper で内部エラーが発生しました。<br><br>**処置**: 構成によっては、情報フィールドのロールバックが試行される場合があります。それ以外の場合は、手動での更新が必要になる場合があります。 |
| 136127 | WARN | WARNING: Failed to roll back to original info fields after failed update. Temporary mismatches may exist in connection types between servers until they can be successfully synchronized. | **原因**: LifeKeeper で内部エラーが発生しました。<br><br>**処置**: ins_list インフォフィールドをロールバックしようとして失敗しました。 q |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | uickCheck サイクル中に別の試行が行われる場合があります。それ以外の場合は、手動構成が必要になる場合があります。 |
| 136152 | WARN | There was an error retrieving the HSR status from the \"$self->{'hana_sysreplstatus_script'}\" script for site $site_id, setting the value to \"$value\". | 原因: SAP HANA システムレプリケーションのステータスを取得しようとして失敗しました。<br><br>処置:LifeKeeper と SAP のログファイルを調べて、失敗の原因を特定し、見つかった問題を手動で修正してください。SAP HANA リソースが in-service である間、LifeKeeper は自動的にステータスの読み取りと更新を継続して利用可能にしてください。 |
| 136160 | ERROR | Unable to create SAP HANA object. The SID and instance for the SAP HANA database must be provided. | 原因: SAP HANA オブジェクトの作成中に、SAP SID または SAP HANA インスタンス名のいずれかが欠落しています。<br><br>処置: 詳細については、LifeKeeper ログファイルを確認してください。 |
| 136161 | ERROR | Unable to create SAP HANA object. Resource system name and tag name must be provided. | 原因: SAP HANAオブジェクトの作成中に、システム名またはリソースタグ名のいずれかが欠落していた。<br><br>処置: 詳細については、LifeKeeperログファイルを確認してください。 |
| 136162 | ERROR | Could not find any information regarding resource $tag on $sys. | 原因: 指定されたタグを持つSAP HANAリソースに関する情報を取得できなかった。<br><br>処置: 指定されたタグを持つSAP HANAリソースが指定されたサーバーに存在することを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 136168 | ERROR | Unable to check status of SAP Host Agent on server $self->{'sys'}. Command \"$curr_cmd\" returned exit code $ret. | **原因**: 指定されたサーバーでSAP Host Agentプロセスのステータスを判別できなかった。<br><br>**処置**: 詳細については、SAP Host Agentトレースファイル（dev_saphostexecなど）を確認してください。 |
| 136174 | ERROR | Unable to create SAP HANA object. The SID or instance value is missing. | **原因**:SAP HANA オブジェクトの作成時に、SAP SID、SAP HANA インスタンス名、または SAP HANA システムレプリケーション値のいずれかが欠落しています。<br><br>**処置**: 詳細については、LifeKeeper ログファイルを確認してください。 |
| 136182 | ERROR | Failed to $flg_action flag \"${HANA_FLAG_DATA_OUT_OF_SYNC}_$eqv_tag{$sys}\" on server $sys. | **原因**: 指定されたサーバーで !HANA_DATA_OUT_OF_SYNC_<tag> フラグを作成または削除できませんでした。<br><br>**処置**: 詳細については、LifeKeeper ログファイルを確認してください。 |
| 136190 | ERROR | Failed to start SAP Host Agent processes on server $self->{'sys'}. Command \"$hostagent_cmd\" returned $ret. | **原因**: 指定されたサーバーでSAP Host Agentプロセスを開始できませんでした。<br><br>**処置**: 詳細については、SAP Host Agentトレースファイル（dev_saphostexecなど）を確認してください。 |
| 136191 | ERROR | Failed to start SAP OS Collector process on server $self->{'sys'}. Command \"$oscol_cmd\" returned $ret. | **原因**: 指定されたサーバーで SAP OS Collector プロセスを開始できませんでした。<br><br>**処置**: 詳細については、SAP OS コレクタートレースファイル（dev_collなど）を確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 136193 | ERROR | $takeover_text of SAP HANA System Replication for SAP HANA database $self->{'instance'} failed on server $node with exit code $ret. | 原因: SAP HANAシステムレプリケーションで、指定されたデータベースのプライマリマスターとして指定されたサーバーを登録できませんでした。<br><br>処置: 詳細については、SAP HANA トレースファイル（例：nameserver_<hostname>.xxxxx.xxx.trc）を確認してください。 |
| 136197 | ERROR | Update of resource info field for $eqv_tag{$sys} on $sys failed with exit code $setinfo_ret. Current info: [$info]. Attempted new info: [$new_info]. | 原因: 指定されたサーバー上の指定されたリソースの情報フィールドを更新できませんでした。<br><br>処置: 詳細については、LifeKeeper ログファイルを確認してください。 |
| 136202 | EMERG | Failed to disable Autostart for SAP HANA instance $self->{‘instance’} on server $sys with exit code $remexec_ret. Please manually set \"Autostart = 0\" in the instance profile $profile on $sys. | 原因: Autostart パラメーターの値を、指定されたサーバー上の指定された HDB インスタンスプロファイルで変更できませんでした。<br><br>処置: HDB インスタンスプロファイルを手動で編集し、「Autostart = 0」に設定してください。 |
| 136205 | ERROR | Failed start of SAP Start Service for SAP HANA database $instance on server $sys. | 原因: 指定された SAP HANA データベースの SAP Start Service の開始に失敗しました。<br><br>処置: 詳細については、SAP Start Service トレースファイル（sapstartsrv.logなど）を確認してください。 |
| 136208 | ERROR | LifeKeeper was unable to determine the SAP HANA System Replication mode for database $rem_obj->{‘instance’} on server $rem_obj->{‘sys’} while attempting to identify the previous primary replication site. Please resolve the issue and bring the SAP HANA resource in-service on the system | 原因: 指定されたサーバーで SAP HANA システムレプリケーションモードを判別できず、以前のプライマリレプリケーションサイトを特定できませんでした。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | where the database should be registered as primary master. | **処置**: 詳細については、LifeKeeper ログファイルを確認してください。 |
| 136210 | ERROR | Failed start of SAP Start Service for SAP HANA database $rem_obj->{‘instance’} on server $rem_obj->{‘sys’}. Unable to stop the database on the server where it is currently registered as primary master. | **原因**: 指定されたサーバー上の指定された SAP HANA データベースの SAP Start Service の開始に失敗し、その結果現在のプライマリー SAP HANA システムレプリケーションサイトでデータベースを停止できませんでした。<br><br>**処置**: 詳細については、SAP Start Serviceトレースファイル（sapstartsrv.logなど）を確認してください。 |
| 136212 | ERROR | Failed stop of SAP HANA database $rem_obj->{‘instance’} on server $rem_obj->{‘sys’} where it is currently registered as primary master. | **原因**: 指定されたサーバー上の指定された SAP HANA データベースの停止に失敗しました。<br><br>**処置**: 詳細については、SAP HANA トレースファイルと LifeKeeper ログファイルを確認してください。 |
| 136217 | ERROR | Failed start of SAP HANA database $instance on server $sys. | **原因**: 指定されたサーバー上の指定されたSAP HANAデータベースの起動に失敗した。<br><br>**処置**: 詳細については、SAP HANAトレースファイルとLifeKeeperログファイルを確認してください。 |
| 136220 | ERROR | Unable to register $sys as a secondary SAP HANA System Replication site for database $instance. The host name of the current primary replication site was not provided. | **原因**: セカンダリレプリケーションサイトの登録時に、現在のプライマリSAP HANAシステムレプリケーションサイトのホスト名を入力しなかった。<br><br>**処置**: 詳細については、LifeKeeperログファイルを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 136233 | EMERG | WARNING: A temporary communication failure has occurred between servers $self->{'sys'} and $sys. Manual intervention is required in order to minimize the risk of data loss. To resolve this situation, please take one of the following resource hierarchies out of service: $self->{'tag'} on $self->{'sys'} or $eqv{$sys} on $sys. The server that the resource hierarchy is taken out of service on will become the standby server for SAP HANA database $self->{'instance'}. | 原因: 一時的な通信障害により、指定された同等の HANA リソースがそれぞれのホストサーバーで同時に In Service になった。<br><br>処置: セカンダリーレプリケーションサイトになるサーバーで、HANA リソース階層全体を Out of Service にしてください。そのサーバーでデータベースが停止されると、LifeKeeper は次の quickCheck サイクル中にデータベースをセカンダリレプリケーションサイトとして自動的に登録してください。 |
| 136234 | EMERG | WARNING: SAP HANA database $self->{'instance'} is running and registered as primary master on the following servers: $primary_node_list. Manual intervention is required in order to minimize the risk of data loss. To resolve this situation, please stop database $self->{'instance'} on the intended secondary replication site by running the command \'su – $self->{'sid_admin'} -c \"$SAP_CONTROL -nr $self->{'instance_number'} -function StopWait $HANA_STOP_WAIT 5\"\' on that server, then bring resource $self->{'tag'} in-service on the intended primary replication site. Once the resource is in-service, the stopped database will automatically be re-registered as a secondary replication site. | 原因: 指定された SAP HANA データベースが、2つのクラスターサーバーで同時にプライマリーマスターとして実行および登録されている。<br><br>処置: メッセージに示されているコマンドを使用して、スタンバイサーバー上のデータベースを停止してください。データベースが停止すると、LifeKeeperは自動的にスタンバイサーバーをセカンダリレプリケーションサイトとして登録してください。 |
| 136236 | ERROR | Failed to remove ${HANA_FLAG_DATA_OUT_OF_SYNC}_$tag flag on server $sys. | 原因: 指定されたサーバーで !HANA_DATA_OUT_OF_SYNC_ LifeKeeper フラグを削除できませんでした。指定された SAP HANA リソースは、これが削除されるまで指定されたサーバーで In-Service になりません。<br><br>処置: LifeKeeper が実行中で、完全に初期化されていることを確認してください。SAP HANA システムレプリケーションが同期していることを確認できた場 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 合は、コマンド "/opt/LifeKeeper/bin/flg_remove -f '!HANA_DATA_OUT_OF_SYNC_<HANA Tag>'" を使用して out-of-sync フラグを手動で削除できます |
| 136238 | ERROR | Unable to create SAP HANA object. Resource system name and tag name must be provided. | 原因: SAP HANAオブジェクトの作成時に、サーバー名またはリソースタグ名のいずれかが欠落していた。<br><br>処置: 詳細については、LifeKeeperログファイルを確認してください。 |
| 136239 | ERROR | Failed start of SAP Start Service for SAP HANA database $obj->{'instance'} on server $obj->{'sys'}. Unable to determine status of the database on $obj->{'sys'}. | 原因: 指定されたサーバー上の指定されたSAP HANAデータベースのSAP Start Serviceの開始に失敗し、その結果データベースのステータスを判別できなかった。<br><br>処置: 詳細については、SAP Start Serviceトレースファイル（sapstartsrv.logなど）を確認してください。 |
| 136242 | ERROR | Unable to create SAP HANA object. At least one of the SAP HANA System Replication values is missing. | 原因: 指定したシステム上のデータベースに対し、SAP HANA システムレプリケーションパラメーターを特定できませんでした。<br><br>処置: SAP HANA システムレプリケーションが有効になっていて適切に構成されていること、データベースがクラスター内のすべてのサーバーで実行されていることを確認してください。 |
| 136243 | ERROR | Unable to locate the pingnfs utility ($pingnfs) on server $me. Please verify that this utility exists and is executable. | 原因: エクスポートされたNFS共有で利用可能なテストに使用されるpingnfsユーティリティが見つからない。<br><br>処置: pingnfsユーティリティが所定の場所に存在し、実行可能であることを確認 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | してください。 |
| 136245 | ERROR | Critical NFS shares being exported by server $sys ($export_list) are not currently available. Please verify that the NFS server is alive and that all NFS-related services are running. | **原因**: 指定した重要なNFS共有ファイルシステムが現在利用できない。<br><br>**処置**: ファイルシステムをエクスポートするNFSサーバーが稼働していること、および必要なNFS関連サービスがすべて実行されていることを確認してください。 |
| 136248 | ERROR | Unable to open file $crit_mount_file on server $me. Please verify that this file exists and is read-enabled. | **原因**: 指定したサーバー上の重要なNFS共有のマウント情報を含むファイルを開けない。<br><br>**処置**: ファイルが指定した場所に存在し、読み取りが有効になっていることを確認してください。 |
| 136253 | ERROR | The SAP HANA \"takeover with handshake\" feature is available only for SAP HANA versions 2.0 SPS04 and greater. Database $self->{'instance'} cannot be resumed. | **原因**: 一時停止されたデータベースの再開は、2.0 SPS04より前のSAP HANAバージョンではサポートされていない。<br><br>**処置**:「Takeover with Handshake」に関連する機能を使用するには、SAP HANA 2.0 SPS04以降にアップグレードしてください。 |
| 136254 | ERROR | Attempt to resume suspended primary database $self->{'instance'} on server $self->{'sys'} failed with exit code $ret. | **原因**: 指定したサーバーで中断されたデータベースインスタンスを再開しようとしたら失敗した。<br><br>**処置**:LifeKeeperとSAPのログファイルを調べて、失敗の原因を特定してください。操作を再試行するか、対応するLifeKeeperリソースを別のサーバーでIn-Serviceにしてください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 136258 | ERROR | Attempt to register server $node as the secondary SAP HANA System Replication site for database $self->{'instance'} failed with exit code $ret. | 原因: 指定したサーバーを、指定したデータベースインスタンスのセカンダリ SAP HANA システムレプリケーションサイトとして登録しようとしたが失敗しました。<br><br>処置:LifeKeeper と SAP のログファイルを調べて障害の原因を特定し、見つかった問題を手動で修正してください。SAP HANA リソースが In-Service になっている間、LifeKeeper は自動的に、セカンダリ HSR ロールへのバックアップサーバーの登録試行を継続してください。 |
| 136263 | ERROR | Usage: $usage | 原因: SAP HANAリソースのリストアスクリプトの引数が無効。<br><br>処置:次の形式で適切な引数を指定してください：<br>-t <リソースタグ> -i <リソースID> |
| 136265 | ERROR | Error getting resource information for $tag on server $me. | 原因: 指定されたサーバーの指定されたSAP HANAリソースに関する情報を取得できなかった。<br><br>処置:指定されたタグを持つSAP HANAリソースが指定されたサーバーに存在することを確認してください。 |
| 136266 | ERROR | The resource $tag protecting SAP HANA database $instance is not in sync. To protect the data LifeKeeper will not restore the resource on $me. Please restore the resource on the previous source server to allow the resync to complete. | 原因: SAP HANA システムレプリケーションは、バックアップサーバー上でデータベースリソースを In Service にする前に同期されていませんでした。<br><br>処置: 前のプライマリーサーバーで SAP HANA リソースを In Service にし、再同期を完了できるようにしてください。 |
| 136275 | ERROR | Failed to determine SAP HANA System Replication mode for database $instance on server $me. | 原因: 指定されたサーバー上の指定され |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | たデータベースのSAP HANAシステムレプリケーションモードを判定できなかった。<br><br>**処置**: 詳細については、SAP HANAトレースファイル（例：nameserver_<hostname>.xxxxx.xxx.trc）およびLifeKeeperログファイルを確認してください。 |
| 136277 | ERROR | The resource $tag protecting SAP HANA database $instance has the last owner flag on $other->{‘sys’} indicating there are changes on $other->{‘sys’} that have not replicated to all systems. | **原因**: データが同期されていないシステムが複数あります。これは SAP HANA リソースが In Service となっており、データがすべてのスタンバイシステムに同期されていないためです。<br><br>**処置**: どのシステムに最適なデータがあるか、または最大のデータがあるかを手動で判断してください。SAP HANAリソースは、最適なデータを持つシステムで強制的に in-service にする必要があります。 |
| 136300 | WARN | Detected unsupported NFS version \"$found_nfs_vers\" for mount point $mount_loc from shared file system $mount_dev. The default NFS version specified by the pingnfs utility will be used instead. If necessary, this value can be overridden by setting the NFS_VERSION parameter in /etc/default/LifeKeeper. | **原因**: NFS バージョンはサポートされていません。 LifeKeeper は、バージョン 2、3、および 4 をサポートしています。<br><br>**処置**: サポートされている NFS バージョンを使用してください。 |
| 136301 | WARN | Detected unsupported NFS transport protocol \"$found_proto\" for mount point $mount_loc from shared file system $mount_dev. The default protocol specified by the pingnfs utility will be used instead. If necessary, this value can be overridden by setting the NFS_RPC_PROTOCOL parameter in /etc/default/LifeKeeper. | **原因**: NFS プロトコルはサポートされていません。 LifeKeeper は tcp と udp をサポートしています。<br><br>**処置**: サポートされているプロトコルを使用してください。 |
| 136302 | ERROR | Failed to create $action_flag flag on server $node. | **原因**: LifeKeeper HANA Recovery Kit はフラグを作成できませんでした。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: LifeKeeper ログとシステムログでエラーを確認してください。 |
| 136351 | ERROR | Usage: $usage | **原因**: SAP HANA resource quickCheck スクリプトの引数が無効です。<br><br>**処置**: 次の形式で適切な引数を指定してください：<br>-t <リソースタグ> -i <リソースID> |
| 136353 | ERROR | Error getting resource information for $tag. | **原因**: 指定された SAP HANA リソースに関する情報を取得できませんでした。<br><br>**処置**: 指定されたタグを持つ SAP HANA リソースが指定されたサーバーに存在することを確認してください。 |
| 136354 | EMERG | The replication mode of the SAP HANA database $instance corresponding to resource $tag was modified outside of LifeKeeper and is no longer registered as primary master on server $me. Please bring the SAP HANA resource in-service on the server where the database should be registered as primary master. Resource monitoring for $tag will be suspended until the issue is resolved. | **原因**: 指定されたデータベースの SAP HANA システムレプリケーションモードが、LifeKeeper の外部で変更されています。<br><br>**処置**: プライマリーマスターとして登録する必要があるサーバーで、SAP HANA リソースを In Service にしてください。 |
| 136363 | EMERG | LifeKeeper was unable to determine the SAP HANA System Replication mode for database $instance corresponding to resource $tag on server $me. Resource monitoring for $tag will be suspended until the issue is resolved. | **原因**: 指定されたサーバーで SAP HANA システムレプリケーションモードを判定できませんでした。<br><br>**処置**: 詳細については、LifeKeeper ログファイルを確認してください。 |
| 136375 | EMERG | An NFS server exporting a critical shared file system for resource $tag is currently unavailable. Resource monitoring for $tag will be suspended until the issue is resolved. | **原因**: 重要な NFS 共有ファイルシステムが現在利用不可です。<br><br>**処置**: NFS サーバーが稼働しているこ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | と、および必要な NFS 関連サービスがすべて実行されていることを確認してください。 |
| 136376 | EMERG | WARNING: LifeKeeper resource $tag is designed for use in situations where SAP HANA System Replication (HSR) is disabled, but HSR was found to be enabled on server $me. Please ensure that the correct LifeKeeper resource type is being used for your current SAP HANA configuration. | **原因**: 指定した LifeKeeper リソースは、HANA システムレプリケーション（HSR）が無効になっている SAP HANA データベース環境を保護するように設計されているが、指定したサーバーでは HSR が有効になっています。<br><br>**処置**: SAP HANA の構成に応じて、正しい LifeKeeper リソースタイプが使用されていることを確認してください。HSR が無効になっている SAP HANA 構成から、HSR が有効になっている SAP HANA 構成に移行した場合は、対応する SAP HANA リソースを LifeKeeper で再作成する必要があります。 |
| 136377 | EMERG | SAP HANA database $instance corresponding to resource $tag is currently suspended on server $me due to actions performed outside of LifeKeeper. In order to resume the suspended database, execute the command 'su – $obj->{'sid_admin'} -c \"$obj->{'hana_util'} -sr_resumeSuspendedPrimary\"' on server $me. Once the suspended database has been resumed, LifeKeeper will automatically register the other servers in the cluster as secondary replication sites. Otherwise, please bring resource $tag in-service on the server where the database should be registered as primary master. Resource monitoring for $tag will be suspended until the issue is resolved. | **原因**: 指定したサーバー上の SAP HANA データベースインスタンスが、LifeKeeperの外部で行われた操作により停止されました。<br><br>**処置**: 対応する LifeKeeper リソースが現在 In-Service（Active）になっているサーバー上で一時停止中のデータベースを再開する場合は、メッセージに記載されているコマンドを実行してください。それ以外の場合は、目的のプライマリーレプリケーションサイトで LifeKeeper SAP HANA リソースを In-Service にしてください。 |
| 136450 | ERROR | Usage: $usage | **原因**: SAP HANA remove スクリプトに無効な引数が指定されています。<br><br>**処置**: 次の形式で適切な引数を指定してください： |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | <テンプレートタグ> <テンプレートID> |
| 136454 | ERROR | Error getting resource information for $tag. | 原因: 指定された SAP HANA リソースに関する情報を取得できませんでした。<br><br>処置: 指定されたタグを持つ SAP HANA リソースが指定されたサーバーに存在することを確認してください。 |
| 136456 | ERROR | Failed start of SAP Start Service for SAP HANA database $instance on server $me. | 原因: 指定された SAP HANA データベースの SAP Start Service を開始できませんでした。<br><br>処置: 詳細については、SAP Start Service トレースファイル（sapstartsrv.logなど）を確認してください。 |
| 136459 | EMERG | WARNING: LifeKeeper resource $tag is designed for use in situations where SAP HANA System Replication (HSR) is disabled, but HSR was found to be enabled on server $me. Please ensure that the correct LifeKeeper resource type is being used for your current SAP HANA configuration. | 原因: 指定した LifeKeeper リソースは、HANA システムレプリケーション（HSR）が無効になっている SAP HANA データベース環境を保護するように設計されていますが、指定したサーバーでは HSR が有効になっています。<br><br>処置: SAP HANA の構成に応じて、正しい LifeKeeper リソースタイプが使用されていることを確認してください。HSR が無効になっている SAP HANA 構成から、HSR が有効になっている SAP HANA 構成に移行した場合は、対応する SAP HANA リソースを LifeKeeper で再作成する必要があります。 |
| 136462 | ERROR | Failed to remove flag \"${hana::HANA_FLAG_LEAVE_DB_RUNNING}_$tag\" on server $me. This may cause subsequent remove actions for resource $tag on server $me to unintentionally fail to stop the database. | 原因: 指定したサーバーでLifeKeeperの!volatile!hana_leave_db_running_<tag>フラグを削除できない。<br><br>処置: 指定したサーバーでコマンド「/opt/LifeKeeper/bin/flg_remove -f '!volatile!h |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ana_leave_db_running_<tag>'」を実行して、フラグを手動で削除してください。フラグが存在する間は、指定したサーバー上のSAP HANAリソースのOut-of-service操作では、保護対象のデータベースインスタンスが実行されたままになります。 |
| 136525 | ERROR | $cmd::The name of the machine was not specified. | 原因: comm_up スクリプトが必須パラメーターなしで呼び出されました。<br><br>処置: LifeKeeper の内部エラーです。 |
| 136528 | ERROR | $cmd::HANA comm_up timed out, will not in-service AUTORES_ISP resources. | 原因: HANA comm_up スクリプトは、LifeKeeper の初期化 (LK_INITDONE) が完了するのを待ってから、AUTORES_ISP リソースを restore (in-service) します。 このエラーは、割り当てられた時間内に初期化が完了しなかったことを示しています。<br><br>処置: ログを確認して、初期化が時間内に終了しない原因を特定してください。追加の時間が必要な場合は、/etc/default/LifeKeeper の HANA_STARTUP_TIMEOUT を変更してください。 |
| 136550 | ERROR | Usage: $usage | 原因: HANAリソース回復スクリプトの引数が無効。<br><br>処置: 有効なHANA LifeKeeperリソースタグ名とリソースIDの両方を入力してください。<br>入力方法：recover -d <タグ> -n <id> |
| 136555 | ERROR | Error getting resource information for $tag on server $me. | 原因: 指定されたサーバー上の指定されたHANAリソースに関する情報を取得できなかった。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: サーバーがオンラインで、LifeKeeperが実行されており、HANAリソースが存在することを確認してください。 |
| 136556 | EMERG | The replication mode of the SAP HANA database $instance corresponding to resource $tag was modified outside of LifeKeeper and is no longer registered as primary master on server $me. Please bring the SAP HANA resource in-service on the server where the database should be registered as primary master. Resource monitoring for $tag will be suspended until the issue is resolved. | **原因**: 指定されたデータベースのSAP HANAシステムレプリケーションモードが、LifeKeeperの外部で変更されている。<br><br>**処置**: プライマリマスターとして登録する必要があるサーバーで、SAP HANAリソースをIn Serviceにしてください。 |
| 136558 | EMERG | LifeKeeper was unable to determine the SAP HANA System Replication mode for database $instance corresponding to resource $tag on server $me. Resource monitoring for $tag will be suspended until the issue is resolved. | **原因**: 指定されたサーバー上の指定されたデータベースに対し、SAP HANAシステムレプリケーションモードを決定でなかった。<br><br>**処置**: 詳細については、SAP HANAトレースファイルとLifeKeeperログファイルを確認してください。 |
| 136559 | ERROR | Resource $tag is no longer ISP on server $me. Exiting $cmd for $tag. | **原因**: 指定されたSAP HANAリソースが、指定されたサーバー上でISPではなくなった。<br><br>**処置**: 詳細については、LifeKeeperログファイルを確認してください。 |
| 136650 | ERROR | Usage: $usage | **原因**: SAP HANA hana_stop_all_dbs スクリプトに無効な引数が指定されている。<br><br>**処置**: 次の形式で適切な引数を指定してください：<br>hana_stop_all_dbs -t <タグ> |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 136654 | ERROR | Error getting resource information for $tag. | **原因**: 指定されたSAP HANAリソースに関する情報を取得できなかった。<br><br>**処置**: 指定されたタグを持つSAP HANAリソースが指定されたサーバーに存在することを確認してください。 |
| 136658 | ERROR | Failed start of SAP Start Service for SAP HANA database $x->{'instance'} on server $x->{'sys'}. Could not determine status of SAP HANA DB on $x->{'sys'}. | **原因**: 指定されたSAP HANAデータベースのSAP Start Serviceの開始に失敗した。<br><br>**処置**: 詳細については、SAP Start Serviceトレースファイル（sapstartsrv.logなど）を確認してください。 |
| 136661 | ERROR | Failed stop of SAP HANA database $x->{'instance'} on server $x->{'sys'}. | **原因**: 指定されたサーバー上の指定されたSAP HANAデータベースの停止に失敗した。<br><br>**処置**: 詳細については、SAP HANAトレースファイルとLifeKeeperログファイルを確認してください。 |
| 136673 | ERROR | Usage: $usage | **原因**: SAP HANAのhana_takeover_with_handshakeスクリプトで無効な引数が指定された。<br><br>**処置**: 次の形式で適切な引数を指定してください： -t <tag> [-s <target server>] [-b] |
| 136674 | ERROR | Unable to remove the \"${hana::HANA_FLAG_LEAVE_DB_RUNNING}_$tag\" flag on server $isp_node. This may cause subsequent remove actions for resource $tag on server $isp_node to unexpectedly fail to stop the database. | **原因**: 指定したサーバーでLifeKeeperの!volatile!hana_leave_db_running_<tag>フラグを削除できない。<br><br>**処置**: 指定したサーバーでコマンド「/opt/LifeKeeper/bin/flg_remove -f '!volatile!hana_leave_db_running_<tag>'」を実行して、フラグを手動で削除してくださ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | い。フラグが存在する間は、指定したサーバー上のSAP HANAリソースのOut-of-service操作では、保護対象のデータベースインスタンスが実行されたままになります。 |
| 136675 | ERROR | Script $cmd exited unexpectedly due to signal \"$sig\" on server $me. This may leave the $tag resource hierarchy as well as SAP HANA System Replication in an unexpected state. Please verify that the cluster resources are in the expected state. | 原因: hana_takeover_with_handshake スクリプトが、指定したサーバー上の指定したシグナルにより、予期せず終了した。<br><br>処置: SAP HANA クラスターリソースが期待通りの状態であることを確認してください。そうでない場合は、見つかった問題を修正し、目的のプライマリーサーバーで SAP HANA リソース階層を In-Service にしてください。 |
| 136677 | ERROR | Unable to find equivalent SAP HANA resource corresponding to $tag on server $target_node. | 原因: 指定したサーバー上の指定したリソースに対応する同等のSAP HANAリソースが見つからない。<br><br>処置: 指定したリソースタグが正しく、リソースが指定したターゲットサーバーに拡張されており、LifeKeeper がターゲットサーバー上で実行されて完全に初期化されていることを確認してください。 |
| 136678 | ERROR | Unable to obtain information about equivalent SAP HANA resource $tag on server $target_node. Verify that LifeKeeper is running and fully initialized. | 原因: 指定したサーバー上の指定したリソースに対応する同等のSAP HANAリソースが見つからない。<br><br>処置: 指定したリソースタグが正しく、リソースが指定したターゲットサーバーに拡張されており、LifeKeeper がターゲットサーバー上で実行されて完全に初期化されていることを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 136679 | ERROR | Resource $tag is not a SAP HANA resource. | **原因**: 指定したリソースがSAP HANA（データベース/hana）リソースではない。<br><br>**処置**: リソースタグが正しいことを確認してください。 |
| 136680 | ERROR | Resource $tag is designed for use in an environment where SAP HANA System Replication is disabled. Takeover with handshake cannot be performed for this resource type. Please use the standard \"In Service…\" command instead. | **原因**:「Takeover with Handshake」に関連する機能は、SAP HANAシステムレプリケーションが有効な環境でのみ使用できる。<br><br>**処置**: 標準の「In Service…」オプションを使用して、SAP HANAリソースをIn-Serviceにしてください。 |
| 136681 | ERROR | SAP HANA resource $tag is not currently in-service on any server in the cluster. The resource must be in-service and SAP HANA System Replication must be in-sync before performing a takeover with handshake. | **原因**:「Takeover with Handshake」の試行中、指定したSAP HANAリソースがクラスター内のどのサーバーでもIn-Serviceになっていない。<br><br>**処置**: 目的のプライマリーサーバーでSAP HANAリソースをIn-Serviceにしてください。 |
| 136682 | ERROR | SAP HANA resource $tag is currently in-service on multiple servers: $isp_node_list. The resource must be in-service on only one server and SAP HANA System Replication must be in-sync before performing a takeover with handshake. | **原因**:「Takeover with Handshake」の試行中、指定したSAP HANAリソースがクラスター内の複数のサーバーでIn-Serviceになっている。<br><br>**処置**: プライマリーマスターとして登録する予定のサーバーを除くすべてのサーバーで、リソースをOut-of-serviceにしてください。 |
| 136684 | ERROR | Unable to create internal SAP HANA object for resource $tag on server $target_node. Verify that all necessary file systems are mounted and that LifeKeeper is running and fully initialized on $target_node. | **原因**: 指定したサーバー上の指定したインスタンスを表す内部HANAオブジェクトを作成できない。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: データベースインスタンスが正しくインストールされ、必要なファイルシステムがすべてマウントされていることを確認してください。 |
| 136685 | ERROR | Takeover with handshake is only supported in SAP HANA versions 2.0 SPS04 and greater. The SAP HANA software must be upgraded in order to use this feature. Please use the standard \"In Service…\" command instead. | **原因**: 基礎となるSAP HANAデータベースのバージョンが2.0 SPS04未満の場合、「Takeover with Handshake」機能は使用できない。<br><br>**処置**:「Takeover with Handshake」機能を使用するには、SAP HANA 2.0 SPS04以降にアップグレードしてください。 |
| 136686 | ERROR | Takeover with handshake cannot be performed for database $target_obj->{'instance'} on server $target_node because the database is not currently running and registered as primary on any other server in the cluster. | **原因**:「Takeover with Handshake」の試行中、指定したSAP HANAデータベースインスタンスがクラスター内のどのサーバーでも実行されておらず、プライマリーマスターとして登録されていない。<br><br>**処置**: 目的のプライマリーサーバーで対応するSAP HANAリソースをIn-Serviceにしてください。 |
| 136687 | ERROR | SAP HANA database $target_obj->{'instance'} is running and registered as primary on more than one server in the cluster. Please resolve this situation and reattempt the takeover. | **原因**:「Takeover with Handshake」の試行中、指定したSAP HANAデータベースインスタンスが複数のサーバーでプライマリーとして実行および登録されている。<br><br>**処置**: データベースインスタンスがすでに実行されており、対象のプライマリーサーバーにプライマリーとして登録されている場合は、対応するLifeKeeperリソースをそのサーバーで In-Service にしてください。そうでない場合は、LifeKeeper で現在 In-Service になっているサーバーを除くすべてのサーバーでデータベースインスタンスを停止し、LifeKeeperがシステムレプリケーションを再開でき |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | るようにしてから、テイクオーバーを再試行してください。 |
| 136689 | ERROR | Unable to create internal SAP HANA object for resource $tag on server $isp_node. Verify that all necessary file systems are mounted and that LifeKeeper is running and fully initialized on $isp_node. | **原因**: 指定したサーバー上の指定したインスタンスを表す内部HANAオブジェクトを作成できない。<br><br>**処置**: データベースインスタンスが正しくインストールされ、必要なファイルシステムがすべてマウントされていることを確認してください。 |
| 136690 | ERROR | Unable to set the ${hana::HANA_FLAG_LEAVE_DB_RUNNING}_$tag flag on server $isp_node. Aborting takeover with handshake attempt for database $target_obj->{'instance'} on server $target_node. | **原因**:「Takeover with Handshake」の試行中に、指定したサーバーで、LifeKeeperフラグ「!volatile!hana_leave_db_running_<tag>」を設定できなかった。<br><br>**処置**: 詳細については、LifeKeeper のログファイルを確認してください。見つかった問題を修正し、テイクオーバーを再試行してください。 |
| 136692 | ERROR | Takeover with handshake for database $target_obj->{'instance'} failed on server $target_node. | **原因**: 指定したターゲットサーバー上の指定したSAP HANAデータベースインスタンスに対する「Takeover with Handshake」の試行が失敗した。<br><br>**処置**: /etc/default/LifeKeeper で HANA_HANDSHAKE_TAKEOVER_FAILBACK=true が指定されている場合、SAP HANAリソース階層は以前のデータベースホストサーバーで自動的にIn-Serviceに戻ります。そうでない場合は、SAP HANAリソース階層を対象のプライマリーサーバーで手動でIn-Serviceにする必要があります。 |
| 136695 | ERROR | Resource $tag does not exist on server $me. | **原因**: 指定したリソースが指定したサーバーに存在しない。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: リソースタグが正しいことを確認してください。 |
| 136696 | ERROR | Unable to verify the status of resource $tag on server $target_node. Assuming that it is not in-service. | **原因**:「Takeover with Handshake」の成否の確認中に、指定したサーバー上の指定したリソースのステータスを判別できなかった。<br><br>**処置**: 指定したサーバー上でLifeKeeperが実行され、完全に初期化されていること、およびローカルサーバーとターゲットサーバー間のコミュニケーションパスがアクティブであることを確認してください。/etc/default/LifeKeeper で HANA_HANDSHAKE_TAKEOVER_FAILBACK=true が指定されている場合、SAP HANAリソース階層は自動的に以前のデータベースホストサーバーでIn-Serviceになります。そうでない場合は、SAP HANAリソース階層を対象のプライマリーサーバー上で手動でIn-Serviceにする必要があります。 |
| 136697 | ERROR | Resource $res was not successfully brought in-service on server $target_node. | **原因**:「Takeover with Handshake」の試行中に、指定したリソースが指定したサーバーでIn-Serviceにならなかった。<br><br>**処置**: etc/default/LifeKeeper で HANA_HANDSHAKE_TAKEOVER_FAILBACK=true に設定されている場合、SAP HANA リソース階層は自動的に以前のデータベースホストサーバーでサービスを In-Service にしてください。そうでない場合は、SAP HANA リソース階層を対象のプライマリーサーバーで手動で In-Service にする必要があります。 |
| 136698 | ERROR | LifeKeeper is not running or is not fully initialized on server $me. | **原因**:「Takeover with Handshake」の試行中に、LifeKeeperが指定したサーバーで実行されていないか、完全に初期化さ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | れていない。<br><br>**処置**: LifeKeeperを /opt/LifeKeeper/bin/lkstart で起動するか、完全に初期化してから、テイクオーバーを再試行してください。 |
| 136699 | ERROR | Unknown server $target_node. | **原因**: 指定したサーバーのホスト名が認識されない。<br><br>**処置**: サーバーのホスト名が正しく、ローカルサーバーとターゲットサーバーの間でコミュニケーションパスが作成されていることを確認してください。 |
| 136700 | ERROR | Usage: $usage | **原因**: SAP HANA remoteregisterdbスクリプトに無効な引数が与えられました。<br><br>**処置**: 次の形式で適切な引数を指定してください: remoteregisterdb -d <tag> -n <id> |
| 136705 | ERROR | Unable to obtain information about resource $tag on server $me. Exiting $cmd for $tag. | **原因**: 指定されたサーバー上の指定されたSAP HANAリソースに関する情報を取得できませんでした。<br><br>**処置**: 指定されたリソースタグが正しいこと、リソースが指定されたサーバーに存在すること、必要なすべてのファイルシステムがマウントされていること、LifeKeeperが実行中で、指定されたサーバーで完全に初期化されていることを確認してください。 |
| 136706 | ERROR | Resource $tag is no longer ISP on server $me. Exiting $cmd for $tag. | **原因**: SAP HANAリソースが特定のサーバーでIn-Service（ISP）になっていないため、SAP HANA remoteregisterdbスクリプトが終了しています。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 対応は必要ありません。 |
| 136707 | ERROR | The $cmd event is intended for use only in environments in which SAP HANA System Replication is enabled. Exiting $cmd for $tag. | **原因**: SAP HANA remoteregisterdbイベントは、指定されたSAP HANAリソースによって保護されているデータベースに対してSAP HANAシステムレプリケーションが無効になっていることを検出しました。<br><br>**処置**: remoteregisterdbスクリプトが実行されていたサーバーでSAP HANAシステムレプリケーションが有効になっていることを確認してください。必要に応じて、SAP HANA管理ユーザーとしてコマンド 'hdbnsutil -sr_enable —name=<Site Name>' を実行することにより、システムレプリケーションを有効にすることができます。 |
| 136708 | ERROR | Error getting resource information for $tag on server $me. Exiting $cmd for $tag. | **原因**: 指定されたサーバーの指定されたSAP HANAリソースに関する情報を取得できませんでした。<br><br>**処置**: 指定されたタグを持つSAP HANAリソースが指定されたサーバーにあることを確認してください。 |
| 136709 | EMERG | LifeKeeper was unable to determine the SAP HANA System Replication mode for database $instance corresponding to resource $tag on server $me. Resource monitoring for $tag will be suspended until the issue is resolved. | **原因**: 指定されたサーバーの指定されたデータベースに対して、SAP HANAシステムレプリケーションモードを検出できませんでした。<br><br>**処置**:詳細については、SAP HANAトレースファイルとLifeKeeperログファイルを確認してください。 |
| 136710 | EMERG | The replication mode of the SAP HANA database $instance corresponding to resource $tag was modified outside of LifeKeeper and is no longer registered as primary master on server $me. Please bring the S | **原因**: 指定されたデータベースのSAP HANAシステムレプリケーションモードがLifeKeeperの外部で変更されました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | AP HANA resource in-service on the server where the database should be registered as primary master. Resource monitoring for $tag will be suspended until the issue is resolved. | **処置**:プライマリーマスターとして登録する必要のあるサーバーで、SAP HANAリソースをIn-Serviceにしてください。 |
| 136711 | EMERG | SAP HANA database $instance corresponding to resource $tag is currently suspended on server $me due to actions performed outside of LifeKeeper. Please take the SAP HANA resource out of service on server $me and bring it in-service on the server where the database should be registered as primary master. Bringing resource $tag back in-service on $me will resume the suspended database. Resource monitoring for $tag will be suspended until the issue is resolved. | **原因**: 指定されたたサーバー上のSAP HANAデータベースインスタンスが、LifeKeeperの外部で行われた操作により一時停止されています。<br><br>**処置**: 対応するLifeKeeperリソースが現在In-Service（Active）になっているサーバー上で一時停止中のデータベースを再開する場合は、メッセージに記載されているコマンドを実行してください。それ以外の場合は、目的のプライマリーレプリケーションサイトでLifeKeeper SAP HANAリソースをIn-Serviceにしてください。 |
| 136732 | ERROR | There is no LifeKeeper protected resource with tag $tag on system $me. | **原因**: HANA canfailover スクリプトは、HANA リソースではないタグで呼び出されました。<br><br>**処置**: 処置は必要ありません。 |
| 137000 | ERROR | PowerShell is not installed. | |
| 137001 | ERROR | PowerCLI is not installed. | |
| 137002 | ERROR | A valid network interface was not found. | |
| 137005 | ERROR | Failed to attach VMDK. | |
| 137010 | ERROR | Failed to detach VMDK. | |
| 137020 | ERROR | Failed to execute VMDK status checker daemon. | |
| 137030 | ERROR | Disk not specified. | |
| 137031 | ERROR | Cannot get disk uuid for $Disk. Please check your ESXi settings. | |
| 137032 | ERROR | PowerCLI failed. %s | |
| 137034 | ERROR | Cannot bring VMDK resource \"%s\" in se | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | rvice on server \"%s\". | |
| 137037 | ERROR | This system is not a VMware guest. | |
| 137050 | ERROR | Failed to connect to ESXi server $addr. | |
| 137051 | ERROR | There is no ESXi server connected. | |
| 137055 | ERROR | Cannot determine ESXi VM ID because multiple network interfaces were found wit h the MAC address $MAC_ADDR. | |
| 137057 | ERROR | Usable SCSI controller not found. | |
| 137058 | ERROR | Cannot find VMDK with ID $UUID. | |
| 137059 | ERROR | This guest has snapshots present. | |
| 137060 | ERROR | The VMDK with ID $UUID cannot be attac hed to this guest. | |
| 137061 | ERROR | The virtual storage controller has an inco mpatible sharing mode configured. | |
| 137068 | ERROR | The VMDK detection failed. Retry count e xceeded. | |
| 137070 | ERROR | Connect failed. | |
| 137071 | ERROR | Get-LocalVM failed. | |
| 137072 | ERROR | The VMDK has been detached remotely. This server has lost ownership. | |
| 137075 | ERROR | Cannot find virtual SCSI controller $CON TROLLER. | |
| 137076 | ERROR | The virtual storage controller has an inco mpatible sharing mode configured. | |
| 137077 | ERROR | Cannot find VM with MAC address $MA C_ADDR. | |
| 137078 | ERROR | Cannot find VM with MAC address $MA C_ADDR. | |
| 137101 | WARN | Partition information not defined for %s on %s. Retry. | |
| 137102 | ERROR | Partition information not defined for %s on %s. | |
| 137105 | ERROR | Device not specified. | |
| 137106 | ERROR | Cannot get device uuid for $Device. Pleas | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | e check your ESXi settings. | |
| 137107 | ERROR | %s is not shareable with any machine. | |
| 137111 | ERROR | Failed to create dependency \"%s\"-\"%s\" on machine \"%s\". | |
| 137112 | ERROR | Cannot bring VMDKP resource \"%s\" in service on server \"%s\". | |
| 138003 | FATAL | $msg | **原因**: このプロセスのスタックトレースを記録します。<br><br>**処置**: なし |
| 139000 | ERROR | Cannot find the "oci" command in directories of the PATH. Please confirm that it is installed and the PATH is set correctly. | **原因**: oci コマンドの PATH が取得できません。<br><br>**処置**: oci コマンドがインストールされていること、oci コマンドのインストールディレクトリが PATH に追加されていることを確認してください。 |
| 139012 | ERROR | \"oci\" command(request-id:%s) failed with code %s, message \"%s\" and status %s | **原因**: oci コマンドの実行に失敗しました。<br><br>**処置**: ステータスコードやメッセージ内容を元に oci コマンドが正常終了するように設定してください。 |
| 139013 | ERROR | Failed to assign $ip, error:\"$result\" | **原因**: $ip の割り当てに失敗しました。<br><br>**処置**: $result の内容を元にエラーの原因を解決してください。 |
| 139022 | ERROR | \"oci\" command(request-id:%s) failed with code %s, message \"%s\" and status %s | **原因**: oci コマンドの実行に失敗しました。<br><br>**処置**: ステータスコードやメッセージ内容を元に oci コマンドが正常終了する |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ように設定してください。 |
| 139023 | ERROR | Failed to unassign $ip, error:\"$result\" | 原因: $ip の割り当て解除に失敗しました。<br><br>処置: $result の内容を元にエラーの原因を解決してください。remove 処理が実行されるより前に $ip の割り当てが解除されていた場合は、リソースの状態が OSU であることと、OCI コンソール上で $ip が対象の VNIC に割り合てられていないことを確認してください。 |
| 139032 | ERROR | \"oci\" command(request-id:%s) failed with code %s, message \"%s\" and status %s | 原因: oci コマンドの実行に失敗しました。<br><br>処置: ステータスコードやメッセージ内容を元に oci コマンドが正常終了するように設定してください。 |
| 139033 | ERROR | Failed to \"$ocicmdstr\", unknown error:\"$ip_list\" | 原因: コード 139032 以外の原因で oci コマンドが異常終了しました。<br><br>処置: unknown error:"$ip_list" の内容を元に対応してください。 |
| 139034 | ERROR | There is no secondary IPs on $device. | 原因: $device にセカンダリIPアドレスが割り当てられていません。<br><br>処置: ローカルリカバリーが有効の場合は、ローカルリカバリーが実行されたこと、それによって障害から復旧したことを確認してください。 |
| 139035 | ERROR | $ip is not assigned to $device. | 原因: $ip が $device に割り当てられていません。<br><br>処置: ローカルリカバリーが有効の場合 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | は、ローカルリカバリーが実行されたこと、それによって障害から復旧したことを確認してください。 |
| 139042 | ERROR | \"oci\" command(request-id:%s) failed with code %s, message \"%s\" and status %s | 原因: oci コマンドの実行に失敗しました。<br><br>処置: ステータスコードやメッセージ内容を元に oci コマンドが正常終了するように設定してください。 |
| 139043 | ERROR | Failed to assign $ip, error:\"$result\" | 原因: $ip の割り当てに失敗しました。<br><br>処置: $result の内容を元にエラーの原因を解決してください。 |
| 139060 | ERROR | $cmd is invalid. | 原因: $cmd が不正です。<br><br>処置: お問い合わせください。 |
| 139061 | ERROR | OCIVIP does not support IPv6. | 原因: OCIVIP リソースは IPv6 での利用をサポートしません。<br><br>処置: IPv4 をご利用ください。 |
| 139062 | ERROR | $cmd is invalid. | 原因: $cmd が不正です。<br><br>処置: お問い合わせください。 |
| 139063 | ERROR | IPv$ipversion is unknown version. | 原因: IP アドレスのバージョンが不正です。<br><br>処置: IPv4 をご利用ください。 |
| 139070 | ERROR | Failed to access the $IMDS_URL with the curl command, status code is $status_code. | 原因: $IMDS_UDL への curl が失敗しま |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | した。<br><br>**処置**: $IMDS_URL への curl が正常終了することを確認してください。 |
| 139071 | ERROR | Failed to decode JSON, error:\"$e\". | **原因**: JSON のデコードに失敗しました。<br><br>**処置**: Instance Meta Data Serviceから取得した結果が JSON 形式であることを確認してください。 |
| 139073 | ERROR | Cannot find the vnicId of $device. | **原因**: $device に対応する vnicId が取得できませんでした。<br><br>**処置**: Instance Meta Data Service の JSON レコードに vnicId が存在するかどうか確認してください。 |
| 139074 | ERROR | Cannot find the subnetcidr of $device. | **原因**: $device に対応する subnetCidrBlock が取得できませんでした。<br><br>**処置**: Instance Meta Data Service の JSON レコードに subnetcidr が存在するかどうか確認してください。 |
| 139075 | ERROR | Cannot find the vRouterIp of $device. | **原因**: $device に対応する virtualRouterIp が取得できませんでした。<br><br>**処置**: Instance Meta Data Service の JSON レコードに virtualRouterIp が存在するかどうか確認してください。 |
| 139080 | ERROR | \"oci\" command(request-id:%s) failed with code %s, message \"%s\" and status %s | **原因**: `oci` コマンドの実行に失敗しました。<br><br>**処置**: ステータスコードやメッセージ内容を元に `oci` コマンドが正常終了する |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ように設定してください。 |
| 139081 | ERROR | Failed to get subnet id, error:\"$subnetid\" | 原因: subnet id の取得に失敗しました。<br><br>処置: 以下のコマンドで正しく情報を取得できるかどうか確認してください。$vnicid にはノードに割り当てられている各 VNIC の VNIC-ID(OCID) が該当します。<br>`oci network vnic get --vnic-id $vnicid --raw-output --query 'data."subnet-id"'` |
| 139100 | ERROR | IP address is not specified. | 原因: IP アドレスが指定されていません。<br><br>処置: IP アドレスが指定されていることを確認してください。 |
| 139101 | ERROR | Device is not specified. | 原因: デバイスが指定されていません。<br><br>処置: デバイスが指定されていることを確認してください。 |
| 139105 | ERROR | Cannot bring OCIVIP resource $Tag in service on server $SysName. | 原因: $SysName 上の OCIVIP リソース $Tag の restore に失敗しました。<br><br>処置: restore 処理に関するログを参照の上ご対応ください。 |
| 139111 | ERROR | IP-$ip already exists, and device and net mask are inconsistent with those of $ocivipTag. | 原因: OCIVIP リソースの IP アドレスに対応する IP リソースが存在していますが、デバイス情報やネットマスクの情報に矛盾があります。<br><br>処置: 既存の IP リソースを OCIVIP リソースと同一の設定になるように設定を変更し、 OCIVIP リソースと依存関係を作 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 成してください。 |
| 139112 | ERROR | Cannot bring IP resource $iptag in service on server $SysName. | **原因**: $SysName 上の OCIVIP リソースに対応する IP リソース $iptag の restore に失敗しました。<br><br>**処置**: IPリソースの設定を見直し、 IP リソースの restore が成功することを確認してください。そのうえで OCIVIP リソースと依存関係を作成してください。 |
| 139200 | ERROR | A resource with the specified tag name "%s" already exists on the target machine "%s". | **原因**: 拡張先のノードに、指定したタグ名のリソースが既に存在します。<br><br>**処置**: 別のタグ名をご利用いただくか、拡張先ノードの対象タグ名のリソースを削除してください。 |
| 139201 | ERROR | Template resource "%s" does not exist on the server "%s". | **原因**: 拡張元として指定したリソースがノード上に存在しません。<br><br>**処置**: タグ名を正しく指定してください。 |
| 139250 | ERROR | Failed to valuenetOnOCI, error: \"$stderr_log\". | **原因**: `valuentOnOCI` に失敗しました。<br><br>**処置**: $stderr_log の箇所に実際のエラー内容が出力されています。その内容を元にご対応ください。 |
| 140001 | FATAL | cmd:$tag:Insufficient input parameters. \n%s | |
| 140002 | FATAL | cmd:$tag:END failed create of %s on system %s with signal $sig. | |
| 140005 | FATAL | cmd:$tag:Resource already exists on machine %s | |
| 140006 | FATAL | cmd:$tag:ins_create failed on machine %s | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 140007 | FATAL | cmd:$tag:Failed restore of tag %s on system %s | |
| 140020 | FATAL | cmd::Insufficient input parameters. \n%s | |
| 140021 | ERROR | cmd:$template_tag:Template resource \"$template_tag\" on server \"$template_sys\" does not exist | |
| 140030 | FATAL | cmd:$tag:Insufficient input parameters. \n%s | |
| 140031 | FATAL | cmd:$tag:END failed extend of "%s" due to a \"%s\" signal. | |
| 140033 | ERROR | cmd:$template_tag:Template resource \"$template_tag\" on server \"$template_sys\" does not exist | |
| 140036 | FATAL | cmd:$tag:Resource already exists on machine %s | |
| 140037 | FATAL | cmd:$tag:ins_create failed on machine %s | |
| 140050 | FATAL | cmd::Port number cannot be empty. Specify the TCP/IP port number (any value from 1024 to 65535 | |
| 140051 | FATAL | cmd::Specify the TCP/IP port number (any value from 1024 to 65535 | |
| 140060 | FATAL | cmd::$usage | |
| 140061 | FATAL | cmd::resource \"$tag\" on server \"$lbhc::cmd\" does not exist | |
| 140062 | FATAL | cmd::LK::ins_setinfo failed | |
| 140070 | FATAL | $usage | |
| 140071 | FATAL | Port number is not valid. | |
| 140073 | FATAL | Error getting resource information for $tag on server $lbhc::me. | |
| 140200 | FATAL | Unable to create LBHC object ($sys, $tag). | |
| 140202 | ERROR | Daemon already started but the pidfile does not match. | 原因: restore 時にデーモンがすでに実行されているが、pidfile が一致しないことを示します。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 実行中のデーモンを停止してから、 restore を試行してください。 |
| 140204 | FATAL | Could not start the daemon by \"$cmdline\". | |
| 140207 | ERROR | Error deleting resource \"$otherTag\" on \"$otherSys\" with return code of \"$err\". | |
| 140208 | WARN | Resource out of service | **原因**: quickCheck は実行されていますが、リソースは ISP (In-Service Protected) ではありませんでした。 |
| 140209 | ERROR | Daemon not running. | **原因**: quickCheckの確認で、デーモンが停止していました。<br><br>**処置**: ローカルリカバリーを試みます。 |
| 140210 | ERROR | Daemon is running but the pidfile does not match. | **原因**: quickCheck の確認で、デーモンは起動していましたが pidfile が一致しませんでした。<br><br>**処置**: ローカルリカバリーを試みます。 |
| 140211 | ERROR | flg $id-hc-hg is on | **原因**: $id-hc-hg フラグは、quickCheck によって検出されました。<br><br>**処置**: フェイルオーバーを試みます。 |
| 140250 | FATAL | Socket open failed. Check if the port \"$port\" is in use by ‘ss -anp’. | **原因**: ソケットをオープンできませんでした。<br><br>**処置**: ‘ss -anp’ でポートが使用されているかどうかを確認し、使用しているプロセスを強制終了してください。 |
| 140251 | ERROR | Health probe not received. Check that the probe from the load balancer is set correctly, including the firewall. | **原因**: ロードバランサーからのプローブ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | が $HC_TIMEOUT 秒以内に受信されませんでした。<br><br>**処置**: ファイアーウォールを含め、ロードバランサーからのプローブが正しく構成されていることを確認してください。 |
| 140252 | FATAL | Could not create the flag by \"$cmdline\". | |
| 140253 | ERROR | Failed to accept the health probe:$reason | |
| 140254 | FATAL | $errmsg | |
| 140256 | FATAL | Failed to save pid into $lbhc::LKROOT/config/$id.pid. | |
| 140257 | FATAL | Resource already deleted | |
| 140258 | FATAL | kill $pid failed with ret=$ret, kill the process \"$pid\" manually. | |
| 140320 | FATAL | $usage | |
| 140321 | FATAL | Error getting resource information for $tag on server $lbhc::me. | |
| 140330 | FATAL | $usage | |
| 140331 | FATAL | Error getting resource information for $tag on server $lbhc::me. | |
| 140340 | FATAL | $usage | |
| 140341 | FATAL | Error getting resource information for $tag on server $lbhc::me. | |
| 140350 | FATAL | $usage | |
| 140351 | FATAL | Error getting resource information for $tag on server $lbhc::me. | |
| 140360 | FATAL | $usage_l | |
| 140361 | FATAL | invalid ID \"$id\" | |
| 140362 | FATAL | $usage_w | |
| 140363 | FATAL | Error getting resource information for $tag on server $lbhc::me. | |

# 8.1.1. DataKeeper メッセージカタログ

**Control F** を使用して、各カタログの特定のエラーコードを検索します。エラーコードの検索には、画面右上の検索ボックスをご利用ください。

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 104002 | FATAL | $msg | **原因**: このメッセージは内部ソフトウェアエラーを示します。<br><br>**処置**: スタックトレースにエラーの原因が示されます。 |
| 104003 | FATAL | $self->Val('Tag') . " is not an SDR resource"); | **原因**: 非データレプリケーションリソースに対してデータレプリケーションアクションが実行されました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 104010 | ERROR | $self->{'md'}: bitmap merge failed, $action | **原因**: ビットマップのマージ操作に失敗しました。<br><br>**処置**: ターゲットサーバーにミラーや保護されたファイルシステムがマウントされている可能性があります。または、ビットマップファイルがターゲットに見つからない可能性があります。ターゲットサーバーを確認してください。 |
| 104022 | ERROR | $argv[1]: mdadm failed ($ret) | **原因**:「mdadm」コマンドによるミラーへのデバイスの追加に失敗しました。<br><br>**処置**: これは通常、一時的な状態です。 |
| 104023 | ERROR | $_); | **原因**: メッセージに「mdadm」コマンドの結果が含まれています。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 104025 | ERROR | failed to spawn monitor | **原因**: システムは、「mdadm -F」監視プロセスの開始に失敗しました。これは、通常の状態では発生しません。<br><br>**処置**: システムを再起動し、競合の可能性があるものが解決されていることを確認してください。 |
| 104026 | ERROR | cannot create $md | **原因**: ミラーデバイスを作成できませんした。<br><br>**処置**: デバイスがまだ使用中ではないこと、およびミラー作成用の他の全パラメータが正しいことを確認してください。 |
| 104027 | ERROR | $_ | **原因**: このメッセージに「mdadm」コマンドの結果が含まれています。 |
| 104035 | ERROR | Too many failures. Aborting resync of $md | **原因**: 正常範囲とは思われないほど長い時間、デバイスがビジー状態です。<br><br>**処置**: システムを再起動し、デバイスがビジーではなくなっていることを確認してください。 |
| 104036 | ERROR | Failed to start nbd-server on $target (error $port) | **原因**: ターゲットサーバーで nbd-server プロセスを開始できませんでした。<br><br>**処置**: ターゲットディスクデバイスが使用可能であること、およびデバイス ID が変更されていないことを確認してください。 |
| 104037 | ERROR | Failed to start compression (error $port) | **原因**: システムは、「バランス」トンネルプロセスを開始できませんでした。または、ネットワークに問題が発生しました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: ネットワークが正常に実行されていること、および 10000 ～ 10512 の範囲内にある TCP ポートがオープンで未使用の状態にあることを確認してください。ソフトウェアがすべてのシステムに正常にインストールされていることを確認してください。 |
| 104038 | ERROR | Failed to start nbd-client on $source (error $ret) | **原因**: nbd-client プロセスをソースサーバーで開始できませんでした。<br><br>**処置**: レポートされている errno 値を調べ、レポートされている問題を解決してください。たとえば、errno 値 110 は「Connection timed out」(接続のタイムアウト) を示します。これは、ネットワークかファイアーウォールに問題がある可能性があります。 |
| 104039 | ERROR | Failed to add $nbd to $md on $source | **原因**: これは通常、一時的な状態です。<br><br>**処置**: このエラーが続く場合、システムを再起動し、競合の可能性があるものを解決してください。 |
| 104045 | ERROR | failed to stop $self->{'md'} | **原因**: ミラーデバイスを停止できませんした。<br><br>**処置**: デバイスがビジーでないこと、またはマウントされていないことを確認してください。「mdadm —stop」を手動で実行し、デバイスを停止してください。 |
| 104048 | WARN | failed to kill $proc, pid $pid | **原因**: プロセスをシグナル化できませんでした。これは、プロセスが終了していることを示している可能性があります。<br><br>**処置**: 問題のプロセスが実行されていないことを確認し、実行されている場合 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | は、システムを再起動して、終了できないプロセスを消去してください。 |
| 104050 | ERROR | Setting $name on $dest failed: $ret. Please try again. | **原因**: システムは「mirrorinfo」ファイル設定を設定できませんでした。<br><br>**処置**: ネットワークとシステムを確認し、ミラー設定操作を再試行してください。 |
| 104052 | FATAL | Specified existing mount point "%s" is not mounted | **原因**: マウントポイントがアンマウント状態になりました。<br><br>**処置**: マウントポイントがマウントされていることを確認し、操作を再試行してください。 |
| 104055 | ERROR | Failed to set up temporary $type access to data for $self->{'tag'}. Error: $ret | **原因**: ターゲットサーバーでファイルシステムまたはデバイスを使用できません。ミラーされたデータは、ミラーが一時停止してから再開されるまでターゲットサーバーでは使用できません。<br><br>**処置**: ターゲットサーバーを再起動し、競合の可能性があるものを解決してください。 |
| 104057 | ERROR | Failed to undo temporary access for $self->{'tag'} on $self->{'sys'}. Error: $ret. Please verify that $fsid is not mounted on server $self->{'sys'}. | **原因**: ターゲットサーバーでファイルシステムをアンマウントできませんでした。<br><br>**処置**: ファイルシステムおよびデバイスがターゲットサーバーでビジーでないことを確認してください。ターゲットサーバーを再起動し、競合の可能性があるものを解決してください。 |
| 104062 | FATAL | Cannot find a device with unique ID "%s" | **原因**: ターゲットディスクを識別できま |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | せんでした。<br><br>**処置**: 適切なストレージリカバリーキットがターゲットサーバーにインストールされていることを確認してください。ターゲットディスクのデバイス ID が変更されていないことを確認してください。 |
| 104066 | FATAL | Cannot get the hardware ID of device "%s" | **原因**: ターゲットディスクデバイスの一意の ID が見つかりませんでした。<br><br>**処置**: 適切なストレージリカバリーキットがターゲットサーバーにインストールされていることを確認してください。ターゲットディスクのデバイス ID が変更されていないことを確認してください。 |
| 104067 | FATAL | Asynchronous writes cannot be enabled without a bitmap file | **原因**: 無効なパラメータでミラーを作成しようとしました。<br><br>**処置**: ビットマップファイルパラメータを指定するか、同期書き込みを指定する必要があります。 |
| 104068 | FATAL | Failed to extend dependent resource %s to system %s. Error %s | **原因**: 階層の拡張操作に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 104070 | FATAL | Unable to extend the mirror "%s" to system "%s" | **原因**: 階層の拡張操作に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 104071 | ERROR | Failed to restore target device resources on $target->{'sys'} : $err | **原因**: ターゲットサーバーで in service 操作に失敗しました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 104074 | FATAL | Cannot get the hardware ID of device "%s" | **原因**: ミラーに使用しようとしている基本のディスクデバイスを認識するストレージリカバリーキットがありません。<br><br>**処置**: 適切なストレージリカバリーキットがインストールされていることを確認してください。必要に応じ、/opt/LifeKeeper/subsys/scsi/resources/DEVNAME/device_pattern ファイルにデバイス名を入れてください。 |
| 104081 | FATAL | Cannot make the %s filesystem on "%s" (%d): | **原因**:「mkfs」コマンドが失敗しました。<br><br>**処置**:ディスクデバイスが書き込み可能で、エラーがないこと、および選択したファイルシステムのファイルシステムツールがインストールされていることを確認してください。 |
| 104082 | FATAL | %s | **原因**: このメッセージに「mkfs」コマンドの結果が含まれています。 |
| 104083 | FATAL | Cannot create filesys hierarchy "%s" | **原因**: リソースの作成に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 104086 | ERROR | The "%s_data_corrupt" flag is set in "%s/subsys/scsi/resources/netraid/" on system "%s". To avoid data corruption, LifeKeeper will not restore the resource. | **原因**: 予期しないデータ破損を防ぐための警告データとして data corrupt フラグファイルが設定されています。ファイルが削除されるまで、このサーバー上のミラーをリストアできません。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 問題のサーバーでデータが有効な場合、以下のいずれかを実行できます。1) ファイルを削除してミラーをリストアします。2) LifeKeeper GUI または「mirror_action force」コマンドでミラーを強制的にオンラインにします。 |
| 104092 | ERROR | Mirror target resource movement to system %s : status %s | **原因**: 階層のスイッチオーバー操作に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 104099 | ERROR | Unable to unextend the mirror for resource "%s" from system "%s" | **原因**: 階層の拡張解除に失敗しました。<br><br>**処置**: ターゲットサーバーを再起動し、競合の可能性があるものを解決して、操作を再試行します。 |
| 104106 | ERROR | remote 'bitmap -m' command failed on $target->{'sys'}: $ranges | **原因**: ターゲットサーバーでビットマップマージコマンドが失敗しました。これには、次の 2 つの原因が考えられます。1) ビットマップファイルが見つからないか、破損している。2) ミラー（md）デバイスがターゲットでアクティブになっている。<br><br>**処置**: ミラーおよび保護されているファイルシステムがターゲットでアクティブになっていることを確認してください。ターゲットのビットマップファイルが見つからない場合、ミラーを一時停止して再開し、ビットマップファイルを再作成してください。 |
| 104107 | ERROR | Asynchronous writes cannot be enabled without a bitmap file | **原因**: ミラー作成操作で無効なパラメータが指定されました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 104108 | ERROR | Local Partition not available | 原因: ミラー作成操作で無効なパラメータが指定されました。 |
| 104109 | ERROR | Cannot get the hardware ID of device "%s" | 原因: ディスクデバイスの一意の ID を特定できませんでした。<br><br>処置: 適切なストレージリカバリーキットがサーバーにインストールされていることを確認してください。ディスクのデバイス ID が変更されていないことを確認してください。 |
| 104111 | FATAL | Insufficient input parameters for "%s" creation | 原因: ミラー作成操作で無効なパラメータが指定されました。 |
| 104112 | FATAL | Insufficient input parameters for "%s" creation | 原因: ミラー作成操作で無効なパラメータが指定されました。 |
| 104113 | FATAL | Insufficient input parameters for "%s" creation | 原因: ミラー作成操作で無効なパラメータが指定されました。 |
| 104114 | FATAL | Insufficient input parameters for "%s" creation | 原因: ミラー作成操作で無効なパラメータが指定されました。 |
| 104115 | FATAL | Insufficient input parameters for "%s" creation | 原因: ミラー作成操作で無効なパラメータが指定されました。 |
| 104117 | FATAL | Insufficient input parameters for "%s" creation | 原因: ミラー作成操作で無効なパラメータが指定されました。 |
| 104118 | FATAL | Cannot unmount existing Mount Point "%s" | 原因: マウントポイントがビジーです。<br><br>処置: ファイルシステムがビジーではないことを確認してください。ファイルシ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ステムにアクセスする可能性があるプロセスまたはアプリケーションを停止してください。 |
| 104119 | FATAL | Invalid data replication resource type requested ("%s") | 原因: ミラー作成操作で無効なパラメータが指定されました。 |
| 104124 | EMERG | WARNING: A temporary communication failure has occurred between systems %s and %s. In order to avoid data corruption, data resynchronization will not occur. MANUAL INTERVENTION IS REQUIRED. In order to initiate data resynchronization, you should take one of the following resources out of service: %s on %s or %s on %s. The resource that is taken out of service will become the mirror target. | 原因: ソースサーバーとターゲットサーバーの間で、一時的な通信障害 (スプリットブレインシナリオ) が発生しました。<br><br>処置: メッセージテキストに記載されている手順を実行してください。 |
| 104125 | ERROR | failed to start '$cmd $_[2] $user_args' on '$_[3]': | 原因: 指定したコマンドが失敗しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 104126 | ERROR | $_ | 原因: このメッセージに、メッセージ 104125 で失敗したとレポートされているコマンドの結果が含まれています。 |
| 104128 | FATAL | comm path/server not specified | 原因: netraid.down スクリプトが、通信パスまたはサーバー名を指定せずに呼び出されました。このスクリプトは内部で呼び出されるため、常に適切なパラメーターを使用する必要があります。<br><br>処置: このエラーを SIOS サポートに報告してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 104129 | WARN | The replication connection for mirror “%s” (resource: “%s”) is down (reason: %s) | **原因**: ミラーへのレプリケーションの接続がダウン状態です。<br><br>**処置**: ネットワークを確認してください。 |
| 104130 | ERROR | Mirror resize failed on %s (%s). You must successfully complete this operation before using the mirror. Please try again. | **原因**: ミラーのサイズ変更操作で、リストされたシステム上にあるミラーのメタデータの更新に失敗しました。<br><br>**処置**: ミラーを使用するには、サイズへンコウを正常に完了する必要があります。「mirror_resize」を再実行してください (必要に応じて、強制実行する -f を使用)。 |
| 104132 | ERROR | The partition "%s" has an odd number of sectors and system "%s&qquot; is running kernel >= 4.12. Mirrors with this configuration will not work correctly with DataKeeper. Please see the SIOS product documentation for information on how to resize the mirror. | **原因**: ミラーの作成用に選択されたパーティションまたはディスクには奇数のディスクセクターがあるため、DataKeeperで使用するにはサイズを変更する必要があります。<br><br>**処置**:「parted」コマンドを使用してパーティションのサイズを変更するか、可能な場合はプラットフォーム（VMware、AWS）ツールを使用してディスクのサイズを変更します。注記：データが失われる可能性があるため、この操作は慎重に行ってください。 |
| 104136 | ERROR | Extend failed. | **原因**: 階層の拡張操作に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 104143 | ERROR | Mirror resume was unsuccessful ($ret) | **原因**:ミラーを確立できませんした。<br><br>**処置**: 関連するエラーをログで確認 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | し、レポートされている問題を解決してください。 |
| 104144 | ERROR | Unable to stop the mirror access for $self->{'md'} on system $self->{'sys'}. Error: $ret. Use the "mdadm —stop $self->{'md'}" command to manually stop the mirror. | **原因**: ミラーの一時停止時にターゲットノード上に作成されたミラーデバイスを停止できませんでした。<br><br>**処置**: デバイスがビジーでないこと、またはマウントされていないことを確認してください。「mdadm —stop」を手動で実行し、デバイスを停止してください。 |
| 104145 | WARN | Unable to dirty full bitmap.  Setting fullsync flag. | **原因**: ビットマップ全体をダーティにすることによる完全な再同期を実行できませんでした。代わりに fullsync フラグが使用されます。完全な同期は実行されるので、これは致命的なエラーではありません。<br><br>**処置**: ありません |
| 104146 | EMERG | WARNING: The target system %s for mirror %s has the target mirror %s currently active. In order to avoid data corruption, data resynchronization will not occur. MANUAL INTERVENTION IS REQUIRED. In order to initiate data resynchronization, you should reboot system %s. | **原因**: ミラーがターゲットシステムで構成されています。<br><br>**処置**: ターゲットシステムを再起動する必要があります。 その後、DataKeeper はミラーを再同期できるようになります。 |
| 104147 | EMERG | WARNING: The target system %s for mirror %s has the target disk %s currently mounted. In order to avoid data corruption, data resynchronization will not occur. MANUAL INTERVENTION IS REQUIRED. In order to initiate data resynchronization, you should unmount %s on %s. A full resync will occur. | **原因**: ミラーディスクがターゲットシステムにマウントされています。<br><br>**処置**: 完全なミラー再同期を開始するには、ミラーディスクをターゲットシステムでアンマウントする必要があります。ディスクで追跡されていない変更が発生したため、完全な再同期が必要です。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 104148 | EMERG | The storage configuration for mirror "%s (%s)" does not have a unique identifier and may have potential for data corruption in some environments in certain circumstances. Please refer to the SIOS Product Documentation for details on DataKeeper storage configuration options. | 原因: ミラーリング用に選択されたディスクが、オペレーティングシステムにUUIDを提供していません。DataKeeperはUUIDがないとディスクをミラーできません。<br><br>処置: ディスク上にGPTパーティションテーブルを作成して、ディスクパーティションにUUIDを提供できます。 DataKeeper for Linux トラブルシューティング を参照してください。 |
| 104155 | EMERG | The mirror %s cannot be forced online at this time. The underlying disk %s is mounted, indicating possible data corruption. MANUAL INTERVENTION IS REQUIRED. You must unmount %s on %s and restore the mirror to the last known mirror source system. A full resync will need to be performed from the source system to %s. | 原因: ターゲットシステムでミラーディスクがマウントされた。<br><br>処置: データ破損を防ぐため、すぐにディスクをアンマウントしてください。 |
| 104156 | WARN | Resynchronization of "%s" is in PENDING state. Current sync_action is: "%s" | 原因: md デバイスの再同期がペンディング状態であることが検出されました。<br><br>処置: LifeKeeper は再同期を強制実行して問題の修正を試行します。関連するエラーをログで確認してください。エラーを解決した後、/proc/mdstat でペンディング状態がクリアされたこと、および datarep リソースの再同期が進行中であるか、完了したことを確認してください。 |
| 104157 | WARN | /etc/sysconfig/raid-check update failed. Please %s "md%d" to SKIP_DEVS. | 原因: /etc/sysconfig/raid-check を変更して、スキップする MD デバイスのリスト（SKIP_DEVS）にエントリを追加または削除することができません。<br><br>処置: raid-check または SKIP_DEVS に関連するエラーがないか、システムログを確認してください。リストされたmdを手動で追加または削除してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 104158 | EMERG | WARNING: The local disk partition $self->{ part } for data replication device\n$self->{ md } has failed. MANUAL INTERVENTION IS REQUIRED. | **原因**: ミラーのローカルデバイスで問題が発生しました。LKDR_FAILURE でリカバリーアクションが設定されていないため、復旧するには手動による介入が必要です。<br><br>**処置**: システムログと LifeKeeper ログで、ローカルディスクに関連するエラーを確認してください。 |
| 104163 | WARN | The "%s_data_corrupt" flag is set in "%s/subsys/scsi/resources/netraid/" on system "%s". The mirror is being forced online. | **原因**: ミラーは強制的にオンラインになり、data_corrupt フラグを上書きします。指定されたシステム上のデータは、最新のデータとして扱われます。これが正しくない場合、データ破損またはデータ損失につながる可能性があります。<br><br>**処置**: ありません |
| 104164 | ERROR | The "%s_data_corrupt" flag for related mirror resource "%s" is set in "%s/subsys/scsi/resources/netraid/" on system "%s". To avoid data corruption, LifeKeeper will not restore this mirror or any related mirrors in the hierarchy. | **原因**: data_corrupt フラグは、階層内の1つ以上のミラーに存在します。追加データの破損を防ぐため、すべての data_corrupt フラグが解決されるまで、どのミラーも In Service になりません。<br><br>**処置**: LifeKeeperのログを確認して、各ミラーが最後に In Service になっていた場所、つまり各ミラーの最新データが存在する場所を特定します。すべての階層が In Service であった以前のソースでミラーを In Service にし、ミラーがすべてのターゲットと同期できるようにする必要があります。 |
| 104165 | ERROR | The "%s_data_corrupt" flag for related mirror resource "%s" is set in "%s/subsys/scsi/resources/netraid/" on system "%s". The mirror resource "%s" is being forced online. | **原因**: ミラーは強制的にオンラインになっており、data_corrupt フラグを上書きします。指定されたシステム上のデータは、すべてのターゲットと同期される正しいデータとして扱われます。これが最 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 新のデータでない場合、データ破損またはデータ損失につながる可能性があります。<br><br>**処置**: ありません |
| 104170 | ERROR | Failed to create "source" flag file on shared source %s to track mirror source. This may result in a full resync. | **原因**:「ソース」フラグファイルが、ミラーソースを追跡するために指定されたシステムで作成されませんでした。<br><br>**処置**: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104171 | ERROR | Failed to create "source" flag file on %s to track mirror source. Target %s will not be added to mirror. | **原因**:「ソース」フラグファイルが、ミラーソースを追跡するために指定されたシステムで作成されませんでした。<br><br>**処置**: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104172 | ERROR | The "source" flag file on %s does not contain a valid target (%s). Full resync to remaining targets is required. | **原因**:「ソース」フラグファイルには以前のソースのシステム名が含まれている必要がありますが、リストされている名前は構成されたシステムのリストにありませんでした。<br><br>**処置**: この問題を SIOS サポートに報告してください。 |
| 104173 | ERROR | Failed to create "source" flag file on %s to track mirror source. | **原因**:「ソース」フラグファイルが、ミラーソースを追跡するために指定されたシステムで作成されませんでした。<br><br>**処置**: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | Full になっていないか確認してください。 |
| 104174 | ERROR | Failed to create “previous_source” flag file to track time waiting on source. Will not be able to timeout. | 原因:「previous_source」フラグファイルは、ミラーの以前のソースを追跡するためのミラーソース上に作成されませんでした。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104175 | ERROR | Failed to create "data_corrupt" flag file on target "%s". | 原因: リストされたターゲットに「data_corrupt」フラグファイルが作成されませんでした。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104176 | ERROR | The "source" flag file on %s to track mirror source does not exist. Full resync is required. | 原因:「ソース」フラグファイルはシステム上に存在する必要があり、これがないとミラーの整合性を検証できません。データの信頼性を確保するには、完全な再同期が必要です。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104177 | ERROR | Failed to determine amount of time waiting on %s. | 原因: 以前のソースを待機する時間を決定できませんでした。以前のソースが見つからない場合、ターゲットは完全に再同期されて追加されます。<br><br>処置: ありません |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 104178 | ERROR | Failed to update "source" flag file on target "%s", previous source must be merged first. | 原因: ターゲットのソースフラグファイルは、同期されて停止されると更新されるため、次のIn Service時には以前のソースは必要ありません。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104180 | ERROR | Internal Error: "previous_source" has the local system name (%s). | 原因: ローカルシステム名は、previous_source フラグファイルに含めることはできません。<br><br>処置: このエラーを SIOS サポートに報告してください。 |
| 104181 | ERROR | Internal Error: There are no targets waiting on %s to be merged. | 原因: 以前のソースがマージされるのを待機しているターゲットがありません。<br><br>処置: このエラーを SIOS サポートに報告してください。 |
| 104182 | ERROR | Failed to create \"source\" flag file on %s to track mirror source. This may result in a full resync. | 原因: リストされたターゲットに「ソース」フラグファイルが作成されませんでした。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104186 | ERROR | Failed to create "last_owner" flag file on %s to track mirror source. This may allow in-service of mirror on old data. | 原因:「last_owner」フラグファイルがソースに作成されませんでした。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 104187 | WARN | $REM_MACH has ${REM_TAG}_last_owner file, create flag ${FLAGTAG}_data_corrupt. | **原因**: リストされているシステムでは、ミラーが最後にIn Serviceになりました。<br><br>**処置**: リストされているシステムには last_owner ファイルがあります。これは、このシステムには最新のデータがあり、データの損失を防ぐためにミラーをIn Serviceにするのに最適なシステムである可能性が高いことを示しています。 |
| 104188 | WARN | $REM_MACH is not alive, create flag ${FLAGTAG}_data_corrupt. | **原因**: リストされているシステムは稼働していません。<br><br>**処置**: リストされているシステムは稼働していないため、そのシステムがローカルシステムよりも新しいミラーソースであったかどうかを判断できません。このため、ローカルシステムがミラーをIn Serviceにすることを自動的に許可しないでください。 |
| 104200 | EMERG | Continue to wait for %s to merge bitmap and do partial resyncs to all targets, no timeout set. | **原因**: マルチターゲット構成では、すべてのターゲットが部分的に再同期できるように、以前のソースがビットマップをマージできるようになるまでターゲットは構成されません。/etc/defaults/LifeKeeper の LKDR_WAIT_ON_PREVIOUS_SOURCE_TIMEOUT エントリは、無期限に待機するために「-1」に設定されます。<br><br>**処置**: メッセージにリストされている以前のソースのステータスを確認し、クラスターへの再参加を妨げている問題を解決してください。 |
| 104201 | EMERG | To stop waiting for the previous source (forcing a full resync to remaining waiting targets) run: "%s/bin/mirror_action %s fullresync %s %s" on %s. | **原因**: マルチターゲット構成では、ターゲットは構成されず、以前のソースがクラスターに再参加するのを待機します。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: メッセージにリストされているコマンドを実行して、このターゲットおよび再同期を待機している残りのターゲットに対して即時の完全再同期を強制してください。 |
| 104202 | EMERG | Continue to wait for %s to merge bitmap and do partial resyncs to all targets. Continue to wait %d more seconds. | **原因**: マルチターゲット構成では、すべてのターゲットが部分的に再同期できるように、前のソースがビットマップをマージできるようになるまでターゲットは構成されません。 /etc/defaults/LifeKeeper の LKDR_WAIT_ON_PREVIOUS_SOURCE_TIMEOUT エントリは、待機する秒数に設定されます。以前のソースがその時間内にクラスターに参加しない場合、ターゲットは完全に再同期されて追加されます。<br><br>**処置**: メッセージにリストされている以前のソースのステータスを確認し、クラスターへの再参加を妨げている問題を解決してください。 |
| 104203 | EMERG | To stop waiting for the previous source (forcing a full resync to remaining waiting targets) run: \"%s/bin/mirror_action %s fullresync %s %s\" on %s. | **原因**: マルチターゲット構成では、ターゲットは構成されず、以前のソースがクラスターに再参加するのを待機します。<br><br>**処置**: メッセージにリストされているコマンドを実行して、このターゲットおよび再同期を待機している残りのターゲットに対して即時の完全再同期を強制してください。<br><br>**注記**: リストされたターゲットが削除されて戻らない場合でも、このコマンドを実行して待機を停止してください。 |
| 104207 | ERROR | Failed to create "data_corrupt" flag file on "%s". | **原因**: リストされたソースに「data_corrupt」フラグファイルが作成されませんでした。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104208 | ERROR | Failed to create "data_corrupt" flag file on target "%s". | 原因: リストされたターゲットに「data_corrupt」フラグファイルが作成されませんでした。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104209 | ERROR | Failed to create "data_corrupt" flag file on "%s". | 原因: リストされたソースに「data_corrupt」フラグファイルが作成されませんでした。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104210 | ERROR | Failed to create "data_corrupt" flag file on target "%s". | 原因: リストされたターゲットに「data_corrupt」フラグファイルが作成されませんでした。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104211 | ERROR | Failed to create "data_corrupt" flag file on target "%s". | 原因: リストされたターゲットに「data_corrupt」フラグファイルが作成されませんでした。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してくださ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | い。 |
| 104212 | ERROR | The "source" flag file on %s to track mirror source does not exist. Full resync to remaining targets is required. | 原因:「ソース」フラグファイルはシステム上に存在する必要があり、これがないとミラーの整合性を検証できません。データの信頼性を確保するには、完全な再同期が必要です。まだミラーされていないすべてのターゲットでは、完全な再同期が必要です。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください. |
| 104214 | ERROR | Failed to create "source" flag file on %s to track mirror source. | 原因:「ソース」フラグファイルが、ミラーソースを追跡するために指定されたシステムで作成されませんでした。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104216 | ERROR | Failed to create "data_corrupt" flag file on target "%s". | 原因: リストされたターゲットに「data_corrupt」フラグファイルが作成されませんでした。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104217 | ERROR | Failed to create "source" flag file on shared source %s to track mirror source. This may result in a full resync. | 原因:「ソース」フラグファイルが、ミラーソースを追跡するために指定されたシステムで作成されませんでした。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | Full になっていないか確認してください。 |
| 104218 | ERROR | Failed to create "data_corrupt" flag file on target "%s". | 原因: リストされたターゲットに「data_corrupt」フラグファイルが作成されませんでした。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104221 | ERROR | Failed to create "last_owner" flag file on %s to track mirror source. This may allow in-service of mirror on old data. | 原因: リストされたターゲットに「last_owner」フラグファイルが作成されませんでした。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104222 | ERROR | Failed to create "last_owner" flag file to track mirror source". This may allow in-service of mirror on old data. | 原因:「last_owner」フラグファイルは、ミラーが最後にIn Serviceになった場所を特定するために使用されています。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104223 | ERROR | Failed to create "last_owner" flag file to track mirror source". This may allow in-service of mirror on old data. | 原因:「last_owner」フラグファイルは、ミラーが最後にIn Serviceになった場所を特定するために使用されています。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してくださ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | い。 |
| 104224 | ERROR | Failed to create "previous_source" flag file. | 原因:「previous_source」フラグファイルが作成されませんでした。これは、以前のソースのビットマップをマージして完全な同期を回避するために必要です。<br><br>処置: LKROOT（/opt/LifeKeeper）ファイルシステムにエラーがないか、または Full になっていないか確認してください。 |
| 104227 | ERROR | Failed to set %s to %s. | 原因: このメッセージは、nbdドライバー（/sys/block/nbdX）のsysfsパラメーターの設定に失敗したことを示しています。<br><br>処置: このエラーを回避するには、/etc/default/LifeKeeper で次のうち1つ以上を調整する必要がある場合があります。<br>NBD_NR_REQUESTS<br>NBD_SCHEDULER<br>LKDR_ASYNC_LIMIT |
| 104232 | ERROR | Mirror resize failed on %s (%s). Could not set size to %d. | 原因: ミラーのサイズ変更操作に失敗しました。 |
| 104233 | ERROR | Mirror resize failed on %s (%s). Could not set bitmap to %s and bitmap-chunk to %d. | 原因: ミラーのサイズ変更操作に失敗しました。 |
| 104234 | ERROR | The mirror %s failed to resize. You must successfully complete this operation before using the mirror. Please try again. | 原因: ミラーのサイズ変更操作に失敗しました。 |
| 104235 | ERROR | mirror_resize of mirror %s failed due to signal "%s". | 原因: ミラーのサイズ変更操作に失敗しました。 |
| 104236 | EMERG | Resource "%s" is "OSF". The mirror "%s" | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | will wait to replicate data until all resources in the hierarchy%s are in-service. This may indicate inconsistent data. Verify the data is correct before replicating data; replication will continue when all resources in the hierarchy%s are in-service. A full resync may be necessary (see "LKDR_WAIT_TO_RESYNC" in /etc/default/LifeKeeper). | |
| 104237 | WARN | Resource "%s" is "OSU". The mirror "%s" will wait to replicate data until all resources in the hierarchy%s are in-service. To replicate data immediately run: "%s/bin/mirror_action %s resume" on "%s" (see "LKDR_WAIT_TO_RESYNC" in /etc/default/LifeKeeper). | |
| 104238 | ERROR | Unable to read /sys/module/nbd/taint. Assuming that the SIOS nbd module is not loaded. | |
| 104239 | EMERG | $nbd_failure_msg | |
| 104242 | ERROR | Unable to read /proc/modules. Assuming that the SIOS nbd module is not loaded. | |
| 104243 | ERROR | Internal script or routine $caller was called for unsupported kernel module '$module'. Supported kernel modules for this script or routine are: $module_list. | **原因**: 指定されたスクリプトまたはルーチンが、サポートされていないカーネルモジュールに対して呼び出されました。<br><br>**処置**: これは内部エラーです。SIOSカスタマーサポートに連絡してください。 |
| 104244 | ERROR | Output from 'modprobe $module' command: $pretty_modprobe_out | **原因**: modprobeコマンドを使用して指定されたカーネルモジュールをロードできませんでした。<br><br>**処置**: 失敗したmodprobeのログメッセージの出力を確認し、見つかった問題を解決してください。 |
| 104251 | ERROR | There is no LifeKeeper protected resource with tag $tag on system $me. | **原因**: 指定されたタグは、指定されたシステム上のLifeKeeperで保護されている |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | リソースに対応していません。<br><br>**処置**: リソースタグとシステム名が正しいことを確認してください。 |
| 104252 | ERROR | Resource $tag is not a $app/$typ resource. Please use the $ins_app/$ins_typ resource-specific canfailover script instead. | **原因**: scsi/netraid 固有の canfailover スクリプトが、非 scsi/netraid リソースに対して呼び出されました。<br><br>**処置**: 指定されたリソースの適切なアプリケーションとタイプに対応する canfailover スクリプトを使用してください（存在する場合）。 |

# 8.1.2. DB2 Kit メッセージカタログ

**Control F** を使用して、各カタログの特定のエラーコードを検索します。エラーコードの検索には、画面右上の検索ボックスをご利用ください。

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 103001 | ERROR | LifeKeeper has detected an error while trying to determine the node number(s) of the DB partition server(s) for the instance | 原因: db2nodes.cfg にサーバ名が含まれていません。<br><br>処置: db2nodes.cfg が有効であることを確認してください。 |
| 103002 | ERROR | LifeKeeper was unable to get the version for the requested instance "%s" | 原因:「db2level」コマンドが DB2 のバージョンを返しませんでした。<br><br>処置: DB2 の設定を確認してください。 |
| 103003 | ERROR | LifeKeeper has detected an error while trying to determine the node number(s) of the DB partition server(s) for the instance | 原因: DB2 Application Recovery Kit は、DB2 インスタンスのノードを見つけることができませんでした。<br><br>処置: DB2 の設定を確認してください。 |
| 103004 | ERROR | Unable to get the information for resource "%s" | 原因: リソース情報の取得に失敗しました。<br><br>処置: LifeKeeper の設定を確認してください。 |
| 103005 | ERROR | Unable to get the information for resource "%s" | 原因: リソース情報の取得に失敗しました。<br><br>処置: LifeKeeper の設定を確認してください。 |
| 103006 | ERROR | Unable to get the instance information for resource "%s" | 原因: インスタンス情報の取得に失敗しました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: LifeKeeper の設定を確認してください。 |
| 103007 | ERROR | Unable to get the instance home directory information for resource "%s" | **原因**: インスタンスのホームディレクトリのパスの取得に失敗しました。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 103008 | ERROR | Unable to get the instance type information for resource "%s" | **原因**: DB2 Application Recovery Kit は、無効なインスタンスタイプを見つけました。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 103009 | ERROR | LifeKeeper has encountered an error while trying to get the database configuration parameters for database "$DB" | **原因**:「db2 get db cfg for $DB」コマンドの実行中に予期しないエラーが発生しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 103012 | ERROR | LifeKeeper was unable to start the database server for instance "%s" | **原因**: 要求された DB2 インスタンスの起動が失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。レポートされているエラーを修正してから、「restore」操作を再試行してください。 |
| 103013 | ERROR | LifeKeeper was unable to start the database server for instance "%s" | **原因**: 要求された DB2 インスタンスの起動が失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してくだ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | さい。レポートされているエラーを修正してから、「restore」操作を再試行してください。 |
| 103015 | ERROR | An entry for the home directory "%s" of instance "%s" does not exist in "/etc/fstab" | **原因**: マルチパーティションのデータベースのインスタンスのホームディレクトリは、「/etc/fstab」に存在する必要があります。<br><br>**処置**: ホームディレクトリが「/etc/fstab」に存在することを確認してください。 |
| 103016 | ERROR | LifeKeeper was unable to mount the home directory for the DB2 instance "%s" | **原因**: マルチパーティションのデータベースのインスタンスのホームディレクトリをマウントできませんでした。<br><br>**処置**: ホームディレクトリがマウントされていることを確認し、操作を再試行してください。 |
| 103017 | ERROR | Unable to get the instance nodes information for resource "%s" | **原因**: インスタンスノードの取得に失敗しました。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 103018 | ERROR | LifeKeeper was unable to start database partition server "%s" for instance "%s" | **原因**: 要求された DB2 インスタンスの起動が失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。レポートされているエラーを修正してから、「restore」操作を再試行してください。 |
| 103020 | ERROR | LifeKeeper was unable to stop the database server for instance "%s" | **原因**: 要求された DB2 インスタンスのシャットダウンが失敗しました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。レポートされているエラーを修正してから、「remove」操作を再試行してください。 |
| 103021 | ERROR | LifeKeeper was unable to stop the database server for instance "%s" | **原因**: 要求された DB2 インスタンスのシャットダウンが失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。レポートされているエラーを修正してから、「remove」操作を再試行してください。 |
| 103023 | ERROR | Unable to get the instance nodes information for resource "%s" | **原因**: インスタンスノードの取得に失敗しました。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 103024 | ERROR | LifeKeeper was unable to stop database partition server "%s" for instance "%s" | **原因**: 要求された DB2 インスタンスのシャットダウンが失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。レポートされているエラーを修正してから、「remove」操作を再試行してください。 |
| 103026 | ERROR | Unable to get the instance nodes information for resource "%s" | **原因**: インスタンスノードの取得に失敗しました。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 103027 | FATAL | The argument for the DB2 instance is empty | **原因**: 作成操作で無効なパラメータが指定されました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: パラメータを確認し、操作を再試行してください。 |
| 103028 | FATAL | Unable to determine the DB2 instance home directory | **原因**: DB2 Application Recovery Kit は、DB2 インスタンスのホームディレクトリを特定できませんでした。<br><br>**処置**: インスタンスの所有者の名前がインスタンス名と同じであることを確認し、操作を再試行してください。 |
| 103029 | FATAL | Unable to determine the DB2 instance type | **原因**: DB2 Application Recovery Kit は、DB2 インスタンスのタイプを特定できませんでした。<br><br>**処置**: DB2 の設定を確認してください。 |
| 103030 | FATAL | LifeKeeper has detected an error while trying to determine the node number(s) of the DB partition server(s) for the instance | **原因**: DB2 Application Recovery Kit は、DB2 インスタンスのノードを見つけることができませんでした。<br><br>**処置**: DB2 の設定を確認してください。 |
| 103031 | ERROR | The path "%s" is not on a shared filesystem | **原因**: インスタンスのホームディレクトリは、共有ファイルシステム上になければなりません。<br><br>**処置**: パスが共有ファイルシステム上にあることを確認し、作成操作を再試行してください。 |
| 103032 | ERROR | LifeKeeper was unable to get the DB tablespace containers for instance "%s" or the log path for one of its databases | **原因**: LifeKeeper はデータベースのテーブルスペースコンテナの場所を特定できなかったか、コンテナがマウント済みのファイルシステム上のパスにあることを確認できませんでした。<br><br>**処置**: 関連するエラーをログで確認し、 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | レポートされている問題を解決してください。レポートされているエラーを修正してから、「create」操作を再試行してください。 |
| 103033 | ERROR | The path "%s" is not on a shared filesystem | **原因**: データベースのテーブルスペースコンテナのパスは、共有ファイルシステム上になければなりません。<br><br>**処置**: データベースのテーブルスペースコンテナが共有ファイルシステム上にあることを確認し、操作を再試行してください。 |
| 103034 | ERROR | A DB2 Hierarchy already exists for instance "%s" | **原因**: LifeKeeper によってすでに保護されている DB2 インスタンスを保護しようとしました。<br><br>**処置**: 別の DB2 インスタンスを LifeKeeper の保護用に選択する必要があります。 |
| 103035 | ERROR | The file system resource "%s" is not in-service | **原因**: DB2 リソースが依存するファイルシステムは in service である必要があります。<br><br>**処置**: ファイルシステムリソースが in service であることを確認して、「create」操作を再試行してください。 |
| 103036 | ERROR | Unable to create the hierarchy for raw device "%s" | **原因**: LifeKeeper は、リソース {raw device} を作成できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから「create」操作を再試行する必要があります。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 103037 | ERROR | A RAW hierarchy does not exist for the tag "%s" | 原因: LifeKeeper は、raw リソース {tag} を見つけられませんでした。<br><br>処置: LifeKeeper の設定を確認してください。 |
| 103038 | ERROR | LifeKeeper was unable to create a dependency between the DB2 hierarchy "%s" and the Raw hierarchy "%s" | 原因: 要求された親 DB2 リソースと子 Raw リソースとの間の依存関係の作成に失敗しました。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから、「create」操作を再試行してください。 |
| 103039 | ERROR | LifeKeeper could not disable the automatic startup feature of DB2 instance "%s" | 原因: DB2 の設定を更新しようとして、予期しないエラーが発生しました。<br><br>処置: システム起動時にインスタンスが自動で起動されないようにするために、DB2AUTOSTART を手動で更新する必要があります。 |
| 103040 | ERROR | DB2 version "%s" is not installed on server "%s" | 原因: LifeKeeper は DB2 のインストール場所を見つけられませんでした。<br><br>処置: DB2 の設定を確認してください。 |
| 103041 | ERROR | The instance owner "%s" does not exist on target server "%s" | 原因:「canextend」操作または「extend」操作の間に、テンプレートサーバから DB2 インスタンスの所有者を取得できませんでした。<br><br>処置: 指定したサーバに DB2 インスタンスの所有者が存在していることを確認してください。所有者が存在しない場合、クラスタ内のすべてのサーバで、同じ uid および gid を使用して作成する必要 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | があります。 |
| 103042 | ERROR | The instance owner "%s" uids are different on target server "%s" and template server "%s" | 原因: DB2 インスタンスの所有者 {user} のターゲットサーバ {target server} のユーザ ID が、テンプレートサーバ {template server} のユーザ {user} の値と一致しません。<br><br>処置: DB2 インスタンスの所有者 {user} のユーザ ID はクラスタ内のすべてのサーバで一致している必要があります。ユーザ ID の不一致をすべてのサーバで手動で修正し、「canextend」操作を再試行する必要があります。 |
| 103043 | ERROR | The instance owner "%s" gids are different on target server "%s" and template server "%s" | 原因:DB2 インスタンスの所有者 {user} のターゲットサーバ {target server} のグループ ID が、テンプレートサーバ {template server} のユーザ {user} の値と一致しません。<br><br>処置: DB2 インスタンスの所有者 {user} のグループ ID はクラスタ内のすべてのサーバで一致している必要があります。グループ ID の不一致をすべてのサーバで手動で修正し、「canextend」操作を再試行する必要があります。 |
| 103044 | ERROR | The instance owner "%s" home directories are different on target server "%s" and template server "%s" | 原因: ターゲットサーバ {target server} のユーザ {user} のホームディレクトリが、テンプレートサーバ {template server} の DB2 インスタンスの所有者のホームディレクトリと一致しません。<br><br>処置: DB2 インスタンスの所有者 {user} のホームディレクトリは、クラスタ内のすべてのサーバで一致している必要があります。ディレクトリの不一致をすべてのサーバで手動で修正し、「canextend」操作を再試行する必要があります。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 103045 | ERROR | LifeKeeper was unable to get the DB2 "SVCENAME" parameter for the DB2 instance | 原因:「db2 get dbm cfg」コマンドの実行中に予期しないエラーが発生しました。<br><br>処置: DB2 の設定を確認してください。 |
| 103046 | ERROR | Unable to get the value of the DB2 "SVCENAME" parameter for the DB2 instance %s. | 原因: DB2 の「SVCENAME」パラメータが null に設定されています。<br><br>処置: DB2 の設定を確認してください。 |
| 103047 | ERROR | LifeKeeper was unable to get the contents of the "/etc/services" file on the server "%s" | 原因: テンプレートサーバの「/etc/services」に、DB2 インスタンスのサービス名が含まれていません。<br><br>処置:「/etc/services」内にある DB2 インスタンスのサービス名は、クラスタ内のすべてのサーバで一致している必要があります。サービス名の不一致をすべてのサーバで手動で修正し、「canextend」操作を再試行する必要があります。 |
| 103048 | ERROR | LifeKeeper was unable to get the contents of the "/etc/services" file on the server "%s" | 原因: ターゲットサーバの「/etc/services」に、DB2 インスタンスのサービス名が含まれていません。<br><br>処置:「/etc/services」内にある DB2 インスタンスのサービス名は、クラスタ内のすべてのサーバで一致している必要があります。サービス名の不一致をすべてのサーバで手動で修正し、「canextend」操作を再試行する必要があります。 |
| 103049 | ERROR | The "/etc/services" entries for the instance "%s" are different on target server "%s" and template server "%s" | 原因: インスタンスの「/etc/services」エントリに不一致があります。<br><br>処置:「/etc/services」内にある DB2 インスタンスのサービス名は、クラスタ内のすべてのサーバで一致している必要があります。サービス名の不一致をすべての |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | サーバで手動で修正し、「canextend」操作を再試行する必要があります。 |
| 103050 | ERROR | The home directory "%s" for instance "%s" is not mounted on server "%s" | **原因**: LifeKeeper はマルチパーティションインスタンス用の db2nodes.cfg を見つけられませんでした。<br><br>**処置**: ホームディレクトリがマウントされていることを確認し、操作を再試行してください。 |
| 103051 | ERROR | Error getting resource information for resource "%s" on server "%s" | **原因**: テンプレートサーバからのリソース情報の取得に失敗しました。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 103052 | ERROR | LifeKeeper was unable to add instance "%s" and/or its variables to the DB2 registry | **原因**:「db2iset」コマンドの実行中に予期しないエラーが発生しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 103053 | ERROR | Usage: %s instance | |
| 103054 | ERROR | Unable to determine the DB2 instance type | **原因**: DB2 Application Recovery Kit は、DB2 インスタンスのタイプを特定できませんでした。<br><br>**処置**: DB2 の設定を確認してください。 |
| 103055 | ERROR | LifeKeeper has detected an error while trying to determine the node number(s) of the DB partition server(s) for the instance | **原因**: DB2 Application Recovery Kit は、DB2 インスタンスのノードを見つけることができませんでした。<br><br>**処置**: DB2 の設定を確認してください。 |
| 103056 | ERROR | Usage: %s instance | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 103058 | ERROR | Usage: %s instance | |
| 103059 | ERROR | Usage: %s instance | |
| 103060 | ERROR | Unable to determine the DB2 instance home directory | 原因: DB2 Application Recovery Kit は、DB2 インスタンスのホームディレクトリを特定できませんでした。<br><br>処置: インスタンスの所有者の名前がインスタンス名と同じであることを確認し、操作を再試行してください。 |
| 103061 | ERROR | Unable to determine the DB2 instance type | 原因: DB2 Application Recovery Kit は、DB2 インスタンスのタイプを特定できませんでした。<br><br>処置: DB2 の設定を確認してください。 |
| 103062 | ERROR | LifeKeeper has detected an error while trying to determine the node number(s) of the DB partition server(s) for the instance | 原因: DB2 Application Recovery Kit は、DB2 インスタンスの該当するノードを見つけることができませんでした。<br><br>処置: DB2 の設定を確認してください。 |
| 103063 | ERROR | Unable to determine the DB2 install path | 原因: DB2 Application Recovery Kit は、そのインスタンスの DB2 を見つけることができませんでした。<br><br>処置: DB2 の設定を確認してください。 |
| 103064 | ERROR | Usage: nodes -t tag -a add_nodenum \| nodes -t tag -d delete_nodenum \| nodes -t tag -p | |
| 103065 | ERROR | Invalid input provided for "%s" utility operation, characters are not allowed. | 原因:「nodes」コマンドで無効なパラメータが指定されました。<br><br>処置: パラメータを確認し、操作を再試行してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 103066 | ERROR | Unable to get the information for resource "%s" | **原因**: LifeKeeper は、リソース {tag} を見つけられませんでした。<br><br>**処置**: パラメータを確認し、操作を再試行してください。 |
| 103067 | ERROR | The DB2 instance "%s" is not a EEE or Multiple Partition instance | **原因**: リソースタグ {tag} はシングルパーティションインスタンスです。<br><br>**処置**: パラメータを確認し、操作を再試行してください。 |
| 103069 | ERROR | Node "%s" is already protected by this hierarchy | **原因**:「nodes」コマンドで無効なパラメータが指定されました。<br><br>**処置**: パラメータを確認し、操作を再試行してください。 |
| 103070 | ERROR | Node number "%s" is the last remaining node protected by resource "%s". Deleting all nodes is not allowed. | **原因**:「nodes」コマンドで無効なパラメータが指定されました。<br><br>**処置**: パラメータを確認し、操作を再試行してください。 |
| 103071 | ERROR | LifeKeeper is unable to get the equivalent instance for resource "%s" | **原因**:「nodes」コマンドの実行中に予期しないエラーが発生しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 103072 | ERROR | Unable to set NodesInfo for resource "%s" on "%s" | **原因**:「nodes」コマンドの実行中に予期しないエラーが発生しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 103073 | ERROR | Unable to set NodesInfo for resource "%s" on "%s" | 原因:「nodes」コマンドの実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 103074 | ERROR | Usage: %s instance | |
| 103075 | ERROR | Usage: %s instance | |
| 103076 | ERROR | Unable to determine the DB2 instance type | 原因: DB2 Application Recovery Kit は、DB2 インスタンスのタイプを特定できませんでした。<br><br>処置: DB2 の設定を確認してください。 |
| 103077 | ERROR | Unable to determine the DB2 instance home directory | 原因: DB2 Application Recovery Kit は、DB2 インスタンスのホームディレクトリを特定できませんでした。<br><br>処置: インスタンスの所有者の名前がインスタンス名と同じであることを確認し、操作を再試行してください。 |
| 103078 | ERROR | The database server is not running for instance "%s" | 原因: DB2 インスタンスのプロセスチェックで、実行中のプロセスが見つかりませんでした。<br><br>処置: DB2 インスタンスを開始してください。 |
| 103079 | ERROR | LifeKeeper has detected an error while trying to determine the node number(s) of the DB partition server(s) for the instance | 原因: DB2 Application Recovery Kit は、DB2 インスタンスのノードを見つけることができませんでした。<br><br>処置: DB2 の設定を確認してください。 |
| 103080 | ERROR | One or more of the database partition servers for instance "%s" is down | 原因: すべてのデータベースパーティシ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ョンサーバが実行されている必要があります。<br><br>**処置**: すべてのデータベースパーティションサーバが実行されていることを確認し、操作を再試行してください。 |
| 103081 | ERROR | DB2 local recovery detected another recovery process in progress for "%s" and will exit. | |
| 103082 | ERROR | Failed to create flag "%s" | **原因**: DB2 ローカルリカバリ処理を制御するためのフラグを作成しようとして、予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 103083 | ERROR | Failed to remove flag "%s" | **原因**: DB2 ローカルリカバリ処理を制御するためのフラグを削除しようとして、予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 103084 | ERROR | Unable to determine the DB2 instance \"$Instance\" home directory | **原因**: DB2 Application Recovery Kit は、DB2 インスタンスのホームディレクトリを特定できませんでした。<br><br>**処置**: インスタンスの所有者の名前がインスタンス名と同じであることを確認し、操作を再試行してください。 |

# 8.1.3. DMMP Kit メッセージカタログ

**Control F** を使用して、各カタログの特定のエラーコードを検索します。エラーコードの検索には、画面右上の検索ボックスをご利用ください。

| Code | Severity | Message | Cause/Action |
|---|---|---|---|
| 128005 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | 原因: オペレーティングシステムの信号 {signal} によって、{server} 上の {resource} の quickCheck が失敗しました。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128008 | ERROR | Usage: quickCheck -t <tag name> -i <id> | 原因: dmmp デバイスの quickCheck コマンドに不正な引数が指定されたため、コマンドが実行できませんでした。<br><br>処置: すべてのソフトウェアコンポーネントの正しいバージョンが適切にインストールされていることを確認してください。次の正しい引数を指定してコマンドを再実行してください。-t <Resource Tag> および -i <Resource ID> (quickCheck 対象の dmmp デバイスリソースを特定する ID) |
| 128010 | ERROR | quickCheck for "%s" failed checks of underlying paths, initiate recovery. retry count=%s. | 原因: dmmp キットが {count} 回の試行後、デバイスの quickCheck に失敗しました。保護されている dmmp リソースのリカバリが実行されます。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |

| Code | Severity | Message | Cause/Action |
|---|---|---|---|
| 128021 | ERROR | unable to find device for uuid "%s". | 原因: restore 操作時に一意の ID を使用してデバイスを見つけられませんでした。<br><br>処置: リソースが正しく設定されていることを確認してください。リストア対象の dmmp デバイスリソースを特定する正しいデバイス ID でコマンドを再実行してください。 |
| 128025 | ERROR | Device "%s" failed to unlock. | 原因: restore 操作時に正常に動作していない {device} が検出され、ロック解除できませんでした。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128026 | ERROR | Device "%s" failed to lock. | 原因: リストア時に {device} をロックできませんでした。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128031 | ERROR | unable to find device for uuid "%s". | 原因: remove 操作時に一意の ID を使用してデバイスを見つけられませんでした。<br><br>処置: リソースが正しく設定されていることを確認してください。削除対象の dmmp デバイスリソースを特定する正しいデバイス ID でコマンドを再実行してください。 |

| Code | Severity | Message | Cause/Action |
|---|---|---|---|
| 128034 | ERROR | Device "%s" failed to unlock. | **原因**: 削除時に {device} をロック解除できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128036 | ERROR | unable to load existing information for device with uuid "%s". | **原因**: 一意の ID を使用してデバイス情報を読み込めませんでした。<br><br>**処置**: リソースが正しく設定されていることを確認してください。dmmp デバイスリソースを特定する正しいデバイス ID でコマンドを再実行してください。 |
| 128037 | ERROR | unable to load existing information for device "%s". | **原因**: 名前を使用してデバイス情報を読み込めませんでした。<br><br>**処置**: リソースが正しく設定されていることを確認してください。dmmp デバイスリソースを特定する正しいデバイス名でコマンドを再実行してください。 |
| 128038 | ERROR | unable to load existing information for device, no dev or uuid defined. | **原因**: 一意のデバイス ID もデバイス名も指定されていないため、デバイス情報を読み込めませんでした。<br><br>**処置**: リソースが正しく設定されていることを確認してください。dmmp デバイスリソースを特定する正しいデバイス ID またはデバイス名でコマンドを再実行してください。 |
| 128041 | ERROR | unable to load existing information for device with uuid "%s". | **原因**: 一意の ID を使用してデバイス情報を読み込めませんでした。<br><br>**処置**: リソースが正しく設定されている |

| Code | Severity | Message | Cause/Action |
|---|---|---|---|
| | | | ことを確認してください。dmmp デバイスリソースを特定する正しいデバイス ID でコマンドを再実行してください。 |
| 128057 | ERROR | All paths are failed on "%s". | 原因: LifeKeeper は、保護対象の dmmp デバイスへのすべてのパスが障害状態にあることを検出しました。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |
| 128058 | ERROR | could not determine registrations for "%s"! All paths failed. | 原因: LifeKeeper は、保護対象の dmmp {device} の登録を特定できませんでした。dmmp {device} へのすべてのパスが障害状態にあります。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |
| 128059 | WARN | path "%s" no longer configured for "%s", remove from path list. | 原因:LifeKeeper が検出した保護対象の {device} への {path} は、現在は有効ではなく、パスリストから削除されます。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |
| 128060 | WARN | registration failed on path "%s" for "%s". | 原因: LifeKeeper は、保護対象の dmmp {device} の {path} の登録を特定できませんでした。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |
| 128062 | ERROR | all paths failed for "%s". | 原因: LifeKeeper は、保護対象の dmmp |

| Code | Severity | Message | Cause/Action |
|---|---|---|---|
| | | | {device} の有効なパスを確認できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |
| 128072 | ERROR | The daemon "%s" does not appear to be running and could not be restarted. Path failures may not be correctly handled without this daemon. | **原因**: LifeKeeper は dmmp デーモンが実行されていることを確認できなかったため、デーモンを再起動できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |
| 128078 | ERROR | "%s" resource type is not installed on "%s". | **原因**: dmmp デバイスをサポートするための Device Mapper Multipath Recovery Kit がシステムにインストールされていません。<br><br>**処置**: steeleye-lkDMMP Device Mapper Multipath Recovery Kit rpm をシステムにインストールしてください。 |
| 128083 | ERROR | This script must be executed on "%s". | **原因**: dmmp デバイスリソースを作成するための devicehier スクリプトの引数に不正なシステム名が指定されました。<br><br>**処置**: クラスタノードとコミュニケーションパスが適切に設定されていることを確認してください。devicehier スクリプトに正しいシステム名を指定してください。この名前は、コマンドを実行するシステムの名前に一致する必要があります。 |
| 128084 | ERROR | The device %s is not active. | **原因**: リソース作成時に LifeKeeper は、指定された {device} をシステム上の有 |

| Code | Severity | Message | Cause/Action |
|---|---|---|---|
| | | | 効なデバイスとして見つけられませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。次の正しい引数を指定してコマンドを再実行してください。-t <Resource Tag> および -i <Resource ID> (作成する dmmp デバイスリソースを特定する ID) |
| 128086 | ERROR | Failed to create "%s" hierarchy. | **原因**: LifeKeeper は、{device} のためのリソース階層を作成できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128088 | ERROR | Error creating resource "%s" on server "%s" | **原因**: LifeKeeper は、{server} で {tagname} を使用してリソースを作成できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128090 | ERROR | Failed to create dependency "%s"-"%s" on system "%s". | **原因**: LifeKeeper は、{system} で依存関係 {resource tag name} – {resource tag name} を作成できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |

| Code | Severity | Message | Cause/Action |
|---|---|---|---|
| 128091 | ERROR | Error creating resource "%s" on server "%s" | 原因: LifeKeeper は、{system} で {resource} を作成できませんでした。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128101 | ERROR | "%s" constructor requires a valid argument. | 原因: LifeKeeper は、dmmp リソースのオブジェクトを作成できませんでした。<br><br>処置: 次の正しい引数を指定してコマンドを再実行してください。-t <Resource Tag> および -i <Resource ID> (dmmp デバイスリソースを特定する ID) |
| 128102 | ERROR | Invalid tag "%s". | 原因: 指定されたタグ名のリソースインスタンスが見つかりませんでした。<br><br>処置: リソースが正しく設定されていることを確認してください。次の正しい引数を指定してコマンドを再実行してください。-t <Resource Tag> および -i <Resource ID> (dmmp デバイスリソースを特定する ID) |
| 128111 | ERROR | Failed to get registrations for "%s": %s. Verify the storage supports persistent reservations. | 原因: LifeKeeper は {device} の登録の取得に失敗し、メッセージ「bad field in Persistent reservation in cdb」を出力しました。<br><br>処置: このストレージでパーシステントリザベーションがサポートされているか確認してください。隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |

| Code | Severity | Message | Cause/Action |
|---|---|---|---|
| 128112 | ERROR | Failed to get registrations for "%s": %s. Verify the storage supports persistent reservations. | 原因: LifeKeeper は {device} の登録の取得に失敗し、メッセージ「illegal request」を出力しました。<br><br>処置: このストレージでパーシステントリザベーションがサポートされているか確認してください。隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128136 | ERROR | A previous quickCheck with PID "%s" running for device "%s" has been terminated. | 原因: LifeKeeper は、dmmp リソースの restore 操作時に前の quickCheck がまだ実行中であることを検出しました。LifeKeeper は quickCheck を終了しました。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |
| 128137 | ERROR | SCSI reservation conflict on %s during LifeKeeper resource initialization. Manual intervention required. | 原因: LifeKeeper は、dmmp リソースの restore 操作時に {device} で SCSI リザベーションコンフリクトを検出しました。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。手動による介入および {device} でのリザベーションコンフリクトの解消が必要です。 |
| 128138 | ERROR | unable to clear registrations on %s. | 原因: LifeKeeper は、{device} 上のすべての登録を消去できませんでした。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要がありま |

| Code | Severity | Message | Cause/Action |
|---|---|---|---|
| | | | す。 |
| 128140 | WARN | registration failed on path %s for %s. | 原因: LifeKeeper は、{device} の {path} の登録を作成できませんでした。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128143 | ERROR | reserve failed (%d) on %s. | 原因: LifeKeeper は、{device} 上の {resource} の登録を作成できませんでした。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128145 | ERROR | The server ID "%s" returned by "%s" is not valid. | 原因: LifeKeeperは有効なホスト {ID} を生成できませんでした。<br><br>処置: デバイスを登録するための ID は、クラスタ内のサーバを一意に特定する 1 ~ 12 桁の 16 進数で構成されます。隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128146 | ERROR | device failure on %s. SYSTEM HALTED. | 原因:LifeKeeper は {device} で障害を検出しました。サーバを再起動します。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |

| Code | Severity | Message | Cause/Action |
|---|---|---|---|
| 128148 | ERROR | device failure on %s. SYSTEM HALTED DISABLED. | 原因: LifeKeeper は {device} で障害を検出しました。LifeKeeper の設定によって、再起動はスキップされました。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。「SCSIHALT」設定を有効にして、いかなるデバイス障害が検出されたときも再起動が行われるようにしてください。 |
| 128149 | ERROR | device failure or SCSI Error on %s. SENDEVENT DISABLED. | 原因: LifeKeeper は {device} で障害を検出しました。LifeKeeper の設定によって、イベント生成はスキップされました。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。「SCSIEVENT」設定を有効にして、いかなるデバイス障害が検出されたときも sendevent を使用できるようにしてください。 |
| 128150 | ERROR | %s does not have EXCLUSIVE access to %s, halt system. | 原因: LifeKeeper は {server} の {device} でリザベーションコンフリクトを検出しました。サーバを再起動します。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |
| 128151 | ERROR | %s does not have EXCLUSIVE access to %s, halt system DISABLED. | 原因: LifeKeeper は {server} の {device} でリザベーションコンフリクトを検出しました。LifeKeeper の設定によって、再起動はスキップされました。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。「RESERVATIONCONFLICT」設定を有効にして、いかなるリザベーションコンフリクトが検出されたときも再起動 |

| Code | Severity | Message | Cause/Action |
|---|---|---|---|
| | | | が行われるようにしてください。 |
| 128154 | WARN | unable to flush buffers on %s. | **原因**: LifeKeeper は、dmmp リソースの削除時に {device} のバッファを消去できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128157 | WARN | %s utility not found, limited healthcheck for %s. | **原因**: LifeKeeper は、{device} の健全性チェックを行う「dd」ユーティリティを見つけられませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128160 | ERROR | %s failed to read %s. | **原因**: LifeKeeper は、{utility} の使用中に {device} のディスク I/O テストに失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128163 | ERROR | Registration ID "%s" for "%s" is not valid. | **原因**: LifeKeeper は {device} の有効な登録 {ID} の生成に失敗しました。<br><br>**処置**: デバイスを登録するための ID は、デバイスへのパスに由来する 4 桁の 16 進数で構成されます。隣接するログメッセージで詳細および関連メッセージを確 |

| Code | Severity | Message | Cause/Action |
|---|---|---|---|
| | | | 認し、レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128170 | ERROR | Usage: canextend <Template system name> <Template tag name> | |
| 128500 | ERROR | Usage error | 原因: dmmp デバイスの restore コマンドに不正な引数が指定されたため、コマンドが実行できませんでした。<br><br>処置: 次の正しい引数を指定してコマンドを再実行してください。-t <Resource Tag> および -i <Resource ID> (リストアする dmmp デバイスリソースを特定する ID) |
| 128504 | ERROR | "%s" resource type is not installed on "%s". | 原因: dmmp デバイスをサポートするための Device Mapper Multipath Recovery Kit がシステムにインストールされていません。<br><br>処置: steeleye-lkDMMP Device Mapper Multipath Recovery Kit rpm をシステムにインストールしてください。 |
| 128506 | ERROR | Usage error | 原因: dmmp デバイスの devShared コマンドに不正な引数が指定されたため、コマンドが実行できませんでした。<br><br>処置: 次の正しい引数を指定してコマンドを再実行してください。<Template Resource System Name> および <Template Resource Tag> (作成する dmmp デバイスリソースを特定するタグ) |
| 128507 | FATAL | This script must be executed on "%s". | 原因: dmmp デバイスリソースを作成するための devicehier スクリプトの引数 |

| Code | Severity | Message | Cause/Action |
|---|---|---|---|
| | | | に不正なシステム名が指定されました。<br><br>**処置**: devicehier スクリプトに正しいシステム名を指定してください。この名前は、コマンドを実行するシステムの名前に一致する必要があります。 |
| 128511 | ERROR | Failed to get the ID for the device "%s". Hierarchy create failed. | **原因**: dmmp デバイスリソースを作成するための devicehier スクリプトは、指定されたデバイスの SCSI ID を特定できませんでした。<br><br>**処置**: 指定されたデバイスパスが存在すること、およびそのパスがサポート対象の SCSI ストレージアレイのパスであることを確認してください。 |
| 128512 | ERROR | Failed to get the disk ID for the device "%s". Hierarchy create failed. | **原因**: dmmp ディスクリソースを作成するための devicehier スクリプトは、指定されたディスクの SCSI ID を特定できませんでした。<br><br>**処置**: 指定されたデバイスパスが存在すること、およびそのパスがサポート対象の SCSI ストレージアレイのパスであることを確認してください。 |
| 128513 | ERROR | Failed to create the underlying resource for device "%s". Hierarchy create failed. | **原因**: ベースとなる dmmp ディスクリソースの作成に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128515 | ERROR | Error creating resource "%s" on server "%s" | **原因**: dmmp デバイスリソースの作成に失敗しました。 |

| Code | Severity | Message | Cause/Action |
|---|---|---|---|
| | | | **処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128517 | ERROR | Failed to create dependency "%s"-"%s" on system "%s". | **原因**: dmmp デバイスリソースと dmmp ディスクリソースとの間の親 / 子の依存関係の作成に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128519 | ERROR | Error creating resource "%s" on server "%s" | **原因**: 新しく作成した dmmp デバイスリソースを in service にする操作が失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 128521 | ERROR | Either TEMPLATESYS or TEMPLATETAG argument missing | **原因**: dmmp デバイスリソースの extend コマンドに不正な引数が指定されました。<br><br>**処置**:テンプレートシステムおよびタグの正しい名前を指定して dmmp デバイスリソースの extend を再実行してください。 |
| 128540 | ERROR | Usage error | **原因**: dmmp デバイスの SCSI ID を取得する getId コマンドに不正な引数が指定されました。 |

| Code | Severity | Message | Cause/Action |
|---|---|---|---|
| | | | **処置**: 次の正しい引数を指定してコマンドを再実行してください。-i <device path> または -b <device ID>。 |
| 128541 | ERROR | Usage error | **原因**: dmmp デバイスリソースを削除するコマンドに不正な引数が指定されました。<br><br>**処置**: 次の正しい引数を指定してコマンドを再実行してください。-t <dmmp device resource tag>。 |
| 128543 | ERROR | device node "$dev" does not exist. | **原因**: dmmp デバイスリソースをリストアするために必要なデバイスノードが存在しません。リストア時に udev デバイスの作成に割り当てられている待機時間を越えました。<br><br>**処置**: udev がデバイスを作成した後に dmmp デバイスリソースの restore を再実行してください。 |
| 128544 | ERROR | Usage error | **原因**: dmmp デバイスリソースを out of service にする remove コマンドに不正な引数が指定されました。<br><br>**処置**: 次の正しい引数を指定してコマンドを再実行してください。-t <dmmp device resource tag>。 |
| 128550 | ERROR | Failed to get path information.(tag=\"$opt_t\" | |
| 128551 | ERROR | Failed to get the status of path.(tag=\"$opt_t\" | |
| 128552 | ERROR | Failed to check the status of path.(tag=\"$opt_t\" | |
| 128553 | ERROR | Failed to get the resource information.(tag=\"$opt_t\" | |

# 8.1.4. EC2 Kit メッセージカタログ

以下の Recovery Kit for EC2 メッセージカタログには、Recovery Kit for EC2 の使用中に表示される可能性のあるすべてのメッセージのリストを記載しています。

総合メッセージカタログ は、SIOS Protection Suite for Linux の使用中に発生する可能性のあるすべてのメッセージの一覧を提供し、必要に応じて、エラーの原因およびエラー状態を解消するために必要な操作に関する追加の説明も記載しています。このリストで、受信したエラーコードを検索できます。

**Control F** を使用して、各カタログの特定のエラーコードを検索します。エラーコードの検索には、画面右上の検索ボックスをご利用ください。

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 129100 | FATAL | Failed to load instance from LifeKeeper. | 原因: 無効なリソースタグまたは ID が指定されました。<br><br>処置: タグまたは ID が有効であることを確認し、コマンドを再実行してください。 |
| 129103 | FATAL | No resource matches tag "$self->{'tag'}". | 原因: 無効なリソースタグが指定されました。<br><br>処置: タグが有効であることを確認し、コマンドを再実行してください。 |
| 129104 | FATAL | An error occurred setting LifeKeeper resource information | 原因: LifeKeeper で内部エラーが発生しました。 |
| 129110 | ERROR | Could not get the Elastic Network Interface ID for $dev | 原因: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。<br><br>処置: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129111 | ERROR | Failed to get Allocation ID of Elastic IP "$elasticIp". | 原因: EC2 API の呼び出しに失敗しまし |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | た。ネットワークの問題と考えられます。<br><br>**処置**: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129113 | ERROR | Failed to get my instance ID. | **原因**: EC2 インスタンスのメタデータのアクセスに失敗しました。<br><br>**処置**: Amazon コンソールを確認し、操作を再試行してください。 |
| 129114 | ERROR | Failed to get ENI ID. | **原因**: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。<br><br>**処置**: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129116 | ERROR | Failed to associate Elastic IP "$self->{'EIP'}" on "$self->{'DEV'}". | **原因**: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。<br><br>**処置**: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129118 | WARN | $self->{'EIP'} is not associated with any instance. | **原因**: Elastic IP がインスタンスに関連付けられていません。<br><br>**処置**: LifeKeeper は Elastic IP を関連付ける EC2 API を呼び出して、問題の修正を試行します。隣接するログメッセージで詳細を確認してください。 |
| 129119 | WARN | $self->{'EIP'} is associated with another instance. | **原因**: Elastic IP が別のインスタンスに関 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 連付けられています。<br><br>**処置**: LifeKeeper は Elastic IP を関連付ける EC2 API を呼び出して、問題の修正を試行します。隣接するログメッセージで詳細を確認してください。 |
| 129120 | ERROR | Failed to recover Elastic IP. | **原因**: EC2 API が Elastic IP の関連付けに失敗しました。<br><br>**処置**: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129121 | ERROR | Recovery process ended but Elastic IP is not associated with this instance. Please check AWS console. | **原因**: EC2 API が Elastic IP の関連付けに失敗しました。<br><br>**処置**: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129122 | ERROR | Error creating resource \"$target_tag\" with return code of \"$err\". | **原因**: LifeKeeper がサーバ上にリソースインスタンスを作成できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 129123 | ERROR | Failed to get ENI ID. | **原因**: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。<br><br>**処置**: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129124 | WARN | $self->{'EIP'} is associated with another network interface. | **原因**: Elastic IP は正しいインスタンスに関連付けられていますが、ENI が正しく |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ありません。<br><br>**処置**: LifeKeeper は Elastic IP を関連付ける EC2 API を呼び出して、問題の修正を試行します。隣接するログメッセージで詳細を確認してください。 |
| 129125 | ERROR | Link check failed for interface '$dev'. | **原因**: 要求したインターフェースが「NO-CARRIER」となっています。これは、リンクが存在しないことを示します。<br><br>**処置**: ネットワークインターフェースを確認し、リンクをアップ状態にしてください。 |
| 129126 | ERROR | Link check failed for interface '$dev'. Reason: down slave. | **原因**: 要求したインターフェースが「NO-CARRIER」となっています。これは、リンクが存在しないことを示します。<br><br>**処置**: ネットワークインターフェースを確認し、リンクをアップ状態にしてください。 |
| 129129 | WARN | The link for network interface '$self->{'DEV'}' is down. Attempting to bring the link up. | **原因**: 要求したインターフェースが「NO-CARRIER」となっています。これは、リンクが存在しないことを示します。<br><br>**処置**: LifeKeeper は、リンクをアップ状態にして Elastic IP とインターフェースを関連付けることにより、問題の修正を試行します。隣接するログメッセージで詳細を確認してください。 |
| 129130 | ERROR | Failed to modify "$opt_t" to end pint URL "$endpoint". | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 129137 | ERROR | The link for network interface '$self->{'DEV'}' is still down. | **原因**: LifeKeeper がリンクをアップ状態にできませんでした。<br><br>**処置**: インターフェースが有効であり、アップ状態であることを確認してください。隣接するログメッセージで詳細を確認してください。 |
| 129139 | WARN | The link for network interface '$self->{'DEV'}' is down. | **原因**: 要求したインターフェースが「NO-CARRIER」となっています。これは、リンクが存在しないことを示します。<br><br>**処置**: ネットワークインターフェースを確認し、リンクをアップ状態にしてください。 |
| 129140 | ERROR | Could not get ENI ID for $self->{IP}. | **原因**: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。<br><br>**処置**: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129142 | ERROR | Failed to update route table | **原因**: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。<br><br>**処置**: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129143 | ERROR | You must have exactly one IP address resource as the parent of the RouteTable EC2 resource. Please reconfigure your resource hierarchy. | **原因**: ルートテーブルの EC2 リソースは親として IP リソースを 1 つのみ持つ必要があります。<br><br>**処置**: 必要に応じてリソース階層を修復してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 129144 | ERROR | $func called with invalid timeout: $timeout | **原因**: /etc/default/LifeKeeper ファイルに無効なタイムアウト値が指定されました。<br><br>**処置**: /etc/default/LifeKeeper 内にあるすべての EC2_*_TIMEOUT 設定が正しいことを確認してください。 |
| 129145 | ERROR | $func action timed out after $timeout seconds | **原因**: タイムアウト期間中に動作が完了しませんでした。<br><br>**処置**: その動作の EC2_*_TIMEOUT 値 (/etc/default/LifeKeeper 内) を増加することを検討してください。 |
| 129146 | ERROR | failed to run $func with timeout: $@ | **原因**: これは内部エラーです。 |
| 129148 | ERROR | Amazon describe-route-tables call failed (err=%s)(output=%s)<br>It is recommended to confirm the following.<br>————————<br>Verify that you're running the most recent AWS CLI version<br>-> Make sure that you're using the most recent version of the AWS CLI.<br><br>Unable to locate credentials<br>-> Verify that the AWS CLI is installed and configured correctly.<br><br>An error occurred (UnauthorizedOperation) and (AuthFailure)<br>-> Make sure that the AWS IAM role or IAM user has the correct permissions to run the relevant commands.<br>-> Make sure that the time on your Linux or Windows instance is correct.<br>The instance's UTC time should match to AWS UTC time. (The time zone setting is irrelevant.) | **原因**: AWS CLIコマンドの呼び出しに失敗しました。<br><br>**処置**: 出力されたトラブルシュート情報を参考に環境を確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | -> Make sure that you’re using the correct Amazon Simple Token Service (AWS STS) token format.<br>-> Make sure that you’re using the correct credentials to make the API call.<br>If there are multiple sets of credentials on the instance,<br>credential precedence might affect which credentials the instance uses to make the API call.<br>Verify which set of credentials you’re using by running the aws sts get-caller-identity command.<br><br>An error occurred (ExpiredToken)<br>-> Temporary credentials expire at the time interval specified during creation.<br>If the credentials for your IAM role are expired, obtain a new STS token by assuming a new IAM role.<br><br>If your error does not matches to the above errors, you can get more information by using the follow command line.<br>(It is the same with the command line which occurs the current error, and it is added ‘—debug’ option.)<br><br># [実行されたawsコマンド + オプション] —debug | |
| 129150 | ERROR | Elastic IP \"$elasticIp\" is associated with another instance. | 原因: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。<br><br>処置: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129151 | ERROR | Could not get the Association ID for Elastic IP "$elasticIp". | 原因: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられま |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | す。<br><br>**処置**: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129152 | ERROR | Failed to disassociate Elastic IP "$self->{'EIP'}" on "$self->{'DEV'}". | **原因**: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。<br><br>**処置**: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129153 | ERROR | Failed to disassociate Elastic IP "$elasticIp", (err=%s)(output=%s) | **原因**: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。<br><br>**処置**: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129154 | ERROR | Amazon describe-addresses call failed (err=%s)(output=%s) | **原因**: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。<br><br>**処置**: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129155 | ERROR | Amazon describe-address call failed (err=%s)(output=%s) | **原因**: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。<br><br>**処置**: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 129157 | ERROR | curl call failed (err=%s)(output=%s) | 原因: EC2 インスタンスのメタデータのアクセスに失敗しました。<br><br>処置: Amazon コンソールを確認し、操作を再試行してください。 |
| 129159 | ERROR | Amazon associate-address call failed (err=%s)(output=%s) | 原因: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。<br><br>処置: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129160 | ERROR | Amazon describe-addresses call failed (err=%s)(output=%s) | 原因: EC2 API の呼び出しに失敗しました。ネットワークの問題と考えられます。<br><br>処置: ネットワークと Amazon コンソールを確認し、操作を再試行してください。 |
| 129161 | ERROR | Error deleting resource "$otherTag" on "$otherSys" with return code of "$err". | 原因: LifeKeeper がサーバ上のリソースインスタンスを削除できませんでした。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 129162 | ERROR | Could not getRouteTablesByIP | |
| 129163 | ERROR | Could not getRouteTablesByIP | |
| 129164 | ERROR | [$SUBJECT event] mail returned $err | |
| 129165 | ERROR | COMMAND OUTPUT: cat /tmp/err$$ | |
| 129167 | ERROR | snmptrap returned $err for Trap 180 | |
| 129168 | ERROR | COMMAND OUTPUT: cat /tmp/err$$ | |
| 129170 | ERROR | This resource is in the old format. Please update. | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 129171 | ERROR | Error disabling sourceDest checks with return code of \"$ret\". | |
| 129172 | ERROR | Failed to lookup IP from resource $self->{'tag'}. | |
| 129173 | ERROR | Failed to lookup device GUID for IP resource $iptag. | |
| 129174 | ERROR | Could not find eni_id for device GUID $dev. | |
| 129175 | ERROR | Failed to disable sourceDestChecks. (err=%s)(output=%s | |
| 129176 | ERROR | Failed to disable sourceDestChecks for $eni_id. (err=%s)(output=%s | |
| 129177 | ERROR | Error disabling sourceDest checks with return code of \"$ret\". | |
| 129178 | ERROR | Failed to disable sourceDestChecks for $eni_id. (err=%s)(output=%s | |
| 129180 | ERROR | Elastic Network Interface $eni sourceDestChecks are enabled. | |
| 129181 | WARN | WARNING Failed to find IP resources since EC2 resource is extended first. Skipping sourceDestChecks. | |
| 129182 | WARN | WARNING SourceDestCheck was skipped because Amazon describe-network-interface-attribute call failed. Please check if ec2:DescribeNetworkInterfaceAttribute and ec2:ModifyNetworkInterfaceAttribute are granted. | |
| 129403 | ERROR | END failed create of $TAG due to a $sig signal | **原因**: 作成プロセスがシグナルによって中断されました。 |
| 129409 | ERROR | The IP resource $IP_RES is not \"ISP\". | **原因**: IP リソースが in service ではありません。<br><br>**処置**: リソースを in service にして、操作をもう一度行ってください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 129410 | ERROR | Could not find IP resource $IP_RES | **原因**: IP リソースが存在していることを確認し、操作を再試行してください。 |
| 129412 | ERROR | EC2 resource $ID is already protected | **原因**: 指定した ID を持つリソースはすでに存在しています。<br><br>**処置**: 残っている古いリソースをすべてクリーンアップしてから、新しいリソースを再作成してください。 |
| 129416 | ERROR | Error creating resource "$TAG" with return code of "$lcderror". | **原因**: LifeKeeper がリソースインスタンスを作成できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 129418 | ERROR | Dependency creation between "$IP_RES" and "$TAG" failed with return code of "$lcderror". | **原因**: LifeKeeper がリソースの依存関係を作成できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 129420 | ERROR | In-service failed for tag "$TAG". | **原因**: LifeKeeper がリソースインスタンスを in service にできませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 129423 | ERROR | Could not get ENI ID for $dev. | |
| 129425 | ERROR | Failed to update route table | |
| 129426 | ERROR | In-service (dummy) failed for tag "$TAG". | |
| 129800 | ERROR | canextend checks failed for \"$self->{'tag'}\" (err=$ret) | **原因**: ターゲットサーバで拡張前のチェックに失敗しました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
| --- | --- | --- | --- |
| | | | **処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 129801 | ERROR | canextend checks failed for "$self->{'tag'}". EC2_HOME "$self->{EC2_HOME}" does not exist on $me. | |

# 8.1.5. ファイルシステムキットメッセージカタログ

**Control F** を使用して、各カタログの特定のエラーコードを検索します。エラーコードの検索には、画面右上の検索ボックスをご利用ください。

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 124004 | FATAL | resource tag name not specified | **原因**:「quickCheck」操作で無効な引数が指定されました。<br><br>**処置**: 正しい引数が渡されたことを確認してください。 |
| 124005 | FATAL | resource id not specified | **原因**:「quickCheck」操作で無効な引数が指定されました。<br><br>**処置**: 正しい引数が渡されたことを確認してください。 |
| 124007 | FATAL | Failed to get resource information | **原因**: ファイルシステムリソースの情報フィールドに正しい情報が含まれていません。<br><br>**処置**: リソースの情報フィールドに正しい情報を入力するか、最近の「lkbackup」からシステムをリストアして元の情報フィールドをリストアします。 |
| 124008 | ERROR | getId failed | **原因**: ファイルシステムリソースは基本ディスクデバイスを見つけられませんでした。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。リソース階層が有効であること、および必要なストレージキットがすべてインストールされていることを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 124009 | ERROR | LifeKeeper protected filesystem is in service but quickCheck detects the following error: | **原因**: ファイルシステムキットはリソースに問題を発見しました。<br><br>**処置**: この直後に表示されるメッセージで詳細を確認してください。 |
| 124010 | ERROR | "$id" is not mounted | **原因**: ファイルシステムリソースがマウントされていません。<br><br>**処置**: 処置は必要ありません。ローカルリカバリでリソースの再マウントを許可してください。 |
| 124011 | ERROR | "$id" is mounted but with the incorrect mount options (current mount option list: $mntopts, expected mount option list: $infoopts) | **原因**: ファイルシステムリソースが不正にマウントされています。<br><br>**処置**: 処置は必要ありません。ローカルリカバリでリソースの再マウントを許可してください。OS アップグレードにより新しいマウントオプションが自動的に追加された場合、LifeKeeper の Change Mount Options により、ファイルシステムリソースのマウントオプションに追加する必要があるかもしれません。 |
| 124012 | ERROR | "$id" is mounted but on the wrong device (current mount device: $tmpdev, expected mount device: $dev) | **原因**: ファイルシステムリソースに誤ったデバイスがマウントされています。<br><br>**処置**: 処置は必要ありません。ローカルリカバリでリソースの再マウントを許可してください。 |
| 124015 | ERROR | LifeKeeper protected filesystem "$tag" ($id) is $percent% full ($blocksfree free blocks). | **原因**: ディスク使用量が FILESYSFULLERROR の値（%）を超過しました。<br><br>**処置**: ファイルシステムからデータを削除または移行してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 124016 | WARN | LifeKeeper protected filesystem "$tag" ($id) is $percent% full ($blocksfree free blocks). | **原因**: ディスク使用量がFILESYSFULLWARNの値（%）を超過しました。<br><br>**処置**: ファイルシステムからデータを削除または移行してください。 |
| 124020 | FATAL | cannot find device information for filesystem $id | **原因**: ファイルシステムリソースは基本ディスクデバイスを見つけられませんでした。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。リソース階層が有効であること、および必要なストレージキットがすべてインストールされていることを確認してください。 |
| 124029 | ERROR | Failed to find child resource. | **原因**: ファイルシステムリソースは基本ディスクリソースを特定できませんでした。<br><br>**処置**: リソース階層が正しいことを確認してください。 |
| 124032 | FATAL | Script has hung. Exiting. | **原因**: アンマウントする必要があるものの、マウントされているファイルシステムに開かれているファイルが複数のプロセスにありました。それらのプロセスの終了に時間がかかりすぎています。<br><br>**処置**: このエラーが続く場合は、マウントポイントを使用している可能性のあるすべてのソフトウェアを一時的に停止し、アンマウントできるようにしてください。ファイルシステムをそれでもアンマウントできない場合は、サポートに問い合わせてください。 |
| 124042 | ERROR | file system $fs failed unmount; will try again | **原因**: アンマウントする必要があるもの |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | の、マウントされているファイルシステムに開かれているファイルが複数のプロセスにありました。場合によっては、それらのプロセスを消去するために、何度か操作を実行する必要があります。<br><br>**処置**: 処置は必要ありません。プロセスをそのまま続行してください。 |
| 124046 | ERROR | file system $fsname failed unmount | **原因**: ファイルシステムをアンマウントできませんでした。<br><br>**処置**: このエラーが続く場合は、マウントポイントを使用している可能性のあるすべてのソフトウェアを一時的に停止し、アンマウントできるようにしてください。ファイルシステムをそれでもアンマウントできない場合は、サポートに問い合わせてください。 |
| 124049 | ERROR | Local recovery of resource has failed (err=$err) | **原因**: ファイルシステムリソースに、ローカルでは修正できない問題があります。<br><br>**処置**: 処置は必要ありません。リソースを別のシステムにフェイルオーバできるようにしてください。 |
| 124051 | WARN | getId failed, try count : $cnt/$try | |
| 124052 | ERROR | \"$id\" is mounted but filesystem is shutdown state. | |
| 124054 | ERROR | Failed to change mount option from \”$old_opts\” to \”$new_opts\” for migration | |
| 124103 | ERROR | $ERRMSG Script was terminated for unknown reason | **原因**: このメッセージは、通常の状態では表示されません。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 124104 | ERROR | $ERRMSG Required template machine name is null | **原因**: canextend 操作で無効な引数が指定されました。<br><br>**処置**: 引数が正しいことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124105 | ERROR | $ERRMSG Required template resource tag name is null | **原因**: canextend 操作で無効な引数が指定されました。<br><br>**処置**: 引数が正しいことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124106 | ERROR | $ERRMSG Unable to access template resource "$TemplateTagName" | **原因**: リソースの基本ディスク情報を特定できませんでした。<br><br>**処置**: テンプレートシステムで階層が正しいことを確認してから拡張してください。 |
| 124107 | ERROR | $ERRMSG Resource "$TemplateTagName" must have one and only one device resource dependency | **原因**: リソースの階層にある基本デバイスが多すぎます。<br><br>**処置**: テンプレートシステムで階層が正しいことを確認してから拡張してください。 |
| 124108 | ERROR | $ERRMSG Unable to access template resource "$TemplateTagName" | **原因**: テンプレートシステムでリソースを見つけられません。<br><br>**処置**: テンプレートシステムで階層が正しいことを確認してから拡張してください。 |
| 124109 | ERROR | $ERRMSG Can not access canextend for scsi/$DeviceResType resources on | **原因**: ターゲットシステムに、必須のコ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | machine "$TargetSysName" | ンポーネントがいくつか見つかりません。<br><br>**処置**: ターゲットシステムに正しいキットがすべてインストールされ、ライセンスされていることを確認してください。 |
| 124110 | ERROR | $ERRMSG Either filesystem "$TemplateLKId" is not mounted on "$TemplateSysName" or filesystem is not shareable with "$TargetSysName" | **原因**: ファイルシステムがテンプレートシステムのサービス内にないか、ターゲットシステムに拡張するための要件を満たしていません。<br><br>**処置**: リソースがテンプレートシステムのサービス内にあることを確認し、ファイルシステムを拡張するための要件に関する製品ドキュメントを参照してください。 |
| 124111 | ERROR | $ERRMSG File system type "${FSType}" is not supported by the kernel currently running on "${TargetSysName}" | **原因**: カーネルのサポートがないため、ファイルシステムのタイプをターゲットシステムにマウントできません。<br><br>**処置**: ターゲットシステムにそのカーネルモジュールがすべてインストールされており、正常に設定されていることを確認してから、リソースを拡張してください。 |
| 124112 | ERROR | must specify machine name containing primary hierarchy | **原因**: creFShier 操作で無効な引数が指定されました。<br><br>**処置**: 通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124113 | ERROR | must specify primary ROOT tag | **原因**: creFShier 操作で無効な引数が指定されました。<br><br>**処置**: 通常の操作でこのエラーが発生し |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | た場合は、サポートに問い合わせてください。 |
| 124114 | ERROR | must specify primary mount point | **原因**: creFShier 操作で無効な引数が指定されました。<br><br>**処置**: 通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124115 | ERROR | must specify primary switchback type | **原因**: creFShier 操作で無効な引数が指定されました。<br><br>**処置**: 通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124118 | ERROR | dep_remove failure on machine \""$PRIMACH"" for parent "$PRITAG" and child "$DEVTAG." | **原因**: 依存関係の作成後のクリーンアップに失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124119 | ERROR | ins_remove failure on machine ""$PRIMACH"" for \"$PRITAG." | **原因**: インスタンスの作成後のクリーンアップに失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124121 | ERROR | ins_remove failure on machine ""$PRIMACH" | **原因**: リソースの作成後のクリーンアップに失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124122 | ERROR | $ERRMSG Script was terminated for unknown reason | **原因**: このメッセージは、通常の状態で |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | は表示されません。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124123 | ERROR | $ERRMSG Required template machine name is null | **原因**: depstoextend 操作で無効な引数が指定されました。<br><br>**処置**: スクリプトが正常に読み出されたことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124124 | ERROR | $ERRMSG Required template resource tag name is null | **原因**: depstoextend 操作で無効な引数が指定されました。<br><br>**処置**: スクリプトが正常に読み出されたことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124125 | ERROR | $ERRMSG Unable to access template resource "$TemplateTagName" | **原因**: リソースが基本ディスクリソースを特定できませんでした。<br><br>**処置**: 階層とすべての依存関係が正しいことを確認してから拡張してください。 |
| 124126 | ERROR | unextmgr failure on machine ""$PRIMACH" | **原因**: 失敗したリソース拡張操作の後のクリーンアップに失敗しました。<br><br>**処置**: 残りのリソースを手動でクリーンアップし、隣接するログメッセージで詳細を確認してください。 |
| 124128 | ERROR | unextmgr failure on machine ""$PRIMACH"" for "$PRITAG." | **原因**: 失敗したリソース拡張操作の後のクリーンアップに失敗しました。<br><br>**処置**: 残りのリソースを手動でクリーン |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | アップし、隣接するログメッセージで詳細を確認してください。 |
| 124129 | ERROR | $ERRMSG Script was terminated for unknown reason | **原因**: このメッセージは、通常の状態では表示されません。<br><br>**処置**: 追加のログメッセージを探して、詳細を確認してください。 |
| 124130 | ERROR | $ERRMSG Required template machine name is null | **原因**: 拡張操作で無効な引数が指定されました。<br><br>**処置**: スクリプトが正常に読み出されたことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124131 | ERROR | $ERRMSG Required template resource tag name is null | **原因**: 拡張操作で無効な引数が指定されました。<br><br>**処置**: スクリプトが正常に読み出されたことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124132 | ERROR | $ERRMSG Required target mount point is null | **原因**: 拡張操作で無効な引数が指定されました。<br><br>**処置**: スクリプトが正常に読み出されたことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124133 | ERROR | $ERRMSG Unable to access template resource "$TemplateTagName" | **原因**: 拡張中のタグがテンプレートシステムに存在しません。<br><br>**処置**: テンプレートシステムで階層が正しいことを確認してから拡張してくださ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | い。 |
| 124134 | ERROR | $ERRMSG Detected conflict in expected tag name "$TargetTagName" on target machine. | **原因**: 拡張するリソースと同じタグのリソースがすでにターゲットシステムにあります。<br><br>**処置**: 競合しているリソースのどちらか一方を別のタグで再作成してください。 |
| 124135 | ERROR | $ERRMSG Resource "$TemplateTagName" does not have required device resource dependency or unable to access this resource on template machine. | **原因**: リソースまたは基本ディスクリソースがテンプレートシステムで見つかりません。<br><br>**処置**: テンプレートシステムで階層が正しいことを確認してから拡張してください。 |
| 124136 | ERROR | $ERRMSG Resource "$TemplateTagName" must have one and only one device resource dependency | **原因**: リソースは、テンプレートシステムの階層内に複数の基本デバイスを持っています。<br><br>**処置**: 拡張する前に階層が正しいことを確認し、ファイルシステムリソースが単一のディスクリソースにのみ依存していることを確認してください。 |
| 124137 | ERROR | $ERRMSG Can not access extend for scsi/$DeviceResType resources on machine "$TargetSysName" | **原因**: 所定のストレージタイプのサポートに必要なファイルがターゲットシステムにありません。<br><br>**処置**: 必要なキットがターゲットシステムにインストールされ、ライセンスされていることを確認してください。 |
| 124138 | ERROR | $ERRMSG Unable to access target device resource "$DeviceTagName" on machine "$TargetSysName&quot | **原因**: 必要な基本ディスクリソースがターゲットシステムに存在していません。<br><br>**処置**: 隣接するログメッセージで詳細を |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 確認し、ターゲットシステムが拡張対象リソースをホストするために正常に設定されていることを確認してください。 |
| 124141 | ERROR | $ERRMSG Unable to find mount point "$TemplateLKId" mode on template machine | **原因**: テンプレートシステムのマウントポイント詳細を特定できません。<br><br>**処置**: リソースが in service で、テンプレートシステムでアクセス可能であることを確認してから拡張してください。 |
| 124142 | ERROR | $ERRMSG Unable to create or access mount point "$TargetLKId" on target machine | **原因**: ターゲットシステムでマウントポイントを作成できませんでした。<br><br>**処置**: マウントポイントの親ディレクトリが存在し、ターゲットシステムでアクセス可能であることを確認してください。 |
| 124143 | ERROR | $ERRMSG Two or more conflicting entries found in /etc/fstab on "$TargetSysName&quot | **原因**: デバイスまたはマウントポイントはターゲットシステムで複数回マウントされているようです。<br><br>**処置**: マウントポイントがターゲットシステムにマウントされていないことを確認してから拡張してください。 |
| 124144 | ERROR | $ERRMSG Failed to create resource instance on "$TargetSysName" | **原因**: ターゲットシステムでのリソースの作成に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。ターゲットサーバでログを必ず確認してください。 |
| 124145 | ERROR | $ERRMSG Failed to set resource instance state for "$TargetTagName" on \"$TargetSysName" | **原因**: ターゲットシステムでソースの状態を OSU に変更できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細を |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 確認してください。 |
| 124146 | ERROR | must specify machine name containing primary hierarchy | 原因: filesyshier 操作で無効な引数が指定されました。<br><br>処置: スクリプトが正常に読み出されたことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124147 | ERROR | must specify primary mount point | 原因: filesyshier 操作で無効な引数が指定されました。<br><br>処置: スクリプトが正常に読み出されたことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124149 | ERROR | create file system hierarchy failure | 原因: リソースインスタンスを検索するプロセスが失敗しました。<br><br>処置: 隣接するログメッセージで詳細を確認してください。 |
| 124150 | ERROR | create file system hierarchy failure | 原因: システムが /etc/mtab ファイルの読み取りに失敗しました。<br><br>処置: 隣接するログメッセージで詳細を確認してください。 |
| 124152 | ERROR | create file system hierarchy failure | 原因: 基本ディスクリソースを見つけられませんでした。<br><br>処置: 隣接するログメッセージで詳細を確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 124153 | ERROR | create file system hierarchy failure | **原因**: ファイルシステムリソースの作成に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124154 | ERROR | create file system hierarchy failure | **原因**: リソースの情報フィールドを更新できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124155 | ERROR | create file system hierarchy failure | **原因**: スイッチバック戦略をリソースで設定できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124157 | ERROR | create file system hierarchy failure (conflicting entries in /etc/fstab\) | **原因**: マウントポイントを /etc/fstab ファイルから削除できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124160 | ERROR | Unknown error in script filesysins, err=$err | **原因**: このメッセージは、通常の状態では表示されません。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124161 | ERROR | create filesys instance – existid – failure | **原因**: このメッセージは、通常の状態では表示されません。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124163 | ERROR | create filesys instance – ins_list – failure | **原因**: 既存のリソースの確認に失敗しま |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | した。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124164 | ERROR | create filesys instance – newtag – failure | **原因**: システムはリソースの推奨タグの生成に失敗しました。<br><br>**処置**: 通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124168 | ERROR | create filesys instance - ins_create - failure | **原因**: ファイルシステムリソースを作成できませんした。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124169 | ERROR | filesys instance - ins_setstate - failure | **原因**:新しいファイルシステムリソースの状態を初期化できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124173 | ERROR | create filesys instance - dep_create - failure | **原因**: 基本ディスクに対するリソースの依存関係を作成できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124174 | ERROR | machine not specified | **原因**: rmenu_mp 操作で無効な引数が指定されました。<br><br>**処置**: スクリプトが正常に読み出されたことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 124175 | ERROR | mount point not specified | **原因**: rmenu_mp 操作で無効な引数が指定されました。<br><br>**処置**: スクリプトが正常に読み出されたことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124177 | ERROR | unexpected multiple matches found | **原因**: 1 つまたは複数のシステムが、複数回使用されたファイルシステムまたはマウントポイントを示しています。<br><br>**処置**: ファイルシステムデバイスとマウントポイントを確認し、ファイルシステムが 1 回だけマウントされていることを確認してください。追加のログメッセージを探して、詳細を確認してください。 |
| 124178 | ERROR | machine name not specified | **原因**: rmenump 操作で無効な引数が指定されました。<br><br>**処置**: スクリプトが正常に読み出されたことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124180 | ERROR | must specify filesystem type | **原因**: validfstype 操作で無効な引数が指定されました。<br><br>**処置**: スクリプトが正常に読み出されたことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124181 | ERROR | mount point not specified | **原因**: validmp 操作で無効な引数が指定されました。<br><br>**処置**: スクリプトが正常に読み出されたことを確認してください。通常の操作で |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | このエラーが発生した場合は、サポートに問い合わせてください。 |
| 124182 | ERROR | The mount point $MP is not an absolute path | **原因**: 絶対パスではない（「/」で始まらない）マウントポイントが指定されました。<br><br>**処置**:「/」で始まる絶対パスでマウントポイントを指定してください。 |
| 124183 | ERROR | $MP is already mounted on $MACH | **原因**: 要求されたマウントポイントはシステムですでに使用中です。<br><br>**処置**: 使用中ではないマウントポイントを指定するか、操作を再試行する前にアンマウントしてください。 |
| 124184 | ERROR | The mount point $MP is already protected by LifeKeeper on $MACH | **原因**: システムは、指定したマウントポイントをすでに保護しています。<br><br>**処置**: まだ保護されていない別のマウントポイントを選択してください。 |
| 124185 | ERROR | The mount point $MP is not a directory on $MACH | **原因**: マウントポイントが通常のファイルなどの非ディレクトリを参照しています。<br><br>**処置**: ディレクトリを参照するマウントポイントを選択してください。 |
| 124186 | ERROR | The mount point directory $MP is not empty on $MACH | **原因**: 指定したマウントポイントは空でないディレクトリを参照します。<br><br>**処置**: 空のマウントポイントを選択するか、指定したディレクトリの内容を削除してから操作を再試行してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 124187 | ERROR | server name not specified | **原因**: valuepmp 操作で無効な引数が指定されました。<br><br>**処置**: スクリプトが正常に読み出されたことを確認してください。通常の操作でこのエラーが発生した場合は、サポートに問い合わせてください。 |
| 124188 | ERROR | There are no mount points on server $MACH | **原因**: サーバにファイルシステムリソースのマウントポイントがありません。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124194 | WARN | Please correct conflicting "/etc/fstab" entries on server $UNAME for: $FSDEV, $FSNAME | **原因**: ファイルシステムリソースを削除したら、/etc/fstab の一部のエントリを手動でクリーンアップする必要があります。<br><br>**処置**: /etc/fstab ファイルを手動でクリーンアップします。 |
| 124195 | ERROR | getchildinfo found no $OKAPP child for $PTAG | **原因**: システムは子リソースを階層で見つけられませんでした。<br><br>**処置**: 隣接するログメッセージで詳細を確認し、階層が正しいことを確認してから、操作を再試行してください。 |
| 124196 | ERROR | enablequotas - quotacheck may have failed for $FS_NAME | **原因**: 割り当て操作に失敗しました。<br><br>**処置**: lifekeeper ログおよび /var/log/messages の隣接するログメッセージで詳細を確認してください。 |
| 124198 | ERROR | enablequotas - quotaon failed to turn on quotas for $FS_NAME, reason | **原因**: 割り当て操作に失敗しました。<br><br>**処置**: lifekeeper ログおよび /var/log/ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | messages の隣接するログメッセージで詳細を確認してください。 |
| 124200 | ERROR | The device node $dev was not found or did not appear in the udev create time limit of $delay seconds | **原因**: デバイスノード（/dev/...）が undev によって作成されませんでした。これは、ストレージまたはストレージへのサーバ接続の問題を示している可能性があります。<br><br>**処置**: lifekeeper ログおよび /var/log/messages の隣接するログメッセージで詳細を確認してください。 |
| 124201 | WARN | Device $device not found. Will retry wait to see if it appears. | **原因**: これは、udev がストレージのデバイスノードエントリを作成している間、通常の条件下で発生する可能性もあります。このメッセージは繰り返し表示されないはずです。<br><br>**処置**: lifekeeper ログおよび /var/log/messages の隣接するログメッセージで詳細を確認してください。 |
| 124202 | ERROR | Command "$commandwithargs" failed. Retrying …. | **原因**: 指定されたコマンドは失敗しましたが、一時的なものである可能性もあります。この障害は、通常の操作で発生しますが、継続はしないはずです。<br><br>**処置**: このメッセージが何度も表示される場合は、隣接するログメッセージで詳細を確認してください。 |
| 124204 | WARN | cannot make file system $FSNAME mount point | **原因**: マウントポイントディレクトリを作成できませんでした。<br><br>**処置**: マウントポイントを作成できることを確認してください。これは、ファイルシステム権限やマウントオプションなどによる可能性があります。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 124207 | ERROR | "fsck"ing file system $FSNAME failed, trying alternative superblock | **原因**: このメッセージは、通常のファイルシステムチェックに失敗したことを示します。このメッセージは、ext2 ファイルシステムまたは代替スーパーブロックロケーションが使用されているファイルシステムの場合は問題ないと思われます。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124209 | ERROR | "fsck"ing file system $FSNAME with alternative superblock failed | **原因**: これは、代替スーパーブロックロケーションで ext2 ファイルシステム (または代替スーパーブロックロケーションが使用されている他のファイルシステム) の確認が失敗したことを示します。<br><br>**処置**: 隣接するログメッセージで詳細を確認し、実行方法の手順を確かめてください。 |
| 124210 | WARN | POSSIBLE FILESYSTEM CORRUPTION ON $FSNAME ($FPNAME) | **原因**: ミラーソースと同期していないファイルシステムが in service になったか、フェイルオーバされました。<br><br>**処置**: 隣接するログメッセージで詳細を確認し、ファイルシステムを安全に in service にする方法を製品ドキュメントで調べてください。 |
| 124211 | ERROR | Reason for fsck failure ($retval): $ret | **原因**: このログメッセージは一連のメッセージの一部で、fsck プロセスからの実際の終了コードを提供します。<br><br>**処置**: 隣接するログメッセージで詳細を確認し、実行方法の手順を確かめてください。 |
| 124212 | ERROR | "fsck" of file system $FSNAME failed | **原因**: ファイルシステムの確認に失敗し |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ました。これは通常、ファイルシステムが破損したことによるものです。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。製品ドキュメントを参照し、可能性のあるファイルシステムの破損を処理する方法を調べてください。 |
| 124213 | WARN | POSSIBLE FILESYSTEM CORRUPTION ON $FSNAME ($FPNAME) | **原因**: システムまたはユーザが、破損している可能性のあるファイルシステムを in service にしようとしました。これは、ミラーサイトと同期していないときにファイルシステムが切り替えられたか、フェイルオーバされた場合に起こることがあります。<br><br>**処置**: 隣接するログメッセージで詳細を確認し、リソースを安全に in service にする方法を製品ドキュメントで調べてください。 |
| 124214 | ERROR | Reason for fsck failure ($retval) | **原因**: このメッセージは、ファイルシステムチェックの障害に関する前のログメッセージに続いて表示されるもので、fsck プロセスのプロセス終了コードを提供します。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124218 | ERROR | File system $FSNAME was found to be already | **原因**: このメッセージは、一連のメッセージの一部です。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124219 | ERROR | mounted after initial mount attempt failed. | **原因**: このメッセージは、一連のメッセージの一部です。通常の状態では表示さ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | れないはずですが、リソースが in service の場合は致命的なエラーの可能性もあります。<br><br>**処置**: lifekeeper ログおよび /var/log/messages の隣接するログメッセージで詳細を確認してください。 |
| 124220 | ERROR | File system $FSNAME failed to mount. | **原因**: ファイルシステムをマウントできませんでした。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124221 | WARN | Protected Filesystem $ID is full | **原因**: ファイルシステムの空き容量がありません。<br><br>**処置**: 使用していないデータをファイルシステムから削除するか、より大きなファイルシステムに移行してください。 |
| 124222 | WARN | Dependent Applications may be affected <> | **原因**: これは、あるリソースに対する操作が、リソースの階層に基づいて別のリソースにも実行される可能性が高いことを示します。<br><br>**処置**: 示されているリソースが影響を受けても許容できることを確認してから操作を続行してください。 |
| 124223 | ERROR | Put "$t" Out-Of-Service Failed By Signal | **原因**: このメッセージは、通常の状態では表示されません。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124227 | ERROR | Put "$i" Out-Of-Service Failed | **原因**: 操作に失敗しました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124230 | ERROR | Put "$t" In-Service Failed By Signal | **原因**: このメッセージは、通常の状態では表示されません。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124231 | ERROR | Put "$t" In-Service Failed | **原因**: 操作に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124234 | ERROR | Put "$t" In-Service Failed | **原因**: 操作に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 124239 | WARN | Invalid subdirectory cache of $id still exists. This may prevent $id to unmount, causing the resource remove to fail. This is most likely caused by a known issue when mounting a subdirectory of an NFSv3 export. Please see the “LifeKeeper Core – Known Issues / Restrictions” section in the LifeKeeper for Linux Technical Documentation. | **原因**: エクスポートされているディレクトリのサブディレクトリを NFSv3 でマウントする際の既知の原因によって引き起こされる可能性が高いです。<br><br>**処置**: LifeKeeper for Linux テクニカルドキュメンテーションの「LifeKeeper Core – 既知の問題と制限」セクションをご覧ください。 |

# 8.1.6. Gen/App キットメッセージカタログ

**Control F** を使用して、各カタログの特定のエラーコードを検索します。エラーコードの検索には、画面右上の検索ボックスをご利用ください。

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 126105 | ERROR | script not specified - $PTH is a directory | **原因**: 指定したスクリプトパスはディレクトリです。<br><br>**処置**: スクリプトのパスを修正してください。 |
| 126110 | ERROR | script $PTH does not exist | **原因**: 指定したスクリプトパスは存在しません。<br><br>**処置**: スクリプトのパスを修正してください。 |
| 126115 | ERROR | script $PTH is a zero length file | **原因**: 指定したスクリプトは空のファイルです。<br><br>**処置**: スクリプトのファイルパスを修正し、スクリプトの内容を確認してください。 |
| 126117 | ERROR | script $PTH is not executable | **原因**: 指定したスクリプトは実行可能ではありません。<br><br>**処置**: スクリプトのファイルパスを修正し、スクリプトファイルの内容を確認したら、正常な実行権限があることを確認してください。 |
| 126125 | ERROR | required template machine name is null | **原因**: 入力テンプレートマシン名が null です。<br><br>**処置**: 入力テンプレートマシン名を修正してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 126130 | ERROR | required template resource tag name is null | **原因**: 入力テンプレートリソース {tag} が null です。<br><br>**処置**: 入力テンプレートリソースタグ名を修正してください。 |
| 126135 | ERROR | Unable to generate a new tag | **原因**:拡張の際に、「newtag」スクリプトを使用してターゲットノードでテンプレートタグ名と同じ新規タグを生成できませんでした。タグ名はすでに存在しています。<br><br>**処置**: 同じノードで重複するタグ名を避け、ログで詳細を確認してください。 |
| 126140 | ERROR | Unable to generate a new tag | **原因**: 拡張の際に、「newtag」スクリプトを使用してターゲットノードで入力ターゲットタグ名として新規タグを生成できませんでした。タグ名はすでに存在しています。<br><br>**処置**: 同じノードで重複するタグ名を避け、ログで詳細を確認してください。 |
| 126150 | ERROR | unable to remote copy template \"$_lscript\" script file | **原因**: テンプレートスクリプトファイルのリモートコピーに失敗しました。原因としては、テンプレートノードにテンプレートスクリプトファイルがない、またはスクリプトファイルを使用できない、あるいは「lcdrcp」プロセスで何らかのトランザクション障害が発生したことなどが考えられます。<br><br>**処置**: テンプレートスクリプトの有無/使用可能性を確認し、テンプレートノードへの接続を確かめてください。また、関連するエラーをログで確認し、レポートされている問題を解決してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 126155 | ERROR | failed to create resource instance on \"$TargetSysName\ | 原因:「ins_create」によるリソースインスタンスの作成に失敗しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126160 | ERROR | failed to set resource instance state for \"$TargetTagName\" on \"$TargetSysName\ | 原因: GenApp リソース拡張の際に、「ins_setstate」によるリソースインスタンスの状態の設定に失敗しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126170 | ERROR | getlocks failure | 原因:リソース階層の作成時に管理ロックの取得に失敗しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126175 | ERROR | instance create failed | 原因:「appins」による GenApp インスタンスの作成に失敗しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126180 | ERROR | unable to set state to OSU | 原因: GenApp リソース作成の際に、「ins_setstate」によるリソースインスタンスの状態の設定に失敗しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126190 | ERROR | resource restore has failed | 原因: GenApp リソースのリストアに失 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126200 | ERROR | create application hierarchy rlslocks failure | **原因**: GenApp リソースの作成後にロックの解除に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126210 | ERROR | copy $ltype script $lscript failure | **原因**: リソース作成時に、ユーザ提供のスクリプトを適切な GenApp ディレクトリにコピーできませんでした。<br><br>**処置**: ユーザ提供のスクリプトの有無/使用可能性に加え、GenApp ディレクトリを確認してください。また、関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126215 | ERROR | no $ltype script specified | **原因**: GenApp リソース作成時にユーザ定義スクリプトが見つかりません。<br><br>**処置**: 入力アクションスクリプトを確認し、リソース作成を再実行してください。 |
| 126220 | ERROR | no machine name specified | **原因**:GenApp リソース作成時に指定のマシン名が見つかりません。マシン名の入力内容が見つからないため、指定したユーザスクリプトのコピーに失敗しました。<br><br>**処置**: マシン名の入力内容を確認し、リソース作成を再実行してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 126225 | ERROR | no resource tag specified | **原因**:リソース作成時に指定のタグ名が見つかりません。<br><br>**処置**: ソースタグ名の入力内容を確認し、リソース作成を再実行してください。 |
| 126230 | ERROR | $ERRMSG Script was terminated for unknown reason | **原因**: 不明な理由で GenApp リソースの拡張に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126235 | ERROR | $ERRMSG Required template machine name is null | **原因**: GenApp リソース拡張時にテンプレートマシン名の入力が見つかりません。<br><br>**処置**: テンプレートマシン名の入力内容を確認し、リソース拡張を再実行してください。 |
| 126240 | ERROR | $ERRMSG Required template resource tag name is null | **原因**: GenApp リソース拡張時にテンプレートリソースタグ名の入力が見つかりません。<br><br>**処置**: テンプレートリソースタグ名の入力内容を確認し、リソース拡張を再実行してください。 |
| 126245 | ERROR | $ERRMSG Can not access extend for $AppType/$ResType resources on machine \"$TargetSysName\ | **原因**: ターゲットノードでの GenApp リソース拡張の際に、「extend」スクリプトを見つけられませんでした。<br><br>**処置**:「extend」スクリプトの有無/使用可能性を確認し、GenApp リソース拡張を再実行してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 126250 | ERROR | create application failure - ins_list failed | 原因: GenApp リソース作成で「ins_list」を呼び出している際に失敗しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126255 | ERROR | create application failure - unable to generate a new tag | 原因: GenApp リソース作成で新規タグの生成に失敗しました。<br><br>処置: 同じノードで重複するタグ名の使用を避けてください。また、関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126270 | ERROR | create application failure - ins_create failed | 原因:「ins_create」による GenApp インスタンスの作成に失敗しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126275 | ERROR | create application failure - copy_actions failed | 原因:「copy_actions」によるユーザ指定テンプレートスクリプトファイルのコピーに失敗しました。<br><br>処置: テンプレートスクリプトの有無/使用可能性を確認してください。また、関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126290 | ERROR | Unable to obtain tag for resource with id $ID | 原因:リカバリ時に入力 ID による GenApp リソースタグ名のフェッチに失敗しました。<br><br>処置: 入力 ID が正しいこと、および LCD の GenApp リソースの有無/使用可能性を確認してください。また、関連するエラーをログで確認し、レポートされ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ている問題を解決してください。 |
| 126300 | ERROR | generic application recover script for $TAG was not found or was not executable | **原因**: リカバリ時に GenApp リソースのユーザ定義スクリプトの特定に失敗しました。<br><br>**処置**: ユーザ定義スクリプトの有無/使用可能性を確認し、GenApp リカバリプロセスを再実行してください。 |
| 126310 | ERROR | -t flag not specified | **原因**: GenApp リソースリストア時にリソースタグ名の入力が見つかりません。<br><br>**処置**: リソースタグ名の入力内容を確認し、リソースリストアを再実行してください。 |
| 126315 | ERROR | -i flag not specified | **原因**: GenApp リソースリストア時にリソース内部 ID の入力が見つかりません。<br><br>**処置**: リソース内部 ID の入力内容を確認し、リソースリストアを再実行してください。 |
| 126327 | ERROR | END timeout restore of \"$TAG\" (forcibly terminating) | |
| 126335 | ERROR | restore script \"$LCDAS/$APP_RESTOREDIR/$TAG\" was not found or is not executable | **原因**: リストア時に GenApp リソースのユーザ定義スクリプトの特定に失敗しました。<br><br>**処置**: ユーザ定義スクリプトの有無/使用可能性を確認し、GenApp リストアプロセスを再実行してください。 |
| 126340 | ERROR | -t flag not specified | **原因**: GenApp リソース削除時にリソースタグ名の入力が見つかりません。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: リソースタグ名の入力内容を確認し、リソース削除を再実行してください。 |
| 126345 | ERROR | -i flag not specified | **原因**: GenApp リソース削除時にリソース内部 ID の入力が見つかりません。<br><br>**処置**: リソース内部 ID の入力内容を確認し、リソース削除を再実行してください。 |
| 126357 | ERROR | END timeout remove of \"$TAG\" (forcibly terminating) | |
| 126365 | ERROR | remove script \"$LCDAS/$APP_REMOVEDIR/$TAG\" was not found or was not executable | **原因**: 削除時に GenApp リソースのユーザ定義スクリプトの特定に失敗しました。<br><br>**処置**: ユーザ定義スクリプトの有無/使用可能性を確認し、GenApp 削除プロセスを再実行してください。 |
| 126375 | ERROR | Script has hung checking \"$tag\". Forcibly terminating. | **原因**: 待機時間がユーザ定義のタイムアウトを超えたので、タグ名 {tag} の GenApp リソースの「quickCheck」スクリプトが強制的に終了しました。<br><br>**処置**: GenApp リソースのパフォーマンスを確認し、quickChecking を再起動してください。また、関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126380 | ERROR | Usage error: no tag specified | **原因**: GenApp リソース quickCheck 時にリソースタグ名の入力が見つかりません。<br><br>**処置**: リソースタグ名の入力内容を確認し、リソース quickCheck を再試行して |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ください。 |
| 126385 | ERROR | Internal error: ins_list failed on $tag. | 原因: quickCheck プロセスで、「ins_list」による入力タグ名別の GenApp リソース情報のフェッチに失敗しました。<br><br>処置: 入力タグ名を修正し、quickCheck プロセスを再実行してください。また、関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126390 | FATAL | Failed to fork process to execute $userscript: $! | 原因: GenApp リソースの「quickCheck」プロセスで、ユーザ定義の「quickCheck」スクリプトを実行するためにプロセスを分岐させられませんでした。<br><br>処置: ユーザ定義の「quickCheck」スクリプトの有無/使用可能性を確認し、「quickCheck」プロセスを再実行してください。 |
| 126391 | ERROR | quickCheck has failed for \"$tag\". Starting recovery. | 原因: タグ名 {tag} の GenApp リソースが失敗することが、ユーザ定義の健全性監視スクリプトにより判定されました。「quickCheck」およびリカバリプロセスが開始されます。<br><br>処置: ローカルリカバリが完了したら、GenApp リソースのパフォーマンスを確認してください。また、関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 126392 | WARN | ${convtag}_TIMEOUT: This parameter is old. This parameter will not be supported soon. | |
| 126400 | ERROR | -t flag not specified | 原因:GenApp リソース削除プロセスで |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | リソースタグ名の入力が見つかりません。<br><br>**処置**: リソースタグ名の入力内容を確認し、リソース削除プロセスを再実行してください。 |
| 126405 | ERROR | -i flag not specified | <br>**原因**: GenApp リソース削除プロセスでリソース内部 ID の入力が見つかりません。<br><br>**処置**: リソース内部 ID の入力内容を確認し、リソース削除プロセスを再実行してください。 |
| 126478 | ERROR | Failed to create tag \'$new_leaf\'. | <br>**原因**:「creapphier」ユーティリティは、指定されたタグ {new_leaf} を作成できませんでした。<br><br>**処置**: /var/log/lifekeeper.log で「creapphier」からの追加メッセージを確認してください。 |
| 126479 | ERROR | Failed to extend tag \'$new_leaf\'. | <br>**処置**: /var/log/lifekeeper.log で拡張マネージャーからの追加エラーを確認してください。 |
| 126481 | ERROR | Failed to create dependency on \'$sys\' for \'$new_leaf\' to \'$hier{$leaf}{$sys}{'Tag'}\'. | <br>**処置**: /var/log/lifekeeper.log で「dep_create」関数のエラーを確認してください。 |
| 126484 | ERROR | Tag \'$root_tag\' is not in-service. | <br>**原因**: 指定されたタグ{root_tag} は、クラスター内のどのノードでも In Service になっていません。<br><br>**処置**: 指定されたタグ{root_tag} をクラスター内の任意のノードで In Service にし、「create_terminal_leaf」を再実行し |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | てください。 |
| 126485 | ERROR | Tag \'$root_tag_1\' was not found, select the root tag for a hierarchy to add a terminal leaf resource. | **原因**:「create_terminal_leaf」に渡された最初のタグ {root_tag_1} が、ユーティリティが実行されたシステムで見つかりませんでした。<br><br>**処置**: リソースがクラスタ内のノードで In-service であり、すべてのノードへリソースが完全に拡張されていることを確認してください。 |
| 126486 | ERROR | Unable to create leaf tag from \'$tag\'. | **原因**: 一意の端末リーフタグを作成できませんでした。100回試行した後、一意の端末リーフタグを作成できませんでした。<br><br>**処置**: 複数のリーフタグと、問題を暗示している可能性のある /var/log/lifekeeper.log のエラーを確認してください。 |
| 126487 | ERROR | Tag \'$root_tag_2\' was not found, select the root tag for a hierarchy to add a terminal leaf resource. | **原因**:「create_terminal_leaf」に渡された2番目のタグ {root_tag_2} が、ユーティリティが実行されたシステムで見つかりませんでした。<br><br>**処置**: リソースがクラスタ内のノードで In-service であり、すべてのノードへリソースが完全に拡張されていることを確認してください。 |
| 126488 | ERROR | Tag \'$root_tag_1' is not extended to 3 or more systems. | **原因**: 指定されたタグ {root_tag_1} は、3つ以上のノードに拡張されていません。<br><br>**処置**: 指定したタグ {root_tag_1} を最低でも3つのノードに拡張し、「create_terminal_leaf」を再度実行して |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ください。 |
| 126489 | ERROR | $cmd does not support SDRS resources. | **原因**: マルチサイト構成が検知されました。<br><br>**処置**: ありません |
| 126492 | ERROR | Remove resource $tag failed. | |
| 126494 | ERROR | Delete dependency failed on \'$sys\' for \'$tag\' to \'$parent\'. | |
| 126495 | ERROR | New tag \'$new_tag\' was modified during create, expected \'$new_leaf\'. | |
| 126496 | ERROR | Tag \'$root_tag_1\' is not in-service. | |
| 126497 | ERROR | Tag \'$root_tag_2\' is not in-service. | |
| 126498 | ERROR | Tag \'$root_tag_1\' and \'$root_tag_2' are not extended to same systems. | |

# 8.1.7. IP Kit メッセージカタログ

**Control F** を使用して、各カタログの特定のエラーコードを検索します。エラーコードの検索には、画面右上の検索ボックスをご利用ください。

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 123006 | FATAL | Unknown version %s of IP address | 原因: IP アドレスが IPv4 でも IPv6 でも有効ではないようです。<br><br>処置: 有効な IP アドレスを指定してください。 |
| 123008 | ERROR | No pinglist found for %s. | 原因: この IP アドレスの pinglist を開こうとして問題が発生しました。<br><br>処置: この IP アドレスの pinglist を指定していることを確認してください。 |
| 123009 | ERROR | List ping test failed for virtual IP %s | 原因: ping リストのどのアドレスからも応答を受信しませんでした。<br><br>処置: このノードのネットワーク接続および ping リストの IP が存在するシステムを確認してください。 |
| 123013 | ERROR | Link check failed for virtual IP %s on interface %s. | 原因: 要求したインターフェースが「NO-CARRIER」となっています。これは、物理レイヤー接続にリンクが存在しないことを示します。<br><br>処置: インターフェースの物理接続を確認し、物理レイヤーリンクを有効にしてください。 |
| 123015 | ERROR | Link check failed for virtual IP %s on interface %s. | 原因: 要求したインターフェースは bonding インターフェースで、スレーブのいずれか 1 つが「NO-CARRIER」です。これは、物理レイヤー接続にリンク |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | が存在しないことを示します。<br><br>**処置**: スレーブインターフェースの物理接続を確認し、物理レイヤーリンクを有効にしてください。 |
| 123024 | ERROR | IP address seems to still exist somewhere else. | **原因**: IP アドレスはネットワークの別の場所で使用中のようです。<br><br>**処置**: 別の IP アドレスを選択して使用するか、この IP アドレスを特定して、現在の使用を無効にしてください。 |
| 123037 | ERROR | must specify machine name containing primary hierarchy | **原因**: creIPhier に指定した引数が不十分です。<br><br>**処置**: 必要な引数をすべて creIPhier に指定してください。 |
| 123038 | ERROR | must specify IP resource name | **原因**: creIPhier に渡された引数が不十分です。<br><br>**処置**: 必要な引数をすべて creIPhier に指定してください。 |
| 123039 | ERROR | must specify primary IP Resource tag | **原因**:「creIPhier」コマンドにプライマリ IP リソースタグの引数が指定されていません。<br><br>**処置**: 必要な引数をすべて指定してください。 |
| 123042 | ERROR | An unknown error has occurred in utility validmask on machine %s. | **原因**:「validmask」ユーティリティの実行中に予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 123045 | ERROR | An unknown error has occurred in utility getlocks. | 原因:「getlocks」ユーティリティの実行中に予期しないエラーが発生しました。<br><br>処置: 隣接するログメッセージで詳細を確認してください。 |
| 123053 | ERROR | Cannot resolve hostname %s | 原因: IP アドレスに対してホスト名が指定されましたが、システムはホスト名を IP アドレスに解決できませんでした。<br><br>処置: ホスト名が正確かどうかを確認してから、名前の解決（DIS または /etc/hosts）が正常に機能しており、ホスト名の IP を返すことを確認してください。 |
| 123055 | ERROR | An unknown error has occurred in utility %s on machine %s. | 原因: IP リソースの作成中に障害が発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 123056 | ERROR | create ip hierarchy failure: perform_action failed | 原因:作成中に IP アドレスをリストアしようとして、予期しないエラーが発生しました。<br><br>処置: 隣接するログメッセージで詳細を確認してください。 |
| 123059 | ERROR | Resource already exists on machine %s | 原因: すでに存在する IP アドレスを作成しようとしました。<br><br>処置: 既存のリソースを再利用するか、すでに存在している IP アドレスを手動で削除するか、別の IP アドレスを使用してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 123060 | ERROR | ins_create failed on machine %s | **原因**: IP リソースの作成中に予期しない障害が発生しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認してください。 |
| 123064 | ERROR | An unknown error has occurred in utility %s on machine %s. | **原因**: IP リソースの依存関係を作成中に障害が発生しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 123066 | ERROR | An error occurred during creation of LifeKeeper application=comm on %s. | **原因**:「app_create」の呼び出し中に障害が発生しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 123068 | ERROR | An error occurred during creation of LifeKeeper resource type=ip on %s. | **原因**:「typ_create」の呼び出し中に障害が発生しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 123089 | ERROR | Link check failed for virtual IP %s on interface %s. | |
| 123091 | ERROR | the link for interface %s is down | **原因**: 要求したインターフェースが「NO-CARRIER」となっています。これは、物理レイヤー接続にリンクが存在しないことを示します。<br><br>**処置**: インターフェースの物理接続を確認し、物理レイヤーリンクを有効にしてください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 123093 | ERROR | the ping list check failed | **原因**: ping リストのどのアドレスからも応答を受信しませんでした。<br><br>**処置**: このノードのネットワーク接続および ping リストの IP が存在するシステムを確認してください。 |
| 123095 | ERROR | broadcast ping failed | **原因**: ブロードキャスト ping から応答を受信しませんでした。<br><br>**処置**: サブネット上の 1 つ以上のホストがブロードキャスト ping に応答することを確認してください。仮想 IP が正しいネットワークインターフェースにあることを確認してください。ブロードキャスト ping ではなく、pinglist の使用を検討してください。 |
| 123096 | ERROR | $msg | **原因**: 仮想 IP の実行可能性の判定に使用するブロードキャスト ping に失敗しました。<br><br>**処置**: このリソースの ping リストがプロパティパネルで正常に設定されていることを確認するか、/etc/default/LifeKeeper 設定ファイルに NOBCASTPING=1 を追加してブロードキャスト ping のチェックを確実に無効にしてください。 |
| 123097 | ERROR | exec_list_ping(): broadcast ping failed. | **原因**: 仮想 IP の実行可能性の判定に使用するブロードキャスト ping に失敗しました。<br><br>**処置**: このリソースの ping リストがプロパティパネルで正常に設定されていることを確認するか、/etc/default/LifeKeeper 設定ファイルに NOBCASTPING=1 を追加してブロードキャスト ping のチェックを確実に無効 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | にしてください。 |
| 123299 | ERROR | Unable to open %s. Reason %s | |
| 123410 | ERROR | Usage error OSUquickCheck | |
| 123411 | ERROR | OSUquickCheck: both tag and id name not specified | |
| 123412 | ERROR | resource $Tag not found on local server | |
| 123414 | ERROR | The link for network interface $IPObj->{'device'} is down | |
| 123415 | ERROR | No pinglist found for $IPObj->{'ipaddr'} | |
| 123416 | ERROR | List ping test failed for virtual IP $IPObj->{'ipaddr'} | |

# 8.1.8. LB Health Check メッセージカタログ

Control F を使用して、各カタログの特定のエラーコードを検索します。エラーコードの検索には、画面右上の検索ボックスをご利用ください。

LifeKeeper for Linux の使用中に発生する可能性のあるすべてのメッセージの一覧は 総合メッセージカタログ をご覧ください。

| Code | Severity | Message | Cause/Action |
|---|---|---|---|
| 140001 | FATAL | cmd:$tag:Insufficient input parameters. \n%s | |
| 140002 | FATAL | cmd:$tag:END failed create of %s on system %s with signal $sig. | |
| 140005 | FATAL | cmd:$tag:Resource already exists on machine %s | |
| 140006 | FATAL | cmd:$tag:ins_create failed on machine %s | |
| 140007 | FATAL | cmd:$tag:Failed restore of tag %s on system %s | |
| 140020 | FATAL | cmd::Insufficient input parameters. \n%s | |
| 140021 | ERROR | cmd:$template_tag:Template resource \"$template_tag\" on server \"$template_sys\" does not exist | |
| 140030 | FATAL | cmd:$tag:Insufficient input parameters. \n%s | |
| 140031 | FATAL | cmd:$tag:END failed extend of "%s" due to a \"%s\" signal. | |
| 140033 | ERROR | cmd:$template_tag:Template resource \"$template_tag\" on server \"$template_sys\" does not exist | |
| 140036 | FATAL | cmd:$tag:Resource already exists on machine %s | |
| 140037 | FATAL | cmd:$tag:ins_create failed on machine %s | |
| 140050 | FATAL | cmd::Port number cannot be empty. Specify the TCP/IP port number (any value from 1024 to 65535 | |
| 140051 | FATAL | cmd::Specify the TCP/IP port number (any value from 1024 to 65535 | |

| Code | Severity | Message | Cause/Action |
|---|---|---|---|
| 140060 | FATAL | cmd::$usage | |
| 140061 | FATAL | cmd::resource \"$tag\" on server \"$lbhc::cmd\" does not exist | |
| 140062 | FATAL | cmd::LK::ins_setinfo failed | |
| 140070 | FATAL | $usage | |
| 140071 | FATAL | Port number is not valid. | |
| 140073 | FATAL | Error getting resource information for $tag on server $lbhc::me. | |
| 140200 | FATAL | Unable to create LBHC object ($sys, $tag). | |
| 140201 | INFO | Daemon already started. | restore 時に既にデーモンが起動していることを示しています。 |
| 140202 | ERROR | Daemon already started but the pidfile does not match. | restore 時に既にデーモンが起動していることを示していますが、pidfile がマッチしてません。起動中のデーモンを停止してから restore を試みます。 |
| 140203 | INFO | Health check received. | restore時に、ロードバランサーからのプローブを受信したことを示します。 |
| 140205 | INFO | Daemon not started, the pidfile ($pidfile) is already removed. | remove 時にデーモンがすでに起動していません。pidfile も削除されています 。 |
| 140206 | INFO | Daemon not started. | remove 時にデーモンがすでに起動していません。 |
| 140207 | ERROR | Error deleting resource \"$otherTag\" on \"$otherSys\" with return code of \"$err\". | |
| 140208 | WARN | Resource out of service | quickCheckが実行されていますが、リソースは起動中（ISP）ではありませんでした。 |
| 140209 | ERROR | Daemon not running. | quickCheck でデーモンが停止しています。ローカルリカバリーを試みます。 |
| 140210 | ERROR | Daemon is running but the pidfile does not match. | quickCheck でデーモンが起動していますが、pidfile がマッチしてません。ローカルリカバリーを試みます。 |
| 140211 | ERROR | flg $id-hc-hg is on | quickCheck で $id-hc-hg フラグを検出しました。フェイルオーバーを試みます。 |
| 140250 | FATAL | Socket open failed. Check if the port \"$port\" is in use by ‘ss -anp’. | ソケットをオープンできませんでした。ポートが使用されているかどうか ‘ss |

| Code | Severity | Message | Cause/Action |
|---|---|---|---|
| | | | -anp' で確認し、使用中のプロセスを kill してください。 |
| 140251 | ERROR | Health probe not received. Check that the probe from the load balancer is set correctly, including the firewall. | ロードバランサーからのプローブを $HC_TIMEOUT 秒以内に受信しませんでした 。ロードバランサーからのプローブが正しく設定されているか、ファイアーウォールも含めて確認してください。 |
| 140252 | FATAL | Could not create the flag by \"$cmdline\". | |
| 140253 | ERROR | Failed to accept the health probe:$reason | |
| 140254 | FATAL | $errmsg | |
| 140256 | FATAL | Failed to save pid into $lbhc::LKROOT/config/$id.pid. | |
| 140257 | FATAL | Resource already deleted | |
| 140258 | FATAL | kill $pid failed with ret=$ret, kill the process \"$pid\" manually. | |
| 140320 | FATAL | $usage | |
| 140321 | FATAL | Error getting resource information for $tag on server $lbhc::me. | |
| 140330 | FATAL | $usage | |
| 140331 | FATAL | Error getting resource information for $tag on server $lbhc::me. | |
| 140340 | FATAL | $usage | |
| 140341 | FATAL | Error getting resource information for $tag on server $lbhc::me. | |
| 140350 | FATAL | $usage | |
| 140351 | FATAL | Error getting resource information for $tag on server $lbhc::me. | |
| 140360 | FATAL | $usage_l | |
| 140361 | FATAL | invalid ID \"$id\" | |
| 140362 | FATAL | $usage_w | |
| 140363 | FATAL | Error getting resource information for $tag on server $lbhc::me. | |

# 8.1.9. Recovery Kit for Oracle Cloud Infrastructure メッセージカタログ

Recovery Kit for Oracle Cloud Infrastructure 固有のメッセージ一覧と、エラーの原因およびエラー状態を解消するために必要な操作を記載します。

SIOS Protection Suite for Linux の使用中に発生する可能性のあるすべてのメッセージの一覧は 総合メッセージカタログ をご覧ください。

| コード | Severity | メッセージ | 原因/対応 |
|---|---|---|---|
| 139000 | ERROR | Cannot find the "oci" command in directories of the PATH. Please confirm that it is installed and the PATH is set correctly. | **原因**: oci コマンドの PATH が取得できません。<br><br>**対応**: oci コマンドがインストールされていること、oci コマンドのインストールディレクトリが PATH に追加されていることを確認してください。 |
| 139012 | ERROR | \"oci\" command(request-id:%s) failed with code %s, message \"%s\" and status %s | **原因**: oci コマンドの実行に失敗しました。<br><br>**対応**: ステータスコードやメッセージ内容を元に oci コマンドが正常終了するように設定してください。 |
| 139013 | ERROR | Failed to assign $ip, error:\"$result\" | **原因**: $ip の割り当てに失敗しました。<br><br>**対応**: $result の内容を元にエラーの原因を解決してください。 |
| 139022 | ERROR | \"oci\" command(request-id:%s) failed with code %s, message \"%s\" and status %s | **原因**: oci コマンドの実行に失敗しました。<br><br>**対応**: ステータスコードやメッセージ内容を元に oci コマンドが正常終了するように設定してください。 |
| 139023 | ERROR | Failed to unassign $ip, error:\"$result\" | **原因**: $ip の割り当て解除に失敗しまし |

| コード | Severity | メッセージ | 原因/対応 |
|---|---|---|---|
| | | | た。<br><br>**対応**: $result の内容を元にエラーの原因を解決してください。remove 処理が実行されるより前に $ip の割り当てが解除されていた場合は、リソースの状態が OSU であることと、OCI コンソール上で $ip が対象の VNIC に割り合てられていないことを確認してください。 |
| 139032 | ERROR | \"oci\" command(request-id:%s) failed with code %s, message \"%s\" and status %s | **原因**: `oci` コマンドの実行に失敗しました。<br><br>**対応**: ステータスコードやメッセージ内容を元に `oci` コマンドが正常終了するように設定してください。 |
| 139033 | ERROR | Failed to \"$ocicmdstr\", unknown error:\"$ip_list\" | **原因**: コード 139032 以外の原因で `oci` コマンドが異常終了しました。<br><br>**対応**: unknown error:"$ip_list" の内容を元に対応してください。 |
| 139034 | ERROR | There is no secondary IPs on $device. | **原因**: $device にセカンダリIPアドレスが割り当てられていません。<br><br>**対応**: ローカルリカバリーが有効の場合は、ローカルリカバリーが実行されたこと、それによって障害から復旧したことを確認してください。 |
| 139035 | ERROR | $ip is not assigned to $device. | **原因**: $ip が $device に割り当てられていません。<br><br>**対応**: ローカルリカバリーが有効の場合は、ローカルリカバリーが実行されたこと、それによって障害から復旧したことを確認してください。 |

| コード | **Severity** | メッセージ | 原因/対応 |
|---|---|---|---|
| 139042 | ERROR | \"oci\" command(request-id:%s) failed with code %s, message \"%s\" and status %s | **原因**: `oci` コマンドの実行に失敗しました。<br><br>**対応**: ステータスコードやメッセージ内容を元に `oci` コマンドが正常終了するように設定してください。 |
| 139043 | ERROR | Failed to assign $ip, error:\"$result\" | **原因**: $ip の割り当てに失敗しました。<br><br>**対応**: $result の内容を元にエラーの原因を解決してください。 |
| 139060 | ERROR | $cmd is invalid. | **原因**: $cmd が不正です。<br><br>**対応**: お問い合わせください。 |
| 139061 | ERROR | OCIVIP does not support IPv6. | **原因**: OCIVIP リソースは IPv6 での利用をサポートしません。<br><br>**対応**: IPv4 をご利用ください。 |
| 139062 | ERROR | $cmd is invalid. | **原因**: $cmd が不正です。<br><br>**対応**: お問い合わせください。 |
| 139063 | ERROR | IPv$ipversion is unknown version. | **原因**: IP アドレスのバージョンが不正です。<br><br>**対応**: IPv4 をご利用ください。 |
| 139070 | ERROR | Failed to access the $IMDS_URL with the curl command, status code is $status_code. | **原因**: $IMDS_UDL への curl が失敗しました。<br><br>**対応**: $IMDS_URL への curl が正常終了することを確認してください。 |

| コード | **Severity** | メッセージ | 原因/対応 |
|---|---|---|---|
| 139071 | ERROR | Failed to decode JSON, error:\"$e\". | **原因**: JSON のデコードに失敗しました。<br><br>**対応**: Instance Meta Data Serviceから取得した結果が JSON 形式であることを確認してください。 |
| 139073 | ERROR | Cannot find the vnicId of $device. | **原因**: $device に対応する vnicId が取得できませんでした。<br><br>**対応**: Instance Meta Data Service の JSON レコードに vnicId が存在するかどうか確認してください。 |
| 139074 | ERROR | Cannot find the subnetcidr of $device. | **原因**: $device に対応する subnetCidrBlock が取得できませんでした。<br><br>**対応**: Instance Meta Data Service の JSON レコードに subnetcidr が存在するかどうか確認してください。 |
| 139075 | ERROR | Cannot find the vRouterIp of $device. | **原因**: $device に対応する virtualRouterIp が取得できませんでした。<br><br>**対応**: Instance Meta Data Service の JSON レコードに virtualRouterIp が存在するかどうか確認してください。 |
| 139080 | ERROR | \"oci\" command(request-id:%s) failed with code %s, message \"%s\" and status %s | **原因**: `oci` コマンドの実行に失敗しました。<br><br>**対応**: ステータスコードやメッセージ内容を元に `oci` コマンドが正常終了するように設定してください。 |
| 139081 | ERROR | Failed to get subnet id, error:\"$subnetid\" | **原因**: subnet id の取得に失敗しました。 |

| コード | Severity | メッセージ | 原因/対応 |
|---|---|---|---|
| | | | **対応**: 以下のコマンドで正しく情報を取得できるかどうか確認してください。$vnicid にはノードに割り当てられている各 VNIC の VNIC-ID(OCID) が該当します。<br>`oci network vnic get --vnic-id $vnicid --raw-output --query 'data."subnet-id"'` |
| 139100 | ERROR | IP address is not specified. | **原因**: IP アドレスが指定されていません。<br><br>**対応**: IP アドレスが指定されていることを確認してください。 |
| 139101 | ERROR | Device is not specified. | **原因**: デバイスが指定されていません。<br><br>**対応**: デバイスが指定されていることを確認してください。 |
| 139105 | ERROR | Cannot bring OCIVIP resource $Tag in service on server $SysName. | **原因**: $SysName 上の OCIVIP リソース $Tag の restore に失敗しました。<br><br>**対応**: restore 処理に関するログを参照の上ご対応ください。 |
| 139111 | ERROR | IP-$ip already exists, and device and netmask are inconsistent with those of $ocivipTag. | **原因**: OCIVIP リソースの IP アドレスに対応する IP リソースが存在していますが、デバイス情報やネットマスクの情報に矛盾があります。<br><br>**対応**: 既存の IP リソースを OCIVIP リソースと同一の設定になるように設定を変更し、 OCIVIP リソースと依存関係を作成してください。 |
| 139112 | ERROR | Cannot bring IP resource $iptag in service on server $SysName. | **原因**: $SysName 上の OCIVIP リソース |

| コード | **Severity** | メッセージ | 原因/対応 |
|---|---|---|---|
| | | | に対応する IP リソース $iptag の restore に失敗しました。<br><br>**対応**: IPリソースの設定を見直し、 IP リソースの restore が成功することを確認してください。そのうえで OCIVIP リソースと依存関係を作成してください。 |
| 139200 | ERROR | A resource with the specified tag name "%s" already exists on the target machine "%s". | **原因**: 拡張先のノードに、指定したタグ名のリソースが既に存在します。<br><br>**対応**: 別のタグ名をご利用いただくか、拡張先ノードの対象タグ名のリソースを削除してください。 |
| 139201 | ERROR | Template resource "%s" does not exist on the server "%s". | **原因**: 拡張元として指定したリソースがノード上に存在しません。<br><br>**対応**: タグ名を正しく指定してください。 |
| 139250 | ERROR | Failed to valuenetOnOCI, error: \"$stderr_log\". | **原因**: `valuentOnOCI` に失敗しました。<br><br>**対応**: $stderr_log の箇所に実際のエラー内容が出力されています。その内容を元にご対応ください。 |

# 8.1.10. Oracle キットメッセージカタログ

**Control F** を使用して、各カタログの特定のエラーコードを検索します。エラーコードの検索には、画面右上の検索ボックスをご利用ください。

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122500 | ERROR | Usage: %s %s | **原因**: 作成操作で無効なパラメーターが指定されました。<br><br>**処置**: パラメーターが正しいことを確認し、操作を再実行してください。 |
| 122501 | ERROR | DB instance "%s" is already protected on "%s". | **原因**: {server} で LifeKeeper によってすでに保護されている Oracle データベースインスタンス {sid} を保護しようとしました。<br><br>**処置**: 別のデータベースインスタンス {sid} を LifeKeeper の保護用に選択する必要があります。 |
| 122502 | ERROR | Failed to create object instance for Oracle on "%s". | **原因**: 保護されている Oracle インスタンスの内部表現を作成中に予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122503 | ERROR | Unable to locate the oratab file "%s" on "%s". | **原因**: {server} のデフォルトまたは代替ロケーションで oratab ファイルが見つかりませんでした。<br><br>**処置**: oratab ファイルが存在しており、Oracle ユーザの適切な権限を持っていることを確認してください。「create」操作を完了するには有効な oratab ファイルが必要です。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122504 | ERROR | Unable to determine Oracle user for "%s" on "%s". | 原因: Oracle Application Recovery Kit は Oracle データベースインストールバイナリの所有権を特定できませんでした。<br><br>処置: Oracle バイナリの所有者は {server} の有効な非 root ユーザである必要があります。Oracle データベース所有権の権限と所有権を修正し、操作を再試行してください。 |
| 122505 | ERROR | The Oracle database "%s" is not running or no open connections are available on "%s". | 原因: データベースインスタンス {sid} が実行されていません。またはデータベースへの接続が、指定された認証情報では使用できませんでした。<br><br>処置:「create」操作を完了するには、データベースインスタンス {sid} を {server} で起動し、正しい認証情報を指定する必要があります。 |
| 122506 | ERROR | Unable to determine Oracle dbspaces and logfiles for "%s" on "%s". | 原因: 必要なテーブルスペース、ログファイル、関連データベースファイルの場所を特定するクエリーが失敗しました。これは、内部データベースエラーが原因の可能性もあります。<br><br>処置: 隣接するログメッセージおよび関連エラーで詳細を確認してください。Oracle ログ (alert.log) および関連トレースログ (*.trc) で詳細を確認し、レポートされている問題を修正してください。 |
| 122507 | ERROR | Unknown chunk type found for "%s" on "%s". | 原因: 指定したテーブルスペース、ログファイル、または必要なデータベースファイルが LifeKeeper でサポートされているファイルまたはキャラクタデバイスタイプではありません。<br><br>処置: 指定したファイル {database_file} |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | は既存のキャラクタデバイスまたはファイルを参照する必要があります。Oracle インストールドキュメントを参照し、指定したファイル {database_file} を、サポートされているファイルまたはキャラクタデバイスタイプとして再作成してください。 |
| 122508 | ERROR | DB Chunk "%s" for "%s" on "%s" does not reside on a shared file system. | **原因**: 指定したテーブルスペース、ログファイル、またはその他の必要なデータベースファイル {database_file} は、クラスタ内の他のシステムと共有されているファイルシステムに存在します。<br><br>**処置**: コミュニケーションパスが正常に作成されたことを確認するには、LifeKeeper UI または「lcdstatus (1M)」を使用してください。ストレージ保護に必要なアプリケーションリカバリキットがインストールされたことを確認するには、「rpm」を使用してください。ファイルが実際には共有ストレージにないことを確認し、ない場合は、共有ストレージデバイスに移動してください。 |
| 122510 | ERROR | File system create failed for "%s" on "%s". Reason: | **原因**: LifeKeeper は、指定したサーバ {server} でリソース {filesystem} を作成できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから「create」操作を再試行する必要があります。 |
| 122511 | ERROR | %s | **原因**: メッセージに「filesyshier」コマンドの結果が含まれています。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポート |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | されているエラーを修正してください。 |
| 122513 | ERROR | Dependency creation between Oracle database "%s (%s)" and the dependent resource "%s" on "%s" failed. Reason: | **原因**: LifeKeeper は、データベースリソース {tag} と必要な子リソース {childtag} との間に依存関係を作成できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。問題が修正されている場合は、{tag} と {childtag} との間の依存関係を手動で作成できる可能性があります。 |
| 122514 | ERROR | Unable to "%s" on "%s" during resource create. | **原因**: Oracle Application Recovery Kit は、「rlslocks」コマンドで管理ロックを解除できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |
| 122516 | ERROR | Raw device resource created failed for "%s" on "%s". Reason: | **原因**: LifeKeeper は、指定したサーバ {server} でリソース {raw device} を作成できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから「create」操作を再試行する必要があります。 |
| 122519 | ERROR | In-service attempted failed for tag "%s" on "%s". | **原因**: {server} の {tag} に対する「perform_action」コマンドがデータベース {sid} の開始に失敗しました。in service 操作に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正し |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | てから「create」操作を再試行する必要があります。 |
| 122521 | ERROR | Create of app "%s" on "%s" failed with return code of "%d". | **原因**: 内部アプリケーションタイプを作成する「app_create」コマンドの実行中にエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122522 | ERROR | Create of typ "%s" for app "%s" on "%s" failed with return code of "%d". | **原因**: 内部リソースタイプを作成する「typ_create」コマンドの実行中にエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122524 | ERROR | Setting "resstate" for resource "%s" on "%s" failed with return code of "%d". | **原因**: リソースの状態を {state} に設定する「ins_setstate」コマンドの実行中にエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122525 | ERROR | The values specified for the target and the template servers are the same: "%s". | **原因**:「extend」操作でターゲットサーバとテンプレートサーバで指定した値が同じでした。<br><br>**処置**: {target server} と {template server} に対して正しいパラメータを指 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 定する必要があります。{target server} は、{tag} が拡張されるサーバです。 |
| 122526 | ERROR | Unable to locate the oratab file in "/etc" or in "%s" on "%s". | 原因: {server} のデフォルトまたは代替ロケーションで oratab ファイルが見つかりませんでした。<br><br>処置: oratab ファイルが存在しており、Oracle ユーザの適切な権限を持っていることを確認してください。「extend」操作を完了するには有効な oratab ファイルが必要です。 |
| 122527 | ERROR | Unable to retrieve the Oracle user on "%s". | 原因:「canextend」操作または「extend」操作の間に {template server} から Oracle ユーザを取得できませんでした。<br><br>処置: Oracle バイナリの所有者は、{target server} と {template server} の有効なユーザである必要があります。Oracle データベース所有権の権限と所有権を修正し、操作を再試行してください。 |
| 122528 | ERROR | The Oracle user and/or group information for user "%s" does not exist on the server "%s". | 原因: LifeKeeper は、サーバ {server} の Oracle ユーザ {user} に対して Oracle ユーザおよびグループ情報を検索できません。<br><br>処置: 指定した {server} に Oracle ユーザ {user} が存在していることを確認してください。ユーザ {user} が存在しない場合、クラスタ内のすべてのサーバで、同じ uid および gid を使用して作成する必要があります。 |
| 122529 | ERROR | The id for user "%s" is not the same on template server "%s" and target server "%s". | 原因: ターゲットサーバ {target server} にある Oracle ユーザ {user} のユーザ ID |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | が、テンプレートサーバ {template server} にあるユーザ {user} の値と一致しません。<br><br>**処置**:Oracle ユーザ {user} のユーザ ID はクラスタ内のすべてのサーバで一致している必要があります。ユーザ ID の不一致をすべてのサーバで手動で修正し、「extend」操作を再試行する必要があります。 |
| 122530 | ERROR | The group id for user "%s" is not the same on template server "%s" and target server "%s". | **原因**: ターゲットサーバ {target server} にある Oracle ユーザ {user} のグループ ID が、テンプレートサーバ {template server} にあるユーザ {user} の値と一致しません。<br><br>**処置**: Oracle ユーザ {user} のグループ ID はクラスタ内のすべてのサーバで一致している必要があります。グループ ID の不一致をすべてのサーバで手動で修正し、「extend」操作を再試行する必要があります。 |
| 122532 | ERROR | No file system or raw devices found to extend for "%s" on "%s". | **原因**: サーバ {template server} で Oracle リソース {tag} の依存ファイルシステムまたは Raw デバイスリソースが見つかりません。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |
| 122533 | WARN | A RAMDISK (%s) was detected in the ORACLE Database configuration for "%s" on "%s". LifeKeeper cannot protect RAMDISK. This RAMDISK resource will not be protected by LifeKeeper! ORACLE hierarchy creation will continue. | **原因**: 指定したテーブルスペース、ログファイル、またはその他のデータベースファイル {database_file} が ramdisk として検出されました。現在の LifeKeeper 製品では、このタイプのリソースに保護を適用できません。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: ramdisk は保護されません。すべての Oracle データベース操作で必要なデータベースファイル {database_file} が使用可能になるように手動で確認する必要があります。 |
| 122534 | ERROR | Failed to initialize object instance for Oracle sid "%s" on "%s". | **原因**: 保護されている Oracle インスタンスの内部表現を作成中に予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122537 | ERROR | Update of instance info field for "%s" on "%s" failed (%s). | **原因**: 内部リソース情報フィールドを更新する「ins_setinfo」コマンドの実行中にエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122538 | ERROR | Initial connect with query buffer to database "%s" on "%s" failed, testing output. | **原因**: データベースの状態を判別するために Oracle データベース {sid} に接続しようとしたところ、失敗しました。<br><br>**処置**: 指定した認証情報で接続が失敗しました。隣接するログメッセージおよび関連エラーで詳細を確認してください。Oracle ログ (alert.log) および関連トレースログ (*.trc) で詳細を確認し、レポートされている問題を修正してください。 |
| 122542 | ERROR | The "%s [ %s ]" attempt of the database "%s" appears to have failed on "%s". | **原因**: データベースインスタンス {sid} に対して メソッド {action_method} を使 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 用した Oracle アクション {action} がサーバ {server} で失敗しました。<br><br>**処置**: 隣接するログメッセージおよび関連エラーで詳細を確認してください。Oracle ログ (alert.log) および関連トレースログ (*.trc) で詳細を確認し、レポートされている問題を修正してください。 |
| 122543 | ERROR | All attempts to "%s" database "%s" on "%s" failed | **原因**: サーバ {server} の Oracle データベース {sid} に対するアクション {action} を実行するためのあらゆる取り組みが失敗しました。<br><br>**処置**: 隣接するログメッセージおよび関連エラーで詳細を確認してください。Oracle ログ (alert.log) および関連トレースログ (*.trc) で詳細を確認し、レポートされている問題を修正してください。 |
| 122544 | ERROR | Update of "%s" sid "%s" on "%s" failed.  Reason: "%s" "%s" failed: "%s". | **原因**: データベース {sid} の oratab エントリを更新しようとしている際に予期しないエラーが発生しました。このエラーは oratab ファイルを開こうとしている間に発生しました。<br><br>**処置**: システム起動時にデータベースが自動で起動されないようにするために、{sid} の oratab ファイルエントリを手動で更新する必要があります。 |
| 122545 | ERROR | Unable to locate the oratab file in "/etc" or in "%s" on "%s". | **原因**: {server} のデフォルトまたは代替ロケーションで oratab ファイルが見つかりませんでした。<br><br>**処置**: oratab ファイルが存在しており、Oracle ユーザの適切な権限を持っていることを確認してください。「extend」操作を完了するには有効な oratab ファイルが必要です。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122546 | ERROR | Unable to open file "%s" on "%s" (%s). | **原因**: エラー {error} のため、サーバ {server} では、指定したファイル {file} を開くことも、ファイルにアクセスすることもできませんでした。<br><br>**処置**: 指定したファイル {file} の有無や権限を確認してください。隣接するログメッセージおよび関連エラーで詳細を確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122547 | ERROR | (cleanUpPids):Forcefully killing hung pid(s):pid(s)="%s" | **原因**: プロセス {pid} が正常終了するための要求に応答できませんでした。プロセス {pid} は強制終了します。<br><br>**処置**: コマンドラインで、プロセス {pid} が終了していることを確認してください。隣接するログメッセージで詳細および関連メッセージを確認してください。 |
| 122548 | ERROR | Unable to locate the DB utility (%s/%s) on this host. | **原因**: {path/utility} にある Oracle バイナリおよび必要なデータベースユーティリティ {utility} がこのサーバ {server} では見つかりませんでした。<br><br>**処置**: Oracle バイナリおよび必要なソフトウェアユーティリティがインストールされ、サーバ {server} で適切に設定されていることを確認してください。Oracle バイナリを各ノードにローカルインストールするか、クラスタ内の全ノードで使用可能な共有ストレージに配置する必要があります。 |
| 122549 | ERROR | Oracle internal error or non-standard Oracle configuration detected. Oracle User and/or Group set to "root". | **原因**: 検出された Oracle データベースインストールの所有権が、ルートユーザまたはルートグループに解決されています。ルートによる Oracle インストールの所有権は標準的ではない設定です。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: Oracle バイナリの所有者は {server} の有効な非 root ユーザである必要があります。Oracle データベース所有権の権限と所有権を修正し、操作を再試行してください。 |
| 122550 | ERROR | Initial inspection of "%s" failed, verifying failure or success of received output. | **原因**: 前の Oracle クエリー {query} またはコマンド {cmd} が成功を返すことができませんでした。<br><br>**処置**: 隣接するログメッセージおよび関連エラーで詳細を確認してください。Oracle ログ (alert.log) および関連トレースログ (*.trc) で詳細を確認し、レポートされている問題を修正してください。 |
| 122551 | ERROR | Logon failed with "%s" for "%s" on "%s". Please check username/password and privileges. | **原因**: 認証情報 {credentials} を使用してサーバ {server} のデータベースインスタンス {sid} へのログオンに失敗しました。無効なユーザ {user} またはパスワードが指定されました。<br><br>**処置**: Oracle データベースユーザ {user} およびパスワード {password} が有効であることを確認してください。また、Oracle データベースユーザ {user} は、試行されるアクションに対して十分な権限を持っている必要があります。 |
| 122552 | ERROR | %s | **原因**: メッセージに「sqlplus」コマンドの結果が含まれています。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。 |
| 122553 | ERROR | Unable to open file "%s" on "%s" (%s). | **原因**: エラー {error} のため、サーバ {server} では、指定したファイル {file} を開くことも、ファイルにアクセスする |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | こともできませんでした。<br><br>**処置**: 指定したファイル {file} の有無や権限を確認してください。隣接するログメッセージおよび関連エラーで詳細を確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122554 | ERROR | The tag "%s" on "%s" is not an Oracle instance or it does not exist. | **原因**: サーバ {server} の指定されたタグ {tag} は、既存の有効な Oracle リソースインスタンスを参照していません。<br><br>**処置**: UI または「lcdstatus (1M)」を使用して、リソースタグ {tag} の有無を確認してください。コマンド「ora-display」を使用するには、リソースタグ {tag} が Oracle リソースインスタンスである必要があります。 |
| 122555 | ERROR | Failed to create object instance for Oracle on "%s". | **原因**: Oracle リソースインスタンスの認定ユーザ、パスワード、データベースロールを更新している際に、保護されている Oracle インスタンスの内部表現を作成しようとしたところ、予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122557 | ERROR | Update of user and password failed for "%s" on "%s". | **原因**: リソースタグ {tag} のユーザおよびパスワードを更新する要求に失敗しました。指定した認証情報では、サーバ {server} に対する初期検証/接続ができませんでした。<br><br>**処置**: 目的の操作に対して正しい認証情 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 報 {user/password} が指定されていることを確認してください。隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122559 | ERROR | Update of user and password failed for "%s" on "%s". | 原因: サーバ {server} のリソースタグ {tag} のユーザおよびパスワード情報の更新に失敗しました。<br><br>処置: 目的の操作に対して正しい認証情報 {user/password} が指定されていることを確認してください。隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122562 | ERROR | Unable to find the Oracle executable "%s" on "%s". | 原因: 必要な Oracle 実行ファイル {exe} がこのサーバ {server} では見つかりませんでした。<br><br>処置: Oracle バイナリおよび必要なソフトウェアユーティリティがインストールされ、サーバ {server} で適切に設定されていることを確認してください。Oracle バイナリを各ノードにローカルインストールするか、クラスタ内の全ノードで使用可能な共有ストレージに配置する必要があります。 |
| 122566 | ERROR | Unable to find Oracle home for "%s" on "%s". | 原因: Oracle ホームディレクトリ {Oracle home} には、Oracle インスタンス {sid} の正常な動作に必要なファイルが含まれていないようです。<br><br>処置: Oracle ホームディレクトリ {Oracle home} に Oracle バイナリ、有効な spfile{sid}.ora、または init{sid}.ora |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ファイルが含まれていることをコマンドラインで確認してください。 |
| 122567 | ERROR | Oracle SID mismatch. The instance SID "%s" does not match the SID "%s" specified for the command. | **原因**: 保護されている Oracle インスタンスの内部表現を作成中に予期しないエラーが発生しました。指定した内部 ID {id} が、予想した SID {sid} と一致しません。<br><br>**処置**: パラメータが正しいことを確認してください。隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122568 | ERROR | DB Processes are not running on "%s". | **原因**: Oracle インスタンスのプロセスチェックで、サーバ {server} で実行されているプロセスは見つかりませんでした。<br><br>**処置**: ローカルリカバリが有効になっている場合、Oracle インスタンスはローカルで再起動されます。隣接するログメッセージで詳細および関連メッセージを確認してください。 |
| 122572 | ERROR | Failed to create flag "%s" on "%s". | **原因**: スタンバイノードへのフェイルオーバを促す Oracle ローカルリカバリ処理を制御するためのフラグを作成しようとして、予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 122574 | ERROR | all attempts to shutdown the database %s failed on "%s". | **原因**: ローカルリカバリプロセス中に、おそらくデータベース接続数が上限に達したため、Oracle データベースのシャッ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | トダウンに失敗しました。<br><br>**処置**: Oracle ログで、許容接続数の上限に達したことによって起こった接続の障害の有無を確認してください。その障害が見つかった場合は、接続数を増やしてください。また、ローカルリカバリの実行による quickCheck の障害発生が、接続の失敗を引き起こさないように、LK_ORA_NICE を 1 に設定してください。 |
| 122597 | ERROR | Failed to create object instance for Oracle on "%s". | **原因**: 保護されている Oracle インスタンスの内部表現を作成中、拡張前のチェックの際に予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから、pre-extend を再試行してください。 |
| 122598 | ERROR | Failed to create object instance for Oracle on "%s". | **原因**: 作成中の Oracle インスタンスの内部表現を作成しているときに、Oracle ホームディレクトリの有効性を確認しようとして、予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから、「create」を再試行してください。 |
| 122599 | ERROR | Failed to create object instance for Oracle on "%s". | **原因**: 保護中の Oracle インスタンスの内部表現を作成しているときに、テンプレートシステムで Oracle ユーザをルックアップしようとして、予期しないエラーが発生しました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから、extend を再試行してください。 |
| 122600 | ERROR | Failed to create object instance for Oracle on "%s". | **原因**: 保護中の Oracle インスタンスの内部表現を作成しているときに、リソースプロパティを表示しようとして、予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから、リソースプロパティの表示を再試行してください。 |
| 122601 | ERROR | Failed to create object instance for Oracle on "%s". | **原因**: 保護中の Oracle インスタンスの内部表現を作成しているときに、有効なデータベース認証を確認しようとして、予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから、コマンドを再試行してください。 |
| 122603 | ERROR | Failed to create object instance for Oracle on "%s". | **原因**: 保護中の Oracle インスタンスの内部表現を作成しているときに、Oracle リソースインスタンスに対してヘルスチェックを実行しようとして、予期しないエラーが発生しました。<br><br>**処置**: 正しい引数が quickCheck コマンドに渡されていることを確認し、隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから、restore を再試行してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122604 | ERROR | Failed to create object instance for Oracle on "%s". | 原因: 保護中の Oracle インスタンスの内部表現を作成しているときに、Oracle リソースインスタンスに対してローカルリカバリを実行しようとして、予期しないエラーが発生しました。<br><br>処置: 正しい引数が「recover」コマンドに渡されていることを確認し、隣接するログメッセージと関連メッセージで詳細を確認してください。レポートされているエラーを修正してから、recover を再試行してください。 |
| 122606 | ERROR | The Oracle database "%s" is not running or no open connections are available on "%s". | 原因: データベースインスタンス {sid} が実行されていません。またはデータベースへの接続が、指定された認証情報では使用できません。<br><br>処置: 選択した操作を完了するには、データベースインスタンス {sid} を {server} で起動し、正しい認証情報を指定する必要があります。 |
| 122607 | ERROR | The Oracle database "%s" is not running or no open connections are available on "%s". | 原因: データベースインスタンス {sid} が実行されていません。またはデータベースへの接続が、指定された認証情報では使用できません。<br><br>処置: 選択した操作を完了するには、データベースインスタンス {sid} を {server} で起動し、正しい認証情報を指定する必要があります。 |
| 122608 | ERROR | Failed to create object instance for Oracle on "%s". | 原因:「remove」操作で、Oracle リソースを Out of Service にするために必要なリソースオブジェクトインスタンスを作成できませんでした。<br><br>処置: 正しい引数が「remove」コマンドに渡されていることを確認し、隣接するロ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | グメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから、restore を再試行してください。 |
| 122609 | ERROR | Failed to create object instance for Oracle on "%s". | 原因:「restore」操作で、Oracle リソースを In Service にするために必要なリソースオブジェクトインスタンスを作成できませんでした。<br><br>処置: 正しい引数が「restore」コマンドに渡されていることを確認し、隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから、「restore」を再試行してください。 |
| 122610 | ERROR | Unable to "%s" on "%s" during resource create. | 原因: Oracle Application Recovery Kit は、リソース作成中に「getlocks」コマンドで管理ロックを作成できませんでした。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してから、create を再試行してください。 |
| 122611 | ERROR | %s | 原因: 要求された親 Oracle リソースと子 File System リソースとの間の依存関係の作成に失敗しました。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してから、create 操作を再試行してください。 |
| 122612 | ERROR | %s | 原因: 要求された親 Oracle リソースと子 Raw リソースとの間の依存関係の作成に失敗しました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してから、create 操作を再試行してください。 |
| 122613 | ERROR | %s | **原因**: 要求された親 Oracle リソースと子 Raw リソースとの間の依存関係の作成に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してから、create 操作を再試行してください。 |
| 122614 | ERROR | %s | **原因**: 要求された親 Oracle リソースと子 Listener リソースとの間の依存関係の作成に失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してから、create 操作を再試行してください。 |
| 122616 | ERROR | %s | **原因**: Oracle データベースの要求された起動またはシャットダウンに失敗しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してから、「restore」または「remove」操作を再試行してください。 |
| 122618 | ERROR | Dependency creation between Oracle database "%s (%s)" and the dependent resource "%s" on "%s" failed.  Reason: | **原因**: LifeKeeper は、データベースリソース {tag} と必要な子リソース {childtag} との間に依存関係を作成できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細お |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | よび関連メッセージを確認してください。問題が修正されている場合は、{tag} と {childtag} との間の依存関係を手動で作成できる可能性があります。 |
| 122619 | ERROR | Dependency creation between Oracle database "%s (%s)" and the dependent resource "%s" on "%s" failed. Reason: | **原因**: LifeKeeper は、データベースリソース {tag} と必要な子リソース {childtag} との間に依存関係を作成できませんでした。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。問題が修正されている場合は、{tag} と {childtag} との間の依存関係を手動で作成できる可能性があります。 |
| 122625 | ERROR | Unable to find the Oracle executable "%s" on "%s". | **原因**: quickCheck プロセスが、Oracle の実行可能ファイル「sqlplus」を見つけることができませんでした。<br><br>**処置**: Oracle 設定を確認し、隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされている問題を修正してください。 |
| 122626 | ERROR | Unable to find the Oracle executable "%s" on "%s". | **原因**: 削除プロセスが、Oracle の実行可能ファイル「sqlplus」を見つけることができませんでした。<br><br>**処置**: Oracle 設定を確認し、隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされている問題を修正してください。 |
| 122627 | ERROR | Unable to find the Oracle executable "%s" on "%s". | **原因**: リストアプロセスが、Oracle の実行可能ファイル「sqlplus」を見つけることができませんでした。<br><br>**処置**: Oracle 設定を確認し、隣接するロ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | グメッセージで詳細および関連メッセージを確認してください。レポートされている問題を修正してください。 |
| 122628 | ERROR | Unable to find the Oracle executable "%s" on "%s". | 原因: 復旧プロセスが、Oracle の実行可能ファイル「sqlplus」を見つけることができませんでした。<br><br>処置: Oracle 設定を確認し、隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされている問題を修正してください。 |
| 122632 | ERROR | Oracle SID mismatch. The instance SID "%s" does not match the SID "%s" specified for the command. | 原因: 削除の際に、削除プロセスに渡されるリソースインスタンス {sid} が、{sid} の内部リソースインスタンス情報と一致しません。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 122633 | ERROR | Oracle SID mismatch. The instance SID "%s" does not match the SID "%s" specified for the command. | 原因: リストアの際に、restore に渡されるリソースインスタンス {sid} が、{sid} の内部リソースインスタンス情報と一致しません。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 122634 | ERROR | Oracle SID mismatch. The instance SID "%s" does not match the SID "%s" specified for the command. | 原因: リソースリカバリの際に、recovery に渡されるリソースインスタンス {sid} が、{sid} の内部リソースインスタンス情報と一致しません。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122636 | ERROR | END failed hierarchy "%s" of resource "%s" on server "%s" with return value of %d | 原因:{server} で Oracle リソース階層 {tag} の作成に失敗しました。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 122638 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | 原因:サーバ {server} の Oracle データベースリソース {tag} で作成アクションに失敗しました。シグナル {sig} が作成プロセスによって受信されました。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122640 | ERROR | Error creating resource "%s" on server "%s" | 原因: Oracle リソースインスタンス {tag} を {server} に作成しようとして、予期しないエラーが発生しました。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 122641 | ERROR | Error creating resource "%s" on server "%s" | 原因: Oracle リソースインスタンス {tag} を {server} に作成しようとして、予期しないエラーが発生しました。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 122642 | ERROR | Error creating resource "%s" on server "%s" | 原因: Oracle リソースインスタンス {tag} を {server} に作成しようとして、予期しないエラーが発生しました。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認し、レポート |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | されているエラーを修正してください。 |
| 122643 | ERROR | Cannot extend resource "%s" to server "%s" | 原因: LifeKeeper は、{server} でリソース {resource} を拡張できませんでした。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 122644 | ERROR | Error getting resource information for resource "%s" on server "%s" | 原因: リソースインスタンス情報 {tag} を {server} で取得しようとして、予期しないエラーが発生しました。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正し、extend を再試行してください。 |
| 122645 | ERROR | Cannot access canextend script "%s" on server "%s" | 原因: {server} で依存関係の子リソース用の「canextend」スクリプトが見つからなかったので、LifeKeeper は拡張前のチェックを実行できませんでした。<br><br>処置: LifeKeeper の設定を確認してください。 |
| 122646 | ERROR | Error getting resource information for resource "%s" on server "%s" | 原因: リソースインスタンス情報 {tag} を {server} で取得しようとして、予期しないエラーが発生しました。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正し、extend を再試行してください。 |
| 122647 | ERROR | Cannot extend resource "%s" to server "%s" | 原因: LifeKeeper は、{server} でリソース {resource} を拡張できませんでし |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | た。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 122648 | ERROR | Resource with either matching tag "%s" or id "%s" already exists on server "%s" for App "%s" and Type "%s" | **原因**: データベースリソースの拡張中に、同じ {tag} または {id} で異なるリソースアプリケーションおよびタイプを使用しているリソースインスタンスが見つかりました。<br><br>**処置**: リソース ID は一意のものにする必要があります。Oracle リソースインスタンスに合致する ID のリソースインスタンスを削除してください。 |
| 122649 | ERROR | Error creating resource "%s" on server "%s" | **原因**: Oracle リソースインスタンス {tag} を {server} に作成しようとして、予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 122650 | ERROR | Cannot access extend script "%s" on server "%s" | **原因**: 依存関係にある子リソースのスクリプト {extend} がサーバ {server} で見つけられなかったので、サーバ {server} にデータベースリソース {resource} を拡張する要求が失敗しました。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 122651 | ERROR | Cannot extend resource "%s" to server "%s" | **原因**: 依存関係にある子リソースの拡張でエラーが発生したため、サーバ {server} にデータベースリソース {resource} を拡張する要求が失敗しました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 隣接するログメッセージで詳細および関連メッセージを確認し、レポートされているエラーを修正してください。 |
| 122654 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | **原因**: quickCheck プロセスがシグナルを受信したため、データベース {sid} のヘルスチェックが終了しました。これはおそらく、quickCheck プロセスの処理に、割り当てられているよりも長い時間が必要となっているためです。<br><br>**処置**: Oracle リソースのヘルスチェック時間は ORACLE_QUICKCHECK_TIMEOUT の設定で制御します。ヘルスチェックプロセスの完了に今より長い時間を割り当てる場合は、45 秒よりも大きな値に設定してください。 |
| 122655 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | **原因**: 削除プロセスがシグナルを受信したため、データベース {sid} を「Out of Service」にする要求が終了しました。これはおそらく、削除プロセスの処理に、割り当てられているよりも長い時間が必要となっているためです。<br><br>**処置**: Oracle リソースの削除時間は ORACLE_REMOVE_TIMEOUT の設定で制御します。削除プロセスの完了に今より長い時間を割り当てる場合は、240 秒よりも大きな値に設定してください。 |
| 122659 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | **原因**: リストアプロセスがシグナルを受信したため、データベース {sid} を「In Service」にする要求が終了しました。これはおそらく、リストアプロセスの処理に、割り当てられているよりも長い時間が必要となっているためです。<br><br>**処置**: Oracle リソースのリストア時間は |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ORACLE_RESTORE_TIMEOUT の設定で制御します。リストアプロセスの完了に今より長い時間を割り当てる場合は、240 秒よりも大きな値に設定してください。 |
| 122663 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | **原因**: リカバリプロセスがシグナルを受信したため、失敗したデータベースのリカバリが終了しました。これはおそらく、リカバリプロセスの処理に、割り当てられているよりも長い時間が必要となっているためです。<br><br>**処置**: Oracle リソースのリカバリ時間は ORACLE_RESTORE_TIMEOUT と ORACLE_REMOVE_TIMEOUT の設定で制御します。リカバリの完了に今より長い時間を割り当てる場合は、一方または両方を 240 秒よりも大きな値に設定してください。 |
| 122670 | ERROR | Update of "%s" sid "%s" on "%s" failed.  Reason: "%s" "%s" failed: "%s". | **原因**: データベース {sid} の oratab エントリを更新しようとしている際に予期しないエラーが発生しました。更新プロセスで使用する一時ファイルを開こうとして、エラーが発生しました。<br><br>**処置**: システム起動時にデータベースが自動で起動されないようにするために、{sid} の oratab ファイルエントリを手動で更新する必要があります。 |
| 122671 | ERROR | Update of "%s" sid "%s" on "%s" failed.  Reason: "%s" "%s" failed: "%s". | **原因**: データベース {sid} の oratab エントリを更新しようとしている際に予期しないエラーが発生しました。更新プロセスで使用する一時ファイルを閉じようとして、エラーが発生しました。<br><br>**処置**: システム起動時にデータベースが自動で起動されないようにするために、 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | {sid} の oratab ファイルエントリを手動で更新する必要があります。 |
| 122672 | ERROR | Update of "%s" sid "%s" on "%s" failed. Reason: "%s" "%s" failed: "%s". | 原因: データベース {sid} の oratab エントリを更新しようとしている際に予期しないエラーが発生しました。一時ファイルの名前を oratab に変更しようとして、エラーが発生しました。<br><br>処置: システム起動時にデータベースが自動で起動されないようにするために、{sid} の oratab ファイルエントリを手動で更新する必要があります。 |
| 122673 | ERROR | Unable to log messages queued while running as oracle user %s on %s. Reason: $! | 原因: メッセージをログファイルに追加しようとして、予期しないエラー {reason} が発生しました。これらのメッセージは、Oracle ユーザとして実行している間に生成されました。<br><br>処置: 障害に関する理由を参照し、修正処置を実行してください。 |
| 122674 | ERROR | Unable to open %s Reason: %s. | 原因: Oracle データベースへの接続を開き、データベース {cmd} を実行しようとしたときに、予期しないエラーが発生しました。<br><br>処置: 隣接するログメッセージで詳細および関連メッセージを確認してください。また、Oracle ログ (alert.log) および関連トレースログ (*.trc) で詳細を確認し、レポートされている問題を修正してください。 |
| 122680 | ERROR | Unable to find Oracle home for “%s” on “%s”. | 原因: Oracle ホームディレクトリ {Oracle home} には、Oracle インスタンス {sid} の正常な動作に必要なファイルが含まれていないようです。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: Oracle ホームディレクトリ {Oracle home} に Oracle バイナリ、有効な spfile{sid}.ora、または init{sid}.ora ファイルが含まれていることをコマンドラインで確認してください。 |
| 122681 | ERROR | Failed to create object instance for Oracle on "%s". | **原因**: 保護されている Oracle インスタンスの内部表現を作成中に予期しないエラーが発生しました。<br><br>**処置**: 隣接するログメッセージで詳細および関連メッセージを確認してください。レポートされているエラーを修正してから操作を再試行する必要があります。 |
| 122682 | ERROR | Unable to find the Oracle executable "%s" on "%s". | **原因**: 必要な Oracle 実行ファイル {exe} がこのサーバ {server} では見つかりませんでした。<br><br>**処置**: Oracle バイナリおよび必要なソフトウェアユーティリティがインストールされ、サーバ {server} で適切に設定されていることを確認してください。Oracle バイナリを各ノードにローカルインストールするか、クラスタ内の全ノードで使用可能な共有ストレージに配置する必要があります。 |
| 122683 | ERROR | Backup node %s is unreachable; abort username/password changes. | **原因**: バックアップノード {node} に接続できません。保留中のユーザー名/パスワードに対する変更は中断されました。<br><br>**処置**: 指定したバックアップノード {node} 上でSIOS LifeKeeper が起動中で、すべてのコミュニケーションパスが確立されていることが確認できたら、再度操作を行ってください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122684 | ERROR | The restore of %s has timed out on server %s. The default TIMEOUT is 300 seconds. To increase the TIMEOUT, set ORACLE_RESTORE_TIMEOUT in /etc/default/LifeKeeper. | **原因**: サーバー {server} 上で Oracle リソース {resource} の restore 処理がタイムアウトしました。これは、In Service 処理が設定された時間を経過すると発生します。<br><br>**処置**: Oracle リソースのリストア時間は ORACLE_RESTORE_TIMEOUT の設定で制御します。リストアプロセスの完了に今より長い時間を割り当てる場合は、240 秒よりも大きな値に設定してください。 |
| 122685 | ERROR | The remove of %s has timed out on server %s. The default TIMEOUT is 300 seconds. To increase the TIMEOUT, set ORACLE_REMOVE_TIMEOUT in /etc/default/LifeKeeper. | **原因**: サーバー {server} 上で Oracle リソース {resource} の remove 処理がタイムアウトしました。 これは、Out of Service 処理が設定された時間を経過すると発生します。<br><br>**処置**: Oracle リソースの削除時間は ORACLE_REMOVE_TIMEOUT の設定で制御します。削除プロセスの完了に今より長い時間を割り当てる場合は、240 秒よりも大きな値に設定してください。 |
| 122686 | ERROR | The quickCheck of %s has timed out on server %s. The default TIMEOUT is 45 seconds. To increase the TIMEOUT, set ORACLE_QUICKCHECK_TIMEOUT in /etc/default/LifeKeeper. | **原因**: サーバー {server} 上で Oracle リソース {resource} に対する quickCheck がタイムアウトしました。 これは、health check 処理が設定された時間を経過すると発生します。<br><br>**処置**: Oracle リソースのヘルスチェック時間は ORACLE_QUICKCHECK_TIMEOUT の設定で制御します。ヘルスチェックプロセスの完了に今より長い時間を割り当てる場合は、45 秒よりも大きな値に設定してください。 |
| 122687 | ERROR | Usage: %s %s | **原因**: getalertlog スクリプトに対して無 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 効なパラメーターが指定されました。<br><br>処置: Oracle リソースのタグ名が、最初のコマンドラインのパラメーターとして提供される必要があります。パラメーターが正しいかどうかを確認し、再度操作を行ってください。 |
| 122702 | ERROR | Failed start OHAS. | |
| 122704 | ERROR | Failed start ASM. | |
| 122706 | ERROR | Failed start Diskgroup. | |
| 122708 | ERROR | Failed stop Diskgroup. | |
| 122710 | ERROR | Failed stop ASM. | |
| 122712 | ERROR | Failed stop OHAS. | |
| 122713 | ERROR | OHAS is not running. | |
| 122714 | ERROR | ASM is not running. | |
| 122715 | ERROR | Diskgroup is not running. | |
| 122716 | ERROR | $usage | |
| 122717 | ERROR | Template resource \"$template_tag\" on server \"$template_sys\" does note exist | |
| 122718 | ERROR | Cannot extend resource \"$template_tag\" to server \"$me\" | |
| 122719 | ERROR | $usage | |
| 122720 | ERROR | END failed create of \"$Tag\" due to a \"$sig\" signal | |
| 122722 | ERROR | Unable to getlocks on $me during resouce create. | |
| 122725 | ERROR | Error creating resource $Tag. Error ($rc | |
| 122726 | ERROR | END failed hierarchy create of resource $Tag with return value of $ecode. | |
| 122728 | ERROR | Unable to rlslocks on $me during resource create. | |
| 122729 | ERROR | $usage | |
| 122730 | ERROR | END failed extend of \"$Tag\" due to a | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | \"$sig\" signal | |
| 122732 | ERROR | Template resource \"$TemplateTag\" on server \"$TemplateSys\" does not exist | |
| 122733 | ERROR | Error creating resouce \"$Tag\"on server \"$me\" | |
| 122735 | ERROR | END failed hierarchy extend of resource $Tag with return value of $ecode. | |
| 122736 | ERROR | $usage | |
| 122737 | ERROR | Failed to create object instance for Oracle ASM on $oracle::me | |
| 122738 | ERROR | Usage : $cmd $usage | |
| 122739 | ERROR | Failed to create object instance for Oracle ASM on \"$me\". | |
| 122740 | ERROR | $usage | |
| 122742 | ERROR | backup node $back_dead_name is unreachable; abort protection Diskgroup changes. | |
| 122744 | ERROR | Update of protection Diskgroup failed for \"$tag\" on \"$oarcleasm::me\" | |
| 122746 | ERROR | Update of protection Diskgroup failed foor \"$tag\" on \"$remote\" | |
| 122750 | ERROR | Failed to init the object. | |
| 122751 | ERROR | Usage: %s %s | |

# 8.1.11. Oracle Listener キットメッセージカタログ

**Control F** を使用して、各カタログの特定のエラーコードを検索します。エラーコードの検索には、画面右上の検索ボックスをご利用ください。

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122005 | ERROR | Unable to "%s" on "%s" | 原因:「getlocks」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122007 | ERROR | Unable to "%s" on "%s" | 原因:「rlslocks」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122009 | ERROR | The path %s is not a valid file. | 原因: listener.ora ファイルがありません。<br><br>処置: ファイルが存在していることを確認し、操作を再試行してください。 |
| 122010 | ERROR | The listener user does not exist on the server %s. | 原因:「Stat」コマンドでユーザ ID を取得できませんでした。<br><br>処置: 操作を再試行してください。 |
| 122011 | ERROR | The listener user does not exist on the server %s. | 原因: UID が passwd ファイルにありません。<br><br>処置: UID が passwd ファイルに存在していることを確認し、操作を再試行して |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ください。 |
| 122012 | ERROR | The listener user does not exist on the server %s. | 原因:ユーザ名が passwd ファイルにありません。<br><br>処置: ユーザ名が passwd ファイルに存在していることを確認し、操作を再試行してください。 |
| 122023 | ERROR | The %s command failed (%d) | 原因: このメッセージに「lsnrctl」コマンドの戻りコードが含まれています。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122024 | ERROR | $line | 原因: メッセージに「lsnrctl」コマンドの結果が含まれています。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122039 | ERROR | Usage error | 原因: リストア操作で無効なパラメータが指定されました。<br><br>処置: パラメータを確認し、操作を再試行してください。 |
| 122040 | ERROR | Script $cmd has hung on the restore of \"$opt_t\". Forcibly terminating. | 原因: リスナーの restore スクリプトがタイムアウト値に達しました。<br><br>処置: listener.ora が有効であること、および /etc/default/LifeKeeper の LSNR_START_TIME（デフォルトでは 35 秒）がリスナーの起動に必要な時間以上の値に設定されていることを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122041 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | 原因: LifeKeeper は、{server} でリソース {resource} をリストアできませんでした。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122045 | ERROR | Error getting resource information for resource "%s" on server "%s" | 原因: リソース情報の取得に失敗しました。<br><br>処置: LifeKeeper の設定を確認してください。 |
| 122046 | ERROR | Usage error | 原因: リストア操作で無効なパラメータが指定されました。<br><br>処置: パラメータを確認し、操作を再試行してください。 |
| 122049 | ERROR | The script $cmd has hung on remove of \"$opt_t\".  Forcibly terminating. | 原因: リスナーの remove スクリプトがタイムアウト値に達しました。<br><br>処置: listener.ora が有効であること、および /etc/default/LifeKeeper の LSNR_STOP_TIME (デフォルトでは 35 秒) がリスナーの停止に必要な時間以上の値に設定されていることを確認してください。 |
| 122051 | ERROR | Error getting resource information for resource "%s" on server "%s" | 原因: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。<br><br>処置: LifeKeeper の設定を確認してください。 |
| 122055 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | 原因: LifeKeeper は、{server} でリソース {resource} に対して quickCheck を実 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 行できませんでした。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122057 | ERROR | Error getting resource information for resource "%s" on server "%s" | **原因**: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 122064 | WARN | The %s level is set to %s a %s will not occur. | **原因**: リスナーの最低限の保護レベルは起動と監視です。<br><br>**処置**: リスナーを手動で起動してください。 |
| 122066 | ERROR | Script has hung checking \"$tag\". Forcibly terminating. | **原因**: リスナーの quickCheck スクリプトがタイムアウト値に達しました。<br><br>**処置**: listener.ora が有効であること、および /etc/default/LifeKeeper の LSNR_STATUS_TIME (デフォルトでは 15 秒) がリスナーの確認に必要な時間以上の値に設定されていることを確認してください。 |
| 122067 | ERROR | Usage error | **原因**: quickCheck 操作で無効なパラメータが指定されました。<br><br>**処置**: パラメータを確認し、操作を再試行してください。 |
| 122069 | ERROR | Usage error | **原因**: 削除操作で無効なパラメータが指定されました。<br><br>**処置**: パラメータを確認し、操作を再試 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 行してください。 |
| 122072 | ERROR | %s: resource "%s" not found on local server | **原因**: 復旧操作で無効なパラメータが指定されました。<br><br>**処置**: パラメータを確認し、操作を再試行してください。 |
| 122074 | WARN | The local recovery attempt has failed but %s level is set to %s preventing a failover to another node in the cluster. With %s recovery set all local recovery failures will exit successfully to prevent resource failovers. | **原因**: オプションのリスナーリカバリレベルがローカルリカバリにのみ設定されています。<br><br>**処置**: リソースツリーを手動でスイッチオーバしてください。 |
| 122078 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | **原因**:LifeKeeper は、{server} でリソース {resource} を復旧できませんでした。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122082 | ERROR | Error getting resource information for resource "%s" on server "%s" | **原因**: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 122083 | ERROR | $cmd has hung checking \"$tag\". Forcibly terminating | **原因**: recover スクリプトがシグナルによって停止されました。<br><br>**処置**: listener.ora が有効であることを確認してください。 |
| 122084 | ERROR | Cannot extend resource "%s" to server "%s" | **原因**: LifeKeeper は、{server} でリソー |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ス {resource} を拡張できませんでした。<br><br>**処置**: パラメータを確認し、操作を再試行してください。 |
| 122085 | ERROR | Usage: %s %s | **原因**: canextend 操作で無効なパラメータが指定されました。<br><br>**処置**: パラメータを確認し、操作を再試行してください。 |
| 122086 | ERROR | The values specified for the target and the template servers are the same. Please specify the correct values for the target and template servers. | **原因**: ターゲットサーバに指定した値とテンプレートサーバに指定した値が同じです。<br><br>**処置**: メッセージテキストに記載されている手順を実行してください。 |
| 122087 | ERROR | Error getting resource information for resource "%s" on server "%s" | **原因**: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 122088 | ERROR | Error getting resource information for resource "%s" on server "%s" | **原因**: リソース情報からのリスナーユーザ名の取得に失敗しました。<br><br>**処置**: リソース情報フィールドが有効であることを確認し、操作を再試行してください。 |
| 122089 | ERROR | The listener user %s does not exist on the server %s. | **原因**: ユーザ名が passwd ファイルにありません。<br><br>**処置**: ユーザ名が passwd ファイルに存在していることを確認し、操作を再試行してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122090 | ERROR | The id for user %s is not the same on template server %s and target server %s. | **原因**: ユーザ ID はどちらのサーバでも同じものにする必要があります。<br><br>**処置**: ユーザ ID を同じものにしてください。 |
| 122091 | ERROR | The group id for user %s is not the same on template server %s and target server %s. | **原因**: グループ ID はどちらのサーバでも同じものにする必要があります。<br><br>**処置**: グループ ID を同じものにしてください。 |
| 122092 | ERROR | Cannot access canextend script "%s" on server "%s" | **原因**: {server} で「canextend」スクリプトを見つけられなかったので、LifeKeeper は拡張前のチェックを実行できませんでした。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 122097 | ERROR | Usage: %s %s | **原因**:「configActions」操作で無効な引数が指定されました。<br><br>**処置**: 引数を確認し、操作を再試行してください。 |
| 122098 | ERROR | Error getting resource information for resource "%s" on server "%s" | **原因**: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 122099 | ERROR | Unable to update the resource %s to change the %s to %s on %s. | **原因**: LifeKeeper は情報フィールドに情報を入力できませんでした。<br><br>**処置**: LifeKeeper を再起動し、操作を再試行してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122100 | ERROR | Error getting resource information for resource "%s" on server "%s" | 原因: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。<br><br>処置: LifeKeeper の設定を確認してください。 |
| 122101 | ERROR | Unable to update the resource %s to change the %s to %s on %s. | 原因: LifeKeeper は {server} で情報フィールドに情報を入力できませんでした。<br><br>処置: {server} で LifeKeeper を再起動し、操作を再試行してください。 |
| 122103 | ERROR | Usage: %s %s | 原因: 作成操作で無効なパラメータが指定されました。<br><br>処置: パラメータを確認し、操作を再試行してください。 |
| 122124 | ERROR | END failed hierarchy "%s" of resource "%s" on server "%s" with return value of %d | 原因: LifeKeeper は、{server} でリソース {resource} を作成できませんでした。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122126 | ERROR | Unable to "%s" on "%s | 原因:「rlslocks」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122127 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | 原因: LifeKeeper は、{server} でリソース {resource} を作成できませんでした。<br><br>処置: 関連するエラーをログで確認し、 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | レポートされている問題を解決してください。 |
| 122129 | ERROR | Unable to "%s" on "%s | 原因:「getlocks」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122131 | ERROR | Error creating resource "%s" on server "%s" | 原因: LifeKeeper は、{server} でリソース {resource} を作成できませんでした。<br><br>処置: LifeKeeper の設定を確認してください。 |
| 122133 | ERROR | Unable to create a file system resource hierarchy for the file system %s. | 原因:「filesyshier」の実行中に予期しないエラーが発生しました。<br><br>処置: 隣接するログメッセージで詳細を確認してください。 |
| 122135 | ERROR | Unable to create a dependency between parent tag %s and child tag %s. | 原因:「dep_create」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122140 | ERROR | Resource "%s" is not ISP on server "%s" Manually bring the resource in service and retry the operation | 原因: リスナーリソースが依存する IP リソース {tag} は ISP の必要があります。<br><br>処置: メッセージテキストに記載されている手順を実行してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122141 | ERROR | Unable to create a dependency between parent tag %s and child tag %s. | 原因:「dep_create」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122144 | ERROR | Usage: %s %s | 原因:「create_ins」操作で無効なパラメータが指定されました。<br><br>処置: パラメータを確認し、操作を再試行してください。 |
| 122145 | ERROR | An error has occurred in utility %s on server %s. View the LifeKeeper logs for details and retry the operation. | 原因:「app_create」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122146 | ERROR | An error has occurred in utility %s on server %s. View the LifeKeeper logs for details and retry the operation. | 原因:「typ_create」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122147 | ERROR | An error has occurred in utility %s on server %s. View the LifeKeeper logs for details and retry the operation. | 原因:「newtag」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122148 | ERROR | Error creating resource "%s" on server "%s" | 原因: LifeKeeper は、{server} でリソース {resource} を作成できませんでした。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: LifeKeeper の設定を確認してください。 |
| 122149 | ERROR | An error has occurred in utility %s on server %s. View the LifeKeeper logs for details and retry the operation. | **原因**:「ins_setstate」の実行中に予期しないエラーが発生しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122150 | ERROR | Error creating resource "%s" on server "%s" | **原因**: LifeKeeper は、{server} でリソース {resource} を作成できませんでした。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 122151 | ERROR | Usage: %s %s | **原因**:「depstoextend」操作で無効な引数が指定されました。<br><br>**処置**: 引数を確認し、操作を再試行してください。 |
| 122152 | ERROR | Usage: %s %s | **原因**:「extend」操作で無効なパラメータが指定されました。<br><br>**処置**: パラメータを確認し、操作を再試行してください。 |
| 122153 | ERROR | Error getting resource information for resource "%s" on server "%s" | **原因**: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 122154 | ERROR | Cannot extend resource "%s" to server "%s" | **原因**: LifeKeeper は、{server} でリソー |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ス {resource} を拡張できませんでした。 |
| 122155 | ERROR | Resource with either matching tag "%s" or id "%s" already exists on server "%s" for App "%s" and Type "%s" | **原因**: リスナーリソースの拡張中に、同じ {tag} または {id} で異なるリソースアプリケーションおよびタイプを使用しているリソースインスタンスが見つかりました。<br><br>**処置**: リソース ID は一意のものにする必要があります。Oracle Listener リソースインスタンスに合致する ID のリソースインスタンスを削除してください。 |
| 122156 | ERROR | Cannot access extend script "%s" on server "%s" | **原因**: LifeKeeper は、{server} でスクリプト EXTEND が見つからなかったので、リソース階層を拡張できませんでした。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 122157 | ERROR | Usage: %s %s | **原因**:「getConfigIps」操作で無効な引数が指定されました。<br><br>**処置**: 引数を確認し、操作を再試行してください。 |
| 122158 | ERROR | The file %s is not a valid listener file. The file does not contain any listener definitions. | **原因**: リスナー定義が見つかりませんでした。<br><br>**処置**: リスナー定義が listener.ora にあることを確認し、操作を再試行してください。 |
| 122159 | ERROR | Usage: %s %s | **原因**:「getSidListeners」操作で無効な引数が指定されました。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 引数を確認し、操作を再試行してください。 |
| 122160 | ERROR | The file %s is not a valid listener file. The file does not contain any listener definitions. | **原因**: リスナー定義が見つかりませんでした。<br><br>**処置**: リスナー定義が listener.ora にあることを確認し、操作を再試行してください。 |
| 122161 | ERROR | Usage: %s %s | **原因**:「lsn-display」操作で無効な引数が指定されました。<br><br>**処置**: 引数を確認し、操作を再試行してください。 |
| 122162 | ERROR | Error getting resource information for resource "%s" on server "%s" | **原因**: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 122163 | ERROR | Usage: %s %s | **原因**: updateHelper 操作で無効なパラメータが指定されました。<br><br>**処置**: パラメータを確認し、操作を再試行してください。 |
| 122164 | ERROR | END failed hierarchy "%s" of resource "%s" on server "%s" with return value of %d | **原因**: LifeKeeper は、{server} でリソース {resource} を更新できませんでした。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122166 | ERROR | Usage: %s %s | 原因:「updateHelper」操作で無効なパラメータが指定されました。<br><br>処置: パラメータを確認し、操作を再試行してください。 |
| 122170 | ERROR | Unable to create a dependency between parent tag %s and child tag %s. | 原因:「dep_create」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122171 | ERROR | Unable to create a dependency between parent tag %s and child tag %s. | 原因:「dep_create」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122172 | ERROR | Usage: %s %s | 原因:「updIPDeps」操作で無効な引数が指定されました。<br><br>処置: 引数を確認し、操作を再試行してください。 |
| 122173 | ERROR | END failed hierarchy "%s" of resource "%s" on server "%s" with return value of %d | 原因: LifeKeeper は、{server} でリソース {resource} を更新できませんでした。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122175 | ERROR | Unable to "%s" on "%s | 原因:「rlslocks」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | レポートされている問題を解決してください。 |
| 122177 | ERROR | Unable to "%s" on "%s | 原因:「getlocks」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122180 | ERROR | Unable to create a dependency between parent tag %s and child tag %s. | 原因:「dep_create」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122181 | ERROR | Unable to create a dependency between parent tag %s and child tag %s. | 原因:「dep_create」の実行中に予期しないエラーが発生しました。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 122183 | ERROR | The path %s is not a valid file. | 原因: listener.ora ファイルがありません。<br><br>処置: ファイルが存在していることを確認し、操作を再試行してください。 |
| 122185 | ERROR | The file %s is not a valid listener file. The file does not contain any listener definitions. | 原因: LifeKeeper は有効なリスナー定義の検索に失敗しました。<br><br>処置: 有効なリスナー定義が listener.ora にあることを確認し、操作を再試行してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122186 | ERROR | The value specified for %s cannot be empty. Please specify a value for this field. | 原因:設定ファイルまたは実行可能ファイルの {path} フィールドが空です。<br><br>処置: 空でない値を {path} に入力し、操作を再試行してください。 |
| 122187 | ERROR | The path %s is not a valid file or directory. | 原因: 定義済み {path} が無効です。<br><br>処置: {path} が存在していることを確認し、操作を再試行してください。 |
| 122188 | ERROR | The path %s is not a valid file or directory. | 原因: {path} がありません。<br><br>処置: {path} が存在していることを確認し、操作を再試行してください。 |
| 122189 | ERROR | The value specified for %s cannot be empty. Please specify a value for this field. | 原因: 設定および実行パスフィールドが空です。<br><br>処置: フィールドのパスを入力してください。 |
| 122190 | ERROR | Usage: %s %s | 原因:「valid_rpath」操作で無効な引数が指定されました。<br><br>処置: 引数を確認し、操作を再試行してください。 |
| 122191 | ERROR | The values specified for the target and the template servers are the same. | 原因: valid_rpath の無効な引数。<br><br>処置: 引数を確認し、操作を再試行してください。 |
| 122192 | ERROR | Unable to find the configuration file "oratab" in its default locations, /etc/oratab or %s on "%s" | 原因: /etc/oratab または {path} に oratab ファイルがありません。<br><br>処置: oratab ファイルが {path} にあるこ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | と、または /etc/default/Lifekeeper の ORACLE_ORATABLOC が有効なパスに設定されていることを確認してください。 |
| 122193 | ERROR | Unable to find the configuration file "oratab" in its default locations, /etc/oratab or %s on "%s" | 原因: /etc/oratab または {path} に oratab ファイルがありません。<br><br>処置: oratab ファイルが {path} にあること、または /etc/default/Lifekeeper の ORACLE_ORATABLOC が有効なパスに設定されていることを確認してください。 |
| 122194 | ERROR | Unable to find the configuration file "oratab" in its default locations, /etc/oratab or %s on "%s" | 原因: /etc/oratab または {path} に oratab ファイルがありません。<br><br>処置: oratab ファイルが {path} にあること、または /etc/default/Lifekeeper の ORACLE_ORATABLOC が有効なパスに設定されていることを確認してください。 |
| 122195 | ERROR | Unable to find the configuration file "oratab" in its default locations, /etc/oratab or %s on "%s" | 原因: /etc/oratab または {path} に oratab ファイルがありません。<br><br>処置: oratab ファイルが {path} にあること、または /etc/default/Lifekeeper の ORACLE_ORATABLOC が有効なパスに設定されていることを確認してください。 |
| 122196 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | 原因:LifeKeeper は、{server} でリソース {resource} を削除できませんでした。<br><br>処置: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122197 | ERROR | Unable to find the configuration file \"oratab\" in its default locations, /etc/oratab or $listener::oraTab on \"$me\" | 原因: /etc/oratab または {path} に oratab ファイルがありません。<br><br>処置: oratab ファイルが {path} にあること、または /etc/default/Lifekeeper の ORACLE_ORATABLOC が有効なパスに設定されていることを確認してください。 |
| 122198 | ERROR | remove for $okListener failed. | |

# 8.1.12. Oracle PDB キットメッセージカタログ

Use **Control F** Control F を使用して、各カタログの特定のエラーコードを検索します。エラーコードの検索には、画面右上の検索ボックスをご利用ください。

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122251 | ERROR | Update of pluggable database info field for "%s" on "%s" failed (%s). | |
| 122252 | ERROR | Initial connect with query buffer to database "%s" on "%s" failed, testing output. | |
| 122253 | ERROR | The Oracle database "%s" is not running or no open connections are available on "%s". | |
| 122261 | ERROR | The Oracle resource (%s) and dependency are not set on %s. | |
| 122262 | ERROR | Usage: %s %s | |
| 122263 | ERROR | The restore of %s has timed out on server %s. The default TIMEOUT is 300 seconds. To increase the TIMEOUT, set ORACLE_RESTORE_TIMEOUT in /etc/default/LifeKeeper. | |
| 122264 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | |
| 122268 | ERROR | Failed to create object instance for Oracle on "%s". | |
| 122269 | ERROR | no dependency for Oracle on "%s". | |
| 122270 | ERROR | Unable to find the Oracle executable "%s" on "%s". | |
| 122271 | ERROR | Usage: %s %s | |
| 122272 | ERROR | The remove of %s has timed out on server %s. The default TIMEOUT is 300 seconds. To increase the TIMEOUT, set ORACLE_REMOVE_TIMEOUT in /etc/default/LifeKeeper. | |
| 122273 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | |
| 122277 | ERROR | Usage: %s %s | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122278 | ERROR | Failed to create object instance for Oracle on "%s". | |
| 122279 | ERROR | no dependency for Oracle on "%s". | |
| 122280 | ERROR | Unable to find the Oracle executable "%s" on "%s". | |
| 122281 | ERROR | The quickCheck of %s has timed out on server %s. The default TIMEOUT is 45 seconds. To increase the TIMEOUT, set ORACLE_QUICKCHECK_TIMEOUT in /etc/default/LifeKeeper. | |
| 122282 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | |
| 122284 | ERROR | Failed to create object instance for Oracle on "%s". | |
| 122285 | ERROR | no dependency for Oracle on "%s". | |
| 122287 | ERROR | Unable to find the Oracle executable "%s" on "%s". | |
| 122288 | ERROR | Usage: %s %s | |
| 122291 | ERROR | Cannot extend resource "%s" to server "%s" | |
| 122292 | ERROR | The values specified for the target and the template servers are the same: "%s". | |
| 122294 | ERROR | Cannot access canextend script "%s" on server "%s" | |
| 122295 | ERROR | Usage: %s %s | |
| 122296 | ERROR | DB instance "%s" is not protected on "%s". | |
| 122297 | ERROR | Failed to create object instance for OraclePDB on "%s". | |
| 122298 | ERROR | Unable to locate the oratab file "%s" on "%s". | |
| 122299 | ERROR | END failed hierarchy "%s" of resource "%s" on server "%s" with return value of %d | |
| 122301 | ERROR | Unable to "%s" on "%s" during resource create. | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122302 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | |
| 122304 | ERROR | Unable to "%s" on "%s" during resource create. | |
| 122305 | ERROR | Unable to determine Oracle user for "%s" on "%s". | |
| 122306 | ERROR | Error creating resource "%s" on server "%s" | |
| 122308 | ERROR | Dependency creation between Oracle pluggable database "%s (%s)" and the dependent resource "%s" on "%s" failed. Reason | |
| 122309 | ERROR | %s | |
| 122311 | ERROR | In-service attempted failed for tag "%s" on "%s". | |
| 122312 | ERROR | Usage: %s %s | |
| 122313 | ERROR | Create of app "%s" on "%s" failed with return code of "%d". | |
| 122314 | ERROR | Create of typ "%s" for app "%s" on "%s" failed with return code of "%d". | |
| 122316 | ERROR | Create of resource tag via "newtag" on "%s" failed. | |
| 122318 | ERROR | Error creating resource "%s" on server "%s" | |
| 122320 | ERROR | Setting "resstate" for resource "%s" on "%s" failed with return code of "%d". | |
| 122321 | ERROR | Error creating resource "%s" on server "%s" | |
| 122322 | ERROR | Usage: %s %s | |
| 122323 | ERROR | Usage: %s %s | |
| 122324 | ERROR | Usage: %s %s | |
| 122325 | ERROR | Cannot extend resource "%s" to server "%s" | |
| 122326 | ERROR | Resource with either matching tag "%s" or id "%s" already exists on server "%s" for App "%s" and Type "%s" | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 122327 | ERROR | Error creating resource "%s" on server "%s" | |
| 122328 | ERROR | Cannot access extend script "%s" on server "%s" | |
| 122329 | ERROR | Cannot extend resource "%s" to server "%s" | |
| 122330 | ERROR | Usage: %s %s | |
| 122331 | ERROR | Failed to create object instance for OraclePDB on "%s". | |
| 122332 | ERROR | Usage: %s %s | |
| 122334 | ERROR | Backup node %s is unreachable; abort protection PDB changes. | |
| 122336 | ERROR | Update of protection PDB failed for "%s" on "%s". | |
| 122339 | ERROR | Usage: %s %s | |
| 122340 | ERROR | Usage: %s %s | |
| 122341 | ERROR | Usage: %s %s | |
| 122342 | ERROR | Failed to create object instance for OraclePDB on "%s". | |
| 122343 | ERROR | Usage: %s %s | |
| 122344 | ERROR | END failed %s of "%s" on server "%s" due to a "%s" signal | |
| 122348 | ERROR | Failed to create flag "%s" on "%s". | |
| 122350 | ERROR | Failed to create object instance for Oracle on "%s". | |
| 122351 | ERROR | no dependency for Oracle on "%s". | |
| 122353 | ERROR | Unable to find the Oracle executable "%s" on "%s". | |
| 122356 | ERROR | Usage: %s %s | |
| 122357 | ERROR | Failed to create object instance for OraclePDB on "%s". | |
| 122358 | ERROR | The selected oracle SID "%s" is not a CDB. | |
| 122359 | ERROR | No protectable PDB found for the | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | selected SID "%s". | |
| 122360 | ERROR | No protected Oracle database found on "%s". | |

# 8.1.13. SCSI キットメッセージカタログ

**Control F** を使用して、各カタログの特定のエラーコードを検索します。エラーコードの検索には、画面右上の検索ボックスをご利用ください。

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 125102 | ERROR | `printf 'Template resource "%s" on server "%s" does not exist' $TemplateTagName $TemplateSysName` | 原因: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。 |
| 125103 | ERROR | `printf '%s is not shareable with any machine.' $DEV` | 原因: デバイスは他のシステムと共有されていないようです。<br><br>処置: デバイスがクラスタ内のすべてのサーバからアクセス可能であることを確認してください。すべての関連ストレージドライバおよびソフトウェアが正常にインストールおよび設定されていることを確認してください。 |
| 125104 | ERROR | `printf 'Failed to create disk hierarchy for "%s" on "%s"' $PRIMACH $DEV` | 原因: 物理ディスクを保護するリソースの作成に失敗しました。<br><br>処置: 隣接するログメッセージで詳細を確認し、レポートされている問題を解決してください。 |
| 125107 | ERROR | `printf 'Template resource "%s" on server "%s" does not exist' $TemplateTagName $TemplateSysName` | 原因: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。 |
| 125114 | ERROR | `printf 'Template resource "%s" on server "%s" does not exist' $TemplateTagName $TemplateSysName` | 原因: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。 |
| 125120 | ERROR | `printf 'Template resource "%s" on server "%s" does not exist' $TemplateTagName $TemplateSysName` | 原因: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。 |
| 125123 | ERROR | `printf 'Cannot access depstoextend script "%s" on server "%s"' $depstoextend` | 原因: {server} でスクリプト |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | $TargetSysName` | {DEPSTOEXTEND}を見つけられなかったので、LifeKeeper はリソース階層に対して拡張前のチェックを実行できませんでした。<br><br>**処置**: LifeKeeper の設定を確認してください。 |
| 125126 | ERROR | `printf 'Template resource "%s" on server "%s" does not exist' $ChildTag $TemplateSysName` | **原因**: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。 |
| 125129 | ERROR | `printf 'Template resource "%s" on server "%s" does not exist' $TemplateTagName $TemplateSysName` | **原因**: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。 |
| 125155 | ERROR | SCSI $DEV failed to lock. | **原因**: SCSI デバイスをロックする際に問題が発生しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認し、レポートされている問題を解決してください。 |
| 125164 | ERROR | SCSI $INFO failed to unlock. | **原因**: SCSI デバイスをロック解除する際に問題が発生しました。<br><br>**処置**: 隣接するログメッセージで詳細を確認し、レポートされている問題を解決してください。 |
| 125181 | ERROR | `printf 'Template resource "%s" on server "%s" does not exist' $TemplateTag  $TemplateSysName` | **原因**: LifeKeeper は、{server} でリソース {tag} を見つけられませんでした。 |
| 125194 | ERROR | Failed to check disk.(tag=\"$opt_t\" | |

# 8.1.14. Quick Service Protection Recovery Kit メッセージカタログ

**Control F** を使用して、各カタログの特定のエラーコードを検索します。エラーコードの検索には、画面右上の検索ボックスをご利用ください。

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 134003 | ERROR | catch a \"$sig\" signal | **原因**: create処理がシグナルによって中断されました。<br><br>**処置**: 前後のログメッセージを確認してください。 |
| 134004 | ERROR | Unable to getlocks on $server during resource create. Error ($rc) | **原因**: リソース階層の作成時に管理ロックの取得に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 134005 | ERROR | The service \"$serviceName\" is not supported on $server. Error ($rc) | **原因**: サービスが存在しないか保護できないサービスです。<br><br>**処置**: 適切なサービス名を入力してください。 |
| 134006 | ERROR | The service \"$serviceName\" is already protected on $server. | **原因**: すでに保護されているサービスです。<br><br>**処置**: このサービスを保護するためのリソースを作成することはできません。 |
| 134007 | ERROR | Error creating resource $tag. Error ($rc) | **原因**: QSPリソースの作成に失敗しました。<br><br>**処置**: 前後のログメッセージを確認して、エラーの原因を取り除いてくださ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | い。 |
| 134011 | ERROR | In-service attempted failed for tag $tag. | **原因**: QSPリソースの起動に失敗しました。<br><br>**処置**: 保護したいサービスに関連したログを確認して、問題を解決してください。 |
| 134015 | ERROR | Unable to rlslocks on $server during resource create. Error ($rc) | **原因**: QSPリソースの作成後にロックの解除に失敗しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 134103 | ERROR | Template resource \"$template_tag\" on server \"$template_sys\" does not exist | **原因**: テンプレートシステムでリソースを見つけられません。<br><br>**処置**: テンプレートシステムで階層が正しいことを確認してから拡張してください。 |
| 134104 | ERROR | Template resource \”$template_tag\” on server \”$template_sys\” is not QSP resource (app=$ins[1], res=$ins[2]) | **原因**: テンプレートリソースはQSPリソースではありません。<br><br>**処置**: テンプレートリソースと同じタイプに拡張してください。 |
| 134105 | ERROR | The service \"$service\" is not supported on $me. Error ($check) | **原因**: 拡張先のサーバにサービスが存在しません。<br><br>**処置**: 拡張先サーバにサービスをインストールしてから拡張してください。 |
| 134106 | ERROR | The service \"$service\" is already protected on $me. | **原因**: 拡張先サーバに同じIDのリソース |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | が既にあります。<br><br>**処置**: 同じサービスのリソースを作成することはできません。 |
| 134203 | ERROR | catch a \"$sig\" signal | **原因**: リソース拡張処理がシグナルによって中断されました。<br><br>**処置**: 前後のログメッセージを確認してください。 |
| 134204 | ERROR | Template resource \"$template_tag\" on server \"$template_sys\" does not exist | **原因**: テンプレートシステムでリソースを見つけられません。<br><br>**処置**: テンプレートシステムで階層が正しいことを確認してから拡張してください。 |
| 134208 | ERROR | Error creating resource \"$tag\" on server \"$me\" | **原因**: LifeKeeperはターゲットサーバーにリソースインスタンスを作成できませんでした。<br><br>**処置**: 前後のログメッセージを確認して、エラーの原因を取り除いてください。 |
| 134401 | ERROR | timeout $cmd for \"$tag\". Forcibly terminating. | **原因**: サービスの起動処理が規定時間を超えても終了しません。<br><br>**処置**: 保護しているサービスについて確認し、restore操作を再試行してください。また、関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 134405 | FATAL | Failed to fork process to execute service command: $! | **原因**: forkが失敗しました。これはシステムエラーです。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: forkが失敗する理由を特定してください。 |
| 134407 | ERROR | service command has failed for \"$tag\" | **原因**: serviceコマンドの実行に失敗しました。<br><br>**処置**: startオプションでserviceコマンドを手動実行するとエラーになります。エラーメッセージを参考にしてエラーの原因を取り除いてください。 |
| 134501 | ERROR | timeout $cmd for \"$tag\". Forcibly terminating. | **原因**: サービスの停止処理が規定時間を超えても終了しません。<br><br>**処置**: 保護しているサービスについて確認し、remove操作を再試行してください。また、関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 134505 | FATAL | Failed to fork process to execute service command: $! | **原因**: forkが失敗しました。これはシステムエラーです。<br><br>**処置**: forkが失敗する理由を特定してください。 |
| 134507 | ERROR | service command has failed for \"$tag\" | **原因**: serviceコマンドの実行に失敗しました。<br><br>**処置**: stopオプションでserviceコマンドを手動実行するとエラーになります。エラーメッセージを参考にしてエラーの原因を取り除いてください。 |
| 134601 | ERROR | timeout $cmd for \"$tag\". Forcibly terminating. | **原因**: サービスの監視処理が規定時間を超えても終了しません。<br><br>**処置**: 保護しているサービスについて確 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 認してください。また、関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 134605 | FATAL | Failed to fork process to execute service command: $! | 原因: forkが失敗しました。これはシステムエラーです。<br><br>処置: forkが失敗する理由を特定してください。 |
| 134607 | ERROR | service command has failed for \"$tag\" | 原因: serviceコマンドの実行に失敗しました。<br><br>処置: statusオプションでserviceコマンドを手動実行するとエラーになります。エラーメッセージを参考にしてエラーの原因を取り除いてください。 |
| 134701 | ERROR | timeout $cmd for \"$tag\". Forcibly terminating. | 原因: サービスの回復処理が規定時間を超えても終了しません。<br><br>処置: 保護しているサービスについて確認してください。また、関連するエラーをログで確認し、レポートされている問題を解決してください。 |
| 134706 | FATAL | Failed to fork process to execute service command: $! | 原因: forkが失敗しました。これはシステムエラーです。<br><br>処置: forkが失敗する理由を特定してください。 |
| 134708 | ERROR | service command has failed for \"$tag\" | 原因: serviceコマンドの実行に失敗しました。<br><br>処置: startオプションでserviceコマンドを手動実行するとエラーになります。エラーメッセージを参考にしてエラーの原 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 因を取り除いてください。 |
| 134803 | ERROR | tag \"$tag\" does not exist on server \"$me\" | 原因: 指定されたタグが存在しません。これは内部エラーです。 |
| 134804 | ERROR | app type \”$ins[1]\” is not $app | 原因: 指定されたタグはQSPリソースではありません。これは内部エラーです。 |
| 134805 | ERROR | res type \”$ins[2]\” is not $res | 原因: 指定されたタグはQSPリソースではありません。これは内部エラーです。 |
| 134823 | ERROR | tag \"$tag\" does not exist on server \"$me\" | 原因: 指定されたタグが存在しません。これは内部エラーです。 |
| 134824 | ERROR | app type \”$ins[1]\” is not $app | 原因: 指定されたタグはQSPリソースではありません。これは内部エラーです。 |
| 134825 | ERROR | res type \”$ins[2]\” is not $res | 原因: 指定されたタグはQSPリソースではありません。これは内部エラーです。 |
| 134843 | ERROR | tag \"$tag\" does not exist | 原因: 指定されたタグが存在しません。これは内部エラーです。 |
| 134844 | ERROR | app type \”$ins[1]\” is not $app | 原因: 指定されたタグはQSPリソースではありません。これは内部エラーです。 |
| 134845 | ERROR | res type \”$ins[2]\” is not $res | 原因: 指定されたタグはQSPリソースではありません。これは内部エラーです。 |

# 8.1.15. Quorum メッセージカタログ

**Control F** を使用して、各カタログの特定のエラーコードを検索します。エラーコードの検索には、画面右上の検索ボックスをご利用ください。

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 135002 | ERROR | Socket connection to $host failed (not a timeout): $@ | |
| 135802 | ERROR | Periodic storage quorum check has not been done for %lld ms, which exceeds 2*QWK_STORAGE_HBEATTIME (%lld) ms. | |
| 135806 | ERROR | qwk_config() failed. | |
| 135807 | ERROR | thread_initialize() failed. | |
| 135808 | ERROR | state_monitor_initialize() failed. | |
| 135809 | ERROR | state_monitor() failed. | |
| 135810 | ERROR | start_server() failed. | |
| 135812 | ERROR | ‘fopen’ for %s failed: %m | |
| 135813 | ERROR | ‘qwk_object_path=’ line cannot be found for %s. | |
| 135814 | ERROR | ‘%s’ is unknown config. | |
| 135815 | ERROR | configuration of hbeattime ‘%d’ is incorrect. | |
| 135816 | ERROR | configuration of numhbeats ‘%d’ is incorrect. | |
| 135817 | ERROR | configuration of timeout_multiplier ‘%d’ is incorrect. | |
| 135818 | ERROR | configuration of lcmhbeattime ‘%d’ is incorrect. | |
| 135819 | ERROR | configuration of lcmnumhbeats ‘%d’ is incorrect. | |
| 135820 | ERROR | configuration of qwk_object_type is incorrect. | |
| 135821 | ERROR | configuration of my_node is incorrect. | |
| 135822 | ERROR | configuration of number_of_node ‘%d’ is incorrect. | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 135823 | ERROR | configuration of number_of_object ‘%d’ is incorrect. | |
| 135824 | ERROR | configuration of object node is incorrect. | |
| 135825 | ERROR | configuration of object path is incorrect. | |
| 135826 | ERROR | my_node is not include in qwk_objects. | |
| 135827 | ERROR | ‘open’ for %s failed: %m | |
| 135828 | ERROR | ‘read’ for %s failed: %m | |
| 135829 | ERROR | ‘popen’ for %s failed: %m | |
| 135830 | ERROR | ‘open’ for %s failed: %m | |
| 135831 | ERROR | ‘write’ for %s failed: %m | |
| 135832 | ERROR | ‘popen’ for %s failed: %m | |
| 135833 | ERROR | (bug) buffer overflow | |
| 135834 | ERROR | (bug) data is corrupted | |
| 135835 | ERROR | ‘signature=’ line cannot be found. | |
| 135836 | ERROR | signature ‘%s’ does not match. | |
| 135837 | ERROR | ‘local_node=’ line cannot be found. | |
| 135838 | ERROR | local_node ‘%s’ does not match. | |
| 135839 | ERROR | ‘time=’ line cannot be found. | |
| 135840 | ERROR | ‘sequence=’ line cannot be found. | |
| 135841 | ERROR | sequence ‘%s’ scan failed. | |
| 135842 | WARN | ‘node=’ line cannot be found. index=%d | |
| 135843 | ERROR | ‘commstat=’ line cannot be found. index=%d | |
| 135844 | ERROR | ‘checksum=’ line cannot be found. | |
| 135845 | ERROR | checksum ‘%s’ scan failed. | |
| 135846 | ERROR | checksum does not match. | |
| 135849 | ERROR | qwk object was not found. | |
| 135851 | ERROR | failed to read qwk object. | |
| 135852 | ERROR | failed to decode node_info. | |
| 135854 | WARN | sequence backed down from %llu to %llu. | |
| 135855 | ERROR | ‘malloc’ for %zu failed: %m | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 135856 | ERROR | thread_create() failed. index=%d | |
| 135870 | ERROR | (bug) data is corrupted | |
| 135871 | ERROR | format error in request. | |
| 135872 | ERROR | format error in quorum_verify request. lkevent cannot be found. | |
| 135873 | ERROR | format error in witness_verify request. lkevent cannot be found. | |
| 135874 | ERROR | format error in witness_verify request. target_node cannot be found. | |
| 135875 | ERROR | ‘%s’ is unknown command. | |
| 135877 | ERROR | qwk_receive() did not receive full header. Close socket. | |
| 135878 | ERROR | Request is too long. Close socket. | |
| 135879 | ERROR | qwk_receive() did not receive full request. Close socket. | |
| 135880 | ERROR | do_request() failed. | |
| 135881 | ERROR | qwk_send() did not send full header. Close socket. | |
| 135882 | ERROR | qwk_send() did not send full response. Close socket. | |
| 135884 | ERROR | qwk_accept() failed. | |
| 135885 | ERROR | thread_create() failed. | |
| 135886 | ERROR | cannot create socket. | |
| 135887 | ERROR | ‘bind’ failed: %m | |
| 135888 | ERROR | ‘listen’ failed: %m | |
| 135889 | ERROR | create_sockets() failed. | |
| 135890 | ERROR | thread_create() failed. | |
| 135891 | ERROR | ‘pthread_attr_init’ failed: %m | |
| 135892 | ERROR | ‘pthread_attr_setstacksize’ failed: %m | |
| 135893 | ERROR | ‘pthread_create’ failed: %m | |
| 135894 | ERROR | requst is too long. header.size=%zu | |
| 135895 | ERROR | qwk_send() failed for header. | |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 135896 | ERROR | qwk_send() failed for request. | |
| 135897 | ERROR | qwk_send() failed for termination. | |
| 135898 | ERROR | qwk_receive() did not receive full header. | |
| 135899 | ERROR | Response buffer is not enough large. Server sent %zu bytes. | |
| 135900 | ERROR | qwk_receive() did not receive full response. | |
| 135901 | ERROR | ‘%s’ is unknown command. | |
| 135903 | ERROR | Cannot create socket. | |
| 135904 | ERROR | ‘connect’ failed: %m | |
| 135905 | ERROR | request_send() failed. | |
| 135906 | ERROR | request_receive() failed. | |
| 135907 | ERROR | ‘%s’ is unknown lkevent. | |
| 135908 | ERROR | ‘%s’ is unknown qwktype. | |
| 135909 | ERROR | ‘%s’ is unknown node state. | |
| 135910 | ERROR | ‘%s’ is unknown quorum state. | |
| 135911 | ERROR | ‘accept’ failed: %m | |
| 135912 | ERROR | You must install the LifeKeeper license key for Storage Quorum Witness Kit. | |
| 135913 | ERROR | Cannot find the configuration for node ‘%s’ in the configuration file. If the configuration of the cluster is changed after the initialization, it needs to be initialized again. Please see the SIOS product documentation for information on how to reinitialize using qwk_storage_init command. | |
| 135914 | WARN | The first cycle did not complete before timeout (%lld ms). | 原因: Storage モードの Quorum オブジェクトへの最初の書き込みが、指定された時間（ミリ秒）でタイムアウトしました。<br><br>処置: QWK_STORAGE_HBEATTIME と QWK_STORAGE_NUMHBEATS に大きな値を設定します。(どちらか一方また |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | は両方)<br><br>このメッセージが出力された後、135802 メッセージが頻繁に出力される場合は、Quorum/Witness の Storage モードページのトラブルシューティングの項目を参照してください。 |
| 135916 | WARN | The first cycle did not complete before timeout (%lld ms). | **原因**: Storage モードの Quorum オブジェクトへの最初の書き込みが、指定された時間（ミリ秒）でタイムアウトしました。<br><br>**処置**: QWK_STORAGE_HBEATTIME と QWK_STORAGE_NUMHBEATS に大きな値を設定します。(どちらか一方または両方)<br><br>このメッセージが出力された後、135802 メッセージが頻繁に出力される場合は、Quorum/Witness の Storage モードページのトラブルシューティングの項目を参照してください。 |
| 135917 | ERROR | params is NULL. return 0 as timeout msec. | **原因**: 内部処理に問題が発生しました。<br><br>**処置**: サポートにお問い合わせください。 |
| 135918 | ERROR | s3_read_qwk_object for %s failed: %d (%s | |
| 135919 | ERROR | s3_write_qwk_object for %s failed: %d (%s | |
| 135920 | ERROR | %s is empty | |
| 135921 | ERROR | mkstemp for %s failed: %m | |
| 135922 | ERROR | mkstemp for %s failed: %m | |
| 135998 | EMERG | An unrecoverable error was detected in qwk_storage_server. Stop restarting qwk_storege_server. | |

| コード | 重大度 | メッセージ | 原因/処置 |
| --- | --- | --- | --- |
| 135999 | ERROR | -c $cmd | |

# 8.1.16. GUI メッセージカタログ

**Control F** を使用して、各カタログの特定のエラーコードを検索します。エラーコードの検索には、画面右上の検索ボックスをご利用ください。

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 000200 | ERROR | pam_start() failed | |
| 000201 | ERROR | pam_authenticate failed (user %s, retval %d) | |
| 000202 | ERROR | pam_end() failed?!?! | |
| 000203 | ERROR | Did not find expected group ‘lkguest’ | |
| 000204 | ERROR | Did not find expected group ‘lkoper’ | |
| 000205 | ERROR | Did not find expected group ‘lkadmin’ | |
| 000208 | ERROR | pam_setcred establish credentials failed (user %s, retval %d) | **原因**: ユーザー {user} の有効なログイン資格情報を確立できません。pam_setcred 呼び出しで返された値は次の通りです：{retval}<br><br>**処置**: 詳細については、/var/log/security および /var/log/messages を確認して下さい。 |
| 000209 | ERROR | pam_setcred delete credentials failed (user %s, retval %d) | **原因**: ユーザー {user} の有効なログイン資格情報を確立できません。pam_setcred 呼び出しで返された値は次の通りです：{retval}<br><br>**処置**: 詳細については、/var/log/security および /var/log/messages を確認して下さい。 |
| 000902 | ERROR | Error removing system name from loopback address line in /etc/hosts file. You must do this manually before starting the GUI server. | **原因**: システム名が /etc/hosts ファイルから削除されていません。<br><br>**処置**: システム名を手動で削除し、GUI サーバを再起動して、次のコマンドを入力します: run <action name> |

| コード | 重大度 | メッセージ | 原因/処置 |
| --- | --- | --- | --- |
| 000918 | ERROR | LifeKeeper GUI Server error during Startup | **原因**: 異常な状態により、GUI サーバが終了しました。<br><br>**処置**: 関連するエラーをログで確認し、レポートされている問題を解決してください。 |

# 8.1.17. SAP メッセージカタログ

**Control F** を使用して、各カタログの特定のエラーコードを検索します。エラーコードの検索には、画面右上の検索ボックスをご利用ください。

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 112000 | FATAL | Usage: "%s %s". Specify the correct usage for the requested command. | 原因: SAP HANA createスクリプトに無効なパラメーターが指定された。<br><br>処置: 適切なパラメーターをSAP HANA createスクリプトに指定してください。 |
| 112004 | ERROR | Neither the SID/instance pair nor the tag parameter were specified for the internal "%s" routine on %s. If this was a command line operation, specify the correct parameters. Otherwise, consult the troubleshooting documentation. | 原因: 内部SAPオブジェクトの作成において無効なパラメーターが指定された。<br><br>処置: SIDおよびインスタンスまたはLifeKeeper のリソースタグ名のいずれかを指定してください。 |
| 112013 | ERROR | The file "%s" does not exist or was not readable on %s. Verify that the specified file exists and/or is readable. | 原因: sapservices ファイルが存在しないか、読み込み不可。<br><br>処置: sapservices ファイルが存在し、読み込み可能であることを確認してください。 |
| 112017 | ERROR | Error getting resource information for resource "%s" on server "%s" | 原因: 指定したサーバー上で指定したリソースに関連する情報を見つけることができない。<br><br>処置: 指定したサーバーに指定したリソースが存在していて、すべての必要なファイルシステムがマウントされているかを確認してください。 |
| 112019 | ERROR | The attempt to update the resource information field for resource %s has failed on %s. View the resource properties manually using "ins_list -t <tag>" to verify that the resource is | 原因: 指定したサーバーで指定したリソースの LifeKeeper リソース情報フィールドを更新できない。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | functional. | **処置**: 指定したサーバーにリソースが存在しているか、もしくはLifeKeeper が起動しているかを確認してください。 |
| 112022 | ERROR | An error occurred while trying to find the IP address corresponding to "%s" on %s. Verify the IP address or host name exists in DNS or the hosts file. | **原因**: 指定したサーバーにIPアドレスもしくはホスト名が見つからない。<br><br>**処置**: IPアドレスもしくはDNS名がDNSもしくはローカルホストファイルに存在しているかを確認してください。 |
| 112024 | FATAL | There was an error verifying the NFS connections for SAP related mount points on $me. One or more NFS servers is not operational and needs to be restarted. | **原因**: SAP_NFS_CHECK_DIRS パラメーター に掲載されている少なくとも一つの NFS 共有ファイルシステムを現在使用できない。<br><br>**処置**: NFSサーバーの起動、すべての必要とされるNFS関連のサービスの起動、すべての必要とされるファイルシステムがエクスポートされていることを確認してください。 |
| 112028 | ERROR | Unable to determine the user name for the SAP administrative user for resource %s on %s. | **原因**: 指定したサーバー上に指定したリソース情報を確認することができない。<br><br>**処置**: 指定したサーバーに指定したリソースが存在していて、すべての必要なファイルシステムがマウントされているかを確認してください。 |
| 112029 | ERROR | The canextend script "%s" either does not exist or is not executable on %s. | **原因**: SAP canextend スクリプトが存在していないか、実行できない。<br><br>**処置**: SAP canextend スクリプトが存在していて、実行可能か確認してください。 |
| 112037 | ERROR | Unable to create an internal object for the SAP instance using SID %s, instance %s, | **原因**: 指定した SAP インスタンスを表す |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | and tag %s on server %s. Verify that all necessary SAP file systems are mounted and accessible before reattempting the operation. | ための内部の SAP オブジェクトを作成することができない。<br><br>処置: SAP インスタンスが正常にインストール、設定されていて、すべての必要なファイルシステムがマウントされているかを確認してください。 |
| 112040 | ERROR | The SAP Directory "%s" ("%s") does not exist on %s. Verify that the directory exists and that the SAP software is properly installed. | 原因: 指定したSAP インストールディレクトリが指定したサーバーに存在していない。<br><br>処置: SAP ソフトウェアが正常にインストールされすべての必要となるファイルシステムがマウントされていることを確認してください。 |
| 112041 | ERROR | The required utility "%s" was not found or was not executable on %s. Verify the SAP installation and location of the required utility. | 原因: saphostexec または saposcol が見つからないか実行できない。<br><br>処置: SAP Host Agent package が正しくインストールされ、すべての必要となるファイルシステムがマウントされていることを確認してください。 |
| 112042 | ERROR | One or more SAP or LifeKeeper validation checks has failed on %s. Please update the SAP software on this host to include the SAPHOST and SAPCONTROL packages. | 原因: saphostexec または saposcol が見つからないか実行できない。<br><br>処置: SAP Host Agent package が正しくインストールされ、すべての必要となるファイルシステムがマウントされていることを確認してください。 |
| 112048 | ERROR | The SAP instance %s is already under LifeKeeper protection on server %s. Choose another SAP instance to protect or specify the correct instance. | 原因: 指定した SAP インスタンスはすでに指定したサーバーでLifeKeeper により保護されている。<br><br>処置: まだLifeKeeper により保護されていない SAP インスタンスを選択してく |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ださい。 |
| 112049 | ERROR | Unable to locate the SAP Mount directory on %s. Verify that all SAP file systems are mounted and accessible before reattempting the operation. | **原因**: SAP のマウント (sapmnt) ディレクトリの場所を特定できない。<br><br>**処置**: すべての必要となるファイルシステムがマウントされ、すべての必要となる SAP インスタンスプロファイルがアクセス可能になっていることを確認してください。 |
| 112050 | ERROR | Detected multiple virtual IP addresses/ host names for instance %s on %s. Verify that the instance is configured correctly. | **原因**: 指定したサーバー上に指定した SAP インスタンスに対する複数の仮想 IP もしくはホスト名が検出された。<br><br>**処置**: インスタンスに関連する仮想 IP もしくはホスト名が正しく構成されているかを確認してください。 |
| 112051 | ERROR | The "%s" or "%s" value in the default profile "%s" is still set to the physical host name on %s. The value(s) must be set to a virtual host name. | **原因**: 指定したサーバーで指定したホスト名のパラメーターが物理サーバーのホスト名に指定されている。<br><br>**処置**: ホスト名のパラメーターは仮想ホスト名に設定してください。 |
| 112053 | ERROR | Detected multiple instances under SID %s with the same instance number (%s) on %s. Each instance within a particular SID must have a unique instance number. | **原因**: 同じインスタンス番号のSAP インスタンスが、同じ SAP SID 下に検出された。<br><br>**処置**: SAP環境を再構築することで、指定した SAP SID 下の各インスタンスが一意のインスタンス番号をもつことができます。 |
| 112056 | ERROR | The NFS export for the path "%s" required by the instance %s for the "%s" directory does not have an NFS hierarchy protecting it on %s. You must create an | **原因**: 指定したファイルシステムのNFS エクスポートが LifeKeeper により保護されていません。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | NFS hierarchy to protect this NFS export before creating the SAP resource hierarchy. | **処置**: エクスポートしたファイルシステムの LifeKeeper NFS 階層を作成し、SAP リソースを作成してください。 |
| 112057 | ERROR | Unable to create a file system resource hierarchy for the file system “%s” on “%s”. | **原因**: 指定したサーバーで指定したファイルシステムを保護するための LifeKeeper ファイルシステムリソース階層を作成することができない。<br><br>**処置**: 詳細について LifeKeeper とシステムログを確認してください。 |
| 112058 | ERROR | Unable to create a dependency between parent tag “%s” and child tag “%s” on “%s”. | **原因**: 指定したサーバーで指定したリソース間での依存関係を作成できない。<br><br>**処置**: 詳細について LifeKeeper のログを確認してください。 |
| 112060 | ERROR | All attempts at local recovery for the SAP resource “%s” have failed on “%s”. A failover to the backup server will be attempted. | **原因**: 指定したサーバーで指定した SAP リソースを復旧できない。<br><br>**処置**: SAP リソース階層のフェイルオーバーは自動的に行われます。ユーザーの介入は必要ありません。 |
| 112061 | ERROR | The values specified for the target and the template servers are the same. Please specify the correct values for the target and template servers. | **原因**: SAP リソース階層の拡張中に提供されたテンプレート/ターゲットサーバーが同一。<br><br>**処置**: テンプレート・ターゲットサーバーの正しい名前を提供し、拡張処理を再度実行してください。 |
| 112062 | ERROR | Unable to find the home directory "%s" for the SAP administrative user "%s" on %s. Verify that the SAP software is installed correctly. | **原因**: 指定したサーバーで指定した SAP ユーザーのホームディレクトリが見つからない。<br><br>**処置**: SAP ソフトウェアが正常にインス |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | トールされ、指定した SID に対する適切な SAP 管理ユーザーがサーバーに存在していることを確認してください。 |
| 112063 | ERROR | The SAP administrative user "%s" does not exist on %s. Verify that the SAP software is installed correctly or create the required SAP user on %s. | 原因: 指定した SAP 管理ユーザーが指定したサーバー上に存在していない。<br><br>処置: SAP ソフトウェアが正常にインストールされているか確認し、必要に応じて SAP 管理ユーザーを作成してください。 |
| 112064 | ERROR | The group ID for user "%s" is not the same on template server "%s" and target server "%s". Please correct the group ID for the user so that it is the same on the template and target servers. | 原因: 指定したSAP 管理ユーザーのグループ IDがテンプレート/ターゲットサーバーで一致していない。<br><br>処置: 指定した SAP 管理ユーザーのグループ ID を修正することで、テンプレート/ターゲットサーバーで同一となります。 |
| 112065 | ERROR | The user ID for user "%s" is not the same on template server "%s" and target server "%s". Please correct the user ID so that it is the same on the template and target servers. | 原因: 指定した SAP 管理ユーザーの ID がテンプレート/ターゲットサーバーで一致していない。<br><br>処置: 指定した SAP 管理ユーザーの ID を修正することで、テンプレート/ターゲットサーバーで同一となります。 |
| 112066 | ERROR | Required SAP utilities could not be found in "%s" on %s. Verify that the SAP software is installed correctly. | 原因: 必要な SAP 実行ファイルもしくは SAP インスタンスプロファイルを配置することができない。<br><br>処置: SAP ソフトウェアが正常にインストールされ構成されていて、すべての必要なファイルシステムがマウントされているか確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 112069 | ERROR | The command “%s” is not found in the “%s” perl module (“%s”) on “%s”. Please check the command specified and retry the operation. | **原因**: 指定したコマンドが指定したサーバーの sap perl modedule に見つからない。<br><br>**処置**: ユーザーから開始したコマンドアクションによりエラーとなった場合は、正しいルーチン名が remoteControl スクリプトに渡されているかを確認してください。通常のLifeKeeper 運用において発生したエラーの場合は、製品サポートへお問い合わせください。 |
| 112071 | ERROR | The file “%s” exists, but was not read and write enabled on “%s”. Enable read and write permissions on the specified file | **原因**: 指定したファイルが指定したサーバーで読み込み/書き込み権限をもっていない。<br><br>**処置**: 指定したサーバーで読み込み/書き込み権限を有効にしてください。 |
| 112073 | ERROR | Unable to create an internal object for the SAP instance using either SID "%s" and instance "%s" or tag "%s" on server "%s". Either the values specified for the object initialization (SID/instance pair or tag, system) were not valid, or an error occurred while attempting to gather information about the SAP instance. If all specified parameters are correct, verify that all necessary SAP file systems are mounted and accessible before reattempting the operation. | **原因**: 指定した SAP インスタンスを示すための内部 SAP オブジェクトを作成することができない。<br><br>**処置**: SAP インスタンスが正常にインストール、設定されていて、すべての必要なファイルシステムがマウントされているかを確認してください。 |
| 112074 | WARN | Warning: The profile file %s for SID %s and Instance %s has Autostart is enabled on %s. Disable Autostart for the specified instance by setting Autostart=0 in the profile file | **原因**: Autostart パラメーターが指定したサーバーの指定したインスタンスプロファイルで有効になっている。<br><br>**処置**: 指定した SAP インスタンスがインスタンスプロファイルの中で ‘Autostart = 0’に設定されており、Autostartが有効になっています。Autostartを無効にしてください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 112076 | FATAL | Unable to start the sapstartsrv service for SAP SID $sid and SAP instance $Instance on $me. Verify that the sapservices file is correct and the process can be started manually. | 原因: 指定した SAP インスタンスに対して SAP Start Service (sapstartsrv) のプロセスを開始できない。<br><br>処置: sapservices ファイルが sapstartsrv プロセスを開始するための適切なコマンドを含んでいて、そのプロセスを手動で開始できるかを確認してください。 |
| 112077 | ERROR | Unable to stop the sapstartsrv service for SAP SID %s and SAP instance %s on %s. Verify that the sapservices file is correct and the process can be stopped manually. | 原因: 指定した SAP インスタンスに対して SAP Start Service (sapstartsrv) プロセスを停止することができない。<br><br>処置: sapservices ファイルが sapstartsrv プロセスを開始するための適切なコマンドを含んでいて、そのプロセスを手動で開始できるかを確認してください。 |
| 112078 | ERROR | ERSv1 is only supported in two-node clusters. Resource %s is unable to be extended to system %s. Upgrade to ERSv2 in order to extend the hierarchy to three or more nodes. | 原因: ERSv1インスタンスを表す SAP リソースを 3つ以上のノードに拡張することができない。<br><br>処置: ERS インスタンスを表す SAP リソースを3つ以上のノードに拡張する場合は、ERSv2 へアップグレードしてください。 アップグレード手順は、オンラインプロダクトドキュメント内に記載されています。 |
| 112082 | WARN | Instance %s is running a different version of the enqueue server than its corresponding enqueue replication server. This configuration is not supported by SAP and will lead to unexpected resource behavior. See SAP Note 2711036 – "Usage of the Standalone Enqueue Server 2 in an HA Environment" for more details. Please review the online product documentation for instructions on how to modify the instance profiles for the | 原因: エンキューサーバーとエンキューレプリケーションサーバーのバージョンが一致していない。<br><br>処置: オンラインプロダクトドキュメントに記載のインスタンスプロファイルの修正方法を確認の上、エンキューサーバーとエンキューレプリケーションサーバーのバージョンを一致させてください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | enqueue server and enqueue replication server so that they use the same version. | |
| 112086 | ERROR | The ERS resource corresponding to resource %s is in-service and maintaining backup locks on a remote system. Bringing resource %s in-service on %s would result in a loss of the backup lock table. Please bring resource %s in-service on the system where the corresponding ERS resource is currently in-service in order to maintain consistency of the lock table. In order to force resource %s in-service on %s, either (i) run the command \'/opt/LifeKeeper/bin/flg_create -f sap_cs_force_restore_%s\' as root on %s and reattempt the in-service operation or (ii) take the corresponding ERS resource out of service on the remote system. Both of these actions will result in a loss of the backup lock table. | 原因: ASCS/SCS リソースに対応する ERS インスタンスがクラスタ内の異なるサーバーで起動し、バックアップのエンキューテーブルを保存していたので、ASCS/SCS リソースに対して in-service 処理が試された。<br><br>処置: バックアップのエンキューテーブルを取得するためにERSインスタンスが起動しているサーバーで、ASCS/SCS リソースを in-sevice にしてください。ASCS/SCS リソースを強制的に in-service にする場合は、 '/opt/LifeKeeper/bin/flg_create -f sap_cs_force_restore_<ASCS/SCS Tag>' コマンドを実行し in-service 処理を試みるか、またはリモートサーバーで対応する ERS リソースを out of service にしてください。これら両方の操作を実行すると、バックアップのエンキューテーブルが失われます。 |
| 112089 | ERROR | The internal "%s" routine was called for a resource with unsupported instance type %s. This method supports only SAP instance types TYPE_CS, TYPE_ERS, or TYPE_NEW_ERS (1, 2, or 5). | 原因: GetEnqVersion ルーチンがサポートされていない SAP インスタンスのタイプに対して呼び出された。インスタンスタイプ 1 (TYPE_CS), 2 (TYPE_ERS), 5 (TYPE_NEW_ERS) のみがサポート対象です。<br><br>処置: こちらは内部エラーです。SIOS カスタマーセンターへお問い合わせください。 |
| 112092 | ERROR | The profile "%s" either does not exist or cannot be read on %s. Unable to determine whether enqueue replication is enabled for resource %s. Please verify that the file exists and can be read. | 原因: 指定したサーバー上に指定したインスタンスプロファイルが存在していない、もしくは読み取りができない。<br><br>処置: ファイルが存在していて、読み取り可能かどうかを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 112095 | ERROR | Error creating resource “%s” on server “%s” | **原因**: 指定したサーバーに指定したリソースを作成することができない。<br><br>**処置**: 指定したサーバーにSAP ソフトウェアが正常にインストールされ、すべての必要なファイルシステムがマウントされアクセス可能になっていることを確認してください。 |
| 112096 | ERROR | Resource %s is not currently in-service on server %s. Manually bring the resource in-service and retry the operation. | **原因**: 依存関係の作成中に、指定したリソースがサーバーで In Service ではありませんでした。<br><br>**処置**: 指定したサーバーでリソースを In Service にし、再度操作を実施してください。 |
| 112101 | ERROR | Error getting resource information for resource “%s” on server “%s” | **原因**: 指定したサーバー上の指定したリソース情報を確認することができない。<br><br>**処置**: 指定したサーバーに指定したリソースが存在していてすべての必要なファイルシステムがマウントされているかを確認してください。 |
| 112102 | ERROR | Cannot extend resource “%s” to server “%s” | **原因**: 指定したサーバーへ拡張したリソースを作成することができない。<br><br>**処置**: SAP ソフトウェアがターゲットサーバーに正常にインストールされ、すべての必要なファイルシステムがターゲットシステムからアクセス可能になっていることを確認してください。 |
| 112103 | ERROR | Error creating resource “%s” on server “%s” | **原因**: 指定したサーバーに指定したリソースを作成することができない。<br><br>**処置**: 指定したサーバーにSAP ソフトウェアが正常にインストールされ、すべて |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | の必要なファイルシステムがマウントされアクセス可能になっていることを確認してください。 |
| 112104 | ERROR | The extend script "%s" either does not exist or is not executable on %s. | **原因**: 指定したサーバーに指定した extend スクリプトが存在しないか実行することができない。<br><br>**処置**: すべての必要なリカバリーキットがインストールされ、extend スクリプトが実行可能か確認してください。 |
| 112106 | ERROR | Unable to create an internal SAP object for resource "%s" on %s. If the tag is correct, verify that all necessary SAP file systems are mounted and accessible. | **原因**: 指定したサーバー上の指定したリソース情報を確認することができませんでした。<br><br>**処置**: 指定したサーバーに指定したリソースが存在していてすべての必要なファイルシステムがマウントされているかを確認してください。 |
| 112112 | ERROR | Error getting resource information for resource “%s” on server “%s” | **原因**: 指定したサーバー上の指定したリソース情報を確認することができませんでした。<br><br>**処置**: 指定したサーバーに指定したリソースが存在していてすべての必要なファイルシステムがマウントされているかを確認してください。 |
| 112115 | FATAL | Error getting resource information for resource \”$tag\” on server \”$sap::me\”. | **原因**: 指定したサーバー上の指定したリソース情報を確認することができませんでした。<br><br>**処置**: 指定したサーバーに指定したリソースが存在していてすべての必要なファイルシステムがマウントされているかを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 112119 | ERROR | Error getting resource information for resource “%s” on server “%s” | **原因**: 指定したサーバー上の指定したリソース情報を確認することができませんでした。<br><br>**処置**: 指定したサーバーに指定したリソースが存在していてすべての必要なファイルシステムがマウントされているかを確認してください。 |
| 112123 | ERROR | Error getting resource information for resource “%s” on server “%s” | **原因**: 指定したサーバー上の指定したリソース情報を確認することができませんでした。<br><br>**処置**: 指定したサーバーに指定したリソースが存在していてすべての必要なファイルシステムがマウントされているかを確認してください。 |
| 112124 | ERROR | Resource with either matching tag “%s” or id “%s” already exists on server “%s” for App “%s” and Type “%s” | **原因**: 指定したリソースタグ名もしくはID および同一のアプリケーション、タイプをもつリソースがすでに指定したサーバーに存在しています。<br><br>**処置**: SAP インスタンスが指定したサーバーでまだLifeKeeper に保護されていないことを確認してください。まだ保護されていない場合は、別のリソースタグ名を選択してください。 |
| 112125 | ERROR | Unable to create an SAP object for resource %s on system %s. | **原因**: 指定のサーバーで指定されたリソースを示すための SAP オブジェクトを作成することができない。<br><br>**処置**: 指定したサーバーにSAP ソフトウェアが正常にインストールされ、すべての必要なファイルシステムがマウントされアクセス可能になっていることを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 112126 | FATAL | Usage: “%s %s” Specify the correct usage for the requested command. | 原因: SAP canextend スクリプトの使用に誤りがあります。<br><br>処置: canextend スクリプトに対する正しいパラメーターを指定してください。: canextend <template server> <template tag> |
| 112127 | FATAL | Usage: \”$cmd $usage\” Specify the correct usage for the requested command. | 原因: SAP delete スクリプトの使用に誤りがある。<br><br>処置: SAP delete スクリプトに対する正しいパラメーターを指定してください。: delete [-U] -t <tag> -i <id> |
| 112128 | FATAL | Usage: “%s %s” Specify the correct usage for the requested command. | 原因: SAP depstoextend スクリプトの使用に誤りがある。<br><br>処置: depstoextend スクリプトに対する正しいパラメーターを指定してください。: depstoextend <template server> <template tag> |
| 112129 | FATAL | Usage: “%s %s” Specify the correct usage for the requested command. | 原因: SAP extend スクリプトの使用に誤りがある。<br><br>処置: extend スクリプトに対する正しいパラメーターを指定してください。: extend <template server> <template tag> <switchback> <target tag> |
| 112130 | FATAL | Usage: \”$cmd $usage\” Specify the correct usage for the requested command. | 原因: SAP quickCheck スクリプトの使用に誤りがある。<br><br>処置: quickCheck スクリプトに対する正しいパラメーターを指定してください。: quickCheck -t <tag> -i <id> |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 112131 | FATAL | Usage: “%s %s” Specify the correct usage for the requested command. | 原因: SAP recover スクリプトの使用に誤りがある。<br><br>処置: recover スクリプトに対する正しいパラメーターを指定してください。: recover -d <tag> |
| 112132 | FATAL | Usage: \”$cmd $usage\”. Specify the correct usage for the requested command. | 原因: SAP remoteControl スクリプトの使用に誤りがある。<br><br>処置: remoteControl スクリプトに対する正しいパラメーターを指定してください。: remoteControl <tag> <remote instance> <remote cmd> <remote cmd option> <primary system> <primary tag> |
| 112133 | FATAL | Usage: “%s %s”. Specify the correct usage for the requested command. | 原因: SAP remove スクリプトの使用に誤りがある。<br><br>処置: remove スクリプトに対する正しいパラメーターを指定してください。: remove -t <tag> -i <id> |
| 112134 | FATAL | Usage: \”$cmd $usage\”. Specify the correct usage for the requested command. | 原因: SAP restore スクリプトの使用に誤りがある。<br><br>処置: restore スクリプトに対する正しいパラメーターを指定してください。: restore -t <tag> -i <id> |
| 112137 | ERROR | The required parameter \"parent\" was either not provided or was invalid in the $func routine on $me. | 原因: CleanUpルーチンの使用に誤りがある。<br><br>処置: こちらは内部エラーです。SIOS カスタマーセンターへお問い合わせください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 112138 | ERROR | At least one required process for instance %s was not started successfully during "%s" on server %s. Please check the LifeKeeper and system logs for additional information. | **原因**: 指定したサーバーで、指定した SAP インスタンスに対する少なくとも1つのプロセスが正常に開始されませんでした。<br><br>**処置**: LifeKeeper/システムログ, SAP トレースファイルなどから原因を調査し、操作を再度行ってください。 |
| 112140 | FATAL | The tag parameter was not specified for the internal \"$func\" routine on $me. If this was a command line operation, specify the correct parameters. Otherwise, consult the troubleshooting documentation. | **原因**: 指定したサーバーでタグパラメーターが GetLK ルーチンに指定されていません。<br><br>**処置**: こちらは内部エラーです。SIOS カスタマーセンターへお問い合わせください。 |
| 112141 | ERROR | Either the SID ("%s") or instance ("%s") parameter was not specified for the "%s" routine on %s. | **原因**: SID もしくは SID のどちらかが指定したサーバーの StatusSapServer ルーチン に指定されていません。<br><br>**処置**: こちらは内部エラーです。SIOS カスタマーセンターへお問い合わせください。 |
| 112142 | ERROR | Either the SID ("%s"), instance ("%s"), or instance number ("%s") parameter was not specified for the "%s" routine on %s. | **原因**: SID、インスタンス、インスタンス番号のいずれかが指定のサーバーの StartSapServer ルーチンに提供されていない。<br><br>**処置**: こちらは内部エラーです。SIOS カスタマーセンターへお問い合わせください。 |
| 112143 | ERROR | The SID, instance, or instance number parameter was not specified for the "%s" routine on %s. | **原因**: SID、インスタンス、インスタンス番号のいずれかが指定のサーバーの StopSapServer ルーチンで指定されていない。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: こちらは内部エラーです。SIOS カスタマーセンターへお問い合わせください。 |
| 112173 | ERROR | The file “%s” does not exist or was not readable on “%s”. Verify that the specified file exists and/or is readable. | **原因**: 指定したファイルが指定したサーバーに存在しないか、読み込み不可になっている。<br><br>**処置**: ファイルが存在しているかを確認および/もしくは権限を修正して読み込み可能な状態にしてください。 |
| 112174 | ERROR | The file “%s” does not exist or was not readable on “%s”. Verify that the specified file exists and/or is readable. | **原因**: 指定したファイルが指定したサーバーに存在しないか、読み込み不可になっています。<br><br>**処置**: ファイルが存在しているかを確認および/もしくは権限を修正して読み込み可能な状態にしてください。 |
| 112175 | ERROR | The file “%s” does not exist or was not readable on “%s”. Verify that the specified file exists and/or is readable. | **原因**: 指定したファイルが指定したサーバーに存在しないか、読み込み不可になっている。<br><br>**処置**: ファイルが存在しているかを確認および/もしくは権限を修正して読み込み可能な状態にしてください。 |
| 112194 | ERROR | There was an error verifying the NFS connections for SAP related mount points on “%s”. One or more NFS servers is not operational and needs to be restarted. | **原因**: マウントポイントが /etc/default/LifeKeeper の SAP_NFS_CHECK_DIRS エントリに記述されている少なくとも一つの 重要なNFS 共有システムが現在無効となっている。<br><br>**処置**: すべての必要な NFS 共有ファイルシステムがアクセス可能であることを確認し、現在操作不可となっている NFS サーバーを再起動してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 112195 | FATAL | There was an error verifying the NFS connections for SAP related mount points on $me. One or more NFS servers is not operational and needs to be restarted. | **原因**: マウントポイントが /etc/default/LifeKeeper の SAP_NFS_CHECK_DIRS エントリに記述されている少なくとも一つの 重要なNFS 共有システムが現在無効となっている。<br><br>**処置**: すべての必要な NFS 共有ファイルシステムがアクセス可能であることを確認し、現在操作不可となっている NFS サーバーを再起動してください。 |
| 112196 | WARN | There was an error verifying the NFS connections for SAP related mount points on “%s”. One or more NFS servers is not operational and needs to be restarted. | **原因**: マウントポイントが /etc/default/LifeKeeper の SAP_NFS_CHECK_DIRS エントリに記述されている少なくとも一つの 重要なNFS 共有システムが現在無効となっている。<br><br>**処置**: すべての必要な NFS 共有ファイルシステムがアクセス可能であることを確認し、現在操作不可となっている NFS サーバーを再起動してください。 |
| 112201 | ERROR | The internal object value "%s" was empty. Unable to complete "%s" on %s. | **原因**: リソースタグの引数が GetLKEquivルーチンに提供されていません。<br><br>**処置**: こちらは内部エラーです。SIOS カスタマーセンターへお問い合わせください。 |
| 112203 | ERROR | The internal object value "%s" was empty. Unable to complete "%s" on %s. Additional information available in the LifeKeeper and system logs. | **原因**: SAP インスタンス番号が IsInstanceRunning ルーチンに提供されていません。<br><br>**処置**: こちらは内部エラーです。SIOS カスタマーセンターへお問い合わせください。 |
| 112204 | ERROR | The internal object value "%s" was empty. Unable to complete "%s" on %s. | **原因**: saphostexec もしくは saposcol コ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | マンドのどちらを使用するか判断できない。<br><br>**処置**: SAP kernel 7.3より前のSAP NetWeave に対して SAP_SRVHOST_CMD, SAP_HOSTCTL_CMD, SAP_OSCOL_CMD を使用している場合は、チューニングが正しく設定されていることを確認してください。 |
| 112205 | ERROR | The internal object value "%s" was empty. Unable to complete "%s" on %s. | **原因**: SAP インスタンスが 指定したサーバーのSAPRemExec ルーチンに提供されなかった。<br><br>**処置**: こちらは内部エラーです。SIOS カスタマーセンターへお問い合わせください。 |
| 112206 | ERROR | The required action parameter was not provided. Unable to complete "%s" on %s. | **原因**: 指定したサーバーで SAPRemExec ルーチンに対するアクションが提供されませんでした。<br><br>**処置**: こちらは内部エラーです。SIOS カスタマーセンターへお問い合わせください。 |
| 112208 | ERROR | Error getting the value of "%s" or "%s" from the default profile "%s" on %s. Verify that the specified value exists. | **原因**: 指定したサーバーで提供されたプロファイルから仮想IPもしくはホスト名を取得できない。<br><br>**処置**: プロファイルに適切なエントリーが存在していることを確認してください。 |
| 112209 | FATAL | Unable to gather required information from the SAP default profile for SID $sid ($DPFL) on $me. Verify that the default profile exists and is accessible. | **原因**: 指定したサーバーで提供されたプロファイルから SAP インスタンスに関する情報を取得できない。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: SAP ソフトウェアが正常にインストールされ提供されたプロファイルが存在および読み込み可能となっていることを確認してください。 |
| 112214 | ERROR | Unable to determine the status of the path "%s" ("%s") on %s. The path on %s may require the execution of the command: "mount <ip>:<export> %s". Verify that the SAP software is correctly installed and that all SAP file systems are mounted and accessible. | **原因**: 指定したパスのファイルシステムのステータスを判断できません。<br><br>**処置**: SAP ソフトウェアが正常にインストールされすべての必要となるファイルシステムがマウントされていることを確認してください。 |
| 112219 | ERROR | [HACONNECTOR:%s] Unable to write to file "%s" on %s. If the file already exists, manually enable write permissions on it. | **原因**: 指定したファイルが指定したサーバーで読み込み/書き込み権限をもっていない。<br><br>**処置**: 指定したサーバーの読み込み/書き込み権限を有効にしてください。 |
| 112220 | ERROR | Unable to start the sapstartsrv service for SID %s and SAP instance %s on %s. Verify that the sapservices file is correct and the process can be started manually. | **原因**: 指定した SAP インスタンスに対する SAP Start Service (sapstartsrv) プロセスを開始できない。<br><br>**処置**: sapservices ファイルが sapstartsrv プロセスを開始するための適切なコマンドを含んでいて、そのプロセスを手動で開始できるかを確認してください。 |
| 112221 | ERROR | The internal "%s" routine was called for a resource with unsupported instance type %s. This method supports only SAP instance types TYPE_CS, TYPE_ERS, or TYPE_NEW_ERS (1, 2, or 5). | **原因**: サポートされていないインスタンスタイプの内部SAPオブジェクトに対して、指定されたルーチンが呼び出された。<br><br>**処置**: こちらは内部エラーです。SIOS カスタマーセンターへお問い合わせください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 112222 | ERROR | The internal "%s" routine was called for a resource with unsupported instance type %s. This method supports only SAP instance types TYPE_CS, TYPE_ERS, or TYPE_NEW_ERS (1, 2, or 5). | **原因**: サポートされていないインスタンスタイプの内部SAPオブジェクトに対して、指定されたルーチンが呼び出された。<br><br>**処置**: こちらは内部エラーです。SIOSカスタマーセンターへお問い合わせください。 |
| 112223 | ERROR | The internal "%s" routine was called for a resource with unsupported instance type %s. This method supports only SAP instance type TYPE_CS (1). | **原因**: サポートされていないインスタンスタイプの内部SAPオブジェクトに対して、指定されたルーチンが呼び出された。<br><br>**処置**: こちらは内部エラーです。SIOSカスタマーセンターへお問い合わせください。 |
| 112224 | ERROR | The internal \"%s\" routine was called for a resource with unsupported instance type %s. This method supports only SAP instance type TYPE_CS (1). | **原因**: サポートされていないインスタンスタイプの内部SAPオブジェクトに対して、指定されたルーチンが呼び出された。<br><br>**処置**: こちらは内部エラーです。SIOSカスタマーセンターへお問い合わせください。 |
| 112225 | ERROR | The internal \"%s\" routine was called for a resource with unsupported instance type %s. This method supports only SAP instance type TYPE_NEW_ERS (5). | **原因**: サポートされていないインスタンスタイプの内部SAPオブジェクトに対して、指定されたルーチンが呼び出された。<br><br>**処置**: こちらは内部エラーです。SIOSカスタマーセンターへお問い合わせください。 |
| 112226 | ERROR | The internal "%s" routine was called for a resource with unsupported instance type %s. This method supports only SAP instance types TYPE_CS, TYPE_ERS, and TYPE_NEW_ERS (1, 2, and 5). | **原因**: サポートされていないインスタンスタイプの内部SAPオブジェクトに対して、指定されたルーチンが呼び出され |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | た。<br><br>**処置**: こちらは内部エラーです。SIOS カスタマーセンターへお問い合わせください。 |
| 112227 | ERROR | The internal "%s" routine was called for a resource with unsupported instance type %s. This routine only supports SAP instance type TYPE_CS (1). | **原因**: サポートされていないインスタンスタイプの内部SAPオブジェクトに対して、指定されたルーチンが呼び出された。<br><br>**処置**: こちらは内部エラーです。SIOS カスタマーセンターへお問い合わせください。 |
| 112228 | ERROR | The internal "%s" routine was called for a resource with unsupported instance type %s. This method supports only SAP instance type TYPE_NEW_ERS (5). | **原因**: サポートされていないインスタンスタイプの内部SAPオブジェクトに対して、指定されたルーチンが呼び出された。<br><br>**処置**: こちらは内部エラーです。SIOS カスタマーセンターへお問い合わせください。 |
| 112229 | ERROR | The profile \"%s\" either does not exist or cannot be read on %s. Unable to determine whether enqueue replication is enabled for resource %s. Please verify that the file exists and can be read. | **原因**: 指定したサーバー上に指定したインスタンスプロファイルが存在していない、もしくは読み取りができない。<br><br>**処置**: ファイルが存在していて、読み取り可能かどうかを確認してください。 |
| 112325 | ERROR | The $SAP_CONTROL utility cannot be located. Verify that all necessary file systems are mounted and that the $SAP_CONTROL utility can be located in the $sapadmin user's PATH. | **原因**: SAP インスタンスの管理に必要な sapcontrol ユーティリティが配置できない。<br><br>**処置**: 全ての必要なファイルシステムがマウントされ、sapcontrol ユーティリティがSAP 管理ユーザーの Path に配置されていることを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 112326 | ERROR | The \"which $SAP_CONTROL\" command for user $sapadmin returned $sapcmd as the location of the $SAP_CONTROL utility, but the utility could not be found or was not executable in this location. Verify that all necessary file systems are mounted and that the $SAP_CONTROL utility can be located in the $sapadmin user's PATH. | 原因: SAP インスタンスの管理に必要な sapcontrol ユーティリティが配置できない。<br><br>処置: 全ての必要なファイルシステムがマウントされ、sapcontrol ユーティリティがSAP 管理ユーザーのパスに配置されていることを確認してください。 |
| 112433 | ERROR | Unsupported SAPENQ_VERSION ($enqversion) for resource $tag on $me. Unable to obtain enqueue replication status. | 原因: 指定したサーバーで指定したリソースのSAPENQ_VERSIONパラメーターにサポートされていない値が検知された。<br><br>処置: 指定したリソースの info ファイルで、SAPENQ_VERSION に有効な値 (現在使用しているエンキューサーバーのバージョンを表す1 または 2) が設定されていることを確認してください。 |
| 112437 | ERROR | Profile \"$srvpf\" not found on $me. Unable to obtain enqueue replication status for instance $inst. | 原因: 指定したサーバー上に指定したインスタンスプロファイルが存在していない、もしくは読み取りができない。<br><br>処置: ファイルが存在していて、読み取り可能かどうかを確認してください。 |
| 112438 | ERROR | Unsupported SAPENQ_VERSION ($enqversion) for resource $tag on $me. Unable to obtain enqueue replication status for instance $inst. | 原因: 指定したサーバーで指定したリソースのSAPENQ_VERSION パラメーターにサポートされていない値が検知された。<br><br>処置: 指定したリソースの info ファイルで、SAPENQ_VERSION に有効な値 (現在使用しているエンキューサーバーのバージョンを表す1 または 2) が設定されていることを確認してください。 |
| 112470 | ERROR | The instance profile %s could not be found on %s. Verify that the SAP software is installed correctly and that all | 原因: 指定したサーバー上に指定したイ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | necessary file systems are mounted and accessible. | ンスタンスプロファイルが存在していない、もしくは読み取りができない。<br><br>**処置**: ファイルが存在していて、読み取り可能かどうかを確認してください。 |
| 112490 | ERROR | [HACONNECTOR] Unable to determine the corresponding tag for resource with ID \"$res\" on $me. | **原因**: 指定したサーバー上に指定したリソースID をもつ LifeKeeper SAP リソースを見つけることができない。<br><br>**処置**: fra コマンドの —res 引数として提供されるソース ID が有効な LifeKeeper SAP リソースに対応しているか確認してください。 |
| 112507 | ERROR | [HACONNECTOR] At least one required process for the instance was not killed successfully during the fra migrate action on $me. Aborting resource migration. | **原因**: fra マイグレーション実行中に SAP インスタンスが指定したサーバーで正常に停止しなかった。<br><br>**処置**: SAP インスタンスに対して起動中のプロセスを手動で停止し、マイグレーションを再度実行してください。 |
| 112539 | ERROR | [HACONNECTOR] Unable to find rpm information for one or more packages on $me. | **原因**: HA Connector gvi (“Get Version Information”)ルーチンは、LifeKeeper および/または the SAP Recovery Kit のバージョン番号を判別することができなかった。<br><br>**処置**: rpm -q コマンドでLifeKeeper Core および SAP Recovery Kit の rpm 情報を取得できるかを確認してください。 |
| 112976 | ERROR | There is no LifeKeeper protected resource with tag $tag on system $me. | **原因**: SAP canfailover スクリプトに渡されたリソースタグに対応する LifeKeeper リソースがない。<br><br>**処置**: リソースタグ名が正しいことを確 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 認し再度コマンドを実行してください。 |
| 112977 | ERROR | Resource $tag is not a $app/$typ resource. Please use the $ins_app/$ins_typ resource-specific canfailover script instead. | 原因: SAP canfailover スクリプトに渡された LifeKeeper リソースが、SAP（appsuite/sap）リソースではない。<br><br>処置: 指定したリソースに対して適切なタイプ固有の canfailover スクリプトを使用してください。 |

# 8.1.18. SAP HANA Recovery Kit メッセージカタログ

**Control F** を使用して、各カタログの特定のエラーコードを検索します。エラーコードの検索には、画面右上の検索ボックスをご利用ください。

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 136002 | ERROR | Usage: $usage | 原因: SAP HANA createスクリプトに無効な引数が指定された。<br><br>処置: 次の形式で適切な引数を指定してください：<br><リソースタグ> <SAP SID> <HDBインスタンス> [スイッチバックタイプ] [仮想IPリソースタグ] |
| 136003 | ERROR | END failed create of resource $tag on server $me with return value of $errcode. | 原因: SAP HANAリソースの作成中に障害が発生した。<br><br>処置: SAP HANAシステムレプリケーションがプライマリとセカンダリの両方のレプリケーションサイトで完全に構成および有効化されていることを確認し、リソース作成操作を再試行してください。 |
| 136005 | ERROR | An unknown error has occurred in utility rlslocks on server $me. View the LifeKeeper logs for details and retry the operation. | 原因: SAP HANAリソースの作成中にLifeKeeper rlslocksユーティリティで問題が発生する。<br><br>処置: LifeKeeperログファイルで見つかった問題を解決し、リソース作成操作を再試行してください。 |
| 136006 | ERROR | END failed create of resource $tag on server $me with signal $sig. | 原因: SAP HANAリソース作成中のシグナルによる割込み。<br><br>処置: 詳細については、LifeKeeperログファイルを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 136008 | ERROR | An unknown error has occurred in utility getlocks on server $me. View the LifeKeeper logs for details and retry the operation. | 原因: SAP HANAリソースの作成中にLifeKeeper getlocksユーティリティで問題が発生する。<br><br>処置: LifeKeeperログファイルで見つかった問題を解決し、リソース作成操作を再試行してください。 |
| 136009 | ERROR | The SAP HANA product was not found in the directory $obj->{'hana_util_path'} on server $me. Verify that SAP HANA is correctly installed and that all necessary file systems are mounted. | 原因: リソースの作成中に、必須のSAP HANAバイナリが見つからなかった。<br><br>処置: クラスター内のすべてのサーバーでSAP HANAが正しくインストールされ、構成されていることを確認してください。 |
| 136010 | ERROR | Failed to create resource as id $id already exists on system $me. | 原因: 指定されたIDのLifeKeeperリソースがすでにシステムに存在する。<br><br>処置: SAP HANAデータベースがLifeKeeperによってすでに保護されているかどうかを確認してください。 |
| 136011 | ERROR | Failed to create new tag $tag for SAP HANA resource on $me. | 原因: 入力されたSAP HANAリソースタグは別のLifeKeeperリソースによってすでに使用されており、LifeKeeper newtagユーティリティは新しいタグを作成できなかった。<br><br>処置: SAP HANAリソースに対し、別のタグ名を選択してください。 |
| 136012 | ERROR | Failed creation of resource with $tag on system $me. | 原因: LifeKeeperで指定したSAP HANAリソースを作成できなかった。<br><br>処置: LifeKeeperログファイルで見つかった問題を解決し、リソース作成操作を再試行してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 136014 | ERROR | Failed to create resource dependency for parent $tag and child $virtual_ip_tag. | 原因: SAP HANAリソースとそれが依存する仮想IPリソース間の依存関係の作成に失敗した。<br><br>処置: LifeKeeperログファイルで見つかった問題を解決し、リソース作成操作を再試行してください。 |
| 136015 | ERROR | The info field for resource $tag could not be successfully generated using values [SID: $info_sid, Instance: $info_instance, Replication Mode: $info_repl_mode, Site Name: $info_site_name, Operation Mode: $info_oper_mode]. Please verify that SAP HANA System Replication is fully configured and enabled on both the primary and secondary systems before creating the SAP HANA resource. | 原因: SAP HANAの「リソース情報」フィールドの作成時に無効な値が見つかった。<br><br>処置: SAP HANAが正しくインストールおよび構成されていること、SAP HANAシステムレプリケーションがクラスター内のすべてのサーバーで完全に構成され有効になっていることを確認してください。 |
| 136016 | ERROR | The selected server $me is not the primary/source system for SAP HANA System Replication for the selected SID $sid and HDB instance $instance. Please select ‘Cancel’ and start this action on the primary/source HANA System Replication system. | 原因: リソースの作成が、HANAシステムレプリケーションのセカンダリシステムで開始されている。<br><br>処置: HANAシステムレプリケーションのプライマリシステムで作成操作を開始してください。 |
| 136031 | ERROR | END failed extend of resource $target_tag on server $me with return value of $err_code. | 原因: SAP HANAリソースの拡張中に問題が発生した。<br><br>処置: LifeKeeperログファイルで見つかった問題を解決し、リソース拡張操作を再試行してください。 |
| 136033 | ERROR | Usage: $usage | 原因: SAP HANA extendスクリプトに無効な引数が指定された。<br><br>処置: 次の形式で適切な引数を指定してください： |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | <テンプレートシステム> <テンプレートタグ> <スイッチバックタイプ> <ターゲットタグ> |
| 136035 | ERROR | Template resource $template_tag on server $template_sys does not exist. | 原因: 拡張するテンプレートSAP HANAリソースがテンプレートサーバー上に存在しない。<br><br>処置: テンプレートサーバー上に、拡張されているSAP HANAリソースが存在することを確認してください。 |
| 136036 | ERROR | Resource with matching id $target_id already exists on server $me for App $app_type and Type $res_type. | 原因: 同じLifeKeeper IDのSAP HANAリソースがターゲットシステムにすでに存在する。<br><br>処置: SAP HANAデータベースが、ターゲットサーバー上のLifeKeeperによってすでに保護されているかどうかを確認してください。 |
| 136037 | ERROR | Resource with matching tag $target_tag already exists on server $me for App $app_type and Type $res_type | 原因: 同じLifeKeeperリソースタグを持つSAPリソースがターゲットサーバーに既に存在する。<br><br>処置: SAP HANAデータベースが、ターゲットサーバー上のLifeKeeperによってすでに保護されているかどうかを確認してください。 |
| 136039 | ERROR | Error creating resource $target_tag on system $me. | 原因: ターゲットサーバーで同等のSAP HANAリソースを作成できなかった。<br><br>処置: LifeKeeperログファイルで見つかった問題を解決し、リソース拡張操作を再試行してください。 |
| 136040 | ERROR | The target tag ($target_tag) and template tag ($template_tag) must be the same. | 原因: プライマリシステムで使用される |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | リソースタグ名が、セカンダリシステムで使用されるリソースタグ名と異なる。両方の名前は同じでなければならない。<br><br>**処置**: リソースの作成では、プライマリシステムタグおよびセカンダリシステムタグと同じ名前を使用してください。 |
| 136045 | ERROR | Cannot extend resource $template_tag to server $me. | **原因**: SAP HANA canextendスクリプトは、リソースを特定のターゲットシステムに拡張できないことを示している。<br><br>**処置**: LifeKeeperログファイルで見つかった問題を解決し、リソース拡張操作を再試行してください。 |
| 136047 | ERROR | Usage: $usage | **原因**: SAP HANA canextendスクリプトに無効な引数が指定された。<br><br>**処置**: 次の形式で適切な引数を指定してください：<br><テンプレートシステム> <テンプレートタグ> |
| 136048 | ERROR | Resource $template_tag does not exist on server $template_sys. | **原因**: 拡張するテンプレートSAP HANAリソースがテンプレートサーバー上に存在しない。<br><br>**処置**: テンプレートサーバー上に、拡張しているSAP HANAリソースが存在することを確認してください。 |
| 136049 | ERROR | The system user $hana_user does not exist on server $me. | **原因**: SAP HANAデータベースのSAP管理ユーザーが、指定されたサーバーに存在しない。<br><br>**処置**: SAP HANAが正しくインストールおよび構成されていることを確認してください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 136050 | ERROR | The user id for user $hana_user ($template_uid) on template server $template_sys is not the same as user id ($uid) on target server $me. | 原因: SAP管理ユーザーのユーザーIDが、テンプレートサーバーとターゲットサーバーで異なっている。<br><br>処置: SAPのHANAがクラスター内のすべてのサーバーに適切にインストールおよび構成されていることを確認してください。 |
| 136051 | ERROR | The group id for user $hana_user ($template_gid) on template server $template_sys is not the same as group id ($gid) on target server $me. | 原因: SAP管理ユーザーのグループIDが、テンプレートサーバーとターゲットサーバーで異なっている。<br><br>処置: SAPのHANAがクラスター内のすべてのサーバーに適切にインストールおよび構成されていることを確認してください。 |
| 136052 | ERROR | The home directory for user $hana_user ($template_home) on template server $template_sys is not the same as home directory ($user_home) on target server $me. | 原因: SAP管理ユーザーのホームディレクトリが、テンプレートサーバーとターゲットサーバーで異なっている。<br><br>処置: SAPのHANAがクラスター内のすべてのサーバーに適切にインストールおよび構成されていることを確認してください。 |
| 136053 | ERROR | The SAP HANA instance $instance does not exist for $sid on server $me. | 原因: 指定されたSAP HANAデータベースインスタンスのインストールディレクトリが見つからなかった。<br><br>処置: SAPのHANAがクラスター内のすべてのサーバーに適切にインストールおよび構成されていることを確認してください。 |
| 136055 | ERROR | The SAP HANA site name $target_obj->{‘site_name’} on server $me must be different from site name $template_obj->{‘site_name’} on $template_obj->{‘sys’}. | 原因: SAP HANAシステムレプリケーションのサイト名が、プライマリサーバーとセカンダリサーバーの両方で同じにな |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | っている。<br><br>**処置**: セカンダリサーバー上のSAP HANAデータベースを停止し、hdbnsutilユーティリティを使用して、別のサイト名でセカンダリレプリケーションサイトを再登録してください。 |
| 136056 | ERROR | Unable to obtain SAP HANA System Replication parameters for database $instance on server $me. Please verify that SAP HANA System Replication is enabled and properly configured and that the database instance is running on all servers in the cluster. | **原因**: 指定されたシステム上のデータベースのSAP HANAシステムレプリケーションパラメーターを決定できなかった。<br><br>**処置**: SAP HANAシステムレプリケーションが有効で適切に構成されていること、およびデータベースがクラスター内のすべてのサーバーで実行されていることを確認してください。 |
| 136057 | ERROR | Unable to create a HANA object for database $instance on server $me. Please verify that database instance $instance is properly installed. | **原因**: 指定したサーバー上の指定したインスタンスを表す内部HANAオブジェクトを作成できない。<br><br>**処置**: データベースインスタンスが正しくインストールされており、必要なすべてのファイルシステムがマウントされていることを確認してください。 |
| 136083 | ERROR | Usage: $usage | **原因**: SAP HANA deleteスクリプトの引数が無効。<br><br>**処置**: 有効なHANA LifeKeeperリソースタグ名とリソースIDの両方を入力してください。<br>入力方法：delete -t <tag> -i <id> [-U] |
| 136096 | ERROR | One of the required parameters (server, tag, or action) was missing. Unable to set the local recovery policy. | **原因**: 必要なパラメーター（サーバー、タグ、アクション）のいずれかが指定されていません。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **対応**: 必要なパラメーターを指定して、操作を再試行してください。 |
| 136097 | ERROR | Failed to $flg_action local recovery for SAP HANA resource $tag on server $node. | **原因**: 指定されたサーバー上の指定されたSAP HANAリソースのローカルリカバリーの有効化または無効化に失敗しました。<br><br>**対応**: LifeKeeperが実行中で、完全に初期化されていることを確認してから、操作を再試行してください。 |
| 136099 | ERROR | Failed start of SAP Host Agent on server $sys. | **原因**: 指定されたサーバーでSAP Host Agentプロセス（saphostexec, saposcol）を開始できませんでした。<br><br>**対応**: SAP Host Agentプロセスのトレースログを確認し、見つかった問題を修正してから、操作を再試行してください。 |
| 136100 | ERROR | Failed start of SAP Host Agent on server $sys. | **原因**: 指定されたサーバーでSAP Host Agentプロセス（saphostexec, saposcol）を開始できませんでした。<br><br>**対応**: SAP Host Agentプロセスのトレースログを確認し、見つかった問題を修正してから、操作を再試行してください。 |
| 136160 | ERROR | Unable to create SAP HANA object. The SID and instance for the SAP HANA database must be provided. | **原因**: SAP HANAオブジェクトの作成中に、SAP SIDまたはSAP HANAインスタンス名のいずれかが欠落していた。<br><br>**処置**: 詳細については、LifeKeeperログファイルを確認してください。 |
| 136161 | ERROR | Unable to create SAP HANA object. Resource system name and tag name must be provided. | **原因**: SAP HANAオブジェクトの作成中に、システム名またはリソースタグ名のいずれかが欠落していた。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 詳細については、LifeKeeperログファイルを確認してください。 |
| 136162 | ERROR | Could not find any information regarding resource $tag on $sys. | **原因**: 指定されたタグを持つSAP HANAリソースに関する情報を取得できなかった。<br><br>**処置**: 指定されたタグを持つSAP HANAリソースが指定されたサーバーに存在することを確認してください。 |
| 136168 | ERROR | Unable to check status of SAP Host Agent on server $self->{'sys'}. Command \"$curr_cmd\" returned exit code $ret. | **原因**: 指定されたサーバーでSAP Host Agentプロセスのステータスを判別できなかった。<br><br>**処置**: 詳細については、SAP Host Agentトレースファイル（dev_saphostexecなど）を確認してください。 |
| 136174 | ERROR | Unable to create SAP HANA object. The SID or instance value is missing. | **原因**:SAP HANAオブジェクトの作成時に、SAP SID、SAP HANAインスタンス名、またはSAP HANAシステムレプリケーション値のいずれかが欠落していた。<br><br>**処置**: 詳細については、LifeKeeperログファイルを確認してください。 |
| 136182 | ERROR | Failed to $flg_action flag \"${HANA_FLAG_DATA_OUT_OF_SYNC}_$eqv_tag{$sys}\" on server $sys. | **原因**: 指定されたサーバーで !HANA_DATA_OUT_OF_SYNC_<tag> フラグを作成または削除できなかった。<br><br>**処置**: 詳細については、LifeKeeperログファイルを確認してください。 |
| 136190 | ERROR | Failed to start SAP Host Agent processes on server $self->{'sys'}. Command \"$hostagent_cmd\" returned $ret. | **原因**: 指定されたサーバーでSAP Host Agentプロセスを開始できなかった。<br><br>**処置**: 詳細については、SAP Host Agent |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | トレースファイル（dev_saphostexecなど）を確認してください。 |
| 136191 | ERROR | Failed to start SAP OS Collector process on server $self->{'sys'}. Command \"$oscol_cmd\" returned $ret. | **原因**: 指定されたサーバーでSAP OS Collectorプロセスを開始できなかった。<br><br>**処置**: 詳細については、SAP OSコレクタートレースファイル（dev_collなど）を確認してください。 |
| 136193 | ERROR | $takeover_text of SAP HANA System Replication for SAP HANA database $self->{'instance'} failed on server $node with exit code $ret. | **原因**: SAP HANAシステムレプリケーションで、指定されたデータベースのプライマリマスターとして指定されたサーバーを登録できなかった。<br><br>**処置**: 詳細については、SAP HANAトレースファイル（例：nameserver_<hostname>.xxxxx.xxx.trc）を確認してください。 |
| 136197 | ERROR | Update of resource info field for $eqv_tag{$sys} on $sys failed with exit code $setinfo_ret. Current info: [$info]. Attempted new info: [$new_info]. | **原因**: 指定されたサーバー上の指定されたリソースの情報フィールドを更新できなかった。<br><br>**処置**: 詳細については、LifeKeeperログファイルを確認してください。 |
| 136202 | EMERG | Failed to disable Autostart for SAP HANA instance $self->{'instance'} on server $sys with exit code $remexec_ret. Please manually set \"Autostart = 0\" in the instance profile $profile on $sys. | **原因**: Autostartパラメーターの値を、指定されたサーバー上の指定されたHDBインスタンスプロファイルで変更できなかった。<br><br>**処置**: HDBインスタンスプロファイルを手動で編集し、「Autostart = 0」に設定してください。 |
| 136205 | ERROR | Failed start of SAP Start Service for SAP HANA database $instance on server $sys. | **原因**: 指定されたSAP HANAデータベースのSAP Start Serviceの開始に失敗し |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | た。<br><br>**処置**: 詳細については、SAP Start Service トレースファイル（sapstartsrv.logなど）を確認してください。 |
| 136208 | ERROR | LifeKeeper was unable to determine the SAP HANA System Replication mode for database $rem_obj->{‘instance’} on server $rem_obj->{‘sys’} while attempting to identify the previous primary replication site. Please resolve the issue and bring the SAP HANA resource in-service on the system where the database should be registered as primary master. | **原因**: 指定されたサーバーでSAP HANAシステムレプリケーションモードを判別できず、以前のプライマリレプリケーションサイトを特定できなかった。<br><br>**処置**: 詳細については、LifeKeeperログファイルを確認してください。 |
| 136210 | ERROR | Failed start of SAP Start Service for SAP HANA database $rem_obj->{‘instance’} on server $rem_obj->{‘sys’}. Unable to stop the database on the server where it is currently registered as primary master. | **原因**: 指定されたサーバー上の指定されたSAP HANAデータベースのSAP Start Serviceの開始に失敗し、その結果現在のプライマリSAP HANAシステムレプリケーションサイトでデータベースを停止できなかった。<br><br>**処置**: 詳細については、SAP Start Service トレースファイル（sapstartsrv.logなど）を確認してください。 |
| 136212 | ERROR | Failed stop of SAP HANA database $rem_obj->{‘instance’} on server $rem_obj->{‘sys’} where it is currently registered as primary master. | **原因**: 指定されたサーバー上の指定されたSAP HANAデータベースの停止に失敗した。<br><br>**処置**: 詳細については、SAP HANAトレースファイルとLifeKeeperログファイルを確認してください。 |
| 136217 | ERROR | Failed start of SAP HANA database $instance on server $sys. | **原因**: 指定されたサーバー上の指定されたSAP HANAデータベースの起動に失敗した。<br><br>**処置**: 詳細については、SAP HANAトレースファイルとLifeKeeperログファイル |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | を確認してください。 |
| 136220 | ERROR | Unable to register $sys as a secondary SAP HANA System Replication site for database $instance. The host name of the current primary replication site was not provided. | **原因**: セカンダリレプリケーションサイトの登録時に、現在のプライマリSAP HANAシステムレプリケーションサイトのホスト名を入力しなかった。<br><br>**処置**: 詳細については、LifeKeeperログファイルを確認してください。 |
| 136233 | EMERG | WARNING: A temporary communication failure has occurred between servers $self->{'sys'} and $sys. Manual intervention is required in order to minimize the risk of data loss. To resolve this situation, please take one of the following resource hierarchies out of service: $self->{'tag'} on $self->{'sys'} or $eqv{$sys} on $sys. The server that the resource hierarchy is taken out of service on will become the standby server for SAP HANA database $self->{'instance'}. | **原因**: 一時的な通信障害により、指定された同等のHANAリソースがそれぞれのホストサーバーで同時にIn Serviceになった。<br><br>**処置**: セカンダリレプリケーションサイトになるサーバーで、HANAリソース階層全体をOut of Serviceにします。そのサーバーでデータベースが停止されると、LifeKeeperは次のquickCheckサイクル中にデータベースをセカンダリレプリケーションサイトとして自動的に登録します。 |
| 136234 | EMERG | WARNING: SAP HANA database $self->{'instance'} is running and registered as primary master on the following servers: $primary_node_list. Manual intervention is required in order to minimize the risk of data loss. To resolve this situation, please stop database $self->{'instance'} on the intended secondary replication site by running the command \'su – $self->{'sid_admin'} -c \"$SAP_CONTROL -nr $self->{'instance_number'} -function StopWait $HANA_STOP_WAIT 5\"\' on that server, then bring resource $self->{'tag'} in-service on the intended primary replication site. Once the resource is in-service, the stopped database will automatically be re-registered as a secondary replication site. | **原因**: 指定されたSAP HANAデータベースが、2つのクラスターサーバーで同時にプライマリマスターとして実行および登録されている。<br><br>**処置**: メッセージに示されているコマンドを使用して、スタンバイサーバー上のデータベースを停止します。データベースが停止すると、LifeKeeperは自動的にスタンバイサーバーをセカンダリレプリケーションサイトとして登録します。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 136236 | ERROR | Failed to remove ${HANA_FLAG_DATA_OUT_OF_SYNC}_$tag flag on server $sys. | **原因**: 指定されたサーバーで !HANA_DATA_OUT_OF_SYNC_ LifeKeeperフラグを削除できませんでした。指定されたSAP HANAリソースは、これが削除されるまで指定されたサーバーでIn-Serviceになりません。<br><br>**対応**: LifeKeeperが実行中で、完全に初期化されていることを確認します。SAP HANAシステムレプリケーションが同期していることを確認できた場合は、コマンド "/opt/LifeKeeper/bin/flg_remove -f '!HANA_DATA_OUT_OF_SYNC_<HANA Tag>'" を使用してout-of-syncフラグを手動で削除できます |
| 136238 | ERROR | Unable to create SAP HANA object. Resource system name and tag name must be provided. | **原因**: SAP HANAオブジェクトの作成時に、サーバー名またはリソースタグ名のいずれかが欠落していた。<br><br>**処置**: 詳細については、LifeKeeperログファイルを確認してください。 |
| 136239 | ERROR | Failed start of SAP Start Service for SAP HANA database $obj->{'instance'} on server $obj->{'sys'}. Unable to determine status of the database on $obj->{'sys'}. | **原因**: 指定されたサーバー上の指定されたSAP HANAデータベースのSAP Start Serviceの開始に失敗し、その結果データベースのステータスを判別できなかった。<br><br>**処置**: 詳細については、SAP Start Serviceトレースファイル（sapstartsrv.logなど）を確認してください。 |
| 136242 | ERROR | Unable to create SAP HANA object. At least one of the SAP HANA System Replication values is missing. | **原因**: 指定したシステム上のデータベースに対し、SAP HANAシステムレプリケーションパラメーターを特定できない。<br><br>**処置**: SAP HANAシステムレプリケーションが有効になっていて適切に構成されていること、データベースがクラスター |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 内のすべてのサーバーで実行されていることを確認してください。 |
| 136243 | ERROR | Unable to locate the pingnfs utility ($pingnfs) on server $me. Please verify that this utility exists and is executable. | **原因**: エクスポートされたNFS共有で利用可能なテストに使用されるpingnfsユーティリティが見つからない。<br><br>**処置**: pingnfsユーティリティが所定の場所に存在し、実行可能であることを確認してください。 |
| 136245 | ERROR | Critical NFS shares being exported by server $sys ($export_list) are not currently available. Please verify that the NFS server is alive and that all NFS-related services are running. | **原因**: 指定した重要なNFS共有ファイルシステムが現在利用できない。<br><br>**処置**: ファイルシステムをエクスポートするNFSサーバーが稼働していること、および必要なNFS関連サービスがすべて実行されていることを確認してください。 |
| 136248 | ERROR | Unable to open file $crit_mount_file on server $me. Please verify that this file exists and is read-enabled. | **原因**: 指定したサーバー上の重要なNFS共有のマウント情報を含むファイルを開けない。<br><br>**処置**: ファイルが指定した場所に存在し、読み取りが有効になっていることを確認してください。 |
| 136253 | ERROR | The SAP HANA \"takeover with handshake\" feature is available only for SAP HANA versions 2.0 SPS04 and greater. Database $self->{'instance'} cannot be resumed. | **原因**: 一時停止されたデータベースの再開は、2.0 SPS04より前のSAP HANAバージョンではサポートされていない。<br><br>**処置**:「Takeover with Handshake」に関連する機能を使用するには、SAP HANA 2.0 SPS04以降にアップグレードしてください。 |
| 136254 | ERROR | Attempt to resume suspended primary database $self->{'instance'} on server $self- | **原因**: 指定したサーバーで中断されたデ |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | >{'sys'} failed with exit code $ret. | ータベースインスタンスを再開しようとしたら失敗した。<br><br>**処置**:LifeKeeperとSAPのログファイルを調べて、失敗の原因を特定します。操作を再試行するか、対応するLifeKeeperリソースを別のサーバーでIn-Serviceにしてください。 |
| 136258 | ERROR | Attempt to register server $node as the secondary SAP HANA System Replication site for database $self->{'instance'} failed with exit code $ret. | **原因**: 指定したサーバーを、指定したデータベースインスタンスのセカンダリSAP HANAシステムレプリケーションサイトとして登録しようとしたが失敗した。<br><br>**処置**:LifeKeeperとSAPのログファイルを調べて障害の原因を特定し、見つかった問題を手動で修正してください。SAP HANAリソースがIn-Serviceになっている間、LifeKeeperは自動的に、セカンダリHSRロールへのバックアップサーバーの登録試行を継続します。 |
| 136263 | ERROR | Usage: $usage | **原因**: SAP HANAリソースのリストアスクリプトの引数が無効。<br><br>**処置**:次の形式で適切な引数を指定してください：<br>-t <リソースタグ> -i <リソースID> |
| 136265 | ERROR | Error getting resource information for $tag on server $me. | **原因**: 指定されたサーバーの指定されたSAP HANAリソースに関する情報を取得できなかった。<br><br>**処置**:指定されたタグを持つSAP HANAリソースが指定されたサーバーに存在することを確認してください。 |
| 136266 | ERROR | The resource $tag protecting SAP HANA database $instance is not in sync. To protect the data LifeKeeper will not restore the resource on $me. Please restore the re | **原因**: SAP HANAシステムレプリケーションは、バックアップサーバー上でデー |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | source on the previous source server to allow the resync to complete. | タベースリソースをIn Serviceにする前に同期されていなかった。<br><br>**処置**: 前のプライマリサーバーでSAP HANAリソースをIn Serviceにし、再同期を完了できるようにしてください。 |
| 136275 | ERROR | Failed to determine SAP HANA System Replication mode for database $instance on server $me. | **原因**: 指定されたサーバー上の指定されたデータベースのSAP HANAシステムレプリケーションモードを判定できなかった。<br><br>**処置**: 詳細については、SAP HANAトレースファイル（例：nameserver_<hostname>.xxxxx.xxx.trc）およびLifeKeeperログファイルを確認してください。 |
| 136351 | ERROR | Usage: $usage | **原因**: SAP HANA resource quickCheckスクリプトの引数が無効。<br><br>**処置**: 次の形式で適切な引数を指定してください：<br>-t <リソースタグ> -i <リソースID> |
| 136353 | ERROR | Error getting resource information for $tag. | **原因**: 指定されたSAP HANAリソースに関する情報を取得できなかった。<br><br>**処置**: 指定されたタグを持つSAP HANAリソースが指定されたサーバーに存在することを確認してください。 |
| 136354 | EMERG | The replication mode of the SAP HANA database $instance corresponding to resource $tag was modified outside of LifeKeeper and is no longer registered as primary master on server $me. Please bring the SAP HANA resource in-service on the server where the database should be registered as primary master. Resource monitoring for $tag will be suspended until the issue is resolved. | **原因**: 指定されたデータベースのSAP HANAシステムレプリケーションモードが、LifeKeeperの外部で変更されている。<br><br>**処置**: プライマリマスターとして登録する必要があるサーバーで、SAP HANAリソースをIn Serviceにしてください。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| 136363 | EMERG | LifeKeeper was unable to determine the SAP HANA System Replication mode for database $instance corresponding to resource $tag on server $me. Resource monitoring for $tag will be suspended until the issue is resolved. | 原因: 指定されたサーバーでSAP HANAシステムレプリケーションモードを判定できなかった。<br><br>処置: 詳細については、LifeKeeperログファイルを確認してください。 |
| 136375 | EMERG | An NFS server exporting a critical shared file system for resource $tag is currently unavailable. Resource monitoring for $tag will be suspended until the issue is resolved. | 原因: 重要なNFS共有ファイルシステムが現在利用不可である。<br><br>処置: NFSサーバーが稼働していること、および必要なNFS関連サービスがすべて実行されていることを確認してください。 |
| 136376 | EMERG | WARNING: LifeKeeper resource $tag is designed for use in situations where SAP HANA System Replication (HSR) is disabled, but HSR was found to be enabled on server $me. Please ensure that the correct LifeKeeper resource type is being used for your current SAP HANA configuration. | 原因: 指定したLifeKeeperリソースは、HANAシステムレプリケーション（HSR）が無効になっているSAP HANAデータベース環境を保護するように設計されているが、指定したサーバーではHSRが有効になっている。<br><br>処置: SAP HANAの構成に応じて、正しいLifeKeeperリソースタイプが使用されていることを確認してください。HSRが無効になっているSAP HANA構成から、HSRが有効になっているSAP HANA構成に移行した場合は、対応するSAP HANAリソースをLifeKeeperで再作成する必要があります。 |
| 136377 | EMERG | SAP HANA database $instance corresponding to resource $tag is currently suspended on server $me due to actions performed outside of LifeKeeper. In order to resume the suspended database, execute the command 'su – $obj->{'sid_admin'} -c \"$obj->{'hana_util'} -sr_resumeSuspendedPrimary\"' on server $me. Once the suspended database has been resumed, LifeKeeper will automatically register the other servers in the cluster as secondary replicatio | 原因: 指定したサーバー上のSAP HANAデータベースインスタンスが、LifeKeeperの外部で行われた操作により停止された。<br><br>処置: 対応するLifeKeeperリソースが現在In-Service（Active）になっているサーバー上で一時停止中のデータベースを再開する場合は、メッセージに記載され |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | n sites. Otherwise, please bring resource $tag in-service on the server where the database should be registered as primary master. Resource monitoring for $tag will be suspended until the issue is resolved. | ているコマンドを実行してください。それ以外の場合は、目的のプライマリーレプリケーションサイトでLifeKeeper SAP HANAリソースをIn-Serviceにしてください。 |
| 136450 | ERROR | Usage: $usage | 原因: SAP HANA removeスクリプトに無効な引数が指定されている。<br><br>処置: 次の形式で適切な引数を指定してください：<br><テンプレートタグ> <テンプレートID> |
| 136454 | ERROR | Error getting resource information for $tag. | 原因: 指定されたSAP HANAリソースに関する情報を取得できなかった。<br><br>処置: 指定されたタグを持つSAP HANAリソースが指定されたサーバーに存在することを確認してください。 |
| 136456 | ERROR | Failed start of SAP Start Service for SAP HANA database $instance on server $me. | 原因: 指定されたSAP HANAデータベースのSAP Start Serviceを開始できなかった。<br><br>処置: 詳細については、SAP Start Serviceトレースファイル（sapstartsrv.logなど）を確認してください。 |
| 136459 | EMERG | WARNING: LifeKeeper resource $tag is designed for use in situations where SAP HANA System Replication (HSR) is disabled, but HSR was found to be enabled on server $me. Please ensure that the correct LifeKeeper resource type is being used for your current SAP HANA configuration. | 原因: 指定したLifeKeeperリソースは、HANAシステムレプリケーション（HSR）が無効になっているSAP HANAデータベース環境を保護するように設計されているが、指定したサーバーではHSRが有効になっている。<br><br>処置: SAP HANAの構成に応じて、正しいLifeKeeperリソースタイプが使用されていることを確認してください。HSRが無効になっているSAP HANA構成か |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | ら、HSRが有効になっているSAP HANA構成に移行した場合は、対応するSAP HANAリソースをLifeKeeperで再作成する必要があります。 |
| 136462 | ERROR | Failed to remove flag \"${hana::HANA_FLAG_LEAVE_DB_RUNNING}_$tag\" on server $me. This may cause subsequent remove actions for resource $tag on server $me to unintentionally fail to stop the database. | 原因: 指定したサーバーでLifeKeeperの!volatile!hana_leave_db_running_<tag>フラグを削除できない。<br><br>処置: 指定したサーバーでコマンド「/opt/LifeKeeper/bin/flg_remove -f '!volatile!hana_leave_db_running_<tag>'」を実行して、フラグを手動で削除してください。フラグが存在する間は、指定したサーバー上のSAP HANAリソースのOut-of-service操作では、保護対象のデータベースインスタンスが実行されたままになります。 |
| 136550 | ERROR | Usage: $usage | 原因: HANAリソース回復スクリプトの引数が無効。<br><br>処置: 有効なHANA LifeKeeperリソースタグ名とリソースIDの両方を入力してください。<br>入力方法：recover -d <タグ> -n <id> |
| 136555 | ERROR | Error getting resource information for $tag on server $me. | 原因: 指定されたサーバー上の指定されたHANAリソースに関する情報を取得できなかった。<br><br>処置: サーバーがオンラインで、LifeKeeperが実行されており、HANAリソースが存在することを確認してください。 |
| 136556 | EMERG | The replication mode of the SAP HANA database $instance corresponding to resource $tag was modified outside of LifeKeeper and is no longer registered as primary master on server $me. Please bring the SAP HANA resource in-service on the serv | 原因: 指定されたデータベースのSAP HANAシステムレプリケーションモードが、LifeKeeperの外部で変更されている。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | er where the database should be registered as primary master. Resource monitoring for $tag will be suspended until the issue is resolved. | **処置**: プライマリマスターとして登録する必要があるサーバーで、SAP HANAリソースをIn Serviceにしてください。 |
| 136558 | EMERG | LifeKeeper was unable to determine the SAP HANA System Replication mode for database $instance corresponding to resource $tag on server $me. Resource monitoring for $tag will be suspended until the issue is resolved. | **原因**: 指定されたサーバー上の指定されたデータベースに対し、SAP HANAシステムレプリケーションモードを決定でなかった。<br><br>**処置**: 詳細については、SAP HANAトレースファイルとLifeKeeperログファイルを確認してください。 |
| 136559 | ERROR | Resource $tag is no longer ISP on server $me. Exiting $cmd for $tag. | **原因**: 指定されたSAP HANAリソースが、指定されたサーバー上でISPではなくなった。<br><br>**処置**: 詳細については、LifeKeeperログファイルを確認してください。 |
| 136650 | ERROR | Usage: $usage | **原因**: SAP HANA hana_stop_all_dbs スクリプトに無効な引数が指定されている。<br><br>**処置**: 次の形式で適切な引数を指定してください：<br>hana_stop_all_dbs -t <タグ> |
| 136654 | ERROR | Error getting resource information for $tag. | **原因**: 指定されたSAP HANAリソースに関する情報を取得できなかった。<br><br>**処置**: 指定されたタグを持つSAP HANAリソースが指定されたサーバーに存在することを確認してください。 |
| 136658 | ERROR | Failed start of SAP Start Service for SAP HANA database $x->{‘instance’} on server $x->{‘sys’}. Could not determine status of SAP HANA DB on $x->{‘sys’}. | **原因**: 指定されたSAP HANAデータベースのSAP Start Serviceの開始に失敗した。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: 詳細については、SAP Start Service トレースファイル（sapstartsrv.logなど）を確認してください。 |
| 136661 | ERROR | Failed stop of SAP HANA database $x->{'instance'} on server $x->{'sys'}. | **原因**: 指定されたサーバー上の指定されたSAP HANAデータベースの停止に失敗した。<br><br>**処置**: 詳細については、SAP HANAトレースファイルとLifeKeeperログファイルを確認してください。 |
| 136673 | ERROR | Usage: $usage | **原因**: SAP HANAのhana_takeover_with_handshakeスクリプトで無効な引数が指定された。<br><br>**処置**: 次の形式で適切な引数を指定してください： -t <tag> [-s <target server>] [-b] |
| 136674 | ERROR | Unable to remove the \"${hana::HANA_FLAG_LEAVE_DB_RUNNING}_$tag\" flag on server $isp_node. This may cause subsequent remove actions for resource $tag on server $isp_node to unexpectedly fail to stop the database. | **原因**: 指定したサーバーでLifeKeeperの!volatile!hana_leave_db_running_<tag>フラグを削除できない。<br><br>**処置**: 指定したサーバーでコマンド「/opt/LifeKeeper/bin/flg_remove -f '!volatile!hana_leave_db_running_<tag>'」を実行して、フラグを手動で削除してください。フラグが存在する間は、指定したサーバー上のSAP HANAリソースのOut-of-service操作では、保護対象のデータベースインスタンスが実行されたままになります。 |
| 136675 | ERROR | Script $cmd exited unexpectedly due to signal \"$sig\" on server $me. This may leave the $tag resource hierarchy as well as SAP HANA System Replication in an unexpected state. Please verify that the cluster resources are in the expected state. | **原因**: hana_takeover_with_handshakeスクリプトが、指定したサーバー上の指定したシグナルにより、予期せず終了した。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: SAP HANAクラスタリソースが期待通りの状態であることを確認します。そうでない場合は、見つかった問題を修正し、目的のプライマリーサーバーでSAP HANAリソース階層をIn-Serviceにします。 |
| 136677 | ERROR | Unable to find equivalent SAP HANA resource corresponding to $tag on server $target_node. | **原因**: 指定したサーバー上の指定したリソースに対応する同等のSAP HANAリソースが見つからない。<br><br>**処置**: 指定したリソースタグが正しく、リソースが指定したターゲットサーバーに拡張されており、LifeKeeper がターゲットサーバー上で実行されて完全に初期化されていることを確認してください。 |
| 136678 | ERROR | Unable to obtain information about equivalent SAP HANA resource $tag on server $target_node. Verify that LifeKeeper is running and fully initialized. | **原因**: 指定したサーバー上の指定したリソースに対応する同等のSAP HANAリソースが見つからない。<br><br>**処置**: 指定したリソースタグが正しく、リソースが指定したターゲットサーバーに拡張されており、LifeKeeper がターゲットサーバー上で実行されて完全に初期化されていることを確認してください。 |
| 136679 | ERROR | Resource $tag is not a SAP HANA resource. | **原因**: 指定したリソースがSAP HANA（データベース/hana）リソースではない。<br><br>**処置**: リソースタグが正しいことを確認してください。 |
| 136680 | ERROR | Resource $tag is designed for use in an environment where SAP HANA System Replication is disabled. Takeover with handshake cannot be performed for this resource type. Please use the standard \"In Servi | **原因**:「Takeover with Handshake」に関連する機能は、SAP HANAシステムレプリケーションが有効な環境でのみ使用で |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | ce…\" command instead. | きる。<br><br>**処置**: 標準の「In Service…」オプションを使用して、SAP HANAリソースをIn-Serviceにしてください。 |
| 136681 | ERROR | SAP HANA resource $tag is not currently in-service on any server in the cluster. The resource must be in-service and SAP HANA System Replication must be in-sync before performing a takeover with handshake. | **原因**:「Takeover with Handshake」の試行中、指定したSAP HANAリソースがクラスター内のどのサーバーでもIn-Serviceになっていない。<br><br>**処置**: 目的のプライマリーサーバーでSAP HANAリソースをIn-Serviceにしてください。 |
| 136682 | ERROR | SAP HANA resource $tag is currently in-service on multiple servers: $isp_node_list. The resource must be in-service on only one server and SAP HANA System Replication must be in-sync before performing a takeover with handshake. | **原因**:「Takeover with Handshake」の試行中、指定したSAP HANAリソースがクラスター内の複数のサーバーでIn-Serviceになっている。<br><br>**処置**: プライマリーマスターとして登録する予定のサーバーを除くすべてのサーバーで、リソースをOut-of-serviceにしてください。 |
| 136684 | ERROR | Unable to create internal SAP HANA object for resource $tag on server $target_node. Verify that all necessary file systems are mounted and that LifeKeeper is running and fully initialized on $target_node. | **原因**: 指定したサーバー上の指定したインスタンスを表す内部HANAオブジェクトを作成できない。<br><br>**処置**: データベースインスタンスが正しくインストールされ、必要なファイルシステムがすべてマウントされていることを確認してください。 |
| 136685 | ERROR | Takeover with handshake is only supported in SAP HANA versions 2.0 SPS04 and greater. The SAP HANA software must be upgraded in order to use this feature. Please use the standard \"In Service…\" command instead. | **原因**: 基礎となるSAP HANAデータベースのバージョンが2.0 SPS04未満の場合、「Takeover with Handshake」機能は使用できない。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**:「Takeover with Handshake」機能を使用するには、SAP HANA 2.0 SPS04以降にアップグレードしてください。 |
| 136686 | ERROR | Takeover with handshake cannot be performed for database $target_obj->{'instance'} on server $target_node because the database is not currently running and registered as primary on any other server in the cluster. | **原因**:「Takeover with Handshake」の試行中、指定したSAP HANAデータベースインスタンスがクラスター内のどのサーバーでも実行されておらず、プライマリーマスターとして登録されていない。<br><br>**処置**: 目的のプライマリーサーバーで対応するSAP HANAリソースをIn-Serviceにしてください。 |
| 136687 | ERROR | SAP HANA database $target_obj->{'instance'} is running and registered as primary on more than one server in the cluster. Please resolve this situation and reattempt the takeover. | **原因**:「Takeover with Handshake」の試行中、指定したSAP HANAデータベースインスタンスが複数のサーバーでプライマリーとして実行および登録されている。<br><br>**処置**: データベースインスタンスがすでに実行されており、対象のプライマリーサーバーにプライマリーとして登録されている場合は、対応するLifeKeeperリソースをそのサーバーでIn-Serviceにします。そうでない場合は、LifeKeeperで現在In-Serviceになっているサーバーを除くすべてのサーバーでデータベースインスタンスを停止し、LifeKeeperがシステムレプリケーションを再開できるようにしてから、テイクオーバーを再試行してください。 |
| 136689 | ERROR | Unable to create internal SAP HANA object for resource $tag on server $isp_node. Verify that all necessary file systems are mounted and that LifeKeeper is running and fully initialized on $isp_node. | **原因**: 指定したサーバー上の指定したインスタンスを表す内部HANAオブジェクトを作成できない。<br><br>**処置**: データベースインスタンスが正しくインストールされ、必要なファイルシステムがすべてマウントされていること |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | を確認してください。 |
| 136690 | ERROR | Unable to set the ${hana::HANA_FLAG_LEAVE_DB_RUNNING}_$tag flag on server $isp_node. Aborting takeover with handshake attempt for database $target_obj->{'instance'} on server $target_node. | 原因:「Takeover with Handshake」の試行中に、指定したサーバーで、LifeKeeperフラグ「!volatile!hana_leave_db_running_<tag>」を設定できなかった。<br><br>処置: 詳細については、LifeKeeper のログファイルを確認してください。見つかった問題を修正し、テイクオーバーを再試行してください。 |
| 136692 | ERROR | Takeover with handshake for database $target_obj->{'instance'} failed on server $target_node. | 原因: 指定したターゲットサーバー上の指定したSAP HANAデータベースインスタンスに対する「Takeover with Handshake」の試行が失敗した。<br><br>処置: /etc/default/LifeKeeper で HANA_HANDSHAKE_TAKEOVER_FAILBACK=true が指定されている場合、SAP HANAリソース階層は以前のデータベースホストサーバーで自動的にIn-Serviceに戻ります。そうでない場合は、SAP HANAリソース階層を対象のプライマリーサーバーで手動でIn-Serviceにする必要があります。 |
| 136695 | ERROR | Resource $tag does not exist on server $me. | 原因: 指定したリソースが指定したサーバーに存在しない。<br><br>処置: リソースタグが正しいことを確認してください。 |
| 136696 | ERROR | Unable to verify the status of resource $tag on server $target_node. Assuming that it is not in-service. | 原因:「Takeover with Handshake」の成否の確認中に、指定したサーバー上の指定したリソースのステータスを判別できなかった。<br><br>処置: 指定したサーバー上でLifeKeeper |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | が実行され、完全に初期化されていること、およびローカルサーバーとターゲットサーバー間のコミュニケーションパスがアクティブであることを確認してください。/etc/default/LifeKeeper で HANA_HANDSHAKE_TAKEOVER_FAILBACK=true が指定されている場合、SAP HANAリソース階層は自動的に以前のデータベースホストサーバーでIn-Serviceになります。そうでない場合は、SAP HANAリソース階層を対象のプライマリーサーバー上で手動でIn-Serviceにする必要があります。 |
| 136697 | ERROR | Resource $res was not successfully brought in-service on server $target_node. | 原因:「Takeover with Handshake」の試行中に、指定したリソースが指定したサーバーでIn-Serviceにならなかった。<br><br>処置: etc/default/LifeKeeper で HANA_HANDSHAKE_TAKEOVER_FAILBACK=true に設定されている場合、SAP HANAリソース階層は自動的に以前のデータベースホストサーバーでサービスをIn-Serviceにします。そうでない場合は、SAP HANAリソース階層を対象のプライマリーサーバーで手動でIn-Serviceにする必要があります。 |
| 136698 | ERROR | LifeKeeper is not running or is not fully initialized on server $me. | 原因:「Takeover with Handshake」の試行中に、LifeKeeperが指定したサーバーで実行されていないか、完全に初期化されていない。<br><br>処置: LifeKeeperを /opt/LifeKeeper/bin/lkstart で起動するか、完全に初期化してから、テイクオーバーを再試行してください。 |
| 136699 | ERROR | Unknown server $target_node. | 原因: 指定したサーバーのホスト名が認識されない。 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | **処置**: サーバーのホスト名が正しく、ローカルサーバーとターゲットサーバーの間でコミュニケーションパスが作成されていることを確認してください。 |
| 136700 | ERROR | Usage: $usage | **原因**: SAP HANA remoteregisterdbスクリプトに無効な引数が与えられました。<br><br>**対応**: 次の形式で適切な引数を指定してください: remoteregisterdb -d <tag> -n <id> |
| 136705 | ERROR | Unable to obtain information about resource $tag on server $me. Exiting $cmd for $tag. | **原因**: 指定されたサーバー上の指定されたSAP HANAリソースに関する情報を取得できませんでした。<br><br>**対応**: 指定されたリソースタグが正しいこと、リソースが指定されたサーバーに存在すること、必要なすべてのファイルシステムがマウントされていること、LifeKeeperが実行中で、指定されたサーバーで完全に初期化されていることを確認します。 |
| 136706 | ERROR | Resource $tag is no longer ISP on server $me. Exiting $cmd for $tag. | **原因**: SAP HANAリソースが特定のサーバーでIn-Service（ISP）になっていないため、SAP HANA remoteregisterdbスクリプトが終了しています。<br><br>**対応**: 対応は必要ありません。 |
| 136707 | ERROR | The $cmd event is intended for use only in environments in which SAP HANA System Replication is enabled. Exiting $cmd for $tag. | **原因**: SAP HANA remoteregisterdbイベントは、指定されたSAP HANAリソースによって保護されているデータベースに対してSAP HANAシステムレプリケーションが無効になっていることを検出しました。<br><br>**対応**: remoteregisterdbスクリプトが実 |

| コード | 重大度 | メッセージ | 原因/処置 |
|---|---|---|---|
| | | | 行されていたサーバーでSAP HANAシステムレプリケーションが有効になっていることを確認します。必要に応じて、SAP HANA管理ユーザーとしてコマンド 'hdbnsutil -sr_enable —name=<Site Name>' を実行することにより、システムレプリケーションを有効にすることができます。 |
| 136708 | ERROR | Error getting resource information for $tag on server $me. Exiting $cmd for $tag. | **原因**: 指定されたサーバーの指定されたSAP HANAリソースに関する情報を取得できませんでした。<br><br>**対応**: 指定されたタグを持つSAP HANAリソースが指定されたサーバーにあることを確認してください。 |
| 136709 | EMERG | LifeKeeper was unable to determine the SAP HANA System Replication mode for database $instance corresponding to resource $tag on server $me. Resource monitoring for $tag will be suspended until the issue is resolved. | **原因**: 指定されたサーバーの指定されたデータベースに対して、SAP HANAシステムレプリケーションモードを検出できませんでした。<br><br>**対応**:詳細については、SAP HANAトレースファイルとLifeKeeperログファイルを確認してください。 |
| 136710 | EMERG | The replication mode of the SAP HANA database $instance corresponding to resource $tag was modified outside of LifeKeeper and is no longer registered as primary master on server $me. Please bring the SAP HANA resource in-service on the server where the database should be registered as primary master. Resource monitoring for $tag will be suspended until the issue is resolved. | **原因**: 指定されたデータベースのSAP HANAシステムレプリケーションモードがLifeKeeperの外部で変更されました。<br><br>**対応**:プライマリーマスターとして登録する必要のあるサーバーで、SAP HANAリソースをIn-Serviceにしてください。 |
| 136711 | EMERG | SAP HANA database $instance corresponding to reource $tag is currently suspended on server $me due to actions performed outside of LifeKeeper. Please take the SAP HANA resource out of service on ser | **原因**: 指定されたたサーバー上のSAP HANAデータベースインスタンスが、LifeKeeperの外部で行われた操作により一時停 |

| コード | 重大度 | メッセージ | 原因/処置 |
| --- | --- | --- | --- |
| | | ver $me and bring it in-service on the server where the database should be registered as primary master. Bringing resource $tag back in-service on $me will resume the suspended database. Resource monitoring for $tag will be suspended until the issue is resolved. | 止されています。<br><br>**対応**: 対応するLifeKeeperリソースが現在In-Service（Active）になっているサーバー上で一時停止中のデータベースを再開する場合は、メッセージに記載されているコマンドを実行してください。それ以外の場合は、目的のプライマリーレプリケーションサイトでLifeKeeper SAP HANAリソースをIn-Serviceにしてください。 |

# 9. LifeKeeper for Linuxサポートマトリックス

サポート対象のオペレーティングシステム
サポート対象のアプリケーション
サポート対象の仮想化環境

## サポート対象のオペレーティングシステム

| 製品 | サポート対象のオペレーティングシステム | LifeKeeper for Linux | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | v9.5.0 | v9.5.1 | v9.5.2 | v9.6.0 | v9.6.1 | v9.6.2 | v9.7.0 |
| LifeKeeper for Linux | Red Hat Enterprise Linux 9 (*18) | | | | | | | 9.0 ~ 9.1 64 ビット (*4) (*5) (*23 ※9.0 のみ) |
| | Red Hat Enterprise Linux 8 (*18) | 8.0, 8.1 64ビット (*4) (*5) | 8.0 ~ 8.3 64ビット (*4) (*5) (*15) (*16) | 8.0 ~ 8.4 64ビット (*4) (*5) | 8.0 ~ 8.4 64ビット (*4) (*5) | 8.0 ~ 8.5 64ビット (*4) (*5) | 8.0 ~ 8.6 64ビット (*4) (*5) (*23) | 8.0 ~ 8.7 64ビット (*4) (*5) (*23 ※8.6 のみ) (*27) |
| | Red Hat Enterprise Linux 7 (*18) | 7.0 ~ 7.8 64ビット (*2) (*13) | 7.0 ~ 7.9 64ビット (*2) (*14) | 7.0 ~ 7.9 64ビット (*2) | 7.0 ~ 7.9 64ビット (*2) | 7.0 ~ 7.9 64ビット (*2) | 7.0 ~ 7.9 64ビット (*2) | 7.0 ~ 7.9 64ビット (*2) |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Red Hat Enterprise Linux 6 | 6.0 ~ 6.10<br>64ビット<br>(*1) | 6.0 ~ 6.10<br>64ビット<br>(*1) | | | | | |
| | SUSE Linux Enterprise Server (SLES) 15<br>(*18) (*19) (*25) | 15.0 SP1<br>64ビット<br>(*5)<br>(*8) | 15.0 SP1 ~ SP2<br>64ビット<br>(*5)<br>(*8)<br>(*16) | 15.0 SP1 ~ SP2<br>64ビット<br>(*5)<br>(*8) | 15.0 SP1 ~ SP3<br>64ビット<br>(*5)<br>(*8) | 15.0 SP1 ~ SP3<br>64ビット<br>(*5)<br>(*8) | 15.0 SP1 ~ SP4<br>64ビット<br>(*5)<br>(*8)<br>(*24)<br>注意：SLES15 SP4にLifeKeeperをインストールするには こちら の手順に従ってlk-sles15sp4-supportをダウンロードし‘setup —addinfo lk-sles15sp4-support’を実行してください。 | 15.0 SP1 ~ SP4<br>64ビット<br>(*5)<br>(*8)<br>(*24) |
| | SUSE Linux Enterprise Server (SLES) 12<br>(*7) (*18) (*25) | 12SP1 ~ SP5<br>64ビット | 12SP1 ~ SP5<br>64ビット<br>(*16) | 12SP1 ~ SP5<br>64ビット | 12SP1 ~ SP5<br>64ビット | 12SP1 ~ SP5<br>64ビット | 12SP1 ~ SP5<br>64ビット | 12SP1 ~ SP5<br>64ビット |
| | SUSE Linux Enterprise Server (SLES) 11<br>(*6) (*25) | | | | | | | |
| | Oracle Linux 9 | | | | | | | 9.0 ~ 9.1<br>64ビ |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | (*18) | | | | | | | ット (*4) (*5) (*23 ※9.0 のみ) |
| | Oracle Linux 8 (*18) | 8.0, 8.1 (*11) | 8.0 ～ 8.2 (UEK R6 は含まない) (*5) (*11) (*16) | 8.0 ～ 8.3 (UEK R6 含む) (*5) (*11) (*17) | 8.0 ～ 8.4 (UEK R6 含む) (*5) (*11) (*17) | 8.0 ～ 8.5 (UEK R6 含む) (*5) (*11) (*17) | 8.0 ～ 8.6 (UEK R6 含む) (*4) (*5) (*23 ) (*25) | 8.0 ～ 8.7 (UEK R6, R7 含む) (*4) (*5) (*17 ) (*18 ) (*23 ) (*27 ※8.7 のみ) |
| | Oracle Linux 7 (*18) | 7.0 ～ 7.8 (UEK R3, R4, R5 含む) 64 ビット (*2) (*13) | 7.0 ～ 7.9 (UEK R3, R4, R5 含む) 64 ビット (*2) (*14) (*16) | 7.0 ～ 7.9 (UEK R3, R4, R5, R6 含む) 64 ビット (*2) (*17) | 7.0 ～ 7.9 (UEK R3, R4, R5, R6 含む) 64 ビット (*2) (*17) | 7.0 ～ 7.9 (UEK R3, R4, R5, R6 含む) 64 ビット (*2) (*17) | 7.0 ～ 7.9 (UEK R3, R4, R5, R6 含む) 64 ビット (*2) (*17) | 7.0 ～ 7.9 (UEK R3, R4, R5, R6 含む) 64 ビット (*2) (*17) |
| | Oracle Linux 6 | 6.3 ～ 6.10 (UEK R3、R4含む) | 6.3 ～ 6.10 (UEK R3、R4含む) | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 64ビット | 64ビット | | | | | |
| | CentOS 8 (*18) | 8.0, 8.1 (*12) | 8.0 ~ 8.2 (*5) (*12) (*16) | 8.0 ~ 8.3 (*5) (*12) | 8.0 ~ 8.3 (*5) (*12) | 8.0 ~ 8.3 (*5) (*12) | 8.0 ~ 8.3 (*5) (*12) | 8.0 ~ 8.3 (*5) (*12) |
| | CentOS 7 (*18) | 7.0 ~ 7.8 64ビット (*2) (*13) | 7.0 ~ 7.9 64ビット (*2) (*14) | 7.0 ~ 7.9 64ビット (*2) | 7.0 ~ 7.9 64ビット (*2) | 7.0 ~ 7.9 64ビット (*2) | 7.0 ~ 7.9 64ビット (*2) | 7.0 ~ 7.9 64ビット (*2) |
| | CentOS 6 (*10) | 6.0 ~ 6.10 64ビット | 6.0 ~ 6.10 64ビット | | | | | |
| | Rocky Linux 9 | | | | | | | 9.0 (*22) (*23) (*26) |
| | Rocky Linux 8 | | | | | 8.4 のみ (*20) (*21) (*22) | 8.4, 8.5, 8.6 (*20) (*21) (*22) (*23 ※8.6 のみ) | 8.4 ~ 8.7 (*20) (*21) (*22) (*23 ※8.6 のみ) (*27) |
| | Miracle Linux 9 (*18) | | | | | | | 9.0 64 ビット (*4) |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | (*5) (*23 ) |
| | Miracle Linux 8 | | | | | | 8.4 | 8.4 ,8.6 (*23 ※8.6 のみ ) |

*1 6.0 は推奨されません
*2 一部のKernel versionでは非同期モードはサポートされません。
*3 7.7でDataKeeperを使用する場合、LifeKeeperインストール時に 特定の手順 に従う必要があります。
*4 RHEL 7からRHEL8、RHEL8からRHEL9のようにKernelをメジャーバージョンアップすることはサポートされません。
*5 DataKeeperにおいてセクターサイズが奇数のディスクは利用できません。
*6 *SLES11SP2はカーネルを3.0.42-0.7.3にアップグレードしてください。*
*7 *SLES12SP3はカーネルを4.4.82-6.9.1にアップグレードしてください。*
*8 SLES12 からSLES15のように Kernel をメジャーバージョンアップすることはサポートされません。
*9 7.5/7.6 ではUEK R5はサポートされません。
*10 *CentOS 6.0 – DataKeeper 設定はサポートされません。この制限は、各バージョンの過去のリリースノートに記載されていません。*
*11 OEL7からOEL8のように Kernel をメジャーバージョンアップすることはサポートされません。
*12 CentOS7からCentOS8のように Kernel をメジャーバージョンアップすることはサポートされません。
*13 7.8で DataKeeperを使用する場合、LifeKeeperインストール時に 特定の手順 に従う必要があります。
*14 7.9で DataKeeperを使用する場合、LifeKeeperインストール時に 特定の手順 に従う必要があります。
*15 8.3で DataKeeperを使用する場合、LifeKeeperインストール時に 特定の手順 に従う必要があります。
*16 重要 :いくつかの Linux リリースの md/raid1カーネルモジュールで発見された問題があります。この問題により、DataKeeper クラスターノードを使用している LifeKeeper for Linux のサポートされているバージョン（v9.3.2～v9.5.1）の部分的な再同期では、すべてのブロックが再同期されない場合があります。影響を受けるディストリビューションおよびカーネルバージョンなどの詳細については こちら をご確認ください。
*17 UEK R6 はカーネルを 5.4.17-2102.202.5以降にアップグレードしてください。
*18 非同期モードで作成し同期モードで拡張したDataKeeperリソースの構成はサポートされません。
*19 SLES 15 SP3はカーネルを 5.3.18-59.5にアップグレードしてください。
*20 Rocky Linux8.4から8.6はAWS/Azure/GCP/OCIなどのクラウド上での利用はサポートされません。
*21 Rocky Linux8.4から8.6はSingle Server Protectionの利用はサポートされません。
*22 SAP Recovery KitとSAP HANA Recovery Kitはサポートされていません。
*23 kernel の問題に起因してシンプロビジョニングディスクで DataKeeper の非同期モードは利用できません。
*24 カーネル 5.14.21-150400.24.21.2 以上が必要です。
*25 *SUSEライブパッチはサポートされていません。*
*26 *Rocky Linux 9.0はGCPとOCIをサポートしていません。*
*27 8.7でDataKeeper を使用する場合、LifeKeeperインストール時に 特定の手順 に従う必要があります

# サポート対象のアプリケーション

| 製品 | サポート対象のアプリケーション | LifeKeeper for Linux | | | | |
|---|---|---|---|---|---|---|
| | | v9.5.0 | v9.5.1 | v9.5.2 | v9.6.0 | v9.6.1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Apache ARK** | Apache HTTP Server | 2.4 | 2.4 | 2.4 | 2.4 | 2.4 |
| **SAP ARK**<br><br>**(SAP**がサポートする**OS**につきましては **LifeKeeper for SAP Solution Page** を参照してください。**)** | SAP NetWeaver | 7.0 including Enhancement Package 1-3,<br><br>7.3 including Enhancement Package 1, 7.4, 7.5 | 7.0 including Enhancement Package 1-3,<br><br>7.3 including Enhancement Package 1, 7.4, 7.5 | 7.0 including Enhancement Package 1-3,<br><br>7.3 including Enhancement Package 1, 7.4, 7.5 | 7.0 including Enhancement Package 1-3,<br><br>7.3 including Enhancement Package 1, 7.4, 7.5 | 7.0 including Enhancement Package 1-3,<br><br>7.3 including Enhancement Package 1, 7.4, 7.5 |
| | SAP NetWeaver AS for ABAP | 7.51 innovation package, 7.52 innovation package | 7.51 innovation package, 7.52 innovation package | 7.51 innovation package, 7.52 innovation package | 7.51 innovation package, 7.52 innovation package | 7.51 innovation package, 7.52 innovation package |
| | SAP S/4HANA Platform | 1809, 1909 Platforms | 1809, 1909, 2020 Platforms | 1809, 1909, 2020 Platforms | 1809, 1909, 2020, 2021 Platforms | 1809, 1909, 2020, 2021 Platforms |
| **SAP MaxDB ARK** | SAP MaxDB | 7.9 | 7.9 | 7.9 | 7.9 | 7.9 |
| **SAP HANA ARK (SAP HANA** がサポートする **OS** につきましては **SAP HANA** サポート構成**)** を参照してください。 | SAP HANA | 2.0 SPS 04<br><br>SAP HANA 2.0 SPS04より前のバージョンはサポートされません。 | 2.0 SPS 04, 05<br><br>SAP HANA 2.0 SPS04より前のバージョンはサポートされません。 | 2.0 SPS 04, 05<br><br>SAP HANA 2.0 SPS04より前のバージョンはサポートされません。 | 2.0 SPS 04, 05, 06<br><br>SAP HANA 2.0 SPS04より前のバージョンはサポートされません。 | 2.0 SPS 04, 05, 06<br><br>SAP HANA 2.0 SPS04より前のバージョンはサポートされません。 |
| **Postfix ARK** | Postfix | (**1) | (**1) | (**1) | (**1) | (**1) |
| **DB2 ARK** | IBM Db2 Advanced, Standard, Community Editions<br><br>**注:** Advanced Enterprise Server | 11.5 | 11.5 | 11.5 | 11.5 | 11.5 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | Edition および Enterprise Server Edition は、Db2 Advanced Edition と呼ばれるようになりました | | | | | |
| | IBM Db2 Enterprise Server Edition (ESE) および Workgroup Server Edition (WSE) | 10.5, 11.1 | 10.5, 11.1 | 10.5, 11.1 | 10.5, 11.1 | 10.5, 11.1 |
| | IBM Db2 Express Edition | 10.5, 11.1 | 10.5, 11.1 | 10.5, 11.1 | 10.5, 11.1 | 10.5, 11.1 |
| | IBM Db2 Advanced Workgroup Server Edition | | 10.5, 11.1 | 10.5, 11.1 | 10.5, 11.1 | 10.5, 11.1 |
| | IBM Db2 Advanced Enterprise Server Edition | | 10.5, 11.1 | 10.5, 11.1 | 10.5, 11.1 | 10.5, 11.1 |
| **PostgreSQL ARK** | PostgreSQL | 9.5, 9.6,10,11,12 | 9.5, 9.6,10,11,12,13 | 9.6,10,11,12,13 | 9.6,10,11,12,13,14 | 9.6,10,11,12,13, |
| | EnterpriseDB Postgres Plus Advanced Server/EDB Postgres Advanced | 9.5, 9.6,10,11,12 | 9.5, 9.6,10,11,12,13 | 9.6,10,11,12,13 | 9.6,10,11,12,13,14 | 9.6,10,11,12,13, |

| | | | | | | |
|---|---|---|---|---|---|---|
| | Server | | | | | |
| | Symfoware Server Enterprise Edition | 12.2, 12.3, 12.4 | 12.2, 12.3, 12.4 | 12.2, 12.3, 12.4 | 12.2, 12.3, 12.4 | 12.2, 12.3, 12.4 |
| | Symfoware Server Standard Edition | 12.2, 12.3, 12.4 | 12.2, 12.3, 12.4 | 12.2, 12.3, 12.4 | 12.2, 12.3, 12.4 | 12.2, 12.3, 12.4 |
| | Symfoware Server Lite Edition | 12.3 | 12.3 | 12.3 | 12.3 | 12.3 |
| | FUJITSU Software Enterprise Postgres Advanced Enterprise Server Edition | | | | | 14SP1 |
| | FUJITSU Software Enterprise Postgres Advanced Edition | 9.5, 10, 11 | 9.5, 10, 11, 12 | 10, 11, 12 | 10, 11, 12, 13 | 10, 11, 12, 13, 1 |
| | FUJITSU Software Enterprise Postgres Standard Edition | 9.5, 9.6, 10, 11 | 9.5, 9.6, 10, 11, 12 | 9.6, 10, 11, 12 | 9.6, 10, 11, 12, 13 | 9.6, 10, 11, 12, 14, 14SP1 |
| | FUJITSU Software Enterprise Postgres Community Edition | 10, 11 | 10, 11, 12 | 10, 11, 12 | 10, 11, 12, 13 | 10, 11, 12, 13, 1 14SP1 |
| | PowerGres on Linux | 11 | 11 | 11 | 11 | 11, 13 |
| | PowerGres Plus | 10 (**7) | 10 (**7) | 10 (**7) | 10 (**7) | 10 (**7) |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Oracle ARK** | Oracle Database Enterprise Edition | 12c, 12c R2, 18c, 19c | 12c, 12c R2, 18c, 19c | 12c, 12c R2, 18c, 19c | 12c, 12c R2, 18c, 19c, 21c | 12c, 12c R2, 18c, 19c, 21c |
| | Oracle Database Standard Edition 2 | 12c, 12c R2, 18c, 19c | 12c, 12c R2, 18c, 19c | 12c, 12c R2, 18c, 19c | 12c, 12c R2, 18c, 19c, 21c | 12c, 12c R2, 18c, 19c, 21c |
| | Oracle Database Standard Edition | | | | | |
| | Oracle Database Standard Edition One | | | | | |
| **Sybase ASE ARK** | SAP Adaptive Server Enterprise | 15.7, 16.0 | 15.7, 16.0 | 15.7, 16.0 | 15.7, 16.0 | 15.7, 16.0 |
| **Samba ARK** | Standard Samba file shares | (**1) | (**1) | (**1) | (**1) | (**1) |
| **NFS Server ARK** | Linux kernel version | Kernel 2.6 以降<br>(**2)<br>(**3) | Kernel 2.6 以降<br>(**2)<br>(**3) | Kernel 2.6 以降<br>(**2)<br>(**3) | Kernel 2.6 以降<br>(**2)<br>(**3) | Kernel 2.6 以降<br>(**2)<br>(**3) |
| **NAS ARK** | NFS version of Mounted NFS file systems from an NFS server or Network Attached Storage (NAS) device | NFSv2<br>(**2)<br>NFSv3<br>NFSv4<br>(**3) | NFSv2<br>(**2)<br>NFSv3<br>NFSv4<br>(**3) | NFSv2<br>(**2)<br>NFSv3<br>NFSv4<br>(**3) | NFSv2<br>(**2)<br>NFSv3<br>NFSv4<br>(**3) | NFSv2<br>(**2)<br>NFSv3<br>NFSv4<br>(**3) |
| **MySQL ARK** | MySQL Community Edition, MySQL | 5.7, 8.0 | 5.7, 8.0 | 5.7, 8.0 | 5.7, 8.0 | 5.7, 8.0 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | Enterprise Edition | | | | | |
| | MariaDB | 10.3, 10.4 (**4) | 10.3, 10.4 (**4) | 10.3, 10.4, 10.5 (**4) | 10.3, 10.4, 10.5 (**4) | 10.3, 10.4, 10.5 (**4) |
| **WebSphere MQ ARK** | WebSphere MQ/IBM MQ | 8.0, 9.0, 9.1 | 8.0, 9.0, 9.1 | 8.0, 9.0, 9.1, 9.2 | 8.0, 9.0, 9.1, 9.2 | 8.0, 9.0, 9.1, 9.2 |
| **Software RAID (md) ARK** | RHEL6, OL6, CentOS6 | 6.0 ～ 6.7 | 6.0 ～ 6.7 | | | |
| | SLES11 | 11.0 ～ SP1 | | | | |
| **VMDK as Shared Storage ARK** | VMware vSphere | 6.5, 6.7, 7.0 (**6) (**7) (**8) (**9) | 6.5, 6.7, 7.0 (**6) (**7) (**9) | 6.5, 6.7, 7.0 (**6) (**7) (**9) | 6.5, 6.7, 7.0 (**6) (**7) | 6.5, 6.7, 7.0 (**6) (**7) |

**1 サポート対象のLinuxディストリビューションに付属
**2 NFSv2はRHEL7/ CentOS7/ OL7以降ではサポートされません。
**3 NFS over UDPはRHEL8以降ではサポートされません。
**4 10.1、10.2を除く
**5 9.0.5はサポートされません。
**6 RHEL6.x/CentOS6.x/OEL6.x および SLES11.x をサポートしていません。
**7 LifeKeeper Single Server Protection for Linuxではサポートしていません。
**8 VMDK ARKでリソースの作成する前に別途手順が必要です。詳細については こちら を確認してください。
**9 サービスの停止を伴わないLifeKeeperの停止を実施した場合、システム全体が停止する可能性があります。詳細については こちら をご確認ください。

## サポート対象の仮想化環境

| 製品 | サポート対象の仮想化環境 | LifeKeeper for Linux | | | | | |
|---|---|---|---|---|---|---|---|
| | | v9.5.0 | v9.5.1 | v9.5.2 | v9.6.0 | v9.6.1 | v9.6.2 |
| **LifeKeeper for Linux** | VMware vSphere (***1) | 5.5, 6.0, 6.5, 6.7 | 5.5, 6.0, 6.5, 6.7, 7.0 | 5.5, 6.0, 6.5, 6.7, 7.0 | 6.5, 6.7, 7.0 | 6.5, 6.7, 7.0 | 6.5, 6.7 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | VMware Cloud on AWS (***2) | – | – | – | – | – | – |
| | KVM | (***4) | (***4) | (***4) | (***4) | (***4) | (***4) |
| | Oracle VM Server for x86 (***2) | 3 | 3 | 3 | 3 | 3 | 3 |
| | Nutanix Acropolis Hypervisor（AHV） | 20160925.57<br>20160925.90 | 20160925.57<br>20160925.90 | 20160925.57<br>20160925.90 | 20160925.57<br>20160925.90 | 20160925.57<br>20160925.90 | 20160<br>20160 |

*****1 VMware vSphere v7.0 以降では vSAN の利用は VMware Cloud on AWS 上でのみサポートされます。**
*****2 シングルAZ構成にて、Reservationを使用しない共有ディスク（別途 IOフェンシングの対応 が必須）、及びDataKeeperでの利用が可能です。**
*****3 SDDC バージョン**
*****4 サポート対象の Linux ディストリビューションに付属するバージョン**

LifeKeeper が現在サポートしているディスクアレイストレージモデルやアダプタ、およびそれらの認定の種類の一覧については サポートストレージ一覧 を参照してください。

SAP関連のバージョン要件は「 LifeKeeper for SAP ソリューションページ 」をご確認ください。

## サポートしているクラウド環境

- Amazon EC2
- Microsoft Azure
  - Azure 共有ディスク
- Google Cloud
- Oracle Cloud Infrastructure
- Virtustream – Dell Technologies

LifeKeeperとの組み合わせで利用可能なクラウド環境については こちら を参照してください。

# 10. サポートストレージ

## LifeKeeper for Linux v9 サポートストレージ一覧

下記の表は、LifeKeeper for Linuix v9 環境で利用可能な、認定済の共有ストレージの一覧表です。ハードウェアの構成検討の際に確認してください。

### 認定済み共有ストレージについて

LifeKeeper の共有ストレージとして利用するストレージは、種類によって認定が必要なものと不要なものがあります。

以下のストレージは、認定されたものを使用する必要があります。下記の表に使用予定のストレージが記載されていることを確認してください。

- SCSI / FC / iSCSI / SAS など、SCSI-2/3 Reservation による I/O フェンシングを LifeKeeper で行うことが前提となる、複数ノードで同一のデータを参照する共有ストレージ

### 認定が不要な構成

以下の構成にストレージ認定はありません。

- NAS ストレージ (NAS RecoveryKit が必要です)
- DataKeeper による DataReplication を構成するディスク装置全般(内蔵/外付けを問いません)
- vSphere上でVMDK ARKによって保護される共有ストレージ上の仮想ディスク
- 以下の条件をすべて満たす環境で利用されるストレージ
  - OSやハードウェア、プラットフォームでサポート済みのストレージである
  - LifeKeeper のSCSI Reservation 機能をオフにする
  - LifeKeeper のQuorum/Witnessによるフェンシング機能を採用する
    ※上記3つの条件を満たす構成で利用される共有ストレージは、弊社の認定が無いものであっても利用することができます。

注記 ※共有ストレージが上記の要件を満たす場合、SIOS認証は必要ありません。

### ご利用いただけないハードウェア

USB や IEEE1394 などで接続する、コンシューマー向けストレージの利用はサポートされません。

## Azure

| ベンダー名 (*1) | ストレージモ | バスタイプ | 接続構成 (*2) | サポート可否 | 必要な ARK(*3) (マルチパス ARK等) | サポートする LKのバージョ | 注意事項 |
|---|---|---|---|---|---|---|---|

| | デル名 | | | | | ン(*4) | |
|---|---|---|---|---|---|---|---|
| Azure | 共有ディスク | – | Single Path | YES | SCSI3 RK | v9.7.0 | LifeKeeper は、リージョン内の共有ストレージとして Microsoft Azure 共有ディスクをサポートしており、ローカル冗長ストレージ (LRS) とゾーン冗長ストレージ (ZRS) の両方がサポートされています。 LifeKeeper は、リージョン間の共有ストレージとして Microsoft Azure 共有ディスクをサポートしていません。GEO 冗長ストレージと GEO ゾーン冗長ストレージの両方がサポートされていません。<br><br>以下がサポートされています。<br>• SUSE Linux Enterprise Server (SLES) 15 SP3 以降<br>• Red Hat Enterprise Linux 8.5 以降<br>• Red Hat Enterprise Linux 7.9<br>• Oracle Linux |

## DELL

| ベンダー名 (*1) | ストレージモデル名 | バスタイプ | 接続構成 (*2) | サポート可否 | 必要な ARK(*3) (マルチパス ARK等) | サポートする LKのバージョン(*4) | 注意事項 |
|---|---|---|---|---|---|---|---|
| DELL | Dell EqualLogic: PS4100 PS4110 PS4210 PS6100 PS6110 PS6210 PS6610 | ISCSI | Multi Path (DMMP) | YES | DMMP ARK | v9.0 – v9.7.0 | LUN 数が多い場合 (20 以上) は、 */etc/default/LifeKeeper* の REMOTETIMEOUT 設定を REMOTETIMEOUT=600 に変更してください。 |
| | | | Single Path (DMMP) | YES | DMMP ARK | | |
| | MD3800f MD3820f MD3860f | FC | vSphere (RDM) | No | – | v9.0 – v9.7.0 | |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single Path | No | – | | |
| | MD3400 | SAS | vSphere | No | – | v9.0 – | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | MD3420<br>MD3460 | | (RDM) | | | v9.7.0 | |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single Path | No | – | | |
| | SCv3000 Series:<br>SCv3000<br>SCv3020 | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 | |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single Path | No | – | | |
| | | ISCSI | Multi Path (DMMP) | YES | DMMP ARK | v9.0 – v9.7.0 | |
| | | | Single Path | No | – | | |
| | | SAS | vSphere | No | – | v9.0 – v9.7.0 | |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single Path | No | – | | |
| | SCv2000 Series:<br>SCv2000<br>SCv2020<br>SCv2080 | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 | |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single Path | No | – | | |
| | | ISCSI | Multi Path (DMMP) | YES | DMMP ARK | v9.0 – v9.7.0 | |
| | | | Single Path | No | – | | |
| | | SAS | vSphere | No | – | v9.0 – | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | Multi Path (DMMP) | YES | DMMP ARK | v9.7.0 | |
| | | | Single Path | No | – | | |
| | Dell Storage: SC7020 SC5020 SC4020 | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 | |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single Path | No | – | | |
| | | ISCSI | Multi Path (DMMP) | YES | DMMP ARK | v9.0 – v9.7.0 | |
| | | | Single Path | No | – | | |
| | Dell Compellent: SC9000 SC8000 | FC | vSphere (RDM) | No | – | – | |
| | | | Multi Path (DMMP) | No | – | | |
| | | | Single Path | No | – | | |
| | | ISCSI | Multi Path (DMMP) | YES | DMMP ARK | v9.0 – v9.7.0 | |
| | | | Single Path | No | – | | |

## Fujitsu

| ベンダー名 **(*1)** | ストレージモデル名 | バスタイプ | 接続構成 **(*2)** | サポート可否 | 必要な **ARK(*3)** **(**マルチパス **ARK**等**)** | サポートする **LK**のバージョン**(*4)** | 注意事項 |
|---|---|---|---|---|---|---|---|

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Fujitsu** | ETERNUS:<br>DX60 S2<br>DX80 S2<br>DX90 S2<br>DX410 S2<br>DX440 S2<br>DX8100 S2<br>DX8700 S2<br>DX60 S3<br>DX100 S3<br>DX200 S3<br>DX500 S3<br>DX600 S3<br>DX8700 S3<br>DX8900 S3<br>DX200F<br>DX60 S4<br>DX100 S4<br>DX200 S4<br>DX500 S4<br>DX600 S4<br>DX60 S5<br>DX100 S5<br>DX200 S5<br>DX500 S5<br>DX600 S5<br>DX900 S5<br>AF250<br>AF650<br>AF250 S2<br>AF650 S2<br>AF250 S3<br>AF650 S3<br>DX8900 S4 | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 | ETERNUSをBootデバイスとして使用した場合、プライマリノードからのBootデバイスアクセスが不可となったときに、ファイルシステムエラーが発生し、セカンダリノードへ切り替わらない可能性があります。<br>対応として、以下のようにファイルシステムエラー時にパニックするように設定してください。<br>例：xfsの場合は下記を設定後サーバをリブートしてください。<br><br>`echo 'fs.xfs.panic_mask=127' > /etc/sysctl.d/01-xfs.conf` |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Multi Path (EMPD) | YES | No Multipath ARK | | |
| | | | Single Path | YES | No Multipath ARK | | |
| | | ISCSI | Multi Path (DMMP) | YES | DMMP ARK | v9.0 – v9.7.0 | |
| | | | Multi Path (EMPD) | YES | No Multipath ARK | | |
| | | | Single Path | YES | No Multipath ARK | | |

## Hitachi / Hitachi Vantara

| ベンダー名 (*1) | ストレージモデル名 | バスタイプ | 接続構成 (*2) | サポート可否 | 必要な ARK(*3) (マルチパスARK等) | サポートする LKのバージョン(*4) |
|---|---|---|---|---|---|---|

| | | | | | | |
|---|---|---|---|---|---|---|
| **Hitachi / Hitachi Vantara** | Hitachi Virtual Storage Platform G1000<br>Hitachi Virtual Storage Platform G1500<br>Hitachi Virtual Storage Platform F1500 | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 |
| | | | Multi Path (HDLM) | YES | HDLM ARK | |
| | | | Multi Path (DMMP) | YES | DMMP ARK | |
| | | | Single Path | YES | – | |
| | | ISCSI | Multi Path (HDLM) | No | – | – |
| | | | Single Path | No | – | |
| | Hitachi Virtual Storage Platform | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 |
| | | | Multi Path (HDLM) | YES | HDLM ARK | |
| | | | Single Path | YES | – | |
| | | ISCSI | Multi Path (HDLM) | No | – | – |
| | | | Single Path | No | – | |
| | Hitachi Virtual Storage Platform Mid Range Family:<br>Hitachi Virtual Storage Platform F900 (VSP F900)<br>Hitachi Virtual Storage Platform F700 (VSP F700)<br>Hitachi Virtual Storage Platform F370 (VSP F370)<br>Hitachi Virtual Storage Platform F350 (VSP F350)<br>Hitachi Virtual Storage Platform G900 (VSP G900)<br>Hitachi Virtual Storage Platform G700 (VSP G700)<br>Hitachi Virtual Storage Platform G370 (VSP G370)<br>Hitachi Virtual Storage Platform | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 |
| | | | Multi Path (HDLM) | YES | HDLM ARK | |
| | | | Multi Path (DMMP) | YES | DMMP ARK | |
| | | | Single Path | YES | No Multipath ARK | |
| | | ISCSI | Multi Path (DMMP) | YES | DMMP ARK | v9.0 – v9.7.0 |
| | | | Single Path | YES | No Multipath ARK | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | G350 (VSP G350)<br>Hitachi Virtual Storage Platform G150 (VSP G150)<br>Hitachi Virtual Storage Platform G130 (VSP G130) | | | | | |
| | Hitachi Virtual Storage Platform Mid Range Family:<br>Hitachi Virtual Storage Platform F800 (VSP F800)<br>Hitachi Virtual Storage Platform F600 (VSP F600)<br>Hitachi Virtual Storage Platform F400 (VSP F400)<br>Hitachi Virtual Storage Platform G800 (VSP G800)<br>Hitachi Virtual Storage Platform G600 (VSP G600)<br>Hitachi Virtual Storage Platform G400 (VSP G400)<br>Hitachi Virtual Storage Platform G200 (VSP G200)<br>Hitachi Virtual Storage Platform G100 (VSP G100) | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 |
| | | | Multi Path (HDLM) | YES | HDLM ARK | |
| | | | Multi Path (DMMP) | YES | DMMP ARK | |
| | | | Single Path | YES | No Multipath ARK | |
| | | ISCSI | Multi Path (DMMP) | YES | DMMP ARK | v9.0 – v9.7.0 |
| | | | Single Path | YES | No Multipath ARK | |
| | Hitachi Unified Storage VM (HUS VM) | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 |
| | | | Multi Path (HDLM) | YES | HDLM ARK | |
| | | | Single Path | YES | No Multipath ARK | |
| | | ISCSI | Multi Path (HDLM) | No | – | – |
| | | | Single Path | No | – | |
| | Hitachi Unified Storage 100 Series:<br>Hitachi Unified Storage 150 (HUS 150)<br>Hitachi Unified Storage 130 (HUS 130)<br>Hitachi Unified Storage 110 (HUS 110) | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 |
| | | | Multi Path (HDLM) | YES | HDLM ARK | |
| | | | Single Path | YES | No Multipath | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | ARK | |
| | | ISCSI | Multi Path (HDLM) | No | – | – |
| | | | Single Path | No | – | |
| | Hitachi Virtual Storage Platform E390 (VSP E390)<br>Hitachi Virtual Storage Platform E590 (VSP E590)<br>Hitachi Virtual Storage Platform E790 (VSP E790)<br>Hitachi Virtual Storage Platform E990 (VSP E990)<br>Hitachi Virtual Storage Platform E1090 (VSP E1090)<br>Hitachi Virtual Storage Platform E390H (VSP E390H)<br>Hitachi Virtual Storage Platform E590H (VSP E590H)<br>Hitachi Virtual Storage Platform E790H (VSP E790H)<br>Hitachi Virtual Storage Platform E1090 (VSP E1090H) | FC | vSphere (RDM) | YES | No Multipath ARK | v9.6.1 – v9.7.0 |
| | | | Multi Path (DMMP) | YES | DMMP ARK | |
| | | | Single Path | No | – | |
| | | ISCSI | Multi Path (DMMP) | No | – | – |
| | | | Single Path | No | – | |
| | Hitachi Virtual Storage Platform 5100 (VSP 5100)<br>Hitachi Virtual Storage Platform 5500 (VSP 5500)<br>Hitachi Virtual Storage Platform 5100H (VSP 5100H)<br>Hitachi Virtual Storage Platform 5500H (VSP 5500H) | FC | vSphere (RDM) | YES | No Multipath ARK | v9.6.1 – v9.7.0 |
| | | | Multi Path (DMMP) | YES | DMMP ARK | |
| | | | Single Path | No | – | |
| | | ISCSI | Multi Path (DMMP) | No | – | – |
| | | | Single Path | No | – | |
| | Hitachi Virtual Storage Platform 5200 (VSP 5200)<br>Hitachi Virtual Storage Platform 5500 (VSP 5500)<br>Hitachi Virtual Storage Platform 5200H (VSP 5200H)<br>Hitachi Virtual Storage Platform 5600H (VSP 5600H) | FC | vSphere (RDM) | YES | No Multipath ARK | v9.6.1 – v9.7.0 |
| | | | Multi Path (DMMP) | YES | DMMP ARK | |
| | | | Single Path | No | – | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | ISCSI | Multi Path (DMMP) | No | – | – |
| | | | Single Path | No | – | |

## Hitachi

| ベンダー名 (*1) | ストレージモデル名 | バスタイプ | 接続構成 (*2) | サポート可否 | 必要な ARK(*3) (マルチパス ARK等) | サポートする LKのバージョン(*4) | 注意事項 |
|---|---|---|---|---|---|---|---|
| Hitachi | BR1200 | ISCSI | Multi Path (HDLM) | No | – | – | |
| | | | Multi Path (DMMP) | No | – | | |
| | | | Single Path | No | – | | |
| | | SAS | Multi Path (RDAC) | YES | No Multipath ARK | v9.0 – v9.7.0 | シングルパスとマルチパスのいずれの場合も RDAC ドライバを使用する必要があります。<br><br>HDLM および HDLM ARK を使用する構成はサポートされません。 |
| | | | Single Path (RDAC) | YES | No Multipath ARK | | |
| | BR1250 | ISCSI | Multi Path (HDLM) | No | – | – | |
| | | | Multi Path (DMMP) | No | – | | |
| | | | Single Path | No | – | | |
| | | SAS | Multi Path (NECSPS) | YES | NEC SPS ARK | v9.0 – v9.7.0 | シングルパスとマルチパスのいずれの場合も SPS NEC iStorage StoragePathSavior (NECSPS) Recovery Kit を使用する必要があります。HDLM および HDLM ARK を使用する構成はサポートされません。 |
| | | | Single Path (NECSPS) | YES | NEC SPS ARK | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | SPS NEC iStorage StoragePathSavior (NECSPS) Recovery Kit を RHEL7 系環境で使用する場合は、LifeKeeper for Linux v9.1 以上をご使用ください。 |
| | BR1650E<br>BR 1650S | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 | |
| | | | Multi Path (HDLM) | YES | HDLM ARK | | |
| | | | Single Path | YES | – | | |
| | | ISCSI | Multi Path (HDLM) | No | – | – | |
| | | | Single Path | No | – | | |

## HPE

| ベンダー名 **(*1)** | ストレージモデル名 | バスタイプ | 接続構成 **(*2)** | サポート可否 | 必要な **ARK(*3)** **(**マルチパス **ARK**等**)** | サポートする **LK**のバージョン**(*4)** | 注意事項 |
|---|---|---|---|---|---|---|---|
| HPE | Alletra9060 | FC | Multi Path (DMMP) | YES | DMMP ARK | v9.0 – v9.7.0 | |
| | Primera A630, A650, A670, C630, C650, C670 | | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 | |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single Path | No | – | | |
| | Nimble Storage: AF20 AF20Q AF40 | ISCSI | Multi Path (DMMP) | YES | DMMP ARK | v9.2.2 – v9.7.0 | |
| | | | Single | No | – | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | AF60<br>AF80<br>AF1000<br>AF3000<br>AF5000<br>AF7000<br>AF9000<br>CS1000H<br>CS1000<br>CS3000<br>CS5000<br>CS7000<br>HF20C<br>HF20H<br>HF20<br>HF40<br>HF60 | | Path | | | | |
| | 3PAR:<br>3PAR StoreServ 8200<br>3PAR StoreServ 8400<br>3PAR StoreServ 8440<br>3PAR StoreServ 8450 | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 | |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single Path | No | – | | |
| | 3PAR:<br>3PAR StoreServ 7400<br>3PAR StoreServ 7400c<br>3PAR StoreServ 7440<br>3PAR StoreServ 7440c<br>3PAR StoreServ 7450 | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 | **注記** 3PAR StoreServ 7400 は、デフォルトのパスチェッカとのリザベーション競合を返します。この競合を回避するには、”/etc/default/LifeKeeper” 内に次のパラメータを設定 (追加) してください。<br>DMMP_REGISTRATION_TYPE=hba<br>ユーザーフレンドリーデバイスマッピングはサポートされません。“multipath.conf” に次のパラメータを設定してください。<br>“user_friendly_names no”<br><br>* vSphere(RDM)で使用する場合は、上記の設定は必要ありません。 |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single Path | No | – | | |
| | | ISCSI | Multi | YES | DMMP | v9.0 – | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 3PAR StoreServ 7450c | | Path (DMMP) | | ARK | v9.7.0 | 注記 3PAR StoreServ 7400 は、リザベーション競合を返す場合があります。この競合を回避するため、”/etc/default/LifeKeeper” 内に次のパラメータを設定 (追加) してください。<br>DMMP_REGISTER_IGNORE=TRUE |
| | | | Single Path | No | – | | |
| | 3PAR: 3PAR StoreServ 7200 3PAR StoreServ 7200c | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 | 注記 3PAR StoreServ 7200 は、デフォルトのパス チェッカーとの予約の競合を返します。 この競合を回避するには、”/etc/default/LifeKeeper” で次のパラメータを設定 (追加) してください。<br>DMMP_REGISTRATION_TYPE=hba |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single Path | No | – | | |
| | | ISCSI | Multi Path (DMMP) | No | – | – | |
| | | | Single Path | No | – | | |
| | 3PAR: 3PAR StoreServ 9450 3PAR StoreServ 20800 R2 3PAR StoreServ 20840 R2 3PAR StoreServ 20850 R2 | FC | vSphere (RDM) | YES | No Multipath ARK | v9.2.1 – v9.7.0 | |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single Path | No | – | | |
| | | ISCSI | Multi Path (DMMP) | No | – | – | |
| | | | Single Path | No | – | | |
| | 3PAR: 3PAR StoreServ 20450 3PAR StoreServ 20800 3PAR StoreServ | FC | vSphere (RDM) | YES | No Multipath ARK | v9.1 – v9.7.0 | |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single Path | No | – | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 20840 3PAR StoreServ 20850 | ISCSI | Multi Path (DMMP) | No | – | – | |
| | | | Single Path | No | – | | |
| | 3PAR StoreServ 10400 | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 | |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single Path | No | – | | |
| | | ISCSI | Multi Path (DMMP) | No | – | – | |
| | | | Single Path | No | – | | |
| | 3PAR StoreServ 10800 | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 | 注記 3PAR StoreServ 10800 FC は、デフォルトのパスチェッカー設定との予約競合を返します。 この競合を回避するには、”/etc/default/LifeKeeper” で次のパラメータを設定 (追加) してください。<br>DMMP_REGISTRATION_TYPE=hba<br><br>ユーザーフレンドリーデバイスマッピングはサポートされません。“multipath.conf” に次のパラメータを設定してください。<br>“user_friendly_names no”<br><br>* vSphere(RDM)で使用する場合は、上記の設定は必要ありません。 |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single Path | No | – | | |
| | | ISCSI | Multi Path (DMMP) | No | – | – | |
| | | | Single Path | No | – | | |
| | MSA: MSA2060 | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single Path | No | – | | |
| | | ISCSI | Multi Path (DMMP) | No | – | – | |
| | | | Single Path | No | – | | |
| | | SAS | vSphere (RDM) | No | – | | |
| | | | Multi Path (DMMP) | No | – | – | |
| | | | Single Path | No | – | | |
| | MSA: MSA2050 MSA2052 | FC | vSphere (RDM) | YES | No Multipath ARK | v9.2 – v9.7.0 | |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single Path | No | – | | |
| | | ISCSI | Multi Path (DMMP) | YES | DMMP ARK | v9.2 – v9.7.0 | |
| | | | Single Path | No | – | | |
| | | SAS | vSphere (RDM) | No | – | | |
| | | | Multi Path (DMMP) | No | – | – | |
| | | | Single Path | No | – | | |
| | MSA: MSA2042 | FC | vSphere (RDM) | YES | No Multipath | v9.1 – v9.7.0 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | ARK | | |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single Path | No | – | | |
| | | ISCSI | Multi Path (DMMP) | YES | DMMP ARK | v9.1 – v9.7.0 | |
| | | | Single Path | No | – | | |
| | | SAS | vSphere (RDM) | YES | No Multipath ARK | v9.1 – v9.7.0 | |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single Path | No | – | | |
| | MSA: MSA1040 MSA2040 | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 | |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single Path | No | – | | |
| | | ISCSI | Multi Path (DMMP) | YES | DMMP ARK | v9.0 – v9.7.0 | |
| | | | Single Path | No | – | | |
| | | SAS | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 | |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single | No | – | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | Path | | | | |
| | StoreVirtual (LeftHand) Series LeftHand OS version 12.6 | FC | vSphere (RDM) | No | – | – | LeftHand OS (SAN/iQ) バージョン 12.6 は、HPE StoreVirtual (LeftHand) ストレージでサポートされます。<br>HC250 / HC380、および仮想ストレージアプライアンスとしての StoreVirtual VSA を含む、すべての StoreVirtual シリーズに対応しています。<br>仮想ストレージアプライアンスの詳細については、LeftHand OS (SAN/iQ) の各バージョンのリリースノートをご参照ください。 |
| | | | Multi Path (DMMP) | No | – | | |
| | | | Single Path | No | – | | |
| | | ISCSI | Multi Path (DMMP) | YES | DMMP ARK | v9.1 – v9.7.0 | |
| | | | Single Path | No | – | | |
| | HP P9500 | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 | |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single Path | YES | No Multipath ARK | | |
| | XP7 Storage System | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 | |
| | | | Multi Path (HDLM) | YES | HDLM ARK | | |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single Path | YES | No Multipath ARK | | |

## IBM

| ベンダ | ストレージモデル名 | バスタイプ | 接続構成 (*2) | サポート可否 | 必要な ARK(*3) (マルチ | サポートする | 注意事項 |
|---|---|---|---|---|---|---|---|

| ー名 (*1) | | | | | パス ARK等) | LKのバージョン(*4) | |
|---|---|---|---|---|---|---|---|
| IBM | IBM SAN Volume Controller* *IBM TotalStorage Proven | FC | vSphere (RDM) | No | – | v9.0 – v9.7.0 | |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single Path | YES | No Multipath ARK | | |
| | | ISCSI | Multi Path (DMMP) | No | – | – | |
| | | | Single Path | No | – | | |
| | IBM Storwize V7000 | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 | |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single Path | No | – | | |
| | | ISCSI | Multi Path (DMMP) | YES | DMMP ARK Quorum/ Witness Server Kit | v9.0 – v9.7.0 | パートナーテストによりiSCSI (iscsi-initiator-utils-6.2.0.872-34.el6.x86_64) と DMMP (device-mapper-1.02.66-6.el6、device-mapper-multipath-0.4.9-46.el6) を使用する IBM Storwize V7000 (Firmware Version 6.3.0.1)を 認定しています。テストは、LifeKeeper for Linux v7.5 と RHEL 6.2 を使用して行われました。<br><br>**制限事項:** IBM Storwize V7000 は、Quorum/Witness Server Kit および STONITH を組み合わせて使用する必要があります。<br>詳細についてはご利用中の LifeKeeper のバージョン用のオンラインマニュアルにてご確認ください。 |
| | | | Single Path | No | – | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | http://docs.us.sios.com/ |
| | IBM Storwize V3700 | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 | |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single Path | No | – | | |
| | | ISCSI | Multi Path (DMMP) | No | – | – | |
| | | | Single Path | No | – | | |
| | | SAS | vSphere (RDM) | No | – | – | |
| | | | Multi Path (DMMP) | No | – | | |
| | | | Single Path | No | – | | |
| | IBM XIV Storage System | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 | LifeKeeper を使用する IBM XIV Storage System で 32 個以上の LUN を作成する必要がある場合、詳細については IBM の営業担当者にお問い合わせください。 |
| | | | Multi Path (DMMP) | YES | DMMP ARK | | |
| | | | Single Path (DMMP) | YES | DMMP ARK | | |
| | | ISCSI | Multi Path (DMMP) | No | – | – | |
| | | | Single Path | No | – | | |

# Lenovo

| ベンダ | ストレージモデル名 | バスタ | 接続構成 | サポート | 必要なARK(*3) | サポートする |
|---|---|---|---|---|---|---|

| 一名 (*1) | | イプ | (*2) | 可否 | (マルチパス ARK等) | LKのバージョン(*4) |
|---|---|---|---|---|---|---|
| Lenovo | Lenovo Storage V3700 V2<br>Lenovo Storage V3700 V2 XP | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 |
| | | | Multi Path (DMMP) | YES | DMMP ARK | |
| | | | Single Path | No | – | |
| | | ISCSI | Multi Path (DMMP) | No | – | – |
| | | | Single Path | No | – | |
| | | SAS | vSphere (RDM) | No | – | – |
| | | | Multi Path (DMMP) | No | – | |
| | | | Single Path | No | – | |
| | IBM Storwize V3700 | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 |
| | | | Multi Path (DMMP) | YES | DMMP ARK | |
| | | | Single Path | No | – | |
| | | ISCSI | Multi Path (DMMP) | No | – | – |
| | | | Single Path | No | – | |
| | | SAS | vSphere (RDM) | No | – | – |
| | | | Multi Path (DMMP) | No | – | |
| | | | Single Path | No | – | |

## NEC

| ベンダー名 (*1) | ストレージモデル名 | バスタイプ | 接続構成 (*2) | サポート可否 | 必要な ARK(*3) (マルチパス ARK等) | サポートする LKのバージョン(*4) | 注意事項 |
|---|---|---|---|---|---|---|---|

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| NEC | NEC iStorage: M10e M11e M100 M110 M300 M12e M120 M320 | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 | NEC iStorage StoragePathSavior (NECSPS) Recovery Kit を RHEL7 環境で使用する場合は、 LifeKeeper for Linux v9.1 以降をご使用下さい。 |
| | | | Multi Path (NECSPS) | YES | NECSPS ARK | | |
| | | | Single Path (NECSPS) | YES | NECSPS ARK | | |
| | | ISCSI | Multi Path (NECSPS) | YES | NECSPS ARK | v9.0 – v9.7.0 | |
| | | | Single Path (NECSPS) | No | – | | |
| | | SAS | vSphere (RDM) | No | – | v9.0 – v9.7.0 | |
| | | | Multi Path (NECSPS) | YES | NECSPS ARK | | |
| | | | Single Path (NECSPS) | YES | NECSPS ARK | | |
| | NEC iStorage: M310 M310F M320F M500 M510 M700 M710 M710F M520 M720 M720F | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 | |
| | | | Multi Path (NECSPS) | YES | NECSPS ARK | | |
| | | | Single Path (NECSPS) | YES | NECSPS ARK | | |
| | | ISCSI | Multi Path (NECSPS) | YES | NECSPS ARK | v9.0 – v9.7.0 | |
| | | | Single Path (NECSPS) | No | – | | |

## Pure Storage

| ベンダー名 (*1) | ストレージモデル名 | バスタイプ | 接続構成 (*2) | サポート可否 | 必要なARK(*3) (マルチパスARK等) | サポートするLKのバージョン (*4) |
|---|---|---|---|---|---|---|

| | | | | | | |
|---|---|---|---|---|---|---|
| Pure Storage | FA-400 Series | FC | vSphere (RDM) | YES | No Multipath ARK | v9.0 – v9.7.0 |
| | | | Multi Path (DMMP) | YES | DMMP ARK | |
| | | | Single Path | No | – | |
| | | ISCSI | Multi Path (DMMP) | No | – | – |
| | | | Single Path | No | – | |
| | FlashArray//m Series //m10 //m20 //m50 //m70 | FC | vSphere (RDM) | No | – | v9.0 – v9.7.0 |
| | | | Multi Path (DMMP) | YES | DMMP ARK | |
| | | | Single Path | No | – | |
| | FlashArray//x Series //x10 //x20 //x50 //x70 | FC | vSphere (RDM) | No | – | v9.0 – v9.7.0 |
| | | | Multi Path (DMMP) | YES | DMMP ARK | |
| | | | Single Path | No | – | |
| | FlashArray//x Series //X10R2 //X20R2 //X50R2 //X70R2 //X90R2 | FC | vSphere (RDM) | No | – | v9.3.2 – v9.7.0 |
| | | | Multi Path (DMMP) | YES | DMMP ARK | |
| | | | Single Path | No | – | |
| | | iSCSI | vSphere (RDM) | No | – | v9.0 – v9.7.0 |
| | | | Multi Path (DMMP) | YES | DMMP ARK | |
| | | | Single Path | No | – | |

## (*1) EMC ストレージのご利用に関して

LifeKeeper は E-Lab の認定を取得しております。EMC 社のストレージのサポート状況については下記のマトリクスをご参照ください。

https://elabnavigator.dell.com/eln/elnhome

## (*2) 接続構成の説明

| | |
|---|---|
| vSphere (RDM) | VMWare ESX サーバーで共有ディスクを Raw Device Mapping (RDM) で接続した構 |

| | |
|---|---|
| | 成 (*5) |
| Multi Path (DMMP) | Device Mapper Multipath を使用した共有ディスクのマルチパス構成 |
| Multi Path (EMPD) | ETERNUS Multipath Driver を使用した共有ディスクのマルチパス構成 |
| Multi Path (HDLM) | Hitachi Dynamic Link Manager を使用した共有ディスクのマルチパス構成 |
| Multi Path (RDAC) | IBM Redundant Disk Array Controller を使用した共有ディスクのマルチパス構成 |
| Multi Path (NECSPS) | iStorage StoragePathSavior を使用した共有ディスクのマルチパス構成 |
| Single Path (DMMP) | Device Mapper Multipath を使用した共有ディスクのシングルパス構成 |
| Single Path (RDAC) | IBM Redundant Disk Array Controller を使用した共有ディスクのシングルパス構成 |
| Single Path | 特別なマルチパスドライバを用いないシングルパス接続 |

## (*3) マルチパスソフトウェア

マルチパスソフトウェアの接続情報については必ず各ベンダーのサポート情報をご確認ください
マルチパスソフトウェアと LifeKeeper の対応の詳細は、リリースノートでご確認ください

## (*4) サポートする LifeKeeper のバージョン

このドキュメントは、LifeKeeper for Linux v9 を対象としています。
以前のバージョンのサポートストレージにつきましては、各バージョンのリリースノートをご確認ください。

## (*5) VMware vSphere での共有ディスクの利用について

以下の URL にある、VMware vSphere 上で共有ディスクを利用する際の構成ガイドをご確認ください。
https://lkdkuserportal.sios.jp/hc/ja/articles/360037730991

表の iSCSI 接続に関する記載は物理環境と vSphere 環境で共通です。その為、バスタイプ “iSCSI” の接続構成の項目に “vSphere” の掲載は行っていません。

# 11. LifeKeeper for Linux 評価ガイド

DataKeeper for Linux 評価ガイド

クラウド環境向けSIOS Protection Suite for Linux 評価ガイド

# 11.1. DataKeeper for Linux 評価ガイド

## 目的

本文書は、LifeKeeper for Linux（以下LifeKeeper） のDataKeeper for Linux（以下DataKeeper）評価のためのインストール、構成、および使用をサポートすることを目的としています。この製品は、ホストベース、ブロックレベルのデータレプリケーションをリアルタイムで行うことを可能にするものです。
本文書に示すプロセスには次の5つのフェーズがあります。

- フェーズ1 – DataKeeper for Linux インストールの準備
- フェーズ2 – DataKeeper for Linux ストレージの構成
- フェーズ3 – LifeKeeper のインストール
- フェーズ4 – DataKeeper for Linux クラスターの構成
- フェーズ5 – DataKeeper for Linux の環境テスト

# 11.1.1. DataKeeper for Linux 関連用語

以下に示す各用語は本文書を通じて使用されるものです。よくご存じのものもあるかもしれませんが、SIOSによる用語の定義や使い方をもう一度見ておくとよいでしょう。

## ネットワーク通信関連用語

**クロスオーバーケーブル** ： コンピューターデバイスをネットワークスイッチ、ハブ、ルーターに接続することなく直接相互接続するために用いられるケーブル。このケーブルによって他から切り離されたプライベートネットワークが構築され、ネットワーク内のシステム間にクラスター関連のデータレプリケーションのトラフィックを流すことが可能になります。

## LifeKeeper のサーバーのタイプ

**サーバー** ： ソフトウェアのアプリケーションプログラム実行専用のコンピューターシステム。

**クラスター** ： 1つのコンピューターシステムとして動作させるために集められた複数のサーバーの集合。LifeKeeperにおいては高可用性を実現することを目的とするHAクラスターを指す。

**ノード** ： クラスターに参加しているサーバー。

**アクティブノード** ： リソース階層が稼働中（IN SERVICE）であるノード。

**スタンバイノード** ： リソース階層が定義されているが稼働中ではないノード。このノードは、アクティブノードのリソース階層に何らかの不具合が生じた場合に当該リソース階層を稼働させるために使われます。

**プライマリーノード** ： 特定のリソース階層についてLifeKeeperの構成内で優先度が最も高いノード。通常時リソース階層に関するサービスを提供するのがこのノードです。

**セカンダリーノード** ： 特定のリソース階層についてLifeKeeperの構成内で優先度が2番目に高いノード。

**ソースサーバー** ： データレプリケーションが利用されているLifeKeeperクラスター内のアクティブノード。リソース階層が稼働中であり、複製されたパーティション（ソースパーティション）にアクセスして書き込みを行うことが可能です。

**ターゲットサーバー** ： データレプリケーションが利用されているLifeKeeperクラスター内のスタンバイノード。複製されたパーティション（ターゲットパーティション）は、ソースパーティションにDataKeeperによる書き込みがあると更新されます。このパーティションへのアクセスや手動での修正を**してはいけません**。

## DataKeeper 関連用語

**レプリケーション** ： セクターごとのコピーにより、あるパーティションから別のパーティションにデータを転送すること。データの整合性を確保するためレプリケーション実行中は、ターゲットパーティションにアクセスしたり修正を加えたり**してはいけません**。

**非同期** ： データがソース上での書き込みのため直ちに解放され、それと同時にターゲットに送信され、ターゲットがデータを受信し書き込み可能になったらすぐに書き込みが行われるレプリケーション方式。

**変化率** ： 設定された期間内に変化するデータの量。

**圧縮** ： ソースノードとターゲットノードの間のトラフィック量を減らすため必要に応じ実装されるアルゴリズム。9段階の圧縮が可能です。デフォルトでは圧縮はオフになっています。

**スロットル** ： レプリケーションに使われる帯域幅を制限するため必要に応じて実装されるメカニズム。

## LifeKeeper 製品関連用語

**コミュニケーションパス** ： LifeKeeperクラスター内のノード間の通信に対応するメカニズム。単一障害点をなくすため、クラスター内のノード間に複数のコミュニケーションパスを設定することを強くお勧めします。

**ハートビート** ： LifeKeeperクラスター内のノード間で定期的に交換され、ノードのヘルスモニタリングを行うメッセージ。ハートビートメッセージは、コミュニケーションパス上で送信されるクラスターノード間通信の一種です。

**スプリットブレイン** ： クラスターノード間の全コミュニケーションパスに不具合が発生しているが、ノードは全て起動･稼働している状況。この状況ではいずれのシステムも他方に不具合が発生していると判断して、リソースを維持しサービスに投入し続けようとします。

**フェイルオーバー** ： アクティブノードのシステムあるいはリソースに不具合が発生したことによる、リソース階層のスタンバイノードへの予定外の移行。

**スイッチオーバー** ： アクティブノードからスタンバイノードへのリソース階層の計画的移行。

**スイッチバックタイプ** ： リソース稼働中に停止したノードのリカバリー動作を制御する設定。この設定が「intelligent」ならノードは特定のリソースのバックアップとして機能します。設定が「automatic」ならノードは通知なしで自動的かつ積極的にリソースを再取得しようとします。

**リソース** ： LifeKeeperにより保護されるシステムアセット。ディスクパーティション、仮想IPアドレス、アプリケーションなどを指します。

**リソースの拡張** ： 構成済みのLifeKeeperリソースをクラスター内の別のノード上に作成あるいは定義して、リソースが双方のシステムで同時に稼働するのを防ぐ等価関係を構築すること。

**リソース階層** ： リソースを、予め定められた順序に従って上から下に階層状にグループ化したもの。単に「階層」と言われることもあります。

**共有ストレージ** ： クラスター内の全ノードに物理的に接続された1つ又は複数の論理ディスクパーティション。LifeKeeperでは、共有ストレージボリュームには一度に1台のノードしかアクセスできないようになっています。これは正式にはI/Oフェンシングと呼ばれます。

**データレプリケーション（ディスクミラーリング）** ： データの可用性、流通性、正確性が継続的に保証されるように、論理ディスクパーティションを別の物理ハードディスクにリアルタイムでコピーすること。

**ソース** ： レプリケーションに使用されるソースサーバー上のパーティション。データの「ゴールド」コピー。

**ターゲット** ： レプリケーションに使用されるターゲットサーバー上のパーティション。

**切り替え可能IPアドレス** ： クラスター内システム間で切り替わる固有IPアドレス。クライアントシステムがこのアドレスに接続すると仮想IPリソースがアクティブになっているシステムがリクエストに応答します。

# 11.1.2. 評価プロセス

LifeKeeperの評価はテストラボ環境内で行うことを強くお勧めします。実稼働環境にインストールされた評価版ソフトウェアに対してSIOSは責任を負わず、サポートを提供することもありません。
評価期間中に何か疑問が生じた場合は、 evalsupport@us.sios.com 又は各地域のプリセールスエンジニアリング連絡先までご連絡下さい。プリセールスサポートが翌営業日までに連絡し疑問にお答えします。ソフトウェアメンテナンスのライセンスをお持ちのお客様には、販売後テクニカルサポートが年中無休で対応いたします。

**重要：** 評価ライセンスは、SIOS販売チームからのメールでSIOS製品評価版パッケージとライセンスを受け取った日から一定期間のみ有効です。

# 11.1.3. DataKeeper for Linux インストールの準備

## ハードウェアに関する要件

### プライマリーノード、セカンダリーノード

- 評価期間中 Linux からの配布物を利用するのに必要な最低条件を満たすシステムが必要です。
- 最低でも 512MB RAM 。1GB RAM が推奨されます。
- 利用可能なハードディスク容量が 2GB はあることが推奨されます。
- 複数のネットワークインターフェースカード (NIC) が推奨されます。
- データレプリケーション用に1つあるいは複数のディスクパーティションを構成して下さい。これはプライマリーノードではソースパーティション、セカンダリーノードではターゲットパーティションになります。この評価例では MySQL データのレプリケーションを行います（パーティションは `/var/lib/mysql` にマウントされます）。
- 複製対象のパーティションに関して、ターゲットパーティションのサイズは、ソースパーティションのサイズ以上でなければなりません。
- システムパーティション( / )及び起動パーティション( `/boot` )はレプリケーションの対象外です。

## ソフトウェアに関する要件

### プライマリーノード、セカンダリーノード

- Linux ディストリビューション x86_64、 AMD 64
  サポート対象の運用システムリストは、サポートマトリックス にてご覧下さい。
- 現在お使いのパッチ、セキュリティーについては最新へのアップデートをお勧めします。
- 依存関係に関する要件：特にベーシックあるいは最小限の Linux インストールパッケージを選択された場合には、Linux の依存関係 にて依存関係関連ドキュメントをご参照下さい。
- ここでは、ファイアーウォールを無効化します。ファイアーウォールを有効にする場合は、ファイアーウォールを使用した状態での LifeKeeper の実行 を参照してください。
  - `systemctl stop firewalld.service; systemctl disable firewalld.service`（firewalld サービスをサポートしている場合）
  - `systemctl stop SuSEfirewall2.service; systemctl disable SuSEfirewall2.service`（SuSEfirewall2 サービスをサポートしている場合）
  - `systemctl stop iptables.service; systemctl disable iptables.service`（iptables サービスをサポートしている場合）

    - `systemctl stop nftables.service; systemctl disable nftables.service`（nftables サービスをサポートしている場合）

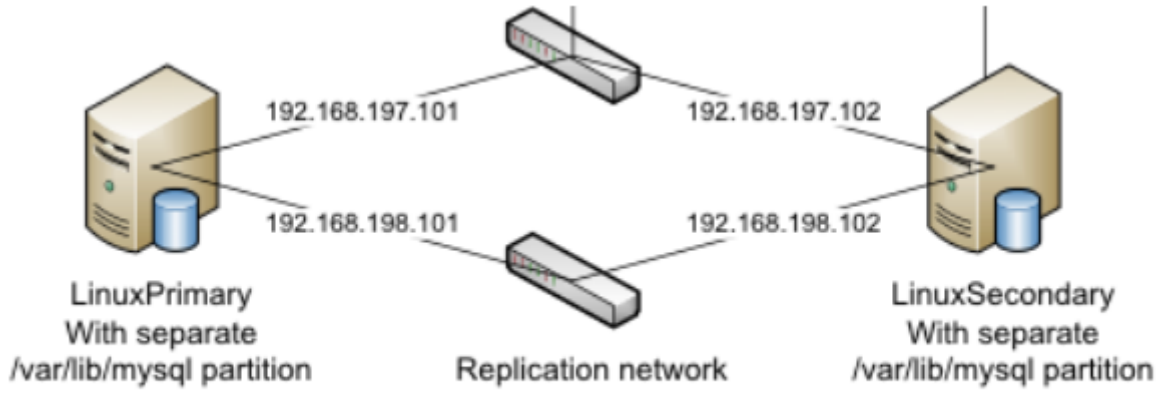
- SELinux を無効化して下さい。無効化方法は SELinux を無効にする を参照してください。

- `/etc/hosts` ファイルの構成をご確認下さい。
    - 127.0.0.1 においてエントリーできるのは localhost.localdomain 及び localhost のみです。
    - ホスト名に関しては、静的アドレスで個別のエントリーを行って下さい。

- PAM による GUI 認証
    - LifeKeeper for Linux では Linux Standard Base (LSB) で提供されている PAM (Pluggable Authentication Module) が利用できるようになりました。
    - ユーザーの識別、認証はシステムの PAM 構成に対して行われます。権限レベルは PAM を通じて付与されるグループメンバーシップによって決まります。
    - ユーザーが GUI にアクセスするには、LifeKeeper の3つのグループ（lkadmin、lkoper、lkguest）のいずれかのメンバーであることが必要です。
    - 詳細については GUI ユーザーの設定 をご覧下さい。

## ネットワークに関する要件

評価を行うには次の例のように機器を構成することをお勧めします。




**Network Configuration Example**

LinuxPrimary、LinuxSecondary は、2つのLANセグメント間でマルチホーミングされています（2つ目の物理ネットワークが利用できない場合でも、クロスオーバーケーブルによって各ノードの2つ目の NIC を接続できます）。この構成では2つ目の NIC はオプションですが、単一障害点をなくすため実稼働環境では2つ目の NIC を使うことを強くお勧めします。
本例の評価シナリオでは、直接接続されたストレージのレプリケーションを行っていきます。

### 重要：変化率の分析

リアルタイムでデータのレプリケーションを行う場合、常時ミラーリング状態でレプリケーションを行うのに十分な帯域幅を確保する必要があります。書き込みアクティビティを時間対帯域幅で収集･分析して変化率分析を行う場合は、Linux システム（物理または仮想）での変化率の測定） をご参照下さい。

## プライマリーノード、セカンダリーノード

- LifeKeeper で保護される全ノードについて、エントリーを伴うホストファイルを構成します。これは通常 /etc/hosts です。
  例:

```
192.168.197.101 LinuxPrimary
192.168.197.102 LinuxSecondary
```

- 切り替え可能 IP アドレスとして使用する未使用 IP アドレスを取得する場合にはネットワーク管理者にお問い合わせ下さい。この切り替え可能 IP アドレスは、構成の過程で後ほど作成します。

- 以下を使ってパブリックネットワーク接続を構成します。
  - 静的 IP アドレス
  - 適切なサブネットマスク
  - 適切なゲートウェイアドレス

- 以下を使ってプライベートネットワーク接続を構成します。
  - 静的 IP アドレス (パブリックネットワークとは別のサブネット上のもの)
  - 適切なサブネットマスク
  - ゲートウェイ IP アドレスは使いません。
  - DNS サーバーアドレスは使いません。

# 11.1.4. DataKeeper for Linux ストレージの構成

> ! レプリケーションには、デバイスに GUID が必要です。

## 始める前に

次の点をご確認下さい。

- 両ノード上にデータレプリケーションに利用できるディスク / パーティションが別途あること。ターゲットのディスク / パーティションのサイズは、ソースのディスク / パーティションのサイズ以上でなければなりません。

## DataKeeper for Linux で使用するローカルストレージのパーティション分割

### プライマリーノード

プライマリーノード上で以下の操作を行って下さい。

1. ミラーのソースとして利用する未使用の空きディスクパーティションがあることを確認して下さい。あるいは新しいパーティションを作っても構いません。ディスクを適切にパーティション分割するには「gdisk」を使います。本例では未使用のディスク /dev/sdb に単一のパーティションを作成します。
   a. `gdisk /dev/sdb`
   b. 「n」を押して新しいパーティションを作成します。
   c. この例では新しいディスクを使用するので、全てデフォルト値を使います（パーティション番号は 1、ディスク全体を使用、パーティションのタイプは「Linux filesystem」）。Enter キーを4回押してこれらのパラメーターを確認して下さい。
   d. 「w」を押してパーティションテーブルを書きます。
   e. 「Y」を押して既存のパーティションに上書きできていることを確認します。

   **例：**

   [root@LinuxPrimary ~] # gdisk /dev/sdb
   GPT fdisk (gdisk) version 0.8.10
   Partition table scan：
   MBR：protective
   BSD：not present
   APM：not present
   GPT：present
   Found valid GPT with protective MBR; using GPT.
   Command (? for help): **n**

Partition number (1-128, default 1): **<enter>**
First sector (34-2047, default = 34) or {+-}size{KMGTP}: **<enter>**
Last sector (34-2047, default = 2047) or {+-}size{KMGTP}: **<enter>**
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300): **<enter>**
Changed type of partition to 'Linux filesystem'
Command (? for help): **w**
Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING PARTITIONS!!
Do you want to proceed? (Y/N): **Y**
OK; writing new GUID partition table (GPT) to /dev/sdb.
Warning: The kernel is still using the old partition table.
The new table will be used at the next reboot.
The operation has completed successfully.
[root@LinuxPrimary ~] #

2. 新しく作成したディスクパーティションをフォーマットして下さい。
   `# mkfs.ext3 /dev/sdb1`

3. 所望の場所（例えば/var/lib/mysql）にパーティションをマウントして下さい。
   `# mount /dev/sdb1 /var/lib/mysql`

4. 注記： `/etc/fstab` にこのパーティションを追加する必要はありません。Lifekeeper が自動的にマウントを行います。

**結果**

```
[root@LinuxPrimary ~] # df /var/lib/mysql
Filesystem 1K-blocks Used Available Use% Mounted on
/dev/sdb1 253855 11083 229666 5% /var/lib/mysql
```

## セカンダリーノード

5. セカンダリーノードでは、上記の手順1（ディスクのパーティション分割）のみを行って下さい。ターゲットのディスク / パーティションのサイズはソースのディスク / パーティションのサイズ以上でなければなりません。ファイルシステムをフォーマットあるいはマウントする必要はありません。

# 11.1.5. LifeKeeperのインストール

簡単にインストールが行えるよう、LifeKeeper にはインストールスクリプトが付いています。スクリプトの最後で所望のApplication Recovery Kit (ARK)を選択する必要があります。ソフトウェアは以下の場所にインストールされます。

| LifeKeeper for Linux Component | Install Location |
|---|---|
| LifeKeeper Software | `/opt/LifeKeeper` |
| LifeKeeper Config File | `/etc/default/LifeKeeper` |

プライマリーノード、セカンダリーノードの**双方**で以下の操作を行って下さい。

## ソフトウェアのダウンロード

1. SIOSから届いたLifeKeeper評価版のメールを開きます。

2. メールに記載されたリンクからLifeKeeperをダウンロードします。一般的には「wget」を使って全ファイルを再帰的にダウンロードするのが最も簡単です。
   a. `# cd /root`
   b. `# wget -r <URL>`
   c. 正常に終了した場合は、以下のディレクトリーリストに類似した内容がダウンロードされているはずです。

```
[root@LinuxPrimary ~] # ls -l <directory> total 63680
-rw-r—r— 1 root root 23163 May 30 14:03 EULA.pdf
-rw-r—r— 1 root root 536 May 30 14:03 readme.txt
-rw-r—r— 1 root root 65179648 May 30 14:03 sps.img
```

   d. 評価用メールに記載されているリンクから評価ライセンスキーをダウンロードして下さい。双方のノードの覚えやすい場所にライセンスファイルを保存して下さい。

## LifeKeeperインストールスクリプトの実行

1. 先にダウンロードしたsps.imgファイル（ISO9660イメージファイル）をループバックマウントし、その中の「setup」スクリプトを実行して下さい。

```
# mount -o loop sps.img /mnt
# cd /mnt
# ./setup
```

2. このプロセスでは、ほとんどの場合Enterキーを押してデフォルト値を受け入れ、次の画面に進むことになります。ただし次の例外がありますのでご注意下さい。
   a. この評価ガイドでは高可用性NFSサーバークラスターは構成しないので「High

AvailabilityNFS（高可用性NFS）」というタイトルの付いた画面では「n」を選択して下さい。

b. 高可用性NFSサービスを作成する場合にはそれに合わせて対応を調整して下さい。

3. 「setup」スクリプトの最後で、試用ライセンスキーをすぐにインストールするか後でインストールするかを選べます。次のステップでライセンスキーをインストールするので、ここでは「n」を選択して構いません。

4. 「setup」の最後の画面で、表示されたリストからDataKeeperを選択して下さい。

5. 以下のRPMをインストールする必要があります。
`steeleye-lkDR-<version>.noarch.rpm`

6. ディストリビューション対応のディスクイメージをアンマウントして下さい。

```
# cd /root
# umount /mnt
```

## 評価ライセンスキーのインストール

設定プロセスの最後のフェーズはライセンスキーのインストールです。LifeKeeperを起動する前に、評価版ソフトウェアとともにダウンロードしたライセンスキーファイル（「.lic」）をインストールする必要があります。

1. 例：試用ライセンスキーをインストールするにはプライマリーノードとセカンダリーノードの双方で「lkkeyins」コマンドを実行して下さい。このコマンドは `/opt/LifeKeeper/bin/lkkeyins` にあります。
`# /opt/LifeKeeper/bin/lkkeyins <path_to_license/<filename>.lic`

2. コマンド/opt/LifeKeeper/bin/lklicmgrによってライセンスキーがインストールされたことを確認して下さい。

```
# /opt/LifeKeeper/bin/lklicmgr
License File: 20220221.lic
Product                          Type    Expiry                    Other
LifeKeeper for Linux             Eval    17 May 2022 (84 days)
SIOS Data Replication ARK        Eval    17 May 2022 (84 days)
```

## LifeKeeperの起動

1. 以下のコマンドでLifeKeeperを起動します。
`# /opt/LifeKeeper/bin/lkstart`

# 11.1.6. DataKeeper for Linuxクラスターの構成

## プライマリーノード

プライマリーノードで次のステップを実行してクラスターを構成して下さい。

- TCPコミュニケーションパスを作成する。
- コミュニケーションパスを検証する。

ハートビートの冗長化のために、開始前にクラスター内の各ノード間に少なくとも2つのTCPコミュニケーションパスを作成しておくことを強くお勧めします。

> **注意:** クラウド環境 では一般にネットワークの内部構成が公開されていないので、物理的に別経路の LAN 回線を 2 つ用意することは困難です。しかしながら、基本的にクラウド側で物理的なネットワークの冗長化が行われている事が予想されるため、 コミュニケーションパスが一つ であっても動作上の信頼性を確保することができます。

## LifeKeeper GUIへのアクセス

LifeKeeper GUIはJavaベースのアプリケーションで、Linuxのネイティブアプリケーションとして実行することができます。
LifeKeeper GUIは、コールバックに対応するJava RMIに基づいています。ホスト名は解決可能なものでなければなりません。そうでない場合、Java 115 又はJava 116エラーが表示されます。

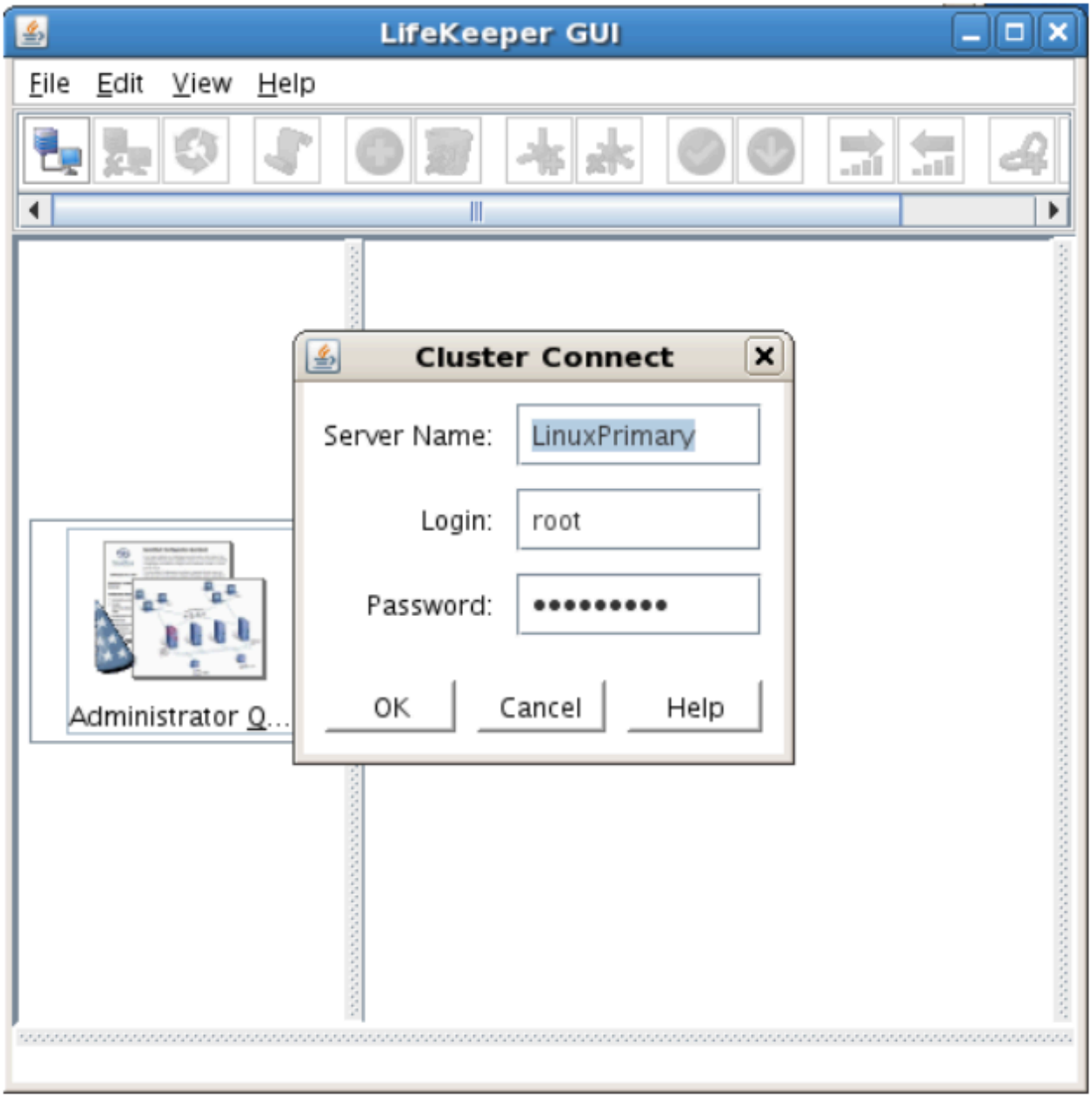
1. 全クラスターノードの短縮ホスト名と完全修飾ホスト名の双方が適切な場所で解決できることを確認して下さい。

   ```
   # ping LinuxPrimary
   # ping LinuxPrimary.domain.com
   # ping LinuxSecondary
   # ping LinuxSecondary.domain.com
   ```

2. 以下によりLifeKeeper Linux GUI Applicationを起動します。
   `/opt/LifeKeeper/bin/lkGUIapp &`

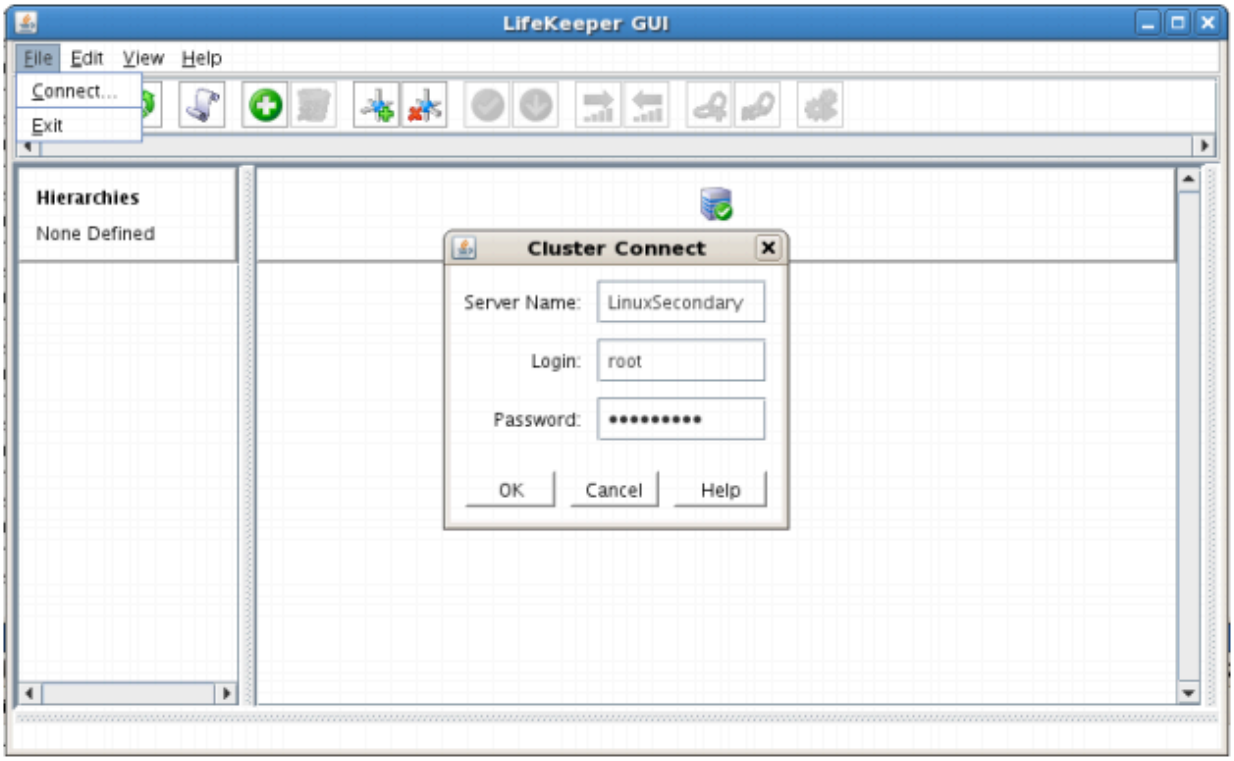
3. ウェブブラウザーからLifeKeeper GUI Appletに接続する場合は、次のURLにアクセスして下さい。
   `http://<hostname>:81`

4. ルート認証情報及び接続しようとするノードの名称を入力し（LifeKeeperがインストールされたノードからGUIを実行している場合、このフィールドには利用中のノードの名前が入ります）、OKをクリックします。

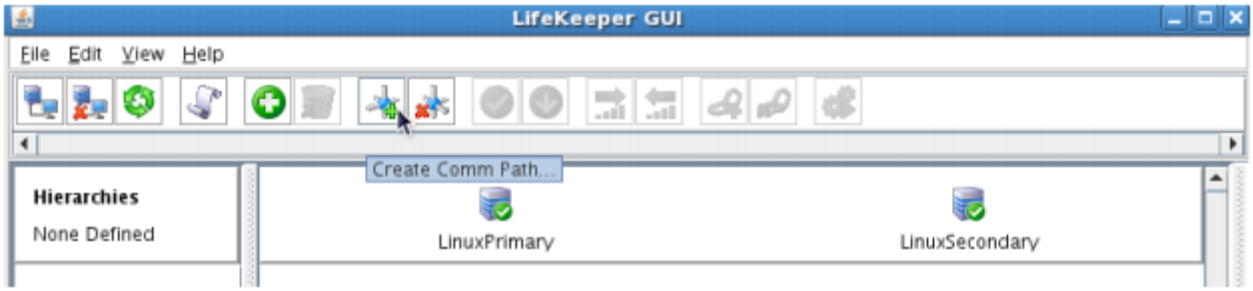



## コミュニケーションパスの作成

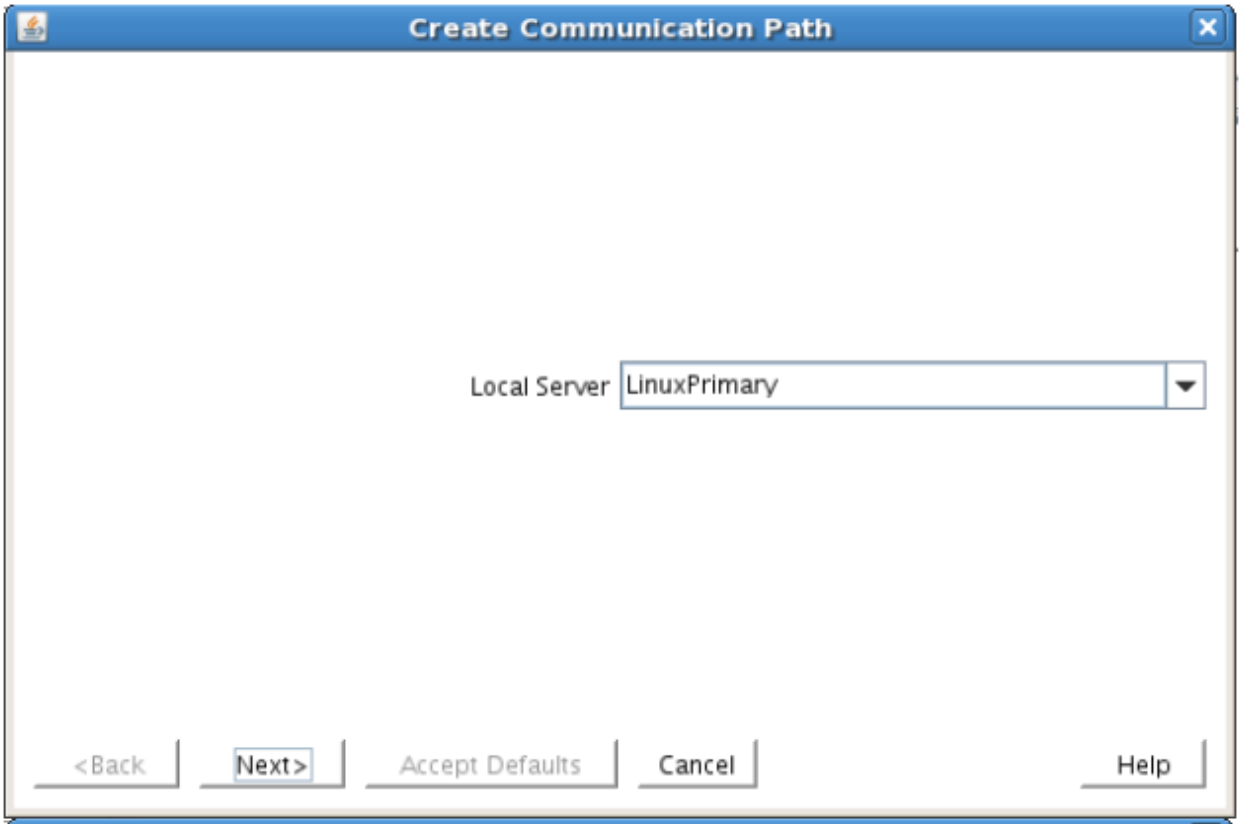
5. LifeKeeper GUIのファイルメニューから「Connect」を選択して下さい。「Cluster Connect」ウインドウが表示されたらセカンダリーノード名、ログイン名、パスワードを入力して下さい。

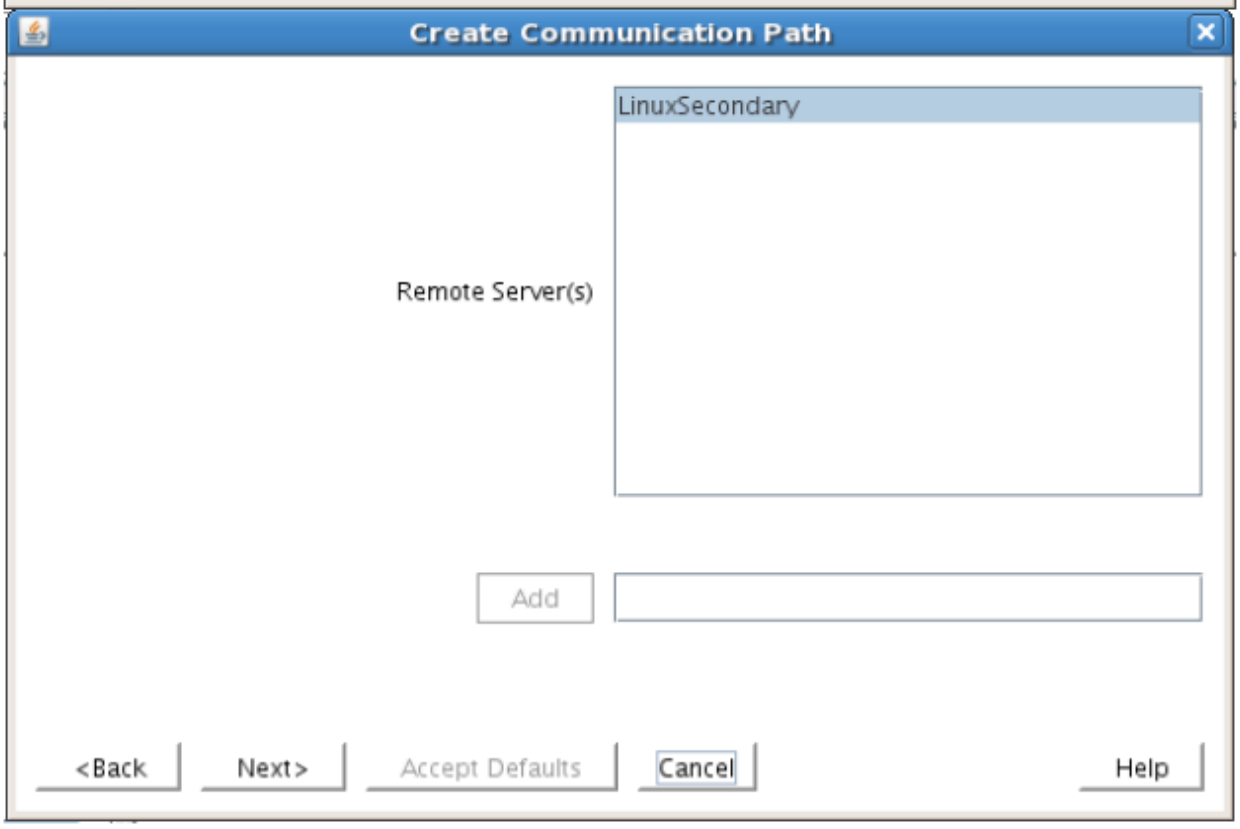



6. LifeKeeper GUI画面のツールバーの「Create Comm Path」ボタンをクリックして下さい。またはいずれかのノードのアイコンを右クリックしてポップアップメニューの「Create Comm Path」をクリックしてもOKです。

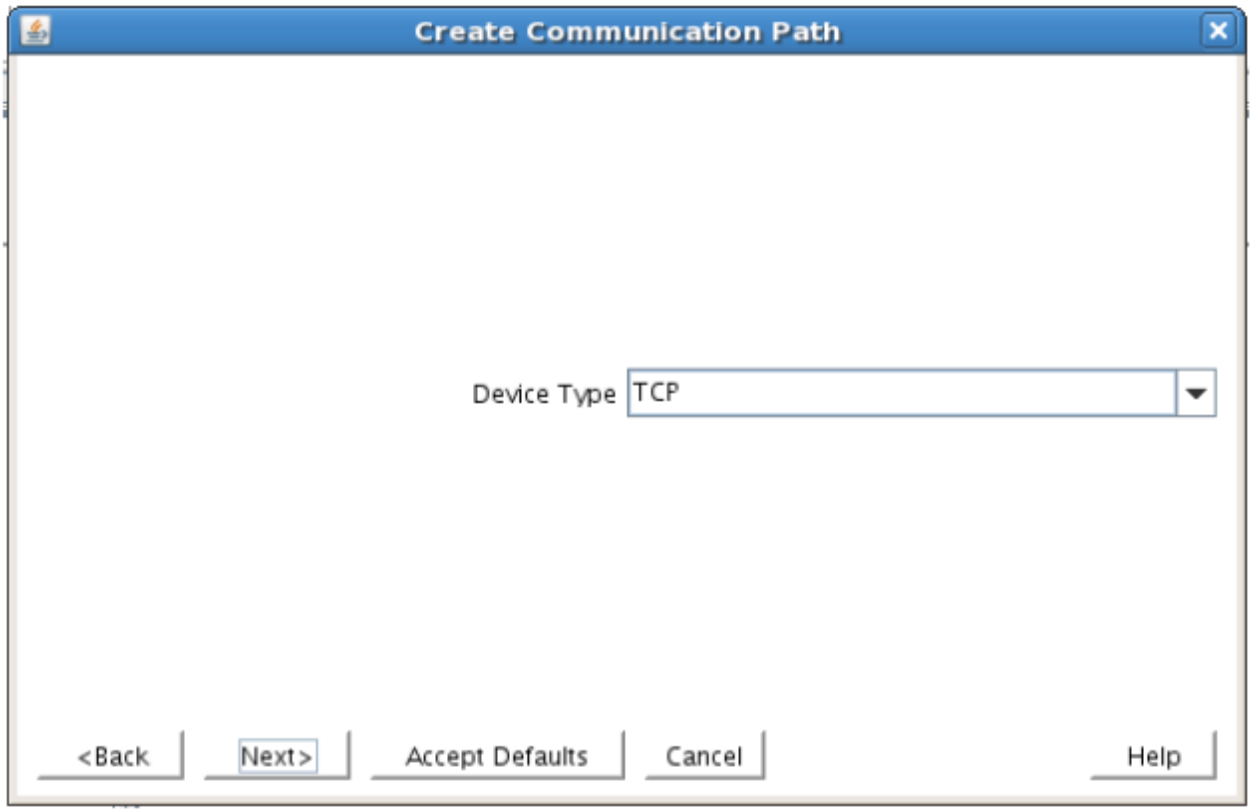



7. リストボックス「Local Server」「Remote Server」から適切なノード名を選んで下さい。リストボックスにノード名が表示されない場合は手動でノード名を入力して「Add Server」ボタンをクリックします。「Add Server」ボタンでノードを追加する場合には、 `ping -a IP_ADDRESS` などを使用してすべてのクラスターノードのコンピューター名が正しく応答することを確認して下さい。応答しない場合には、修正を加えてから次に進みます。「Next」をクリックして下さい。

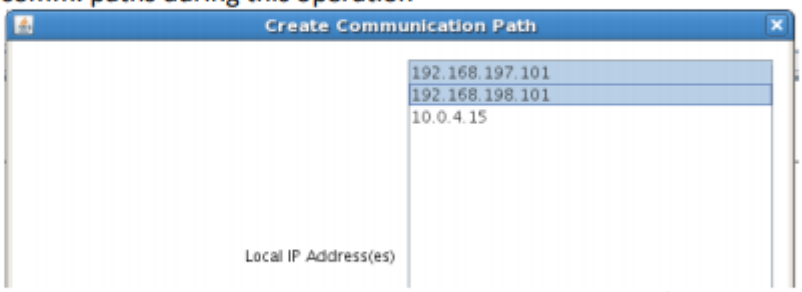


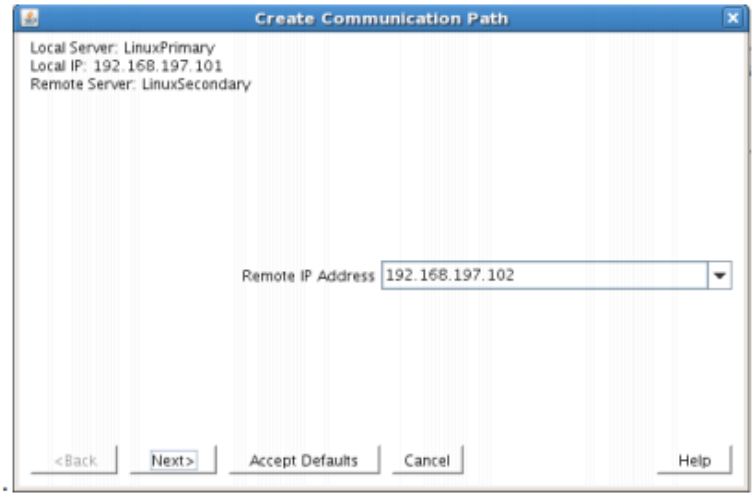




8. 「Device Type」はTCPを選択して「Next」をクリックします。



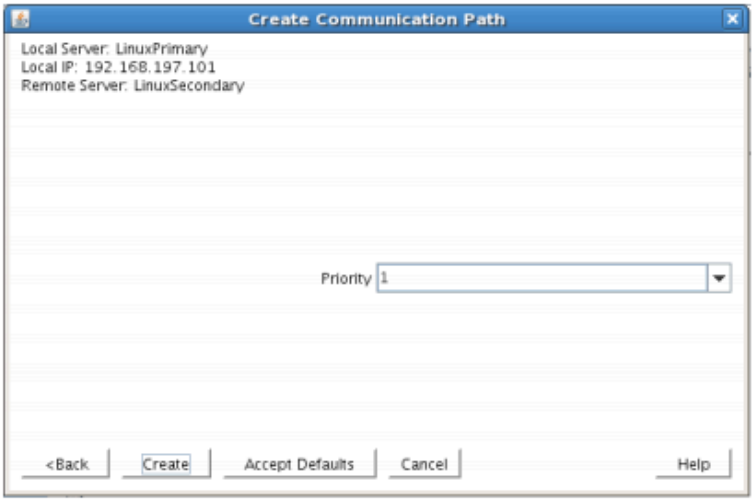

9. 続く一連のダイアログボックスに必要な情報を全て入力して「Next」をクリックします。ダイアログボックスの各フィールドについて、説明や提案が欲しい場合には「Help」をクリックするか、下の表を参照して下さい。

| Field | Tips |
| --- | --- |
| ***For TCP/IP Comm Path...*** | |
| Local IP Address | Choose the IP address to be used by the local server for this comm path. Select both interfaces so that the wizard creates multiple comm. paths during this operation  |
| Remote IP Address | Choose the IP address to be used by the remote server for this comm path |

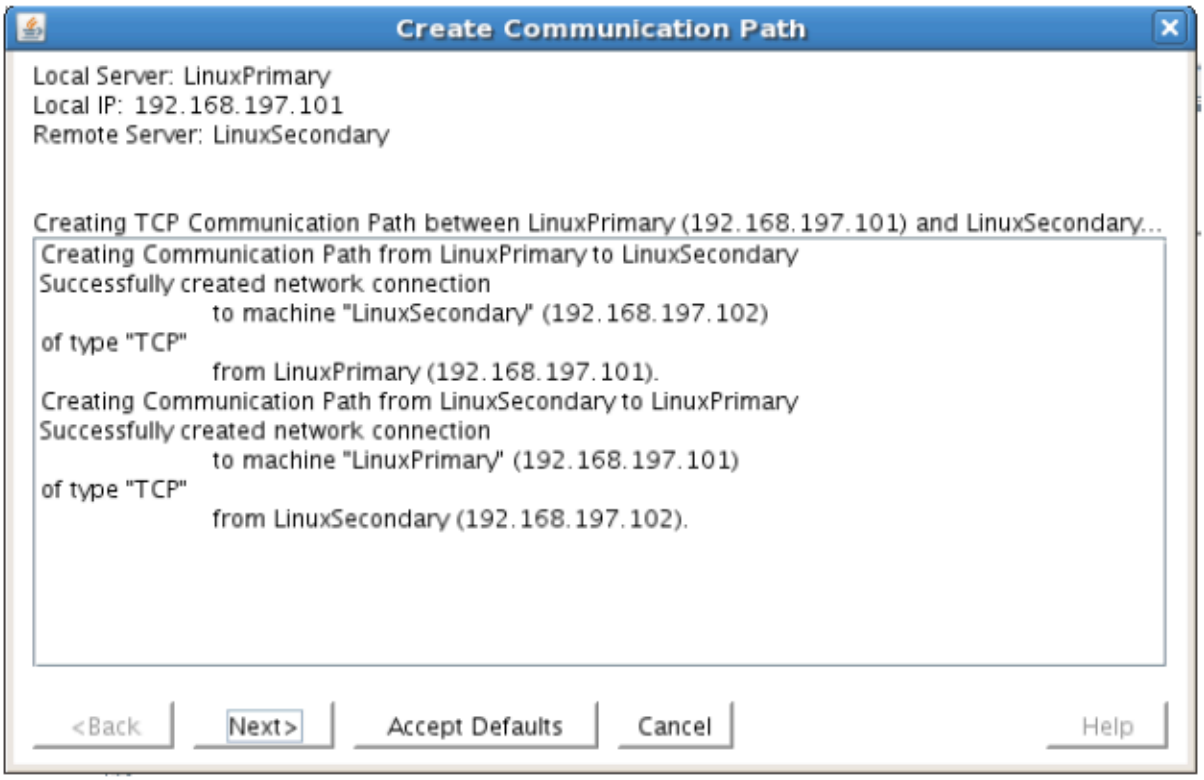



Enter the priority for the comm path on the local server. The priority will be used to determine the order that the comm paths between two servers will be used. Priority 1 is the highest; 99 is the lowest.

Priority




10. 全てのフィールドに必要な情報を入力したら、「Create」を選択して下さい。ネットワークコミュニケーションパスが正常に作成されたことを示すメッセージが表示されます。「Next」をクリックします。



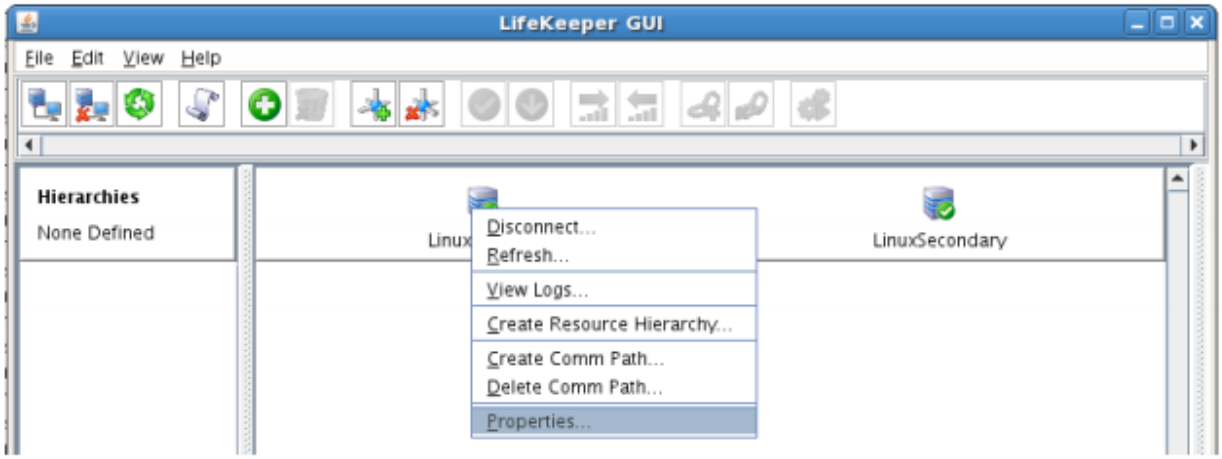

「Device Type」がTCPに設定されており、複数のローカルIPアドレスあるいは複数のリモートノードを選択した場合には、セットアップウィザードに戻って引き続き次のコミュニケーションパスを作成して下さい。

11. 最後のダイアログボックスで「Done」を選択して下さい。

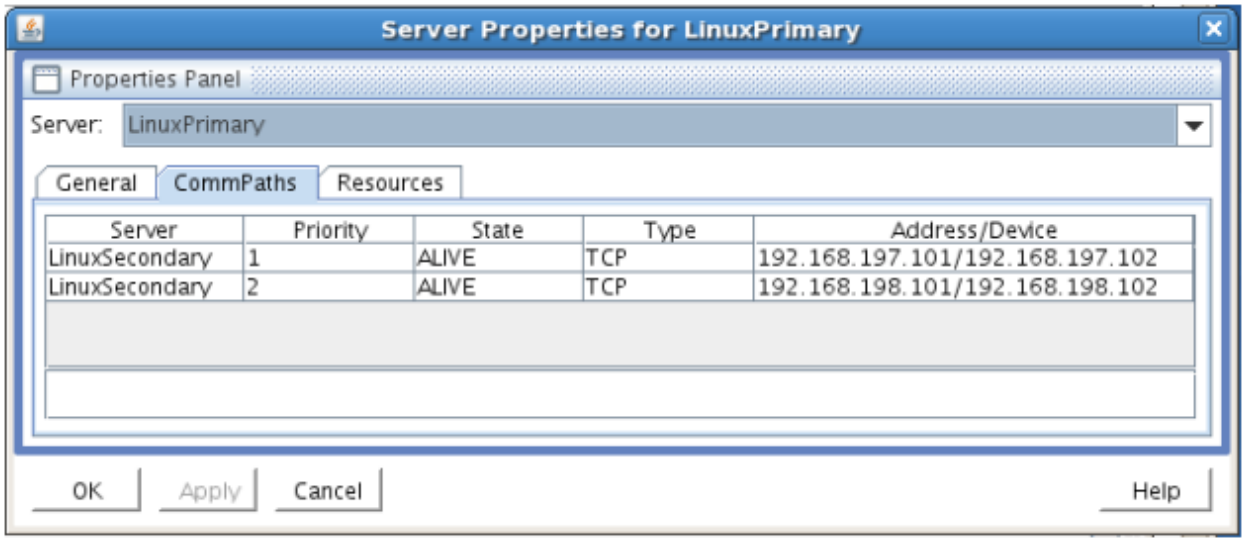
    使用を予定している全てのコミュニケーションパスが定義されるまでこの手順を繰り返して下さい。冗長コミュニケーションパスを作るため、少なくとも2つのコミュニケーションパスを定義することを強くお勧めします。

## コミュニケーションパスの検証

1. ダイアログボックス「Server Properties」を見てコミュニケーションパスが適切に構成されていることを確認します。LifeKeeper GUIから「Edit」、「Server」、「Properties」を選択し、次に「Comm Paths」タブを選択します。




2. 表示される「State」が「ALIVE」であることを確認して下さい。GUIのメインパネル右側のノードのアイコンで確認することもできます。コミュニケーションパスが1つしか作られていない場合、そのノードのアイコンは黄色の警告アイコンになります。警告アイコンは1つのコミュニケーションパスが「ALIVE」であるが、冗長コミュニケーションパスはないことを示しています。少なくとも2つのコミュニケーションパスが作成され、その状態がALIVEであればノードアイコンには緑のハートビートのチェックマークが表示されます。




# LifeKeeper階層の作成

## ミラーの作成とデータレプリケーションの開始

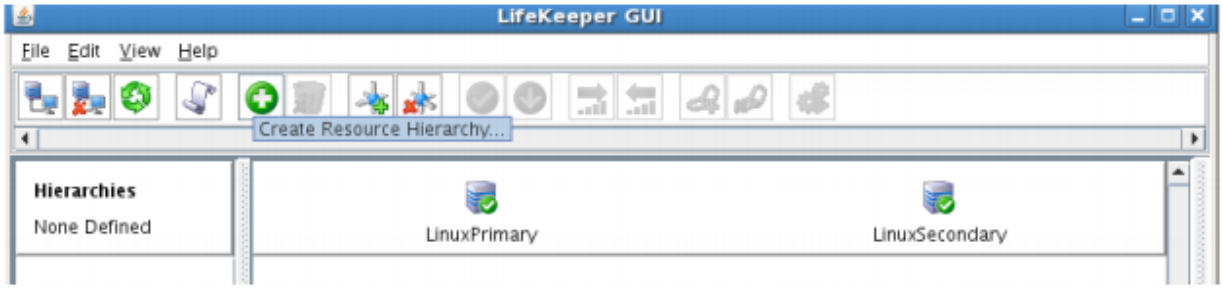
本セクションでは、クラスターノード間でMySQLデータを同期させる際に使われるデータレプリケーションリソースの設定と構成を行います。複製するデータはプライマリークラスターノードのパーティション `/var/lib/mysql` にあります。

以下の点にご注意下さい。

- 複製対象のソースボリュームをプライマリーノードにマウントしなければなりません。

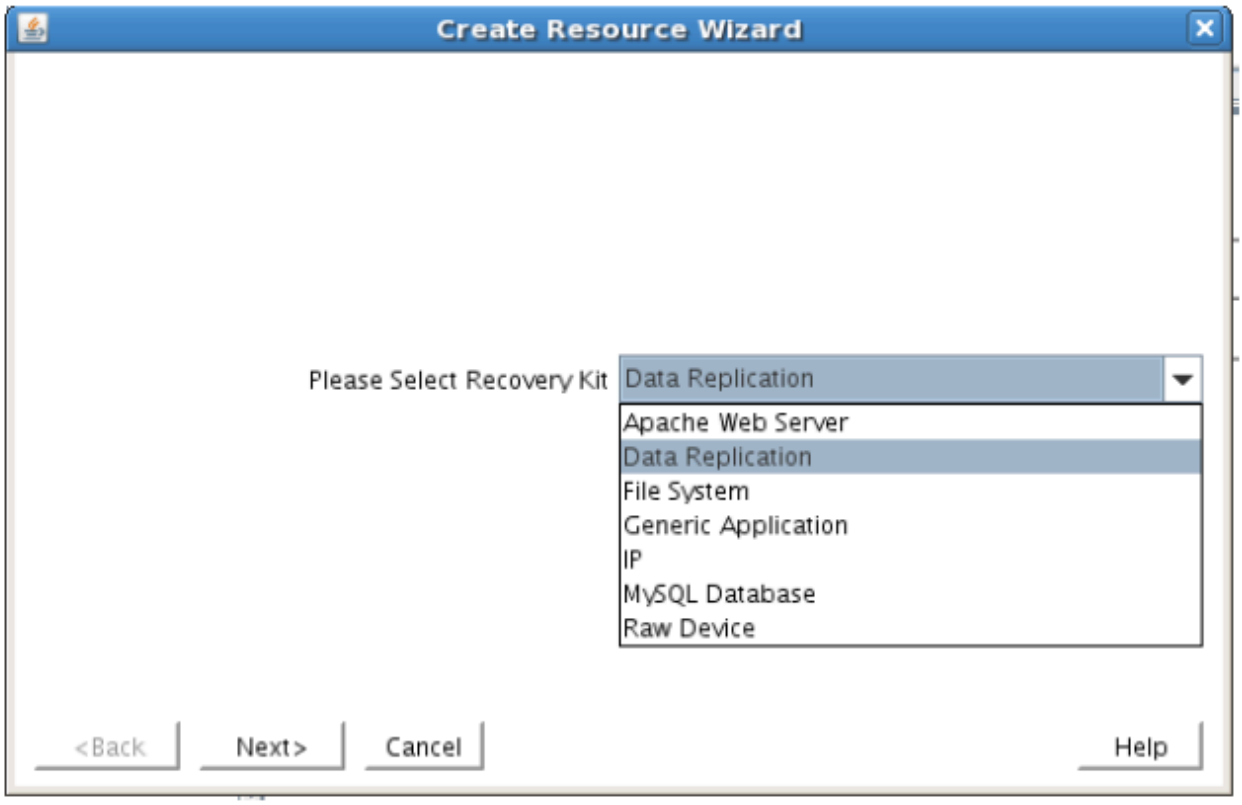
- 複製されたデータを受け取るターゲットボリュームをセカンダリーノードにマウントしてはなりません。
- ターゲットボリュームのサイズは、ソースボリュームのサイズ以上でなければなりません。

1. LifeKeeper GUI画面のツールバーの「Create Resource Hierarchy」をクリックします。




ダイアログボックス「Create Resource Wizard」が表示されます。その画面のドロップダウンリストボックスには、クラスター内にインストールされ認識された全てのRecovery Kitが表示されます。

2. 「Data Replication」を選択し、「Next」をクリックします。

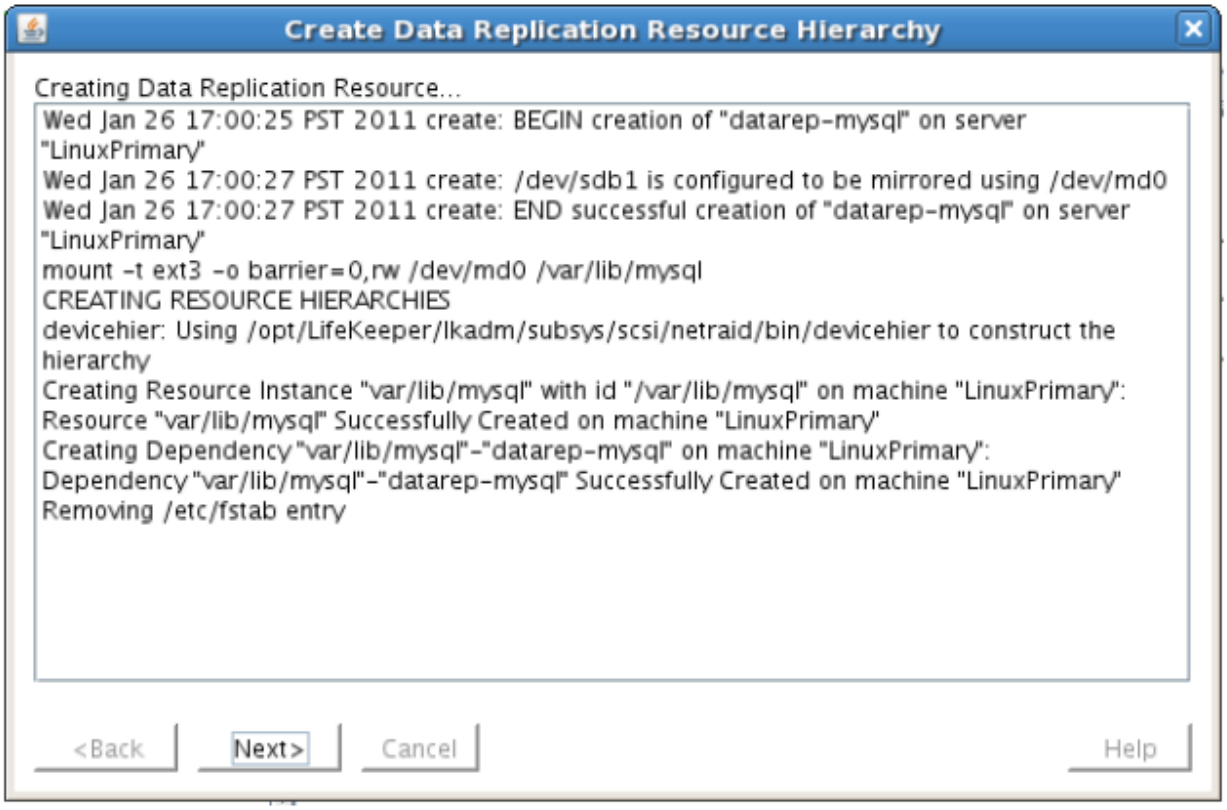



3. 「Data Replication」ウィザードに従って以下のように入力して下さい。

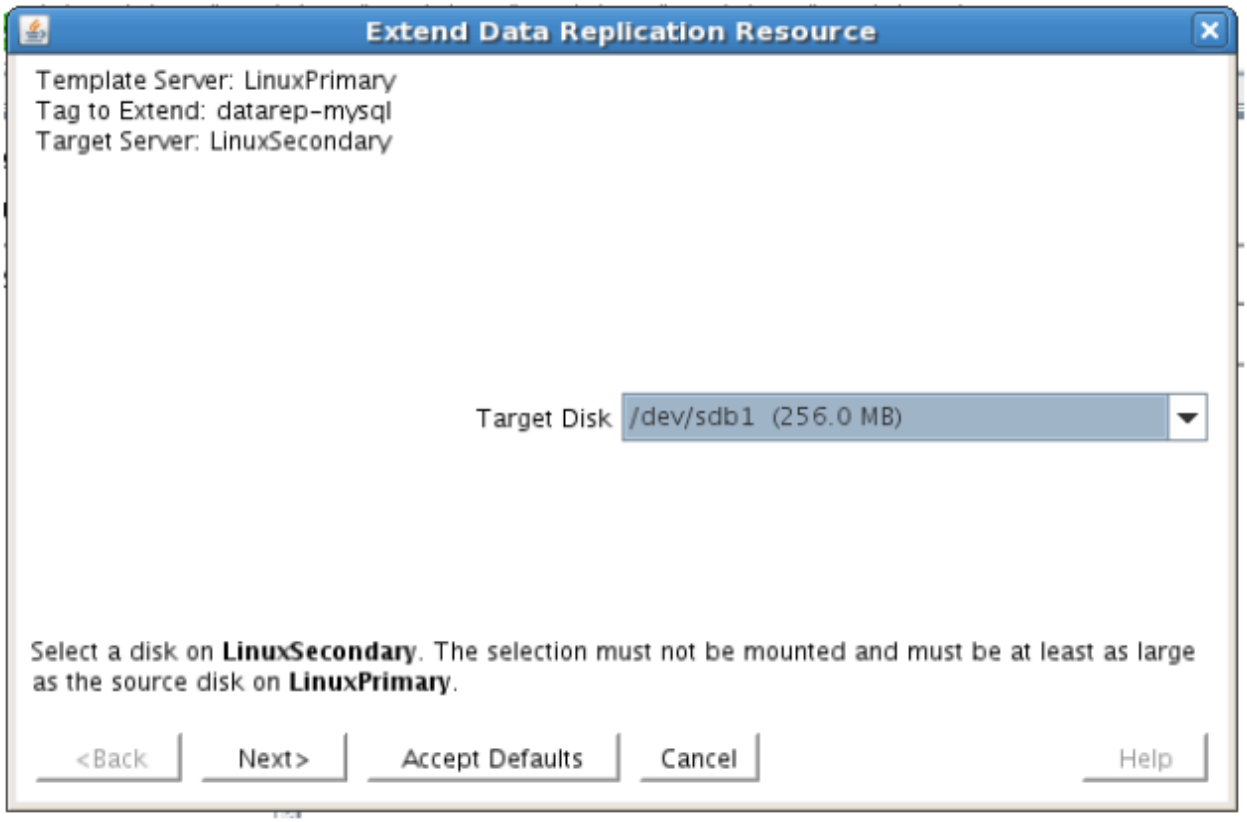
| フィールド | 推奨される入力又は注記の内容 |
|---|---|
| Switchback Type | Intelligent |

| | |
|---|---|
| Server | LinuxPrimary (プライマリークラスターノード即ちミラーソース) |
| Hierarchy Type | 選択：「 **Replicate Existing Filesystem** 」 |
| Existing Mount Point | ここではマウントされた複製対象のパーティションを選択します。この例では「/var/lib/mysql」を選択して下さい。 |
| Data Replication Resource Tag | デフォルトのまま |
| File System Resource Tag | デフォルトのまま |
| Bitmap File | デフォルトのまま （注記：高速SSDストレージを使用する場合は小さなパーティション即ち/bitmapsを作成し、それを利用してビットマップを配置することもできます） |
| Enable Asynchronous Replication | デフォルト（「Yes」）のまま |

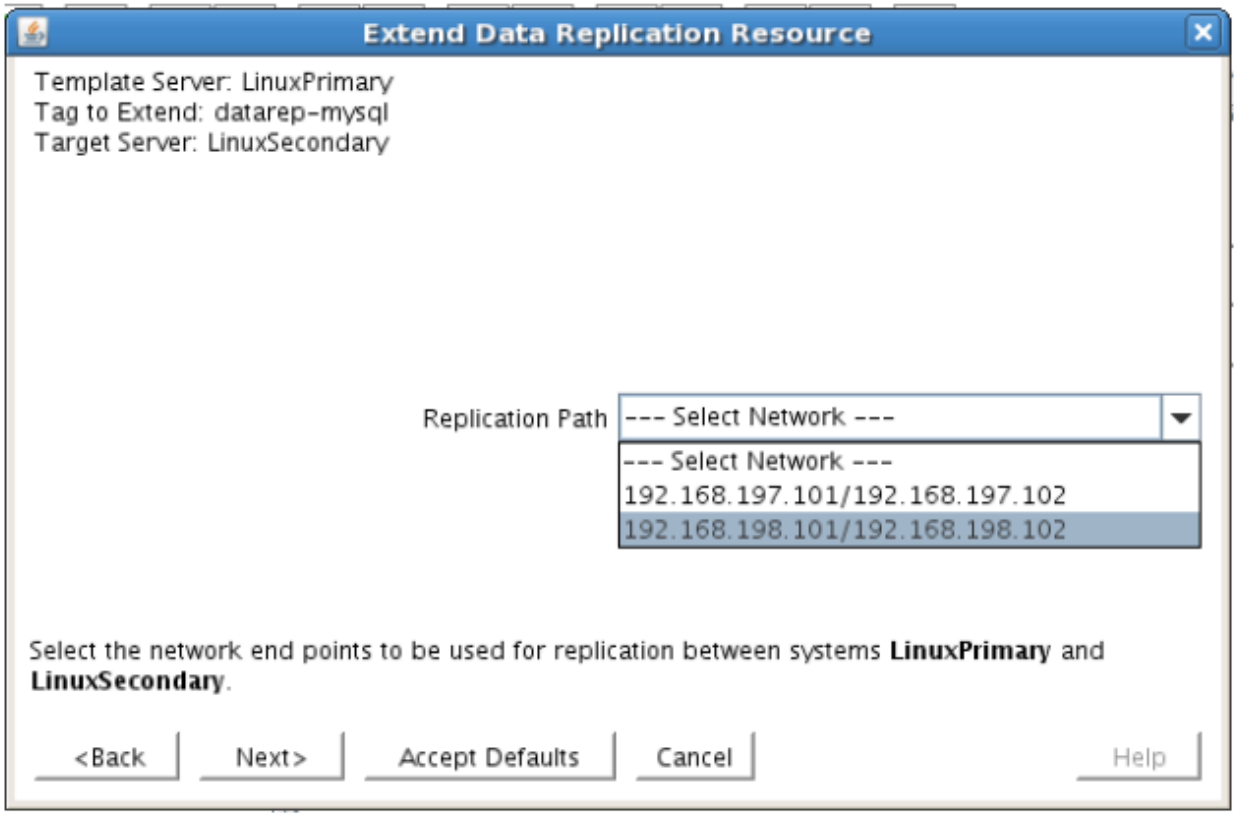
4. 「Next」をクリックしてデータレプリケーションリソース階層の作成を開始します。次のようにGUIにステータスが表示されます。



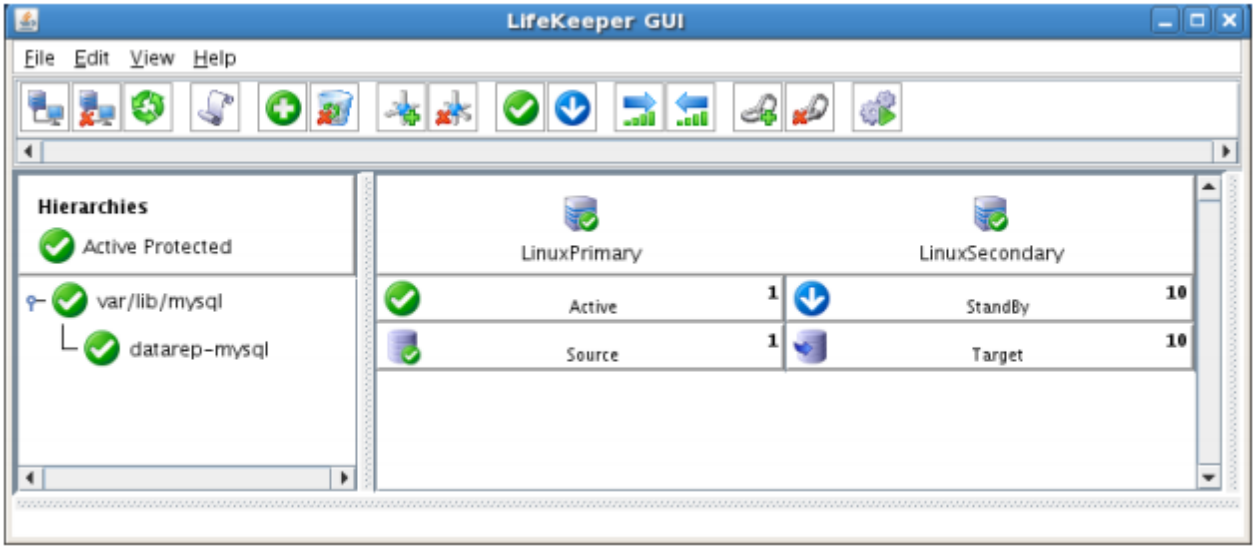

5. 「Next」をクリックしてデータレプリケーションリソースの拡張を始めます。ここでは全てデフォルト設定を選択して下さい。ターゲットディスクが要求された場合には、複製対象のソースボリュームと同じ（又はそれより大きい）サイズの空なパーティションをターゲットサーバー上で選択します。このパーティションはターゲットシステムにマウントしてはなりません。




6. ウィザードを続行すると、レプリケーションを行うネットワークの選択が求められます。一般的には、ユーザー／アプリケーションとレプリケーショントラフィックは分けることをお勧めします。この設定例ではバックエンドネットワーク
192.168.198.X でレプリケーションを行います。



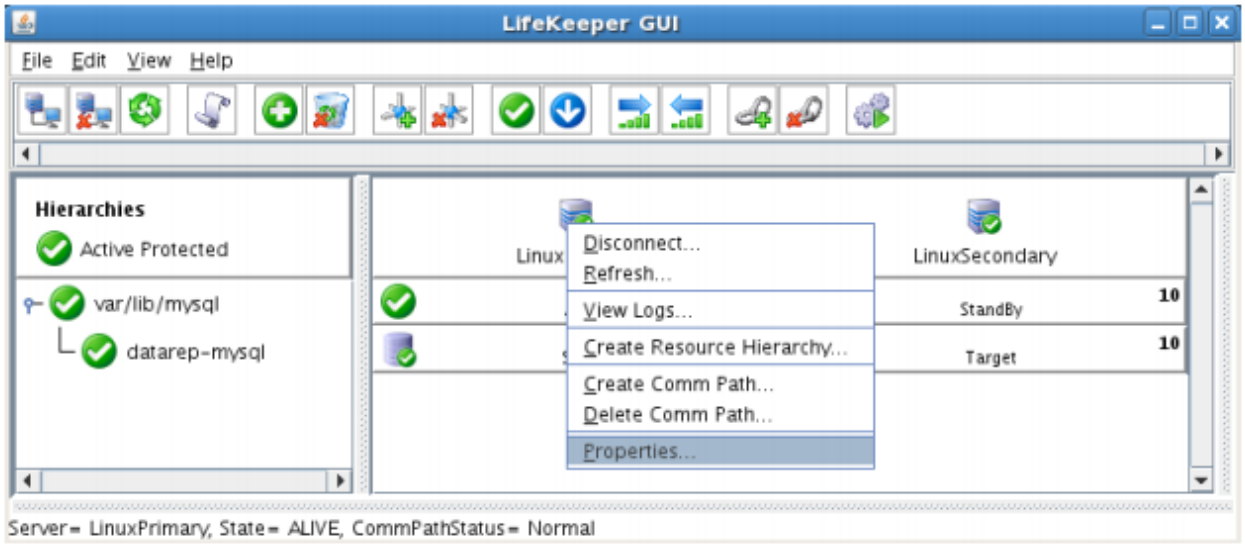
7. 「Next」をクリックしてウィザードを続行します。完了したら次のようにリソース階層が表示されます。




## 自動フェイルオーバーの無効化

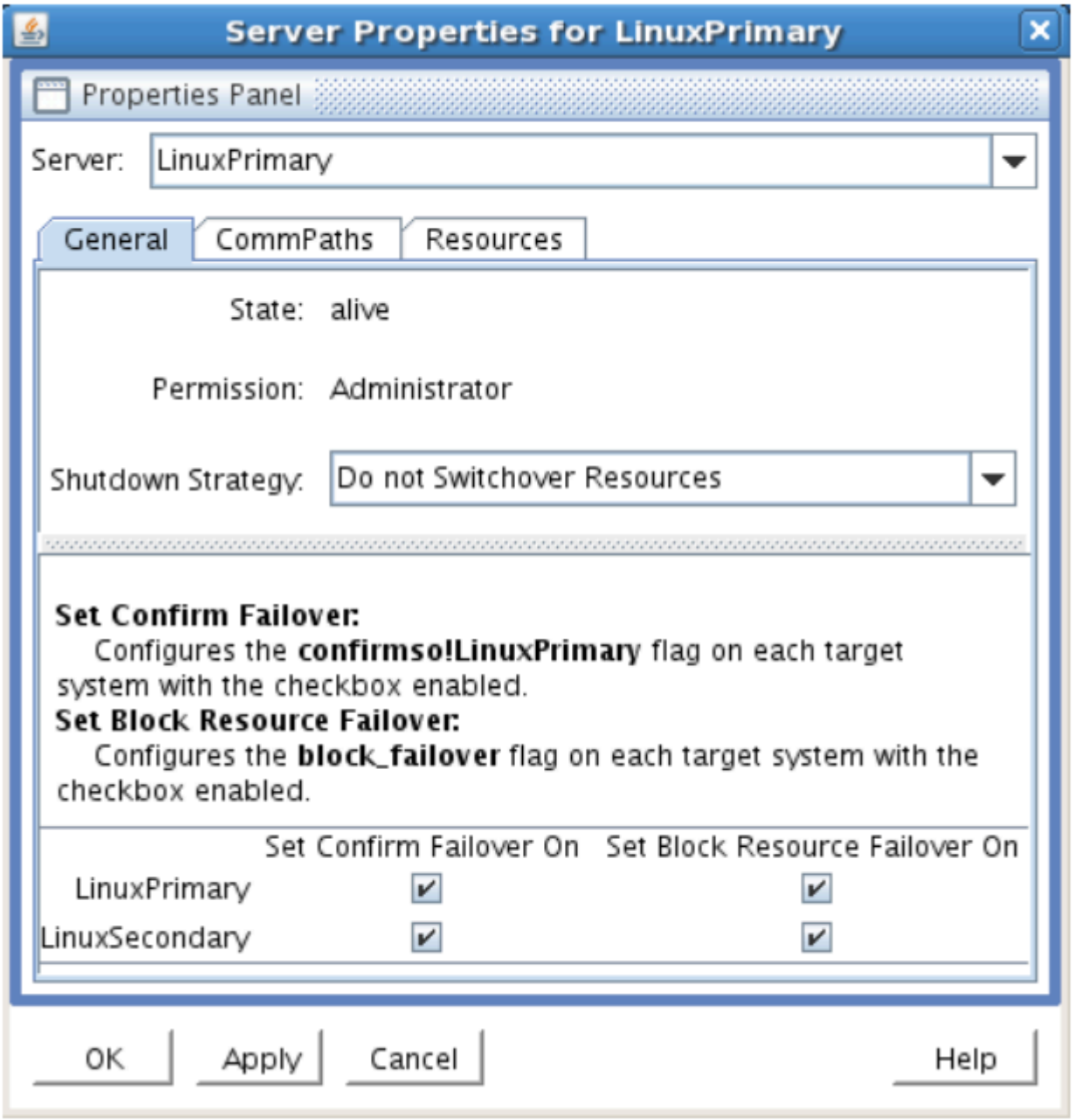
このセクションではスタンバイノードへの自動フェイルオーバーを無効化する操作を見ていきます。

1. LifeKeeper GUIのクラスターノードの一つを右クリックし「Properties」を選択します。




2. 画面の一番上にある「Server」のドロップダウンリストからソースサーバーを選択します。

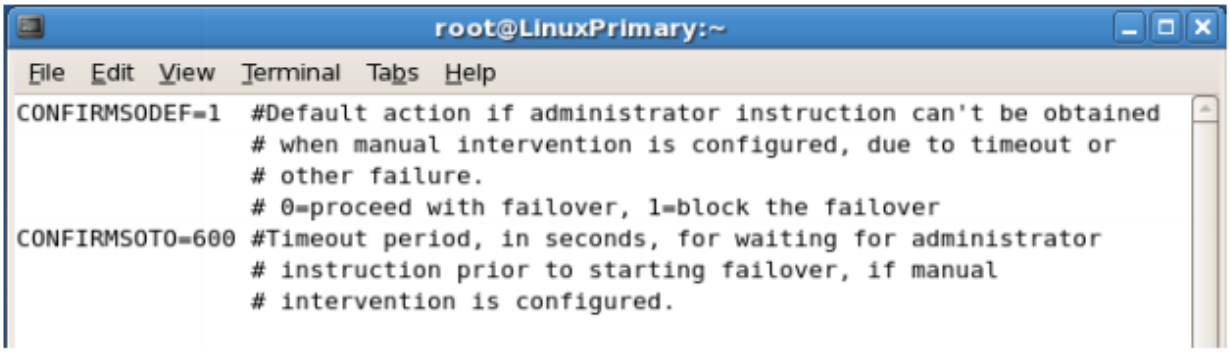
3. 「Server Properties」の画面が読み込まれたらページ下方にある全てのボックスにチェックマークを入れます。以上の手順で自動フェイルオーバーは行われなくなります。




4. 「Apply」をクリックします。

5. 今度は「Server」のドロップダウンリストからターゲットサーバーを選択し、手順2～4を繰り返します。

6. 次に、両ノードの `/etc/default/LifeKeeper` を編集します。
`CONFIRMSODEF=1` と設定します(数字を0から1に変えて下さい)。


root@LinuxPrimary:~
File Edit View Terminal Tabs Help
CONFIRMSODEF=1 #Default action if administrator instruction can't be obtained
# when manual intervention is configured, due to timeout or
# other failure.
# 0=proceed with failover, 1=block the failover
CONFIRMSOTO=600 #Timeout period, in seconds, for waiting for administrator
# instruction prior to starting failover, if manual
# intervention is configured.

# 11.1.7. DataKeeper for Linux の環境テスト

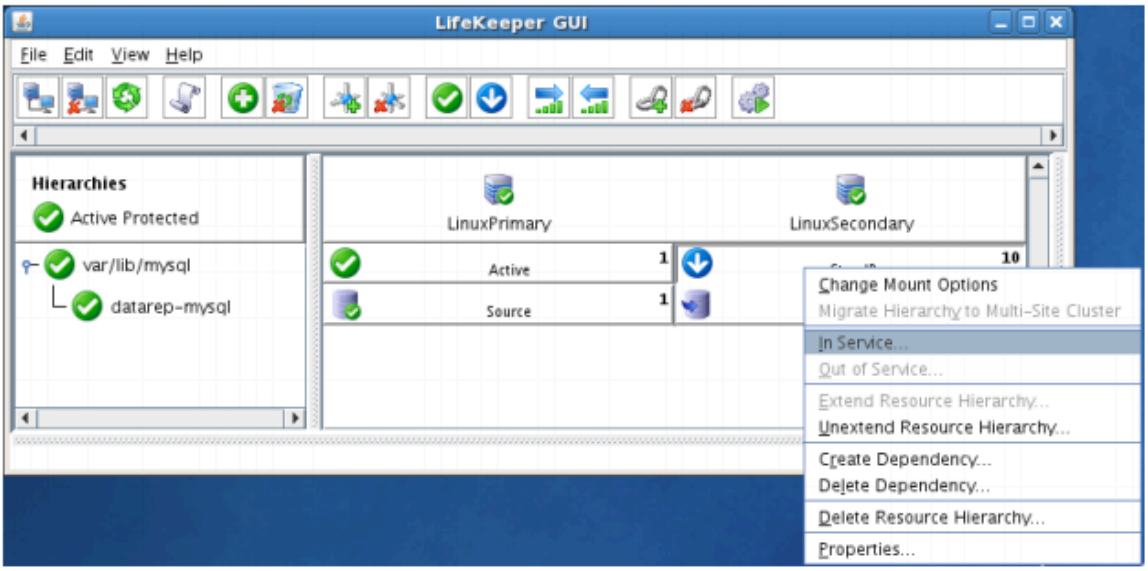
以下のテストシナリオはLifeKeeperの評価を始める際のガイドとしてお使い下さい。 テストを始める前にデータレプリケーションリソースがミラーリング状態であることを確認して下さい。

> 注記： このテスト例では、プライマリーノードはLinuxPrimary 、バックアップあるいはセカンダリーノードはLinuxSecondary と記載されています。

1. 手動スイッチオーバーによってミラーの方向をセカンダリーノードに向けて下さい。

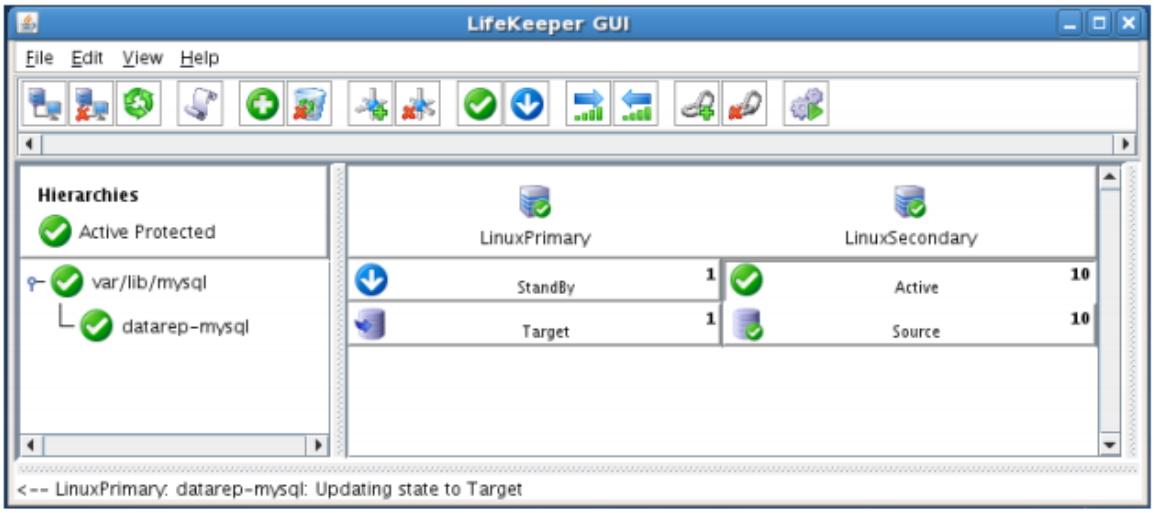
   **手順**

   - LifeKeeper GUIの画面のセカンダリーノード（LinuxSecondary）のリソース階層最上部を右クリックして「IN SERVICE」を選択します。
   - ポップアップ表示されるウインドウの「In Service」をクリックします。



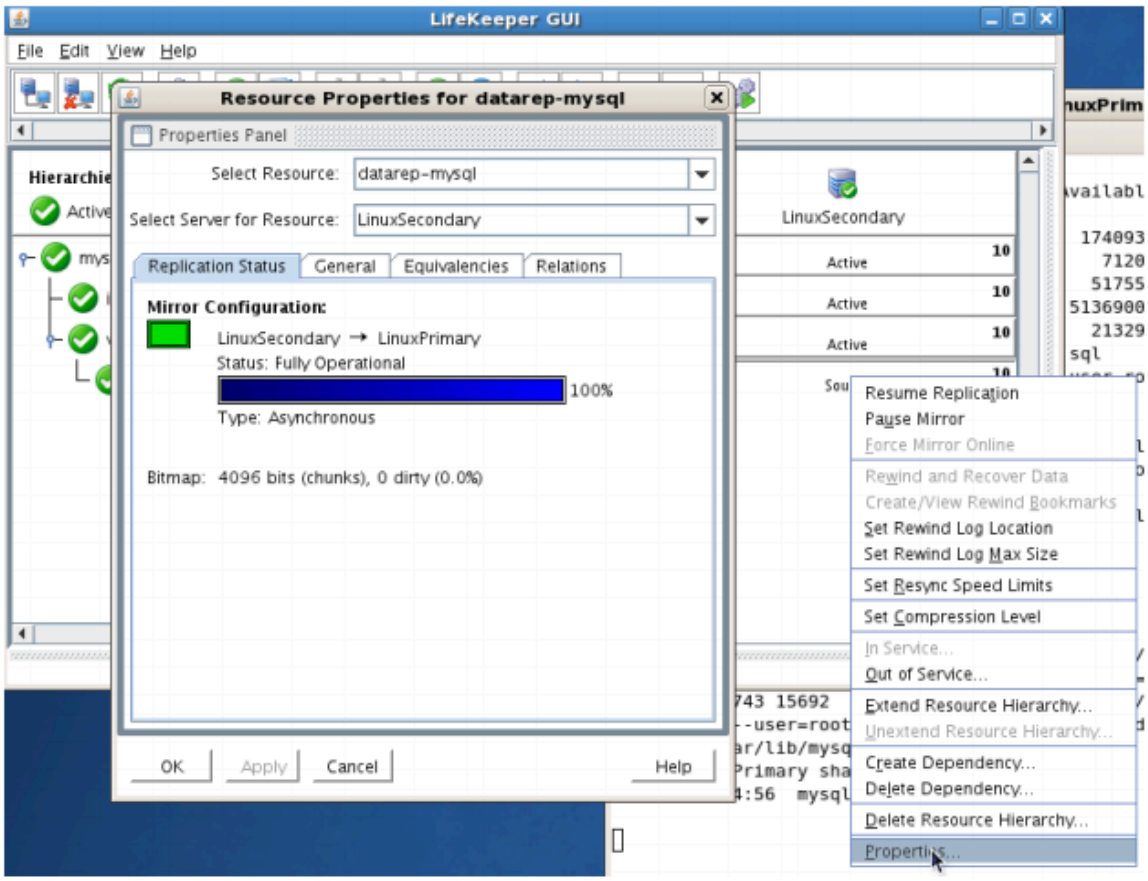

   **予想される結果**

   - アクティブノード（LinuxPrimary）のサービスから全リソースが停止されます。
   - 下位リソースから始まって全リソースがLinuxSecondaryで稼働状態（in sevice）になります。
   - このプロセスの間にミラーの方向が逆になり、今度はデータがLinuxSecondaryからLinuxPrimaryに転送されるようになります。
   - この時点で、LinuxSecondaryにおいて全リソースがアクティブになっています。




テスト/検証

- LifeKeeper GUIを使用して、ミラーの方向が逆になっており逆方向のミラーリングが行われていることを確認して下さい。リソース「datarep-mysql」を右クリックして「Properties」を選択します。




- `df -h` を実行して、複製されたファイルシステム `/var/lib/mysql` が「md」デバイス（例：`/dev/md0`）としてLinuxSecondary にマウントされていることを確認します。

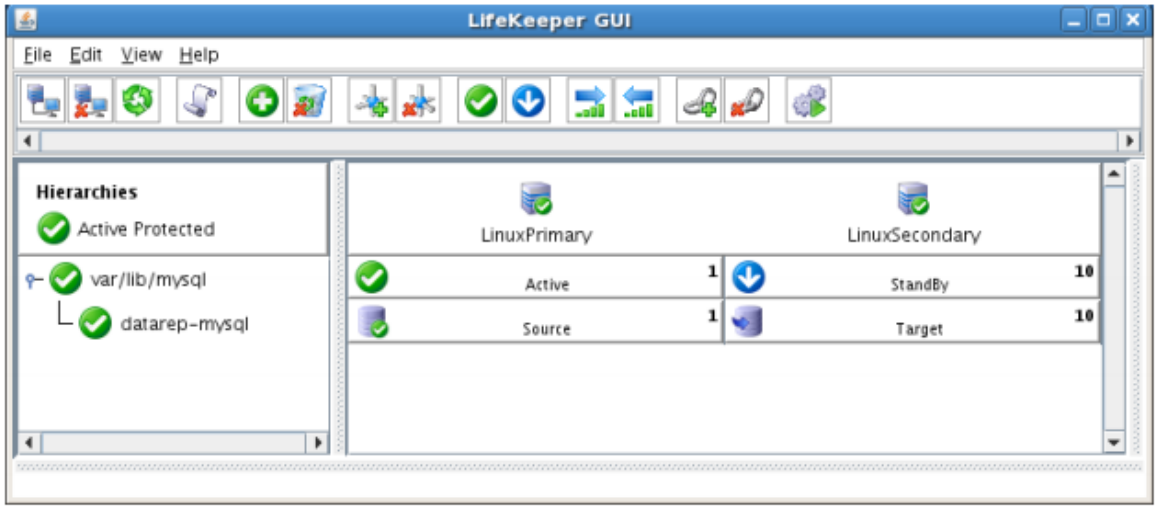
2. 手動スイッチオーバーによってミラーの方向をプライマリーノードへと戻します。

**手順**

- LifeKeeper GUIの画面のプライマリーノード（LinuxPrimary）のリソース階層最上部を右クリックして「In Service」を選択します。
- ポップアップ表示されるウインドウの「In Service」をクリックします。

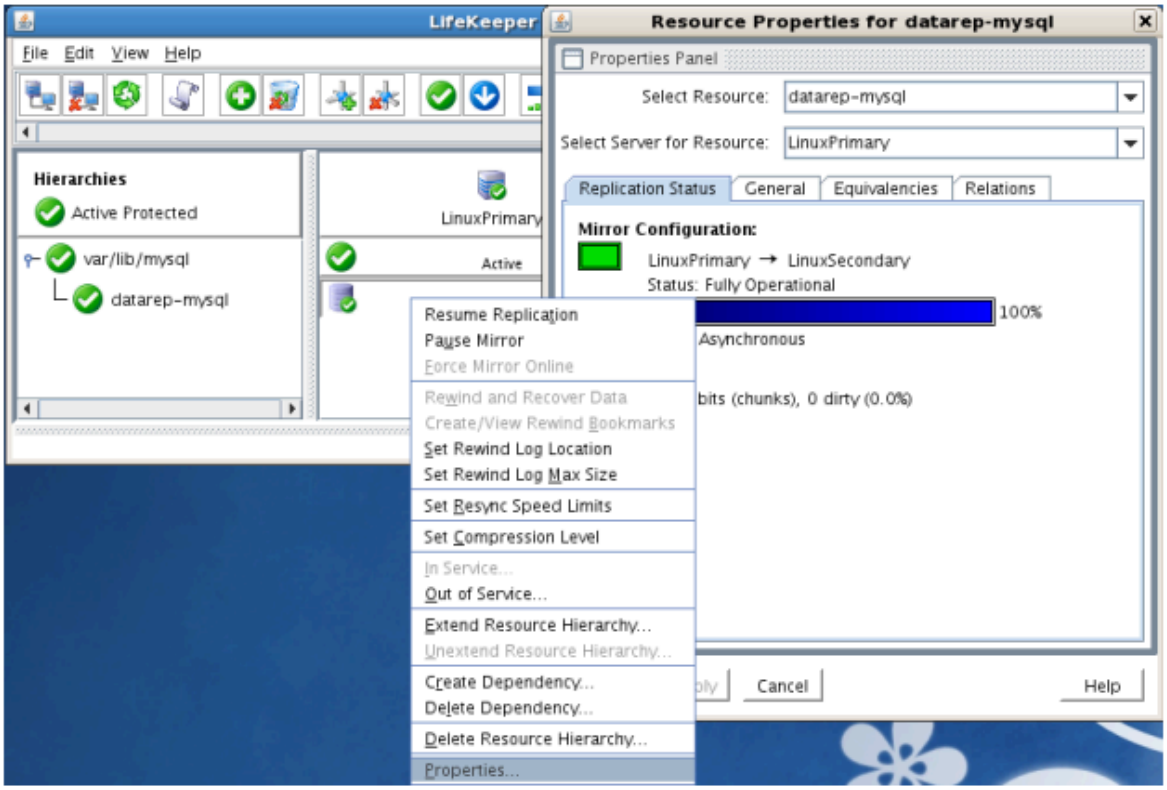
**予想される結果**

- アクティブノード（LinuxPrimary）のサービスから全リソースが停止されます。
- 下位リソースから始まって全リソースがLinuxPrimaryで稼働状態（in sevice）になります。
- このプロセスの間にミラーの方向が逆になり、今度はデータがLinuxPrimaryからLinuxSecondaryに転送されるようになります。




**テスト/検証**

- LifeKeeper GUIを使用して、ミラーの方向が逆になっており逆方向のミラーリングが行われていることを確認して下さい。リソース「datarep-mysql」を右クリックして「Properties」を選択します。




- `df -h` を実行して、複製されたファイルシステム `/var/lib/mysql` が「md」デバイス（例：`/dev/md0`）としてLinuxPrimary にマウントされていることを確認します。

# 11.2. クラウド環境向けLifeKeeper for Linux 評価ガイド

この評価ガイドでは、LifeKeeper の基本概念について説明した後、中核となるユーザーインターフェース要素について説明し、さまざまなリソースがどのように保護されるかを示します。このガイドでは、クラウド環境についていくつかの観点から説明していますが、オンプレミス環境でのLifeKeeperの評価にもご利用いただけます。

(日本語訳されていないページがあります。それらについては英語の情報をご参照ください。)

# 11.2.1. LifeKeeper for Linuxの評価を開始する前に

このセクションでは、ミッションクリティカルなワークロードに高可用性を提供するソフトウェアを評価する際に考慮すべき一般的なトピックについて説明します。

- 高可用性, RTO, and RPO
- LifeKeeper for Linux – 統合コンポーネント
- LifeKeeper for Linuxの利点
- クラウド環境に移行する際のワークロードの分散方法
- パブリッククラウドプラットフォームとそのネットワーク構造の違い
- クライアントがアクティブノードに接続する方法
- ノード間のデータレプリケーションはどのように行われるのか
- 「スプリットブレイン」とは何か、そしてそれを回避する方法

# 11.2.1.1. 高可用性、RTO（目標復旧時間）、RPO（目標復旧時点）

高可用性（HA）とは情報技術用語で、コンピューターのソフトウェアやコンポーネントが 99.99 %以上の時間稼働し、利用可能であることを意味します。アプリケーションやシステムのエンドユーザーが経験するサービスの中断が年間 52.5 分未満であること。このレベルの可用性は、通常、高可用性クラスタリングの使用によって達成できます。これは、冗長化されたサーバー、ネットワーク、ストレージ、およびソフトウェアの使用によって単一障害点を排除することにより、アプリケーションのダウンタイムを削減する構成です。

## 目標復旧時間（RTO）および目標復旧時点（RPO）とは？

高可用性環境は、99.99 %の可用時間に加え、厳格な目標復旧時間と目標復旧時点も満たしています。目標復旧時間（RTO）とは、アプリケーションの障害発生から、アプリケーションの動作と可用性を回復するまでの経過時間を示す指標です。これは、企業がそのアプリケーションの停止を許容できる時間を示す指標です。目標復旧時点（RPO）とは、ダウンタイムの問題からアプリケーションの可用性が回復したときに、データがどれだけ最新であるかを示す指標です。障害発生時に許容できるデータ損失の最大量と表現されることが多いです。SIOS の高可用性クラスターでは、RPO はゼロ、RTO は数分を実現しています。

> ✻ 同期ミラーの場合、ミラーリング状態では RPO が 0 (データ損失なし) になりますが、オフラインまたは再同期時にはそうではありません。 非同期ミラーの RPO は、通常は数秒未満であり、数ミリ秒程度になることもあります。

## 高可用性クラスターとは？

高可用性クラスターでは、重要なアプリケーションはプライマリサーバーノードで実行され、このノードは冗長性確保のために1つまたは複数のセカンダリーノードに接続されています。SIOS LifeKeeper などのクラスタリングソフトウェアは、クラスター化されたアプリケーションと依存するリソースを監視し、それらがアクティブノードで動作していることを確認します。システムレベルの監視は、クラスターノード間のインターバルハートビートによって実現されます。プライマリサーバーで障害が発生した場合、ハートビートのタイムアウト間隔を超えた後にセカンダリサーバーが復旧を開始します。アプリケーションレベルの障害では、クラスタリングソフトウェアがアクティブノードでアプリケーションが利用できないことを検出します。その後、フェイルオーバーと呼ばれるプロセスで、アプリケーションと依存リソースをセカンダリーノードに移動させ、そこで運用を継続し、厳格な RTO が満たされます。

従来のフェールオーバークラスターでは、クラスター内のすべてのノードが同じ共有ストレージ（通常はストレージエリアネットワーク（SAN））に接続されます。フェイルオーバー後、セカンダリーノードは共有ストレージへのアクセスを許可され、厳格な RPO を満たすことができるようになります。

# 11.2.1.2. LifeKeeper for Linux – 統合コンポーネント

SIOS Protection Suiteには、組織のミッションクリティカルなシステムを保護するための以下のソフトウェアコンポーネントが含まれています。

## SIOS LifeKeeper

SIOS LifeKeeperは、サーバー、ファイルシステム、アプリケーション、およびプロセスの高可用性を実現する、完全な耐障害性のあるソフトウェアソリューションを提供します。LifeKeeperは、カスタマイズされたフォールトトレラントなハードウェアを必要としません。 LifeKeeperでは、2つ以上のシステムをネットワーク内でグループ化するだけで、サイト固有の構成データが作成され、自動障害検出と回復を提供します。
障害発生時、LifeKeeperは保護対象のリソースを障害が発生したサーバーから指定されたスタンバイサーバーに移行させます。実際の切り替えの間、ユーザーは短い中断を経験しますが、LifeKeeperはオペレーターの介入なしにスタンバイサーバー上の操作を復元します。

## SIOS DataKeeper

SIOS DataKeeperは、LifeKeeper環境のための統合されたデータレプリケーション機能を提供します。この機能により、LifeKeeperのリソースは、共有および非共有ストレージ環境で動作することができます。

## Application Recovery Kits (ARKs)

Application Recovery Kit (ARK) は、LifeKeeperが特定のアプリケーションまたはサービスを管理および制御するためのツールとユーティリティを含みます。特定のアプリケーションのためにARKがインストールされている場合、LifeKeeperはアプリケーションの状態を監視し、障害が発生した場合、自動的にアプリケーションを回復できます。Recovery Kitはノンインタラクティブであり、LifeKeeperで保護するためにアプリケーション内で変更を加える必要はありません。

# 11.2.1.3. LifeKeeper for Linuxの利点

- SIOSのソフトウェアは、複数のオペレーティングシステムのバージョンと種類（LinuxとWindowsの両方）をサポートしています。
  - オペレーティングシステムに関係なく、一貫したユーザー体験でミッションクリティカルなリソースを保護します。

- SIOSは、オンプレミス、仮想化（VMware）、クラウド環境（AWS、Azure、Google Cloudなどのパブリックプラットフォーム、プライベートホスティングのクラウド環境など）を含む複数の環境に対して高可用性ソリューションを提供します。同じツールを異なる環境間で使用することができます。
  - また、ハイブリッド環境にも対応しています。オンプレミスのノードをプライマリーロケーションとして構成し、セカンダリーノードをプライベートクラウドでホストし、3番目のノードをパブリッククラウドに配置して、DR（災害復旧）オプションを追加する構成を組むことができます。

- SIOSは、Application Recovery Kit（ARK）を含む、アプリケーション対応の保護メカニズムを提供しており、エンタープライズアプリケーションやデータベースの世界的な主要プロバイダーに対応しています。
  - ARKの包括的なライブラリーは、幅広いアプリケーションを「そのまま」保護します。
  - その他の「非標準」またはレガシーアプリケーションは、LifeKeeper Coreの機能である「Generic Application」を使用するか、「Quick Service Protection（QSP）」機能を使用して保護することができます。

- オープンソースのツール（ユーザーが事前に環境パラメーターを手動で設定し、複雑なコマンドラインパラメーターを入力する必要がある）とは異なり、ウィザード形式のインストールおよび設定画面により、保護が必要なリソースを直感的に選択でき、提供する保護のタイプと範囲を選択しやすくなっています。
  - ウィザードは、システムと環境をスキャンして、保護するリソースを特定します。ほとんどの場合、オペレーターはデフォルトの選択項目を確認し、環境固有のホストIDなどを入力するだけでインストールが完了します。
  - これにより、HAソリューションの設定ミスや、システムやアプリケーションの障害による避けられない予期せぬダウンタイムが発生する可能性を低減することができます。

- SIOSは、24時間365日のクリティカルサポートを含む包括的なテクニカルサポートオプションを提供しており、予算、システムの複雑さ、高可用性を必要とするアプリケーションの重要性に応じたサポートオプションを提供しています。

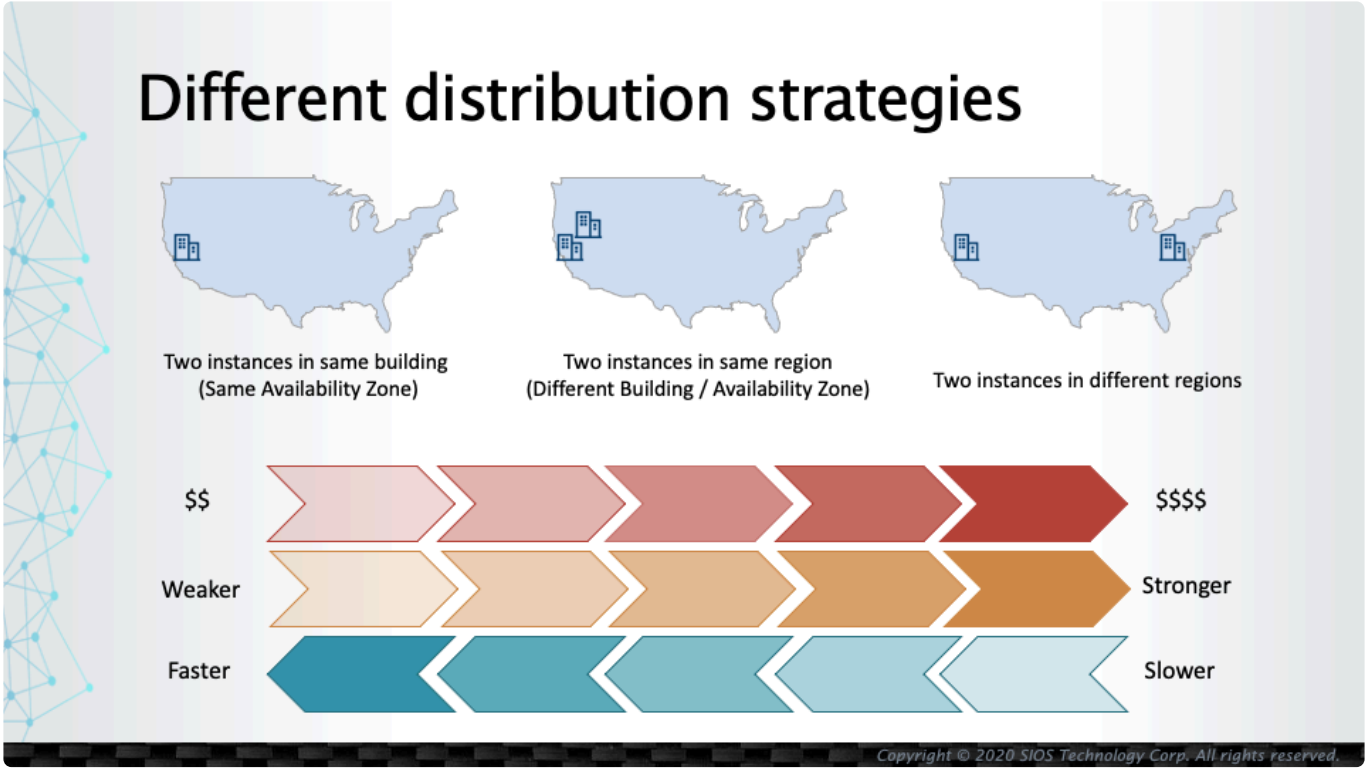
# 11.2.1.4. クラウド環境に移行する際のワークロード（ノード）の分散方法

ワークロード（ノード）をどのように分散させるかは、高可用性を念頭に置いたパブリッククラウドへの移行時によく議論されるテーマです。オンプレミス環境にワークロードが配置されている場合、通常これらのワークロードの配置場所は、確立されたデータセンターの場所によって定義されます。多くの場合、ワークロードをホストするために別の場所を選ぶという選択肢はありません。しかし、パブリッククラウドでは、幅広い地域とアベイラビリティゾーンを選択することができます。

アベイラビリティゾーンとは、一般的に、1つ以上のデータセンター（物理的な場所）が物理的に同じリージョン（カリフォルニア州など）に位置しているようなものです。これらのデータセンターは、異なる地域に位置していても、高速ネットワークで接続されているため、データセンター間の接続遅延を最小限に抑えることができます。（リージョン内の複数のデータセンターでサービスをホスティングすることは、ユーザーにとって透過的でなければならないことに注意してください）。

| | AWS | Azure | Google Cloud | Oracle Cloud | Virtustream | Cross-Subnet Failover |
|---|---|---|---|---|---|---|
| DR リージョンにフェイルオーバー可能 | ✓ | X | X | X | X | X |
| データを DR リージョンにレプリケート可能 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

一般的に、ワークロード間の物理的な距離が遠いほど、耐障害性の高い環境になります。地震などの自然災害が異なる地域で同時に発生することはない（例：米国西海岸と東海岸の両方が同時に発生する）と考えるのが妥当です。しかし、システム全体の障害により、異なるリージョンで同時にサービス停止が発生する可能性はあります（一部のクラウドプロバイダーでは、米国とオーストラリアのように地域をまたいだ同時停止が報告されたことがあります）。異なるクラウドプロバイダー間でDR（災害復旧）計画を策定することを検討すべきかもしれません。

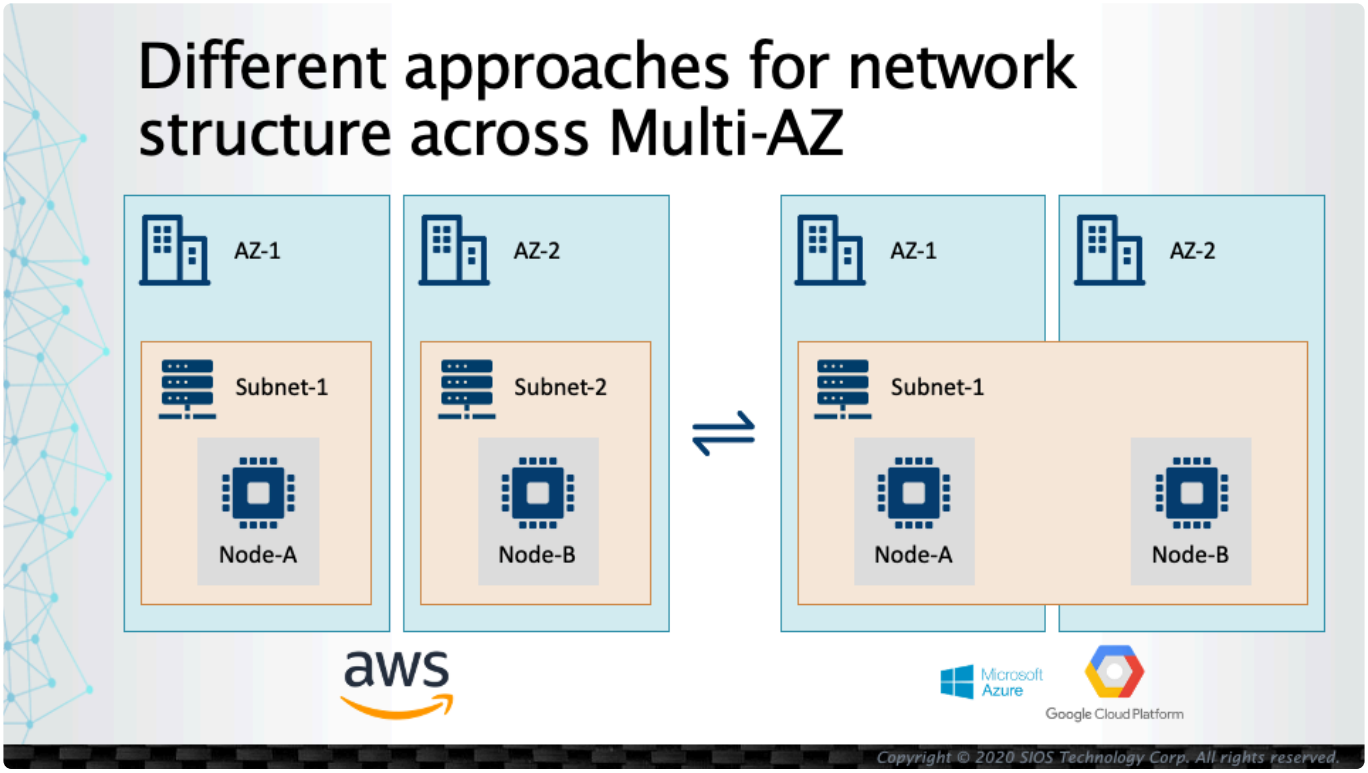



検討に値するもう1つの視点は、リソースを保護するためのコストです。一般に、ワークロード間の距離が遠くなればなるほど、データ転送にかかるコストは高くなります。多くの場合、同じデータセンター（アベイラビリティゾーン）内のノード間のデータ転送は無料ですが、アベイラビリティゾーンをまたぐデータ転送には1GBあたり0.01ドルまたはそれ以上のコストがかかる場合があります。この追加コストは、リージョン間でデータを転送する場合は2倍以上（0.02ドル/GBなど）になる場合があります。さらに、ワークロード間の物理的な距離が遠くなるため、拠点間のノード間のデータ遅延が大きくなることが予想されます。これらの要因を考慮すると、一般的には、同一リージョン内のアベイラビリティゾーンにワークロードを分散させることをお勧めします。

# 11.2.1.5. パブリッククラウドプラットフォームとそのネットワーク構造の違い

パブリッククラウドには、Amazon Web Services (AWS)、Microsoft Azure、Google Cloudなど、いくつかのプラットフォームが存在します。それぞれのインフラには多くの共通点がありますが、いくつかの相違点もあります。多くの場合、リージョンに紐づくVPC（仮想プライベートクラウド）やVNET（仮想ネットワーク）が作成されます。アプリケーションの論理グループに対して、1つまたは複数のVPCを定義することができます。そうすることで、異なるVPCが特別に接続されていない限り、異なるシステムは接続されていない別々のネットワークに分割されます。

VPCでは、さまざまなサブネットを定義できます。目的に応じて、あるサブネットはインターネットにアクセス可能な「パブリック」サブネットとして設定し、あるサブネットはインターネットにアクセスできない「プライベート」サブネットとして設定できます。
一部のクラウドプロバイダー（AzureやGoogle Cloudなど）では、可用性ゾーン（異なるデータセンター）間でサブネットを定義できますが、一部（AWSなど）では、可用性ゾーン間でサブネットを定義できません。後者の場合、可用性ゾーンごとにサブネットを定義する必要があります。




このガイドでは、各ノードに対して異なる可用性ゾーンを使用します。SIOS製品の基本的な機能を理解したら、ワークロードを異なるサブネットに分散させたり、サブネットのIP範囲を変更したり、ネットワークのインターネットへの接続方法を変更するなど、さまざまなシナリオ（お客様自身のネットワークインフラで使用されているもの）を検討することをお勧めします。

# 11.2.1.6. クライアントがアクティブノードに接続する方法

前述したように、高可用性クラスターが構成されると、2つ以上のノードが同時に動作し、ユーザーは「アクティブ」ノードに接続します。アクティブノードで問題が発生すると、「フェイルオーバー」状態が発生し、「スタンバイ」ノードが新しい「アクティブ」ノードになります。フェイルオーバーが発生した場合、クライアントがフェイルオーバー状態を検出して再接続できるようにするメカニズム、またはユーザーのアクティブなクライアントセッションをアクティブノードにシームレスに転送する仕組みが必要です。

## 仮想IPアドレス

通常、「仮想」IPアドレスは、クラスターが構成され、クライアントが仮想IPアドレスを使用してアクティブノードと通信するときに作成されます。フェイルオーバーが発生すると、仮想IPアドレスは新しいアクティブノードに再割り当てされ、クライアントは同じ仮想IPアドレスに再接続します。

例として、IPアドレスが `10.20.1.10`と`10.20.2.10`の2つのノードAとBがあると仮定します。この例では、現在アクティブなノードに割り当てられていると考えられる10.20.0.10の仮想IPアドレスを定義します。

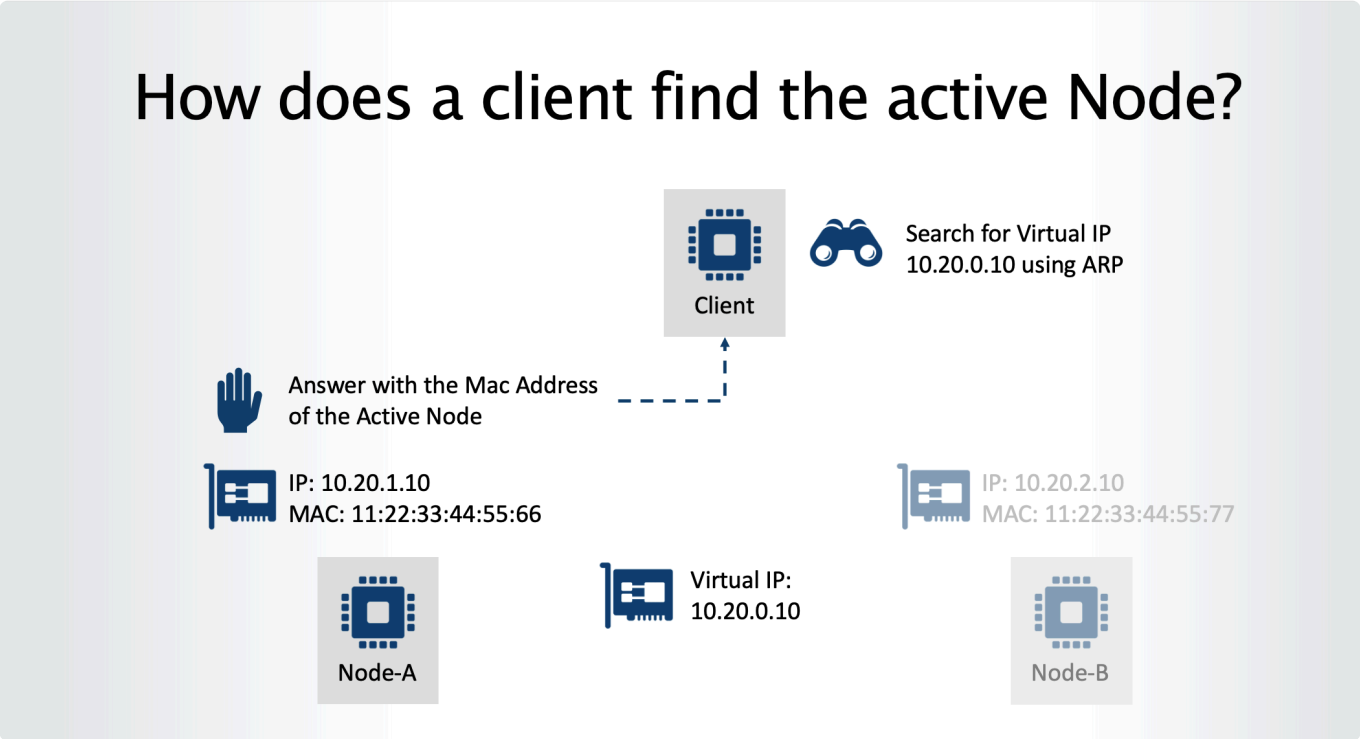
これは、1つのノードの1つのネットワークインターフェースカードに2番目のIPアドレスを割り当てるのと同じです。コマンドip aをアクティブノードで実行すると、両方のIPアドレスが表示されます（以下のLinuxの例の10行目と12行目）。

```
[root@node-a default]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0d:3a:6e:1f:74 brd ff:ff:ff:ff:ff:ff
    inet 10.20.1.10/22 brd 10.20.3.255 scope global noprefixroute eth0
       valid_lft forever preferred_lft forever
    inet 10.20.0.10/22 scope global secondary eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::20d:3aff:fe6e:1f74/64 scope link
       valid_lft forever preferred_lft forever
```

## ARPプロトコル

クライアントがIPアドレスを使用してサーバーを見つけようとする時、クライアントは通常、ARP（アドレス解決プロトコル）を使用してターゲットマシンのMAC（メディアアクセス制御）アドレスを探しま

す。




クライアントがターゲットIPアドレスを見つけるためにメッセージをブロードキャストすると、アクティブノードはそのMACアドレスで応答し、クライアントは要求を解決してそれに接続します。

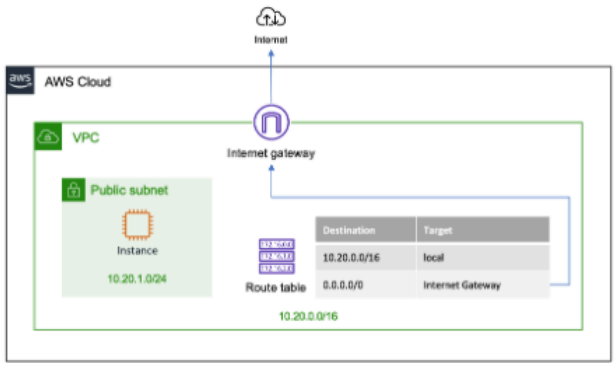
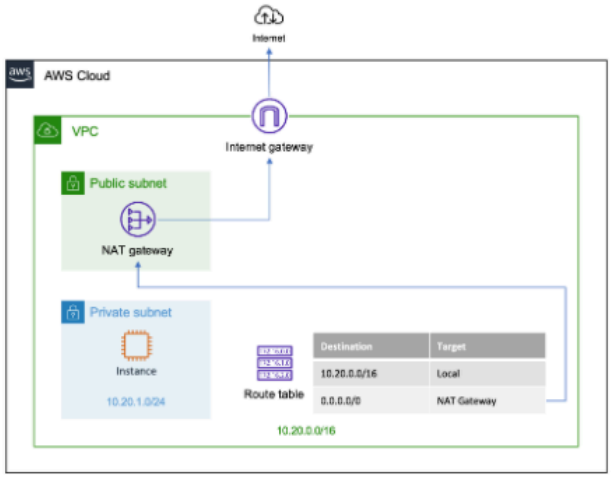
## クラウド環境におけるARPの代替手段

しかし、クラウド環境では、仮想環境で多くのレイヤーが抽象化されているため、ARPを使用してアクティブノードを特定することはできません。特定のクラウド環境で使用されているネットワークインフラに基づいた代替方法が必要になる場合があります。通常、以下のオプションのいずれかを選択します。

- AWS Route Table シナリオ
- AWS Elastic IP シナリオ
- AWS Route53 シナリオ
- Azure Internal Load Balancer シナリオ
- Google Cloud Internal Load Balancer シナリオ

# 11.2.1.6.1. AWS Route Table シナリオ

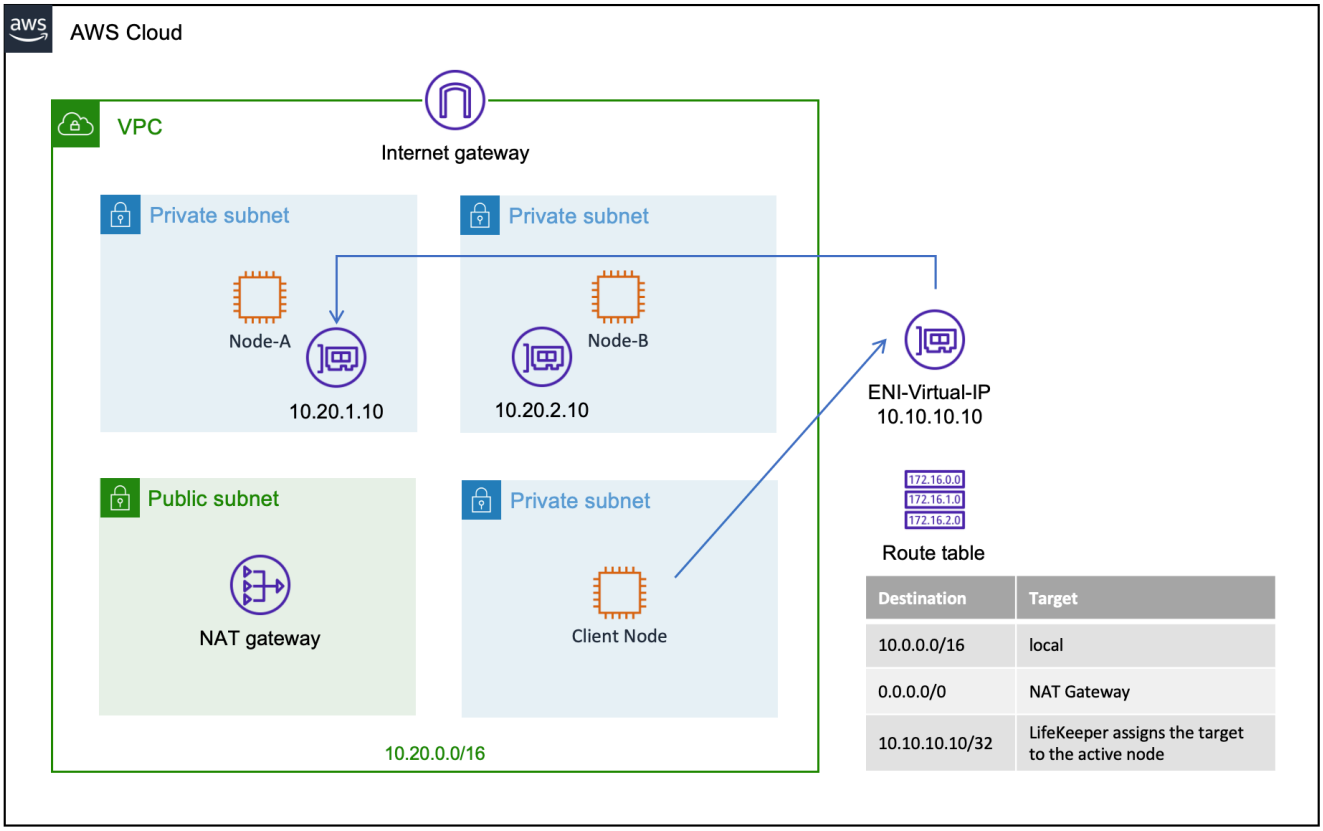
AWSでインスタンスを作成する場合、最初のステップの1つはルートテーブルを作成することです。ルートテーブルは、各宛先に対するルートを定義します。次の図では、インスタンスには2つのルートがあり、1つはVPC境界（ローカルネットワーク）である`10.20.0.0/16`に対するルートです。`10.20.0.0/16`内のトラフィックは、ローカル接続とみなされます。もう1つのルートは`0.0.0.0/0`で、これはIPトラフィックをローカルVPC境界の外（他のVPCまたはインターネット）にルーティングすることを許可するものです。インターネットに接続するには、インターネットゲートウェイが必要である（パブリックサブネットの場合）か、NATゲートウェイを指定する必要があります（プライベートサブネットの場合）。

| When Instances are located in Public Subnet | When Instances are located in Private Subnet |
|---|---|
|  |  |



## VPC内のアクティブなノードの検索

クラスタリング環境でアクティブノードを見つける場合は、 *ルート* を定義できるように、 ***VPC CIDRの外部*** で仮想IPアドレスを定義する必要があります。実際、仮想IPを定義するということは、 *ルートテーブルに新しい宛先を作成する* ことを意味します。

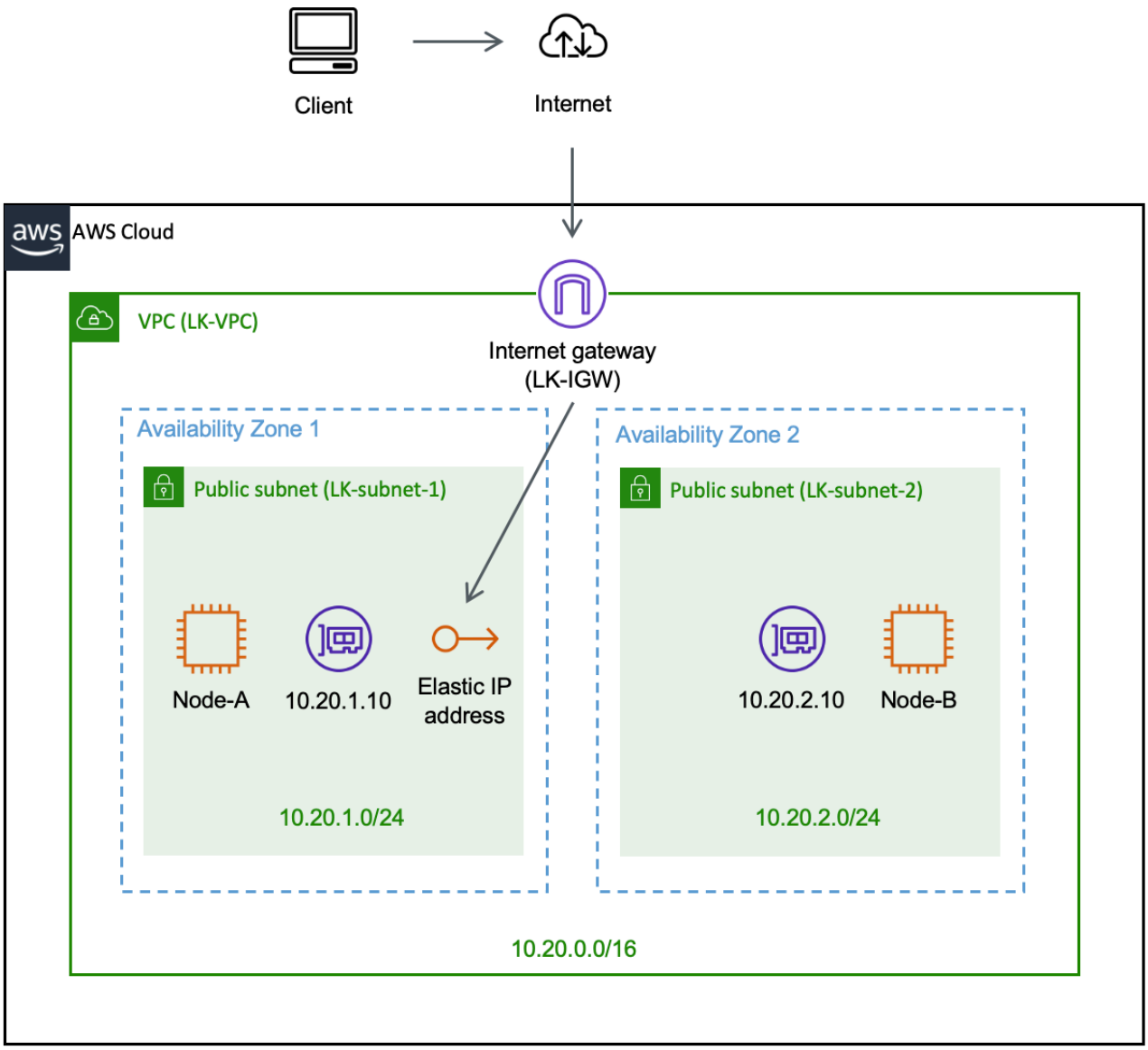



VPC CIDRの外部の新しい宛先が定義されると、SIOS Protection Suiteは、このルートのターゲットをアクティブノード（上記の構成例では10.20.1.10 または 10.20.2.10 ）に変更できます。クライアントノードは、宛先として 10.10.10.10 を探すだけでよく、ルーティングテーブルがトラフィックをアクティブノードに誘導します。ルーティングテーブルは、Recovery Kit for EC2によって動的に更新されます。

このシナリオの技術的な詳細については後述の、 AWS EC2リソースの作成（RouteTableシナリオ） を参照してください。.

# 11.2.1.6.2. AWS Elastic IPシナリオ

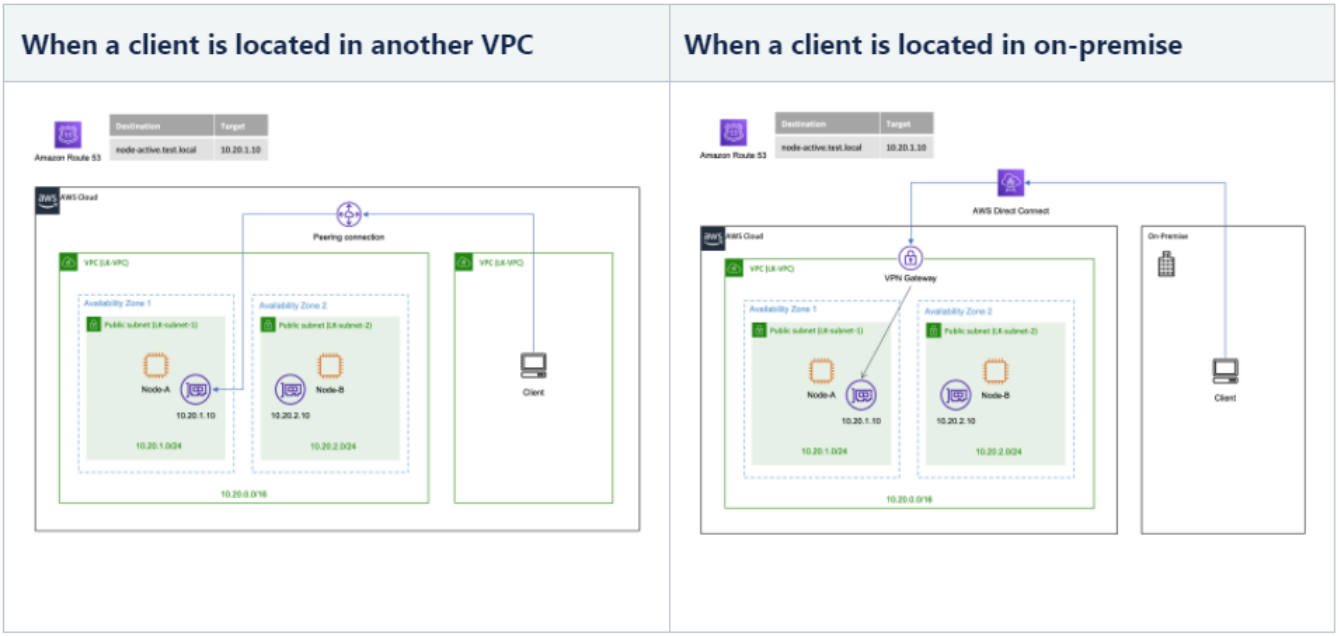
クライアントがインターネット上にあり、アクティブなクラスターノードへのパブリックアクセスが必要な場合は、Elastic IPシナリオを選択する必要があります。この例では、SIOS Recovery Kit for EC2がノード間の切り替え時にElastic IPを切り替え、新しいアクティブノードのENI（Elastic Network Interface）にElastic IPを接続します。




このシナリオの技術的な詳細については AWS EC2リソースの作成（Elastic IPシナリオ）を参照してください。

# 11.2.1.6.3. AWS Route53 シナリオ

前のセクションでは、クライアントがターゲットノードと同じVPC内にある場合のルートテーブルシナリオについて説明しました。しかし、クライアントノードが別のVPC内にある場合、またはオンプレミス環境にある場合（ダイレクトコネクトまたはVPNを介してAWSに接続されている）、ルートテーブルシナリオを使用してアクティブなターゲットノードを指定することはできません。この場合、Route53を使用して、クライアントトラフィックをアクティブなターゲットノードにルーティングできます。




Route53 Recovery Kit（SIOS Protection Suiteの一部として提供）は、VPCピアリングまたはDirect Connectのいずれかでノードに接続できる限り、DNSエントリーを更新してクライアントがアクティブなノードに接続できるようにします。

> このシナリオの技術的な詳細については AWS Route53リソースの作成 を参照してください。

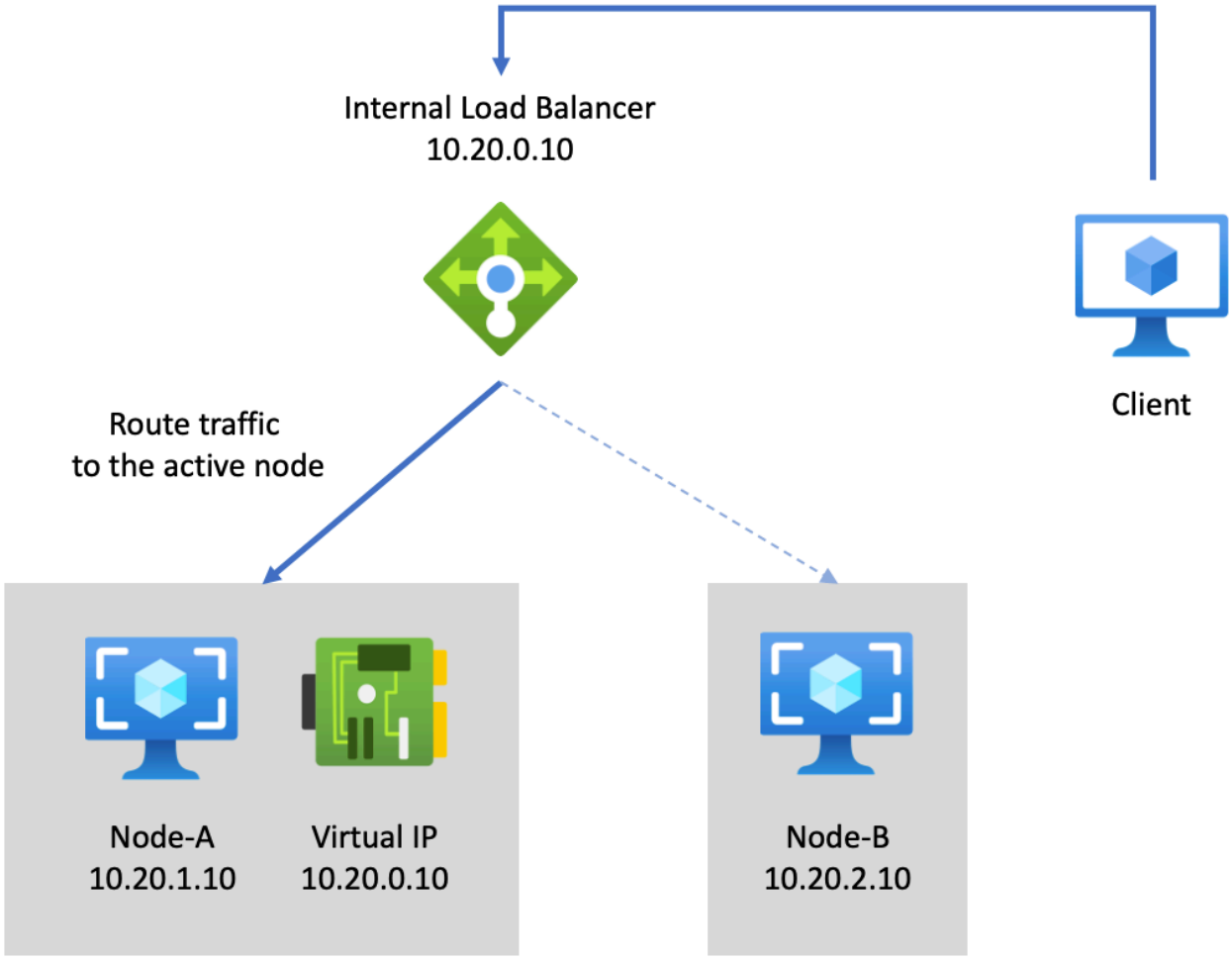
# 11.2.1.6.4. Azure Internal Load Balancer シナリオ

Azureでは、クラウド環境で多くのインフラレイヤーが抽象化されているため、仮想IPを使用してもアクティブノードにトラフィックが解決されない場合があります。この問題を解決するには、内部ロードバランサーを使用してトラフィックをアクティブノードに転送する必要があります。

例として、2つのノード（10.20.1.10 と 10.20.2.10）がクラスターとして動作しており、アクティブノードを指す仮想IP 10.20.0.10が作成されていると仮定します。
この場合、Azure Load Balancerはフロントエンドipアドレス 10.20.0.10 で作成され、同じIPアドレス 10.20.0.10 の仮想IPアドレスが作成されます。

これにより、クライアントは常に仮想IPアドレス 10.20.0.10を介して最初にロードバランサーに接続されるようになります。

ロードバランサーはヘルスプローブを使用して現在どのノードがアプリケーションをホストしているかを判断し、すべてのクライアント接続をそのアクティブなノードに転送します。



このシナリオの技術的な詳細については、Azure – Internal Load Balancerを使用する を参照してください。

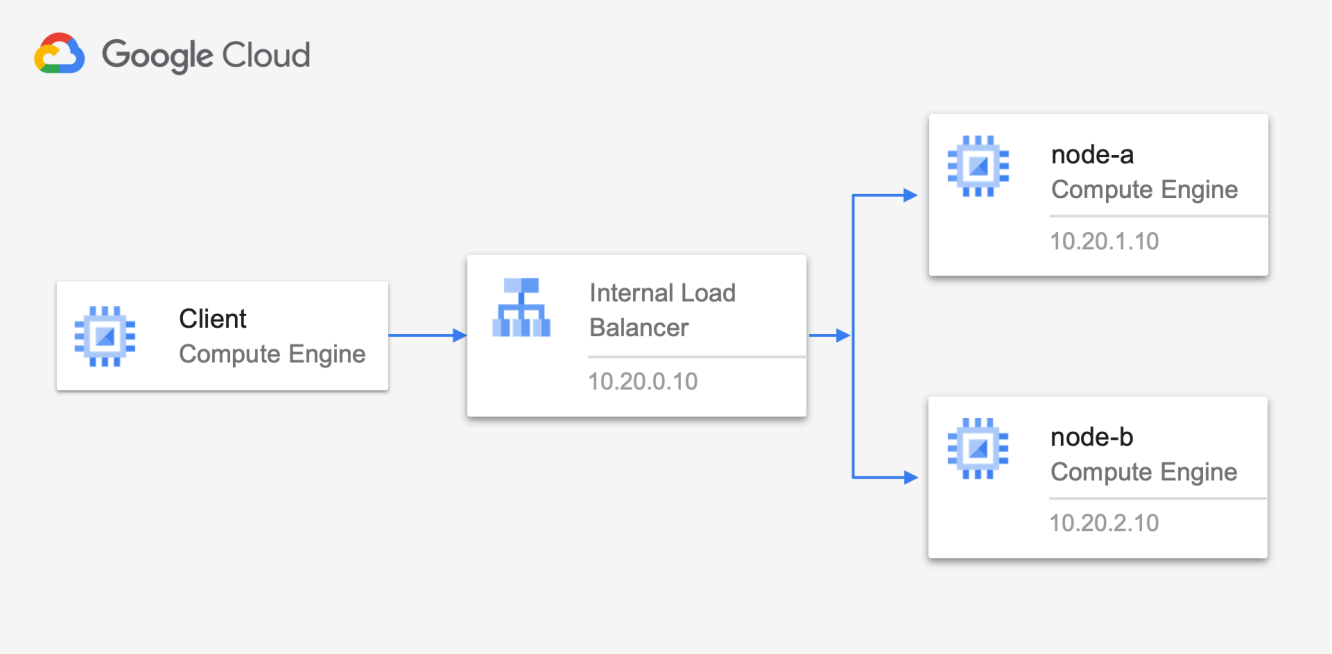
# 11.2.1.6.5. Google Cloud Internal Load Balancer シナリオ

Google Cloudでは、多くのインフラレイヤーが抽象化されているため、仮想IPを使用してもアクティブノードにトラフィックが解決されない場合があります。この問題を解決するには、内部ロードバランサーを使用してアクティブノードにトラフィックを転送する必要があります。

例として、2つのノード (`10.20.1.10` と `10.20.2.10`)がクラスターとして動作し、アクティブノードを指す仮想IP `10.20.0.10` が作成されていると仮定します。

この場合は内部ロードバランサーはフロントエンドIPアドレス`10.20.0.10`で作成されます。

ロードバランサーはヘルスプローブを使用して、現在どのノードがアプリケーションをホストしているかを判断し、すべてのクライアント接続をそのアクティブなノードに転送します。



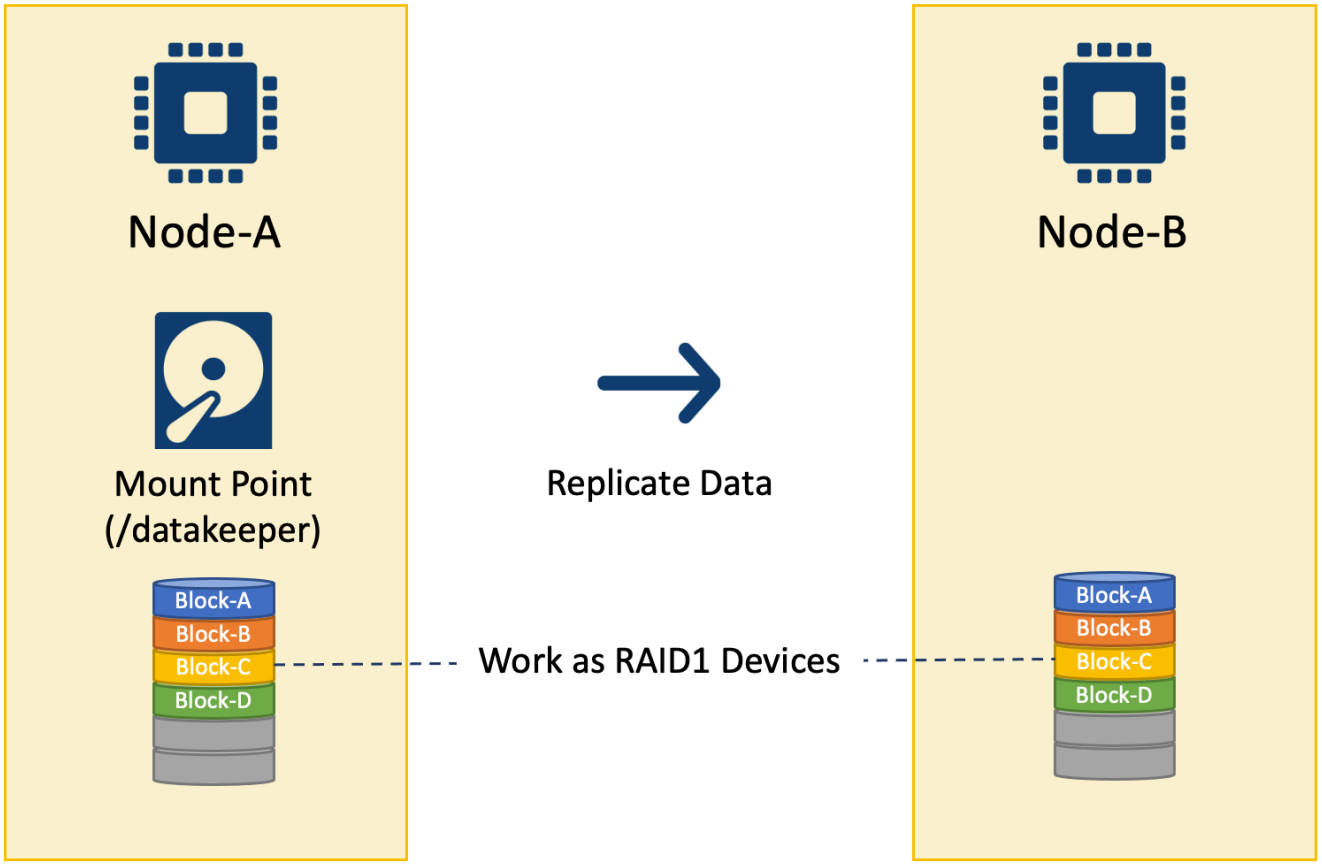

このシナリオの技術的な詳細については Google Cloud – Internal Load Balancerを使用する を参照してください。

# 11.2.1.7. ノード間のデータレプリケーションが行われる方法

従来のデータセンターのシナリオでは、データは一般的にストレージエリアネットワーク（SAN）に保存されます。クラウド環境では通常、共有ストレージはサポートされていません。

SIOS DataKeeperは、レプリケーションテクノロジーを使用して「共有」ストレージを提供し、現在アクティブなデータのコピーを作成します。RAID1デバイス（デバイス間でミラーリングされたデータ）として機能するNetRAIDデバイスを作成します。
データの変更は、ミラーソース（アクティブノード上のディスクデバイス – 下図のノードA）からミラーターゲット（スタンバイノード上のディスクデバイス – 下図のノードB）に複製されます。

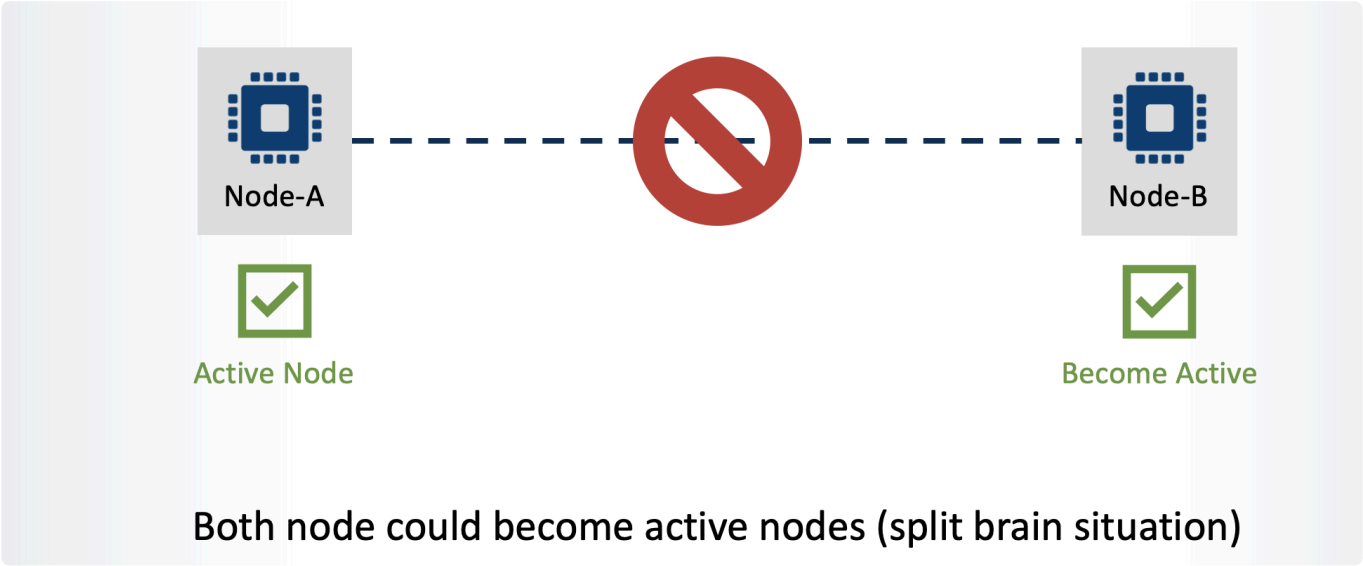
両デバイス間のデータの整合性を保証するため、アクティブノードのみが複製されたデバイス（以下の例では/datakeeperマウントポイント）への書き込みアクセス権を持ちます。複製されたデバイス（/datakeeperマウントポイント）がミラーターゲットの間（スタンバイノード上）はアクセスできません。



# 11.2.1.8.「スプリットブレイン」とは何か、そしてそれを回避する方法

これまで説明してきたように、高可用性クラスター環境では、1つのアクティブノードと、アクティブノードが故障したり応答しなくなったりしたときにサービスを引き継ぐ1つ以上のスタンバイノードが存在します。

これは、ノード間のネットワーク層を考慮するまでは、合理的な仮定であるように聞こえます。ノード間のネットワークパスがダウンした場合はどうなるのでしょうか？

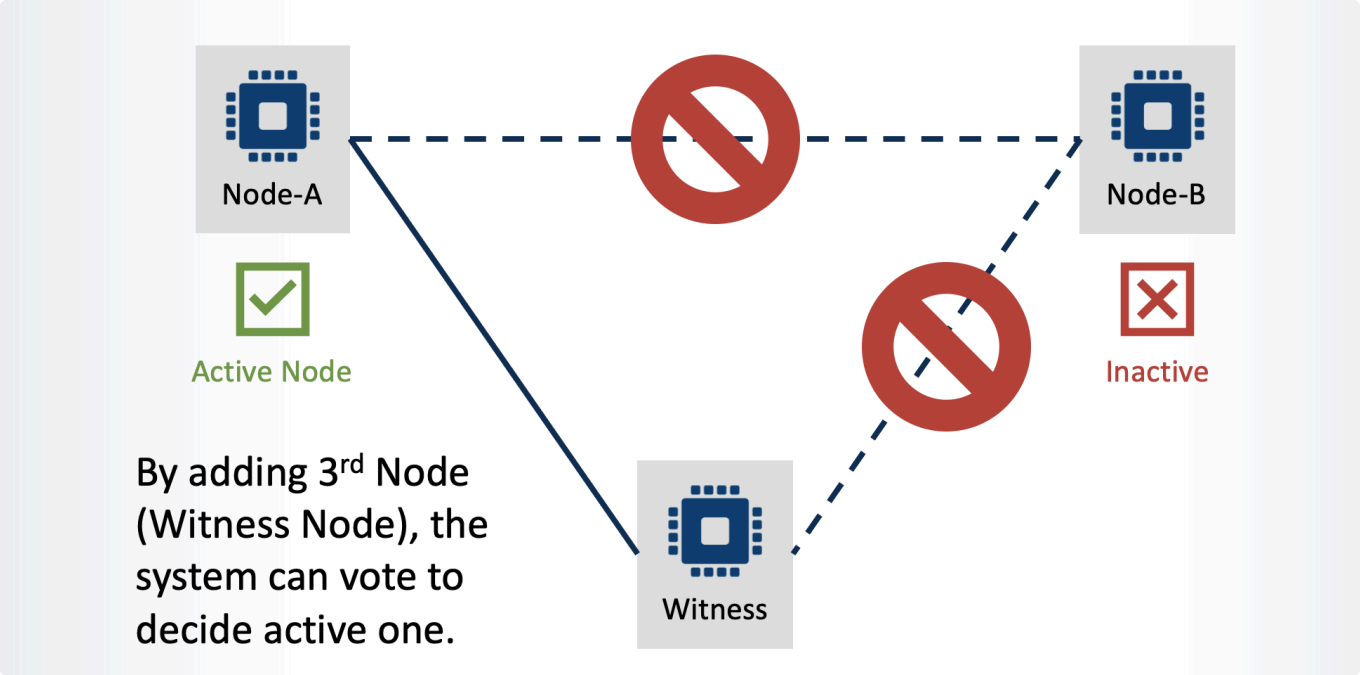



どちらのノードも他方のノードと通信できなくなり、このような状況では、スタンバイサーバーは、アクティブノードで障害が発生したと考えて、自身をアクティブサーバーに昇格させる可能性があります。この結果、両方のノードが「アクティブ」になり、それぞれがもう一方のノードを停止していると見なすことになります。その結果、両方のノードのデータが変更されるため、データの整合性と一貫性が損なわれます。これを「 **スプリットブレイン** 」といいます。

スプリットブレインのシナリオを回避するためには、Quorumノード（「Witness」とも呼ばれます）をクラスター内にインストールする必要があります。偶数ノードで構成されるクラスターにQuorumノードを追加すると、奇数ノード（3、5、7など）が作成され、ノードが投票してクラスター内のアクティブノードとし

て動作するノードを決定します。

以下の例では、ノードBを含むサーバーラックがLAN接続を失っています。このような場合でも、3台目のノードをクラスター環境に追加することで、どのノードをアクティブノードとするかをシステムが判断することができます。




Quorum/Witness機能は、SIOS Protection Suiteに含まれています。インストール時に、Quorumノードだけでなくすべてのノードでで Quorum/Witnessが選択され、Quorumノードを含むすべてのノード間でコミュニケーションパスが定義されます。

Quorumノードは、アクティブなサービスをホストしません。Quorumノードの唯一の役割は、ノード通信に参加してアクティブなノードを特定し、通信が停止した場合に「タイブレーク投票」を提供することです。

SIOSは、Quorumデバイスとして IOフェンシングとストレージ もサポートしており、これらの構成では、追加のQuorumノードは必要ありません。

# 11.2.2. 本書で使用するドキュメントのスタイル

このチュートリアルで説明する手順のほとんどは、2つのノード（node-aとnode-b）で構成される環境について説明しています。ユーザーコマンドの実行例は、本ガイド中に記載しています。以下のセクションでは、コマンドの実行に使用される異なるノードまたはユーザーを示すために使用されるスタイルを示します。

## アクションを実行するノード

特定のノードで特定のコマンドを実行する必要がある場合、プロンプトには以下のようにユーザー名とホスト名が表示されます。:

```
[root@node-a]# <command name and its parameters are listed here>
```

また、以下のようなインラインコメントも表示されます。:

> You need to execute the following steps on the primary node (node-a).

プロンプトでノード名が指定されていない場合、これらのコマンドは両方のノードで実行する必要があります（次の例を参照してください）。:

```
# <command name and its parameters are listed here>
```

また、以下のようなインラインコメントも表示されます。:

> You need to execute the following steps for each node.

## コマンドを実行するユーザー

ガイド内ではコマンドを実行するユーザーは、使用されるプロンプトによって識別されます。

一般ユーザー（例：ほとんどのEC2インスタンスでは `ec2-user`）としてコマンドを実行した場合、次のようなプロンプトが表示されます。

```
$ date
Thu Nov 26 22:58:22 UTC 2020
```

root ユーザーになると、プロンプトは # と記述されます。

```
$ sudo su -
# shutdown -h now
```

特定のユーザーがコマンドを実行する必要がある場合、以下のようにプロンプトにユーザー名が含まれます。

```
[oracle]$ passwd
```

## SIOS GUI（ウィザード）操作時の選択オプション

SIOS Protection Suiteを使用する利点の1つは、ウィザードがシステムをスキャンし、インストール画面内で推奨値を提供することです。ほとんどの場合、デフォルト値を確認･選択するだけで済みます。デフォルト値が選択できる場合は、ガイドではチェックマーク を表示します。チェックマークが表示されていない場合は、特定の値（環境固有のIPアドレスなど）を入力する必要がある場合があります。

ガイドに記載されているパラメーターを変更する場合（例：別のリソース名、別のIPアドレスの選択など）、導入後にパラメーターが出現するたびに更新された値を使用するように注意する必要があります。

# 11.2.3. ネットワークコンポーネントの構成とインスタンスの作成

このセクションでは、各ノードに必要なコンピューティングリソース、ネットワーク構造、およびこれらのコンポーネントの構成に必要なプロセスの概要を説明します。

- 本チュートリアルで使用するネットワーク構成
- 本チュートリアルで使用するコンピューティングリソース

以下のセクションでは、これらのネットワークコンポーネントを設定し、クラウドプラットフォーム上にインスタンスを作成します。

- AWSでゼロからインスタンスを作成する
- Azureでゼロからインスタンスを作成する
- Google Cloudでゼロからインスタンスを作成する

# 11.2.3.1. 本チュートリアルで使用するネットワーク構成

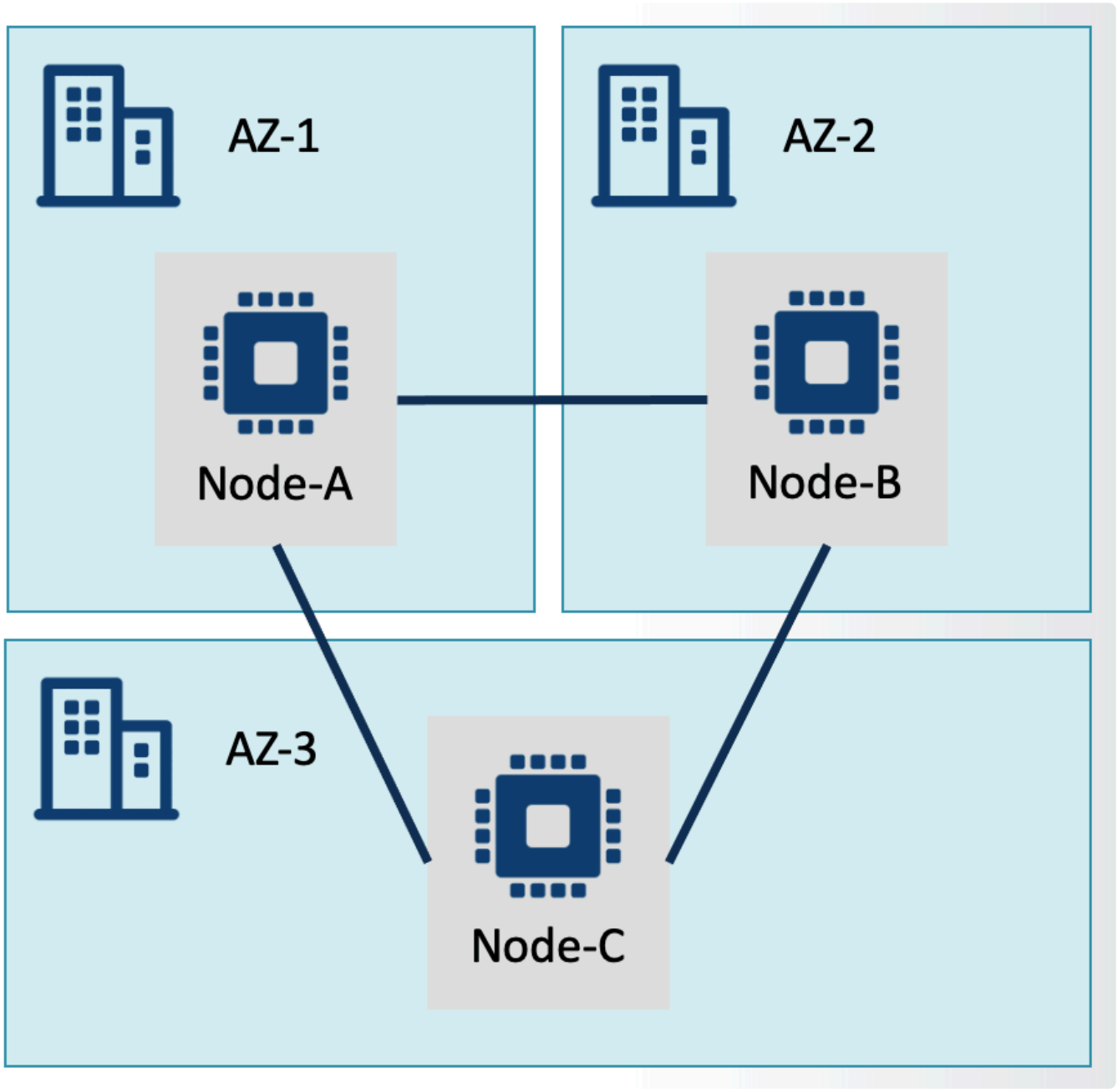
このチュートリアルでは、3つのノードを作成します。Node-AとNode-Bはそれぞれアクティブ・ノードとスタンバイ・ノードとして使用されます。これら2つのノードに加え、Node-C（Quorum/Witnessノード）も作成します。(Quorum/Witness Nodeの必要性については、 What is を参照してください）。

クラウド環境に移行する際のワークロードの分散方法 で説明したように、SIOSは異なるアベイラビリティゾーン（データセンター）内でワークロードを分割して使用することを強く推奨しています。

> この評価ガイドでは、Quorum / Witnessノード（node-c）について説明しますが、学習を容易にするためにこの3番目のノードはスキップしても構いません。しかし、本番環境（または正式なPoCプロジェクト）では、SIOSはQuorum/Witnessノードの使用を強く推奨しています。

以下のトピックでは、異なるアベイラビリティゾーンのノードを作成します。

- AWSでゼロからインスタンスを作成する
- Azureでゼロからインスタンスを作成する
- Google Cloudでゼロからインスタンスを作成する


AZ-1
AZ-2
Node-A
Node-B
AZ-3
Node-C

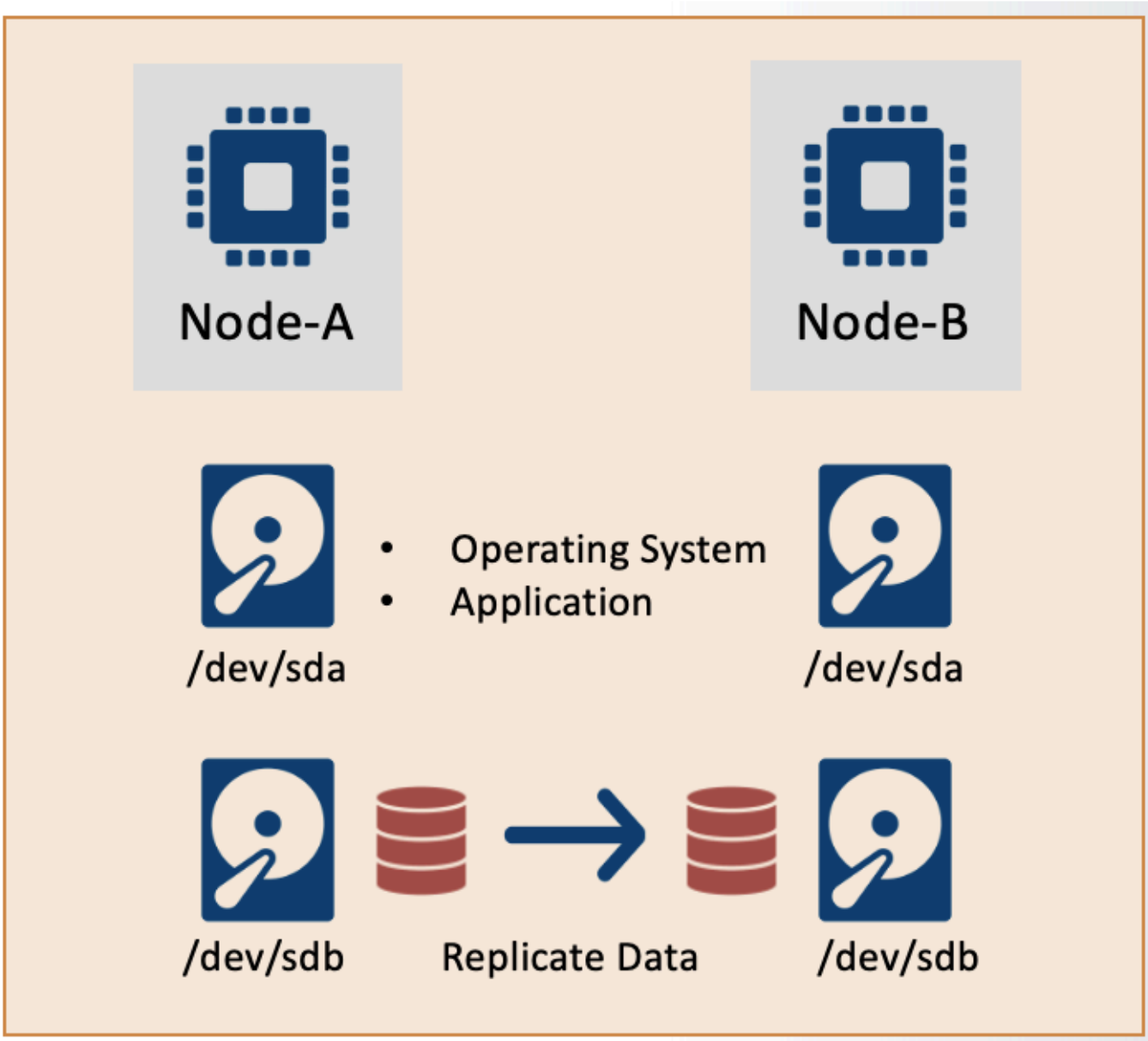
# 11.2.3.2. 本チュートリアルで使用するコンピューティングリソース

前述したように、このチュートリアルのタスクを完了するために2つのノードをネットワークに構成します。

SIOS Protection Suite for Linuxのリリースノート、および保護するリソース（Oracleなど）のリソースプランを参照してください。正しいオペレーティングシステムが選択されていること、および適切なリソース（ディスク容量、メモリ、CPUなど）が利用可能であることを確認してください。

**Note:** このガイドでは、ローカルレプリケーションストレージを使用するクラスターに焦点を当てていますが、LifeKeeperクラスターは、認定された共有ストレージソリューションを使用して構築することもできます。クラスターに共有ストレージを利用する場合は、ストレージとアダプタの設定 を参照してください。

各ノードには2つのディスクが必要であることを理解しておくことが重要です。最初のディスクはシステムの起動に使用され（これにはオペレーティングシステムとLifeKeeperを含むアプリケーションがインストールされています）、もう一方のディスクは「アクティブ」ノードから「スタンバイ」ノードにデータを複製するように構成されます（「スイッチオーバー」が発生するとデータの複製の方向が変わることに注意してください）。




**Note:** /dev/sdaなどのデバイス名は、環境によって異なる場合があります。

このガイドでは、次のリソースが各ノードに接続されていることを前提としています。

| リソース | 最低要件 | Oracleによる評価 |
|---|---|---|
| オペレーティングシステムおよびアプリケーションのインストール用のストレージデバイス | 10 Gb | 30 Gb |
| ノード間のデータレプリケーション用のストレージデバイス | 8 Gb | 20 Gb |
| メモリー | 1 Gb | 2 Gb |

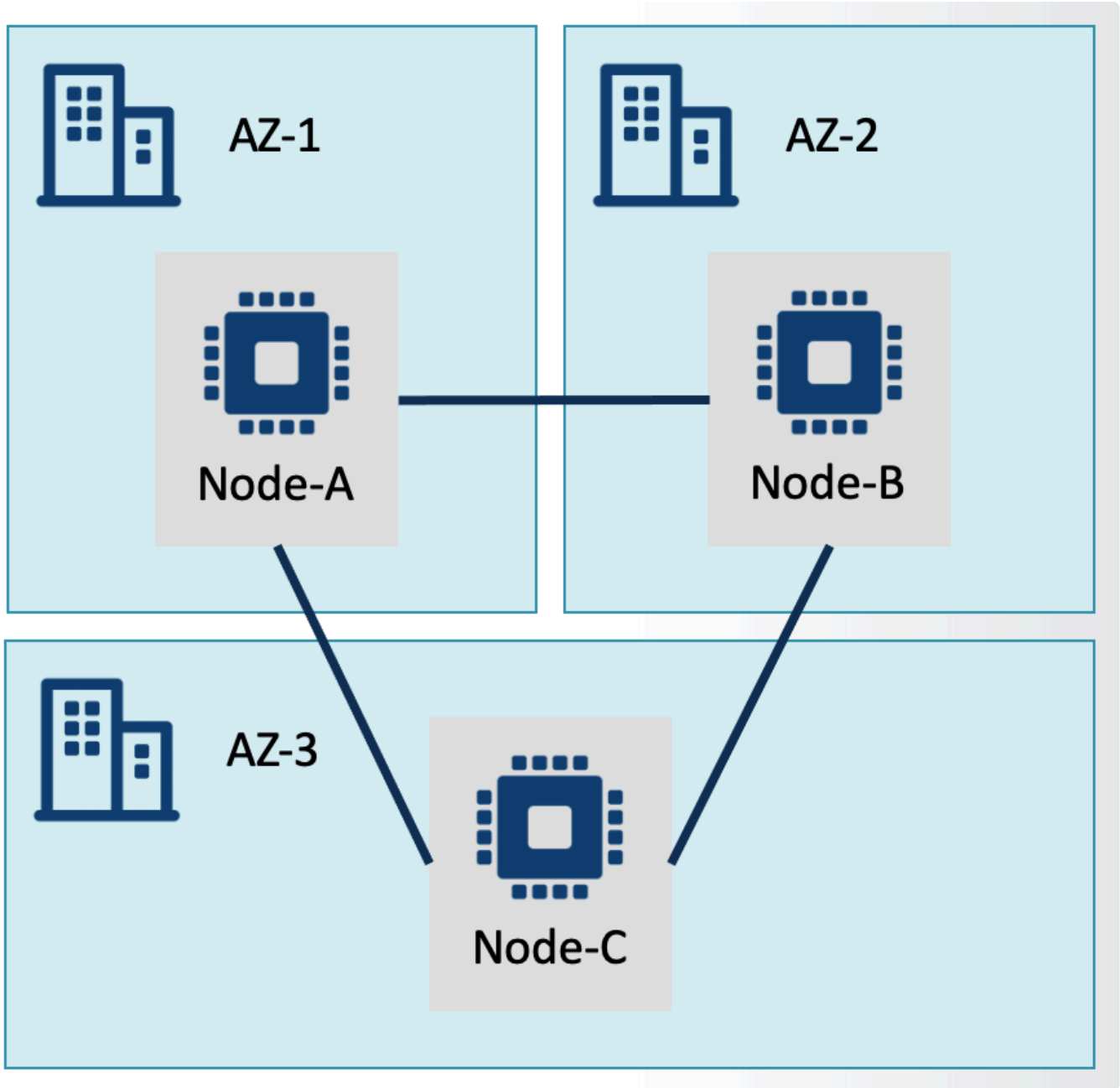
SIOSはリソースの推奨を行うことができないため、特定のアプリケーションを実行する

ために正しいリソースがプロビジョニングされていることを確認するには、アプリケーションベンダー（SAPなど）にお問い合わせください。

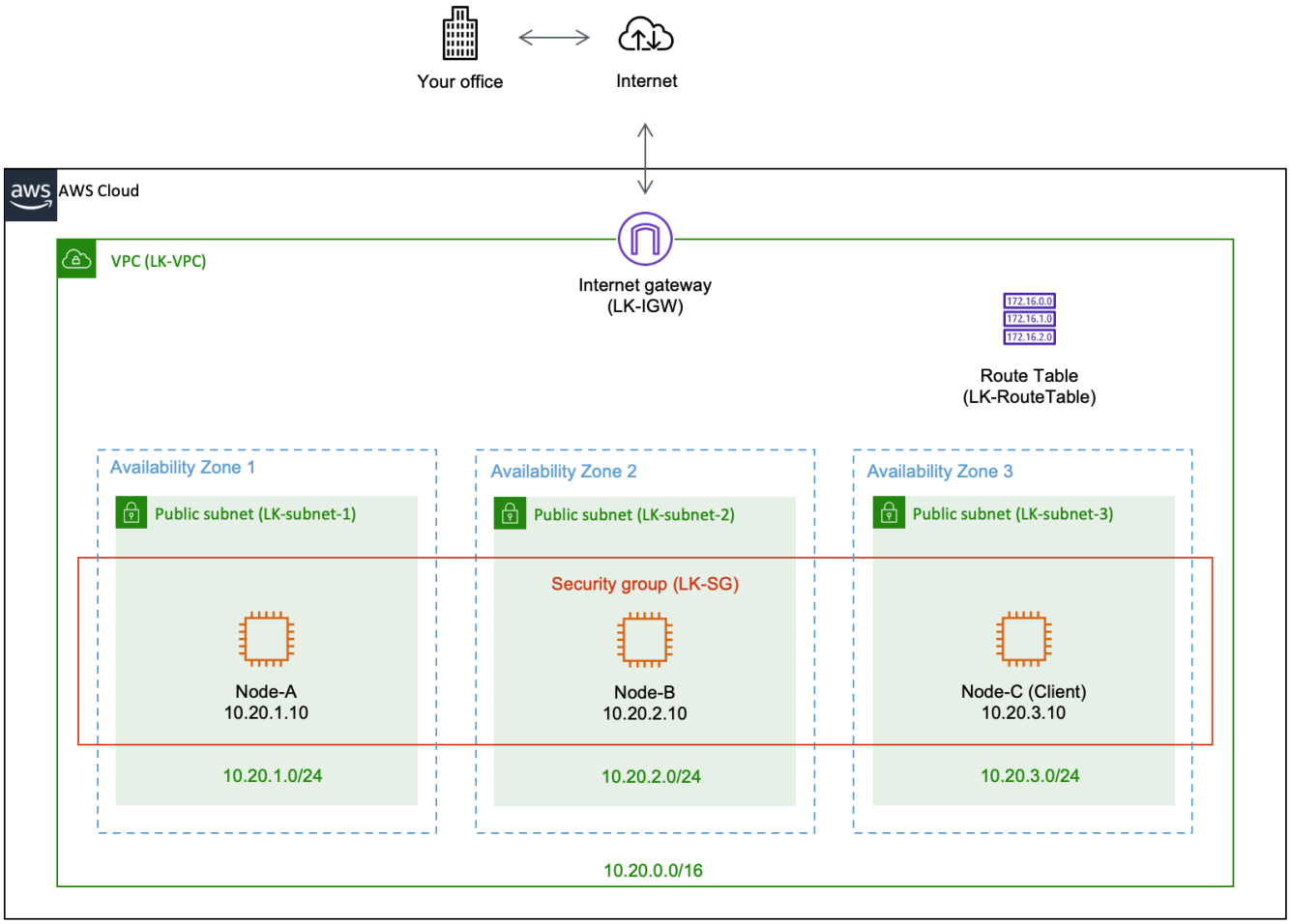
# 11.2.3.3. AWSでゼロからインスタンスを作成する

> **免責事項:** ユーザーインターフェースは地域によって異なり、また今後変更されることがあります。以下のスクリーンショットが実際の体験と異なる場合は、AWSが提供するドキュメントを参照してください。

本評価ガイドでは、以下のネットワーク構成とインスタンスを使用しています。




AWSでは、これらのコンポーネントは下図のように定義することができます。AWSで重要な点は、アベイラビリティゾーンごとにサブネットを作成する必要があることです。インスタンスは異なるアベイラビリティゾーンに分散しているため、これらのインスタンスは異なるサブネットに割り当てられます。




この図に記載されているコンポーネントは、次の表のとおりです。

| コンポーネント | 名前 | パラメータ | 値 |
|---|---|---|---|
| VPC | LK-VPC | IPv4 CIDRs | 10.20.0.0/16 |
| Subnet | **Common values across subnets** | **VPC assignment** | **LK-VPC** |
| | | **Auto-assign public IP** | **Yes** |
| | LK-subnet-1 | IPv4 CIDR | 10.20.1.0/24 |
| | LK-subnet-2 | IPv4 CIDR | 10.20.2.0/24 |
| | LK-subnet-3 | IPv4 CIDR | 10.20.3.0/24 |
| Internet Gateway | LK-IGW | VPC association | LK-VPC |
| Route Table | LK-RouteTable | Subnet association | LK-public-subnet |
| | | Destination=10.20.0.0/16 | local |
| | | Destination=0.0.0.0/0 | LK-IGW |
| Security | LK-SG | Type=All traffic | Source=LK-SG |

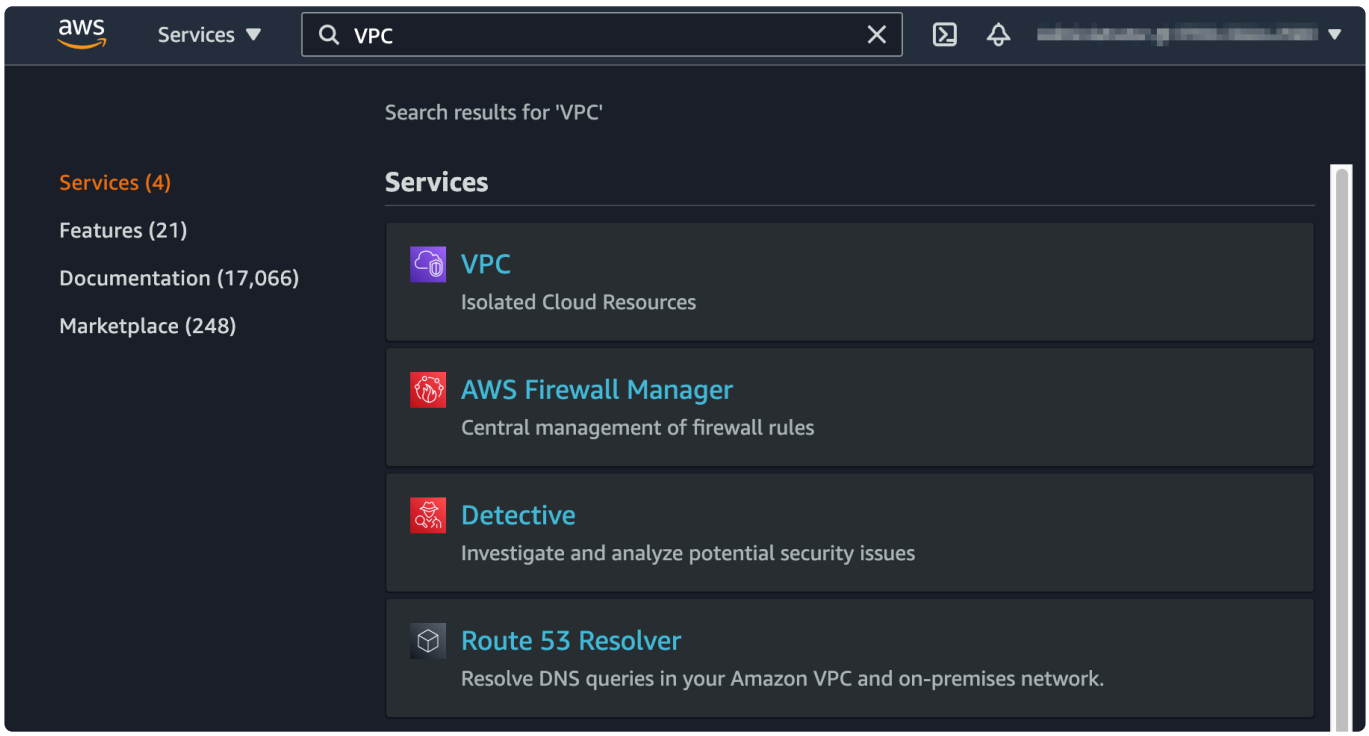
| | | | |
|---|---|---|---|
| Group | | Type=SSH | Source=Your Office's WAN IP |
| Instances | **Common values across instances** | **VPC** | **LK-VPC** |
| | | **Security Group** | **LK-SG** |
| | | **Source/Dest Checking on a network interface (ENI)** | **Disabled** |
| | node-a | Subnet | LK-subnet-1 |
| | | Private IP Address | `10.20.1.10` |
| | node-b | Subnet | LK-subnet-2 |
| | | Private IP Address | `10.20.2.10` |
| | node-c | Subnet | LK-subnet-3 |
| | | Private IP Address | `10.20.3.10` |

以下のセクションでは、これらのネットワークコンポーネントを作成するために必要なプロセスを、最初のインスタンスから順に説明します。

- AWSサービス間の切り替え
- AWSリージョンの決定
- VPCの作成
- サブネットの作成
- インターネットゲートウェイの作成とVPCへの割り当て
- ルートテーブルの作成
- セキュリティグループの作成
- 最初のEC2インスタンスの作成
- 2番目と3番目のインスタンスの作成

# 11.2.3.3.1. AWSサービス間の切り替え

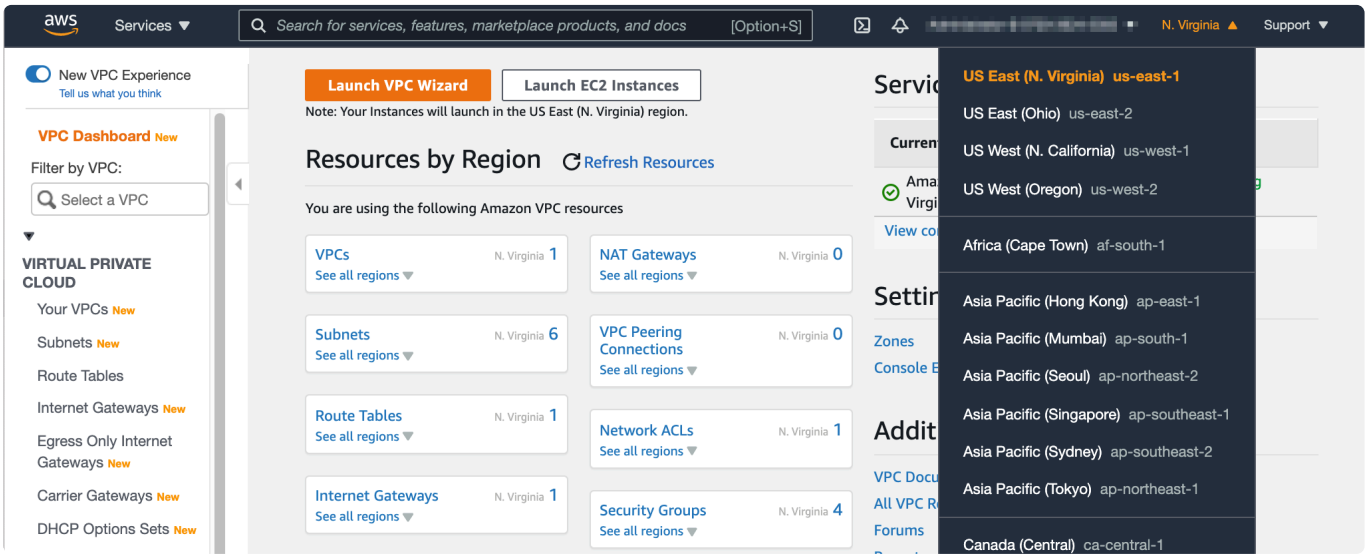
AWSではかなりの数のサービスを利用することができ、その数は増え続けています。別のサービスに移動するには、AWSコンソールの上部にある検索ボックスにサービス名を入力し、一覧からサービスを選択します。下のスクリーンショットでは、VPC（仮想プライベートクラウド）サービスを選択候補として表示しています。



# 11.2.3.3.2. AWSリージョンの決定

AWSには、地理的に多くの場所にリージョンがあります。職場から地理的に近いリージョンを選択することが有益な場合があります。すべてのリージョンに3つ以上のアベイラビリティゾーンがあるわけではないため、選択したリージョンが意図している構成をサポートするように、選択を行う際には注意が必要です。サービスメニューからVPCを選択すると、現在のリージョンがAWSコンソールの右上に表示されます。

詳細については、AWSのドキュメント を参照してください。



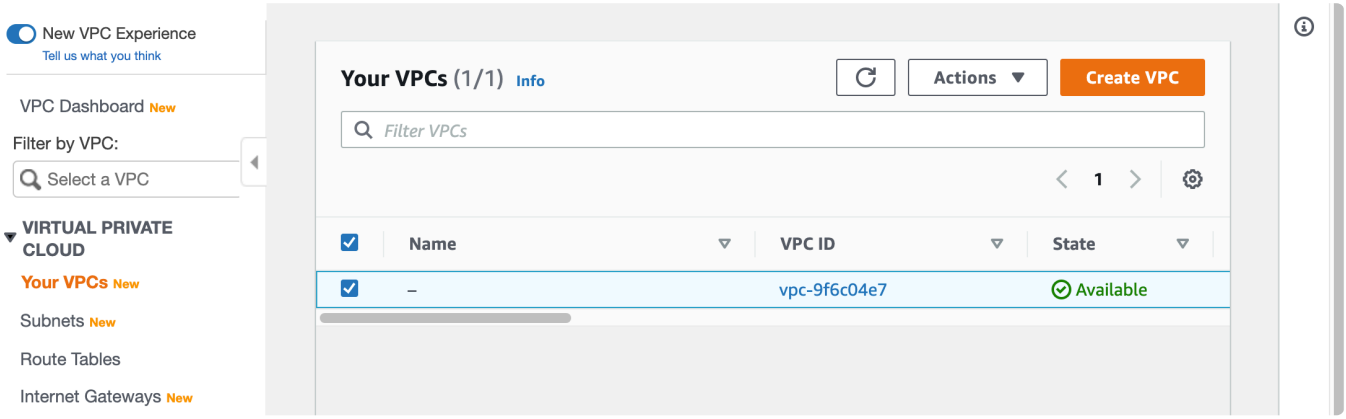
# 11.2.3.3.3. VPCの作成

VPC（Virtual Private Cloud）とは、ローカルネットワークを表すAWSリソースのことです。AWSクラウド内で異なるVPCを定義することで、異なるシステムを論理的に分離することができます。

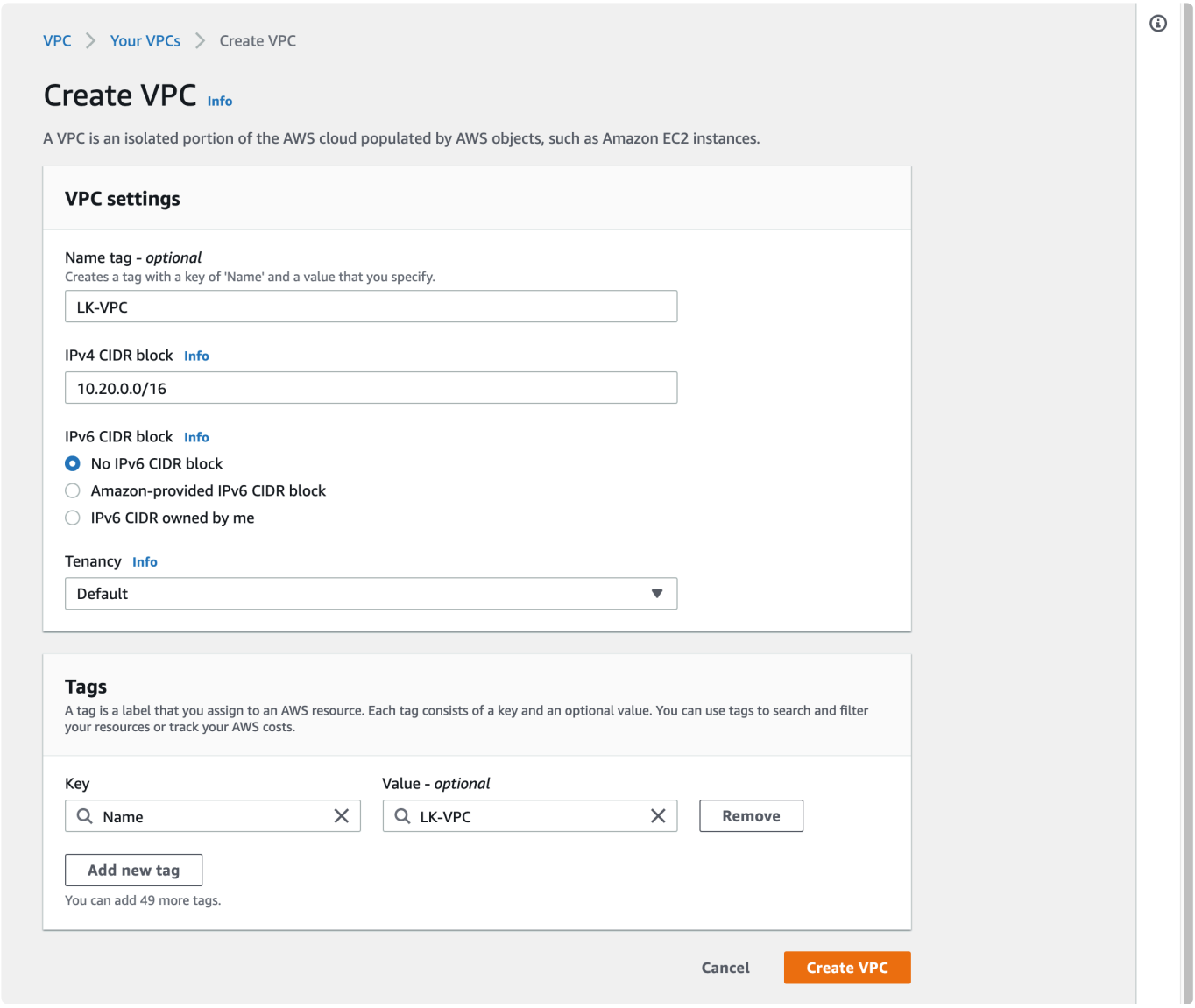
詳細については、AWSのドキュメント を参照してください。
このセクションでは、以下のようにLifeKeeperのテスト用VPC（LK-VPCと命名）を作成します。

1. コンソールの左側にあるナビゲーションペインから [Your VPCs] を選択します。

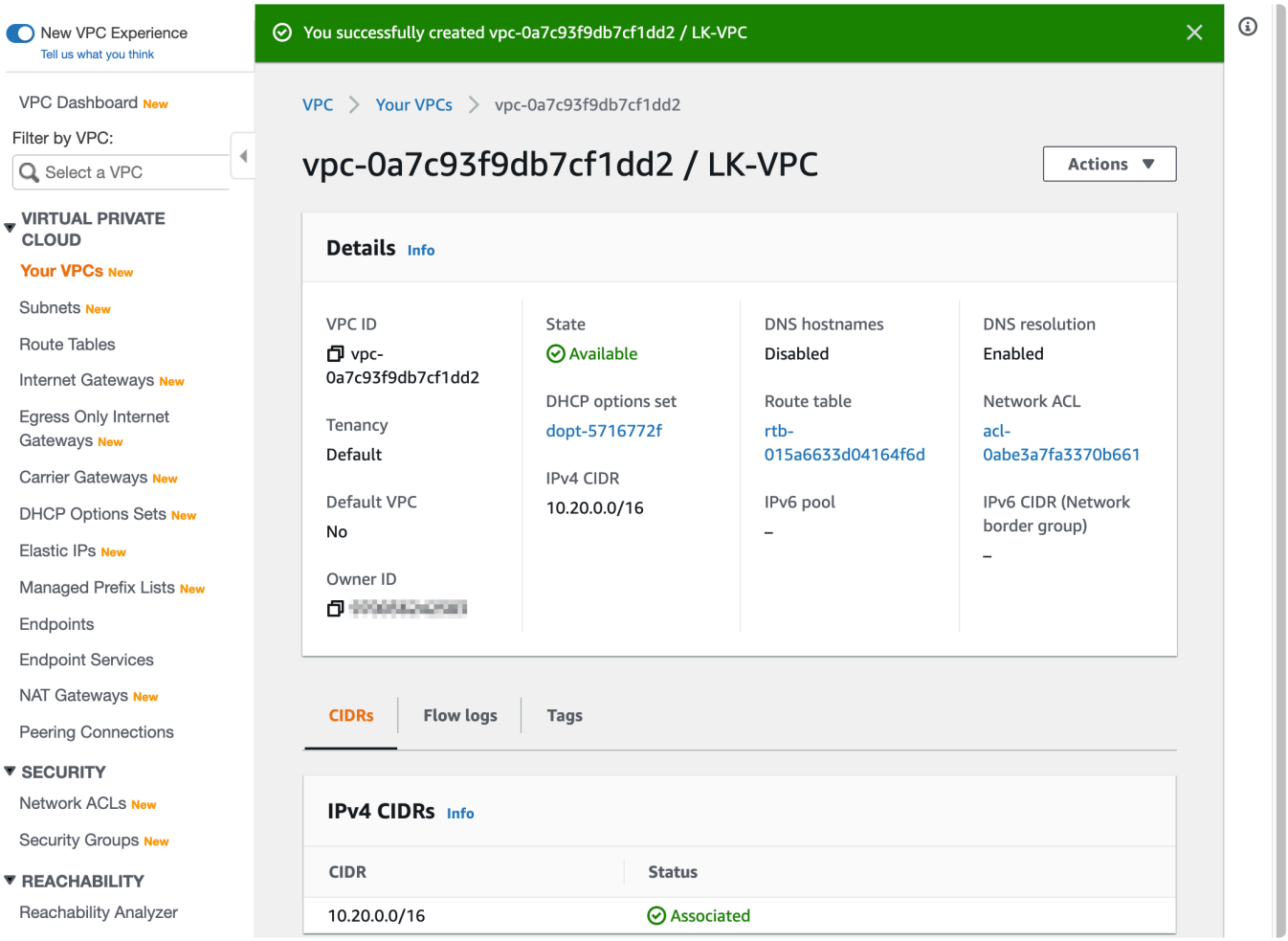
2. AWSが作成したVPC（デフォルトのVPC）がありますが、今回はこの評価ガイドで定義されている仕様に基づき、カスタムVPCを作成します。右上の [Create VPC] を選択します。




3. 「Create VPC」ウィザードで、VPCの名前（LK-VPC）とIPv4 CIDRブロック10.20.0.0/16を入力します。




4. VPC `LK-VPC` が指定されたパラメーターで作成されます。


New VPC Experience
Tell us what you think
VPC Dashboard New
Filter by VPC:
Select a VPC
VIRTUAL PRIVATE CLOUD
Your VPCs New
Subnets New
Route Tables
Internet Gateways New
Egress Only Internet Gateways New
Carrier Gateways New
DHCP Options Sets New
Elastic IPs New
Managed Prefix Lists New
Endpoints
Endpoint Services
NAT Gateways New
Peering Connections
SECURITY
Network ACLs New
Security Groups New
REACHABILITY
Reachability Analyzer
You successfully created vpc-0a7c93f9db7cf1dd2 / LK-VPC
VPC > Your VPCs > vpc-0a7c93f9db7cf1dd2
vpc-0a7c93f9db7cf1dd2 / LK-VPC
Actions
Details Info
VPC ID
vpc-0a7c93f9db7cf1dd2
Tenancy
Default
Default VPC
No
Owner ID
State
Available
DHCP options set
dopt-5716772f
IPv4 CIDR
10.20.0.0/16
DNS hostnames
Disabled
Route table
rtb-015a6633d04164f6d
IPv6 pool
–
DNS resolution
Enabled
Network ACL
acl-0abe3a7fa3370b661
IPv6 CIDR (Network border group)
–
CIDRs
Flow logs
Tags
IPv4 CIDRs Info
CIDR
Status
10.20.0.0/16
Associated

# 11.2.3.3.4. サブネットの作成

AWSにおけるサブネットの基本的な考え方は、オンプレミス環境の場合と同じです。つまり、ネットワークの中のネットワークです。コンポーネント間の論理的または物理的な境界を作成するために、別のサブネットを定義することができます。AWS独自の概念は以下の通りです。

- システムが異なるアベイラビリティゾーン（異なるデータセンター）にまたがって展開されている場合、アベイラビリティゾーンごとに個別のサブネットを定義する必要があります。

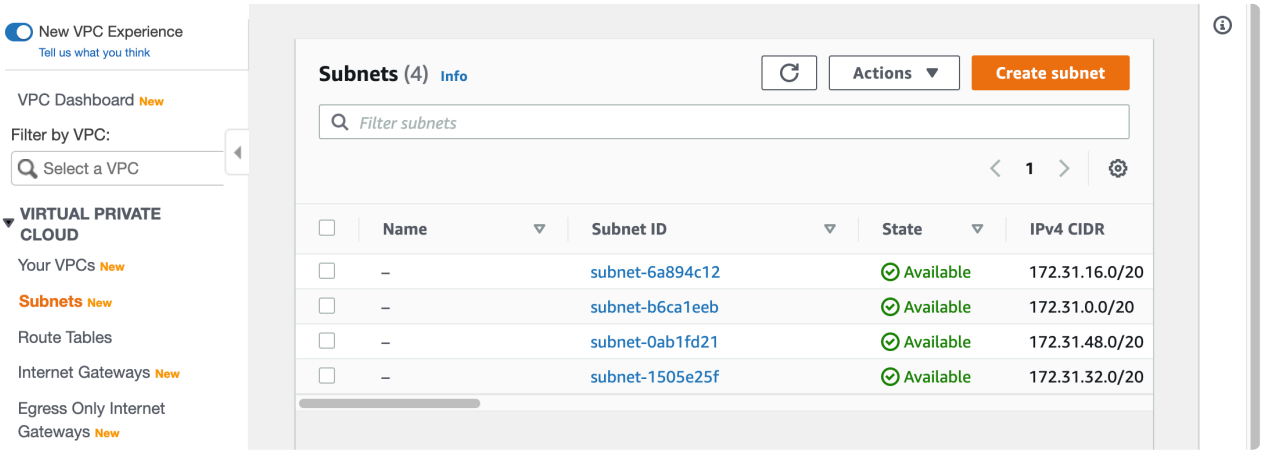
- サブネット内のインスタンスがインターネットから/インターネットへの直接のアクセスを必要とする場合、「パブリック」サブネットとして構成する必要があります。ただし、インスタンスをインターネットから分離する必要がある場合は、「プライベート」サブネットとして構成する必要があります。そのため、アベイラビリティゾーンごとに2つ以上のサブネットを定義する必要がある場合があります。

  詳細については、AWSのドキュメント を参照してください。

このチュートリアルでは、3つのアベイラビリティゾーンを使用し、プロセスを簡略化するために「public」パターンのみを使用します。「public」サブネットは、次のように作成します。

1. 左側のナビゲーションペインから [Subnet] を選択して、[Subnet] ページに移動します。

2. VPCに従って、デフォルトのVPCにはあらかじめサブネットが設定されています。右上の [Create Subnet] を選択します。




3. スクリーンショットに示すように、以下のパラメーターを指定します。

- VPC ID: `LK-VPC`

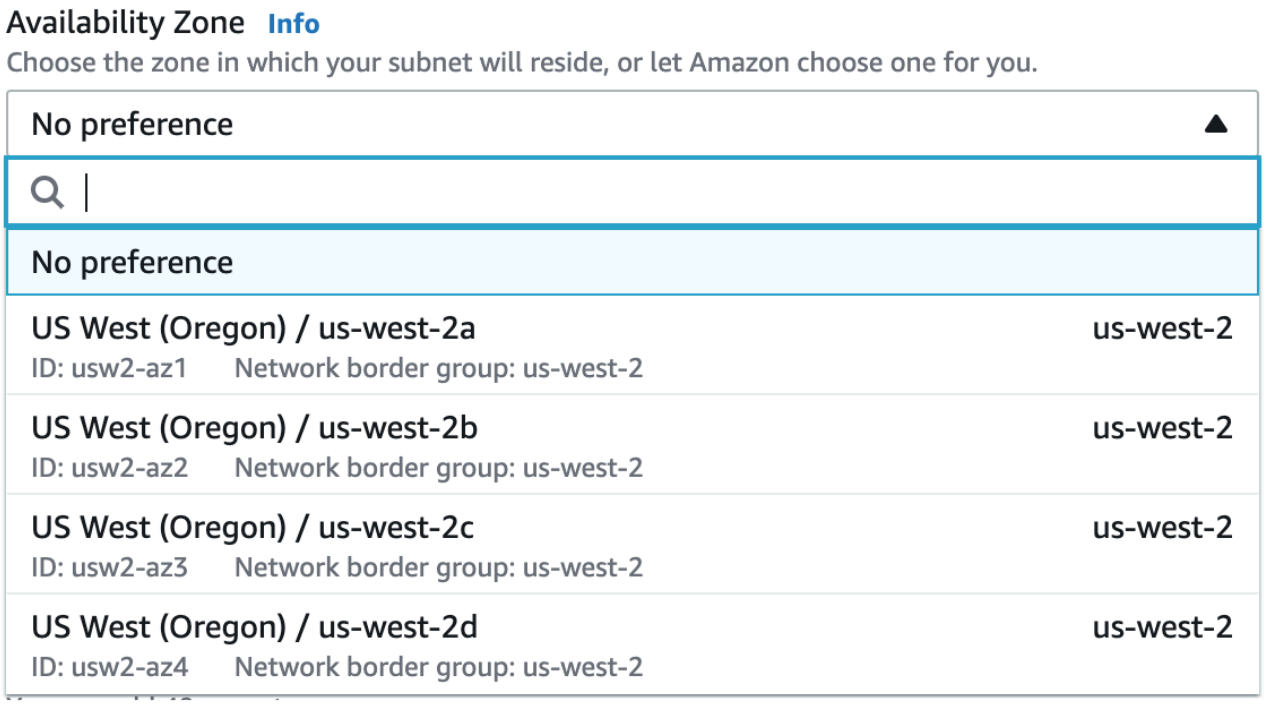
- Subnet Name: `LK-subnet-1`

- Availability Zone: Select the first option.

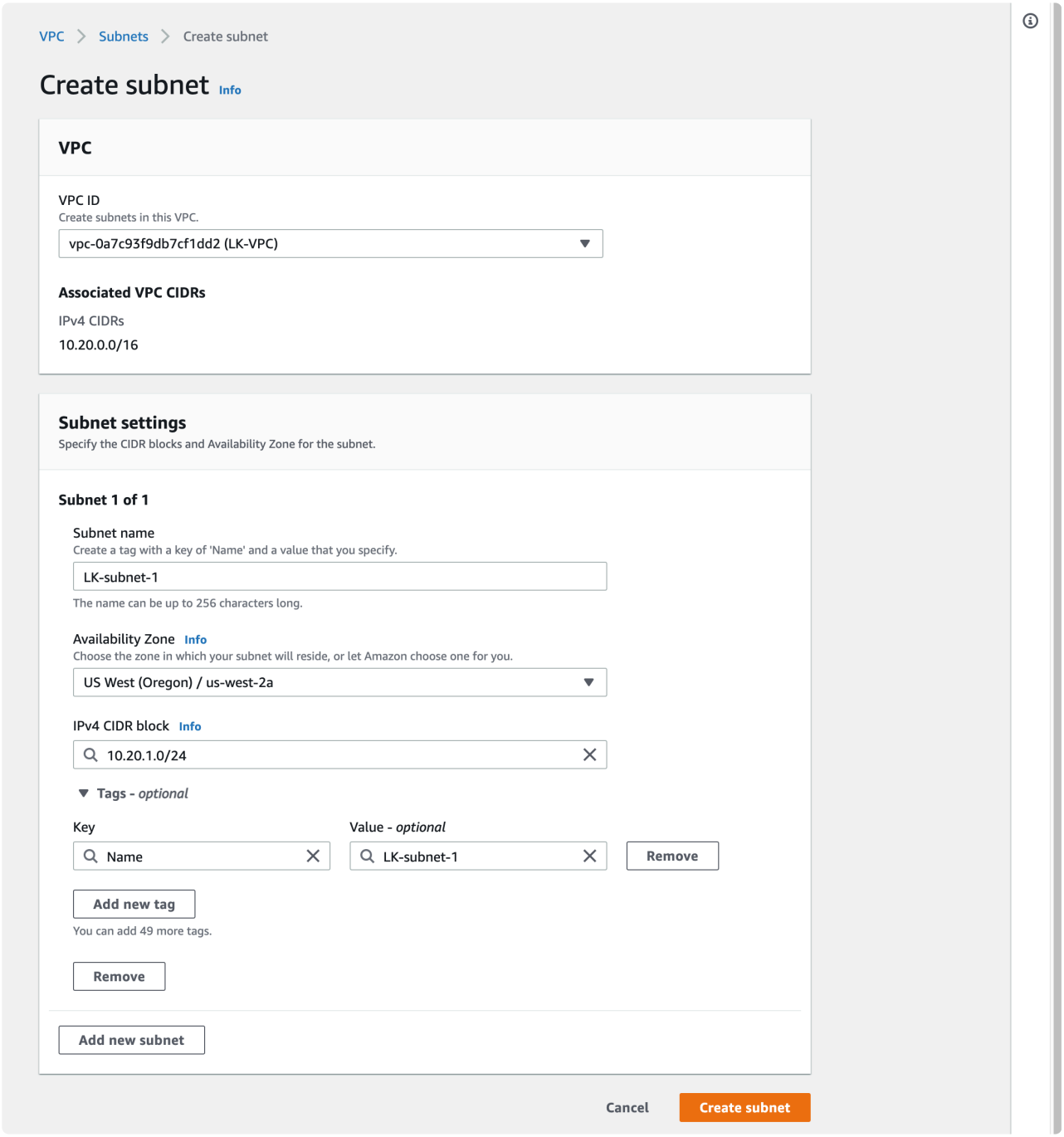
- IPv4 CIDR block: `10.20.1.0/24`

- Tag: Name = `LK-subnet-1`

このウィザードを使用している間、アベイラビリティゾーンのドロップダウンボックスを選択すると、アベイラビリティゾーンのリストが表示されます。




最初のアベイラビリティゾーンを選択します（この場合は`us-west-2a`）

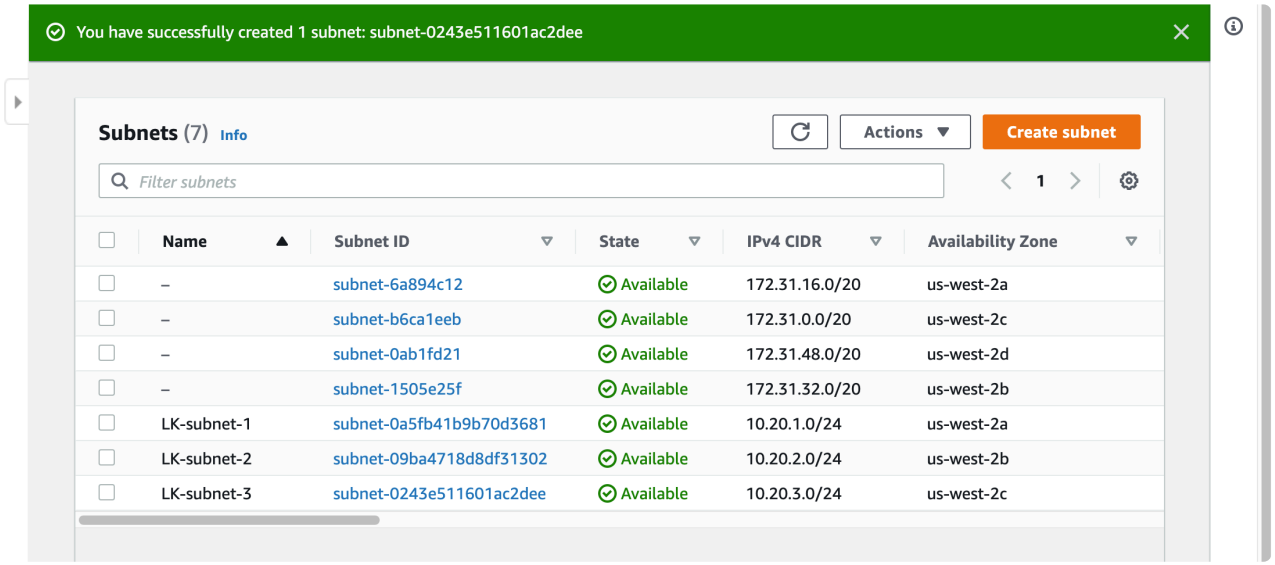



4. これで、アベイラビリティーゾーン `us-west-2a` の最初のサブネットが作成されました。「サブネットページ」に戻ります。

5. 別のアベイラビリティーゾーン `us-west-2b` にCIDRブロック10.20.2.0/24 を使用して2番目のサブネット `LK-subnet-2` を作成します。

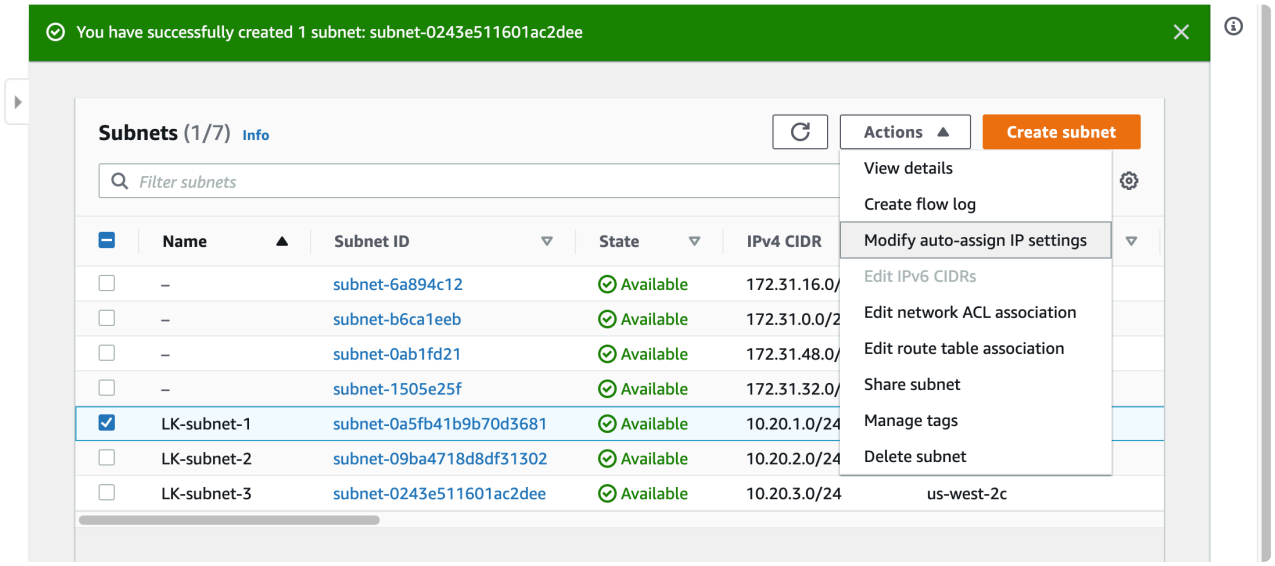
6. アベイラビリティーゾーン `us-west-2c`にCIDRブロック10.20.3.0/24を使用して3番目のサブネットLK-subnet-3を作成します。

7. 3つのサブネットが作成されると、サブネットのリストは次のスクリーンショットのようになります。画像の最初のサブネットは、「デフォルト」VPC用に作成された「デフォルト」サブネットであり、その後に今作成した3つのサブネットが表示されます。

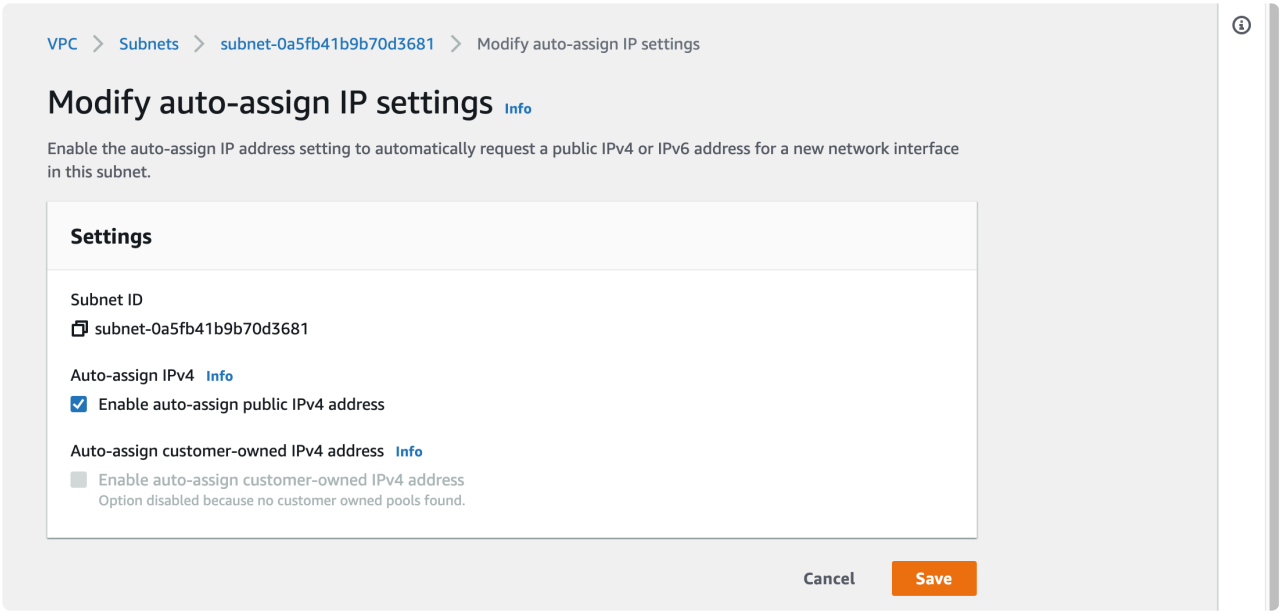



8. このチュートリアルでは、パブリックサブネットを作成します。これは、このサブネットの下に作成するインスタンスにパブリックIPアドレスを割り当てることを意味します。これを行うには、先ほど作成したサブネットを選択し、「アクション」メニューから「自動割り当てIP設定の変更」を選択します。

**Note:** この変更は、3つのサブネットすべてに対して1つずつ行う必要があります。




9. 「パブリックIPv4アドレスの自動割り当てを有効にする」を選択し、変更を保存します。




10. これでサブネットが使えるようになりました。

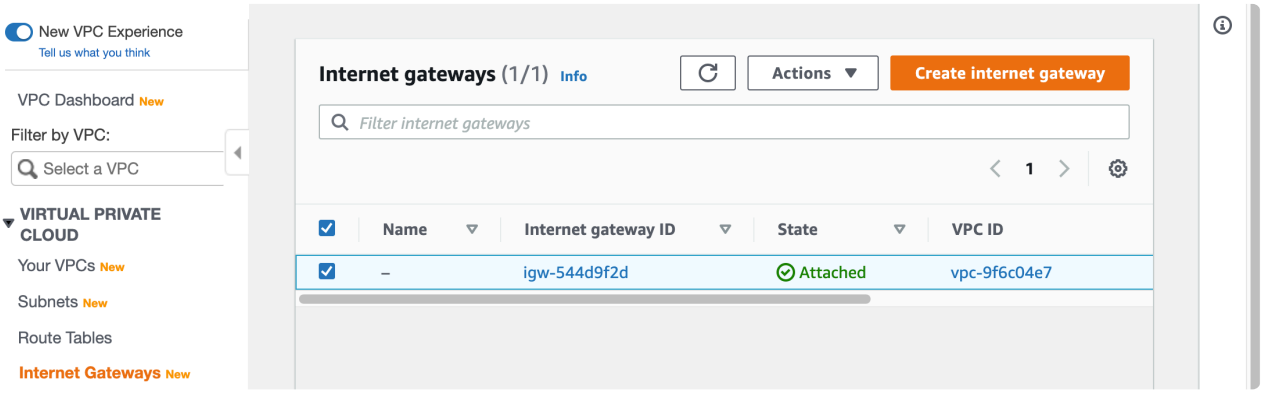
# 11.2.3.3.5. インターネットゲートウェイの作成とVPCへの割り当て

次に、インターネットゲートウェイを作成します。インターネットゲートウェイは、VPCとインターネット間の通信を可能にするVPCコンポーネントです。

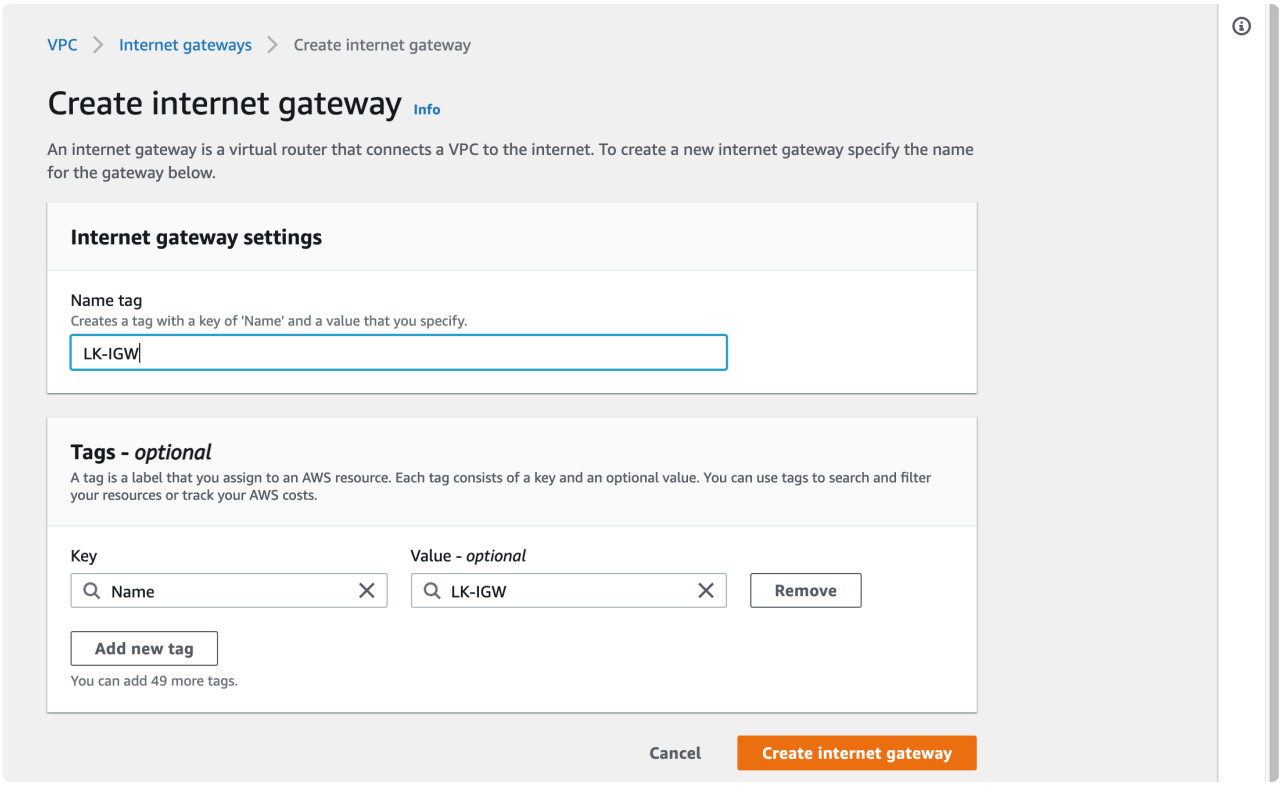
詳細については、AWSのドキュメント を参照してください。

インターネットゲートウェイを作成してVPCに関連付けましょう。

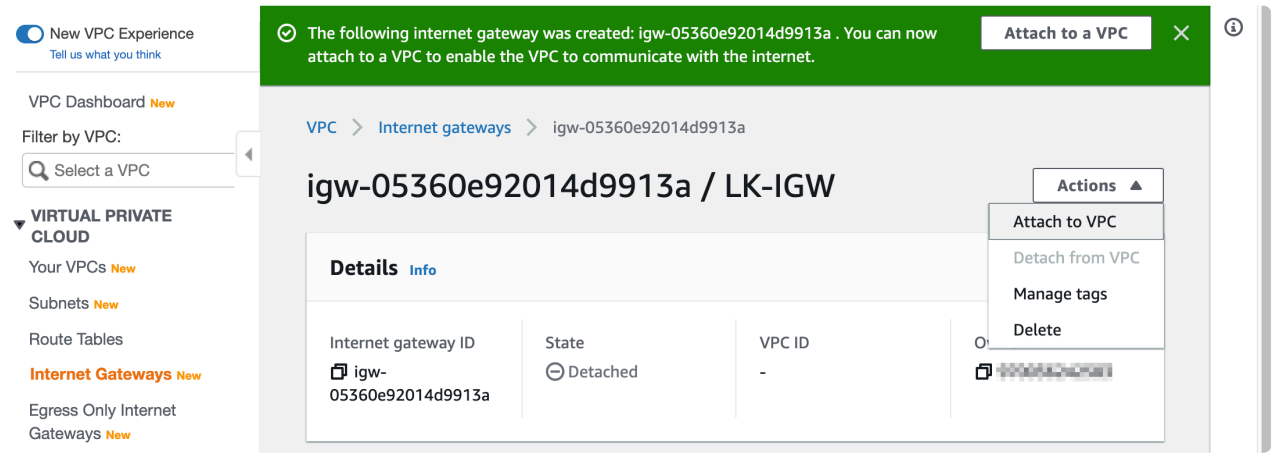
1. 左側のナビゲーションペインから「インターネットゲートウェイ」を選択します。
2. デフォルトVPCの一部として、既存のインターネットゲートウェイが表示される場合があります。右上の「インターネットゲートウェイの作成」を選択します。



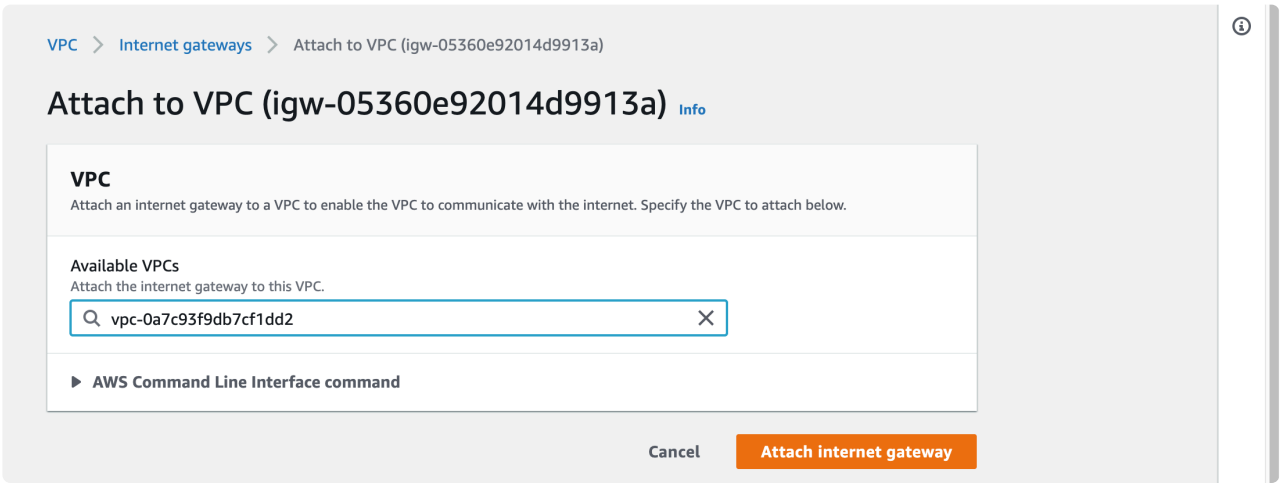

3. インターネットゲートウェイの名称を以下のように指定します。



4. インターネットゲートウェイが作成されたら、「VPCにアタッチ」を選択します。.




5. 関連付けるVPCとして LK-VPC を選択します。




6. インターネットゲートウェイが構成されました。

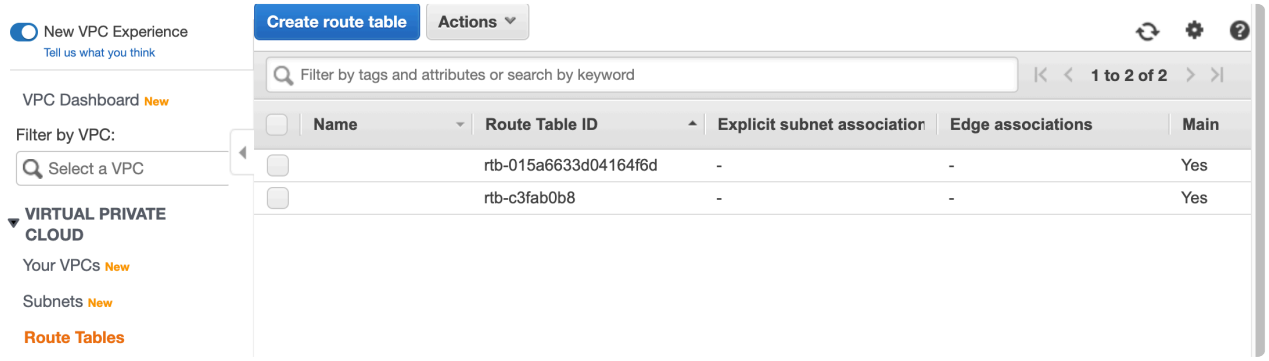
# 11.2.3.3.6. ルートテーブルの作成

ルートテーブルは、サブネット内のインスタンスからのトラフィックを転送する方法を定義します。
詳細については、AWSのドキュメント を参照してください。

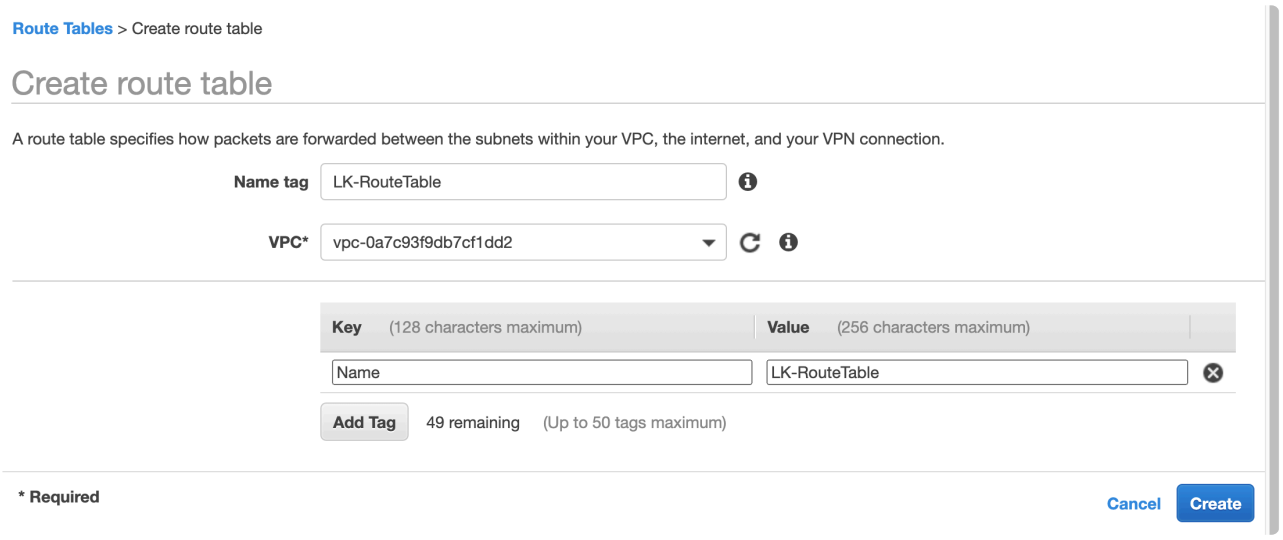
このセクションでは、ルートテーブルを作成し、インターネットゲートウェイを経由してインターネットに接続するルートを追加します。

1. 左側のナビゲーションペインから「ルートテーブル」を選択します。

2. 左上の「ルートテーブルの作成」を選択します。

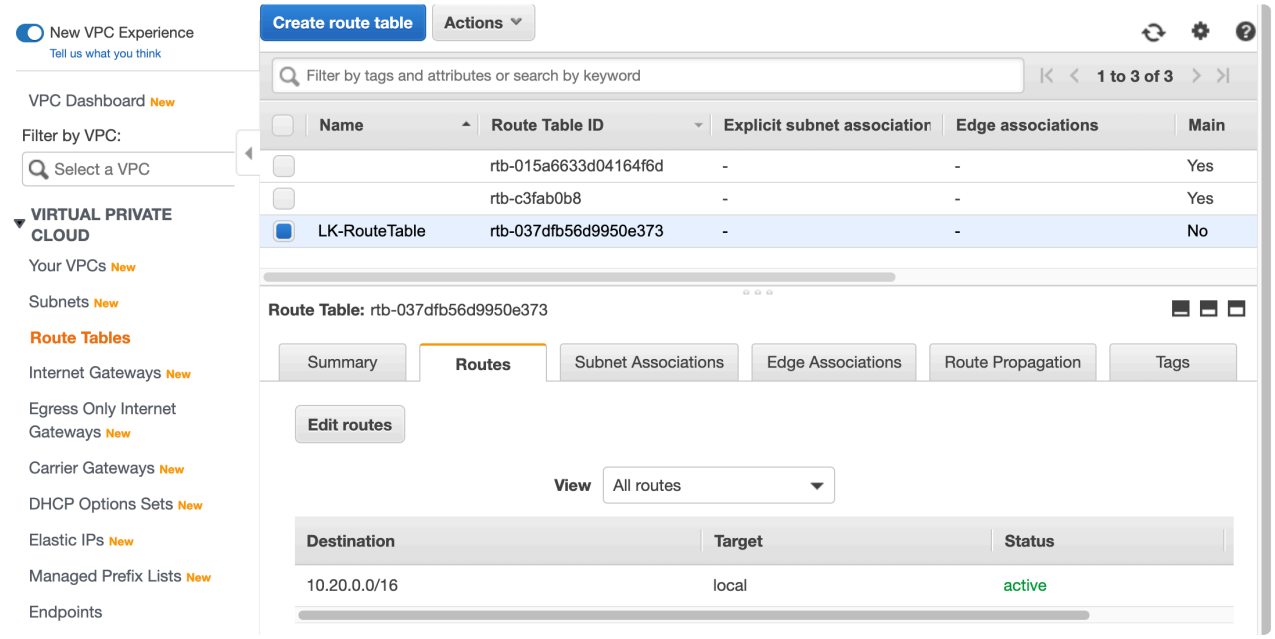



3. このルートテーブルの名前を LK-RouteTable として設定します。また、「名前」タグを定義します。 LK-VPC が関連するVPCとして選択されていることを確認してください。

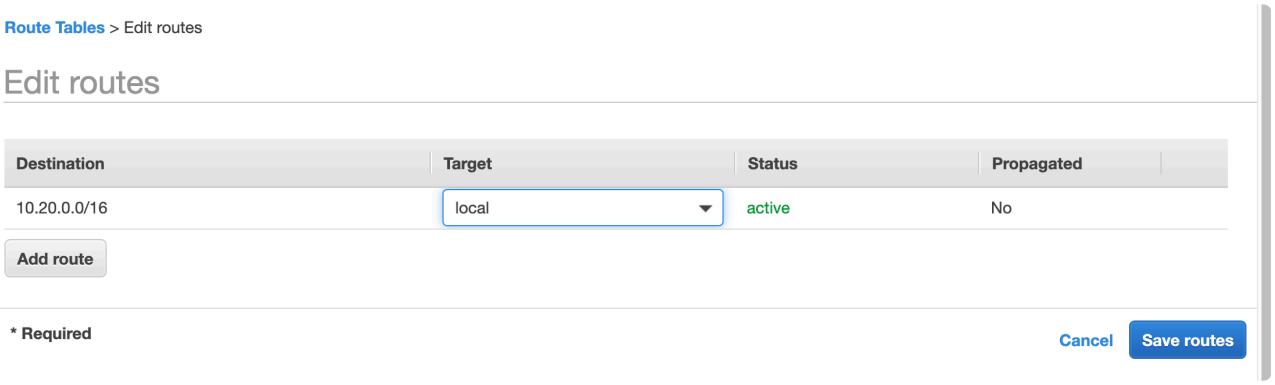



これらのフィールドを定義したら、「作成」を選択します。

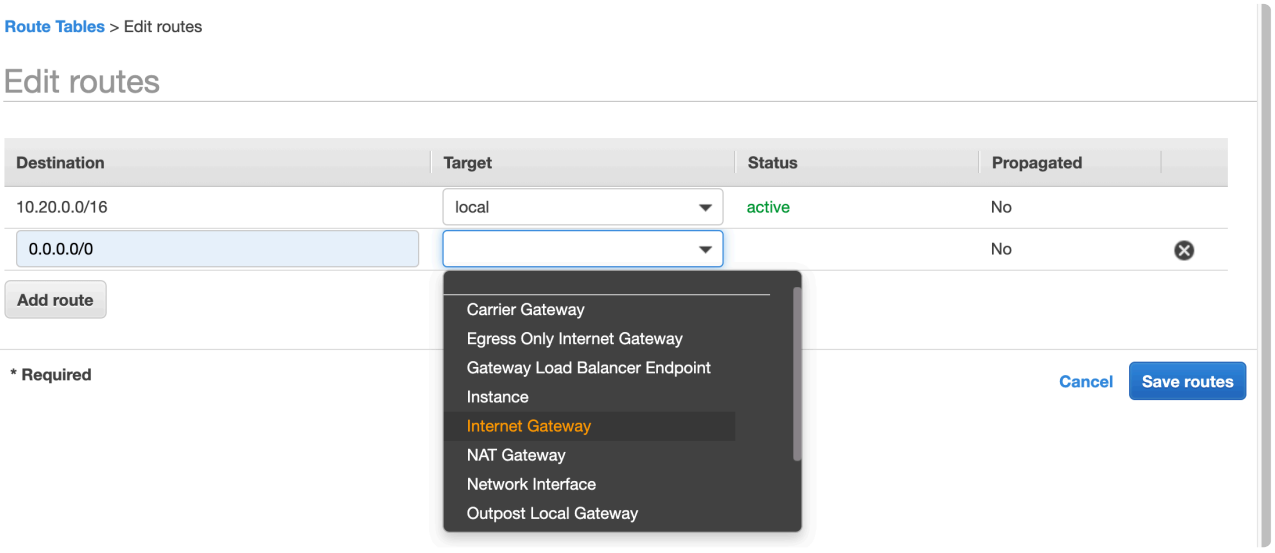
4. 「ルート」ページに移動し、「ルートの編集」をクリックします。



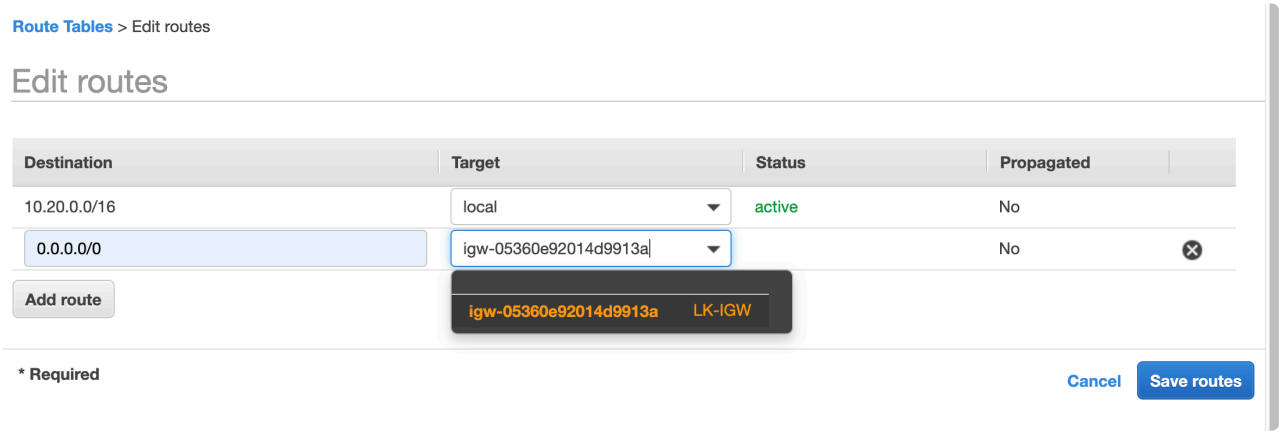

5. ローカルトラフィックはすでに定義されています。 次に、「ルートの追加」をクリックします。




6. 宛先として0.0.0.0/0（任意のIPv4アドレス、つまりインターネット）を入力し、ターゲットとして「インターネットゲートウェイ」を選択します。

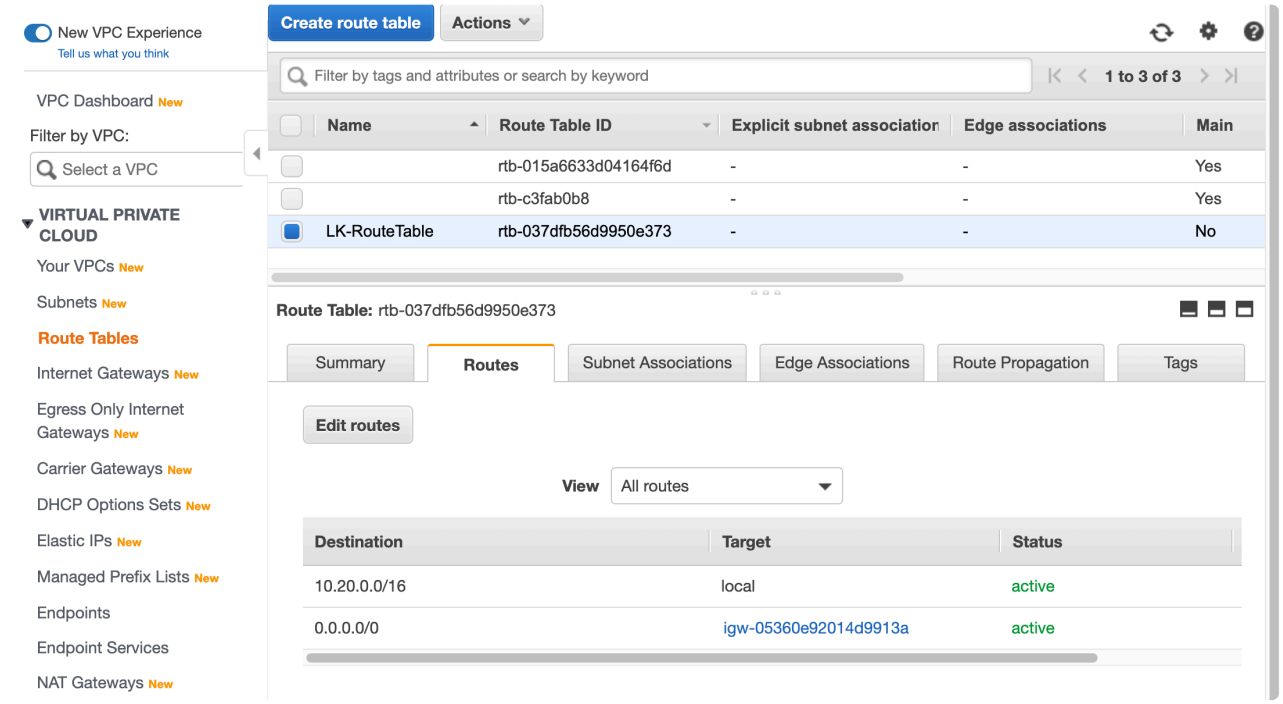



7. インターネットゲートウェイとして「LK-IGW」を選択します。

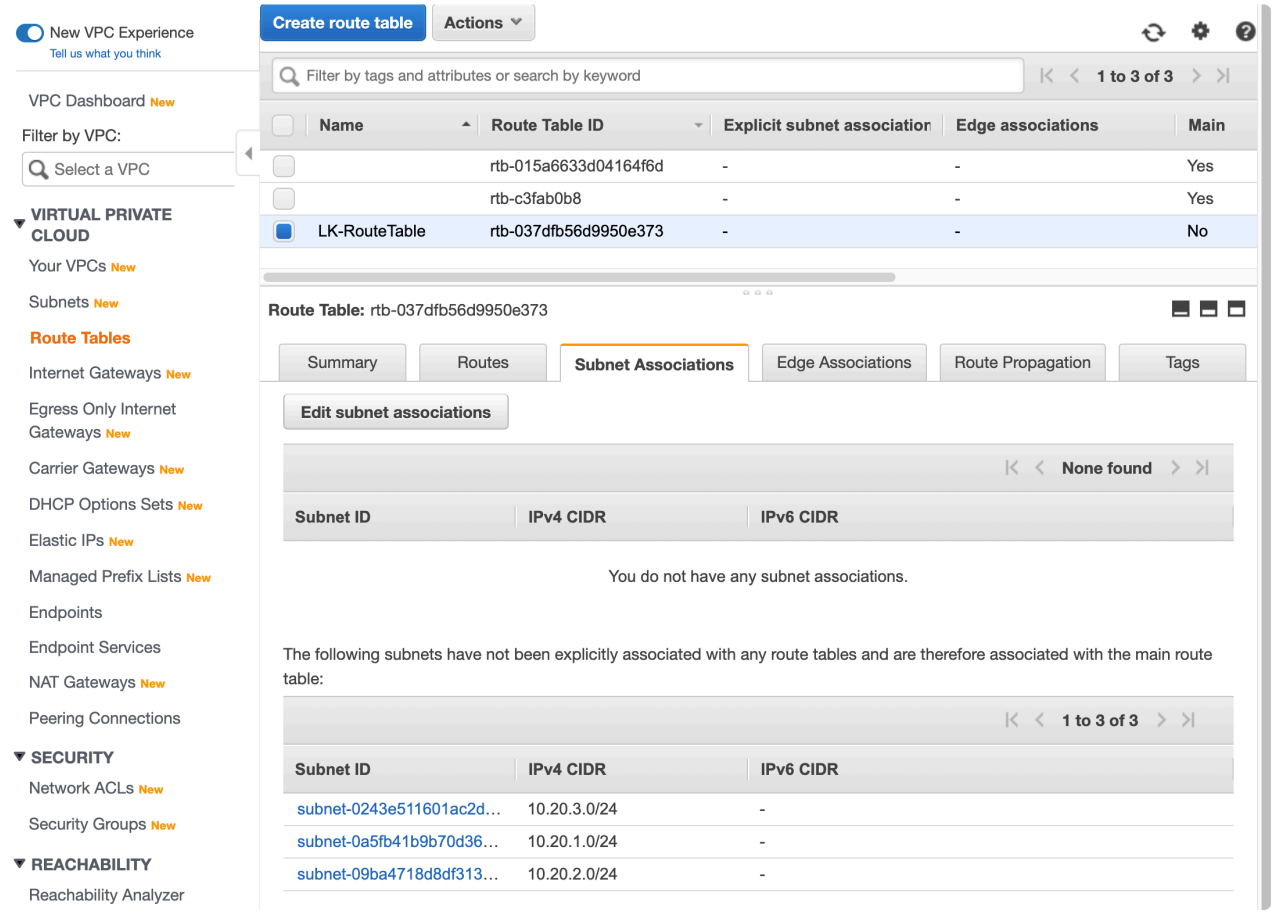



ルートが0.0.0.0/0に定義されたら、「ルートの保存」をクリックしてウィザードを閉じます。

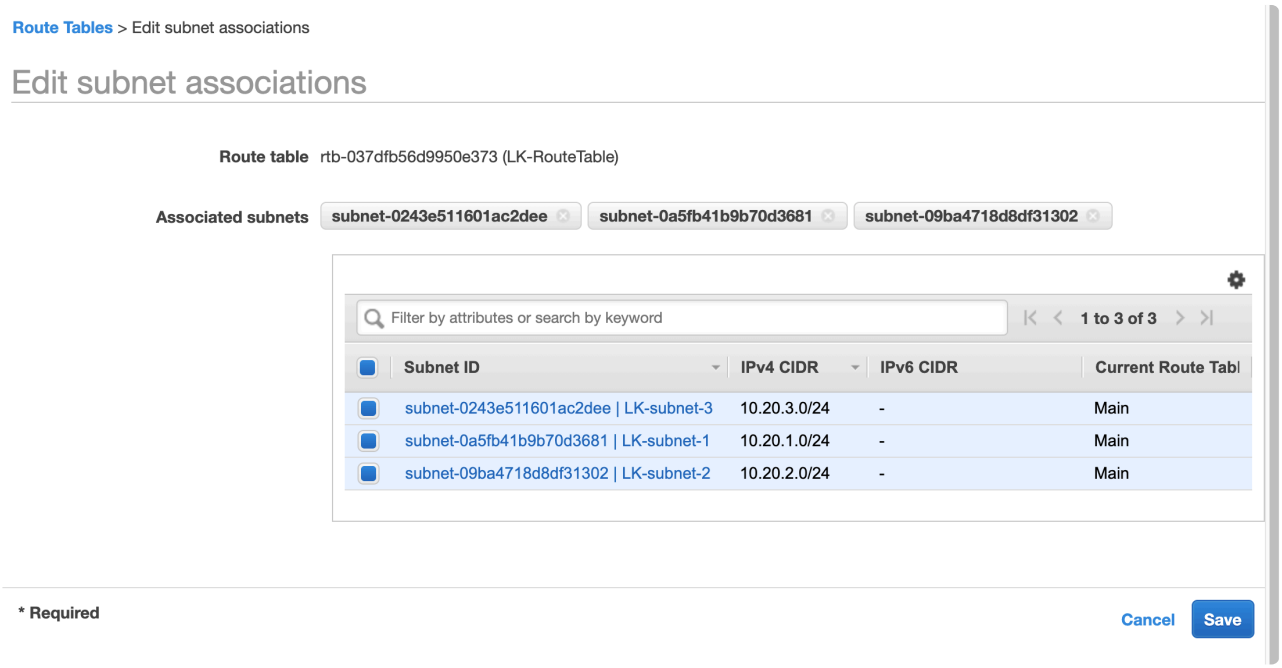
8. これでルートが定義されました。




9. 次に、ルートテーブルと新しく作成したサブネット（LK-subnet-1、LK-subnet-2、LK-subnet-3）の関連付けを行います。サブネットの関連付け」タブを選択します。




10. 「サブネットの関連付けの編集」をクリックし、「LK-subnet-1」、「LK-subnet-2」、および「LK-subnet-3」を選択します。




これで、新しいルートテーブル（「LK-RouteTable」）が定義され、LK-subnet-1/2/3サブネットに関連付けられました。

# 11.2.3.3.7. セキュリティグループの作成

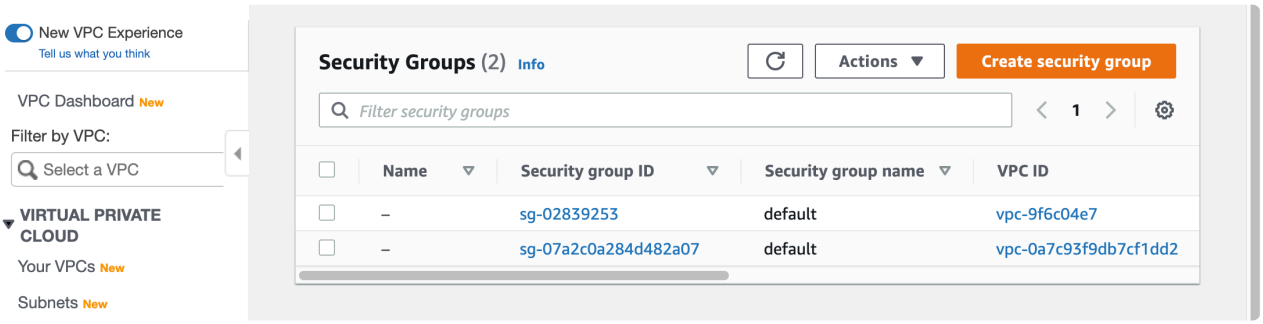
セキュリティグループはファイアーウォールとして機能し、各ソースから「許可」ルールを定義することができます。ソースにはインターネットアドレスやセキュリティグループを指定することができ、セキュリティグループはElastic Network Interfaceに割り当てることができます。ネットワークインターフェースはインスタンスに接続されるため、各インスタンスのファイアーウォールルールとして機能します。

詳細については、 AWSのドキュメント を参照してください。
このセクションでは、次の2つの「許可」ルールを定義します。

- リモートワークの場所からのアクセスを許可する。
- インスタンス（ノード-aとノード-b）間のアクセスを許可する。これは、ソースと同じセキュリティグループを許可してから、両方のインスタンスに同じセキュリティグループを割り当てることによって行われます。

**Note:** 本番環境が定義されると、インスタンスは異なるセキュリティグループに属する可能性があります。この場合、セキュリティグループ間で対応する「許可」ルールを定義する必要があります。

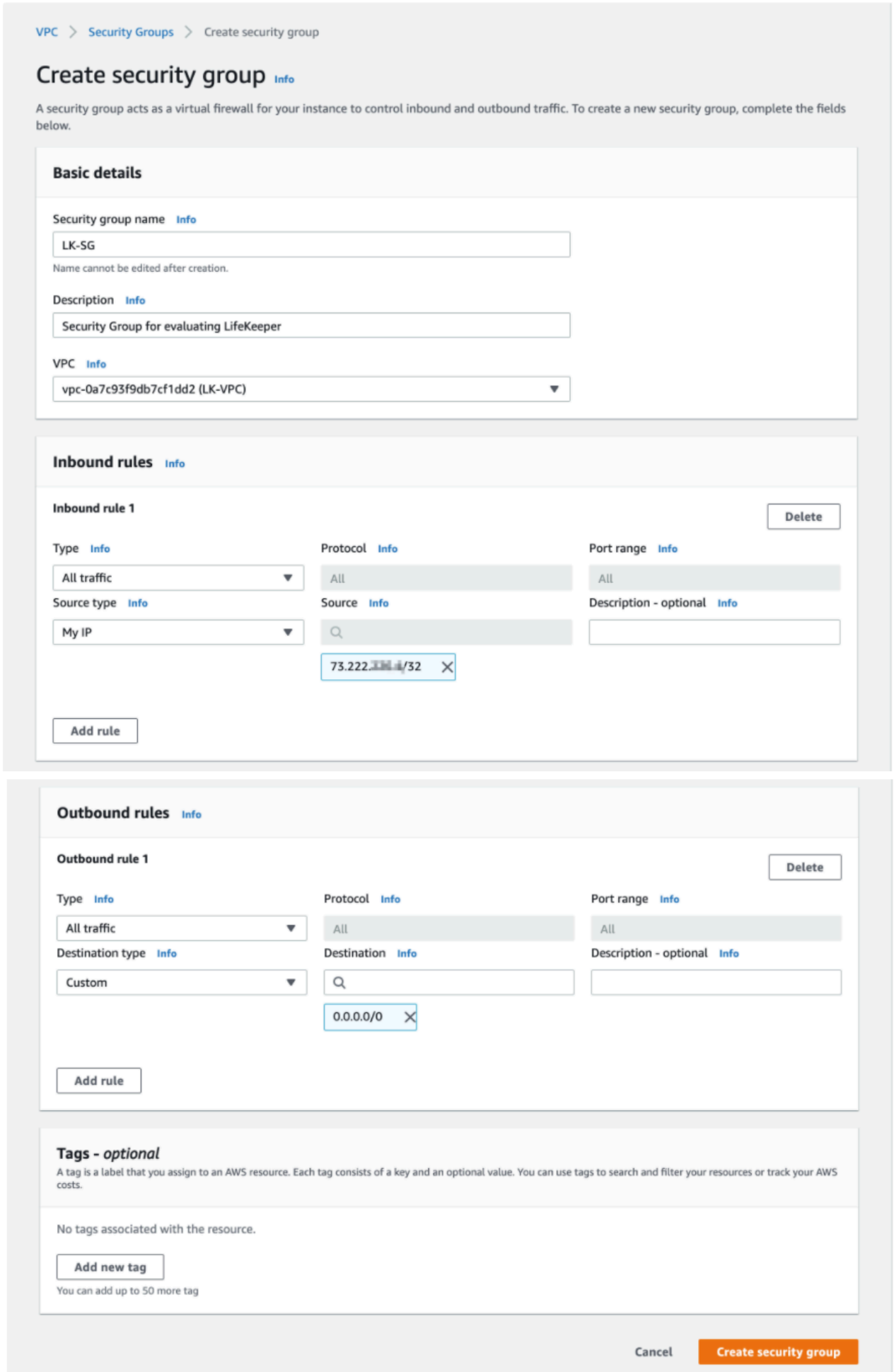
1. 左側のナビゲーションペインから「Security Groups」を選択します。前述のとおり、デフォルトのセキュリティグループがすでに存在します。セキュリティグループの作成」をクリックします。



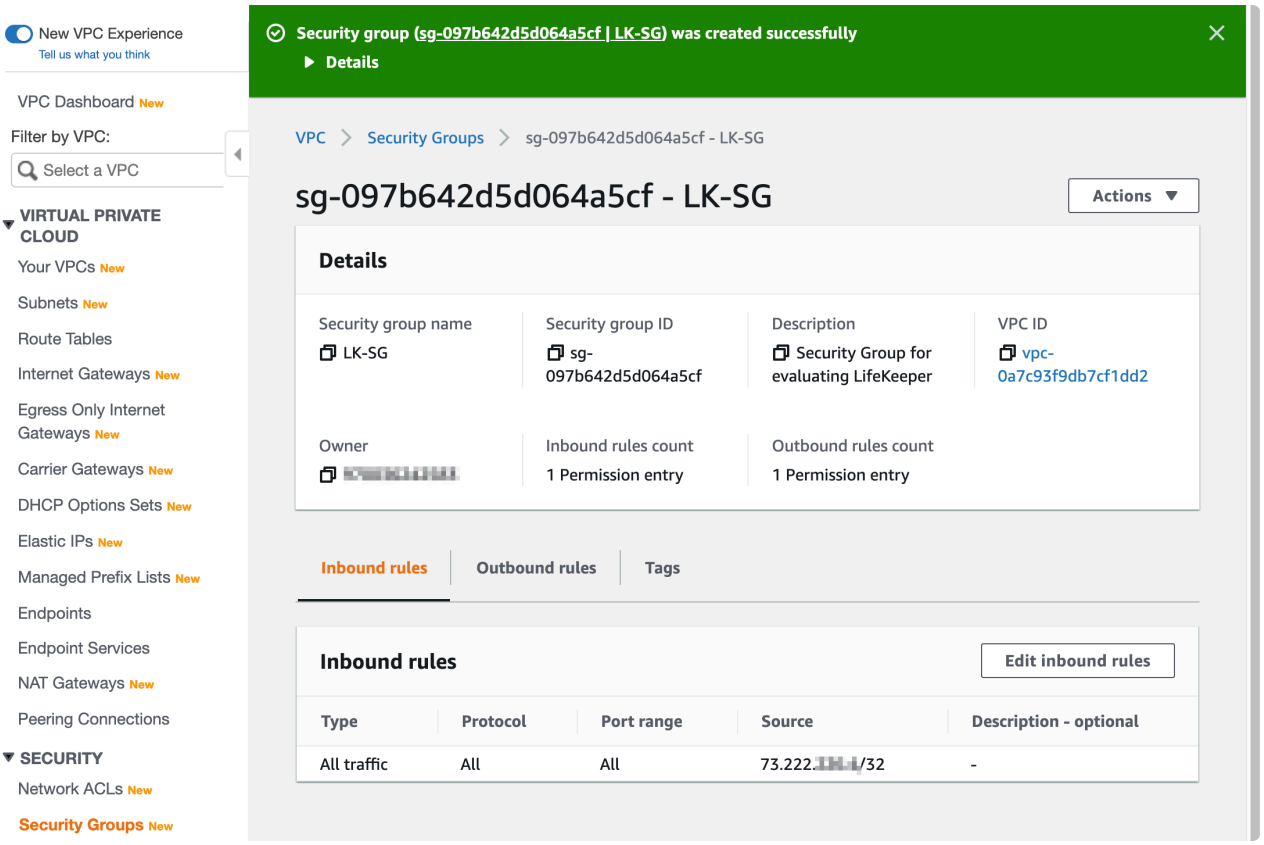

2. 次のパラメーターを使用して、スクリーンショットに示すように、新しいセキュリティグループの値を入力します。

- Name: `LK-SG`
- VPC: `LK-VPC`
- 最初のインバウンドルールを作成します。「ルールの追加」を選択し、タイプとして「すべてのトラフィック」を選択します（現時点では）。次に、ソースタイプとして「MyIP」を選択します。これにより、ユーザーのWANアドレスがソースとして自動的に選択されます。必要に応じて、使用中のWANアドレスに基づいてソースアドレスの範囲を更新します（これを特定のIPアドレスからIPアドレスの範囲に変更することが適切な場合があります）。

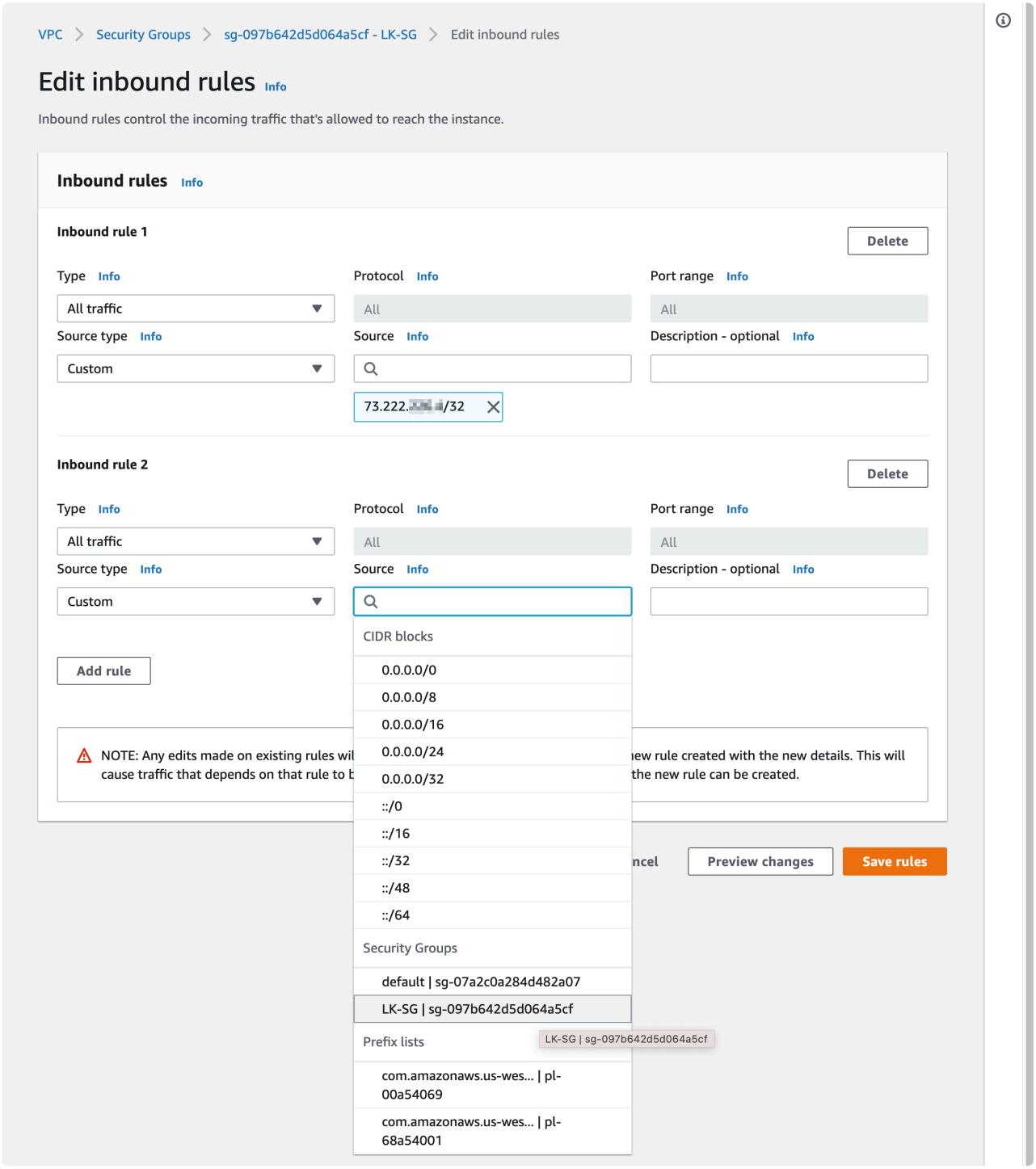
  値を確認したら、右下の「セキュリティグループを作成」をクリックします。


VPC > Security Groups > Create security group
Create security group Info
A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.
Basic details
Security group name Info
LK-SG
Name cannot be edited after creation.
Description Info
Security Group for evaluating LifeKeeper
VPC Info
vpc-0a7c93f9db7cf1dd2 (LK-VPC)
Inbound rules Info
Inbound rule 1
Delete
Type Info
All traffic
Protocol Info
All
Port range Info
All
Source type Info
My IP
Source Info
73.222.[redacted]/32
Description - optional Info
Add rule
Outbound rules Info
Outbound rule 1
Delete
Type Info
All traffic
Protocol Info
All
Port range Info
All
Destination type Info
Custom
Destination Info
0.0.0.0/0
Description - optional Info
Add rule
Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.
No tags associated with the resource.
Add new tag
You can add up to 50 more tag
Cancel
Create security group

3. 新しいセキュリティグループが作成されたら、[インバウンドルールの編集] をクリックして、同じセキュリティグループ内のトラフィックを許可します。

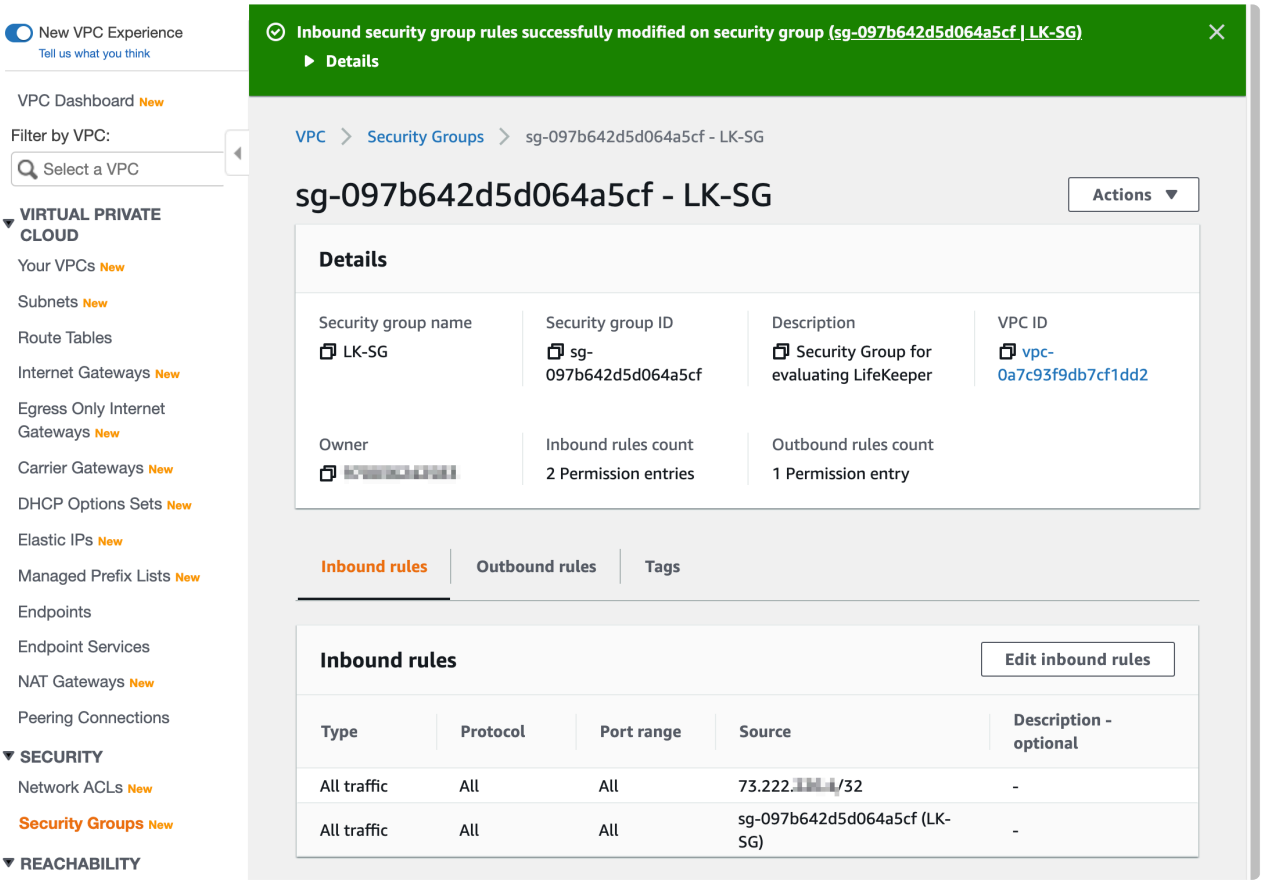



4. [インバウンドルールの編集] ページが表示されたら、[ルールの追加] を選択し、次に「LK-SG」（作成したもの）を選択します。この変更により、「LK-SG」に関連付けられているすべてのインスタンスが相互に通信できるようになります。




変更を確認したら、[ルールの保存] をクリックします。これでセキュリティグループが作成されました。

5. 新しく作成されたセキュリティグループのパラメータを確認します。

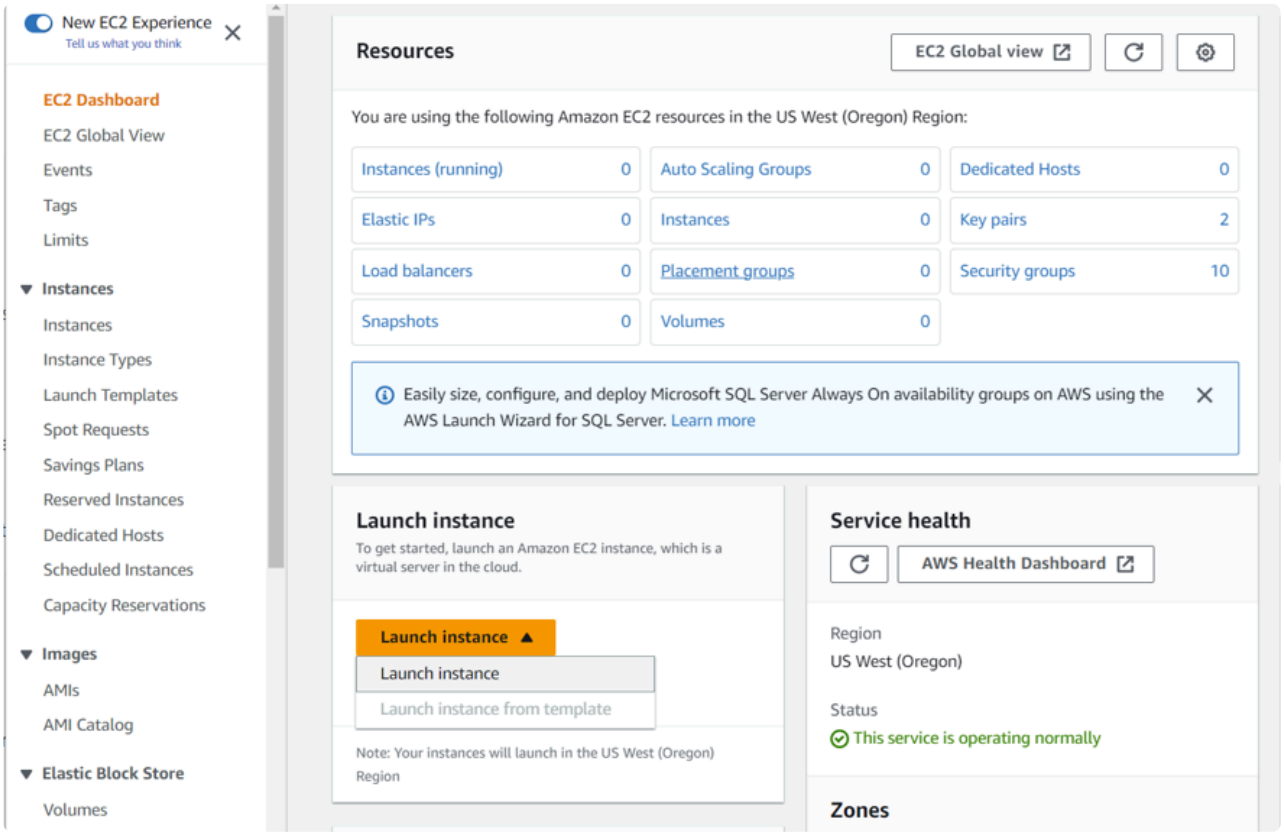

New VPC Experience
Tell us what you think
VPC Dashboard New
Filter by VPC:
Select a VPC
VIRTUAL PRIVATE CLOUD
Your VPCs New
Subnets New
Route Tables
Internet Gateways New
Egress Only Internet Gateways New
Carrier Gateways New
DHCP Options Sets New
Elastic IPs New
Managed Prefix Lists New
Endpoints
Endpoint Services
NAT Gateways New
Peering Connections
SECURITY
Network ACLs New
Security Groups New
REACHABILITY
Inbound security group rules successfully modified on security group (sg-097b642d5d064a5cf | LK-SG)
Details
VPC > Security Groups > sg-097b642d5d064a5cf - LK-SG
sg-097b642d5d064a5cf - LK-SG
Actions
Details
Security group name
LK-SG
Security group ID
sg-097b642d5d064a5cf
Description
Security Group for evaluating LifeKeeper
VPC ID
vpc-0a7c93f9db7cf1dd2
Owner
Inbound rules count
2 Permission entries
Outbound rules count
1 Permission entry
Inbound rules
Outbound rules
Tags
Inbound rules
Edit inbound rules
Type
Protocol
Port range
Source
Description - optional
All traffic
All
All
73.222.
/32
-
All traffic
All
All
sg-097b642d5d064a5cf (LK-SG)
-

# 11.2.3.3.8. 最初の EC2 インスタンスの作成

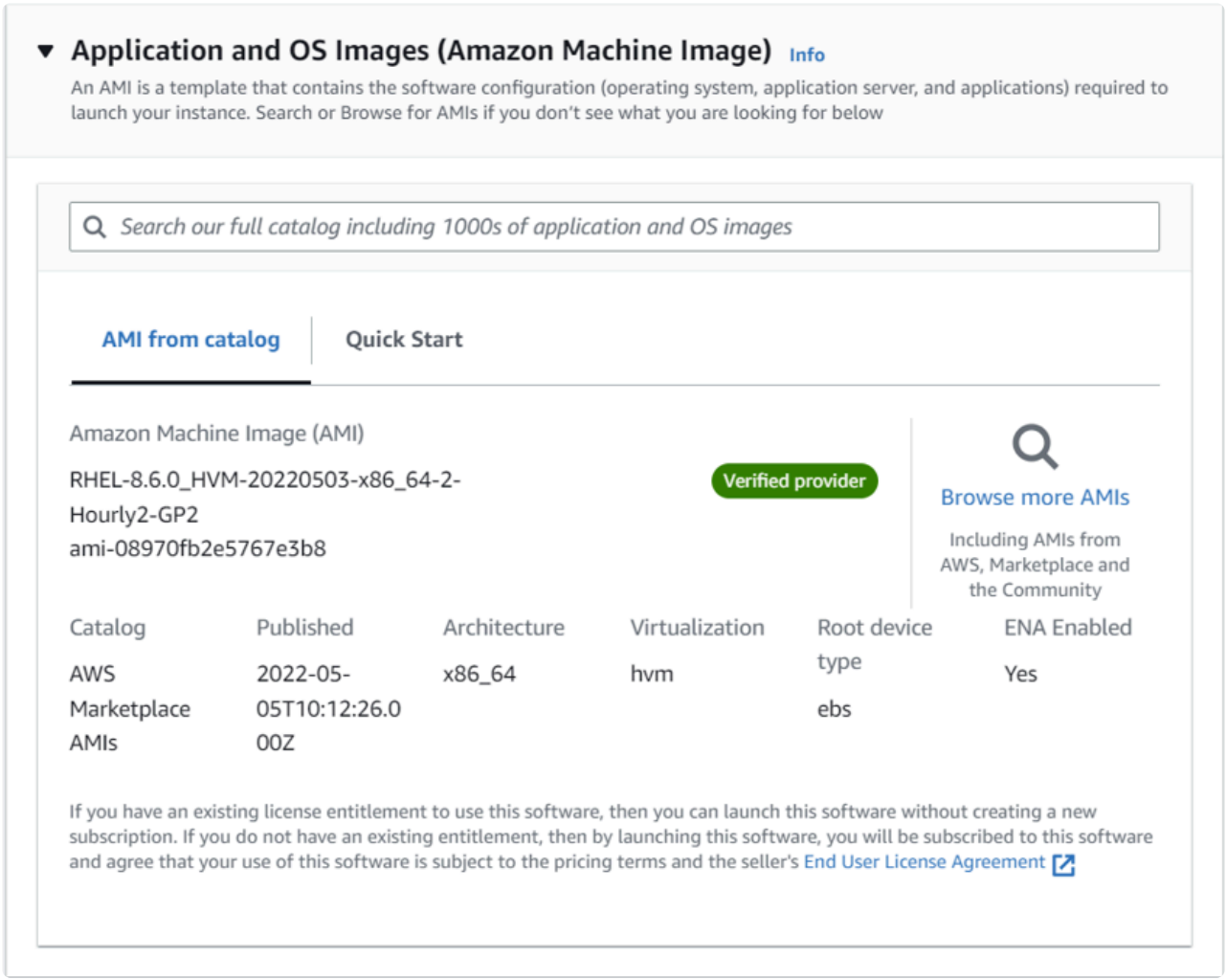
前のセクションでは、ネットワークの構成について説明しました。次に、最初のインスタンスを作成します。本チュートリアルで使用するコンピューティングリソース で説明したように、2つのディスクが必要です。このセクションでは、2番目のディスクを作成する方法についても説明します。

1. 「EC2」サービスに移動します。EC2を選択するには、AWSコンソールの上部の検索ボックスに「EC2」と入力する必要がある場合があります。

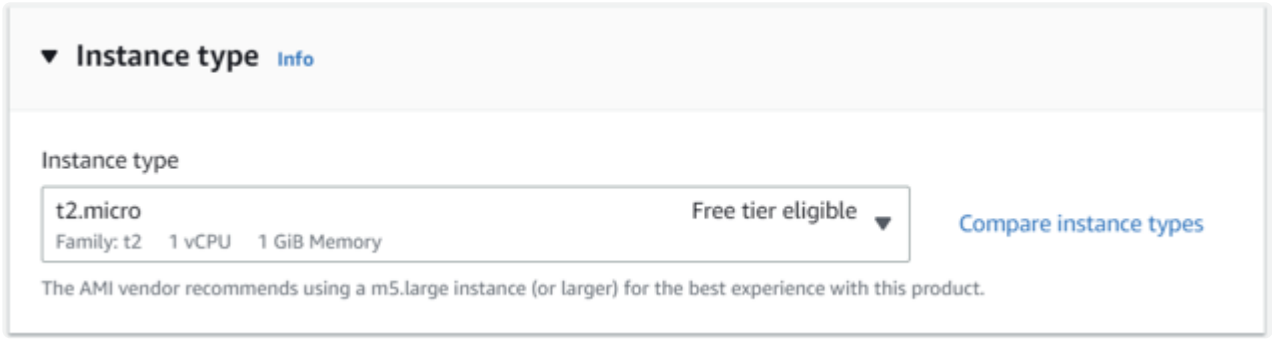
2. 「インスタンスを起動」をクリックします。




3. 「オペレーティングシステム」をクリックします。オペレーティングシステムの名前を入力し、特定のマシンイメージを選択します。オペレーティングシステムとそのバージョンを選択する前に、SIOS Protection Suiteと保護するアプリケーションの両方のシステム要件を確認してください。




4. 「インスタンスサイズ」をクリックします。このチュートリアルでは、 `t2.micro` を使用します。これは、SIOS Protection Suiteを評価するための最小システム要件として定義されており、無料利用枠の使用条件を満たす可能性があるためです。この場合も、保護するアプリケーションによっては、より大きなインスタンスサイズを選択する必要がある場合があります。

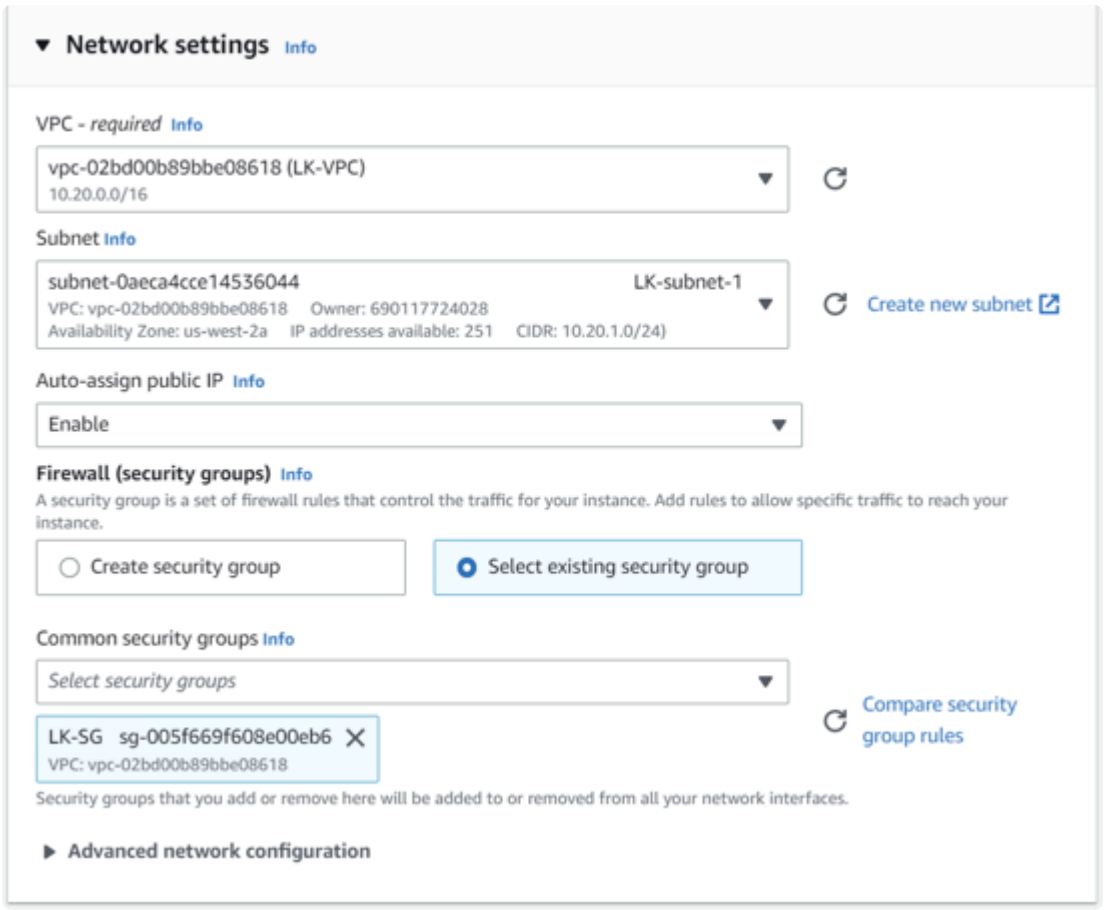


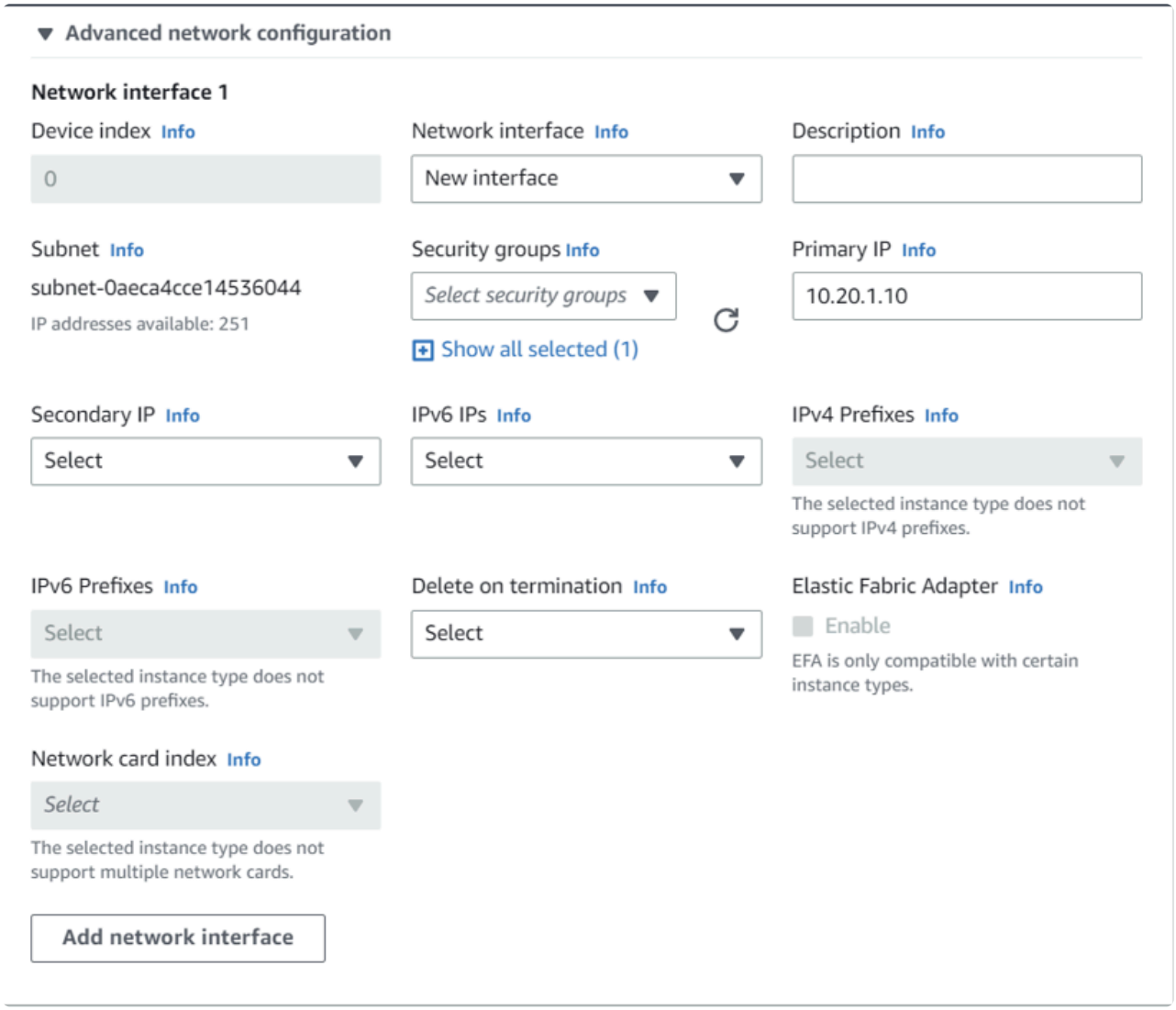

インスタンスサイズを選択したら、[Next: Configure Instance Details] をクリックします。

5. 「インスタンスの詳細設定」ウィザードで、次のパラメーターが使用されていることを確認します。

- VPC: `LK-VPC`
- Subnet: `LK-subnet-1`
- Network Interface: Please enter `10.20.1.10` for the first instance (node-a).

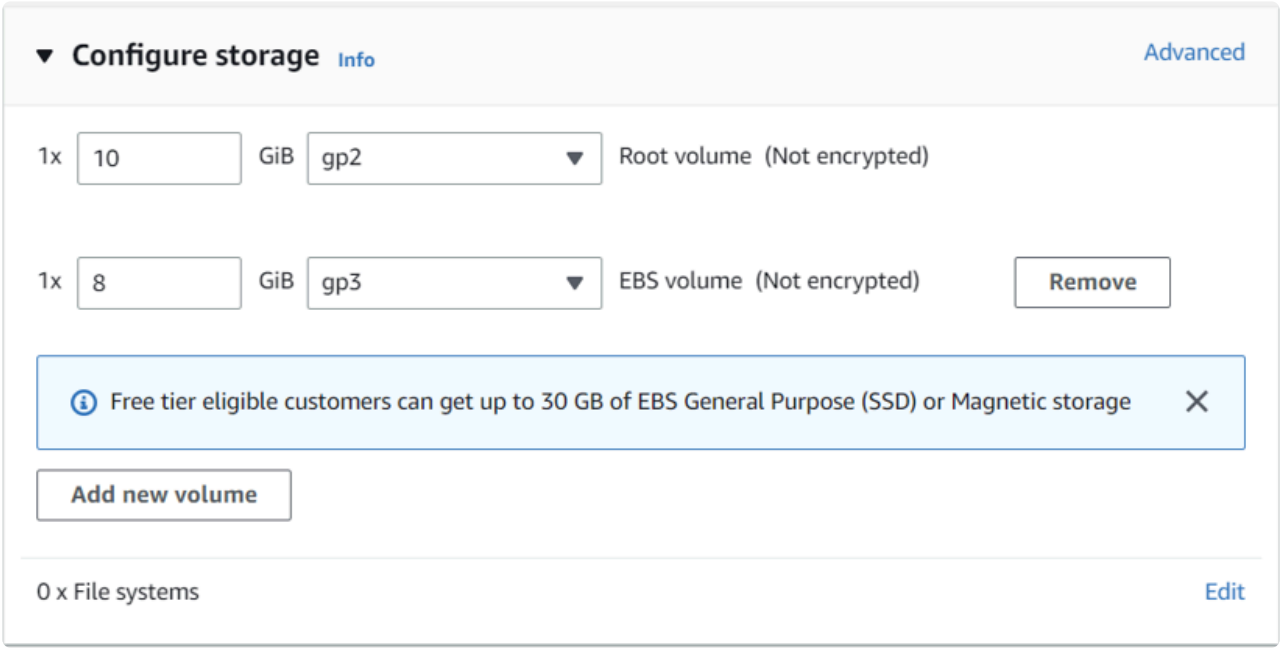
**Note:** node-bおよびnode-cを作成する際には、 node-bおよびnode-cに関連するパラメーター を参照してください。






これらの値を確認したら、[Next: Add Storage] をクリックします。

6. ウィザードでは、すでに最初のストレージデバイス（ボリューム）が作成されています。新しいボリュームの追加」をクリックし、新しいディスクを作成します。このガイドで前述した最小要件のとおり、8GiBのディスクで十分です。ただし、保護（複製）するデータの容量によっては異なる場合があります。

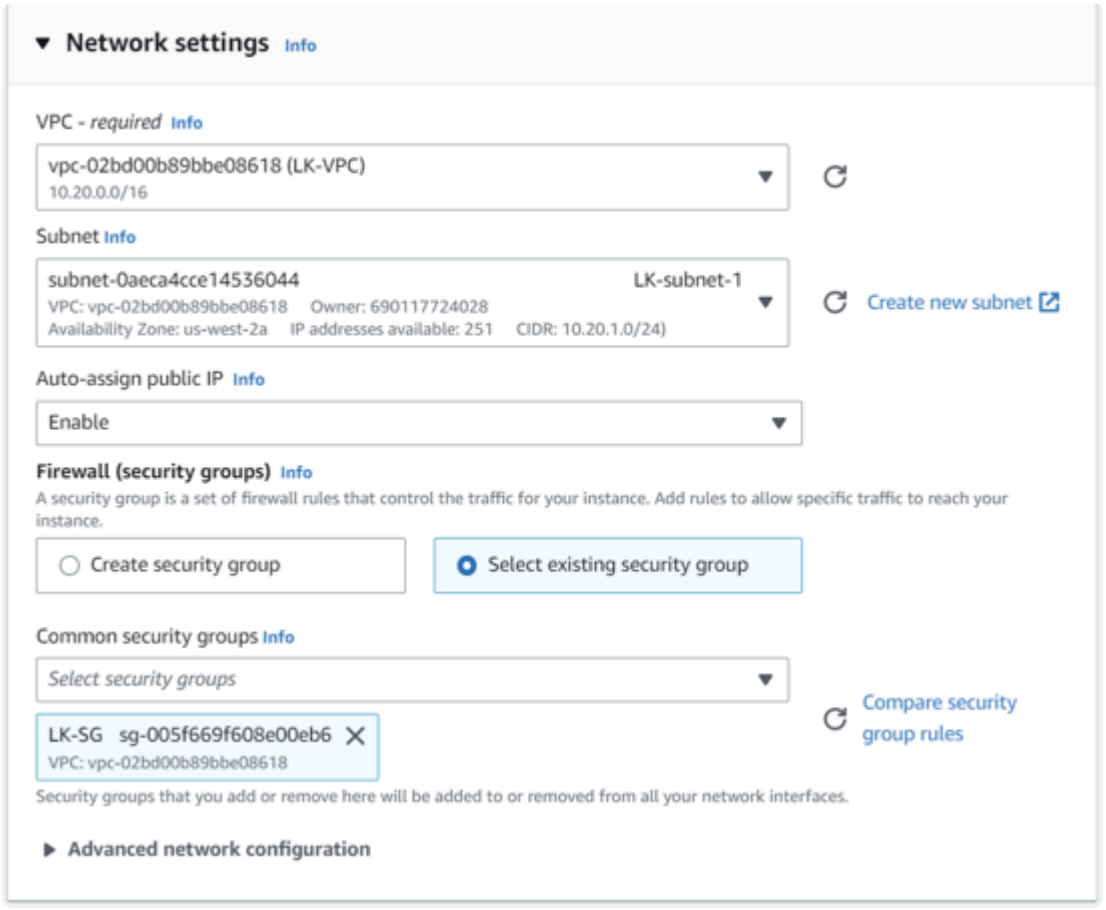



選択内容を確認したら、「Next: Add Tags」をクリックします。

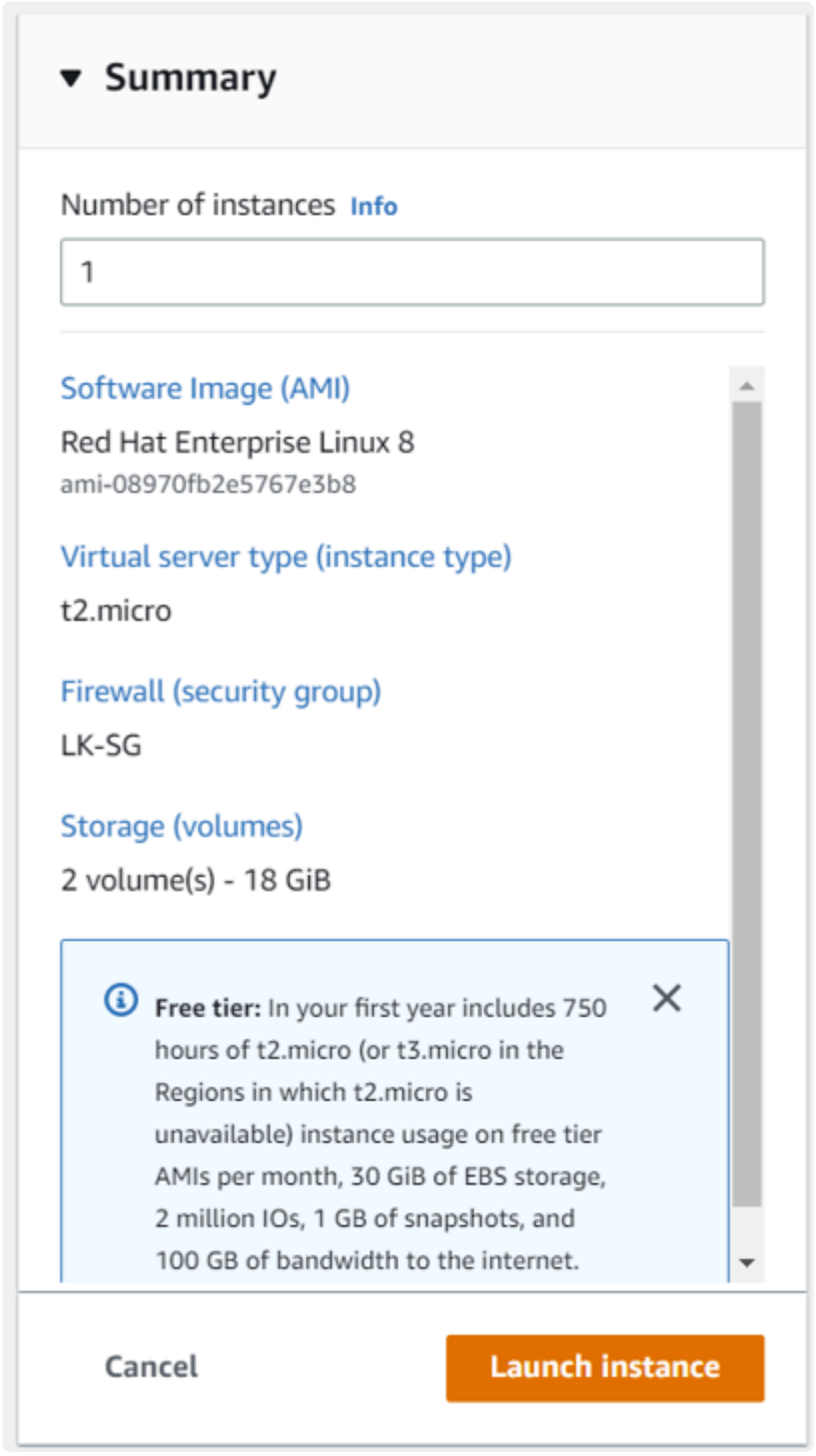
7. キー「Name」と値「Node-A」のタグを追加して、AWSEC2コンソールからインスタンスを識別しやすくします。

「Name」タグを定義したら、「Next: Configure Security Group」をクリックします。

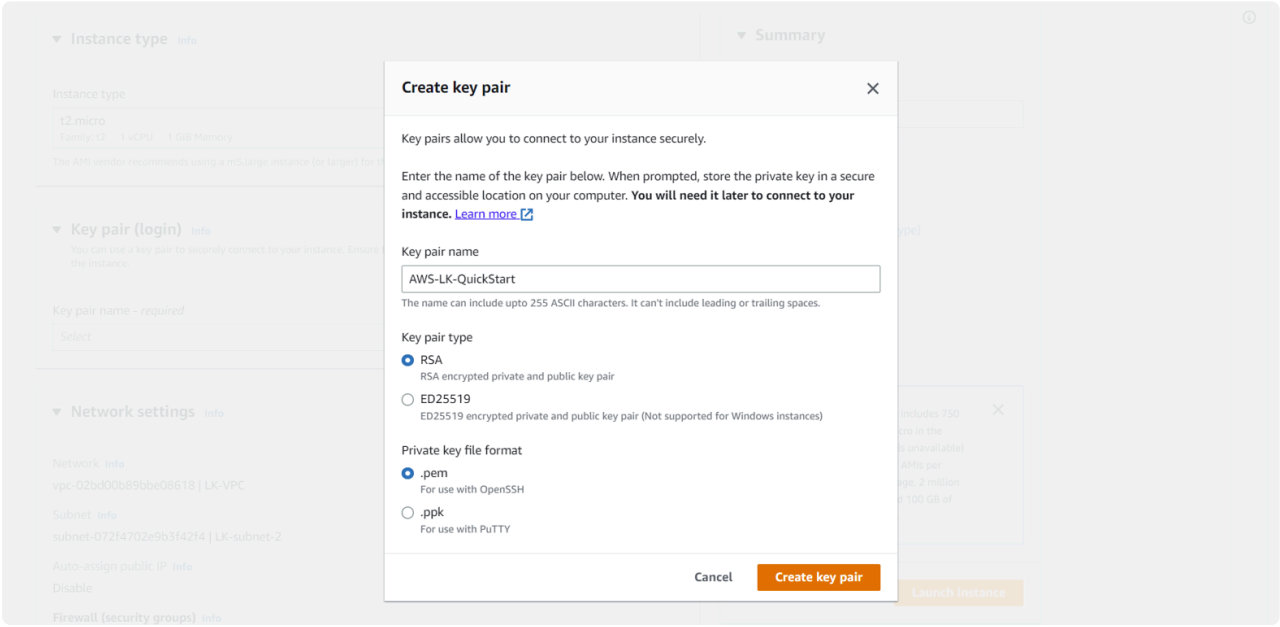
8. 前のセクション（LK-SG）で作成したセキュリティグループを選択します。



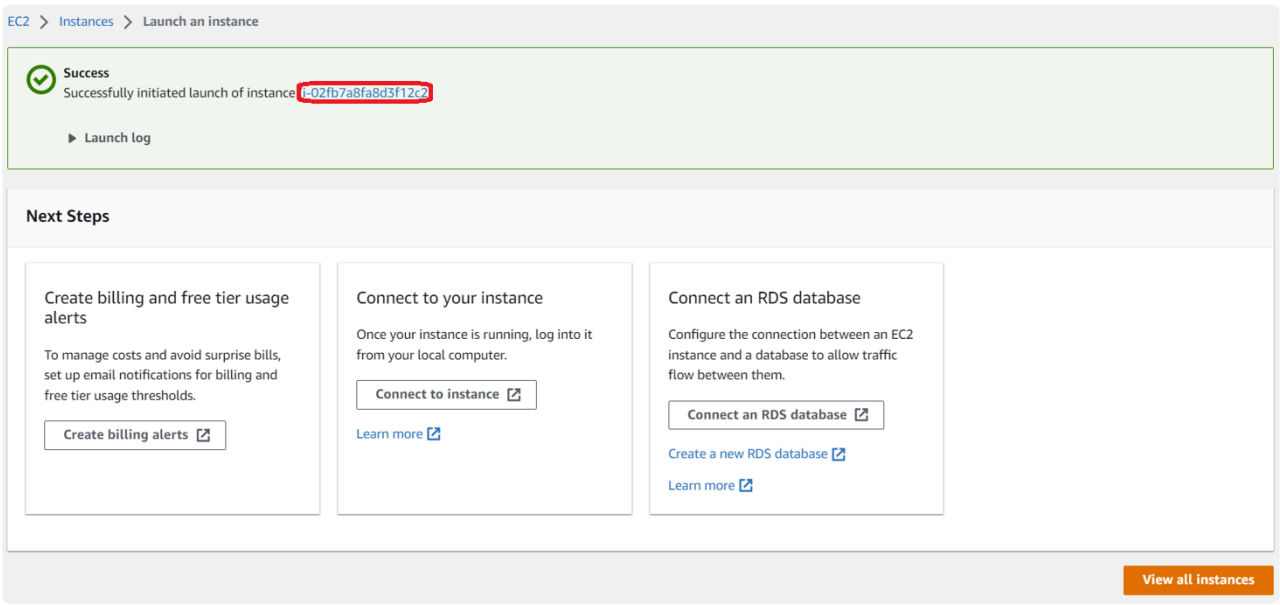

9. 選択した項目を確認し、「起動」をクリックします。




10. 新しく作成されたLinuxEC2インスタンスにアクセスするために、ssh経由で接続します。AWSは、キーペアを使用してユーザーのsshセッションを認証します。キーペアの名前を作成し、ローカルシステムにダウンロードします。EC2インスタンスにアクセスするには、秘密鍵が必要になります。「インスタンスの起動」をクリックします。




11. 「Launch Status」ページが表示されます。インスタンスIDを選択して、詳細を表示します。

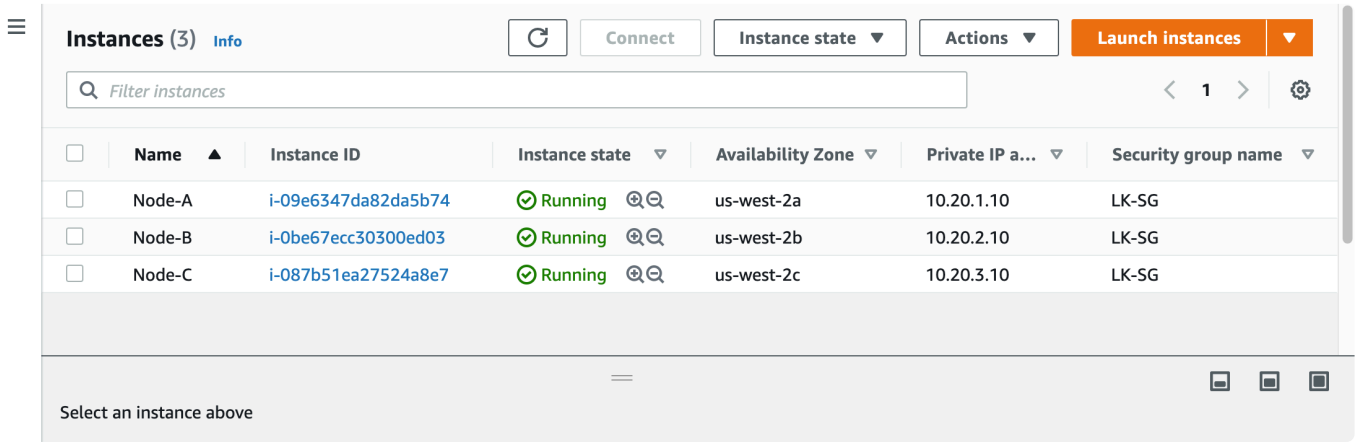


# 11.2.3.3.9. 2番目と3番目のインスタンスの作成

2番目のノード（node-b）と3番目のノード（node-c）を作成するための手順は、最初のノード（node-a）について前述した手順とほぼ同じです。次の表は、3つのノードの違いと、これらのインスタンスを作成するために必要な詳細を示しています。

| 名前 | パラメータ | 値 |
|---|---|---|
| インスタンス間で共通の値 | **VPC** | **LK-VPC** |
| | **Security Group** | **LK-SG** |
| | **Source/Dest Checking on a network interface (ENI)** | 無効 |
| node-a | Subnet | LK-subnet-1 |
| | Private IP Address | `10.20.1.10` |
| | Second Disk (Storage Device) | セカンドディスクが必要です |
| node-b | Subnet | LK-subnet-2 |
| | Private IP Address | `10.20.2.10` |
| | Second Disk (Storage Device) | セカンドディスクが必要です |
| node-c | Subnet | LK-subnet-3 |
| | Private IP Address | `10.20.3.10` |
| | Second Disk (Storage Device) | 2枚目のディスクを接続する必要はありません |

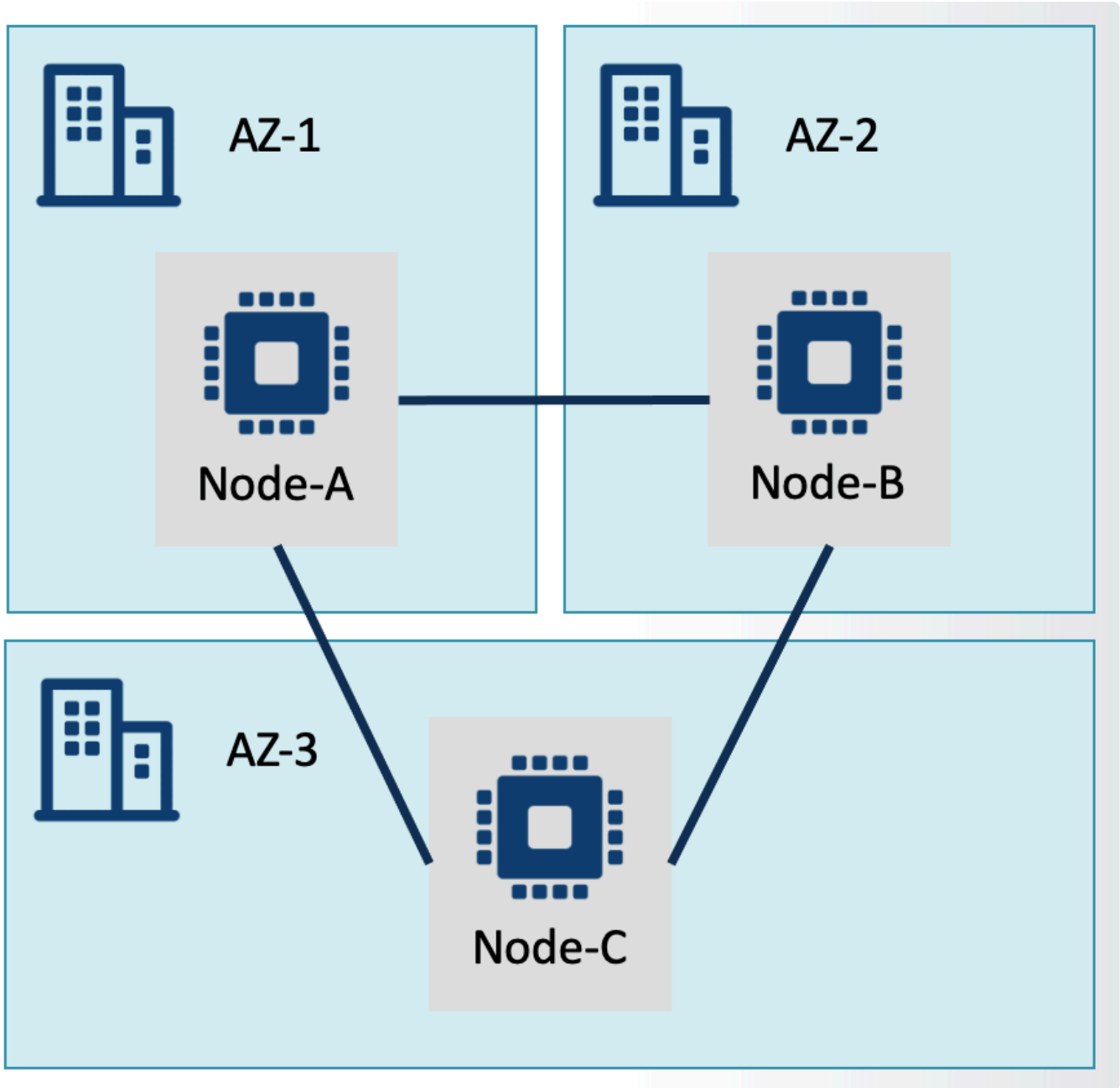
作成すると、インスタンスのリストは次のようになります。



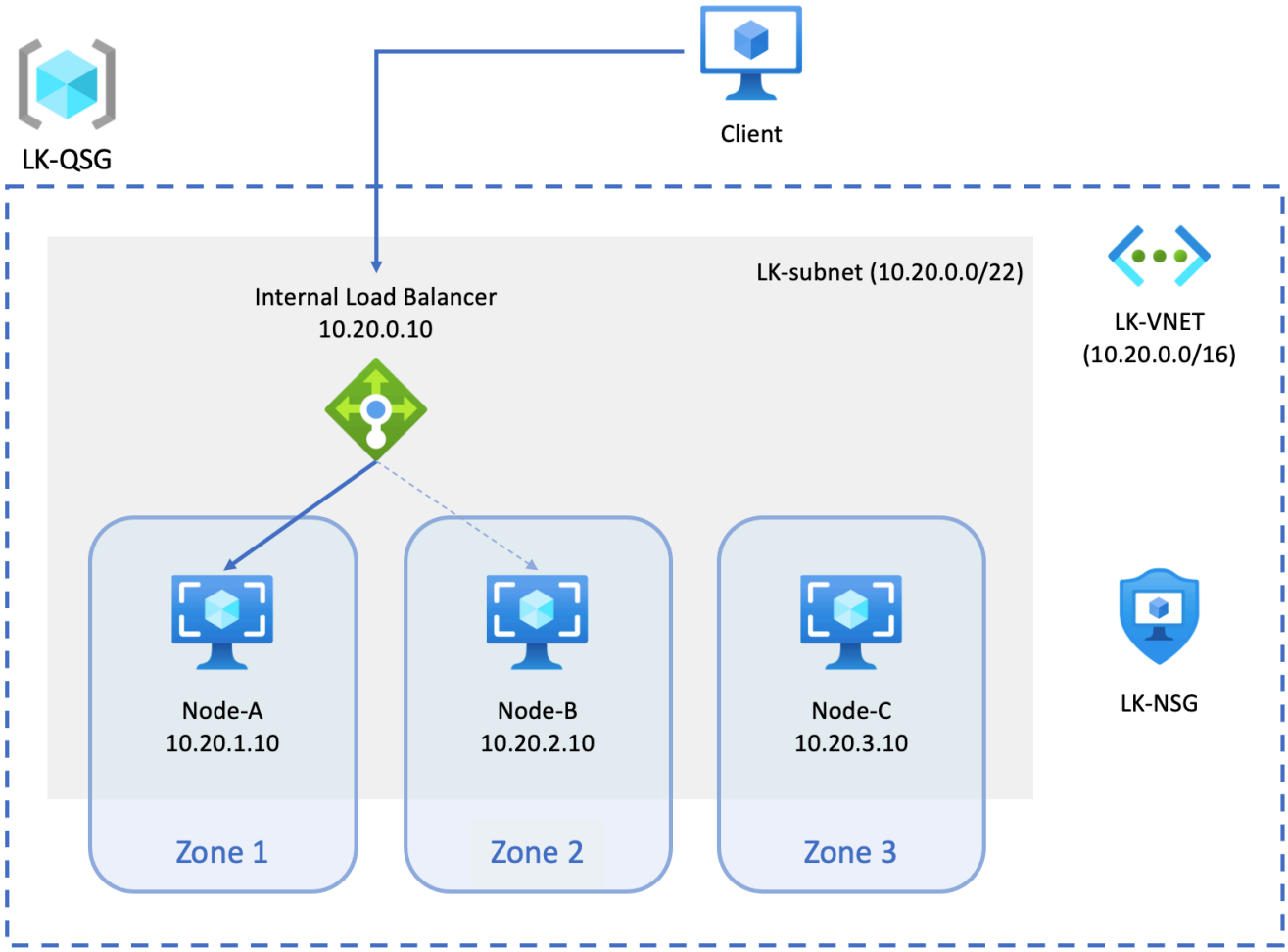
# 11.2.3.4. Azureでゼロからインスタンスを作成する

**免責事項:** ユーザーインターフェースは地域によって異なる場合や、時間の経過とともに変更される場合があります。以下に示すスクリーンショットがお使いのものと異なる場合は、Azureが提供するドキュメントを参照してください。

評価ガイドでは、次のネットワーク構造とインスタンスを使用しています。




Azureでは、これらのコンポーネントは次の図に示すように定義できます。




このセクションでは、内部ロードバランサーを除くこれらのコンポーネントを作成します。内部ロードバランサーは 後で 作成します。

この図に記載されているコンポーネントについて、次の表で説明します。
このネットワーク構造を作成するには、次のコンポーネントを実装する必要があります。

| コンポーネント | 名前 | パラメーター | 値 |
|---|---|---|---|
| Resource Group | LK-QSG | | |
| Virtual network | LK-VNET | IPv4 CIDRs | 10.20.0.0/16 |
| | | Subnet (LK-subnet) | 10.20.0.0/22 |
| Network Security | LK-NSG | Additional Inbound Rule<br><br>Type=SSH | Source=Your Office's WAN IP<br><br>Destination=Any |

| | | | |
|---|---|---|---|
| | | Additional Outbound Rule<br><br>Type=Any | Source=VirtualNetwork<br><br>Destination=AzureLoadBalancer |
| Virtual Machine | Common values across VM | Resource Group | LK-QSG |
| | | Virtual network | LK-VNET |
| | | Subnet | LK-subnet |
| | | Network Security Group | LK-NSG |
| | node-a | Availability Zone (*1) | 1 |
| | | Private IP Address (*2) | 10.20.1.10 |
| | node-b | Availability Zone (*1) | 2 |
| | | Private IP Address (*2) | 10.20.2.10 |
| | node-c | Availability Zone (*1) | 3 |
| | | Private IP Address (*2) | 10.20.3.10 |

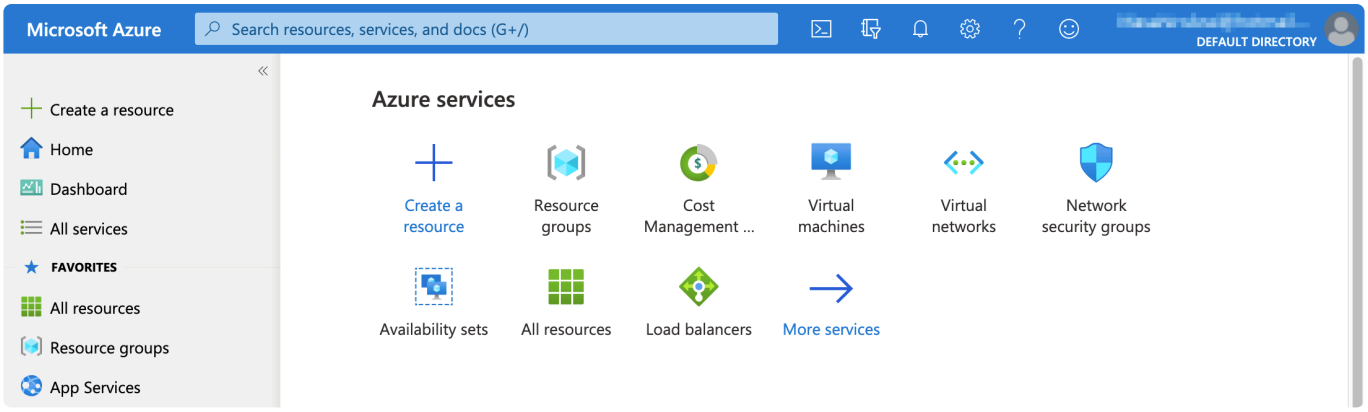
- **(*1)** これは、仮想マシンの作成時に選択する必要があります。
- **(*2)** ウィザードでは、DHCPベースのIPアドレスが自動的に割り当てられます。これは、仮想マシンを作成した後に変更する必要があります。

以下のセクションでは、これらのネットワークコンポーネントを作成するために必要な手順を、最初のインスタンスから順に説明します。

- Azureサービスの切り替え
- Azureリージョンの決定
- リソースグループの作成
- 仮想ネットワークの作成
- ネットワークセキュリティグループの作成
- 1つ目のAzure仮想マシンの作成
- 2番目と3番目の仮想マシンの作成

# 11.2.3.4.1. Azureサービスの切り替え

Azureは、かなりの数のサービスを利用でき、またその数も増えています。代表的なサービスは、左側にあるナビゲーションバーからアクセスするか、ホーム画面（ナビゲーションバーで [ホーム] を選択）にリストアップされています。




特定のサービスが見つからない場合は、上部にある検索ボックスにサービス名を入力してください。

# 11.2.3.4.2. Azure リージョンの決定

Azure には、地理的に多くの場所にリージョンがあります。職場から地理的に近いリージョンを選択することが有益な場合があります。ただし、すべてのリージョンに 3 つのアベイラビリティゾーンがあるわけではないので、選択したリージョンが意図した構成をサポートすることを保証するために、選択の際には注意が必要です。後のステップでリソースグループやその他のリソースを作成する際には、正しいリージョンを選択するようにしてください。

クロスリージョンネットワークのフェールオーバーはサポートされていません。

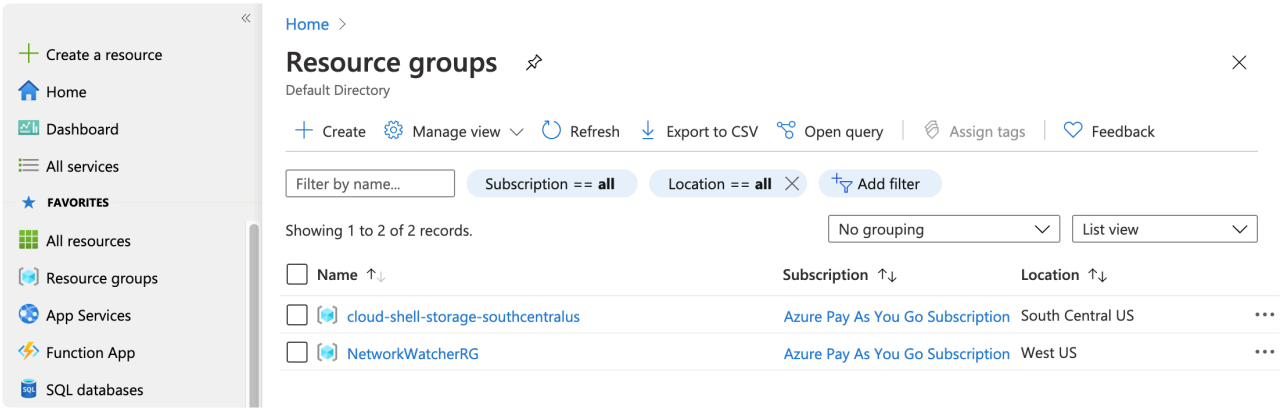
詳細については、Azureのドキュメント を参照してください。

# 11.2.3.4.3. リソースグループの作成

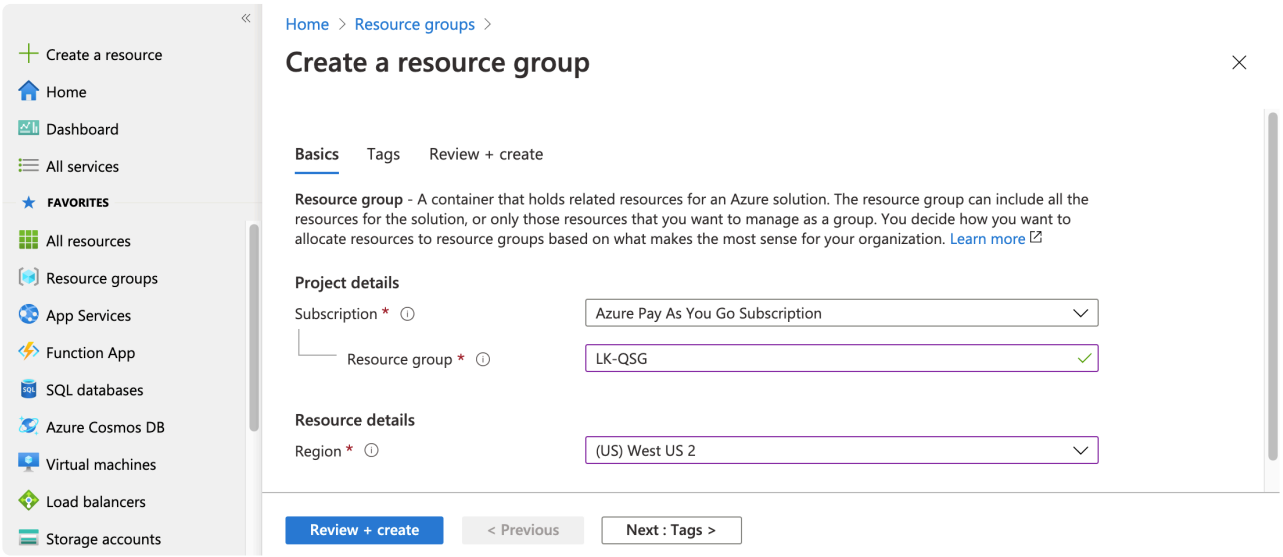
リソースグループは、ネットワークコンポーネント、コンピューターリソース、ストレージなどのリソースを作成するための論理コンテナです。
詳細については、 Azureのドキュメント を参照してください。

このセクションでは、LifeKeeperをテストするためのリソースグループ（LK-QSGという名前を付けます）を次のように作成します。

1. ホーム画面から「リソースグループ」を選択して既存のリソースグループのリストを参照し、上部にある [作成] をクリックします。

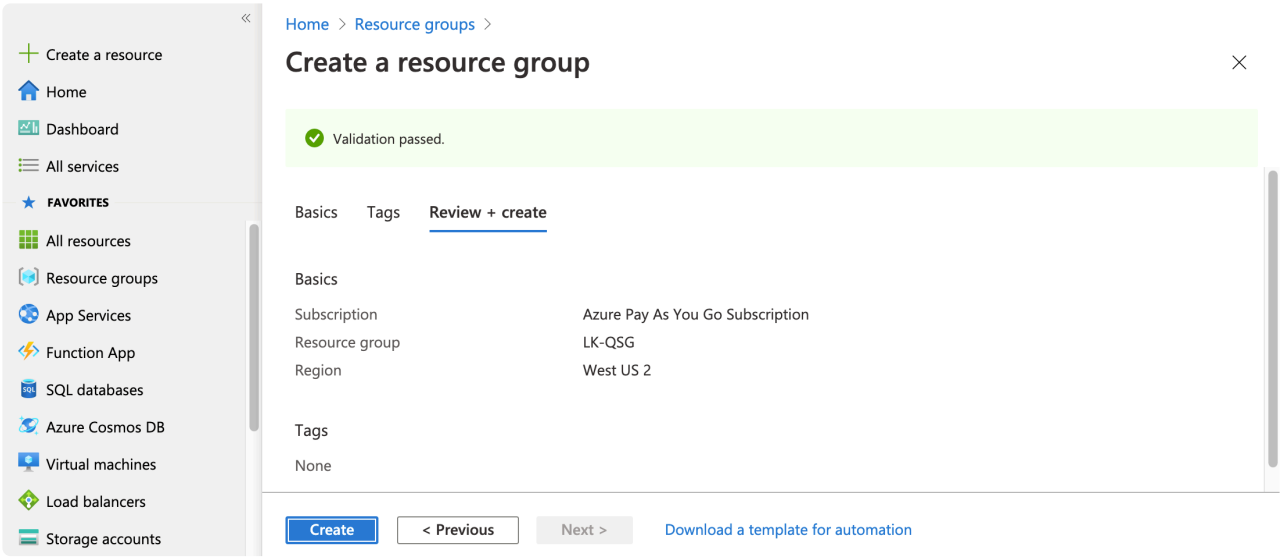



2. リソースグループの名前を LK-QSG と指定し、リージョンを選択してリソースグループを作成します。リソースグループの作成に使用されるリージョンには、3つの可用性ゾーンが必要であることに注意してください。



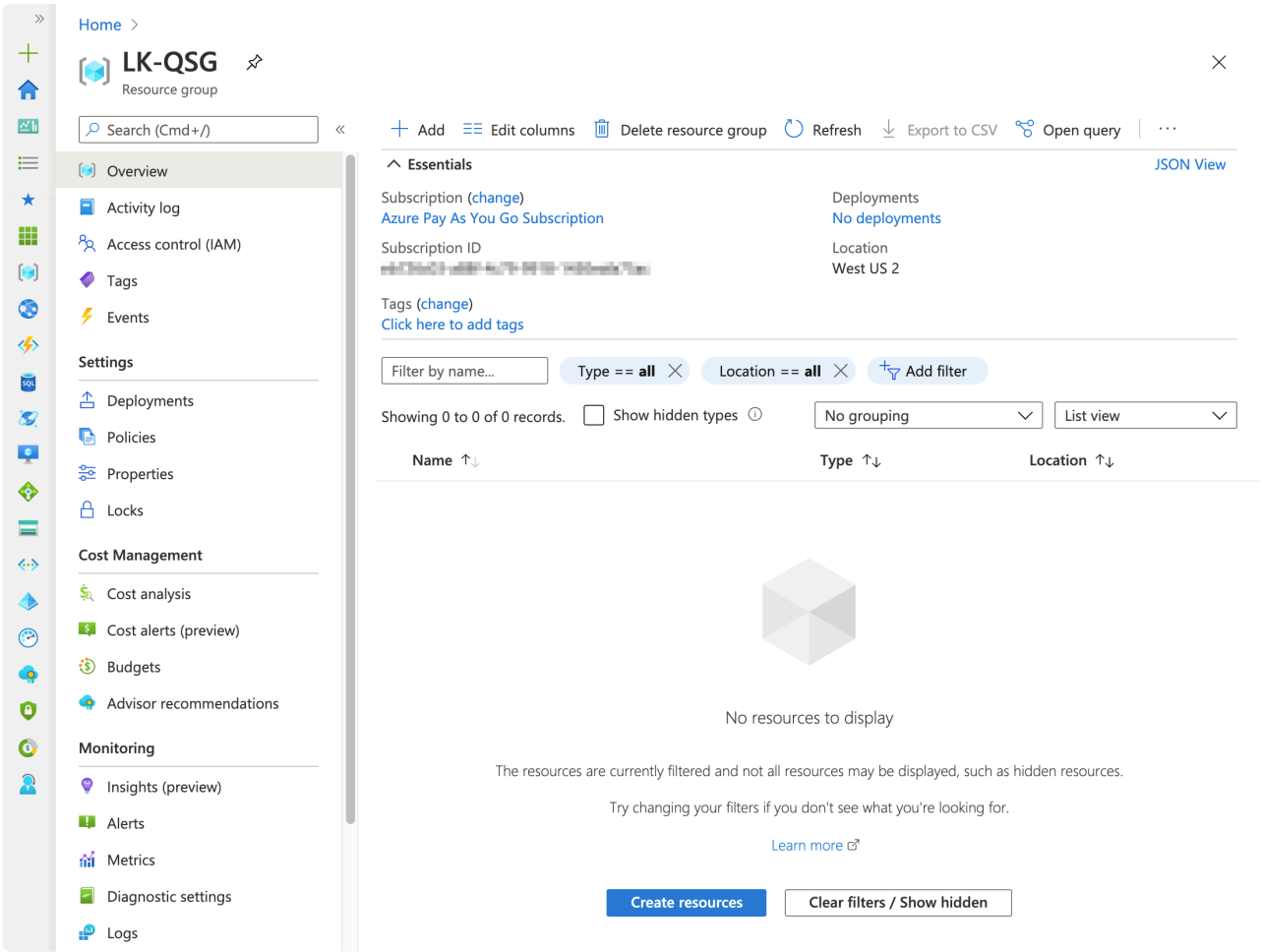

選択したら、[確認と作成] をクリックします。

3. ウィザードがこれらの値を評価し、検証に合格するとリソースを作成できるようになります。




[作成] をクリックして、リソースグループを作成します。

4. リソースグループが作成されました。

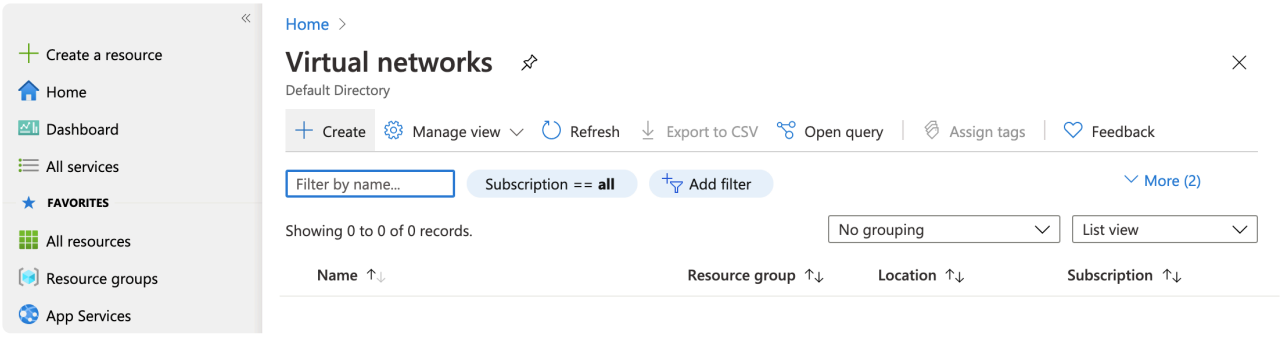


# 11.2.3.4.4. 仮想ネットワークの作成

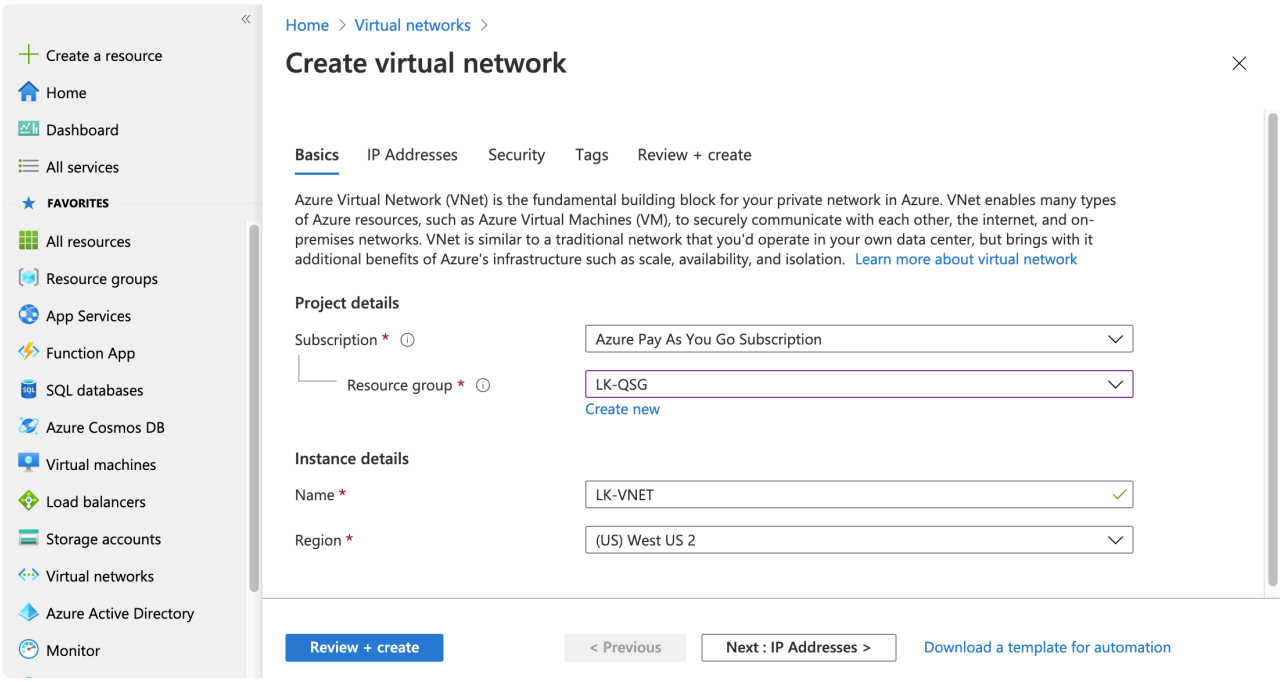
仮想ネットワークは、ローカルネットワークを表すAzureリソースです。異なるシステムを論理的に分離するために、Azureクラウド内でさまざまな仮想ネットワークを定義することができます。

詳細については、 Azureのドキュメント を参照してください。
このセクションでは、LifeKeeper（LK-VNETという名前を付けます）をテストするための仮想ネットワークを以下の手順で作成します。

1. ホーム画面から「仮想ネットワーク」を選択して現在の仮想ネットワークのリストを表示し、上部の [作成] をクリックします。

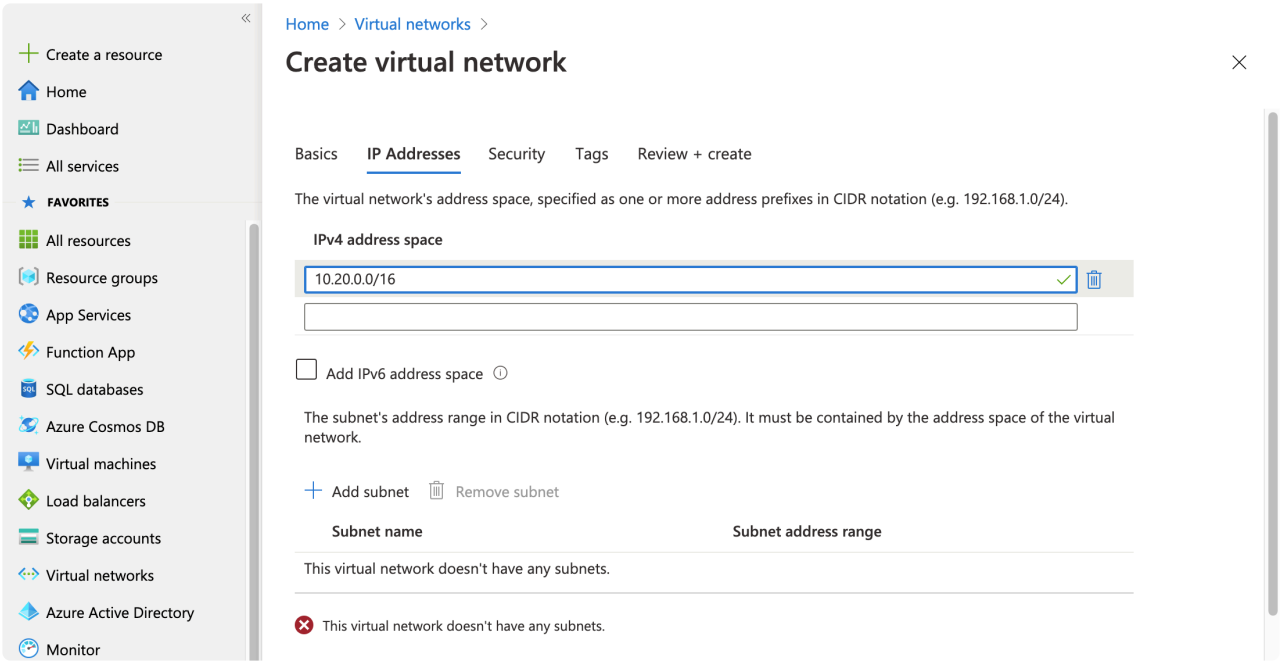



2. 「仮想ネットワーク」ウィザードで、リソースグループLK-QSG、仮想ネットワークLK-VNET、仮想ネットワークを作成するリージョンを選択します。

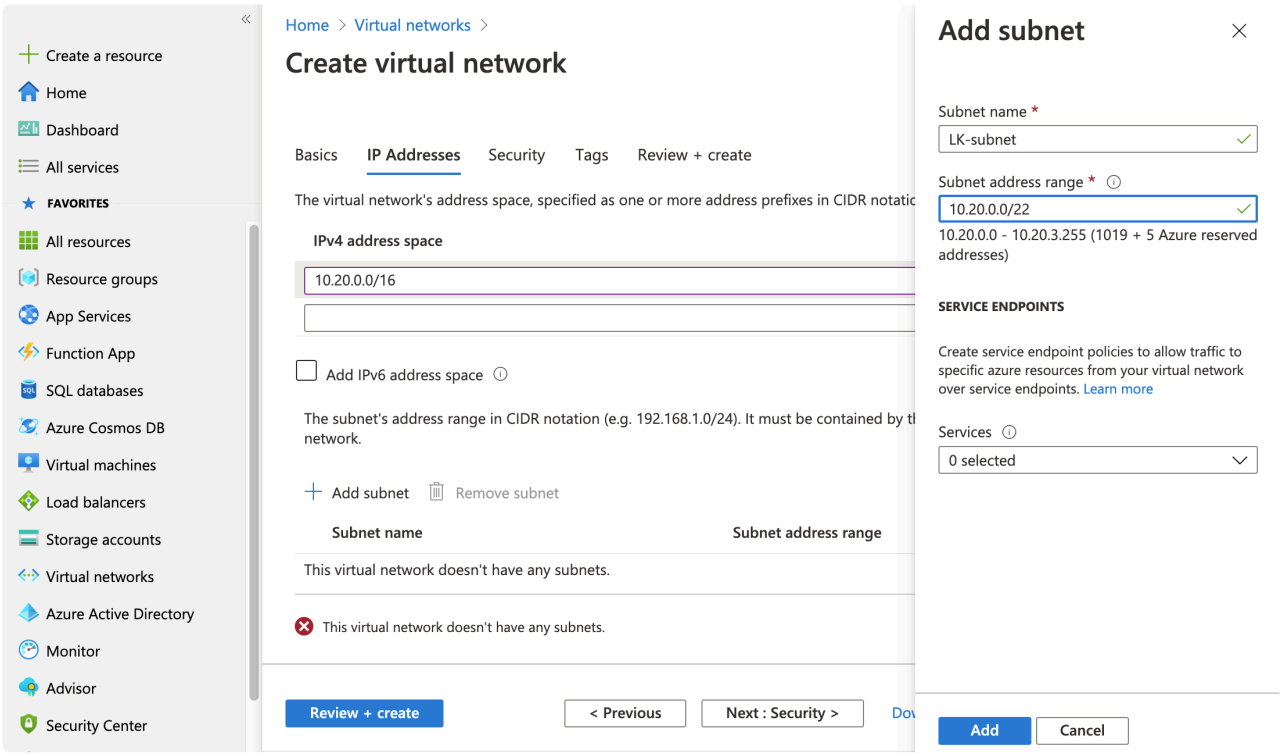



値を入力したら、[次へ: IPアドレス] をクリックします。

3. この仮想ネットワークのCIDRブロックを 10.20.0.0/16と入力します。CIDRブロックを入力したら、[サブネットの追加] をクリックしてサブネットを定義します。

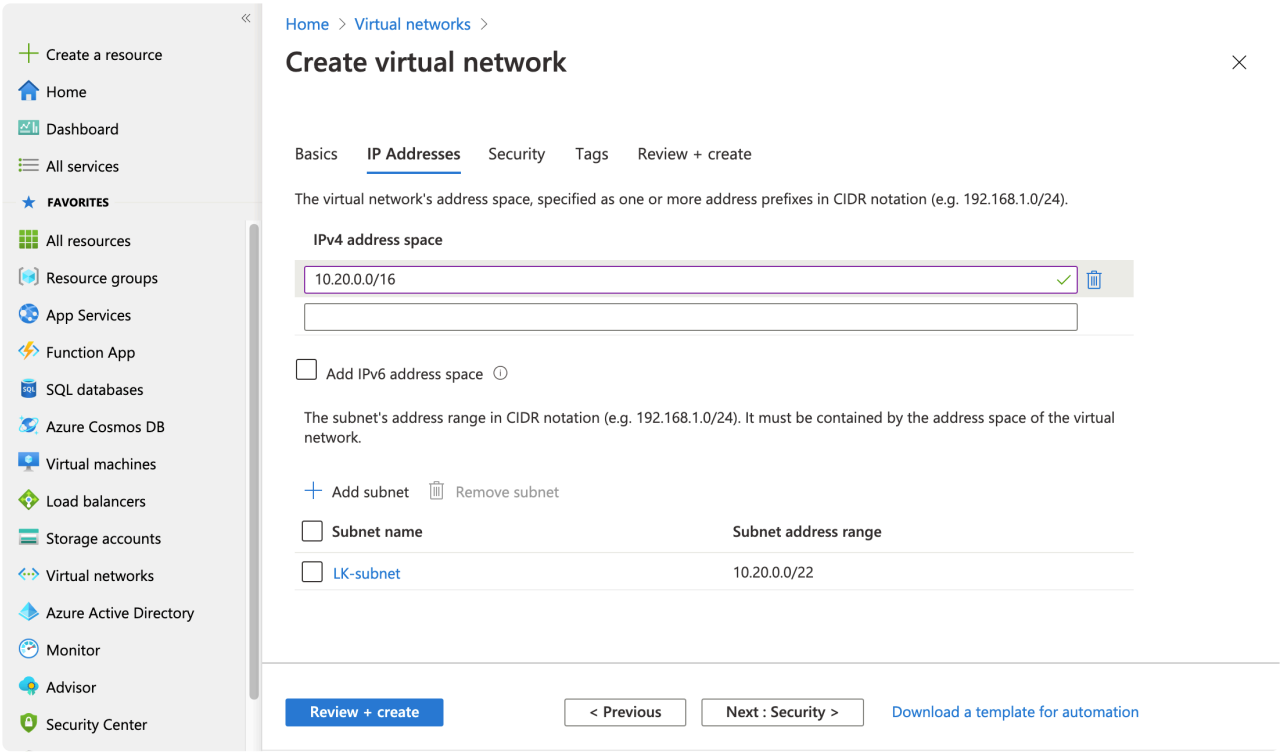



4. サブネットの名前LK-subnetとそのCIDRブロック 10.20.0.0/22を指定します。

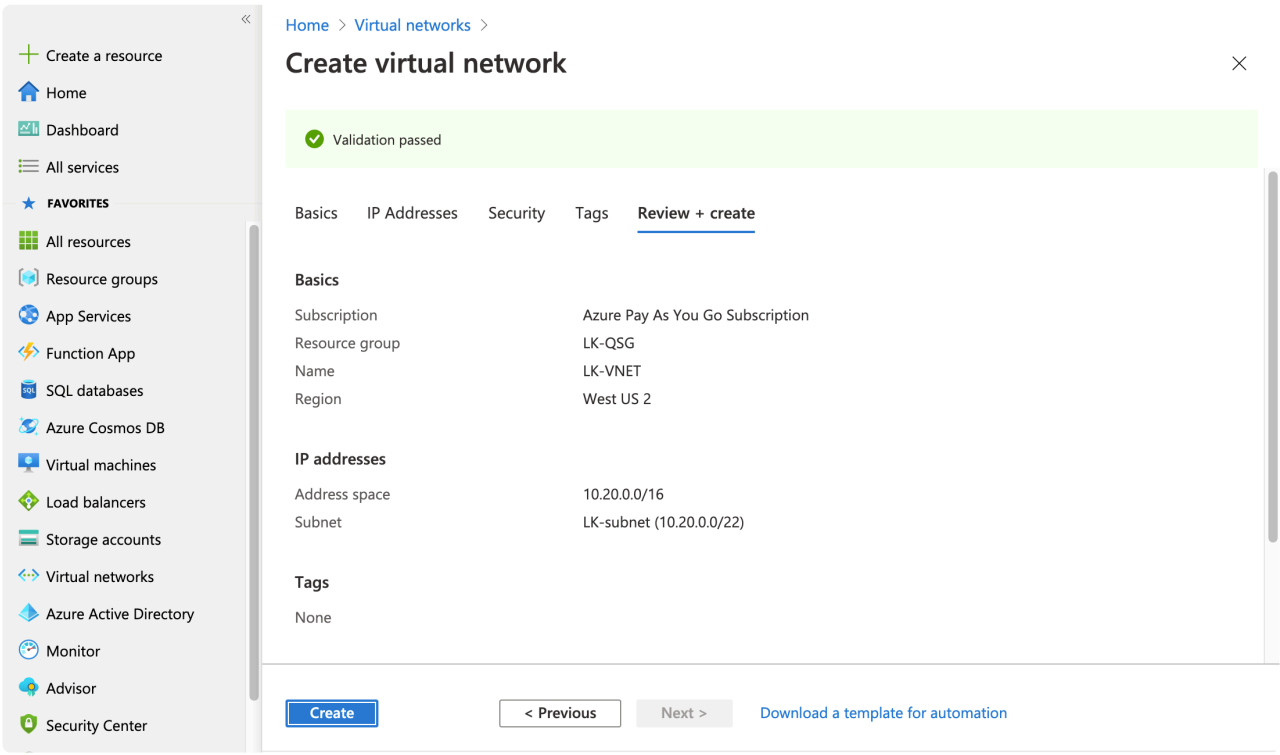



指定したら、下部にある [追加] をクリックします。

5. これで、仮想ネットワークとサブネットの両方のパラメーターが定義されました。[確認と作成] をクリックします。




6. ウィザードがパラメータを検証したら、[作成] をクリックして仮想ネットワークを作成します。

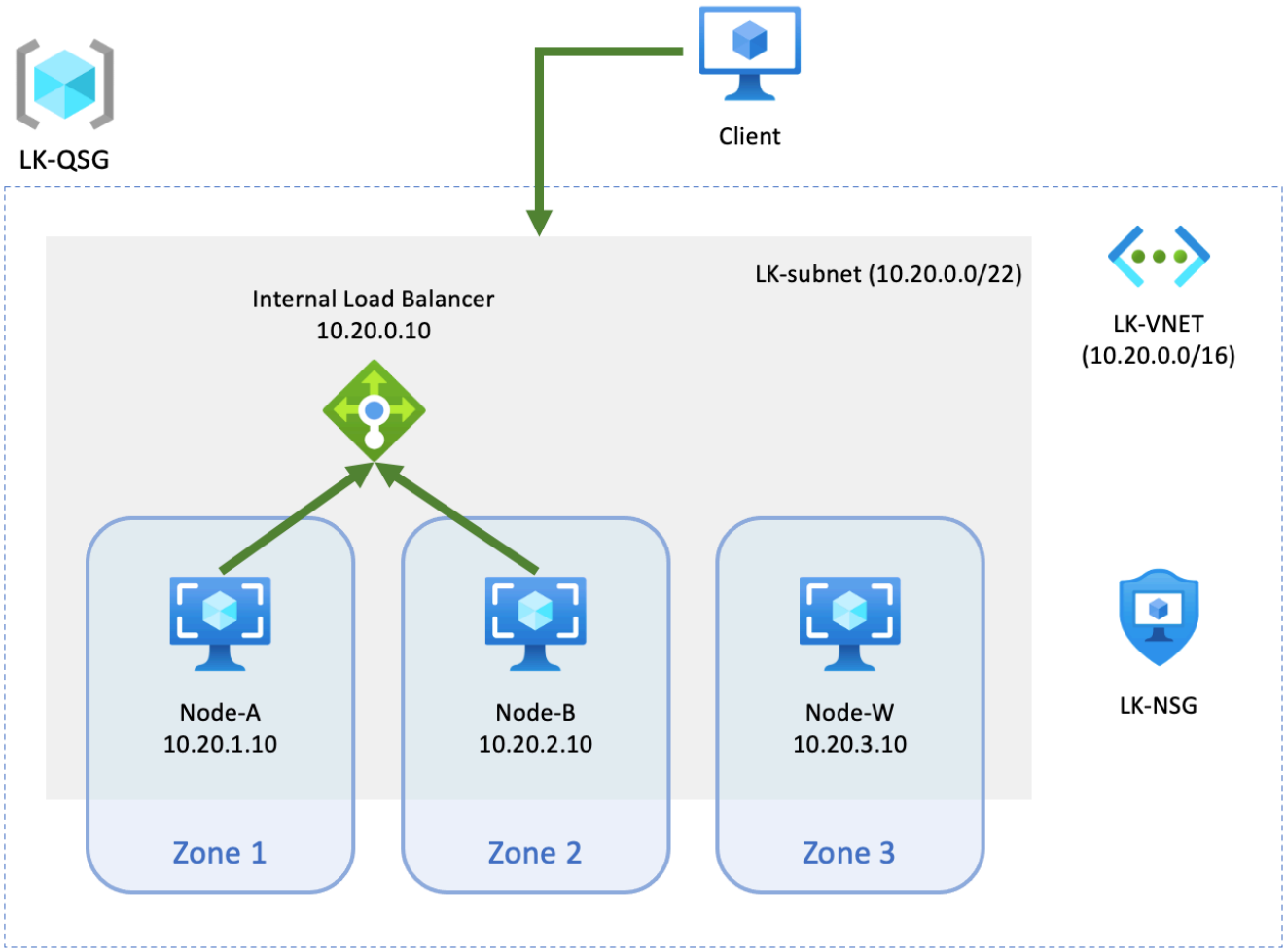


# 11.2.3.4.5. ネットワークセキュリティグループの作成

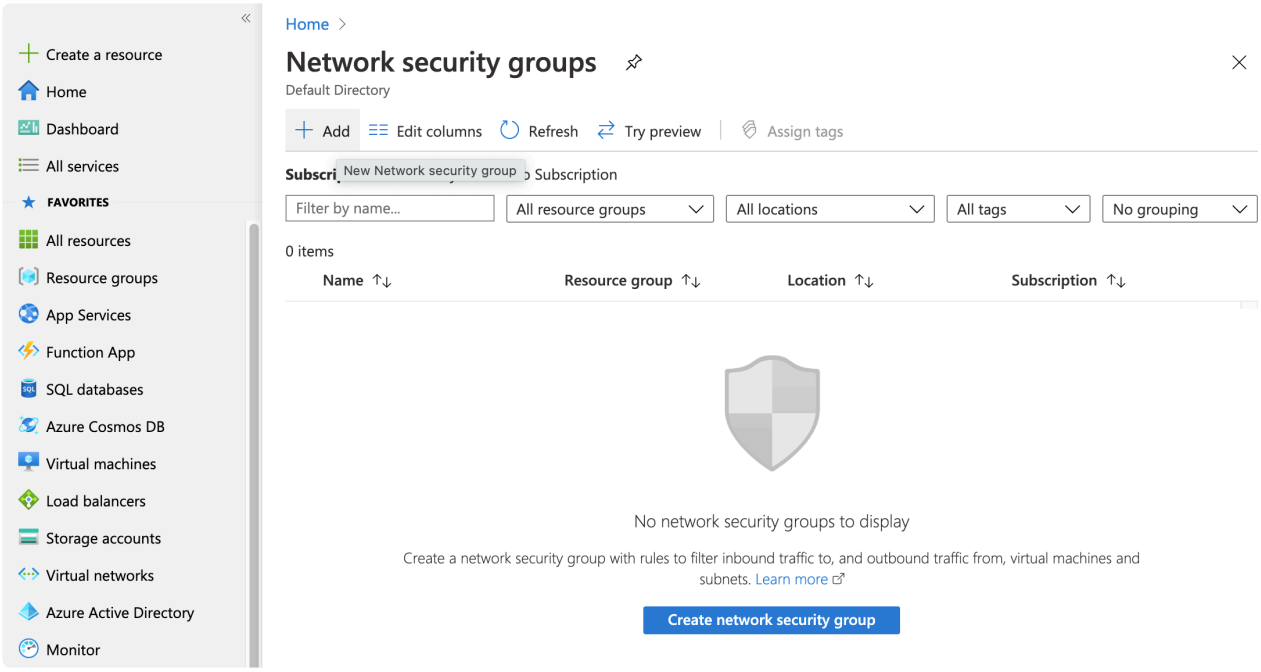
セキュリティグループはファイアーウォールとして機能し、「許可」と「拒否」の両方のルールを定義できます。ソースはインターネットアドレスまたはセキュリティグループにすることができ、セキュリティグループは仮想マシンまたはサブネットに割り当てることができます。
詳細については、 Azureのドキュメント を参照してください。

新しいセキュリティグループには、すでにいくつかの「許可」および「拒否」ルールがあります（これらは優先度65xxxで定義されています）。ここでは、以下のルールを追加して、アクセスを「許可」します（緑色の矢印を参照）。

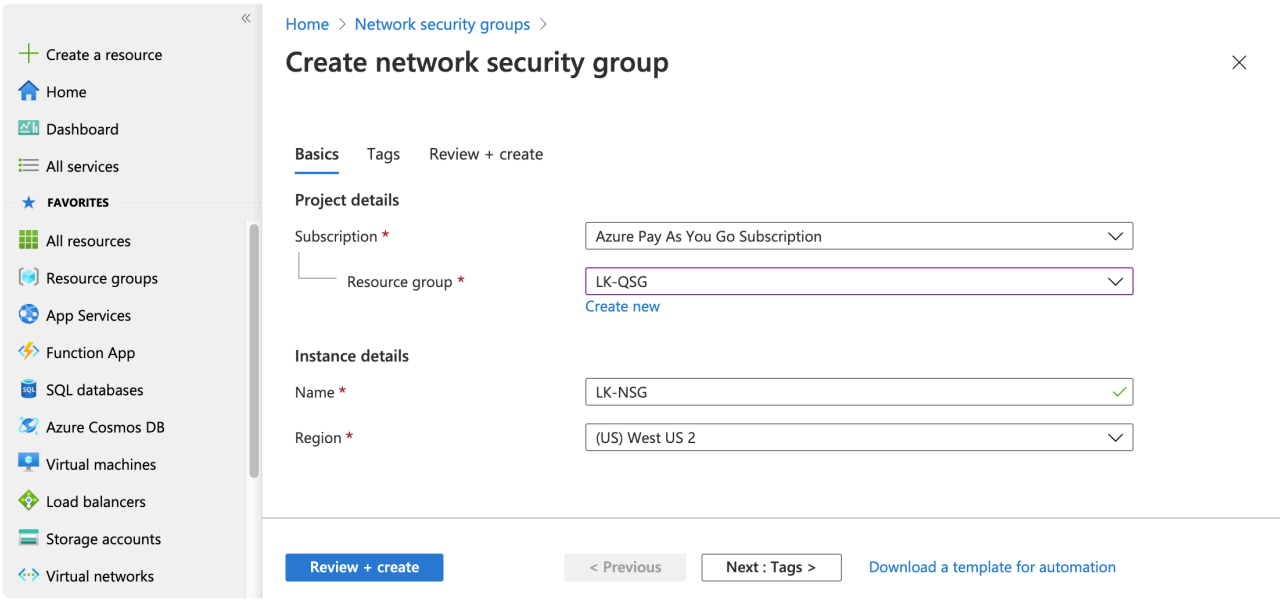
- 作業場所からLKサブネットへのトラフィックを許可する
- 仮想マシンから内部ロードバランサーへのトラフィックを許可する




1. ホーム画面から [ネットワークセキュリティグループ] を選択し、[追加] をクリックして新しいグループを作成します。

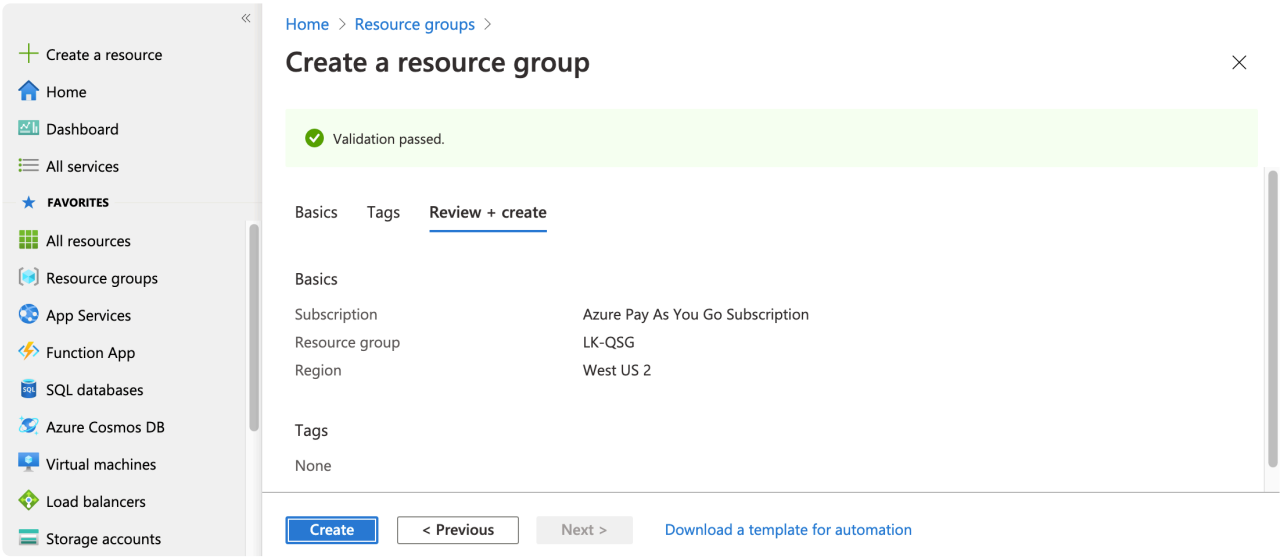



2. ソースグループ LK-QSG を選択し、新しいセキュリティグループの名前を LK-NSGと指定してから、リージョンを選択します。

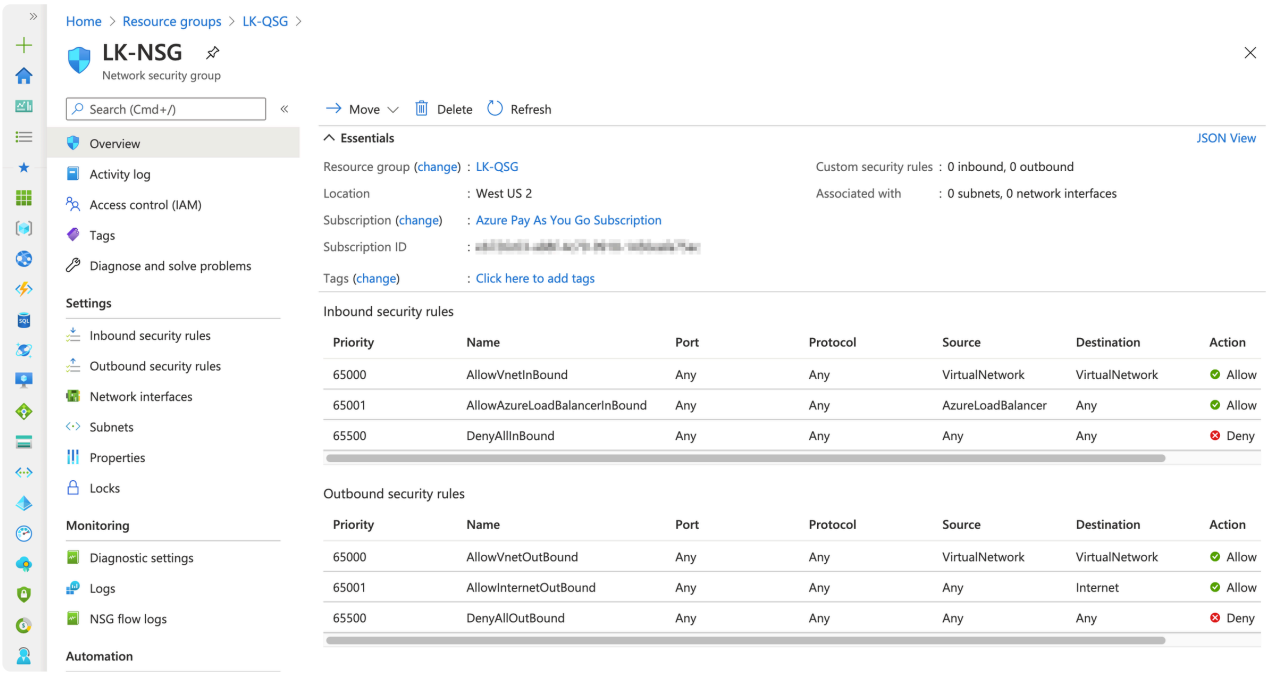



値を指定したら、[確認と作成] をクリックします。

3. ウィザードが入力を検証したら、[作成] をクリックします

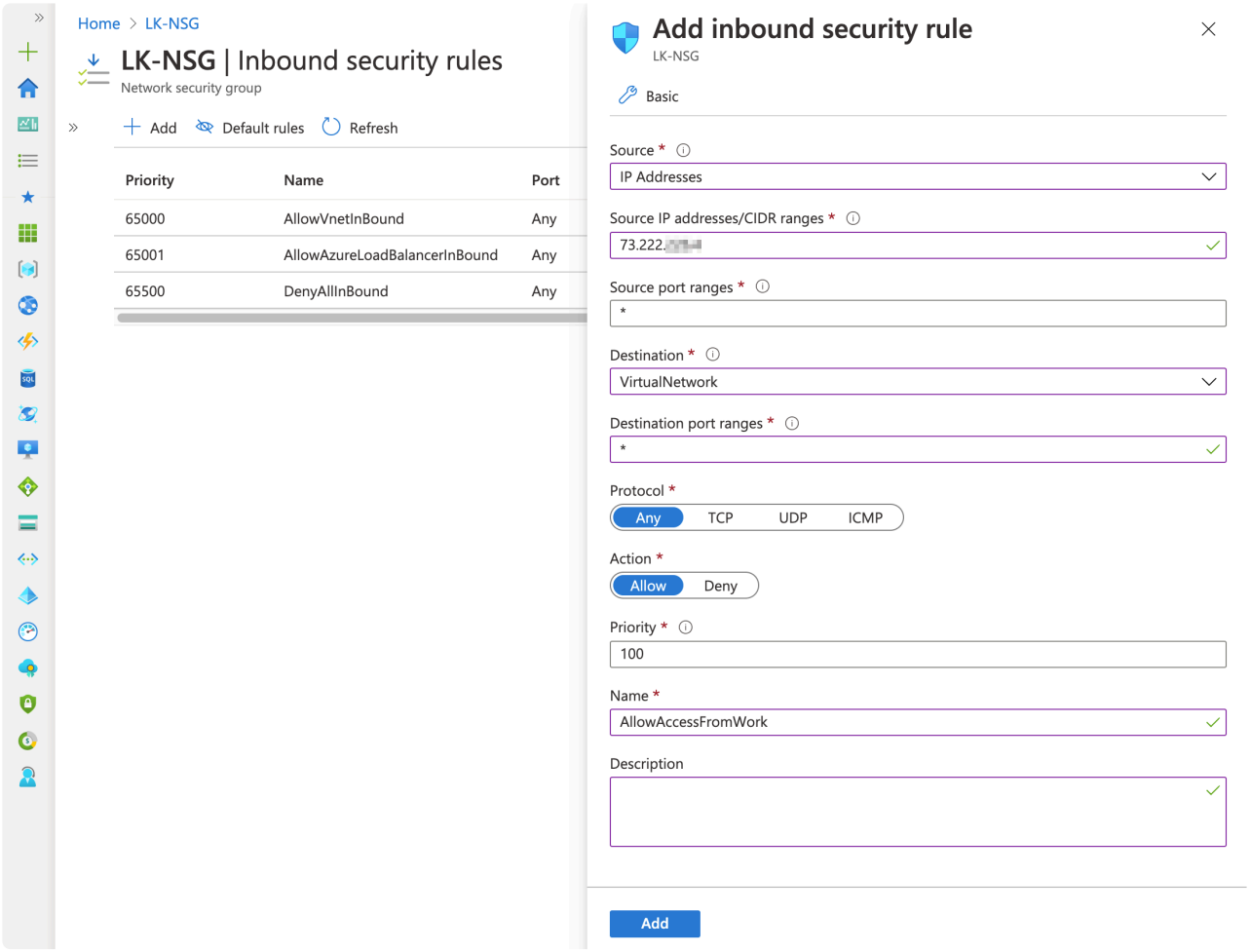



4. 新しいセキュリティグループ LK-NSG が作成されました。すでにいくつかのインバウンドルールとアウトバウンドルールが定義されていることがわかります。




5. 左側からインバウンドセキュリティルールを選択し、[追加] をクリックして新しいルールを入力します。次の値を入力します。

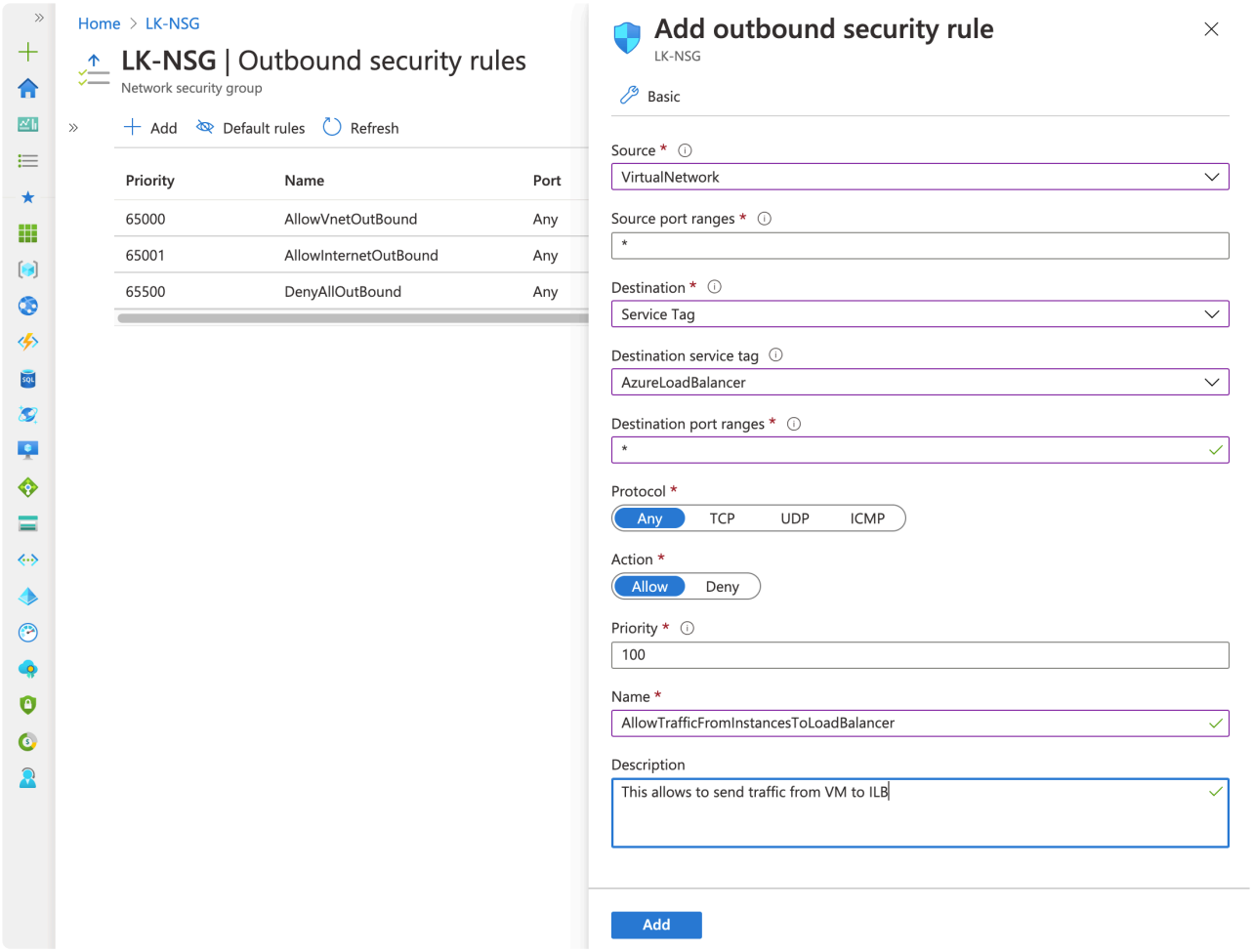
- ソースIP: Your WAN IP address(es)
- ソースのポート範囲: *
- 宛先: 仮想ネットワーク
- 宛先のポート範囲: *
- 名前: AllowAccessFromWork（この構成を表す任意の名前で構いません）




値を入力したら、[追加] をクリックします。

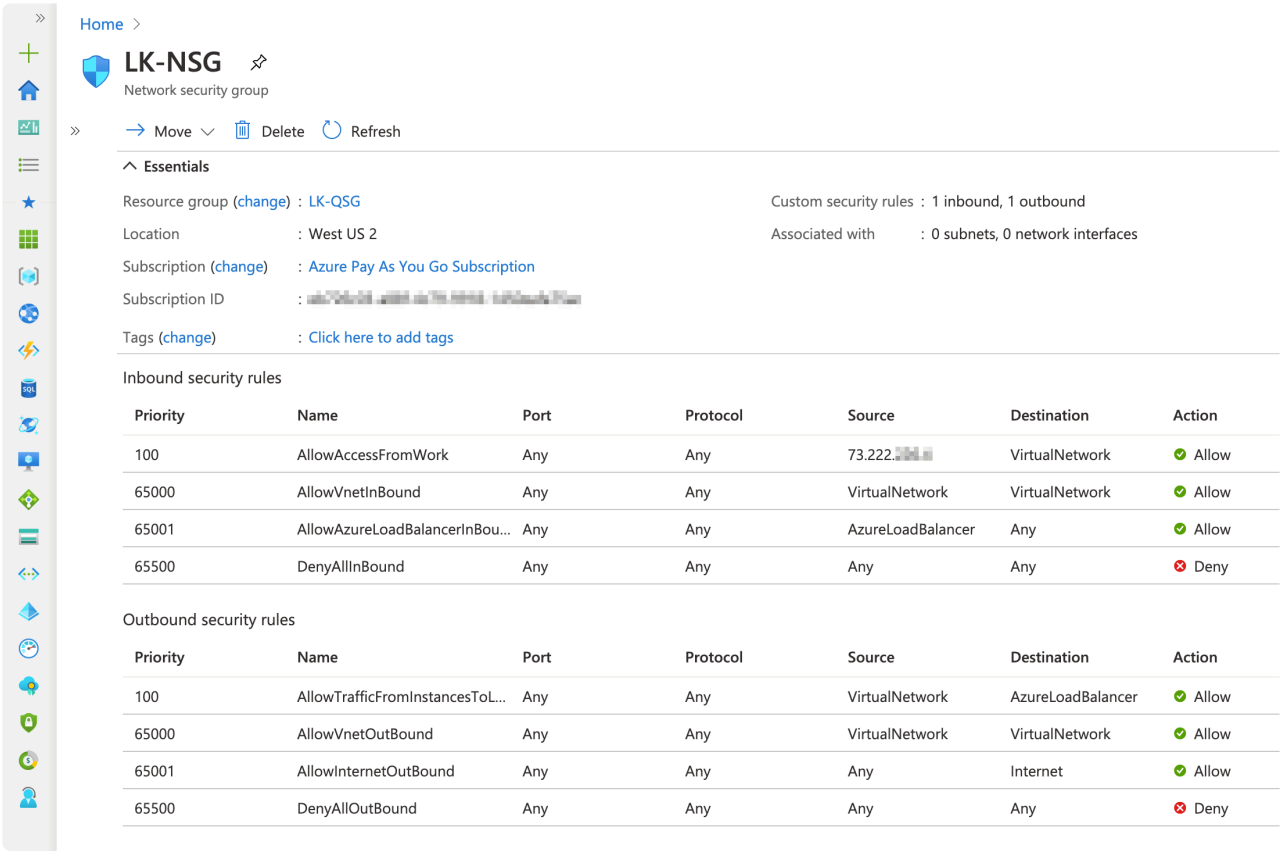
6. 左側から [アウトバウンドセキュリティルール] を選択し、[追加] をクリックして新しいルールを入力します。以下の値を入力します。

- ソース: VirtualNetwork
- ソースのポート範囲: *
- 宛先: Service Tag
- 宛先サービスタグ: AzureLoadBalancer
- 宛先のポート範囲: *
- 名前: AllowTrafficFromInstancesToLoadBalancer（この構成を表す任意の名前で構いません）




値を入力したら、[追加] をクリックします。

7. これで、新しいセキュリティグループが定義されました。



Home >

# LK-NSG

Network security group

Move | Delete | Refresh

**Essentials**

| | | | |
|---|---|---|---|
| Resource group (change) | : LK-QSG | Custom security rules | : 1 inbound, 1 outbound |
| Location | : West US 2 | Associated with | : 0 subnets, 0 network interfaces |
| Subscription (change) | : Azure Pay As You Go Subscription | | |
| Subscription ID | : [redacted] | | |
| Tags (change) | : Click here to add tags | | |

Inbound security rules

| Priority | Name | Port | Protocol | Source | Destination | Action |
|---|---|---|---|---|---|---|
| 100 | AllowAccessFromWork | Any | Any | 73.222.[redacted] | VirtualNetwork | Allow |
| 65000 | AllowVnetInBound | Any | Any | VirtualNetwork | VirtualNetwork | Allow |
| 65001 | AllowAzureLoadBalancerInBou... | Any | Any | AzureLoadBalancer | Any | Allow |
| 65500 | DenyAllInBound | Any | Any | Any | Any | Deny |

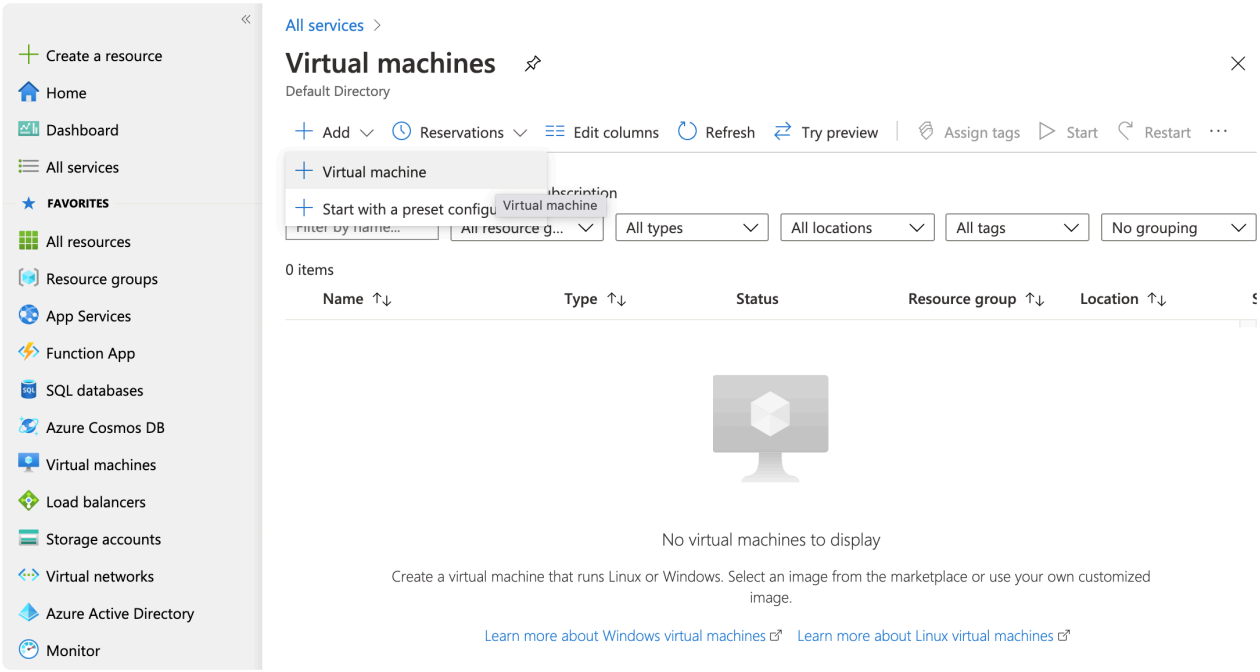
Outbound security rules

| Priority | Name | Port | Protocol | Source | Destination | Action |
|---|---|---|---|---|---|---|
| 100 | AllowTrafficFromInstancesToL... | Any | Any | VirtualNetwork | AzureLoadBalancer | Allow |
| 65000 | AllowVnetOutBound | Any | Any | VirtualNetwork | VirtualNetwork | Allow |
| 65001 | AllowInternetOutBound | Any | Any | Any | Internet | Allow |
| 65500 | DenyAllOutBound | Any | Any | Any | Any | Deny |

# 11.2.3.4.6. 1つ目のAzure仮想マシンの作成

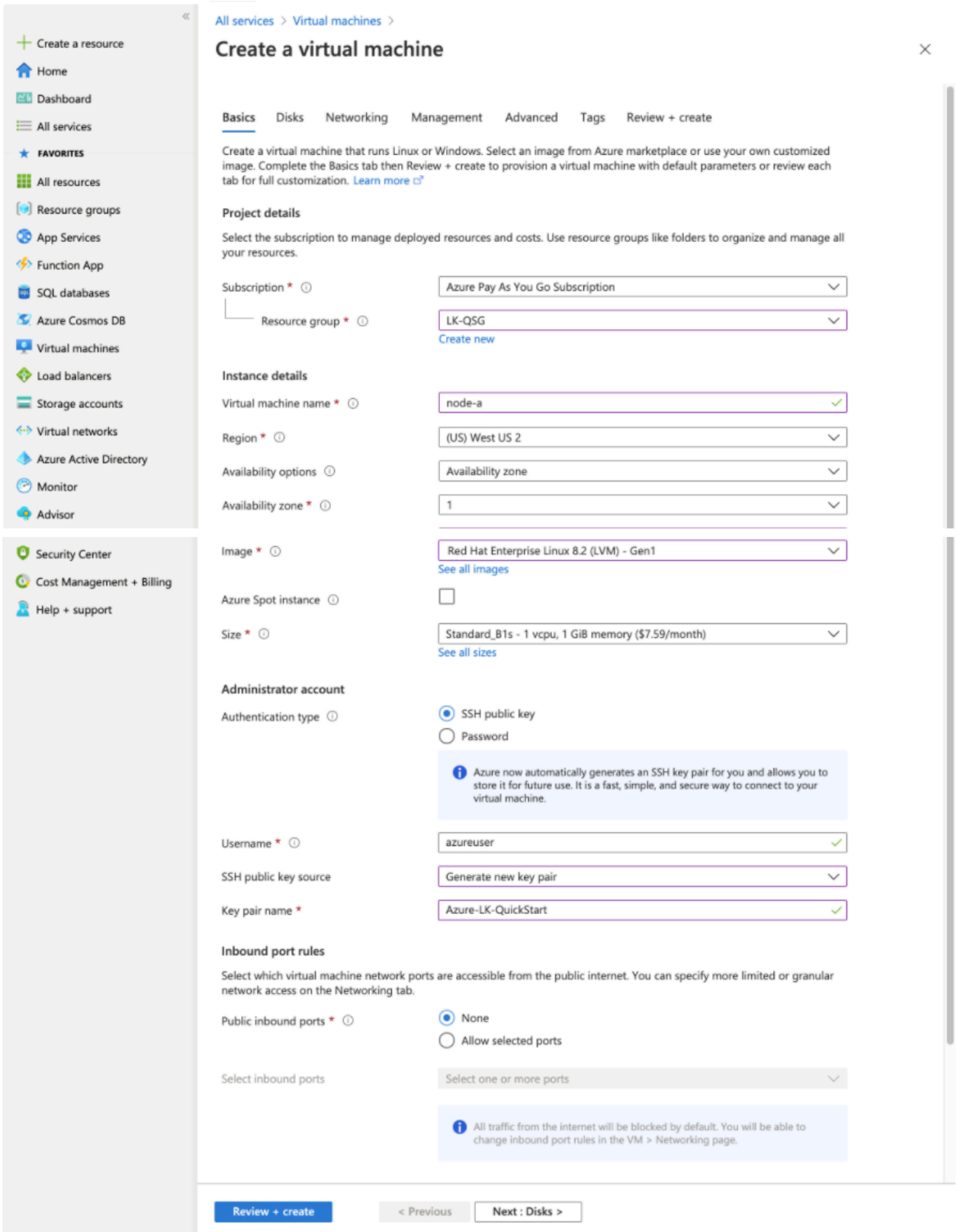
前のセクションでは、ネットワークの構成について説明しました。次に、最初のインスタンスを作成します。このチュートリアルで使用するコンピューティングリソース で説明したように、2つのディスクが必要です。このセクションでは、2番目のディスクを作成する方法についても説明します。

1. 「仮想マシン」に移動し、[追加] をクリックして新しいVMを作成します。



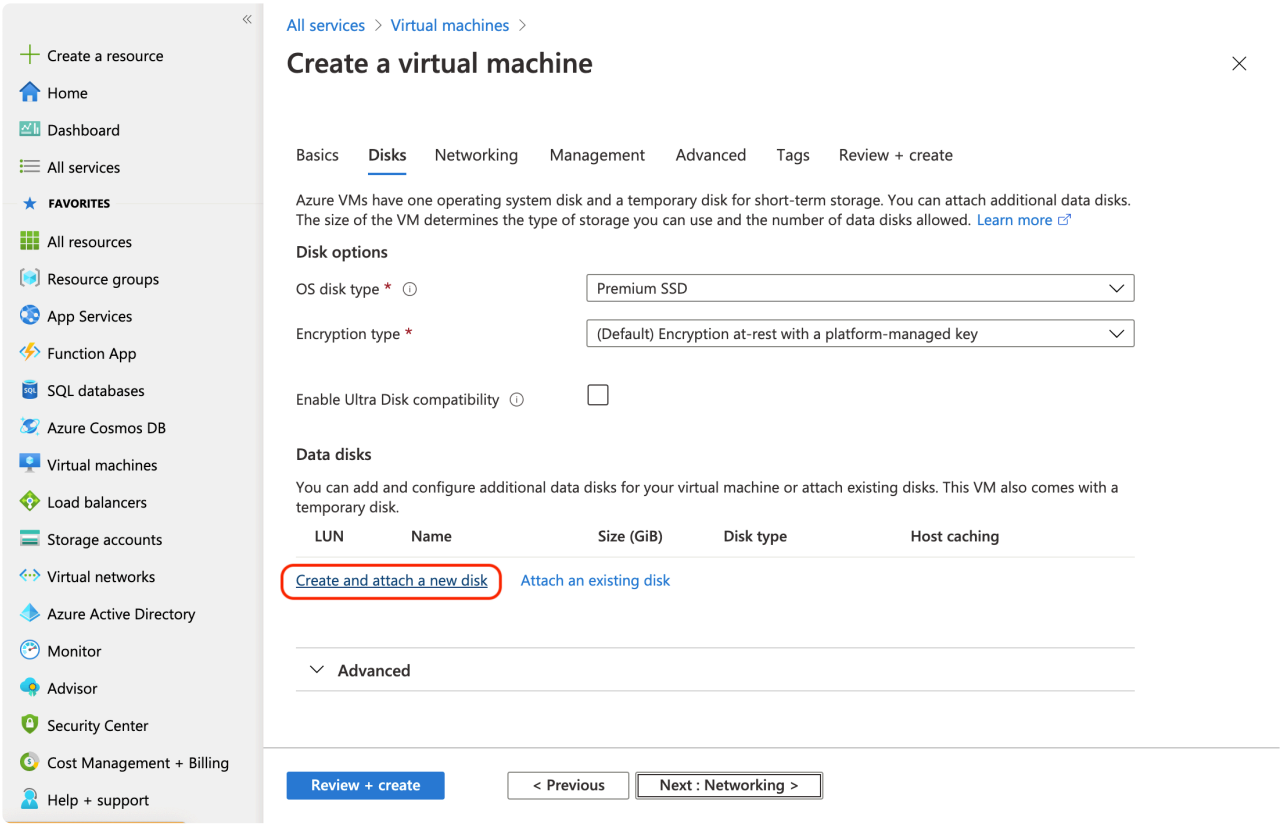

2. 以下の値を入力します。

- リソースグループ: `LK-QSG`
- 仮想マシン名: `node-a` （2番目または3番目のノードを作成する場合は、node-b、node-cになります）
- リージョン: 先ほどの判断で地域を指定する
- 可用性オプション: アベイラビリティゾーン
- 可用性ゾーン: `1` (2つ目、3つ目のノードを作成する場合は `2` および `3` になります。)
- イメージ: 対応OSを選択する
- サイズ: 保護するアプリケーションに応じて、イメージサイズを選択することができます。最小要件は `Standard_B1s` です。
- 認証タイプ: 各オプションを選択することができます。一般的に、クラウド事業者は `SSH 公開鍵` の利用を推奨しています。
- ユーザー名: `azureuser` (デフォルト値のままでも問題ありません。)
- SSH公開鍵ソース: キーを作成したことがない場合は、`Generate new key pair` を生成します。
- 鍵ペア名: 名前を指定します（このスクリーンショットでは `Azure-LK-QuickStart` を使用しています）。
- パブリックインバウンドポート: `None`



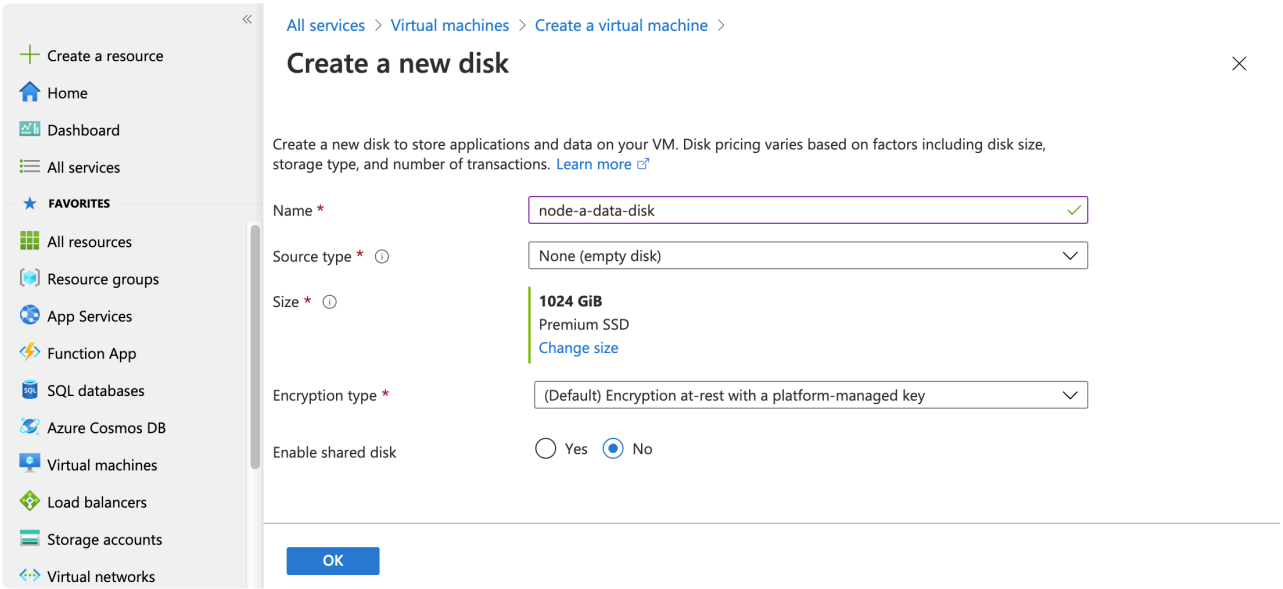

これらの値を入力または選択したら、[Next:Disks] をクリックします。

3. デフォルトでは、ウィザードはオペレーティングシステムを起動するためのディスクを1つ作成します。データレプリケーション用に、追加のディスクが必要です。



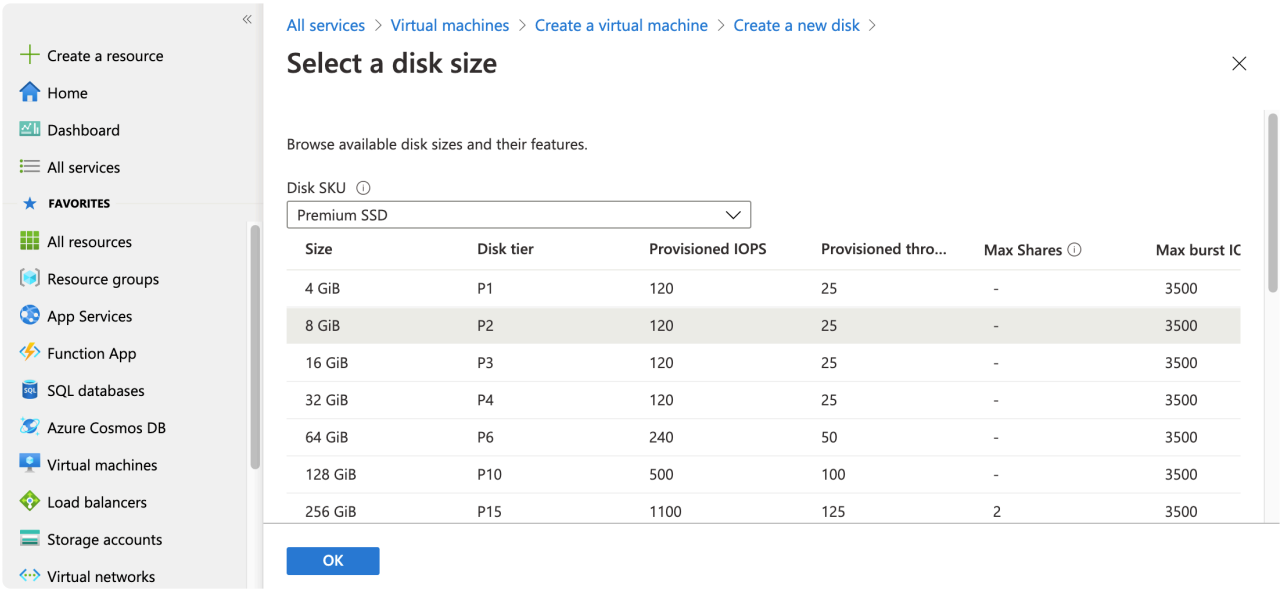

[新しいディスクを作成し接続する] をクリックします。

4. ディスクサイズを選択します。SIOS Protection Suiteを評価するために、最低でも8GiBのディスクを作成してください。

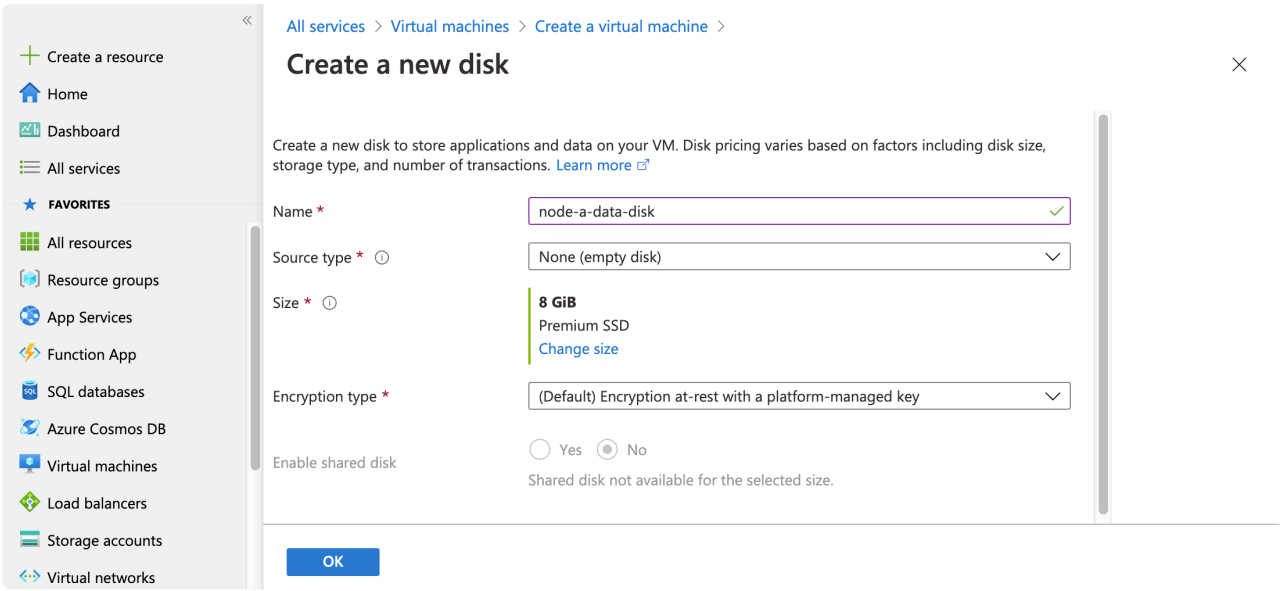



[サイズ変更] をクリックしてディスクサイズを指定します。

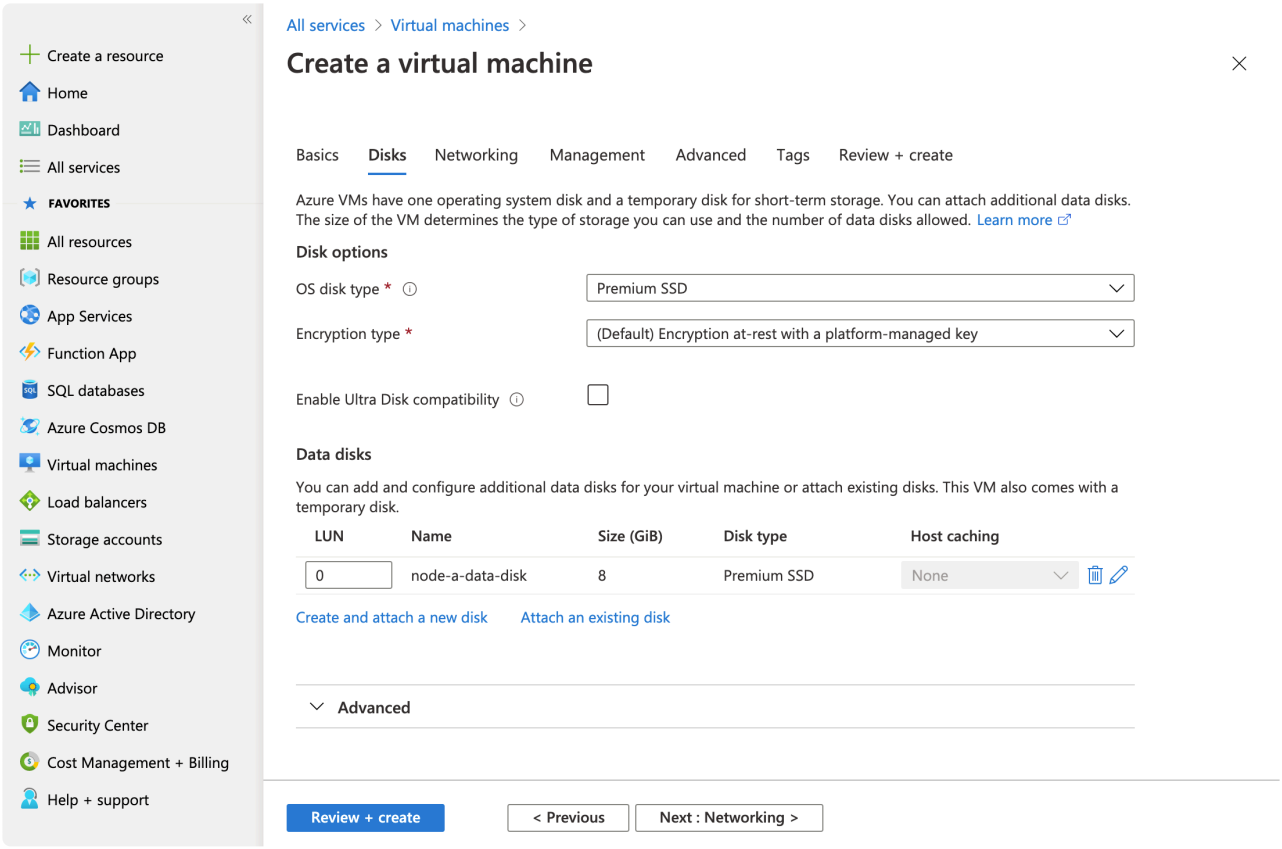
5. ディスクサイズを選択し（SIOS Protection Suiteを評価する場合は8GiB以上を選択）、[OK] を選択します。




6. これでディスクサイズが指定されました。[OK] をクリックして確定します。

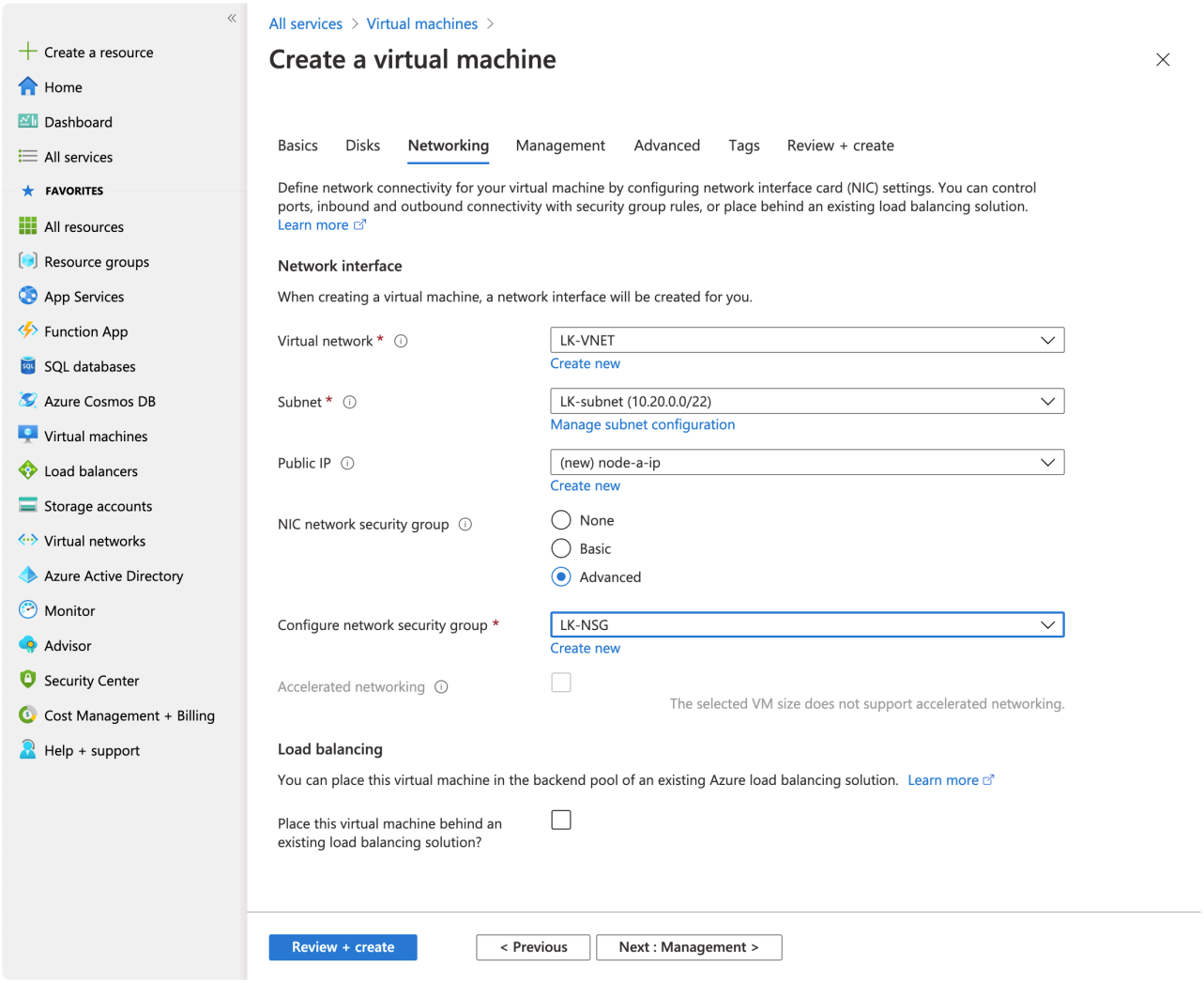



7. ディスクの構成が完了したら、[Next: Networking] をクリックします。



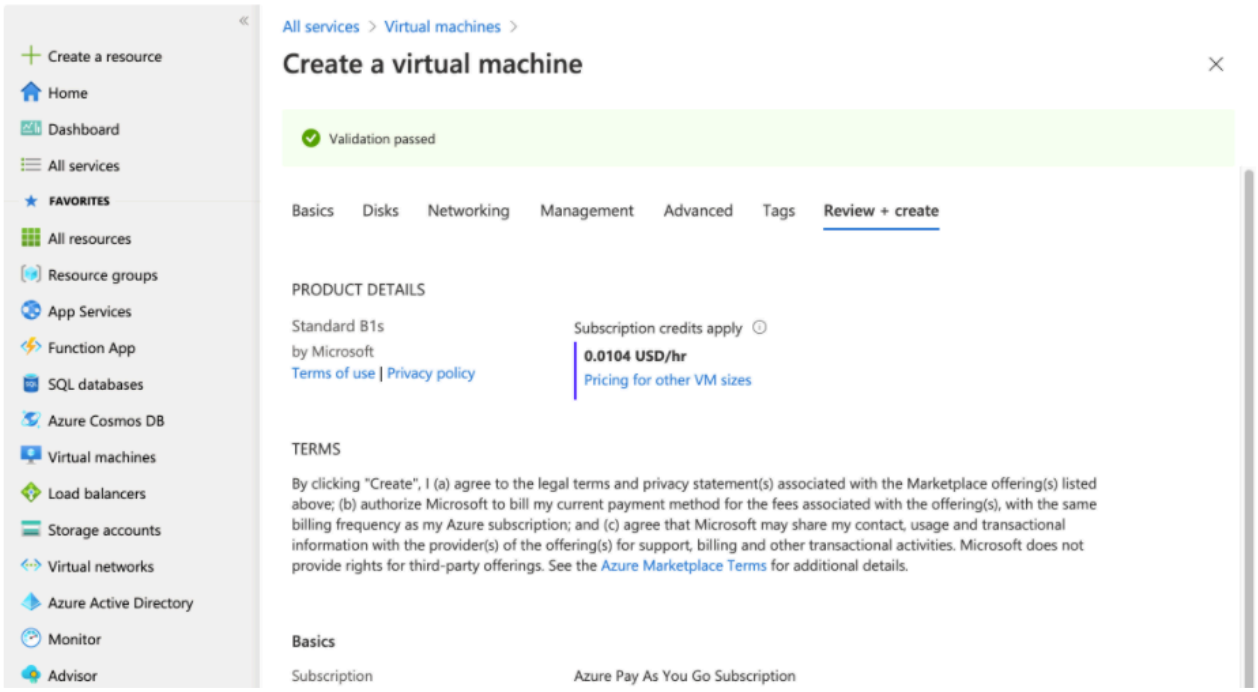

8. ネットワークに関する以下のパラメーターを選択します。

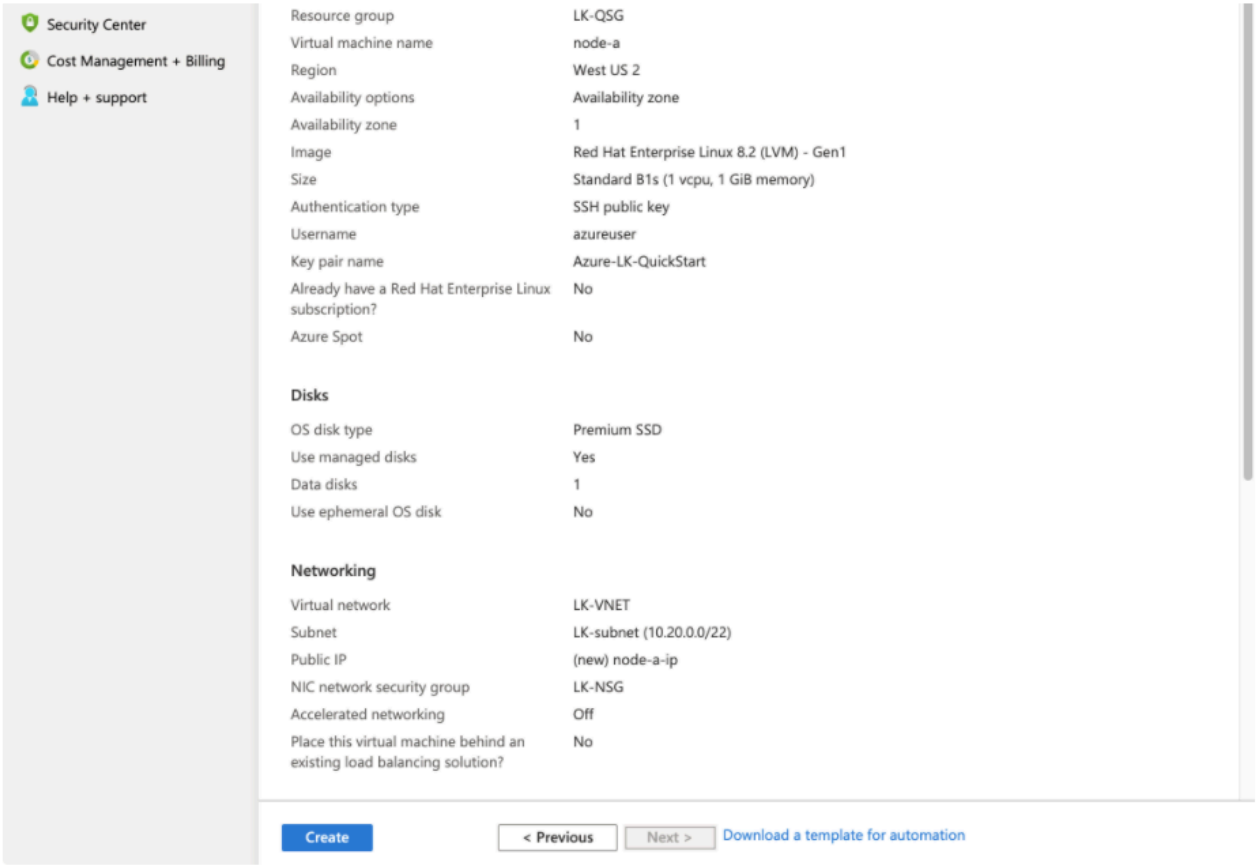
- 仮想ネットワーク: `LK-VNET`
- サブネット: `LK-subnet`
- public IP: `(new) node-a-ip`
- NICネットワークセキュリティグループ: `Advanced`
- ネットワークセキュリティグループ: `LK-NSG`



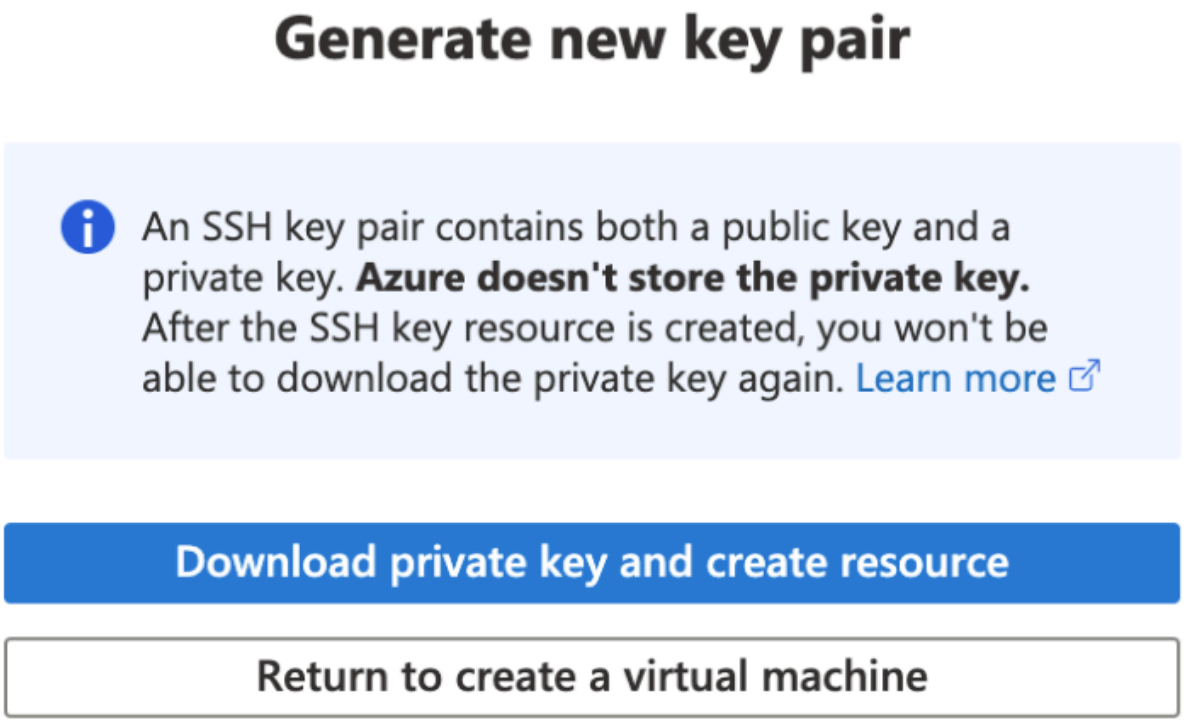

値を指定したら、[確認と作成] をクリックして完了します。

9. ウィザードが入力を検証したら、[作成] をクリックします。



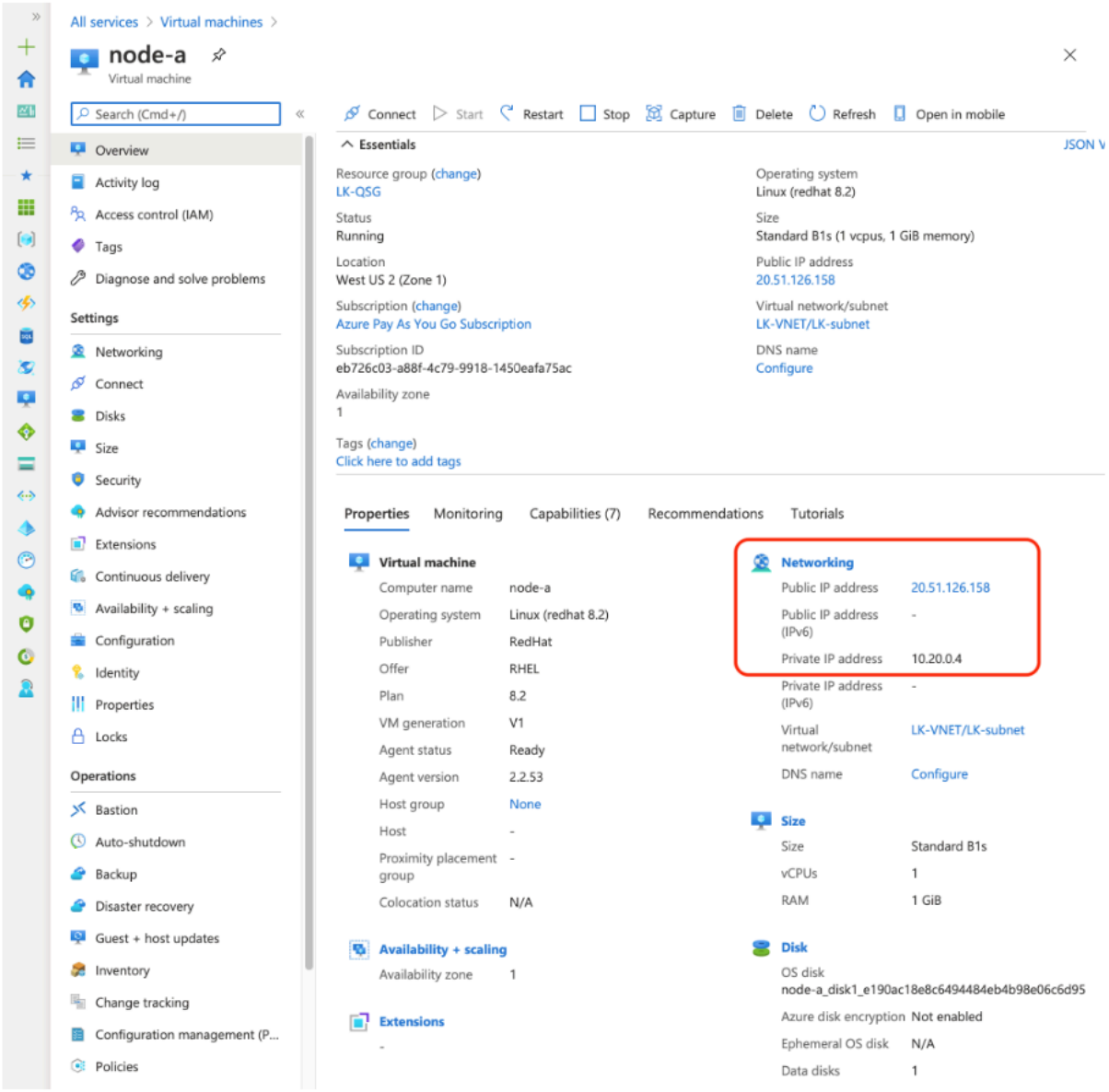



10. キーペアを初めて使用する場合は、秘密キーファイルをダウンロードします。後でインスタンスに接続する際に、キーファイルが必要になります。

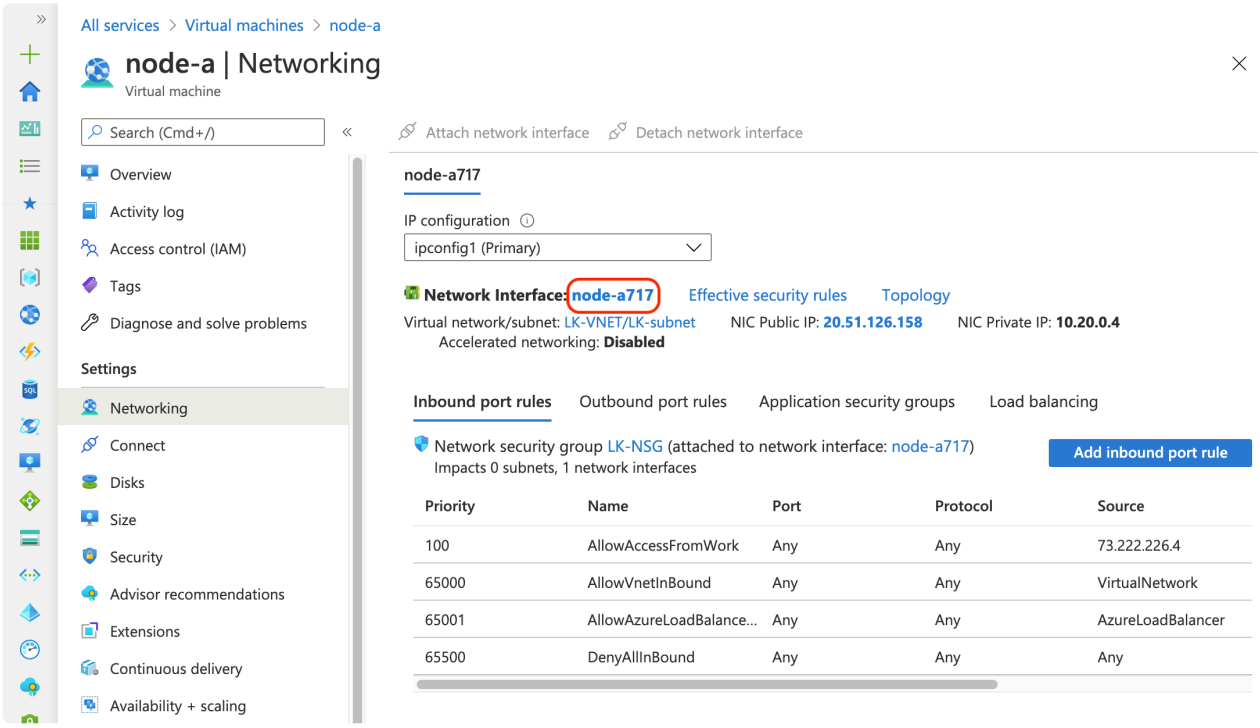



11. 仮想マシンが作成されると、VMの構成を確認できます。IPアドレスがまだ指定されておらず、

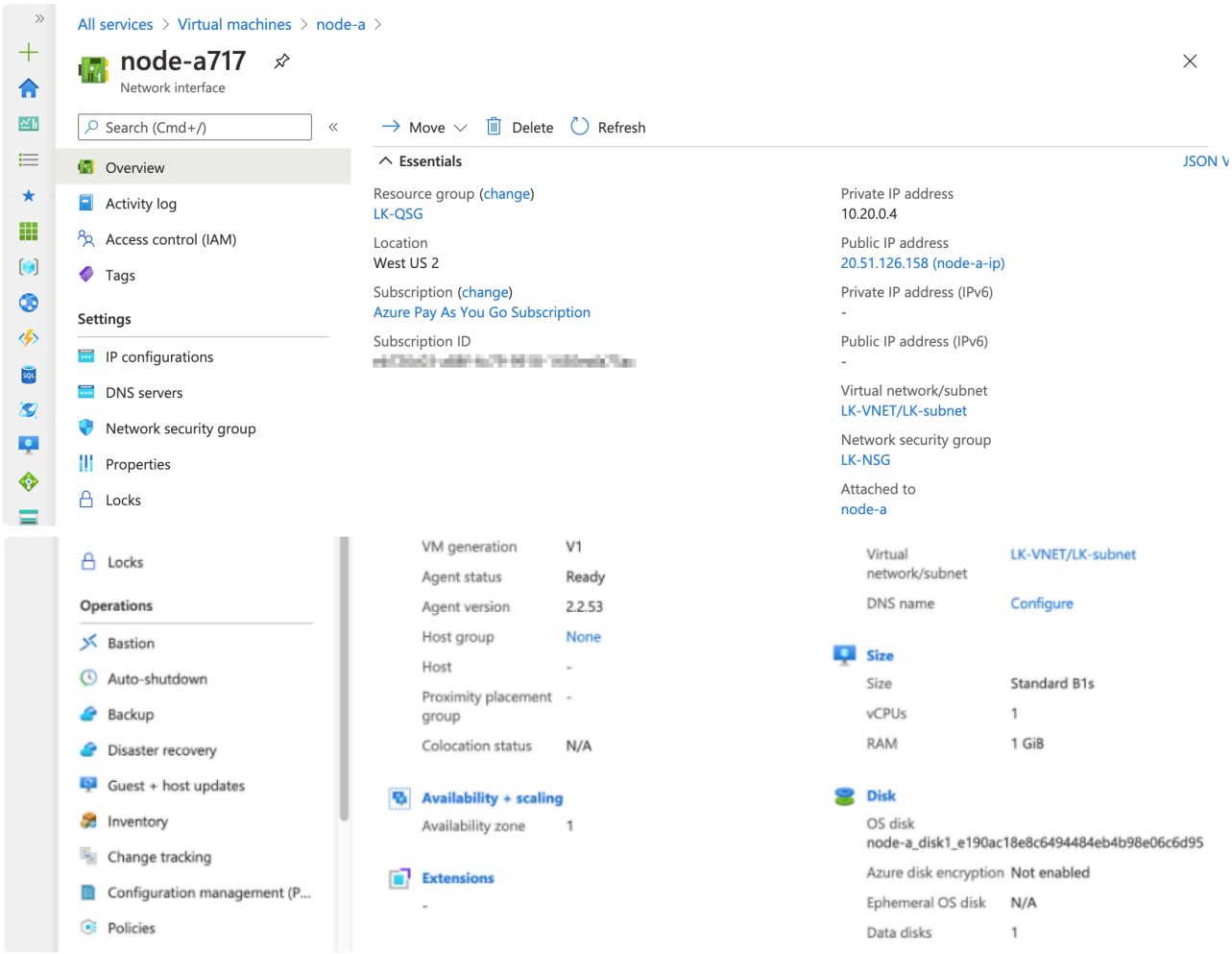
DHCPによって割り当てられたアドレス10.20.0.4が自動的に選択されていることがわかります。「ネットワーク」(青いリンク) をクリックして、ネットワーク設定を変更します。



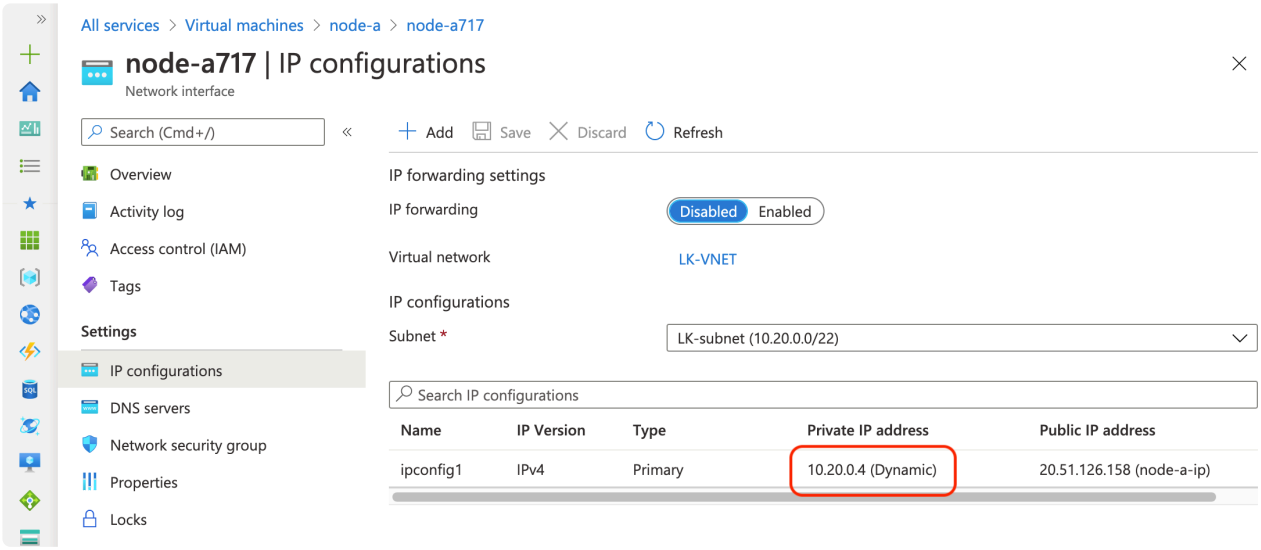

12. 下のスクリーンショットで、仮想マシンにネットワークインターフェイス（node-a717）があることがわかります。 `node-a717` をクリックして、このネットワークカードの設定を変更します。



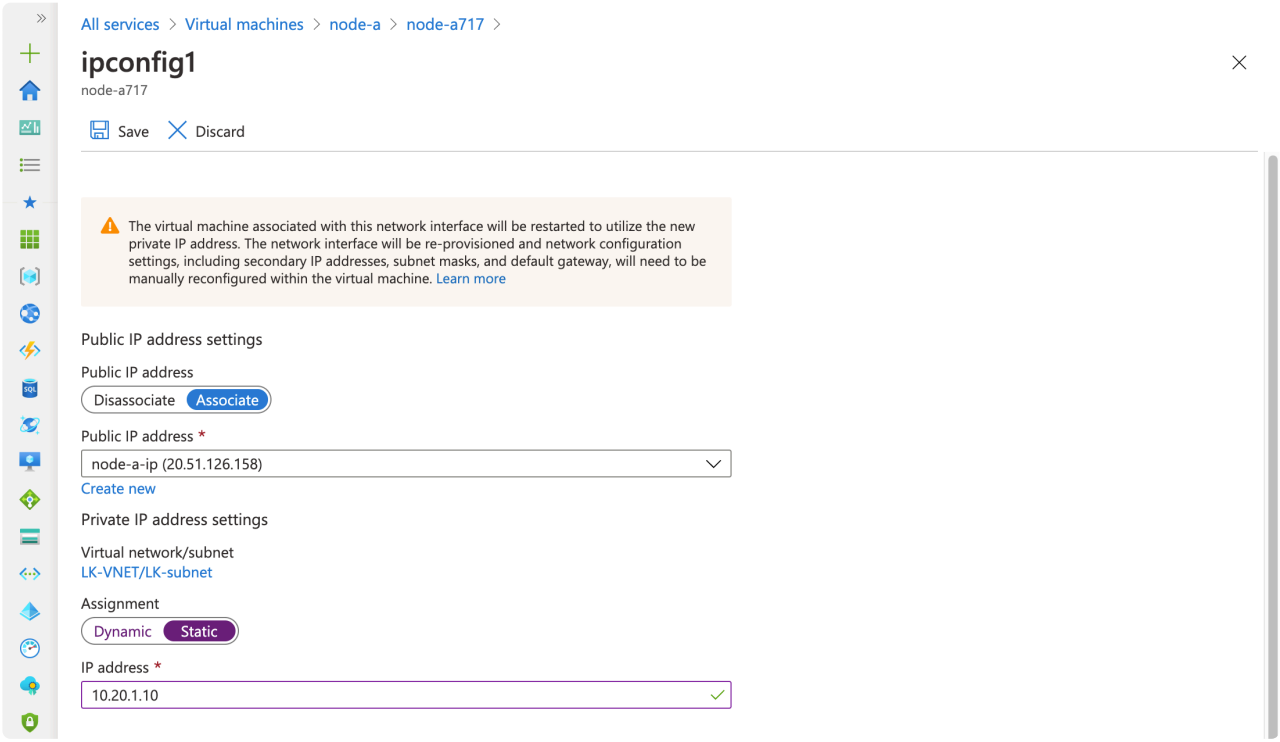

13. これで、ネットワークカードの詳細を確認できます。「IP構成」をクリックします。




14. IPアドレスを選択して構成ページに移動します。




15. 割り当てを `Dynamic` から `Static` に変更し、IPアドレスを10.20.1.10に設定します。




IPアドレスを指定したら、[保存] をクリックします。

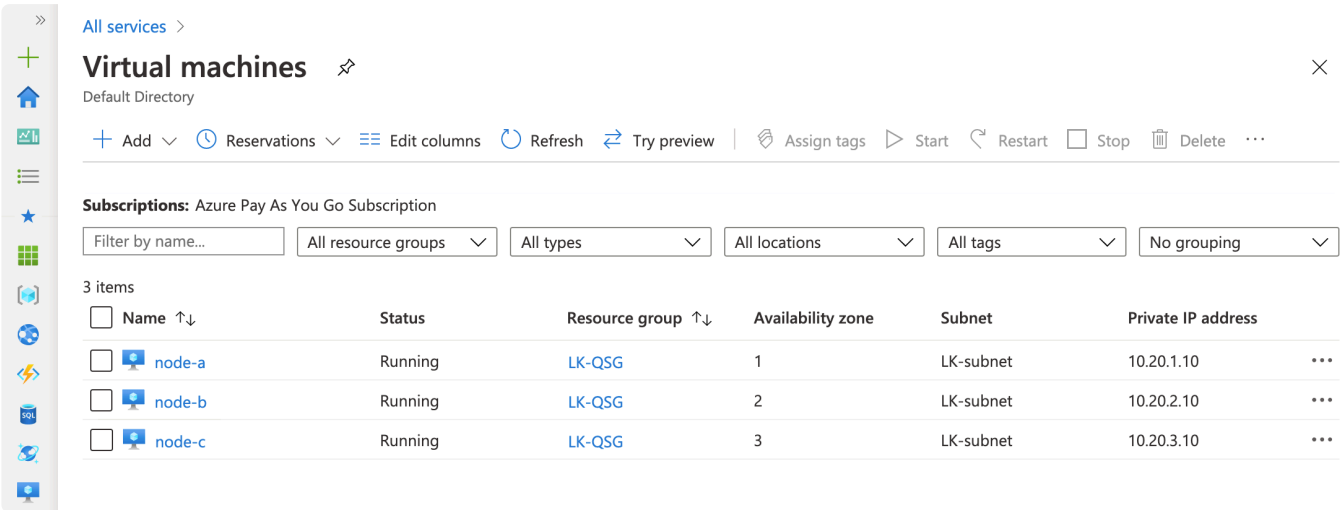
仮想マシンの設定はこれで完了です。

# 11.2.3.4.7. 2番目と3番目の仮想マシンの作成

2番目のノード（node-b）と3番目のノード（node-c）を作成するために必要な手順は、1番目のノード（node-a）について説明した手順とほぼ同じです。次の表は、3つのノードの違いと、これらのインスタンスを作成するために必要な詳細を示しています。

| Name | Parameter | Value |
|---|---|---|
| VM全体で共通の値 | リソースグループ | **LK-QSG** |
| | 仮想ネットワーク | **LK-VNET** |
| | サブネット | **LK-subnet** |
| | ネットワークセキュリティグループ | **LK-NSG** |
| node-a | 可用性ゾーン (*1) | 1 |
| | プライベートIPアドレス (*2) | `10.20.1.10` |
| | 2つ目のディスク（ストレージデバイス） | セカンドディスクが必要です |
| node-b | 可用性ゾーン (*1) | 2 |
| | プライベートIPアドレス (*2) | `10.20.2.10` |
| | 2つ目のディスク（ストレージデバイス） | セカンドディスクが必要です |
| node-c | 可用性ゾーン (*1) | 3 |
| | プライベートIPアドレス (*2) | `10.20.3.10` (node-c) |
| | 2つ目のディスク（ストレージデバイス） | 2番目のディスクを作成する必要はありません |

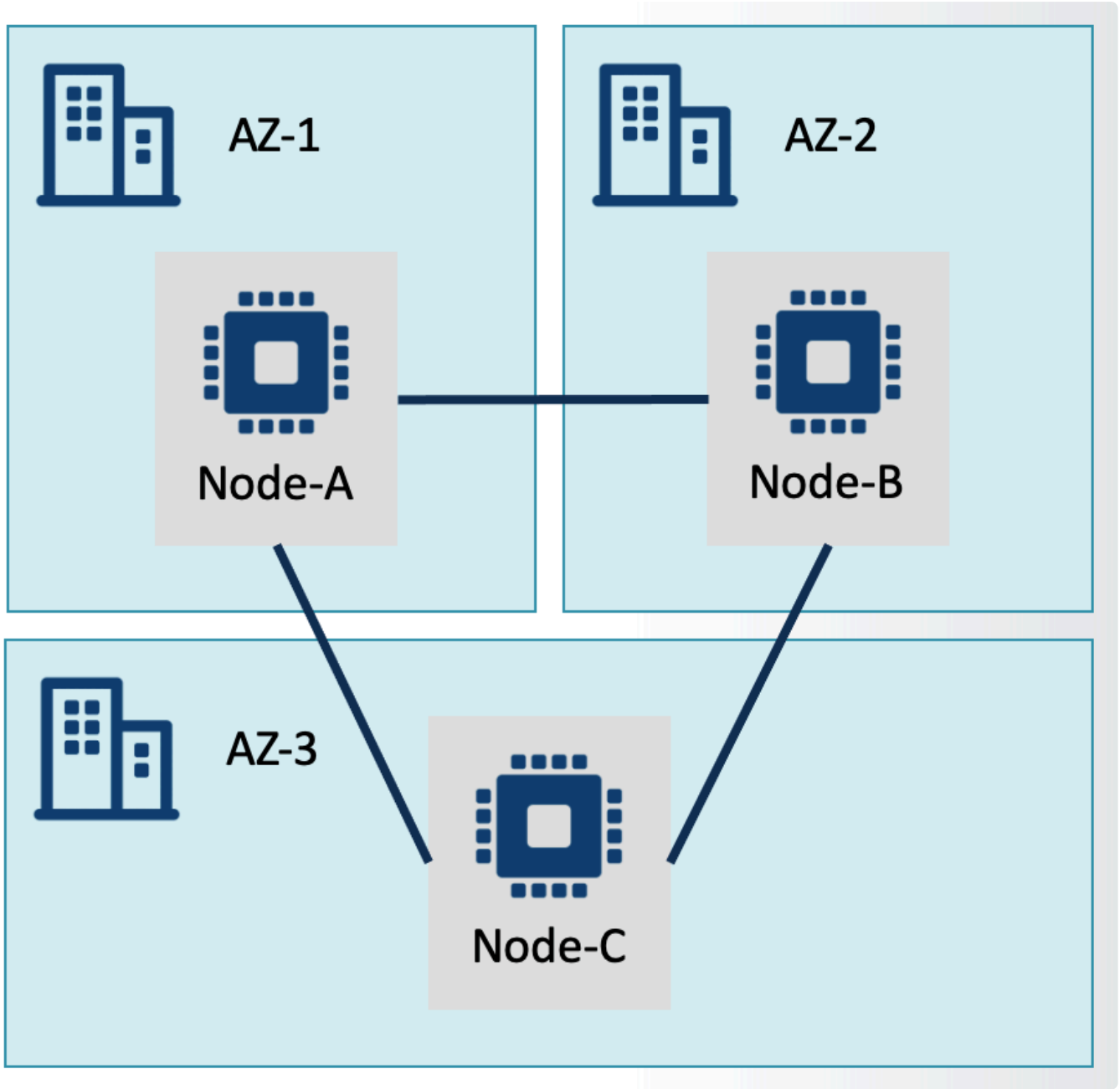
作成すると、インスタンスは次のようになります。



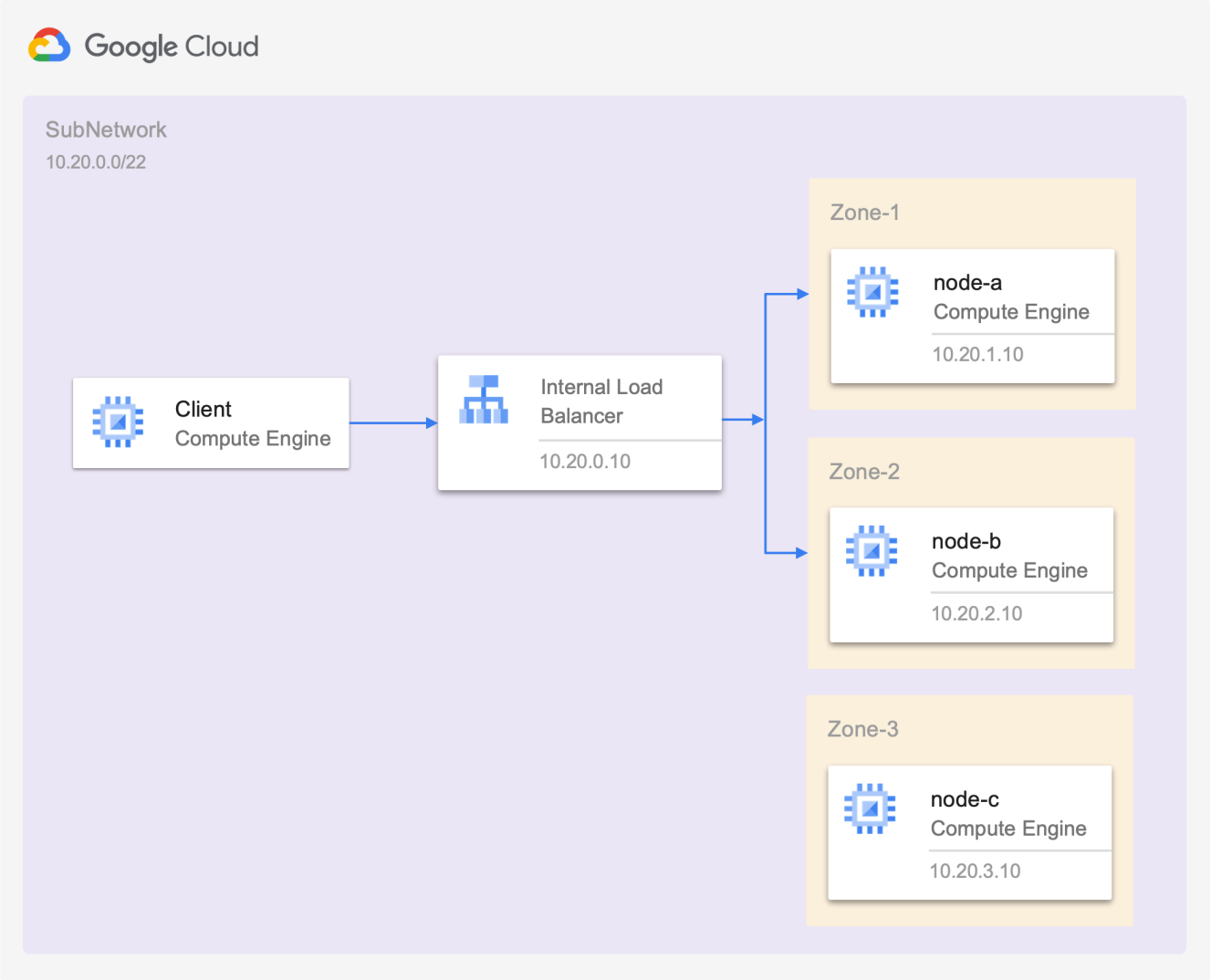
# 11.2.3.5. Google Cloud でゼロからインスタンスを作成する

**免責事項:** ユーザーインターフェースは地域によって異なる場合や、時間の経過とともに変更される場合があります。以下に示すスクリーンショットがお使いのものと異なる場合は、Google Cloudが提供するドキュメントを参照してください。

本評価ガイドでは、次のネットワーク構造とインスタンスを使用しています。




Google Cloudでは、これらのコンポーネントは次の図に示すように定義できます。




このセクションでは、内部ロードバランサーを除くこれらのコンポーネントを作成します。内部ロードバランサーは 後で 作成します。

この図に記載されているコンポーネントについて、次の表で説明します。

このネットワーク構造を作成するには、次のコンポーネントを実装する必要があります。

| コンポーネント | 名称 | パラメーター | 値 |
|---|---|---|---|
| VPC network | lk-vpc | Region | us-west1 (*1) |
| | | Subnet (lk-subnet) | 10.20.0.0/22 |
| Virtual Machine | VM全体で共通の値 | **VPC network** | **lk-vpc** |
| | | **Subnet** | **lk-subnet** |
| | | **Network Tags** | **lk-node** |
| | | **External IP** | **Ephemeral** |
| | node-a | Zone | us-west1-a (*1) |

| | | | |
|---|---|---|---|
| | | Hostname | node-a.internal |
| | | Internal IP | `10.20.1.10` |
| | node-b | Zone | us-west1-b (*1) |
| | | Hostname | node-b.internal |
| | | Internal IP | `10.20.2.10` |
| | node-c | Zone | us-west1-c (*1) |
| | | Hostname | node-b.internal |
| | | Internal IP | `10.20.3.10` |
| Firewall Rules | ルール全体で共通の値 | **Network** | **lk-vpc** |
| | | **Type** | **Ingress** |
| | | **Target** | **lk-node** |
| | | **Action** | **Allow** |
| | fw-allow-ssh | Source | WAN IP Address of work location |
| | | Protocols / ports | all |
| | fw-allow-lk-node-connection | Source | Tags: lk-node |
| | | Protocols / ports | all |
| | fw-allow-health-check (*2) | Source | 130.211.0.0/22, 35.191.0.0/16 |
| | | Protocols / ports | all |

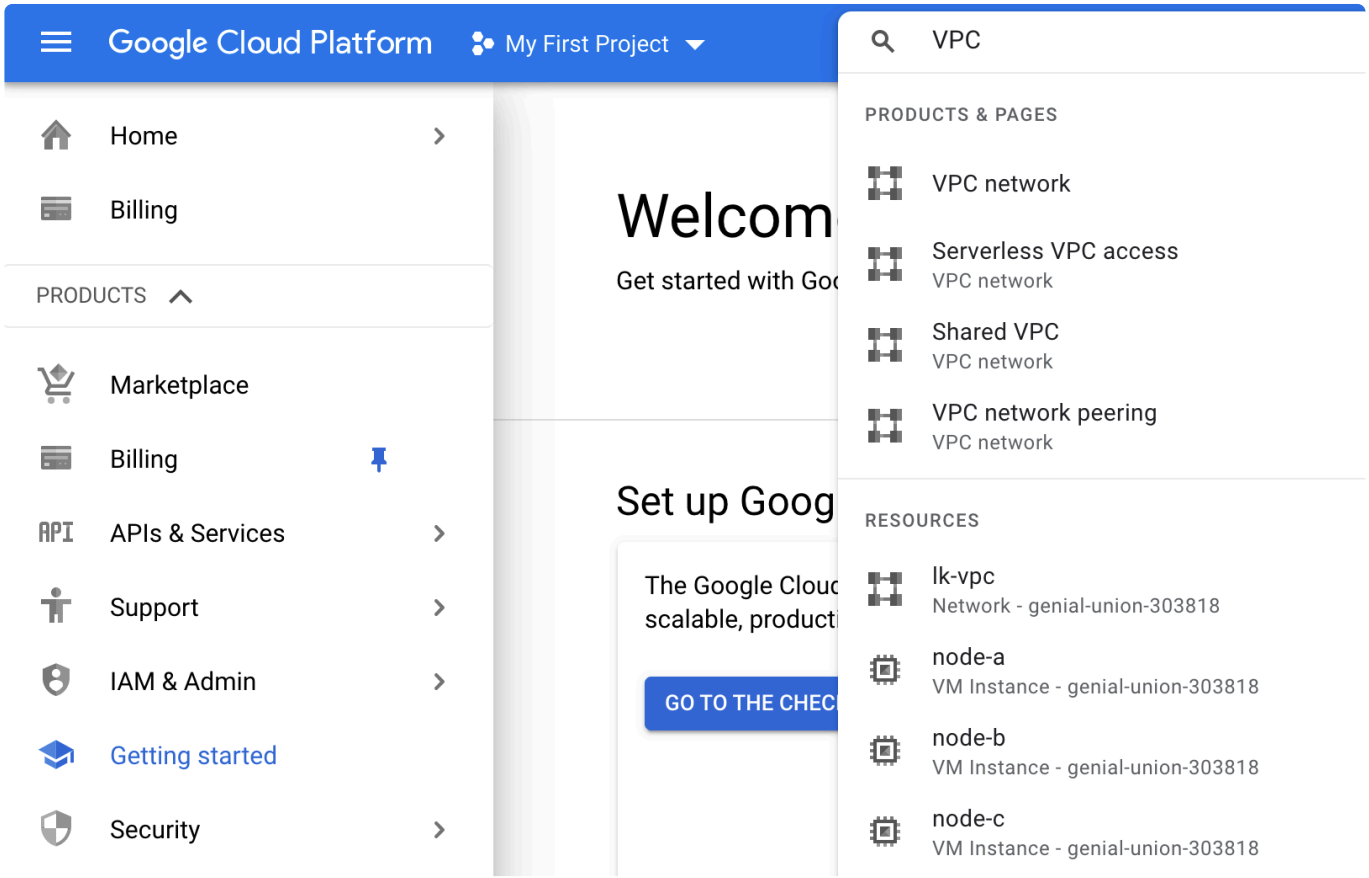
- (*1) この例では、Regionとしてus-west1を使用しています。
- (*2) これは、内部ロードバランサーが各ノードのステータスをチェックできるようにするために必須です。

以下のセクションでは、これらのネットワークコンポーネントを作成するために必要な手順を、最初のインスタンスから順に説明します。

- Google Cloudサービスの切り替え
- Google Cloudリージョンの決定
- プロジェクトの作成
- VPCネットワークの作成
- 新しいSSH認証鍵の作成
- 1つ目のGoogle Cloud VMの作成
- ファイアーウォールルールの設定
- 2番目と3番目の仮想マシンの作成

# 11.2.3.5.1. Google Cloudサービス間の切り替え

Google Cloudでは、利用可能なサービスの数が大幅に増え続けています。別のサービスに移動するには、Google Cloudコンソールの上部にある検索ボックスにサービス名を入力すると便利です。リストからサービスを選択します。以下のスクリーンショットは、VPCネットワークサービスを選択する方法を示しています。



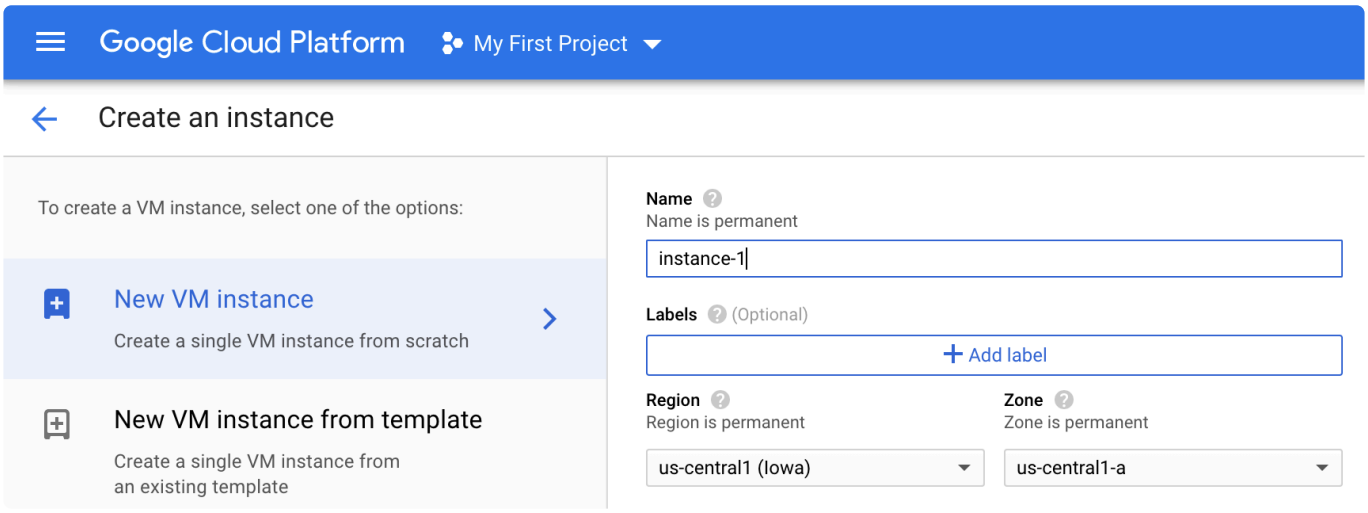
# 11.2.3.5.2. Google Cloud リージョンの決定

Google Cloud には地理的に多くの場所にリージョンがあり、職場から地理的に近いリージョンを選択することが有益な場合があります。各リージョンには 3 つ以上のゾーンがありますが、使用可能なマシンタイプはリージョン間で若干異なります。

クロスリージョンネットワークのフェールオーバーはサポートされていません。

詳細については、Google Cloudのドキュメント を参照してください。

VMを作成する際に、リージョンとゾーンを選択します。VMの作成後は、これらを変更することはできません。



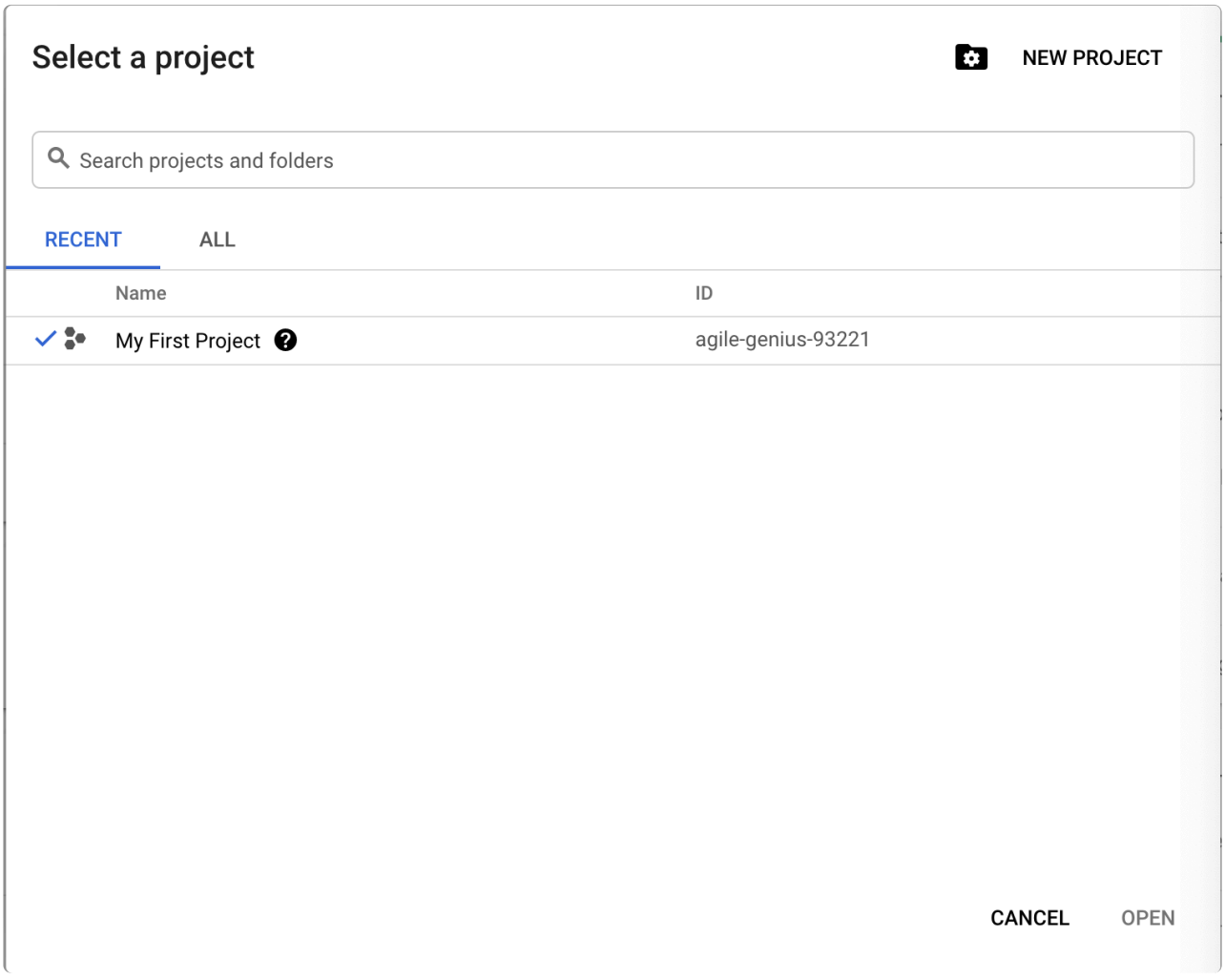
# 11.2.3.5.3. プロジェクトの作成

プロジェクトとは、ネットワークコンポーネント、コンピューターリソース、ストレージなどのリソースを作成できる論理コンテナです。
詳細については、Google Cloudのドキュメント を参照してください。

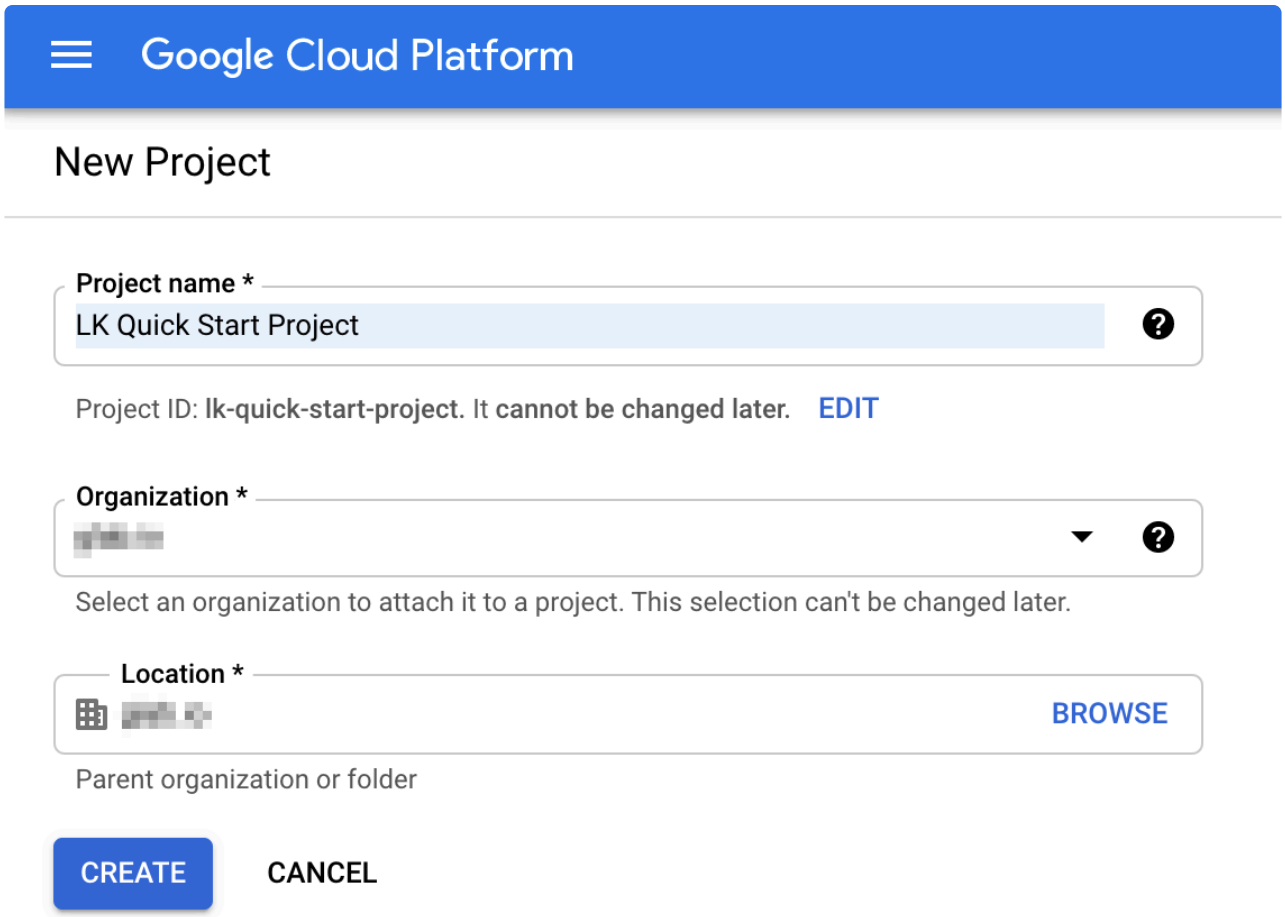
Google Cloudにサインアップしたばかりの場合は、「My First Project」というデフォルトのプロジェクトがあり、このプロジェクトを使用できます。ただし、次のように新しいプロジェクト（ここではLK Quick Start Projectという名前を付けます）を作成することもできます。

1. 上部にある「My First Project」（または作業中の他のアクティブなプロジェクト）を選択します。

2. [プロジェクトの選択] 画面で、右上の [新規プロジェクト] をクリックします。

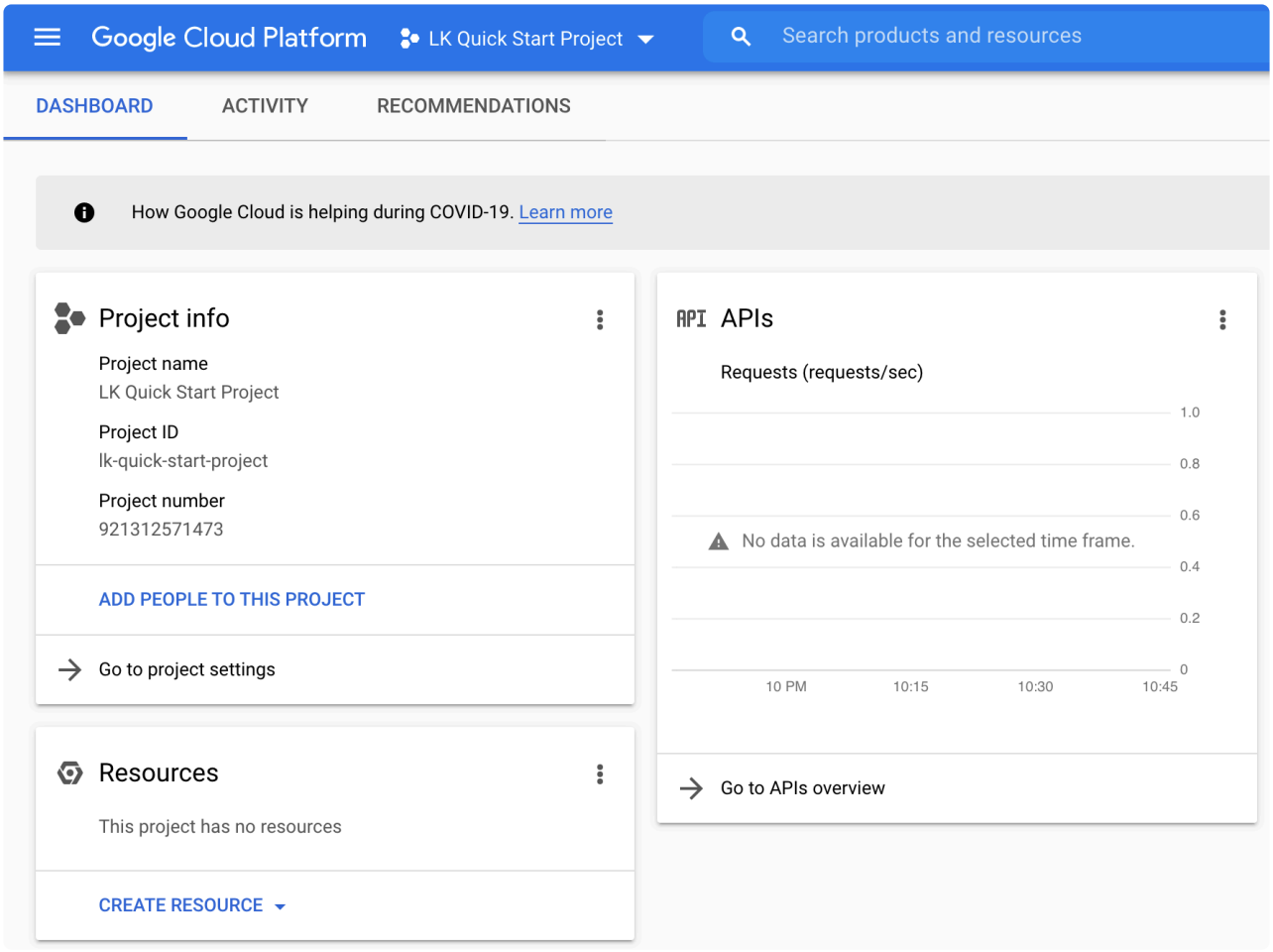


3. プロジェクトの名前を LK Quick Start Projectと指定します。




項目を入力したら、「作成」をクリックします。

これで、新しいプロジェクトが作成されました。

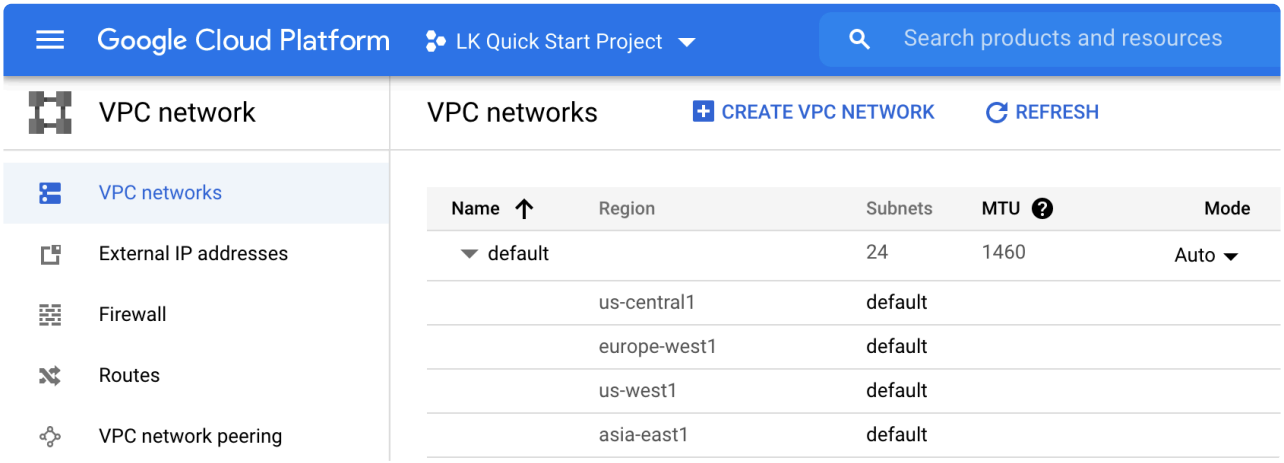

Google Cloud Platform
LK Quick Start Project
Search products and resources
DASHBOARD
ACTIVITY
RECOMMENDATIONS
How Google Cloud is helping during COVID-19. Learn more
Project info
Project name
LK Quick Start Project
Project ID
lk-quick-start-project
Project number
921312571473
ADD PEOPLE TO THIS PROJECT
Go to project settings
Resources
This project has no resources
CREATE RESOURCE
APIs
Requests (requests/sec)
1.0
0.8
0.6
No data is available for the selected time frame.
0.4
0.2
0
10 PM
10:15
10:30
10:45
Go to APIs overview

# 11.2.3.5.4. VPCネットワークの作成

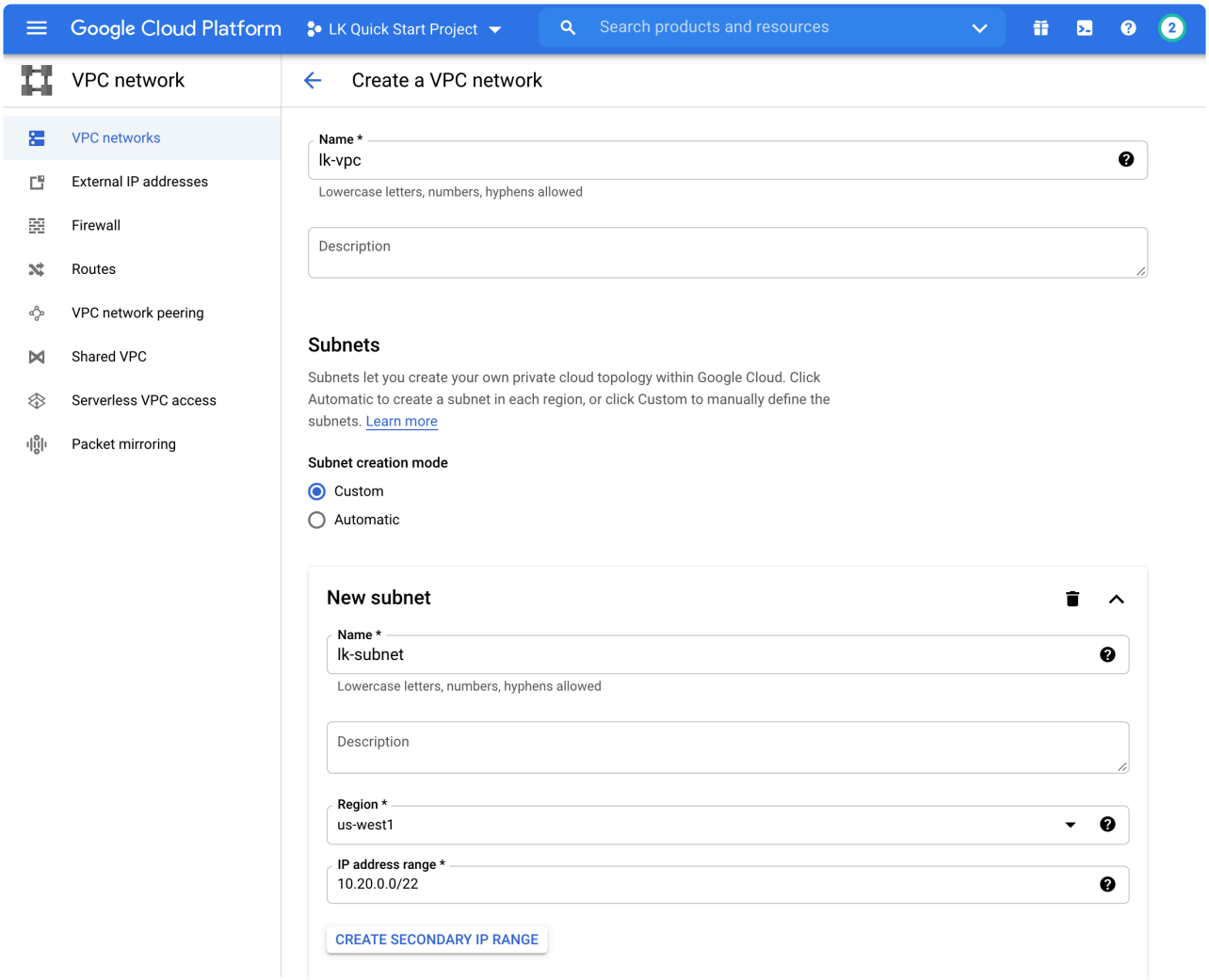
VPCネットワークは、ローカルネットワークを表すGoogle Cloudリソースです。Google Cloud内でさまざまなVPCネットワークを定義して、異なるシステムを論理的に分離できます。
詳細については、Google Cloudのドキュメント を参照してください。

このセクションでは、LifeKeeperをテストするためのVPCネットワーク（名前は `lk-vpc` とします）を次のように作成します。

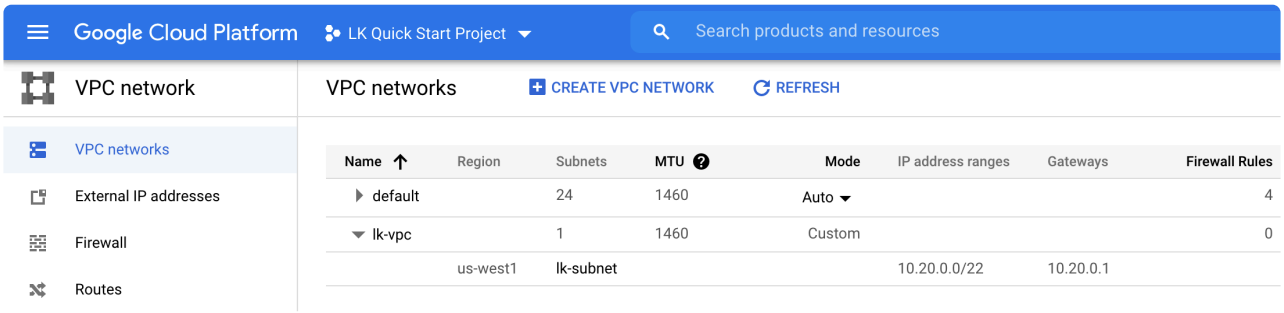
1. [VPCネットワーク] を選択すると、現在のVPCネットワークのリストが表示されます。定義されたリージョンごとに、「デフォルト」のVPCネットワークとそのサブネットがあります。上部にある [VPC ネットワークを作成] をクリックします。




#2.「VPCネットワークの作成」ウィザードで、VPCネットワークの名前を`lk-vpc`とし、VPCネットワークを作成するリージョン、サブネットの名前、CIDRブロック`10.20.0.0/22</font`を設定します。




3. 入力した値を確認したら、下部にある [作成] をクリックします。これで、新しいVPCネットワークが作成されます。

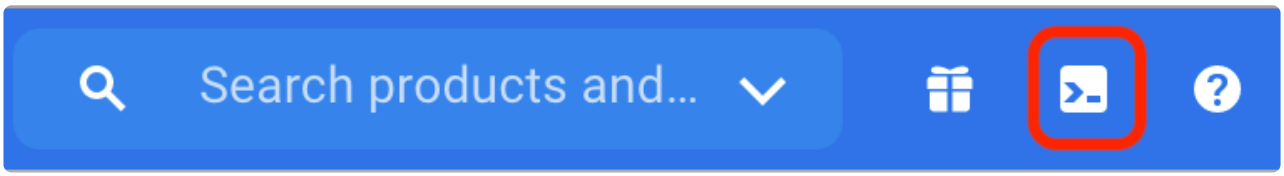


# 11.2.3.5.5. 新しいSSH認証鍵の作成

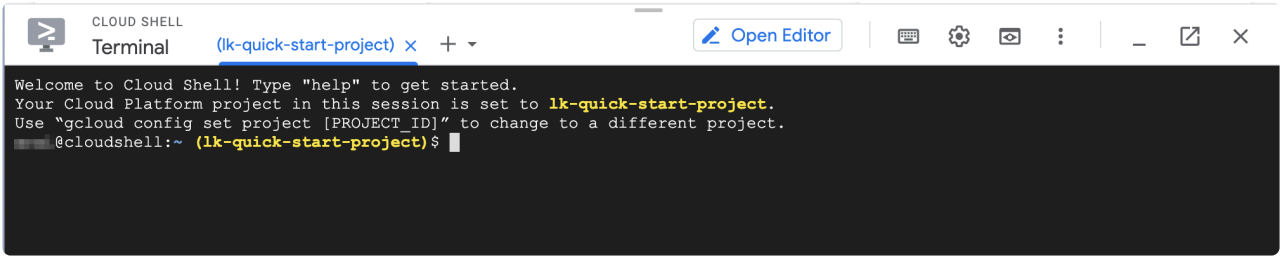
VMに接続するには、SSH認証鍵を使用する必要があります。このセクションでは、Google Cloudコンソールで新しいSSH認証鍵を作成し、秘密鍵をローカルシステムにダウンロードする方法を説明します。

詳細については、Google Cloudのドキュメント を参照してください。

1. コンソールの上部にある [>_] をクリックして、Cloud Shellを起動します。



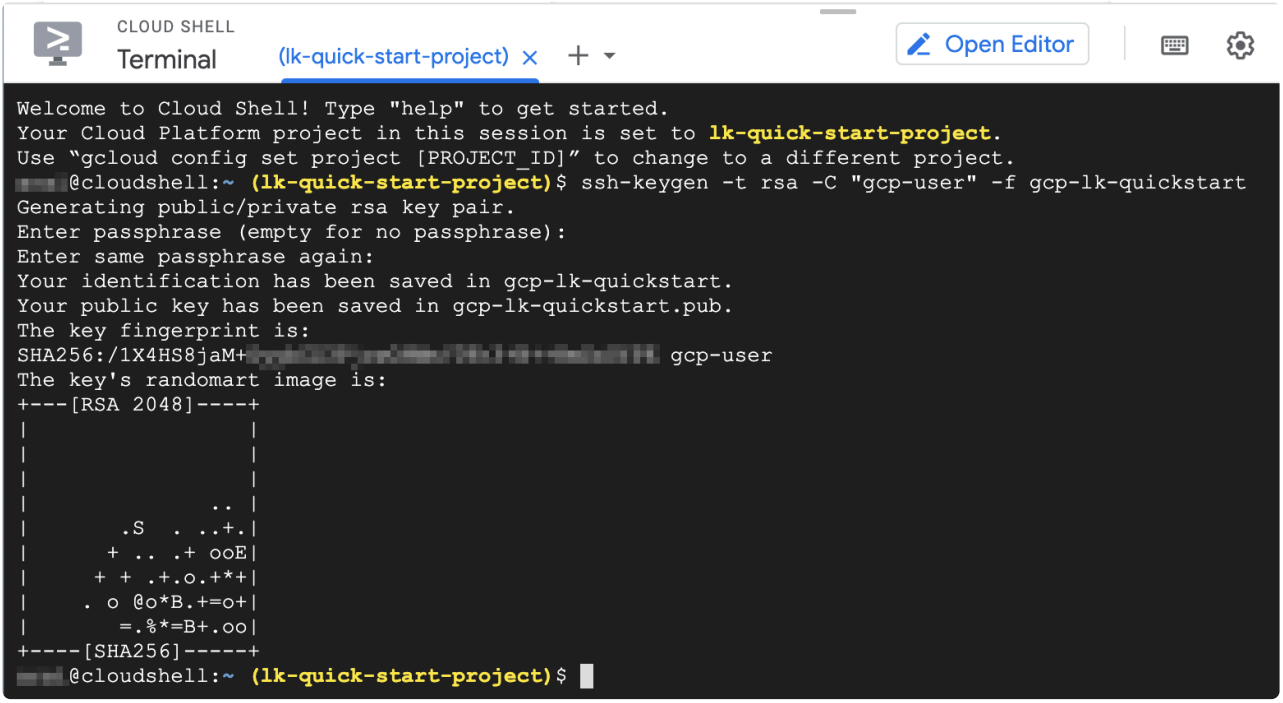

2. Cloud Shellを起動すると、ブラウザーの下部にシェル画面が表示されます。




3. ssh-keygenツールを使用して、新しいSSH認証鍵を作成します。

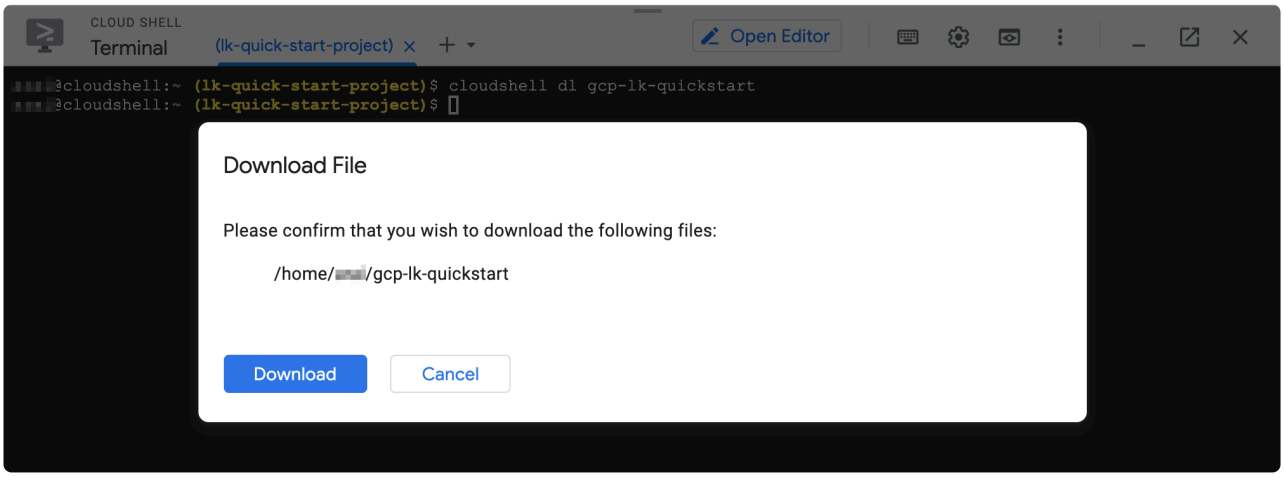
```
ssh-keygen -t rsa -C "gcp-user" -f gcp-lk-quickstart
```

4. `gcp-user`というユーザー名で新しい鍵`gcp-lk-quickstart`が生成されます。




5. 後のステップで、公開鍵 `gcp-lk-quickstart.pub`を新しいVMに登録する必要があります。 `cat gcp-lk-quickstart.pub`と入力して、公開鍵をコピーします。

6. `cloudshell dl gcp-lk-quickstart`で秘密鍵をダウンロードします。

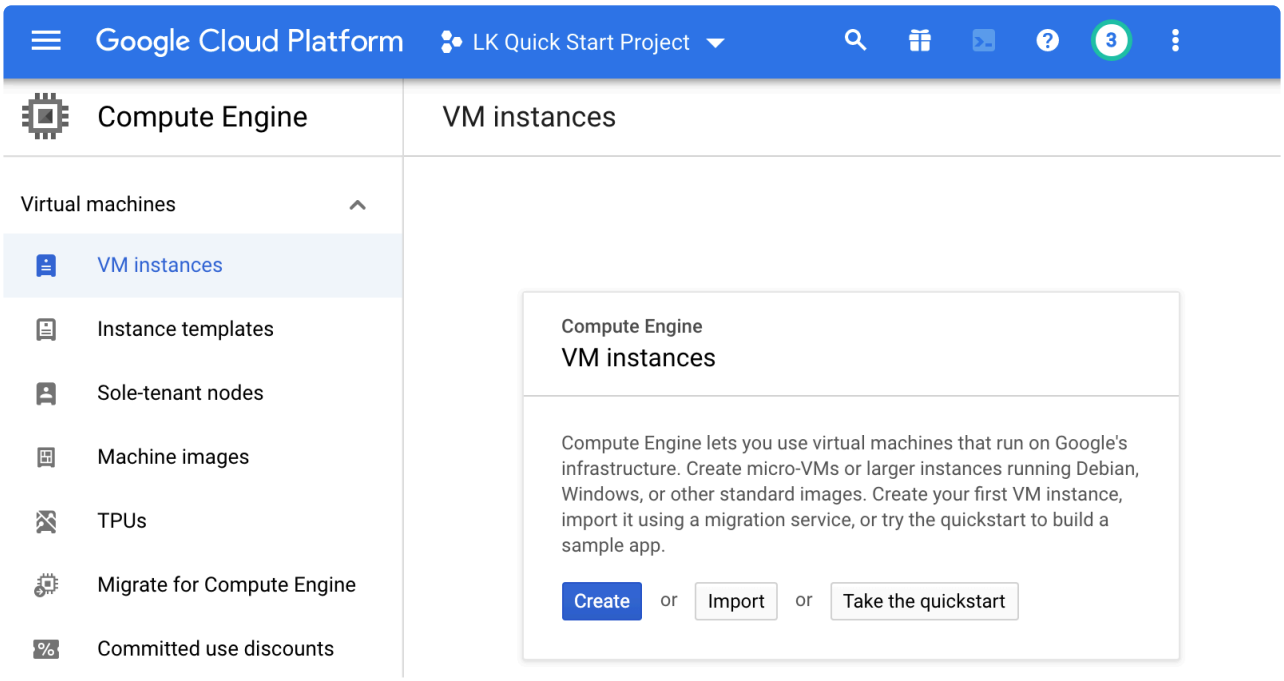



この秘密鍵は、後のステップでVMに接続する際に必要です。

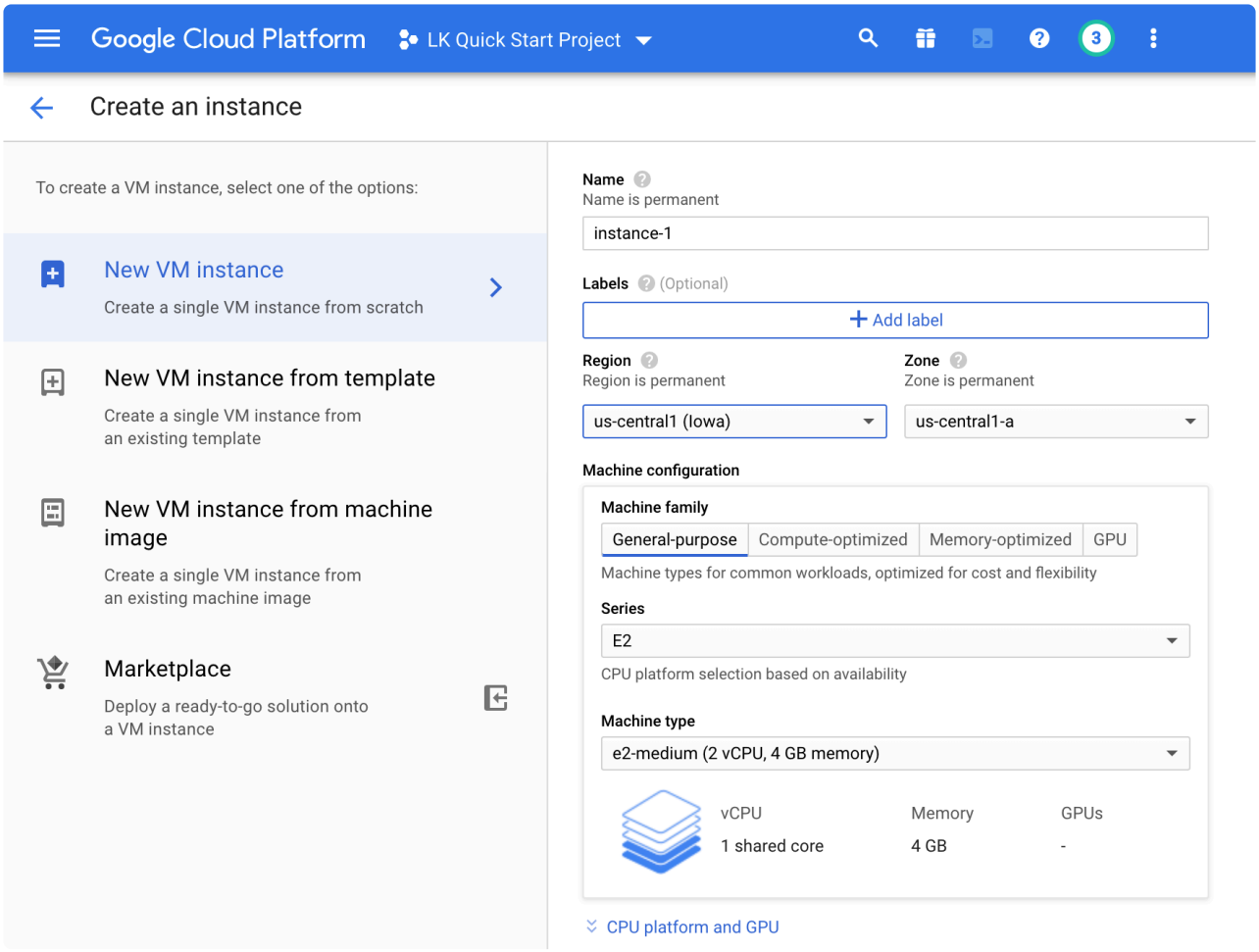
# 11.2.3.5.6. 1つ目のGoogle Cloud VMの作成

前のセクションでは、ネットワークの構成について説明しました。次に、1つ目のVMを作成します。このチュートリアル で使用するコンピューティングリソースで説明したように、2つのディスクが必要です。このセクションでは、2台目のディスクの作成方法についても説明します。

1. [Compute Engine] > [VMインスタンス] に移動し、[作成] をクリックして新しいVMを作成します。

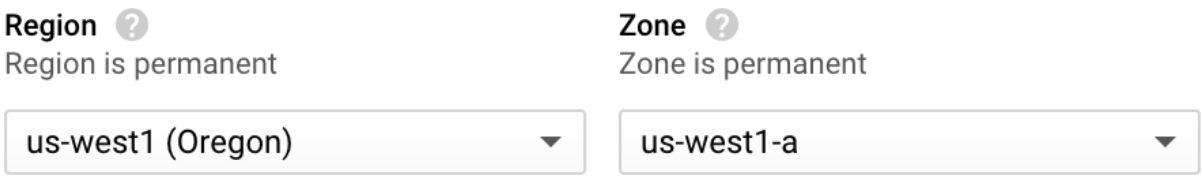



2. 「インスタンスの作成」ウィザードが表示されます。

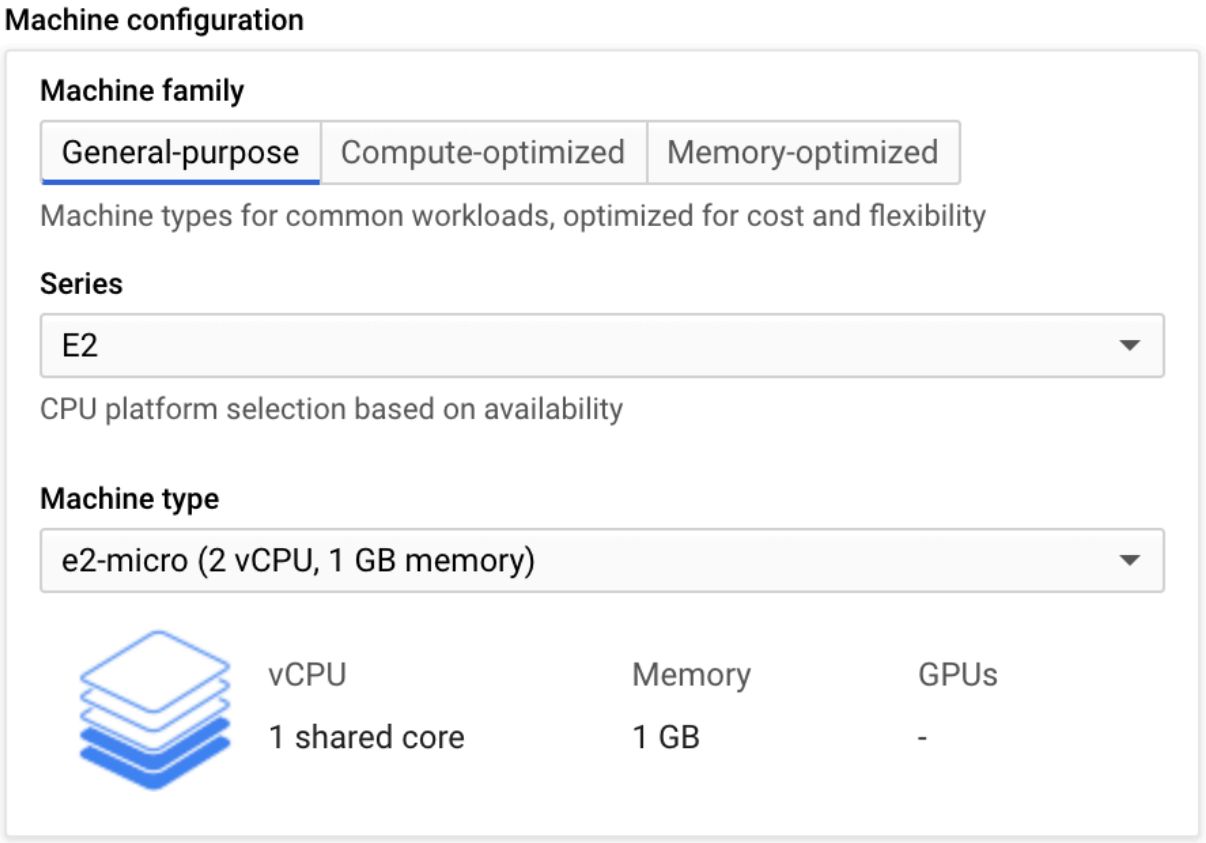



3. 新しいVMの名前、node-aと入力します（2番目または3番目のノードを作成する場合、これはnode-b、node-cになります）。

4. VMのリージョンとゾーンを選択します。近くのリージョンを入力し、ゾーンを選択します。node-aの場合は -a、node-bの場合は -b、node-cの場合は -cで終わるゾーンを選択します。地域によっては、ゾーンの名前が異なる場合がありますので、ご注意ください。

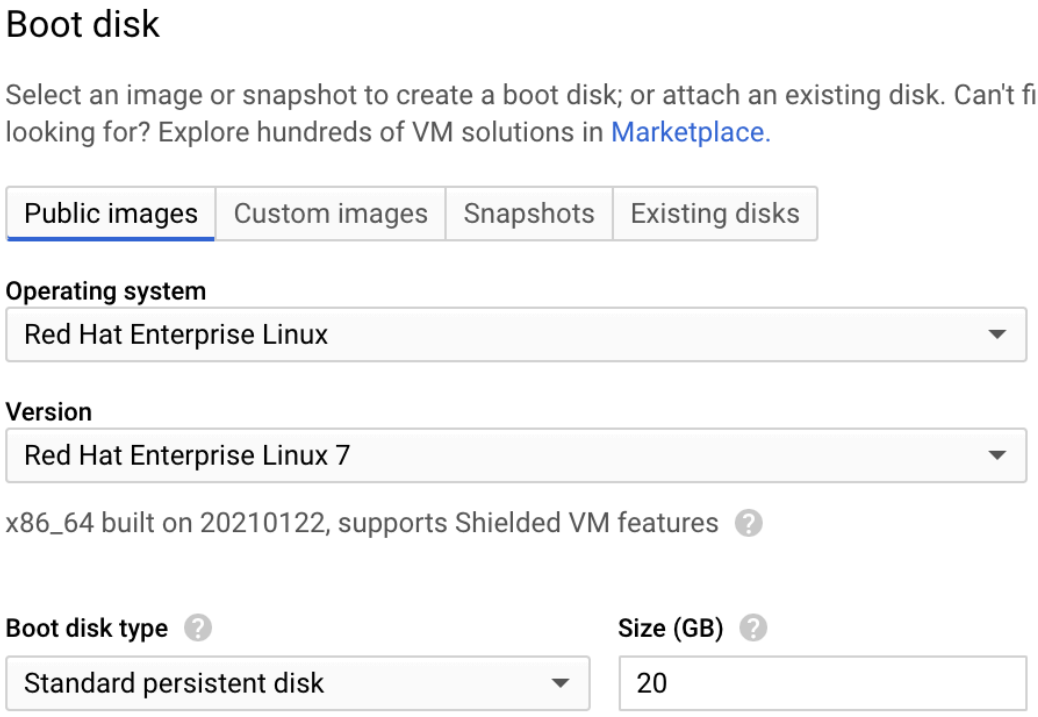



5. マシン構成を選択します。保護しているアプリケーションに応じて、必要なCPUとメモリの要件を選択します。

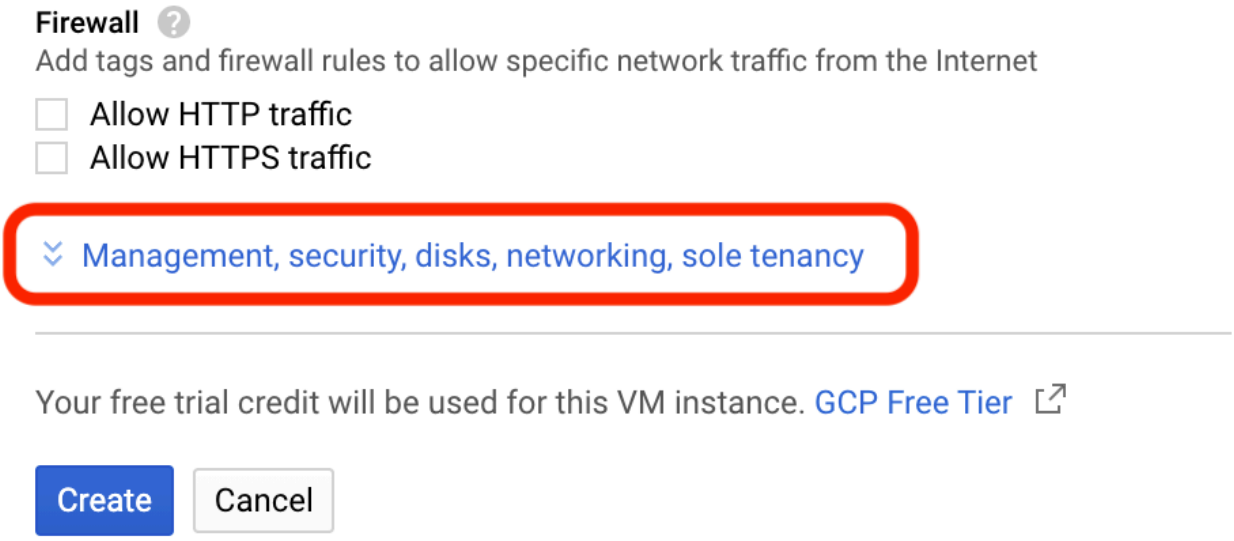



6. オペレーティングシステムを選択します。デフォルトでは、Debian GNU /Linux10が選択されています。LifeKeeperと、使用する予定のデータベースやアプリケーションの両方でサポートされているオペレーティングシステムを選択してください。[変更] をクリックして、Red Hat Enterprise Linux 7を選択します。

> ! SAPまたはSAP HANAクラスターの作成を計画している場合、このステップで「Red Hat Enterprise Linux 8.2 for SAP Applications」ブートイメージを選択します。

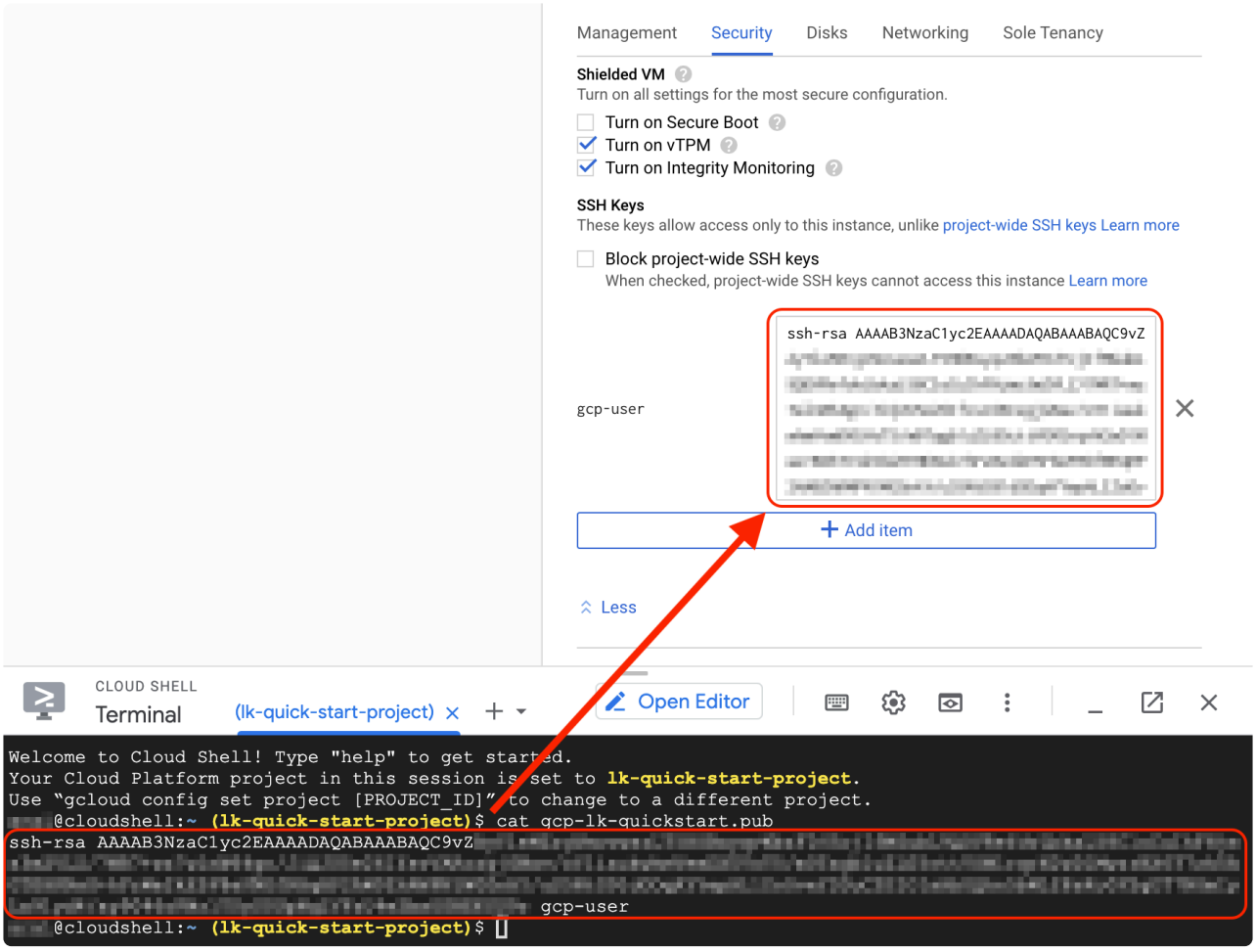



7. [作成] を選択する前に、[管理、セキュリティ、ディスク、ネットワーク、単一テナンシー] のドロップダウンリストを展開します。

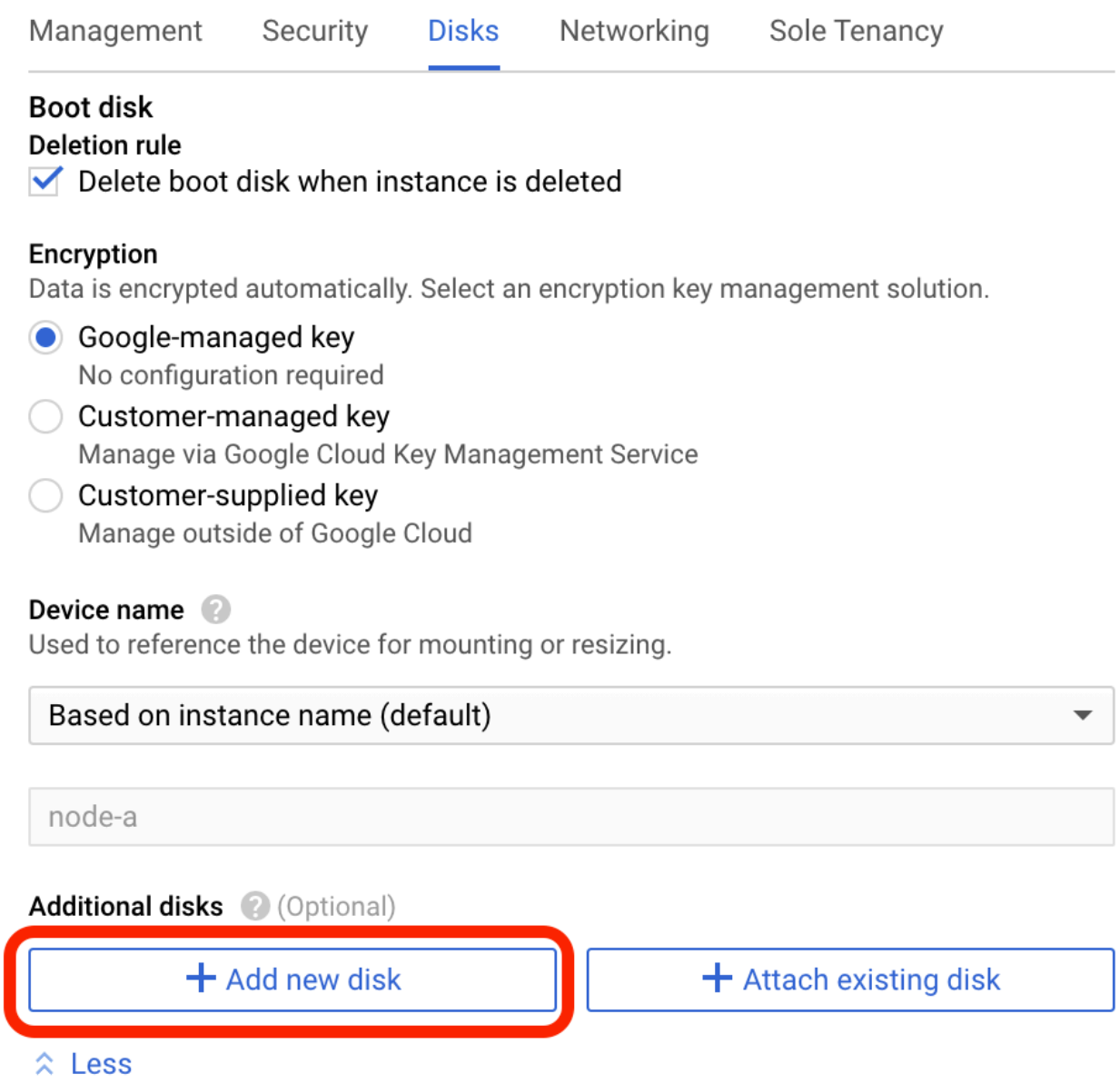



8. [セキュリティ]タブに移動し、新しいSSH認証鍵の作成 セクションでダウンロードした公開鍵を入力します。

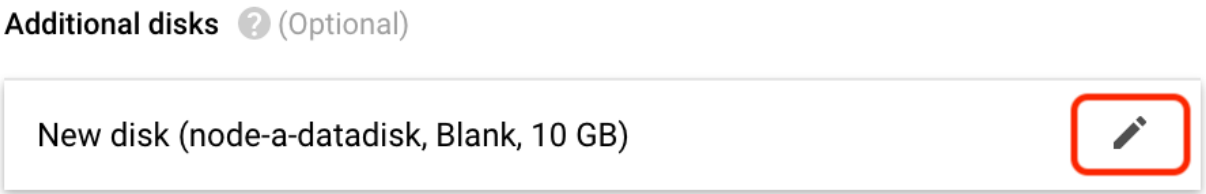
公開鍵をコピーして貼り付ける場合は、空白がないことを確認してください。



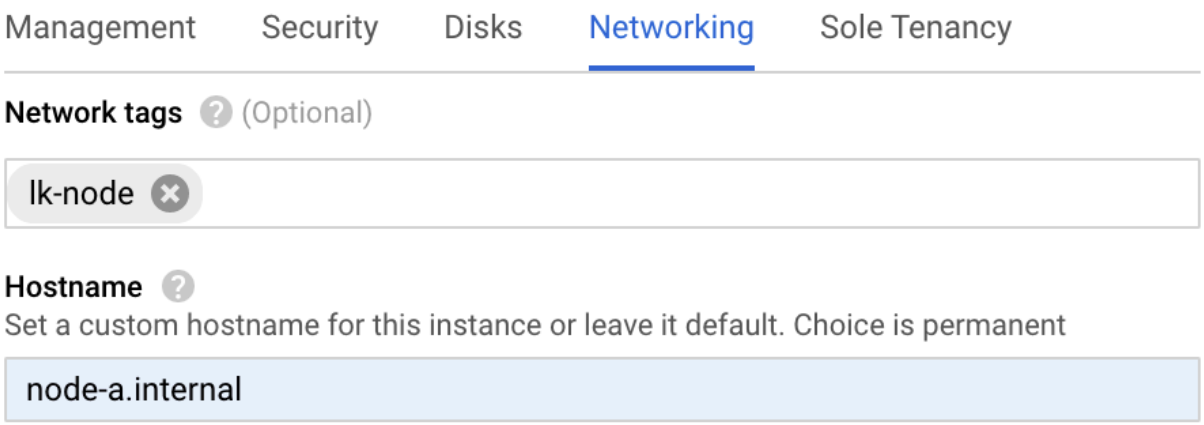

9. 「ディスク」タブに移動します。node-aとnode-bでのデータ複製用に追加のディスクが必要です。ノードを追加するには、[+新しいディスクを追加] をクリックします。



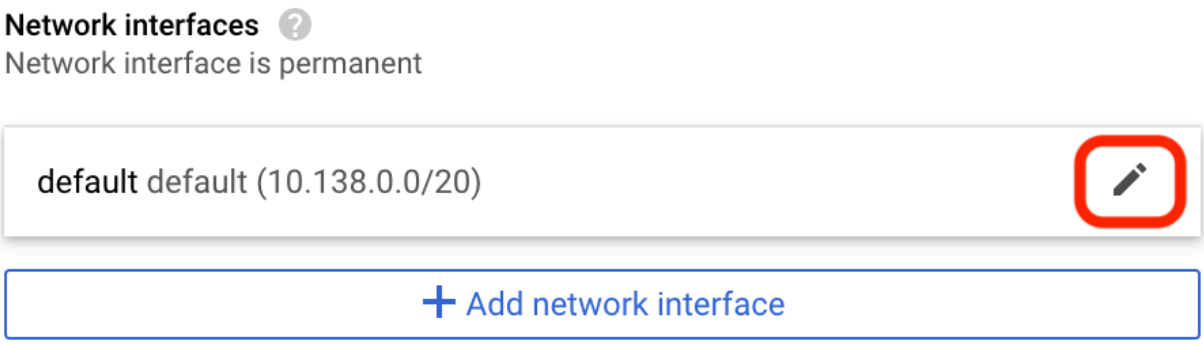

10. ディスクの「名前」と「サイズ」を指定します




11. 「ネットワーキング」タブに移動します。「ネットワークタグ」と「ホスト名」を指定します。「ネットワークタグ」は、ファイアーウォールの設定でVMを識別するために使用されます（次の手順で説明します）。

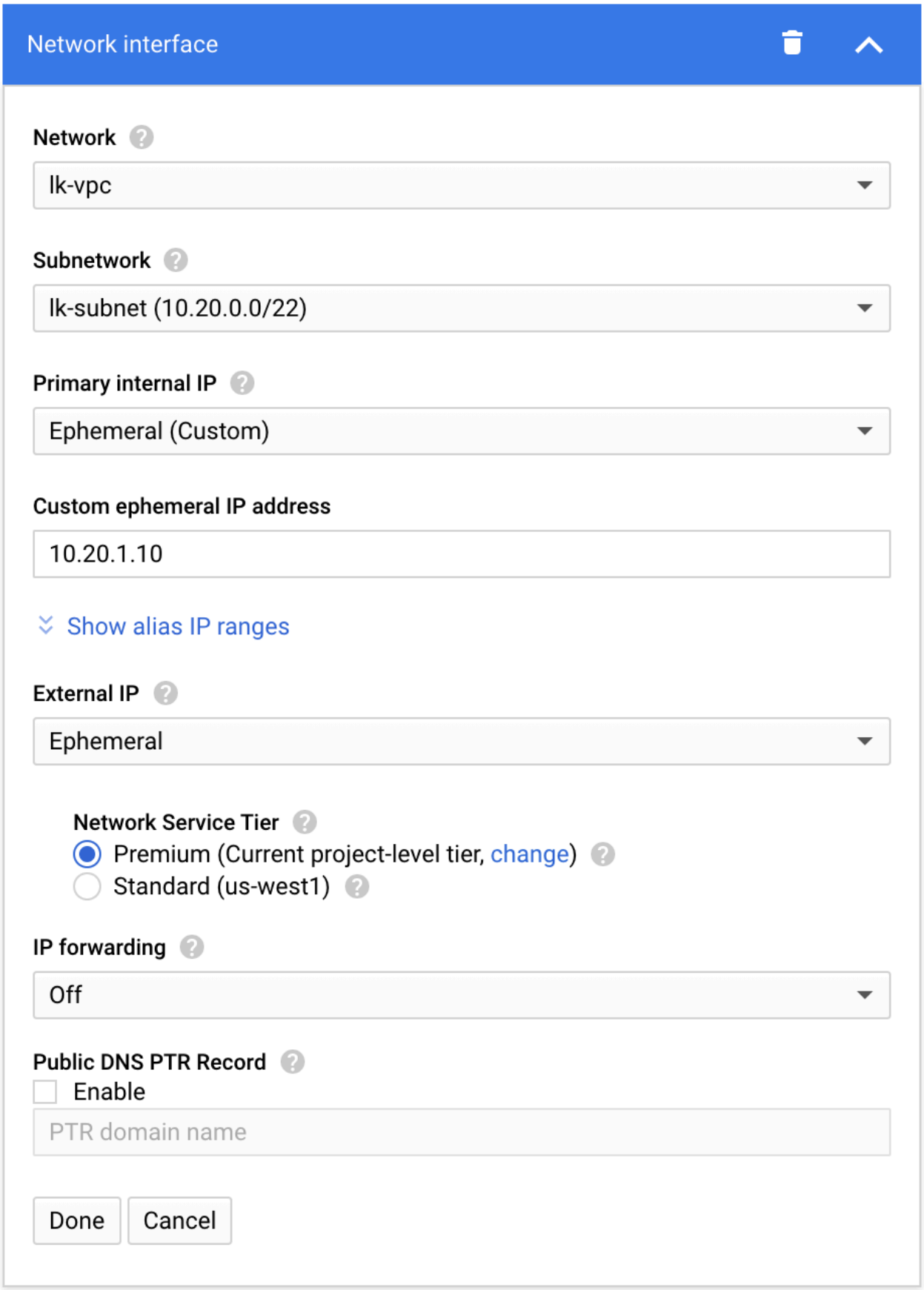



12. ネットワークインターフェースを編集します。「編集」アイコンを選択します。




13. ネットワークインターフェースのパラメータを次のように入力します。

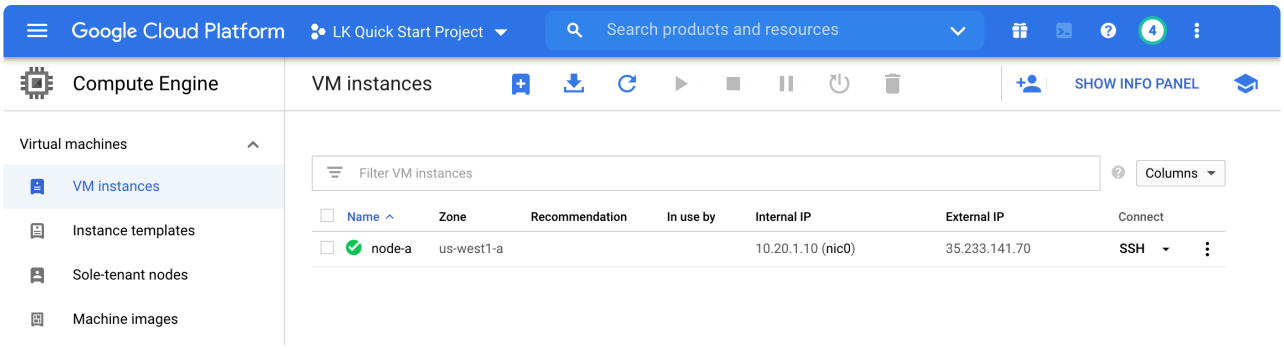
- Network: `lk-vpc`
- Subnetwork: `lk-subnet (10.20.0.0/22)`
- Primary internal IP: Ephemeral (Custom)
- Custom ephemeral IP address: `10.20.1.10` for node-a, `10.20.2.10` for node-b, `10.20.3.10` for node-c.
- External IP: Ephemeral
- IP forwarding: Off




Click “Done” once you enter the values.

Confirm the values you entered, then click “Create” at the bottom.

14. 値を入力したら、[完了] をクリックします。
これで、仮想マシンの構成が完了しました。



# 11.2.3.5.7. ファイアーウォールルールの設定

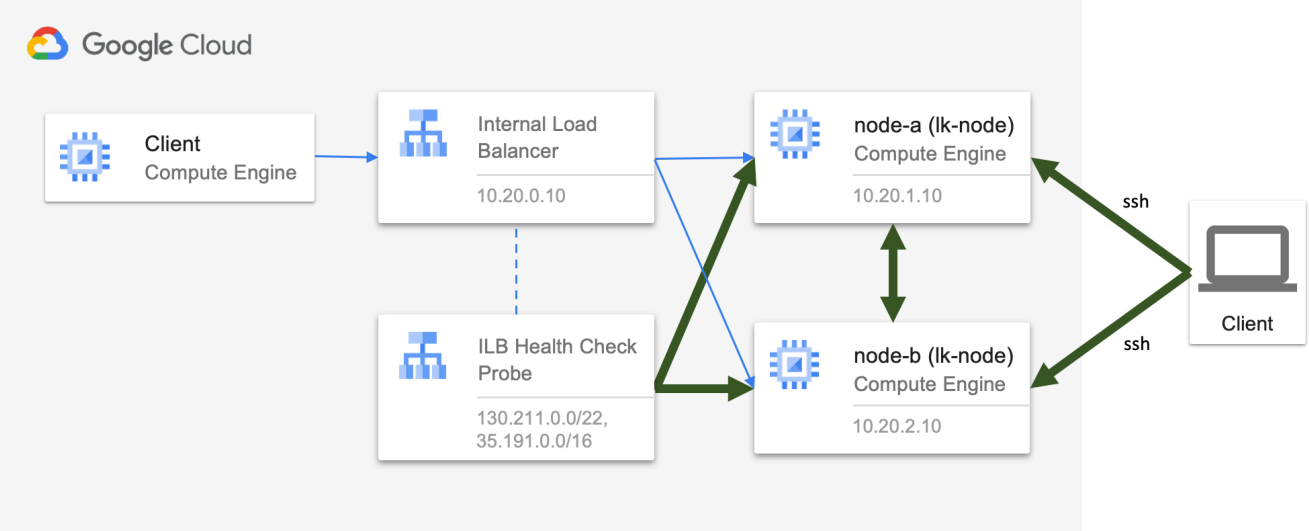
ファイアーウォールルールのセットでは、トラフィックの種類ごとに「許可」と「拒否」の両方のルールを定義することができます。ソースは、インターネットアドレス、内部ロードバランサー、またはVMのグループです。VMのグループは、VMに割り当てられたタグによって識別されます。

詳細については、Google Cloudのドキュメント を参照してください。

[VPCネットワーク] > [ファイアウォール] に移動すると、「デフォルト」のVPCに対して一連のファイアーウォールルールがすでに定義されています。 `lk-vpc`に対して次のルールを定義します。

- `lk-node`タグを使用してリモートワークロケーションからVMへのsshトラフィックを許可します。(`fw-allow-ssh`)
- `lk-node`タグを使用してVM間のすべてのトラフィックを許可します。(`fw-allow-lk-node-connection`)
- `lk-node`タグを使用して、内部ロードバランサーからVMへのすべてのヘルスチェックプローブを許可します。 (`fw-allow-health-check`)

これらのファイアーウォールルールで許可されるトラフィックは、下図の太い矢印で強調表示されています。



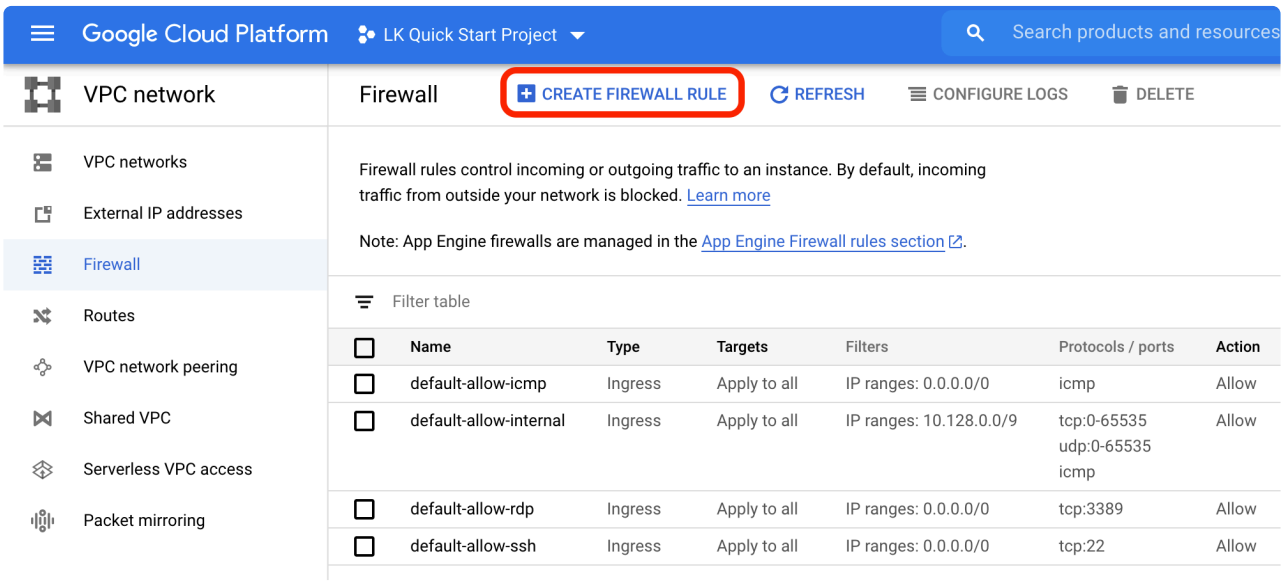

次の表に、これらのルールの構成方法の概要を示します。

| 名称 | パラメーター | 値 |
|---|---|---|
| VM全体で共通の値 | **Network** | **lk-vpc** |
| | **Type** | **Ingress** |
| | **Target** | **lk-node** |

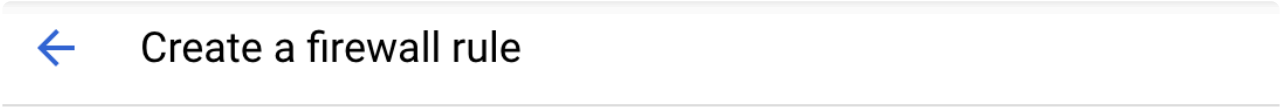
| | Action | Allow |
|---|---|---|
| fw-allow-ssh | Priority | 1000 |
| | Source | WAN IP Address of work location |
| | Protocols / ports | tcp:22 |
| fw-allow-lk-node-connection | Priority | 1100 |
| | Source | Tags: lk-node |
| | Protocols / ports | all |
| fw-allow-health-check (*1) | Priority | 1200 |
| | Source | 130.211.0.0/22, 35.191.0.0/16 |
| | Protocols / ports | all |

(*1) ロードバランサーのヘルスチェックスプローブに関するGoogle Cloudのドキュメント

1. ホーム画面から [VPCネットワーク] > [ファイアウォール] を選択すると、デフォルトのvpcのファイアーウォールルールのリストが表示されます。上部にある [ファイアウォールルールの作成] をクリックして、最初のルール「fw-allow-ssh」を作成します。



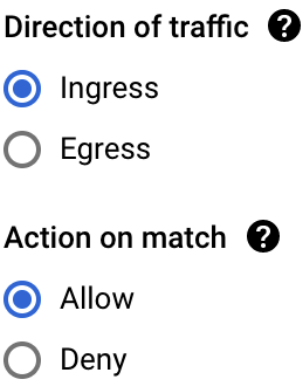

2. 以下のステップでは、作業場所からのssh接続を許可して、作業場所からノードを構成できるようにする方法について説明します。「ファイアウォールルールの作成」ウィザードで、名前を`fw-allow-ssh`と指定します。




3. [ネットワーク] でlk-vpcを選択します。

4. 優先度は0～65535の任意の値を指定できますが、デフォルト値の1000のままにします。他のルールには、1100、1200などを使用できます。これにより、将来追加のルールが必要になった場合に、他の優先度の値を挿入できます。

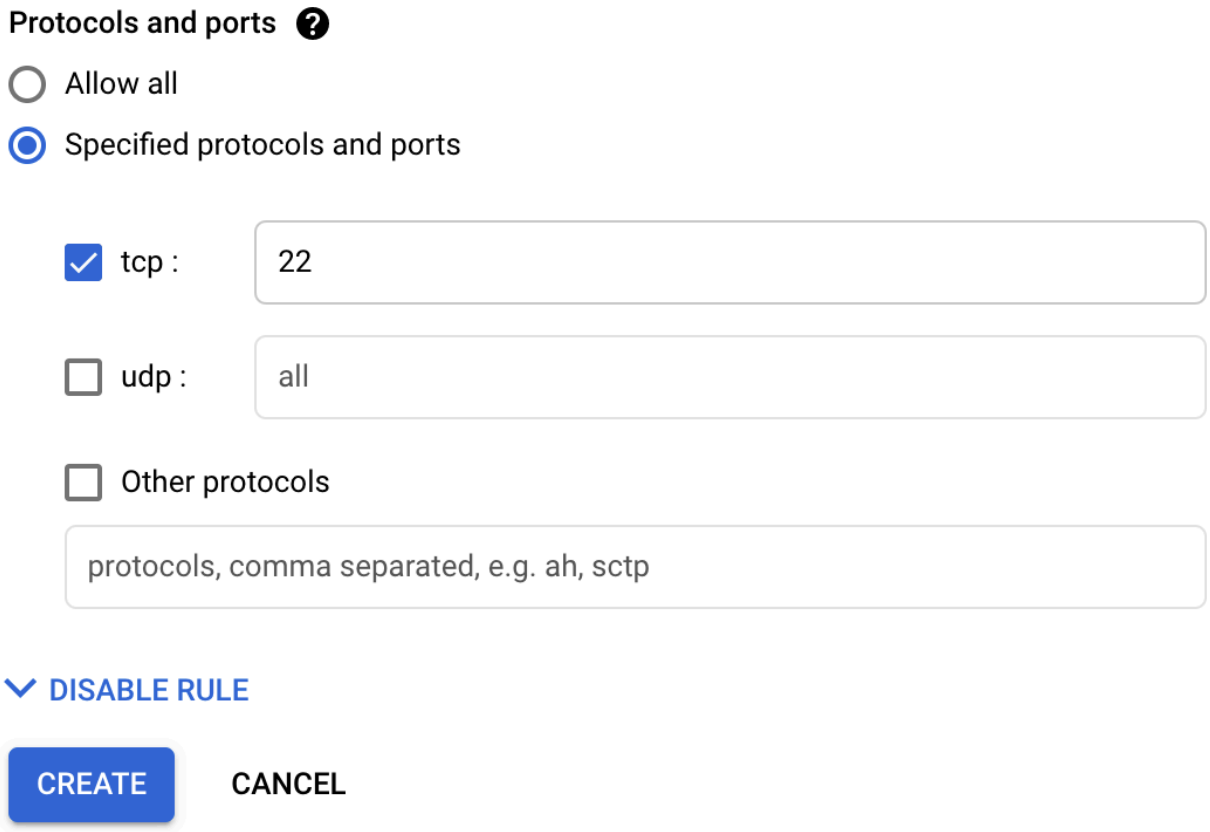
5. [トラフィックの方向] で「Ingress」を選択し、[一致したときのアクション] で「許可」を選択します。この選択により、次の手順で指定したパターンに一致する受信トラフィックが「許可」されます。



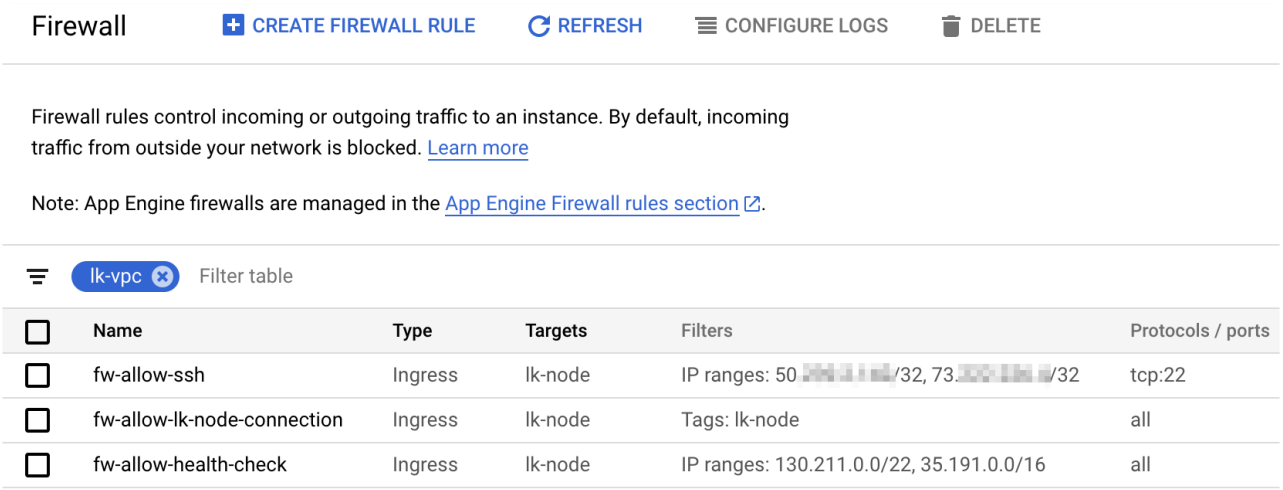

6. ターゲット（トラフィックの宛先）を選択します。必ず、作成したVMに割り当てられているlk-nodeタグを選択してください。

7. ソース（作業場所のWAN IPアドレス）を選択します。

8. 最後に、許可するプロトコルを選択します。入力したすべての値を確認し、ルールを作成します。




これでルールが定義されました。同じ手順で、他のルールも作成します。

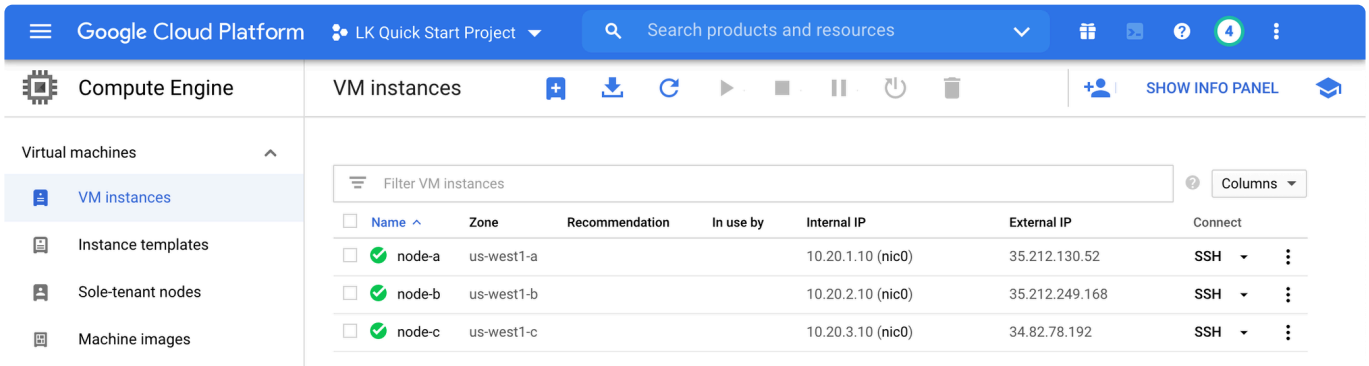

Firewall
CREATE FIREWALL RULE
REFRESH
CONFIGURE LOGS
DELETE
Firewall rules control incoming or outgoing traffic to an instance. By default, incoming traffic from outside your network is blocked. Learn more
Note: App Engine firewalls are managed in the App Engine Firewall rules section.
lk-vpc
Filter table
Name
Type
Targets
Filters
Protocols / ports
fw-allow-ssh
Ingress
lk-node
IP ranges: 50 /32, 73 /32
tcp:22
fw-allow-lk-node-connection
Ingress
lk-node
Tags: lk-node
all
fw-allow-health-check
Ingress
lk-node
IP ranges: 130.211.0.0/22, 35.191.0.0/16
all

# 11.2.3.5.8. 2番目と3番目の仮想マシンの作成

2番目のノード（node-b）と3番目のノード（node-c）を作成するために必要な手順は、最初のノード（node-a）について前述した手順とほぼ同じです。次の表は、3つのノードの違いと、これらのインスタンスを作成するために必要な詳細を示しています。

| 名称 | パラメーター | 値 |
|---|---|---|
| VM全体で共通の値 | **VPC network** | **lk-vpc** |
| | **Subnet** | **lk-subnet** |
| | **Network Tags** | **lk-node** |
| | **External IP** | **Ephemeral** |
| node-a | Zone | us-west1-a (*1) |
| | Hostname | node-a.internal |
| | Internal IP | 10.20.1.10 |
| | Second Disk (Storage Device) | 2個めのディスクが必要です |
| node-b | Zone | us-west1-b (*1) |
| | Hostname | node-b.internal |
| | Internal IP | 10.20.2.10 |
| | Second Disk (Storage Device) | 2個めのディスクが必要です |
| node-c | Zone | us-west1-c (*1) |
| | Hostname | node-c.internal |
| | Internal IP | 10.20.3.10 |
| | Second Disk (Storage Device) | 2個目のディスクは必要ありません |

作成されると、VMのリストは以下のようになります。



# 11.2.4. LifeKeeper for Linuxを実行するためのLinuxノードの設定

このセクションでは、LifeKeeper for Linuxをインストールする前に、Linuxノードを設定する手順の概要を説明します。
Linuxノードに接続するには、Windowsにsshクライアントソフトウェアをインストールする必要がある場合があります。
sshクライアントの利用については sshを使用してWindowsクライアントからLinuxノードに接続する をご参照ください。

## ノードの設定

各ノードについて、以下の手順を実行する必要があります。

- 各インスタンスにホスト名を設定する
- SELinuxを無効にする
- ファイアーウォールを無効にする
- rootユーザーのパスワードを設定する
- x11をインストールする

# 11.2.4.1. sshを使用してWindowsクライアントからLinuxノードに接続する

Linuxノードでは、sshによるログインを受け付けることができるため、sshクライアントが必要です。このページでは、PuTTYを使用して、（パスワードの代わりに）公開鍵/秘密鍵のペアを使用してLinuxノードに接続する方法を説明します。また、Linuxノードに基本構成を作成し、SIOS Protection Suite for Linuxをインストールする方法についても説明します。SIOS Protection Suiteをインストールした後、GUIベースのアプリケーションを使用するためにx11を有効にするためのツールをさらに追加します。

## ツールのダウンロードとインストール

最新のPuTTYソフトウェアをダウンロードしてインストールします。

## 秘密鍵（.pem）の変換

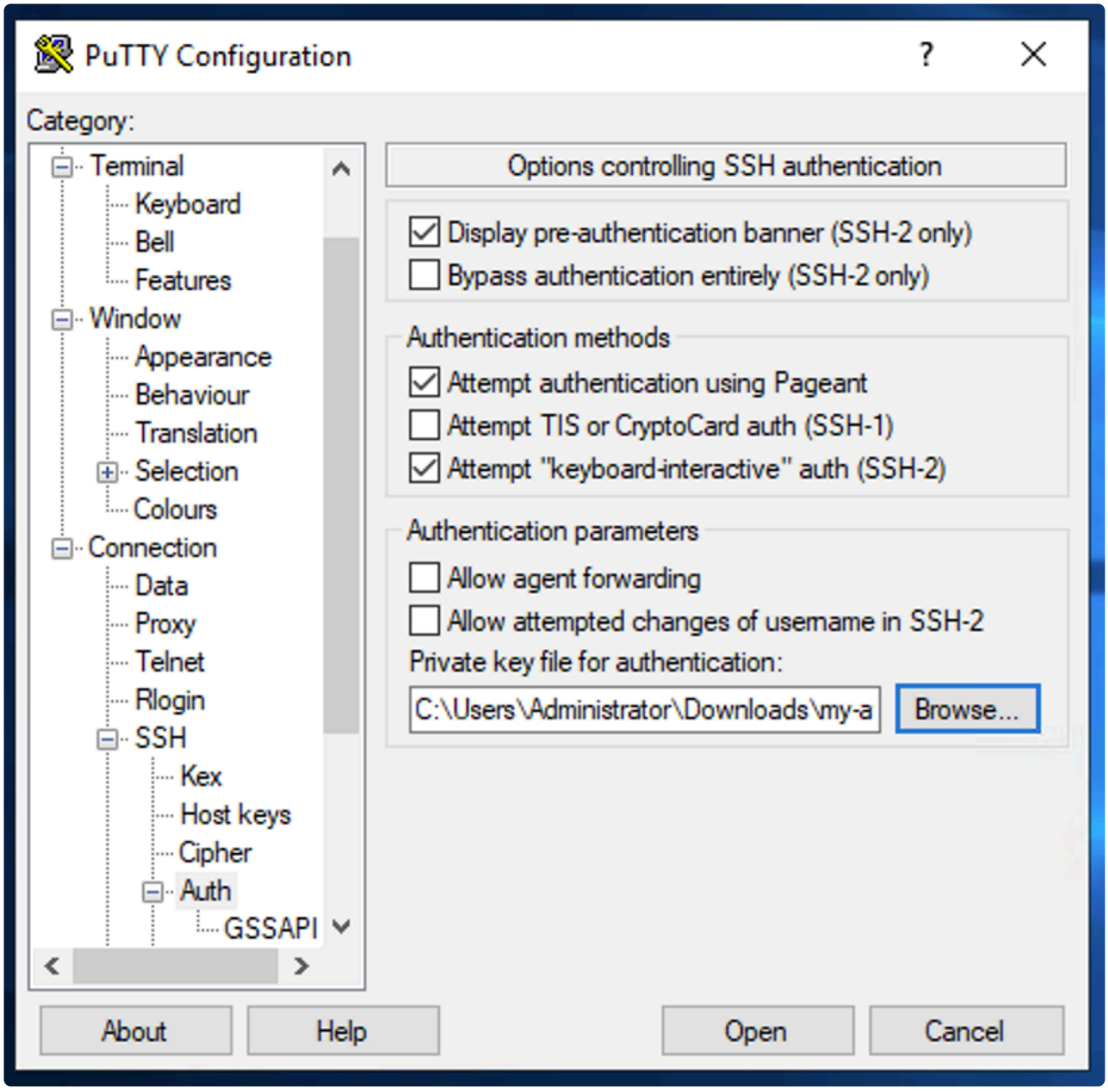
.pem 形式の秘密鍵ファイルがある場合は、それをPuTTY秘密鍵ファイル（`*.ppk`）ファイルに変換します。

ファイルを変換するには、次の手順を実行します。

1. 「PuTTY Key Generator」を起動します。
2. [Load] を選択し、.pemファイルを指定します。
3. .ppk を選択し、[Save private key] をクリックします。

## PuTTYの設定

1. PuTTYを開き、Connection > SSH > Auth を選択します。
2. [Private key file for authentication] で秘密鍵ファイルを選択します。

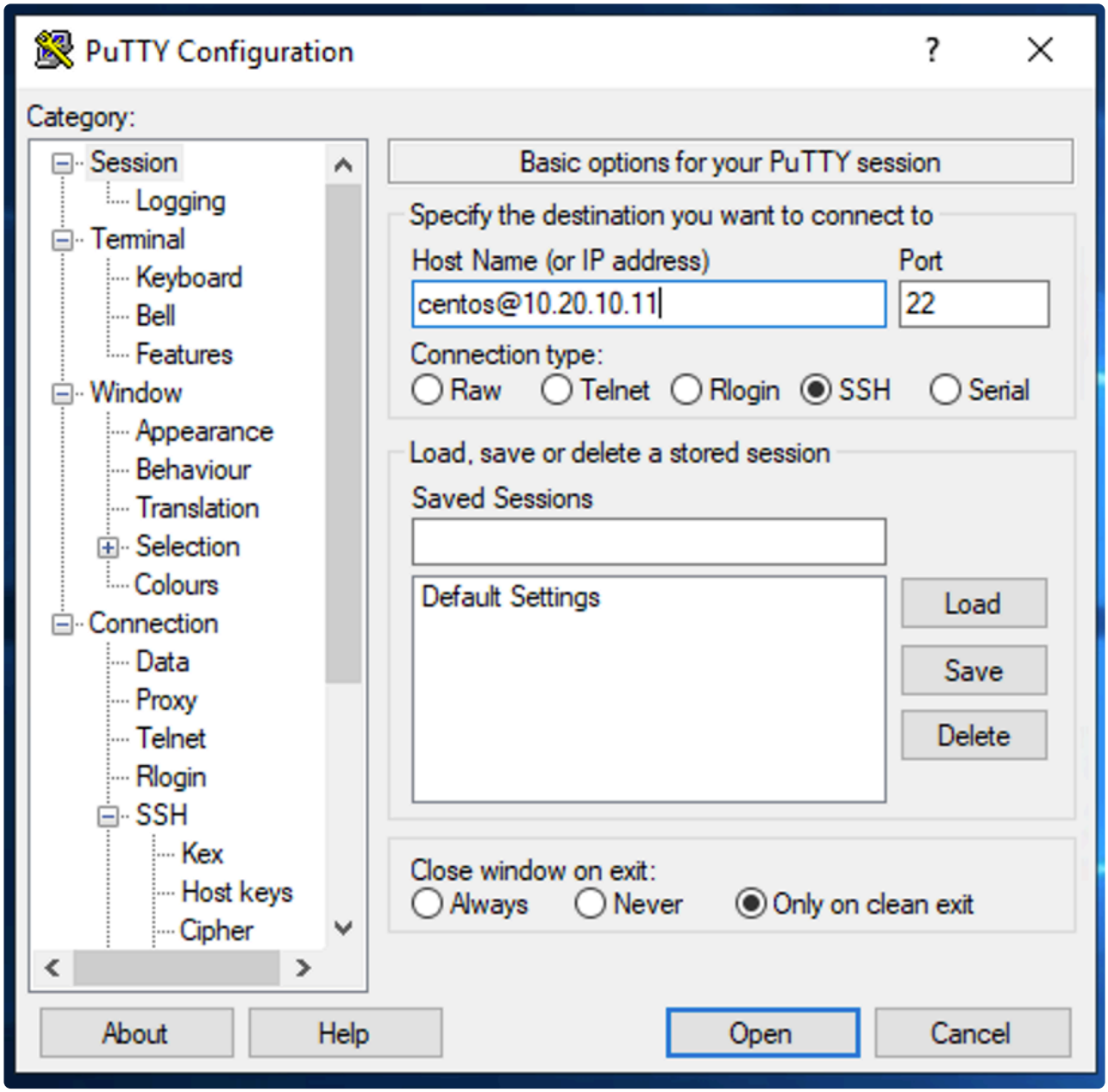



.

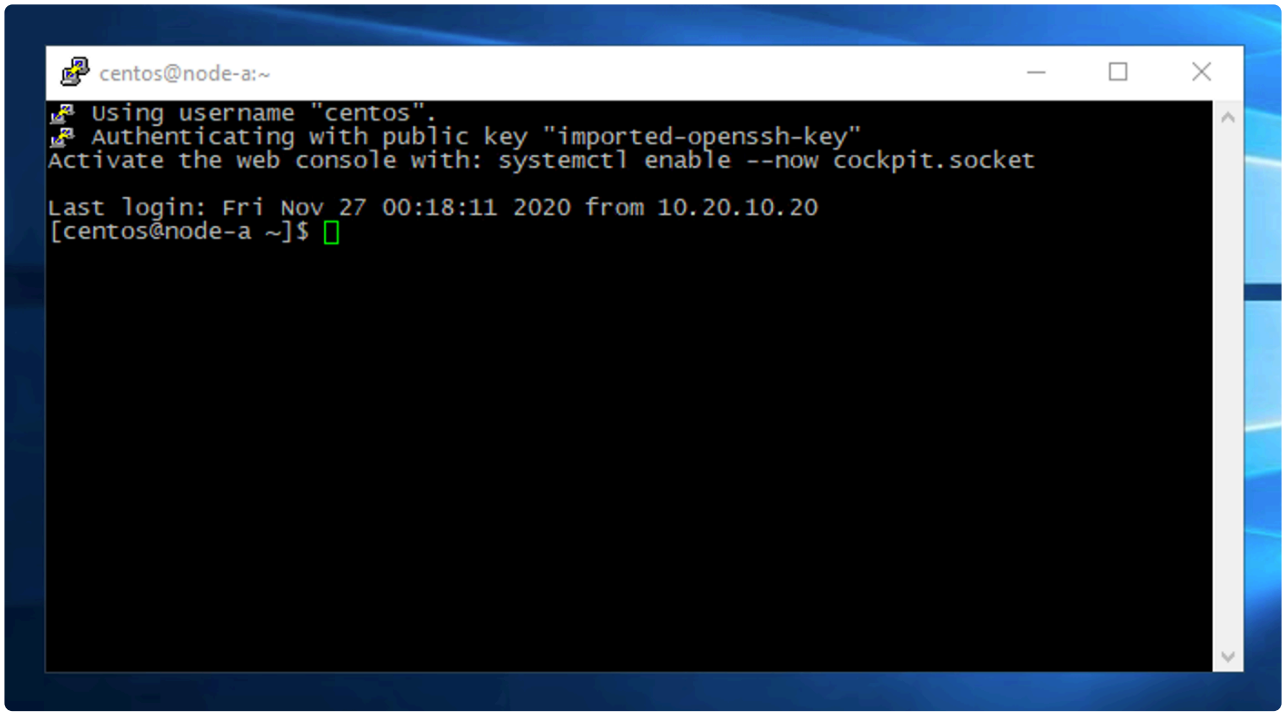
3. 毎回再設定せずに済むように、構成を「Session」ページとして保存することをお勧めします。

## Linuxインスタンスに接続する

1. PuTTYからセッションを開始します。




2. サーバーが以前に接続されていないことを示すセキュリティアラートを確認します。[Yes] を選択して続行します。
   接続されると以下のような表示になります。


centos@node-a:~
Using username "centos".
Authenticating with public key "imported-openssh-key"
Activate the web console with: systemctl enable --now cockpit.socket
Last login: Fri Nov 27 00:18:11 2020 from 10.20.10.20
[centos@node-a ~]$

# 11.2.4.2. 各インスタンスにホスト名を設定する

各ノードに対して、以下のコマンドを実行します。

インスタンスを作成したら各ノードにホスト名を設定し/etc/hostsファイルにエントリを追加して、ノードが相互を特定できるようにします。また、お使いの環境でDNSが使用できる場合は、DNSを使用することもできます。

1. hostnamectlコマンドを使用して、ホスト名を設定します。

```
$ sudo su -
# hostname
ip-10-20-1-10.us-west-2.compute.internal
# hostnamectl set-hostname node-a
```

2. DNSサーバーが利用可能な場合は、サーバーを使用してホスト名を登録します。使用可能なDNSサーバーがない場合は、/etc/hosts を使用してホスト名を解決することをお勧めします。

```
# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
```

3. 次の行を/etc/hostsファイルに追加して、 node-aとnode-b をIPアドレスに解決されるようにします。

```
10.20.1.10 node-a
10.20.2.10 node-b
10.20.3.10 node-c
```

4. /etc/hostsのバックアップを作成し、ファイルを編集して変更を確認します。

```
# cp -p /etc/hosts /etc/hosts.org
# vi /etc/hosts
# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.20.1.10 node-a
10.20.2.10 node-b
10.20.3.10 node-c
```

5. これらのノードに名前でpingできることを確認します。

```
# ping node-b
PING node-b (10.20.2.10) 56(84) bytes of data.
64 bytes from node-b (10.20.2.10): icmp_seq=1 ttl=64 time=0.868 ms
64 bytes from node-b (10.20.2.10): icmp_seq=2 ttl=64 time=0.892 ms
^C
```

# 11.2.4.3. SELinux を無効にする

Security-Enhanced Linux (SELinux) は、強制アクセス制御 (MAC) を含むアクセス制御セキュリティポリシーをサポートするためのメカニズムを提供する Linux カーネルセキュリティモジュールです。

https://selinuxproject.org/page/Main_Page

> SELinux が enforcing モードの場合 LifeKeeper は動作しません。また、SAP 環境で必要な場合を除いて permissive モードは推奨されません。詳細は インストール前の要件 を参照してください。

## 無効化手順

1. ブートローダーを設定して、カーネルコマンドラインに `selinux=0` を追加します。

```
# grubby --update-kernel ALL --args selinux=0
```

2. システムを再起動します。

```
# reboot
```

3. 再起動後、 `getenforce` コマンドが “Disabled” を返すことを確認します。

```
# getenforce
Disabled
```

# 11.2.4.4. ファイアーウォールを無効にする

各ノードに対して以下のコマンドを実行する必要があります。

このセクションでは、LifeKeeper for Linuxをインストールするために、システムにインストールされている適切なファイアーウォールサービス (firewalld、SuSEfirewall2、iptables、nftables など) を無効にします。ファイアーウォールは有効なままにしておくことができますが、LifeKeeperと保護するアプリケーションに必要なポートを設定する必要があります（ここでは示していません）。

## firewalldを無効にする手順

次のコマンドを実行して、firewalldを停止して無効にします。

1. 次のコマンドを実行して現在のステータスを確認します。

```
systemctl status firewalld
```

“Unit firewalld.service could not be found.” と出力された場合(firewalldがインストールされていないことを示します)、ステップ2と3はスキップできます。

2. 次のコマンドを実行して、firewalldを停止して無効にします。

```
systemctl stop firewalld
systemctl disable firewalld
systemctl mask firewalld
```

3. 次のコマンドを実行して現在のステータスを確認します。

```
systemctl status firewalld
```

出力には“Loaded: masked”と“Active: inactive”が含まれている必要があります。

## SuSEfirewall2を無効にする手順

次のコマンドを実行して、SuSEfirewall2を停止して無効にします。

1. 次のコマンドを実行して現在のステータスを確認します。

```
systemctl status SuSEfirewall2
```

> ! “Unit SuSEfirewall2.service could not be found” と出力された場合(SuSEfirewall2がインストールされていないことを示します)、ステップ2と3はスキップできます。

2. 次のコマンドを実行して、SuSEfirewall2 を停止して無効にします。

```
systemctl stop SuSEfirewall2
systemctl disable SuSEfirewall2
systemctl mask SuSEfirewall2
```

3. 次のコマンドを実行して現在のステータスを確認します。

```
systemctl status SuSEfirewall2
```

出力には“Loaded: masked”と“Active: inactive”が含まれている必要があります。

## iptablesを無効にする手順

次の手順を実行してiptablesを無効にします。

1. 次のコマンドを実行して現在のステータスを確認します。

```
systemctl status iptables
```

> ! “Unit iptables.service could not be found”と出力され場合(iptablesがインストールされていないことを示します)、ステップ2と3はスキップできます。

2. 次のコマンドを実行して、iptables を停止して無効にします。

```
systemctl stop iptables
systemctl disable iptables
systemctl mask iptables
```

3. 次のコマンドを実行して現在のステータスを確認します。

```
systemctl status iptables
```

出力には“Loaded: masked” と “Active: inactive”が含まれている必要があります。

## nftablesを無効にする手順

次の手順を使用してnftablesを無効にします。

1. 次のコマンドを実行して現在のステータスを確認します。

```
systemctl status nftables
```

> ! “Unit nftables.service could not be found”と出力され場合(nftablesがインストールされていないことを示します)、ステップ2と3はスキップできます。

2. 次のコマンドを実行して、nftables を停止して無効にします。

```
systemctl stop nftables
systemctl disable nftables
systemctl mask nftables
```

3. 次のコマンドを実行して現在のステータスを確認します。

```
systemctl status nftables
```

出力には“Loaded: masked” と “Active: inactive”が含まれている必要があります。

# 11.2.4.5. rootユーザーのパスワードを設定する

各ノードに対して、以下のコマンドを実行する必要があります。

このチュートリアルでは、SIOS Protection Suite for LinuxのGUIにログインするために root ユーザーを使用し、操作を実行するためにパスワードを必要とします。 root ユーザーのパスワードを定義してください。

1. root ユーザーのパスワードを設定します.

```
# passwd
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

# 11.2.4.6. x11をインストールする

SIOS Protection Suite for the Linux GUIは、X Window Systemを使用します。これは各ノードにインストールする必要があります。

- マシンにx11をインストールします。
- 関連パッケージにx11をインストールし、以下の手順で有効化します。

1. 関連パッケージにx11をインストールします。

```
# yum install  xorg-x11-server-Xorg xorg-x11-xauth -y
Last metadata expiration check: 0:00:51 ago on Wed 18 Nov 2020 12:36:11 AM UT
C.
Dependencies resolved.
================================================================================
====================================
 Package                          Architecture    Version                    Re
pository                           Size
================================================================================
====================================
Installing:
 xorg-x11-server-Xorg             x86_64          1.20.8-6.el8               rh
el-8-appstream-rhui-rpms          1.5 M
 xorg-x11-xauth                   x86_64          1:1.0.9-12.el8             rh
el-8-appstream-rhui-rpms           39 k
(snip)
  xorg-x11-drv-libinput-0.29.0-1.el8.x86_64                 xorg-x11-drv-ves
a-2.4.0-3.el8.x86_64
  xorg-x11-server-Xorg-1.20.8-6.el8.x86_64                  xorg-x11-server-co
mmon-1.20.8-6.el8.x86_64
  xorg-x11-xauth-1:1.0.9-12.el8.x86_64                      xorg-x11-xkb-util
s-7.7-28.el8.x86_64
Complete!
```

2. /etc/ssh/sshd_config の X11Forwarding パラメーターをチェックして、 yesに設定されていることを確認します。yesに設定されていない場合は、ファイルを編集します。

```
1 # grep -e X11Forwarding /etc/ssh/sshd_config
2 X11Forwarding yes
```

3. システムがX11で起動されていると仮定して、起動時にGUIセッションを開始します。そうでない場合は、このステップは無視できます。

```
# systemctl set-default graphical.target
Removed /etc/systemd/system/default.target.
Created symlink /etc/systemd/system/default.target → /usr/lib/systemd/system/g
raphical.target.
```

4. マシンを再起動します。

```
1 # shutdown -r now
```

# 11.2.5. LifeKeeper for Linux のインストール

各ノードに対して、以下のコマンドを実行する必要があります。

## インストール用 ISO イメージのマウント

1. ISO イメージとライセンスキーファイルをローカル環境にダウンロードします。
2. ISO イメージを各ノードにコピーします。
3. ライセンスファイルを各ノードにコピーします。
4. 各ノードにログインし、インストーラーイメージをマウントします。

```
$ ls
evalkeys.txt  sps951.img
$ sudo su
# mount -o loop -t iso9660 sps951.img /media
mount: /media: WARNING: device write-protected, mounted read-only.
```

5. イメージの内容を確認します。

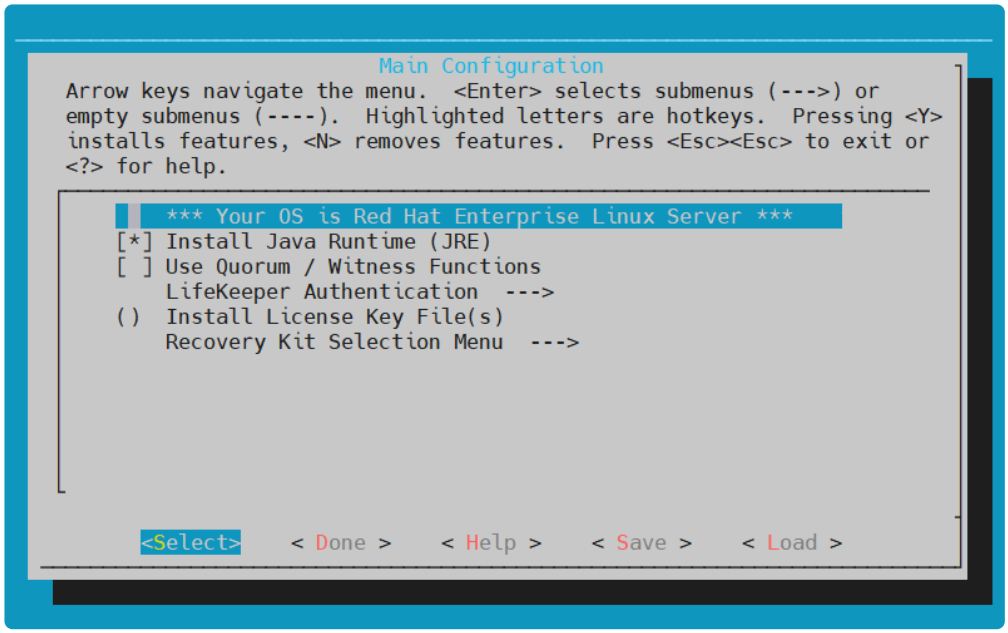
```
# cd /media
# ls
CentOS     core                              java    README        setuplibs
Chef       create_response_file              kits    RHAS          shfuncs
common     HADR-9.5.1-7154.src.rpm           OEL     sapconnector  SuSE
COPYRIGHT  HADR-generic-9.5.1-7154.x86_64.rpm quorum  setup         TRANS.TBL
```

## LifeKeeper for Linux のインストール

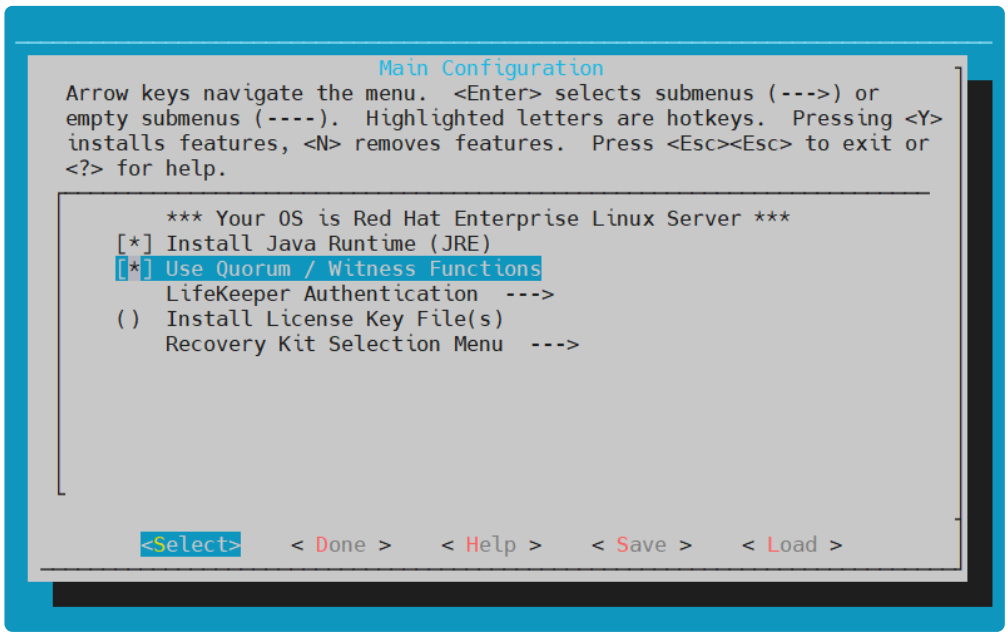
インストーラーは、 `/media/setup` にあります。setup スクリプトを実行して、 LifeKeeper for Linux をインストールします。

```
# ./setup
SIOS Protection Suite for Linux Setup
Validating files.............................OK
Collecting system information.................................done.
Preparing configuration information............done.
```

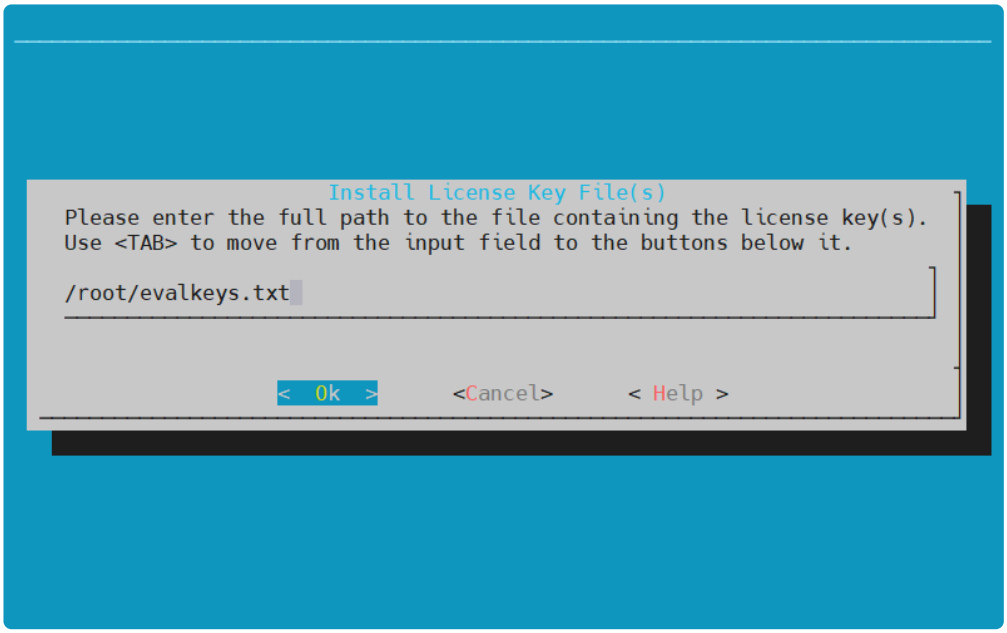
1. GUI インストーラーが以下のように起動します。




2. 「Use Quorum / Witness Functions」にチェックを入れます。

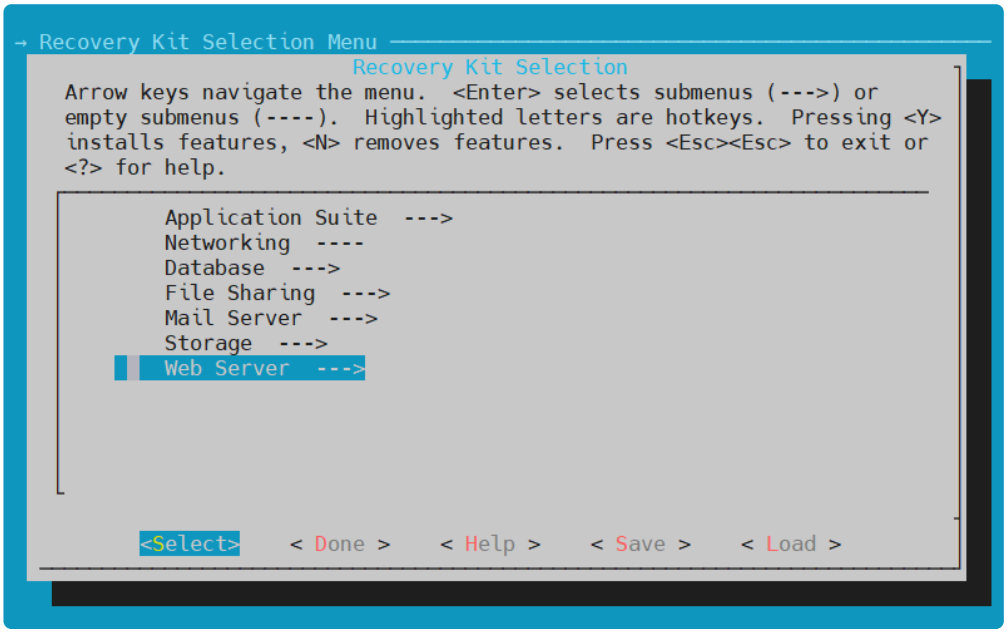



3. ライセンスファイルの場所を指定します。

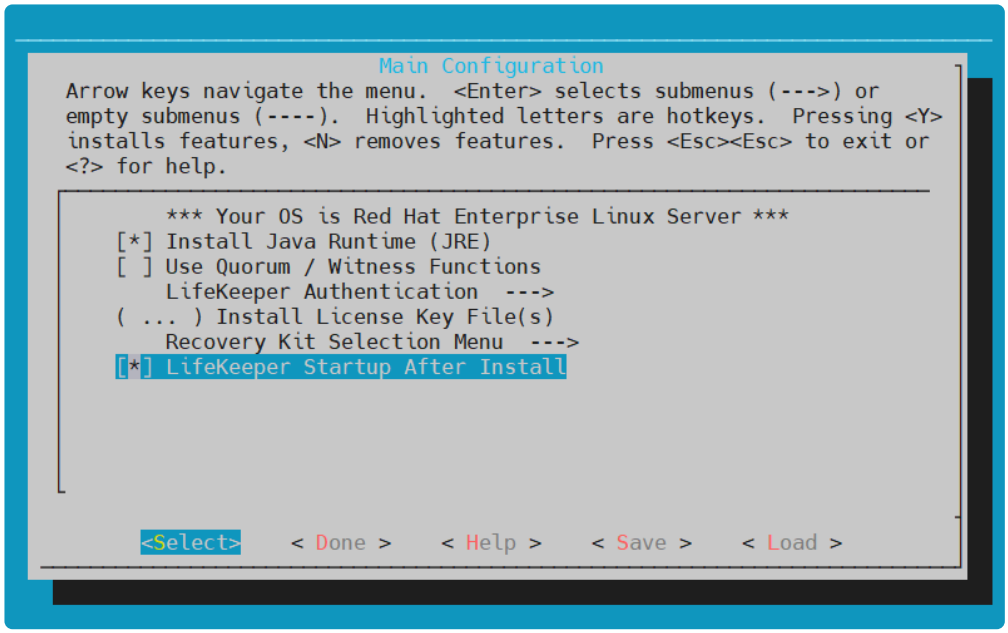



4. インストールする Recovery Kit を選択します。

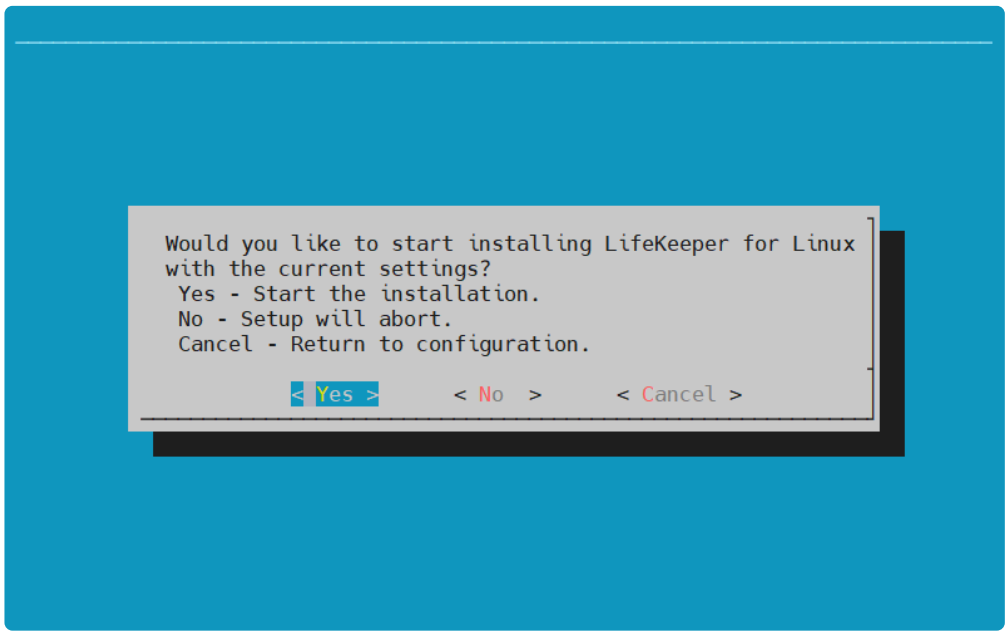
> ! node-c（Witnessノード）では、Recovery Kit を選択しないでください。



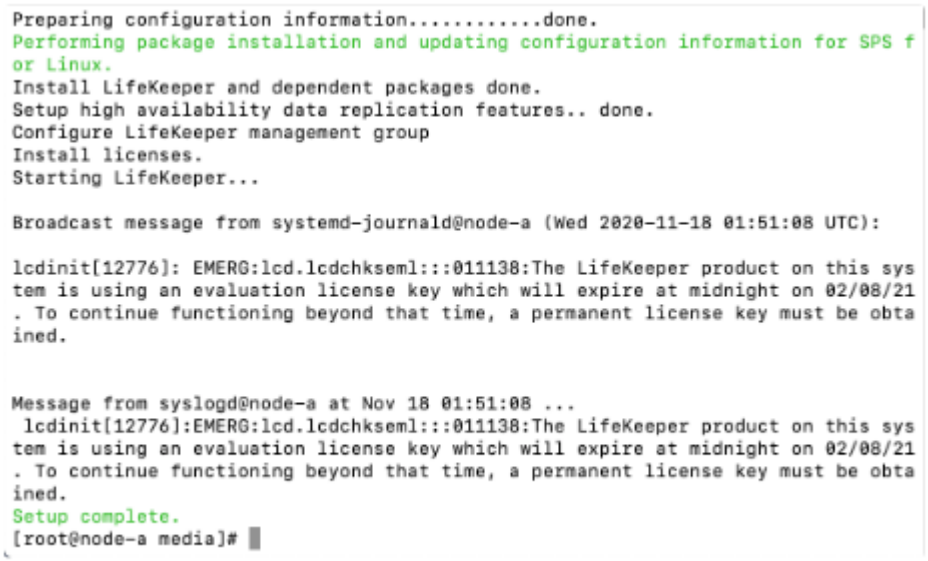

5. 「LifeKeeper Startup After Install」にチェックを入れ、[Done] を選択します。




6. インストールプロセスを開始します。




7. これでインストールは完了です。node-b と node-c についても同じ手順を繰り返します。




## 完了する必要のある環境固有の追加タスク

AWS を使用している場合は、次の手順を実行します。

- PINGブロードキャストを無効にする
- AWS CLIをインストールする
- EC2 Recovery Kitを使用するためのアクセス権限を割り当てる

Azure を使用している場合は、次の手順を実行します。

- PINGブロードキャストを無効にする

# 11.2.5.1. Amazon Web Services

# 11.2.5.1.1. AWS CLIをインストールする

各ノードで以下のコマンドを実行する必要があります。

まず、AWSCLIをインストールする必要があります。詳しい手順は、 AWS CLI の最新バージョンをインストールまたは更新します に記載されています。

CLIをインストールしたら、CLIの場所を `/etc/default/LifeKeeper` ファイルに追加します。

1. `aws` CLIの場所を確認します。

```
# curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
# unzip awscliv2.zip
# sudo ./aws/install
You can now run: /usr/local/bin/aws --version
```

2. `/etc/default/LifeKeeper` を更新して、この場所をPATHに含めます。

```
# vi /etc/default/LifeKeeper
# grep -e PATH= /etc/default/LifeKeeper
PATH=/opt/LifeKeeper/bin:/bin:/usr/bin:/usr/sbin:/sbin:/usr/local/bin
```

**注記:** 実行される AWS CLI のバージョンは、AWS CLI の実行可能ファイルを含むパスの最初の項目に依存します。

すなわち、AWS CLI バージョン 1 のパスを最初に指定すると、AWS CLI バージョン 1 が使用されます。 AWS CLI バージョン 2 のパスを最初に指定すると、AWS CLI バージョン 2 が使用されます。

# 11.2.5.1.2. Recovery Kit for EC2を使用するためのアクセス権限を割り当てる

> ! 以下のステップを完了するには、AWS CLIを各ノードにインストールする必要があります。詳細については、完了する必要のある環境固有の追加タスク を参照してください。

Recovery Kit for EC2を利用するには、RouteTableのエントリーの更新、ENI（Elastic Network Interface）の再割り当てを許可されたIAMロールがインスタンスに必要です。

- ルートテーブルシナリオ（バックエンドクラスタ）
  - ec2:DescribeRouteTables
  - ec2:ReplaceRoute
  - ec2:DescribeNetworkInterfaceAttribute
  - ec2:ModifyNetworkInterfaceAttribute

- Elastic IP シナリオ（フロントエンドクラスタ）
  - ec2:DescribeAddresses
  - ec2:AssociateAddress
  - ec2:DisassociateAddress

そのためには、以下のようなポリシーを作成し（アクセスできるリソースを制限することが望ましい場合があります）、それをIAMロールに割り当てます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "ec2:DisassociateAddress",
                "ec2:DescribeAddresses",
                "ec2:DescribeNetworkInterfaceAttribute",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:AssociateAddress",
                "ec2:DescribeRouteTables",
                "ec2:ReplaceRoute"
            ],
            "Resource": "*"
        }
    ]
}
```

IAMロールを定義したら、それをEC2インスタンスに割り当てます。

# 11.2.5.1.3. AWS – PINGブロードキャストを無効にする

LifeKeeperをクラウド環境で使用する場合、サブネット上の他のマシンを見つけるためにブロードキャストpingを使用できない場合があります。そのため、以下のようにLifeKeeperの構成を変更することで、これを無効にする必要があります。

1. 現在の構成を確認します。

```
# grep -e NOBCASTPING /etc/default/LifeKeeper
NOBCASTPING=0     # Can be used to disable the broadcast ping mechanism
```

2. NOBCASTPING を1に設定する必要があります。

```
# cp -p /etc/default/LifeKeeper /etc/default/LifeKeeper.org
# sed -i -e 's/NOBCASTPING=0/NOBCASTPING=1/g' /etc/default/LifeKeeper
```

3. 変更を確認します。

```
# grep -e NOBCASTPING /etc/default/LifeKeeper
NOBCASTPING=1     # Can be used to disable the broadcast ping mechanism
```

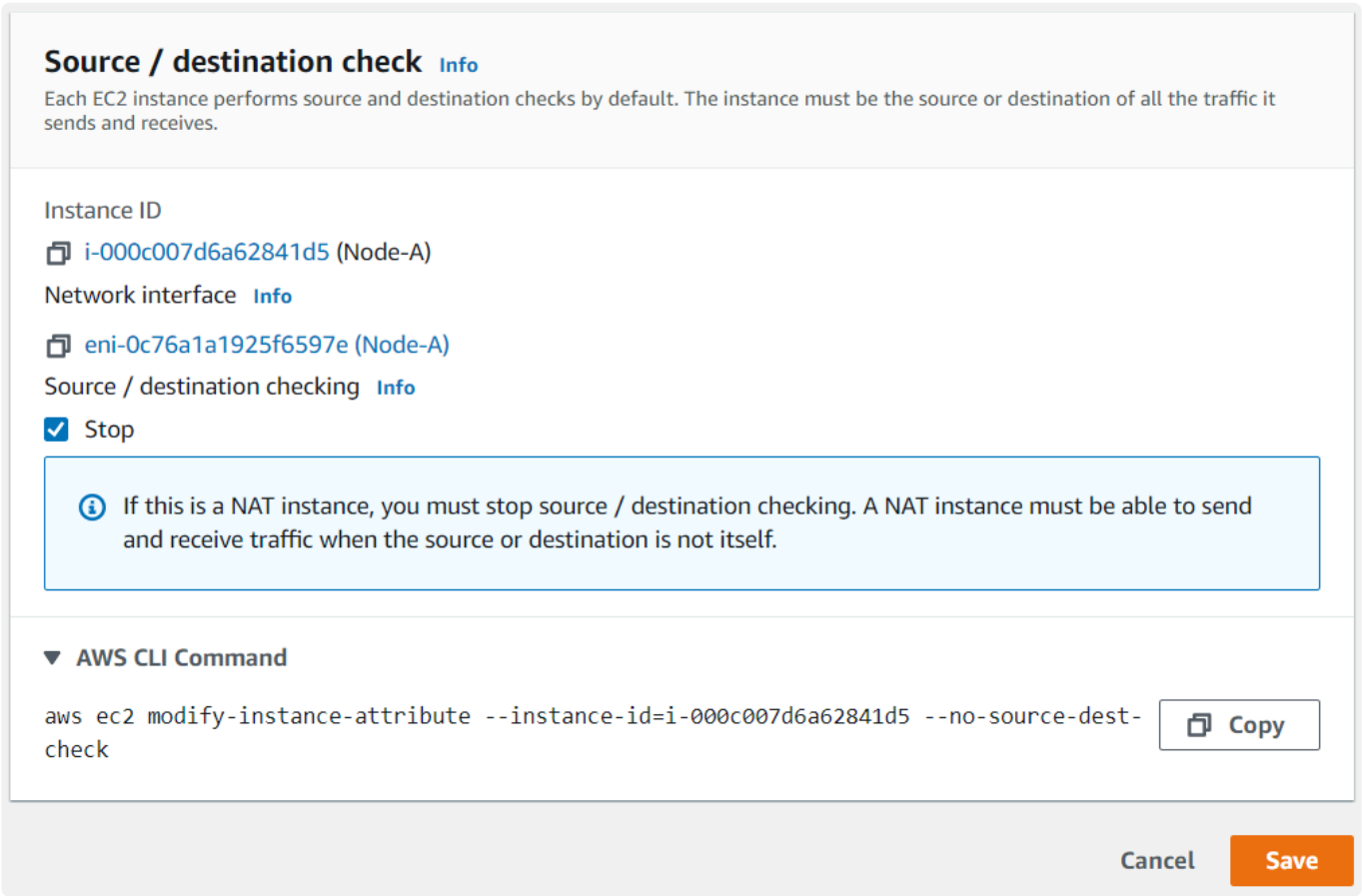
# 11.2.5.1.4. AWS – 送信元/宛先チェックを無効にする

EC2 Recovery Kitリソースが作成されるAWS EC2インスタンスそれぞれについて、送信元/宛先チェックを無効にする必要があります。

> 注記 ：SIOS Protection Suite for Linux v9.6.0以降では、LifeKeeper EC2リソースを作成または拡張するときに、EC2 Recovery Kitによって各サーバーで送信元/宛先チェックが自動的に無効になるため、これらの手動の手順は必要ありません。EC2 Recovery Kitが送信元/宛先チェックを自動的に無効にできるようにするには、ec2:DescribeNetworkInterfaceAttributeおよびec2:ModifyNetworkInterfaceAttribute権限をEC2インスタンスのIAMロールに追加する必要があります。

AWSコンソールで、 **[サービス]** → **[EC2]** → **[インスタンス]** に移動します。送信元/宛先チェックを無効にするEC2インスタンスを右クリックし、 **[Networking]** を選択して、 **[Change source/destination check]** をクリックします。

表示されるダイアログで、 **[Stop]** の横のチェックボックスをクリックし、 **[Save]** をクリックしてインスタンスの設定を更新します。



# 11.2.5.2. Microsoft Azure

# 11.2.5.2.1. Azure – PING ブロードキャストを無効にする

LifeKeeper がクラウド環境で使用されている場合、ブロードキャスト ping を使用してサブネット上の他のマシンを検索できない場合があります。 したがって、LifeKeeper の設定を次のように変更して、これを無効にする必要があります。

1. 現在の構成を確認してください。

```
# grep -e NOBCASTPING /etc/default/LifeKeeper
NOBCASTPING=0     # Can be used to disable the broadcast ping mechanism
```

2. NOBCASTPING は 1 に設定する必要があります。

```
# cp -p /etc/default/LifeKeeper /etc/default/LifeKeeper.org
# sed -i -e 's/NOBCASTPING=0/NOBCASTPING=1/g' /etc/default/LifeKeeper
```

3. 変更を確認します。

```
# grep -e NOBCASTPING /etc/default/LifeKeeper
NOBCASTPING=1     # Can be used to disable the broadcast ping mechanism
```

# 11.2.6. ログインと基本設定タスク

これまでに、3つのノードにそれぞれLifeKeeper for Linuxがインストールされ、使用できるようになっているはずです。

## GUIを使用したLinuxノードへの接続

LifeKeeperには、クラスターを構成するためのGUIアプリケーションがあります。LifeKeeper GUIへのアクセスの詳細については、 x11転送を使用したLinuxノードへの接続 を参照してください。

## 基本設定

このチュートリアルでは、以下のIPアドレス範囲を持つ3つのノードが存在します。

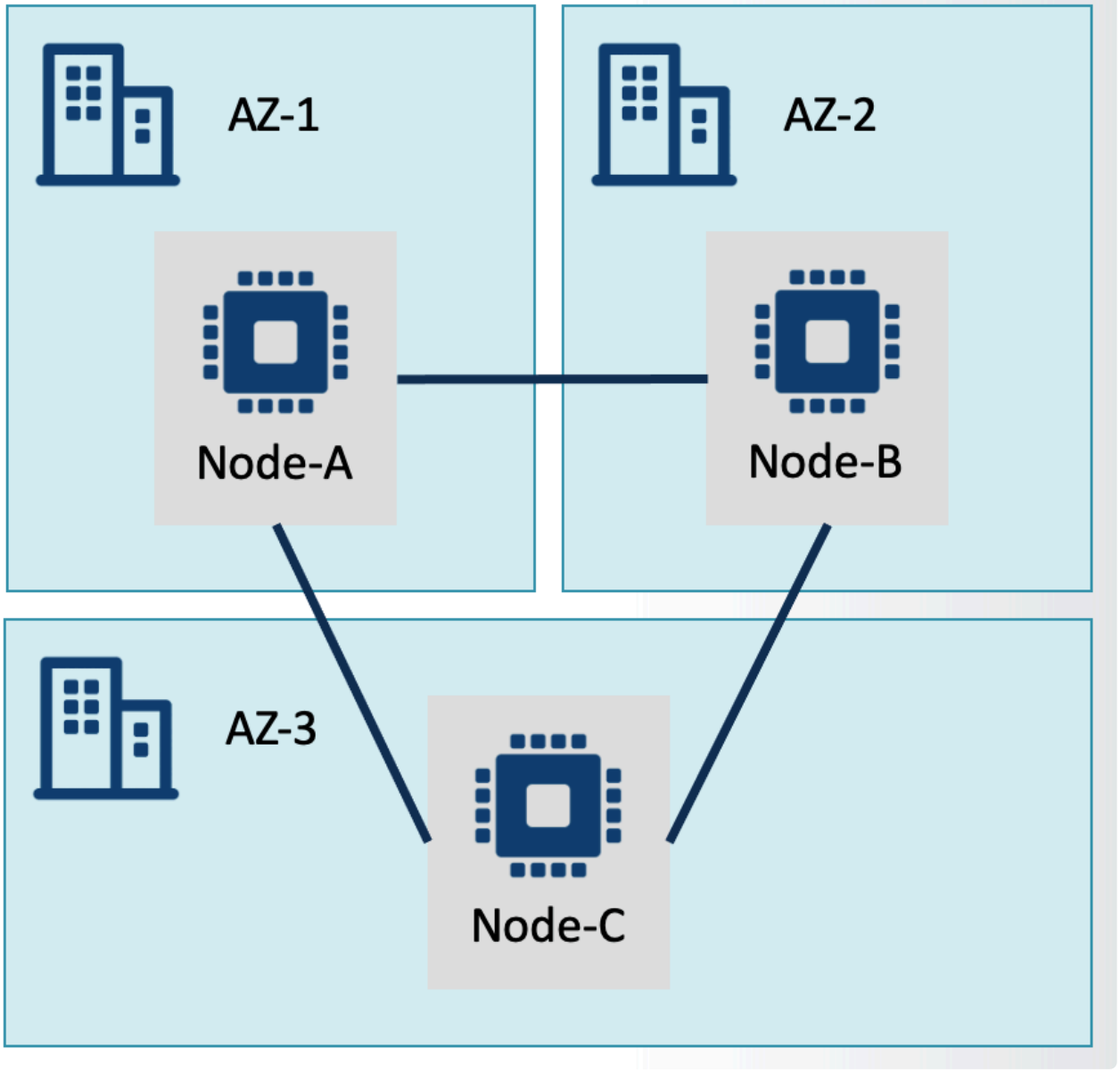
| ノード | IPアドレス |
|---|---|
| node-a (稼働系) | 10.20.1.10 |
| node-b (待機系) | 10.20.2.10 |
| node-c (Quorum / Witness) | 10.20.3.10 |

基本設定は、以下の手順で実施します。

- 1番目のノード(node-a)に接続する
- 他のノードに接続する
- コミュニケーションパスを定義する

**Note:** クラスターには、node-c（Quorum/Witness）を含めて3つのノードがあります。したがって、ノードのすべてのペア間に双方向のコミュニケーションパスを作成するには、3つのコミュニケーションパスを定義する必要があります。

- `node-a <—> node-b`
- `node-a <—> node-c`
- `node-b <—> node-c`


AZ-1
AZ-2
Node-A
Node-B
AZ-3
Node-C

# 11.2.6.1. x11転送を使用したLinuxノードへの接続

X Window Systemは、Unixベースのオペレーティングシステム上でビットマップベースのグラフィカルユーザーインターフェースを実現するために使用されます。これは一般に「X11」と呼ばれています。X11の興味深い使用法の1つは、「ウィンドウ」を端末（オペレーターが使用するマシン）に転送する機能です。元のX11プロセスはサーバー上で実行されますが、X11ユーザーインターフェースはクライアントマシン端末に表示されます。これを「X11転送」と言います。

X11転送を使用するには、ローカルマシンでX11グラフィックサーバーを実行する必要があります（これは、実際にはクライアントマシンで実行されますが、X11サーバーと呼ばれます）。

LifeKeeper GUIはX Window Platformで実行されるため、X11ソフトウェアを実行するクライアントが必要です（これらのノードでGUIログインが直接使用されている場合を除きます）。

## Microsoft Windowsにクライアントソフトウェアをインストールする

Microsoft Windowsを「x11サーバー」として使用するには、Microsoft Windowsクライアントにソフトウェアをインストールする必要があります。以下のページでは、Windowsクライアントにソフトウェアをインストールするために必要な基本的な手順について説明します。

- Microsoft WindowsでのX Windowクライアントソフトウェアのセットアップ

## X11転送を受け入れるようにLinuxノードを設定する

1. 前の手順で設定した一般ユーザーとして、環境変数DISPLAYの現在の値を見つけます。

   ```
   $ echo $DISPLAY
   localhost:10.0
   ```

2. ユーザーのxauthキーの現在のリストを表示し、X11が転送されている現在のディスプレイに関連付けられているキー（この例では10）を特定します。

   ```
   $ xauth list
   node-a/unix:11 MIT-MAGIC-COOKIE-1 7289ba26871a37b94ff359df829e2686
   node-a/unix:10 MIT-MAGIC-COOKIE-1 bae592c842916f23e3ba066ba594c5d0
   ```

3. rootユーザーになります。

   ```
   $ sudo -i
   ```

4. rootユーザーのDISPLAY環境変数を表示します。

   ```
   # echo $DISPLAY
   ```

```
localhost:10.0
```

注記：rootユーザーにDISPLAY変数が設定されていない場合は、手順1で確認したのと同じ値に設定します。

```
# export DISPLAY=localhost:10.0
```

5. rootユーザーの現在のxauthキーのリストを表示します。

```
# xauth list
node-a/unix:1 MIT-MAGIC-COOKIE-1 5b249d176f8da2f7803343f6830002eb
node-a/unix:2 MIT-MAGIC-COOKIE-1 e62f7abaeda1d200576aac40aec2ba83
```

手順1でディスプレイのエントリーが見つからない場合は、手順2で出力されたxauthリストの該当エントリーをrootユーザーの鍵リストに追加します。

```
# xauth add node-a/unix:10 MIT-MAGIC-COOKIE-1
bae592c842916f23e3ba066ba594c5d0
```

6. DISPLAY変数とxauthキーがrootユーザーに正しく設定されると、SSHセッションからrootとしてグラフィカルアプリケーションを実行できるようになります。

```
# /opt/LifeKeeper/bin/lkGUIapp &
[1] 23957
openjdk version "12.0.2" 2019-07-16
OpenJDK Runtime Environment (build 12.0.2+10)
OpenJDK 64-Bit Server VM (build 12.0.2+10, mixed mode, sharing)
Setting up secure random number generator
Random number setup completed
Connecting to rmi://node-a:82/LKRemoteInterface at:
      Sat Mar 13 04:23:19 UTC 2021
Connection to node-a succeeded at:
      Sat Mar 13 04:23:20 UTC 2021
```

# 11.2.6.2. Microsoft Windowsにクライアントソフトウェアをインストールする

Windows クライアントから ssh を使って Linux ノードに接続する のステップを踏んで、 PuTTY のインストールと設定を行いました。ここでは、X11転送を有効にして、ノードに接続し、GUIベースのアプリケーションを使用できるようにする方法を説明します。

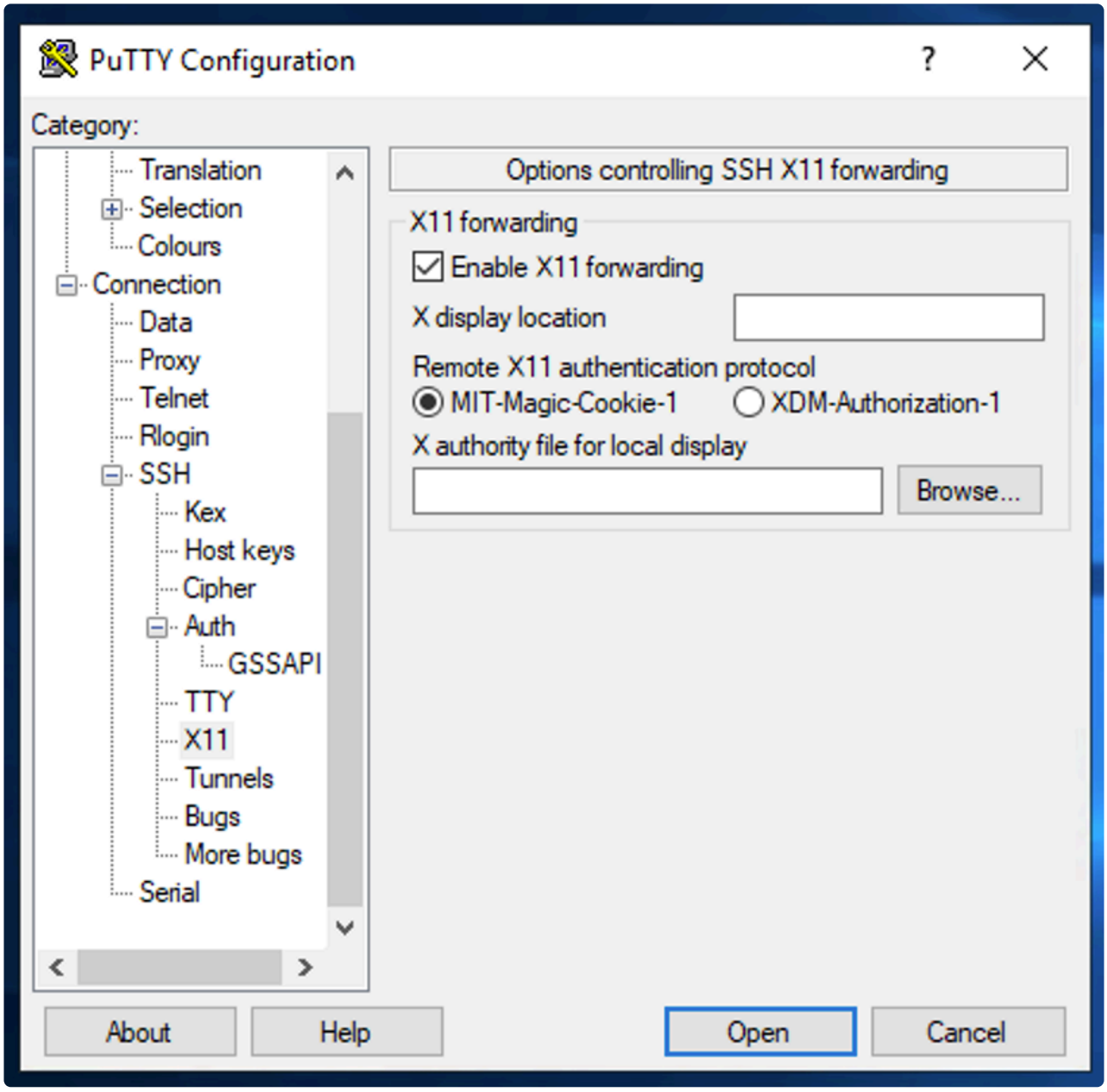
## ツールのダウンロードとインストール

以下の追加ソフトウェアをダウンロードし、インストールします。

- XMing (Microsoft Windows上のX11サーバー、その他のX11 Windowsクライアントが利用可能です。)

## PuTTYの設定

1. PuTTYを起動し、Connectionを表示します > SSH > Auth.
2. ディスプレイの接続 > SSH > X11 and check ‘Enable X11 forwarding’.

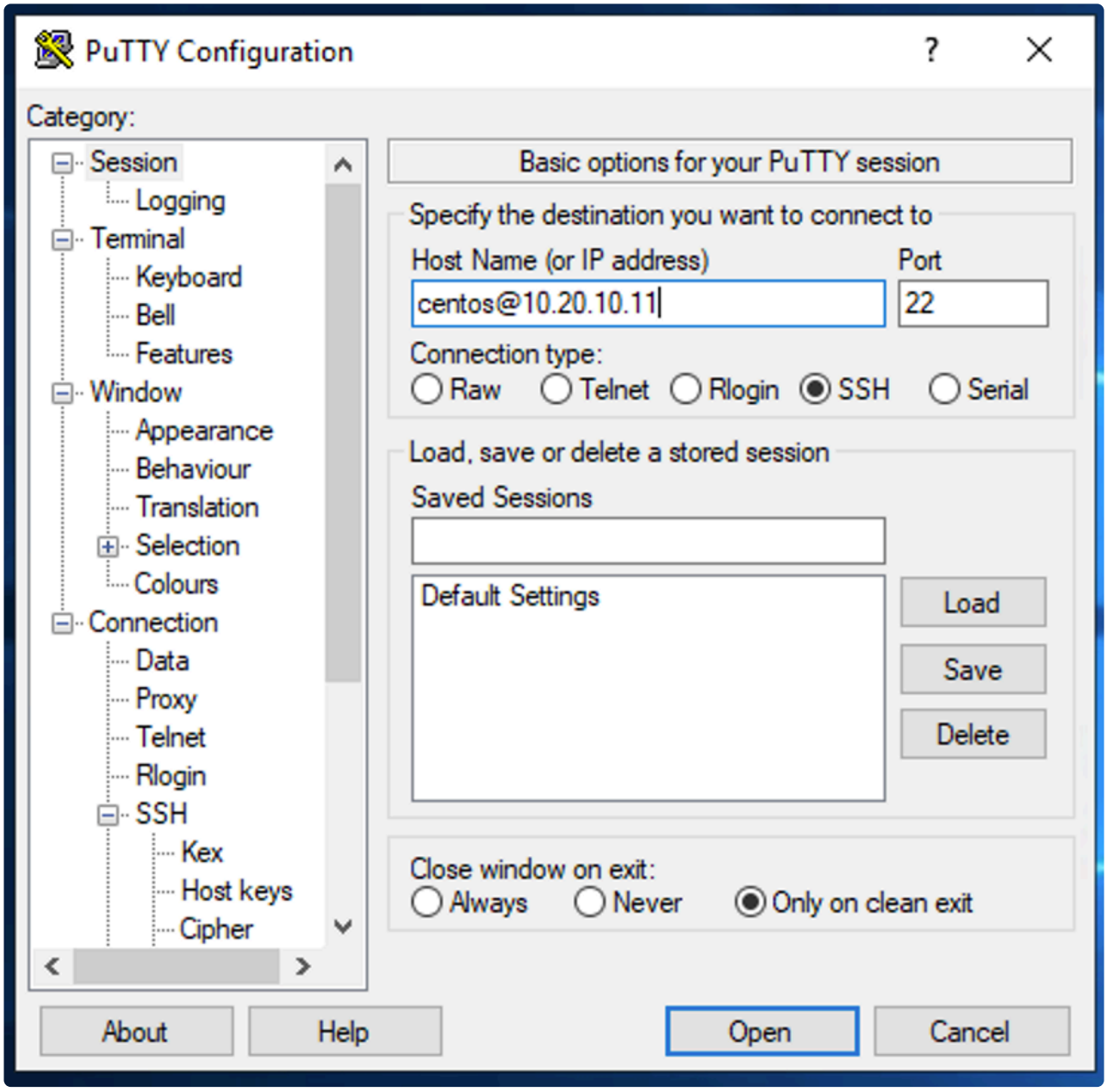



3. 設定は、「セッション」ページで保存することをお勧めします (そうでない場合は、これらのステップを毎回繰り返す必要があります).

## Linuxインスタンスに接続する

1. Xmingを起動します（Mingアイコンがシステムトレイに表示されていることを確認してください）

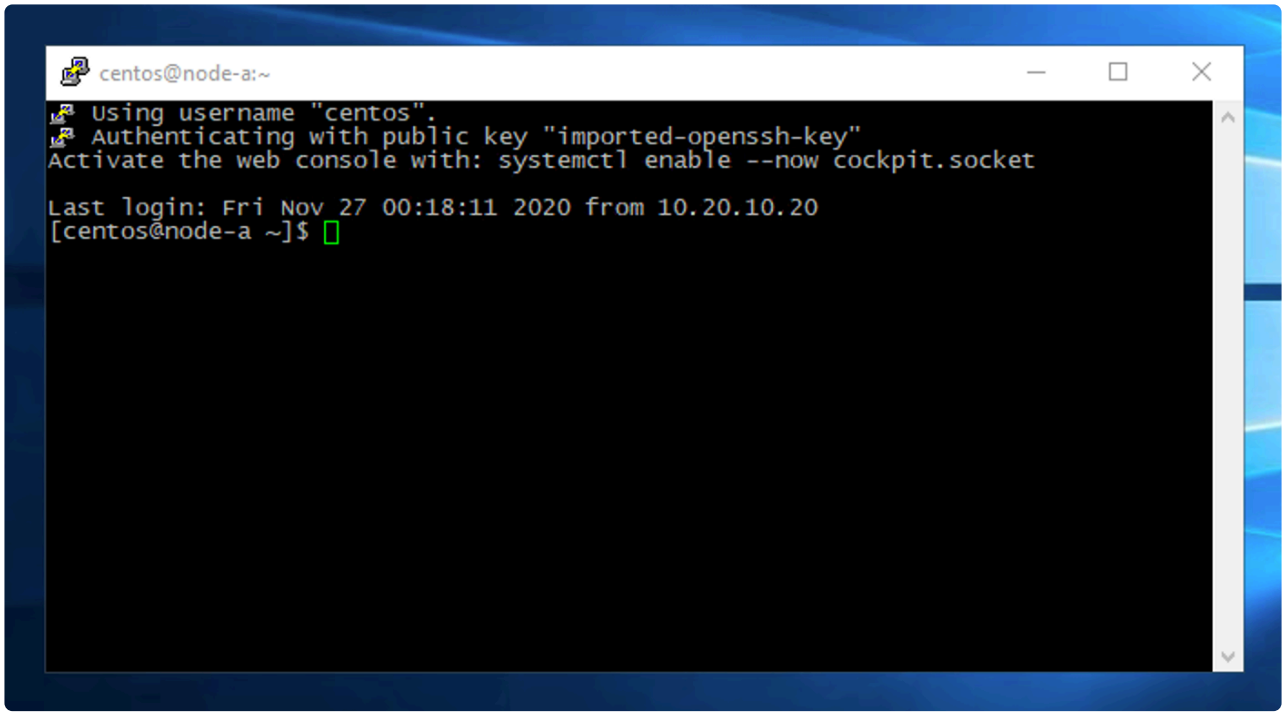
2. PuTTYからセッションを開始します。




3. サーバーが以前に接続したことがないことを示すセキュリティアラートを確認します。「はい」を選択して続行します。

4. 「~/.Xauthorityが存在しない」という警告が表示された場合は、touch でファイルを作成し、ノードに再接続してください。

   これで接続されたはずです。

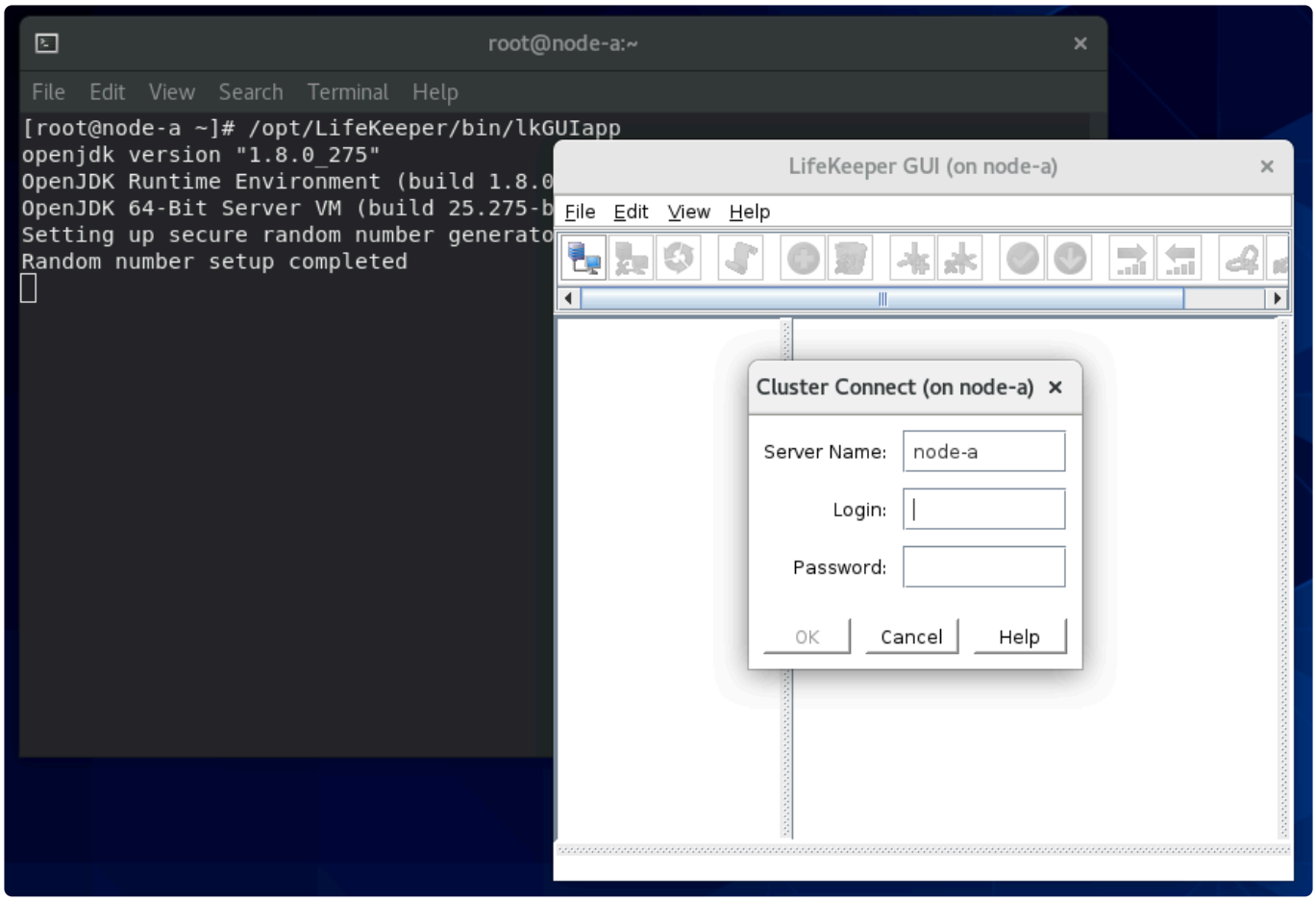

centos@node-a:~
Using username "centos".
Authenticating with public key "imported-openssh-key"
Activate the web console with: systemctl enable --now cockpit.socket
Last login: Fri Nov 27 00:18:11 2020 from 10.20.10.20
[centos@node-a ~]$

# 11.2.6.3. 1番目のノードに接続する (node-a)

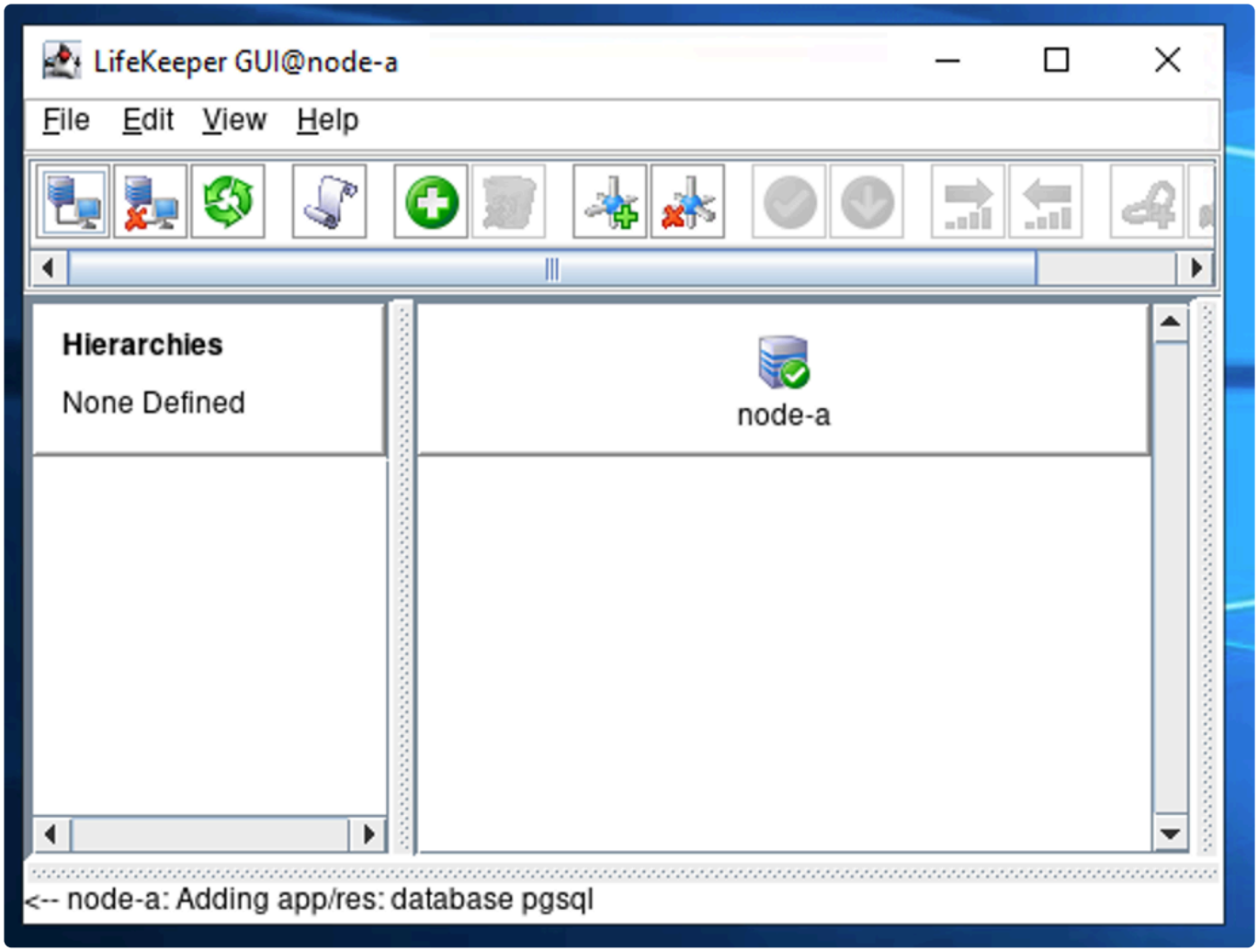
「X11転送」を有効にしてノードにログインしたら、次のコマンドを入力してSIOS Protection Suite for Linux GUIを起動します。

```
# /opt/LifeKeeper/bin/lkGUIapp &
```

ログインダイアログのポップアップで、rootのユーザー名とパスワードを入力し、ログインします。

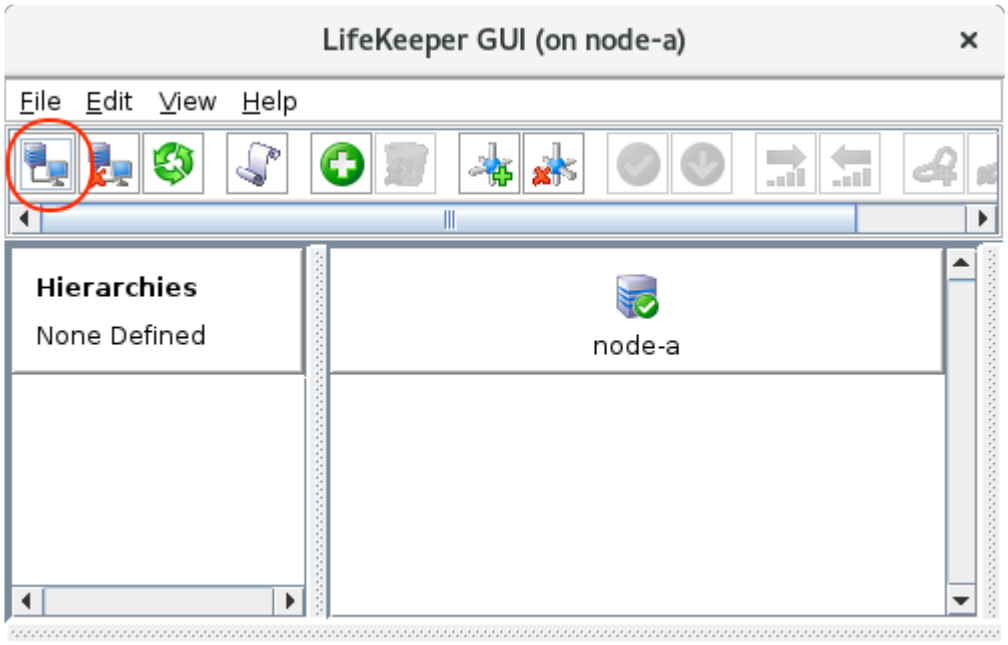



ログインに成功すると、GUIにnode-aが表示されます。

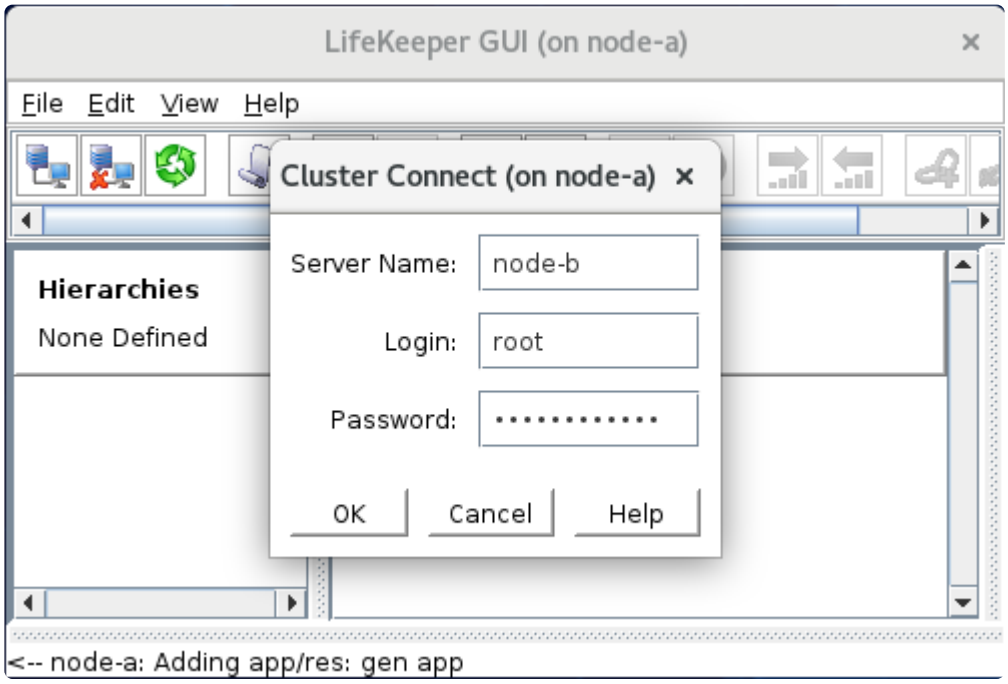

LifeKeeper GUI@node-a
File Edit View Help
Hierarchies
None Defined
node-a
<-- node-a: Adding app/res: database pgsql

# 11.2.6.4. 他のノードに接続する (node-b and node-c)

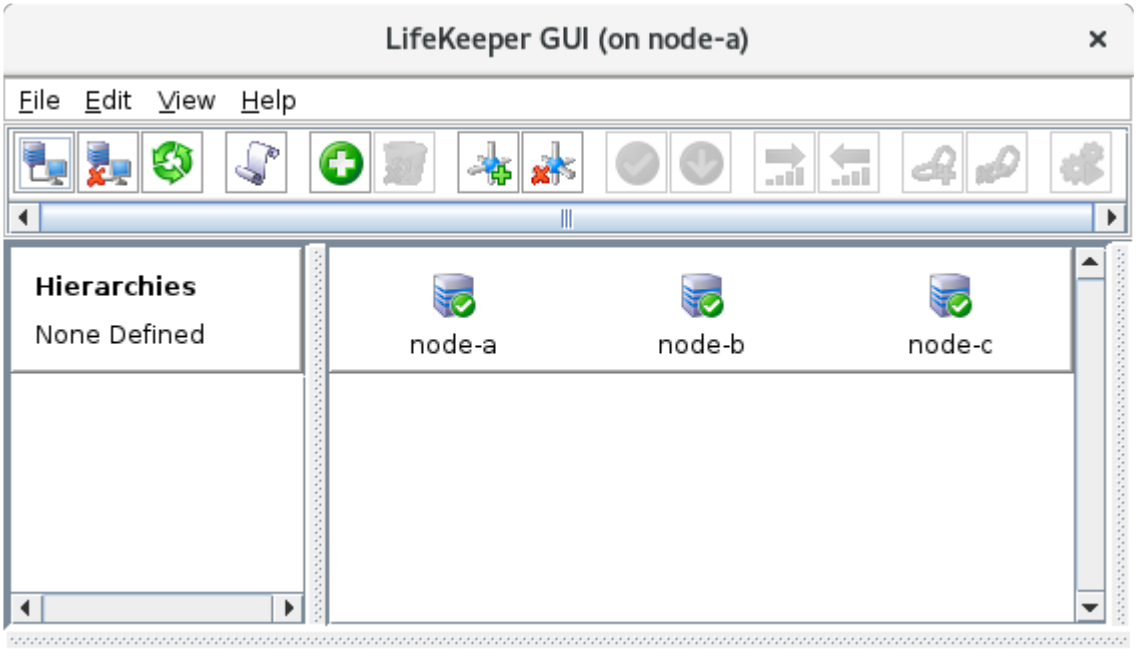
1. LifeKeeper（ツールバーの最初のアイコン）で [接続] を選択します。




2. ログインダイアログが再度表示されます。node-bとrootのユーザ名とパスワードを入力し、2番目のノードに接続します。




3. 同じように node-c に接続します。これで、LifeKeeperのGUIに3つのノードが表示されました。




<-- node-c: Adding app/res: scsi device

# 11.2.6.5. コミュニケーションパスを定義する

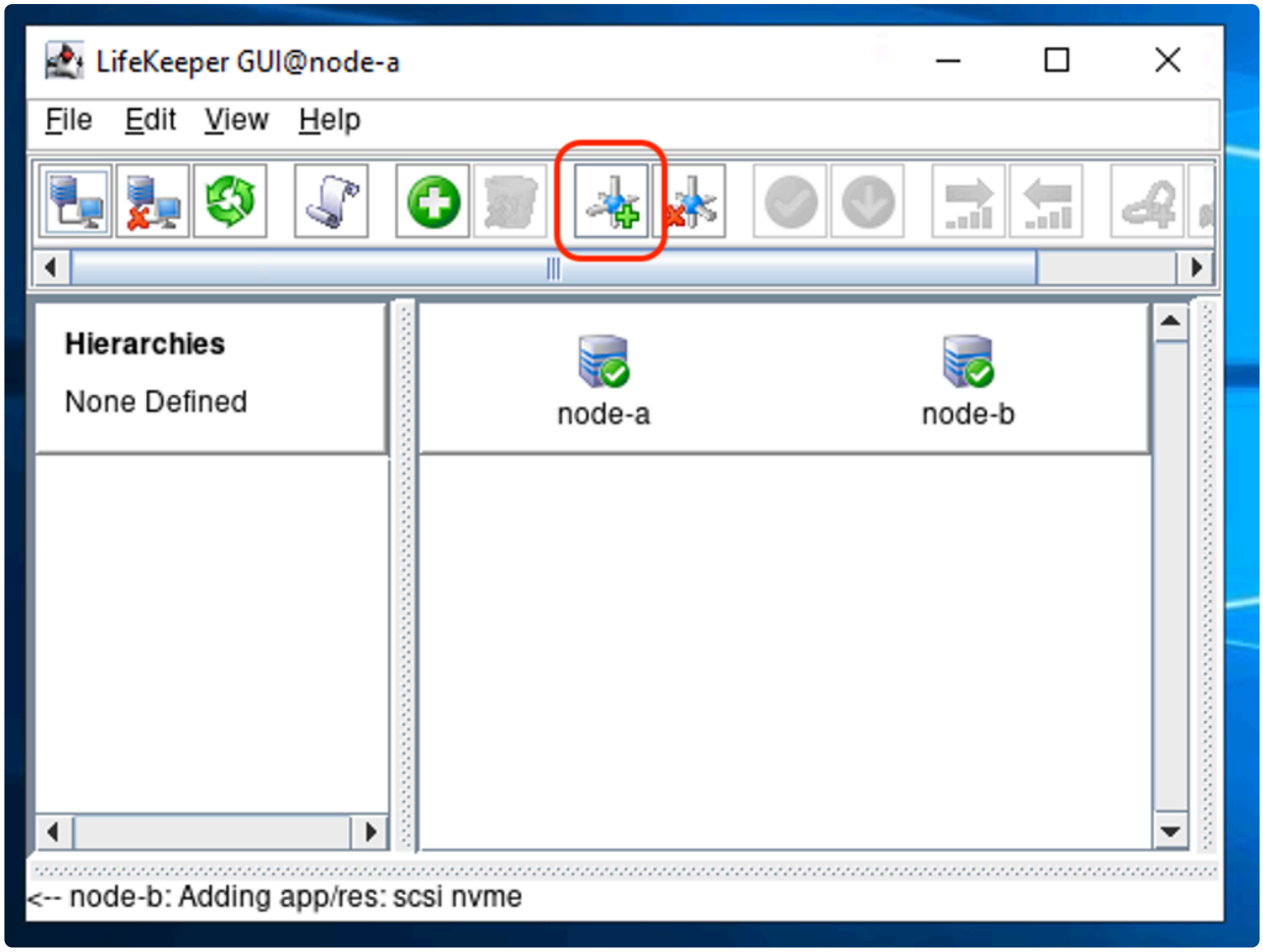
コミュニケーションパスは3つのノードが相互に通信し、ノード間でデータを（データレプリケーションの目的で）転送できるようにするメカニズムです。 一般的なセットアップ、特にオンプレミス環境では、単一障害点を回避するために複数のコミュニケーションパスが必要となります。この例では、各ノードのネットワークカードや、これらのノード間を切り替えるネットワークルーターが障害点となる可能性があります。

ただし、クラウド環境では、基盤となるネットワークレイヤーが抽象化されているため、ノード間のコミュニケーションパスは1つで十分な場合があります（これらのレイヤーは見ることができません）。

次に、ノードのすべてのペア間のコミュニケーションパスを定義します。

> **Important Note:** 3つ以上のノードで構成されるクラスターでは、 ノードのすべてのペア間で コミュニケーションパスを定義する必要があります。3ノードクラスターでは、これは3つのコミュニケーションパスを定義する必要があることを意味します（node-a ↔ node-b、node-a ↔ node-c、node-b ↔ node-c）。

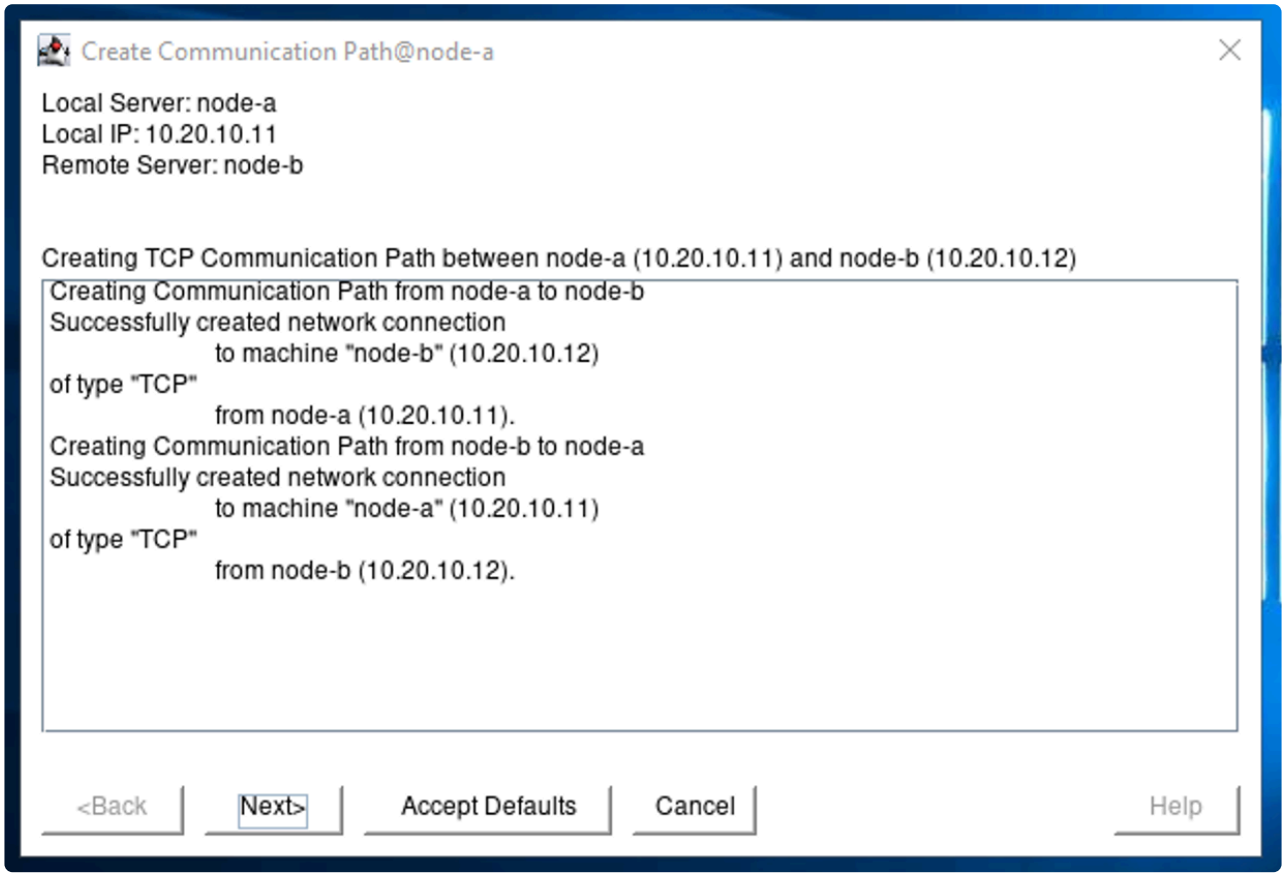
1. [Add communication path] をクリックします。



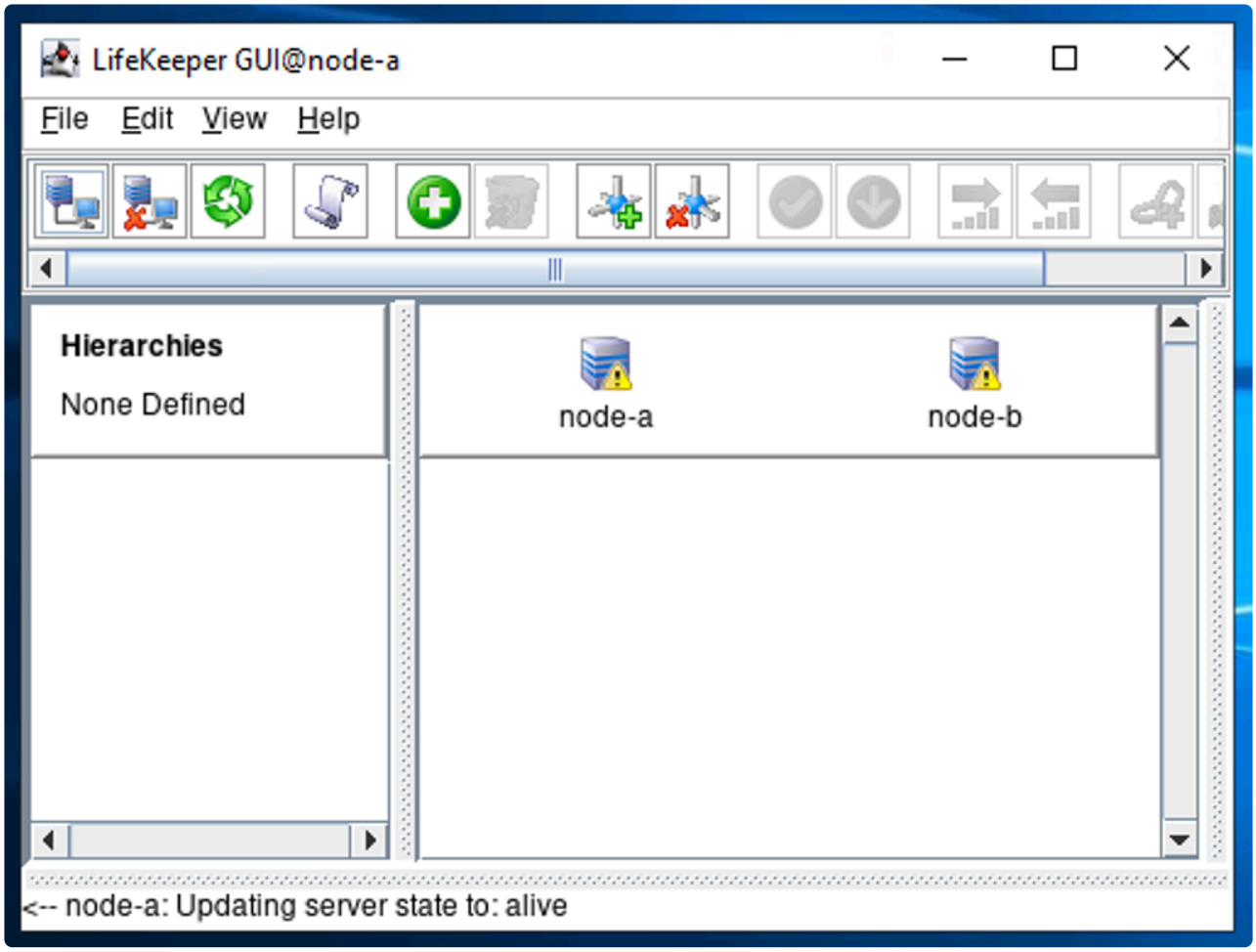

2. 次の値を選択します。ほとんどの場合、これらの値はすでに入力され、選択されているはずです。

| フィールド | 値 |
|---|---|
| Local Server | node-a |
| Remote Server(s) | node-b |
| Device Type | TCP |
| Local IP Address(es) | 10.20.1.10 |
| Remote IP Address | 10.20.2.10 |
| Priority | 1 |

3. 設定内容の概要が表示されます。[Next] を選択します。



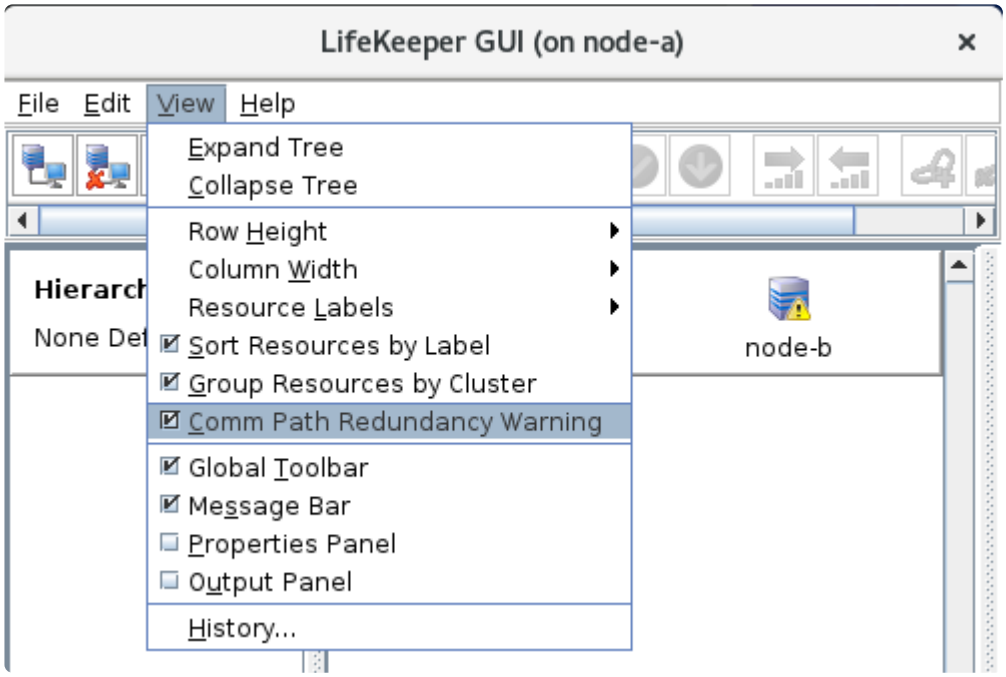

4. A communication path has now been defined.



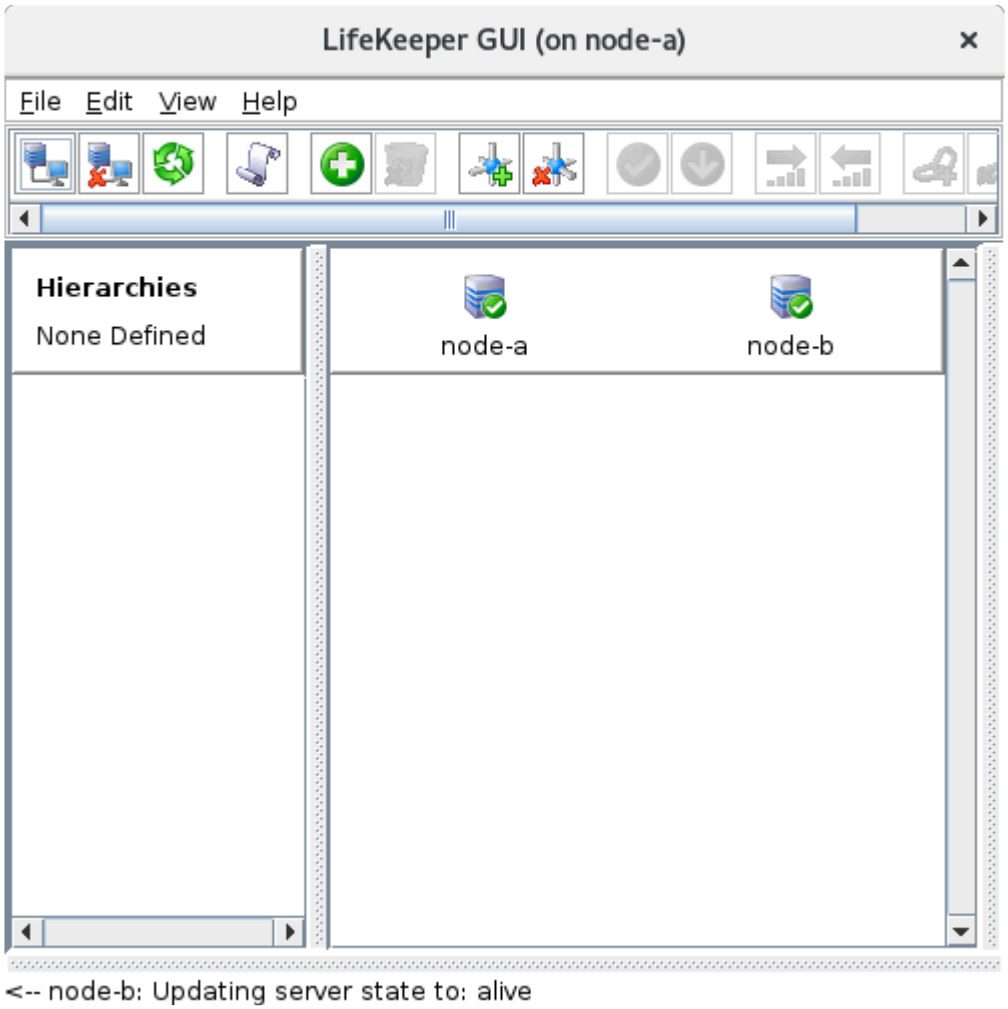

コミュニケーションパスが定義されると、各ノードに黄色の三角形 ( ) が表示される場合があります。これは、コミュニケーションパスしか1つしか定義されていないためです。2つ目のコミュニケーションパスが定義されると、これは緑色のチェックマーク ( ) に変わります。

## コミュニケーションパスの冗長性警告をオフにする

クラウドにデプロイしている場合、この警告は無視できます。警告サインは、LifeKeeper GUIの[View] メニューの [Comm Path Redundancy Warning] のチェックボックスのチェックを外すことでオフにできます。




チェックを外すと、アイコンが から に変わります。



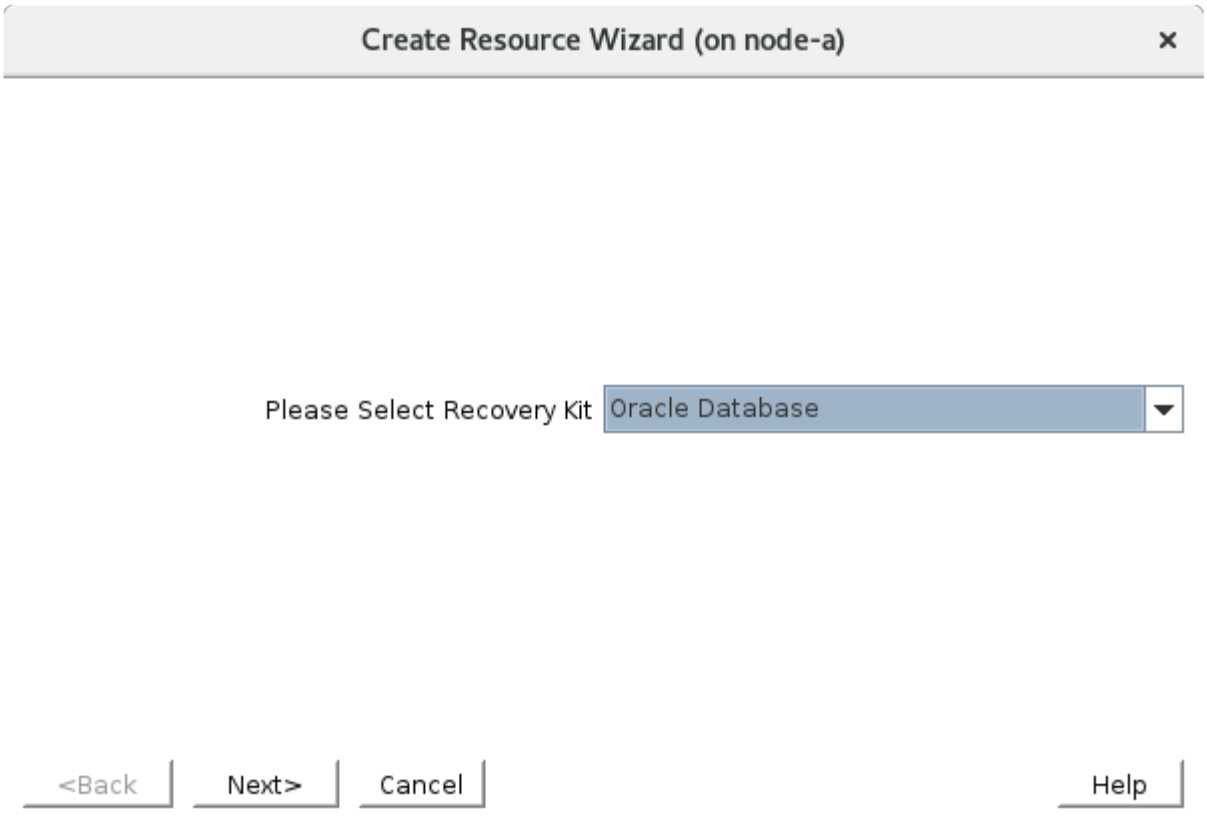
# 11.2.7. リソースの保護

## LifeKeeper を使用した一般的なリソース保護シナリオ

LifeKeeper では、ほとんどの設定をウィザードで完了することができます。ウィザードは保護可能なリソースに関する情報を収集し、一連の簡単な質問に答えることでこれらのリソースを保護することができます。ほとんどの場合、デフォルトのパラメーターを選択するだけで済みます。LifeKeeper リソース作成の標準的な例を以下に示します。

1. リソース作成ウィザード

   a. リソースタイプ（IPリソース、データレプリケーション、Oracle など）を選択します)




   b. スイッチバックタイプを選択します

   c.「アクティブな」サーバー（アプリケーションが現在実行されている場所）を選択します

   d. 保護するIPアドレス、設定ファイルの場所、保護するディスクを選択します

   e. リソースタグ（名前）を選択します

   f. 選択内容を確認します

2. 事前拡張ウィザード

   a. ターゲットサーバー（切り替えやフェイルオーバーの際にアプリケーションホストとして引き

継ぐサーバー）を選択します

b. スイッチバックタイプを選択します

c. 優先度を選択します

d. 選択内容を確認します

3. リソース階層を拡張する

a. 宛先のIPアドレス、設定ファイルの場所、データを複製するディスクを選択します

b. リソースタグ（名前）を選択します

c. 選択内容を確認します

4. 階層の整合性の検証

設定が完了すると、ウィザードは自動的にリソース間の依存関係も定義します（たとえば、データベースリソースを保護するためには、データベースを起動する前に基盤となるディスクをマウントする必要があります）

## ノードの切り替え

これで、リソースを保護する準備が整いました。

前述のとおり、クラウド環境はオンプレミス環境とは動作が異なります。クラウド環境を使用している場合、クラウド環境のアクティブノードへのトラフィックのルーティングに関する関連情報については、クラウド環境でのノードの切り替え をお読みください。IPリソースを作成した後に必要なリソースを設定しますが、次のセクションでIPリソースに選択するパラメータは、使用するクラウド環境によって異なります。

オンプレミス環境で作業している場合は、次のページを確認するだけで十分です。

- IP リソースの作成

クラウド環境で作業している場合は、次のページから該当するシナリオを選択してください。

- クラウド環境でのノードの切り替え

IPリソースの保護が完了したら、スイッチオーバー（スタンバイノードがアクティブノードになる）を開始し、機能をテストすることをお勧めします。

- スタンバイノードに切り替えて、スイッチオーバーが機能していることを確認する

# ディスクレプリケーションの設定

ほとんどのシナリオでは、ノード間でデータレプリケーションを定義する必要があります（特にクラウドで評価する場合）。以下の手順に従って、ノード間のデータレプリケーションを定義してください。

- ノード間のデータレプリケーションはどのように行われるのか？
- ファイルシステムのデータレプリケーションを作成する方法

# リソースを保護する

基本的な準備が完了したら、今度はデータベースやアプリケーションなどのリソースを保護することができます。

- Oracle リソースの保護
- Quick Service Protection を使用した MSSQL の保護
- PostgreSQL リソースの保護
- NFS リソースの保護
- SAP リソースの保護
- SAP HANA リソースの保護

# 11.2.7.1. IP リソースの作成

LifeKeeper 上に作成する最初のリソースは、IP リソースである必要があります。前述 したように、クライアントは仮想 IP アドレスを使用してアクティブノードを見つけることができます。以下は、このセクションで参照されるIP アドレスのリストです。

## ノードのIPアドレス

| フィールド | 値 |
|---|---|
| node-a | 10.20.1.10 |
| node-b | 10.20.2.10 |

## シナリオに基づく IP リソースの IP アドレス

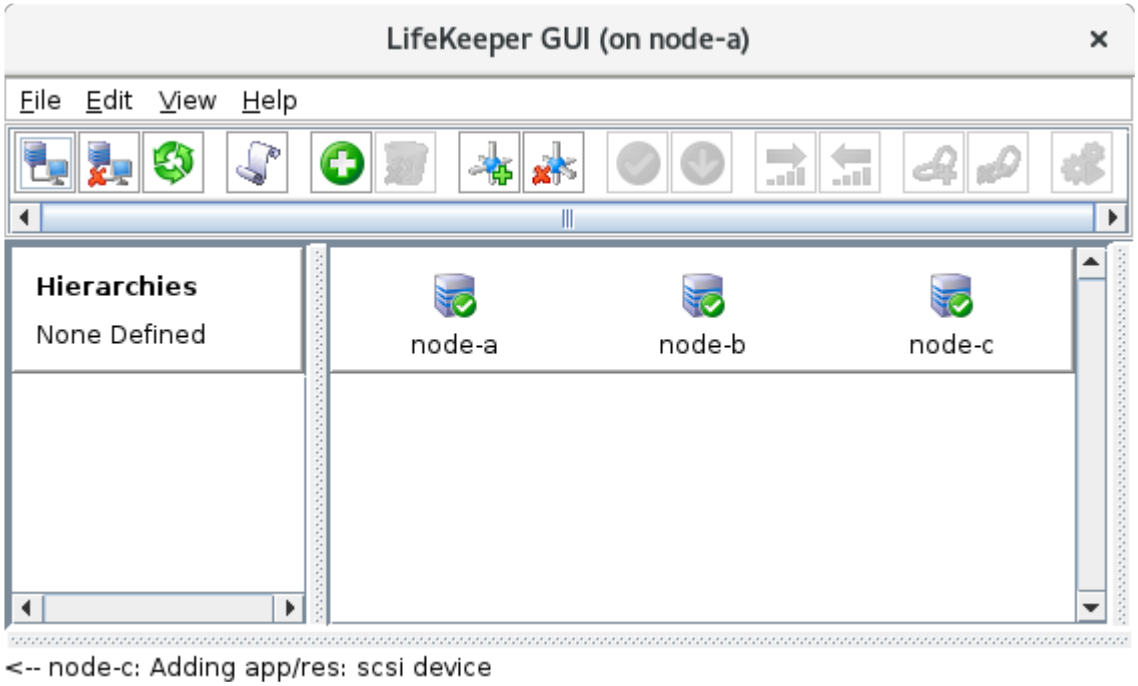
クラウド環境では、IP アドレスを選択する前に クラウド環境でのノードの切り替え を確認し、最も適用可能なシナリオを確認してください。

<table>
<tr><th>プラットフォーム</th><th>シナリオ</th><th>IPアドレスの選択方法</th><th>このガイドで使用されているサンプルIPアドレス</th><th>IPアドレスのデフォルト名</th></tr>
<tr><td rowspan="3">AWS</td><td>ルートテーブルシナリオ</td><td>VPC CIDR の外部にある IP アドレス</td><td>10.10.10.10</td><td>ip-10.10.10.10</td></tr>
<tr><td>Route 53 シナリオ</td><td>「アクティブ」ノードに割り当てられているIPアドレス</td><td>10.20.1.10</td><td>realip</td></tr>
<tr><td>Elastic IP シナリオ</td><td colspan="3" rowspan="2">IPリソースを作成する必要はありません。このページはスキップしてください</td></tr>
<tr><td colspan="2">Google Cloud</td></tr>
<tr><td colspan="2">Azure</td><td rowspan="2">サブネットで使用可能な IP アドレス</td><td rowspan="2">10.20.0.10</td><td rowspan="2">ip-10.20.0.10</td></tr>
<tr><td colspan="2">オンプレミス環境</td></tr>
</table>

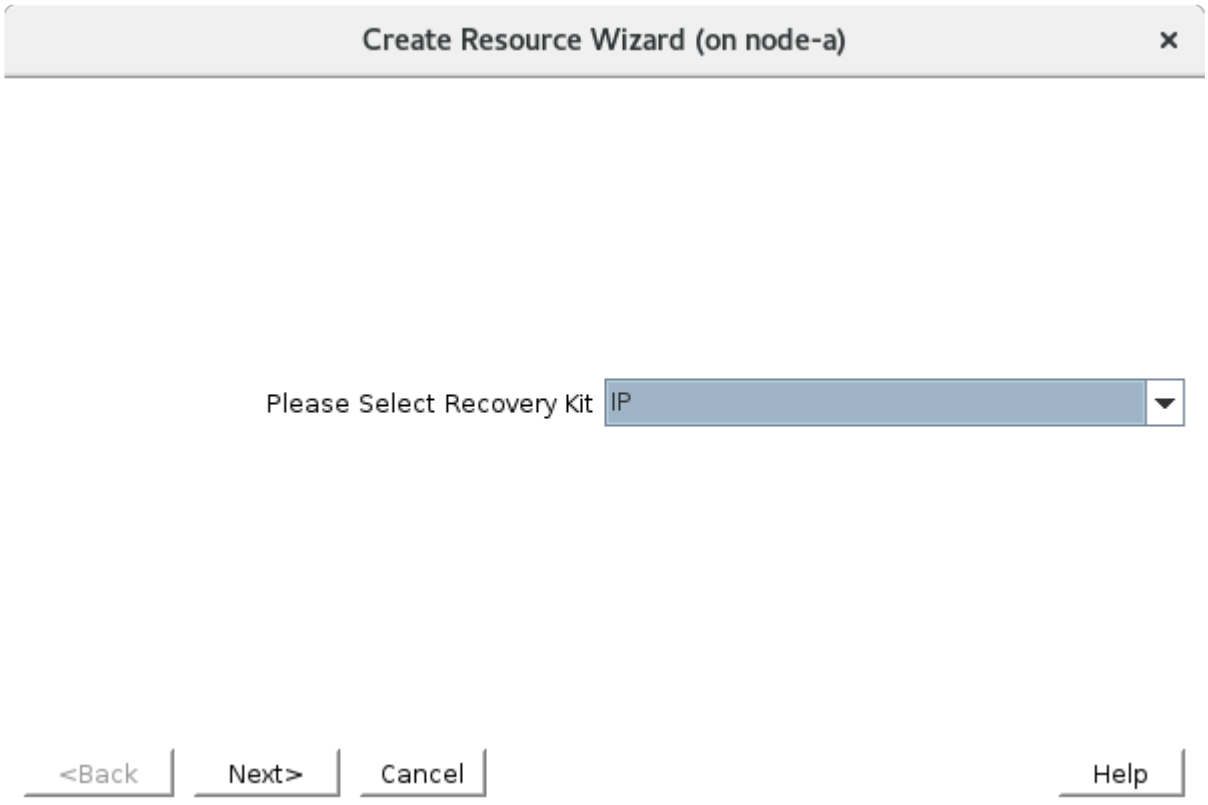
## IP リソースの設定

この例では、AWS の「ルートテーブルシナリオ」で使用する IP アドレスとして 10.10.10.10 を使用しています。

1.LifeKeeper ユーザーインターフェースの ボタンをクリックします（またはEdit > Server > Create Resource Hierarchyを選択します）。




2. リソースの作成ウィザード（node-a 上）が表示され、保護するリソースのタイプを選択できます。「IP」を選択し、「Next >」をクリックします。




3. デフォルト値を選択します。入力が必要なフィールドは、「IP Resource」フィールドの仮想 IP アドレス10.10.10.10 のみです。

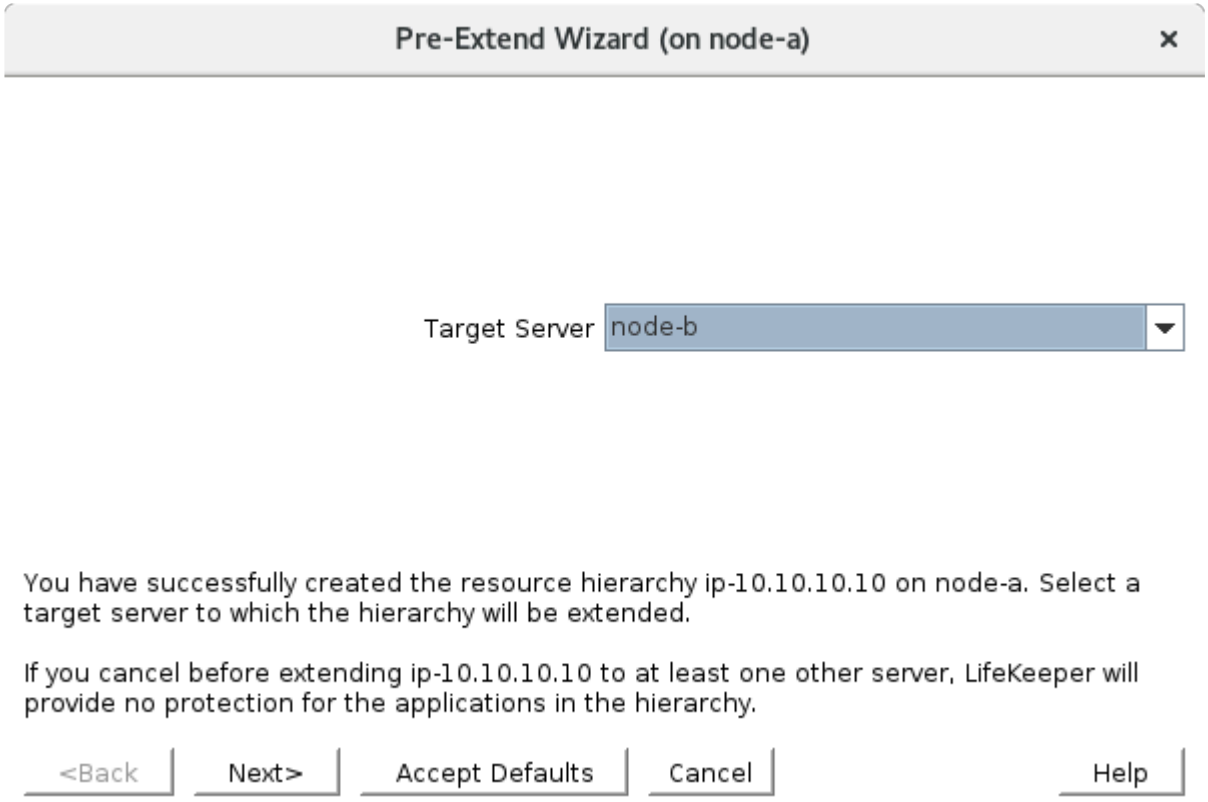
**注記:** チェックマーク ( ✔ )が付いている項目については、デフォルト値を確認の上使用してください。

| フィールド | 値 |
|---|---|
| Switchback Type | Intelligent |
| Server | node-a |
| IP Resource | 10.10.10.10 |
| Net Mask | 255.255.255.0<br>注記：この値は、ネットワークによって異なる場合があります。 |
| Network Interface | eth0 |
| IP Resource Tag | ip-10.10.10.10 |

ネットマスクなどの値は、ローカルシステムの設定方法に基づいて表示されます。そのため、表示されるデフォルト値は異なる場合があります。

4. リソースが正常に作成されたことを確認したら、「Next >」をクリックします。

5. 次のステップは「Pre-Extend Wizard (on node-a)」です。



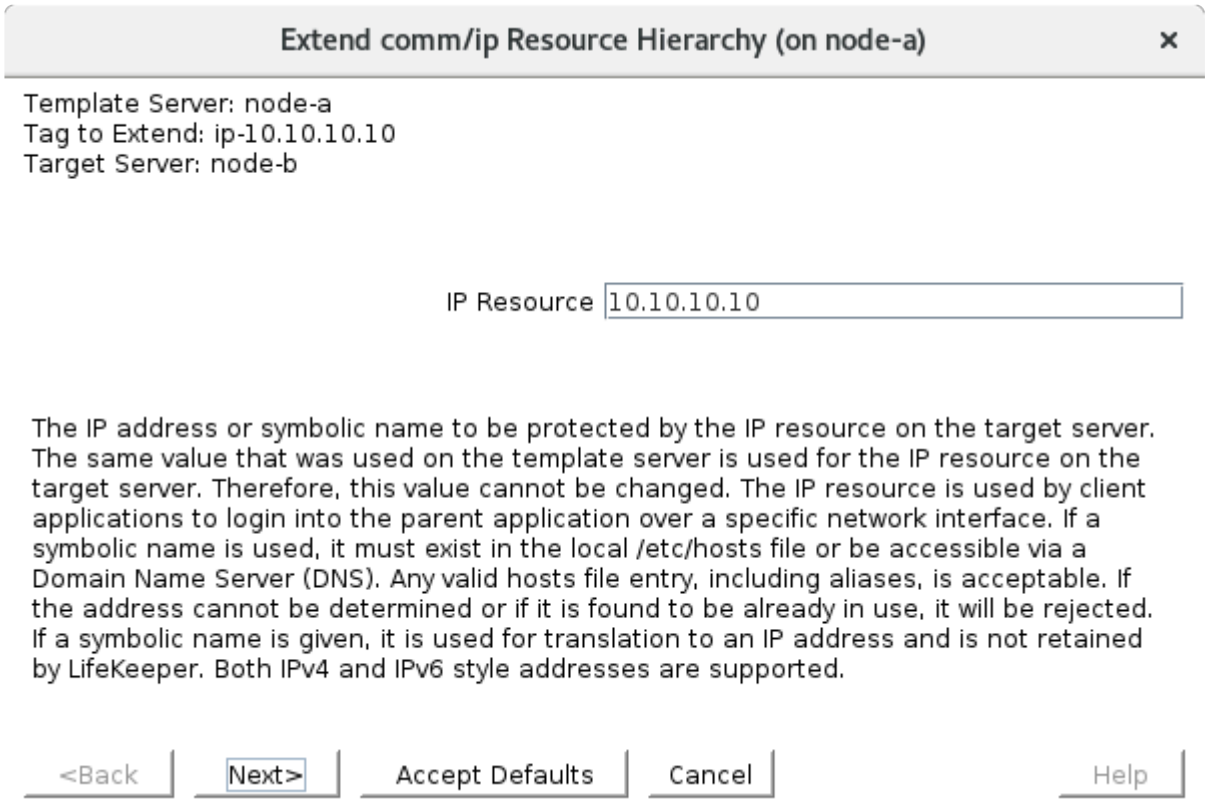

以下のように、デフォルトをすべて選択します。

| フィールド | 値 |
|---|---|

| | |
|---|---|
| Target Server | node-b |
| Switchback Type | Intelligent |
| Template Priority | 1 |
| Target Priority | 10 |

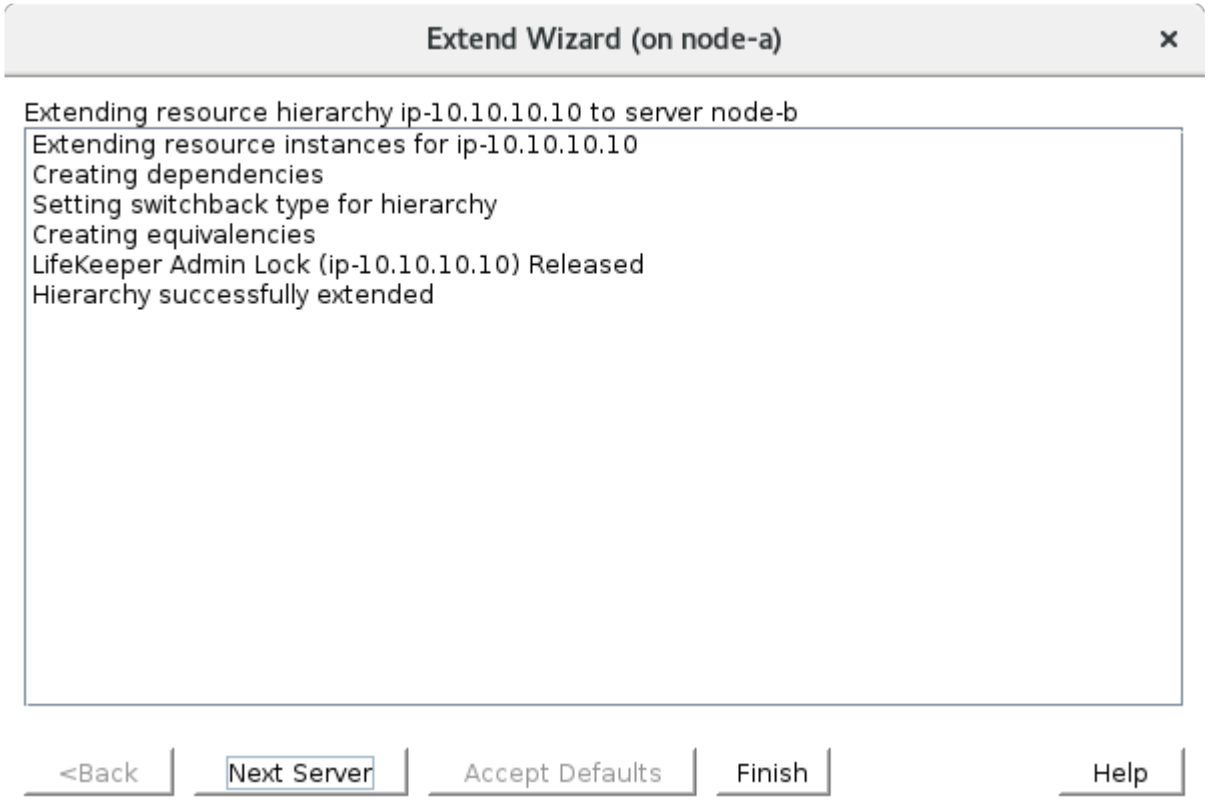
Pre-Extend のチェックは正常に終了しているはずです。「Next」をクリックします。

6. 次は、「Extend comm/IP Resource Hierarchy (on node-a) です。「Next」をクリックし、各項目を入力します。



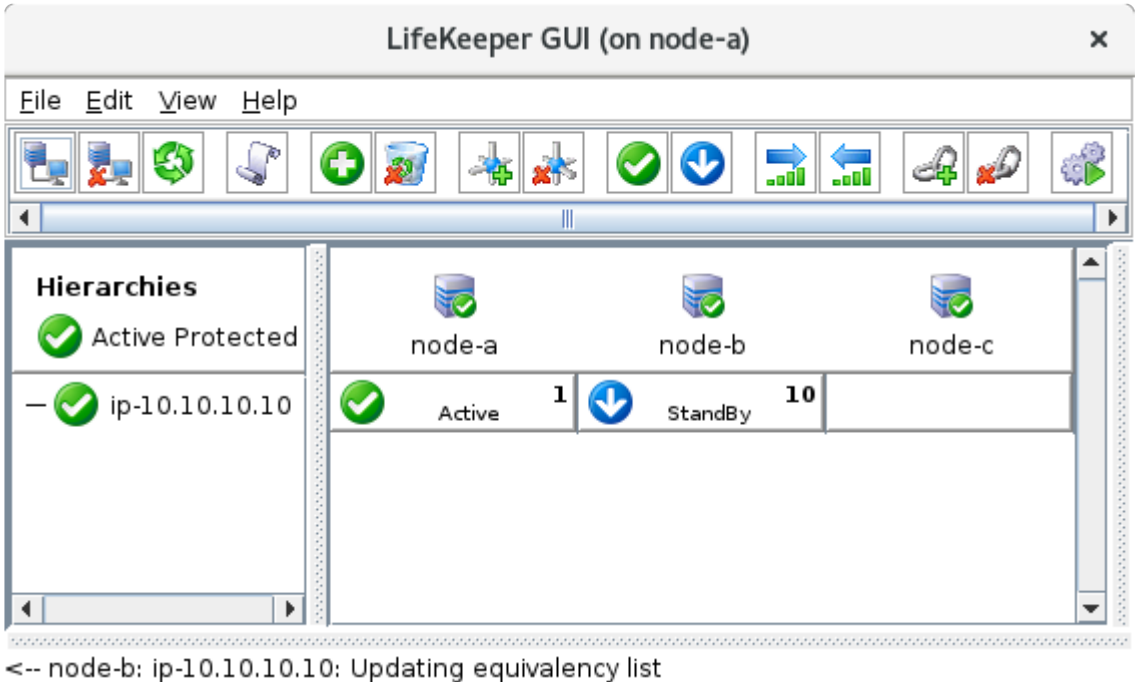

| フィールド | 値 |
|---|---|
| Net Mask | 255.255.255.0<br>注記: この値は、ネットワークによって異なる場合があります。 |
| Network Interface | eth0 |
| IP Resource Tag | ip-10.10.10.10 |

7. これで、リソースが拡張されました。「Finish」を選択します。




8. これで検証が完了し、ウィザードを閉じることができます。

9. IPリソース（ip-10.10.10.10）が LifeKeeper ユーザーインターフェースに表示されます。




<-- node-b: ip-10.10.10.10: Updating equivalency list

これまで見てきたように、リソース作成ウィザードの質問に答えることで、リソースを簡単に保護できます。ほとんどの値は、環境の理解に基づいて事前入力されているため、ユーザーによる操作は最小限で済みます。

# 11.2.7.2. クラウド環境でのノードの切り替え

クラウド環境で運用している場合、ノード間で仮想IPアドレスを切り替えるための手順が追加されます。以下のトピックでは、それぞれのクラウド環境における背景情報と一般的なアプローチについて説明します。

- AWS ルートテーブルシナリオ
- AWS Elastic IP シナリオ
- AWS Route53 シナリオ
- Azure 内部ロードバランサーシナリオ
- Google Cloud 内部ロードバランサーシナリオ

次のセクションでは、各シナリオで必要な設定手順について説明します。

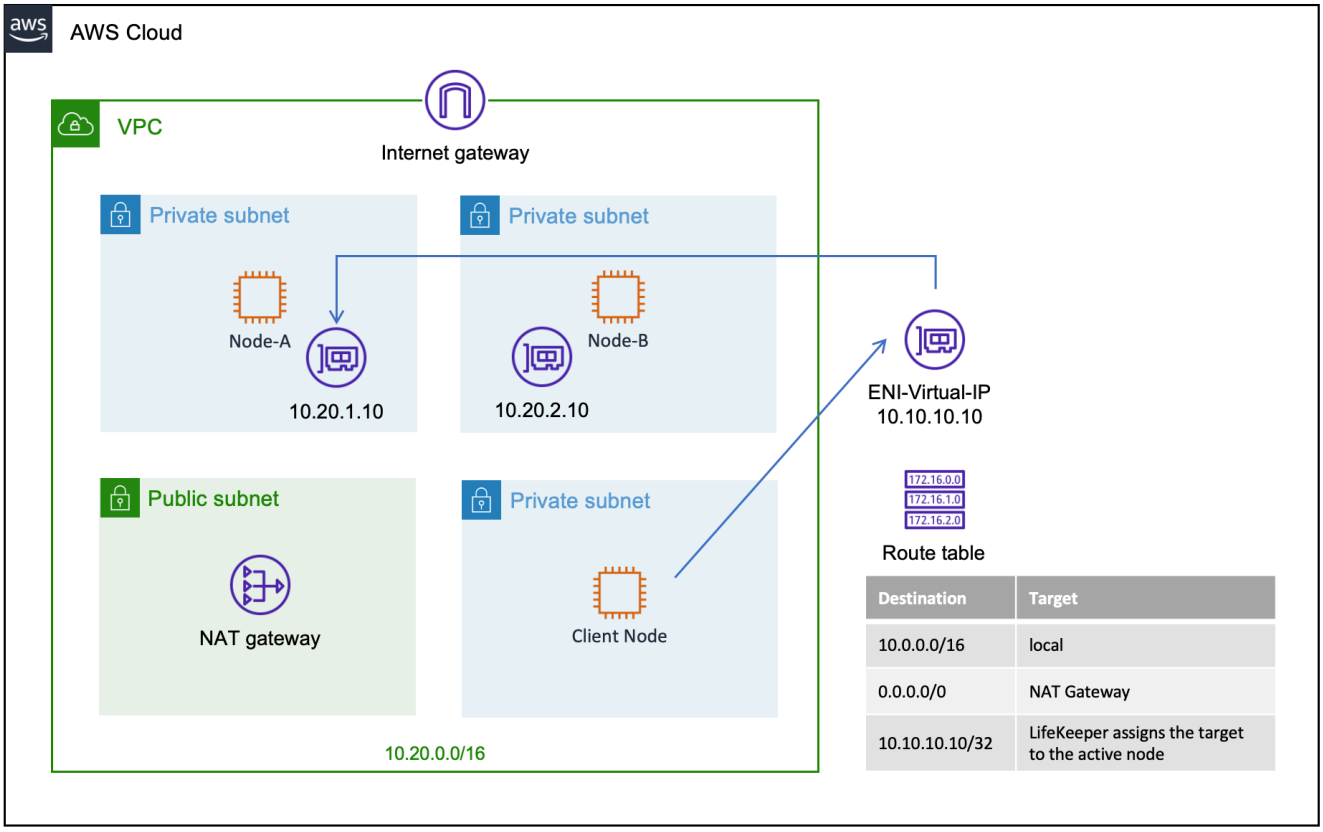
- AWS EC2 リソースの作成 (ルートテーブルシナリオ)
- AWS EC2 リソースの作成 (Elastic IP シナリオ)
- AWS Route53 リソースの作成
- Azure – 内部ロードバランサーの使用
- Google Cloud – 内部ロードバランサーの使用

SIOS Protection Suite は、トラフィックを現在のアプリケーションホストにルーティングするために、内部ロードバランサーからの TCP ヘルスチェックプローブに応答するリソースタイプも提供します。

- ロードバランサーのヘルスチェックに応答する

# 11.2.7.2.1. AWS EC2リソースの作成（ルートテーブルシナリオ）

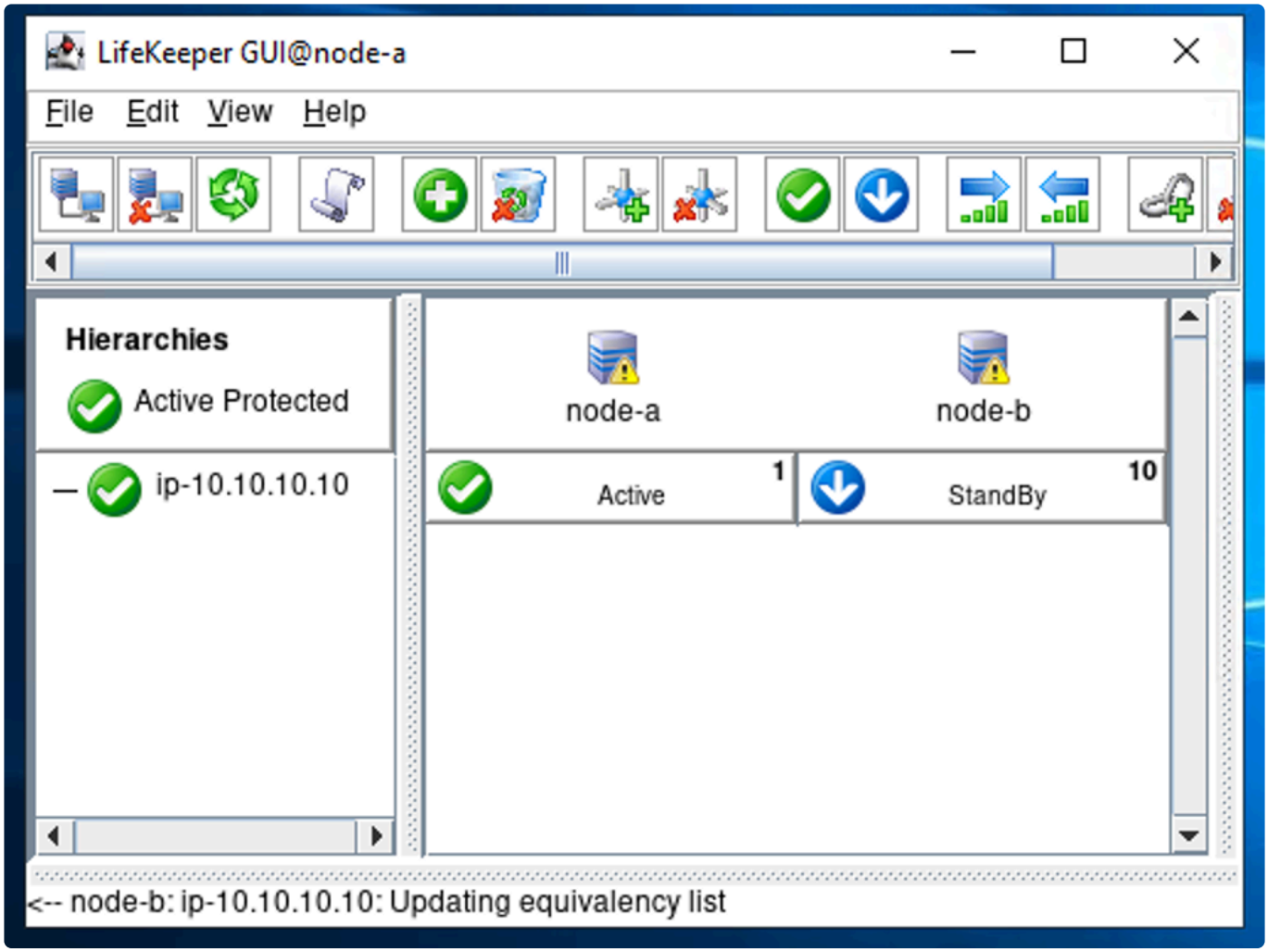
ここでは、アクティブノードを指すエントリをルートテーブルに作成することにより、ルートテーブルを使用してノードを切り替える方法について説明します。




## IP リソースを作成する

これで、IP リソースを作成できます。リソースの IP アドレスは VPC の外部に配置する必要があることに注意してください。この例では、 `10.10.10.10` を使用して IP リソースを作成します（VPC は 10.20.0.0/16 として定義されています）。詳細については、 IP リソースの作成 を参照してください。

IP リソースが作成されると、LifeKeeper ユーザーインターフェースは以下のようになります。




## ルートテーブルのエントリーを更新する権限をインスタンスに割り当てる

> ! 以下の手順を完了するには、AWS CLIを各ノードにインストールする必要があります。

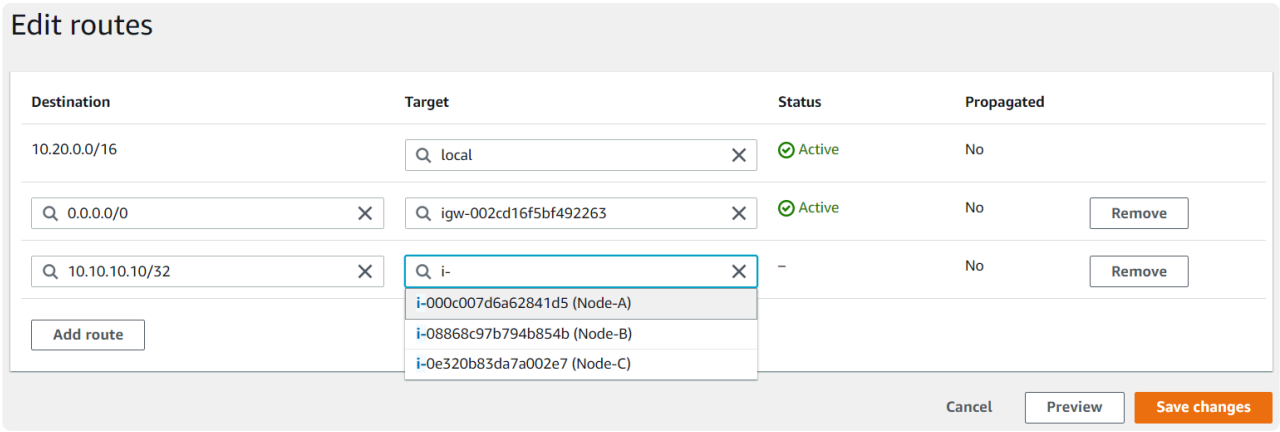
- AWS CLI をインストールする
- EC2 Recovery Kit を使用するための権限を割り当てる

## プライマリーホストの最初のルートテーブルエントリーを追加する

EC2 リソースを作成する前に、まず LK-VPC のルートテーブル（LK-RouteTable）に最初のルートを追加する必要があります。

1. AWS コンソールで、[Services] → [VPC] → [Route Tables] に移動し、LK-VPC に関連付けられているルートテーブル（LK-RouteTable）をクリックし、[Routes] タブをクリックして、次に [EditRoutes] をクリックします。
2. 表示されたダイアログで、以下のパラメーターを指定して新しいルートを追加します。

| フィールド | 値 |
|---|---|
| Destination | 10.10.10.10/32 |
| Target | 「Instance」を選択し、Node-A のインスタンス ID を選択します。 |




3. **[Save changes]** をクリックして、ルートをルートテーブルに追加します。

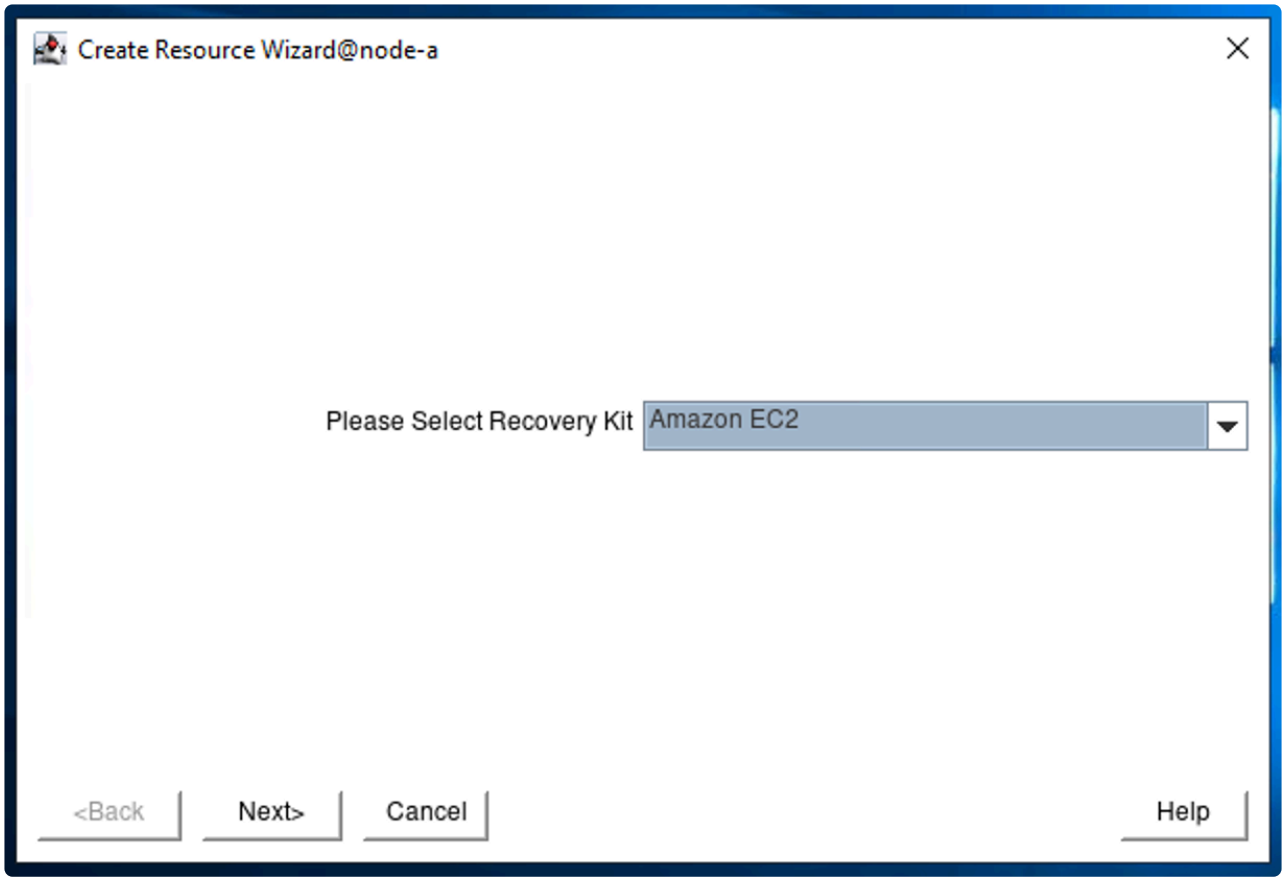
## EC2 リソースの作成（ルートテーブルシナリオ）

> ! EC2 リソースを作成する前に、 AWS – 送信元/宛先チェックを無効にする の手順に従って、すべてのクラスターノードで送信元/宛先チェックが無効になっていることを確認してください。

IPリソースが作成されていれば、EC2リソースの作成は簡単です。

1. アイコンを選択して、「Create Resource Wizard」を開始します。

2. Recovery Kitは「Amazon EC2」を選択します。




3. 「Create Resource Wizard @ node-a」で、以下の値を指定します。

> **注記:** チェックマーク ( ) が付いている項目については、デフォルト値を確認し、推奨の値を使用してください。

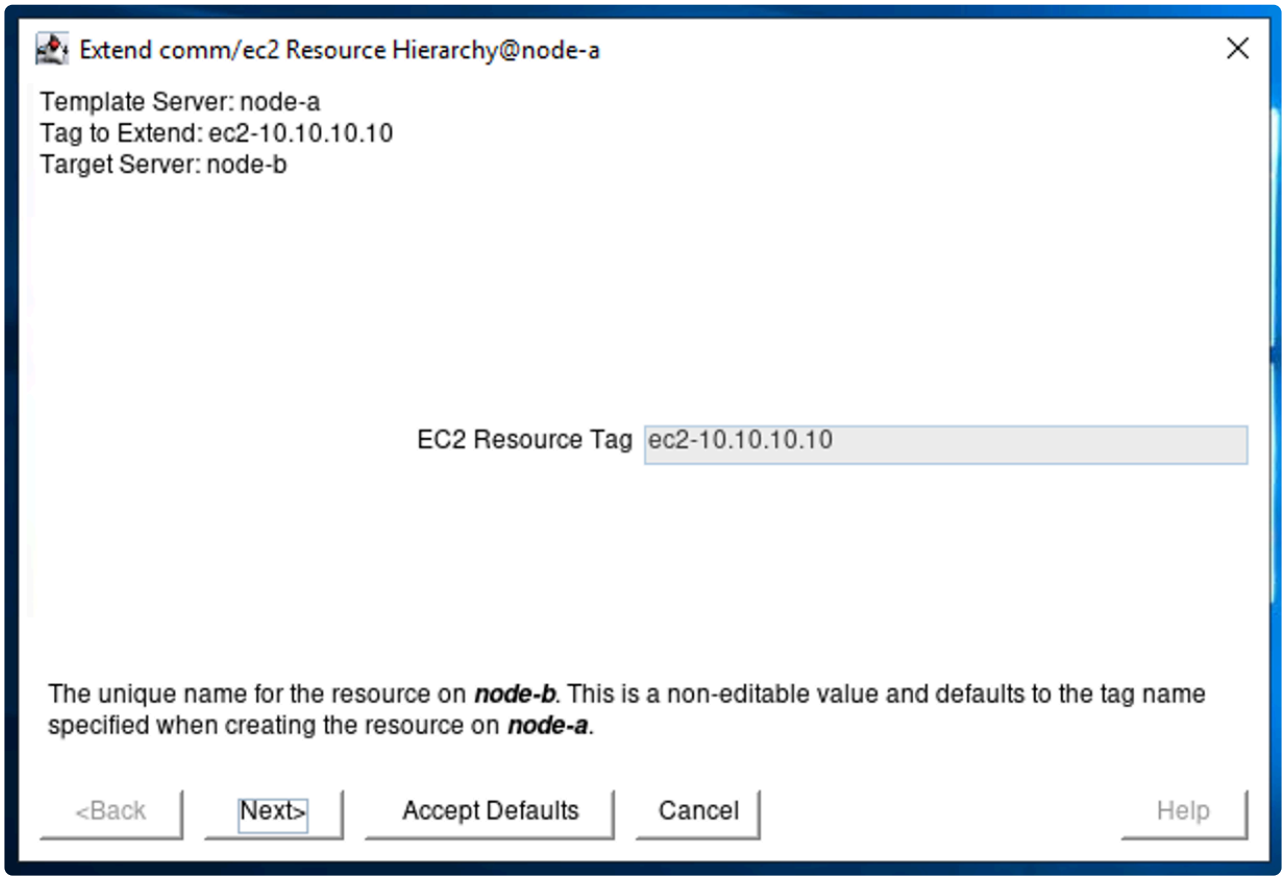
| フィールド | 値 |
|---|---|
| Switchback Type | Intelligent |
| Server | node-a |
| EC2 Resource Type | Route Table (Backend Cluster) |
| IP Resource | ip-10.10.10.10 |
| EC2 Resource Tag | ec2-10.10.10.10 |

4. 「Pre-Extend Wizard」で、以下の値を指定します。

| Field | Value |
|---|---|
| Target Server | node-b |
| Switchback Type | Intelligent |

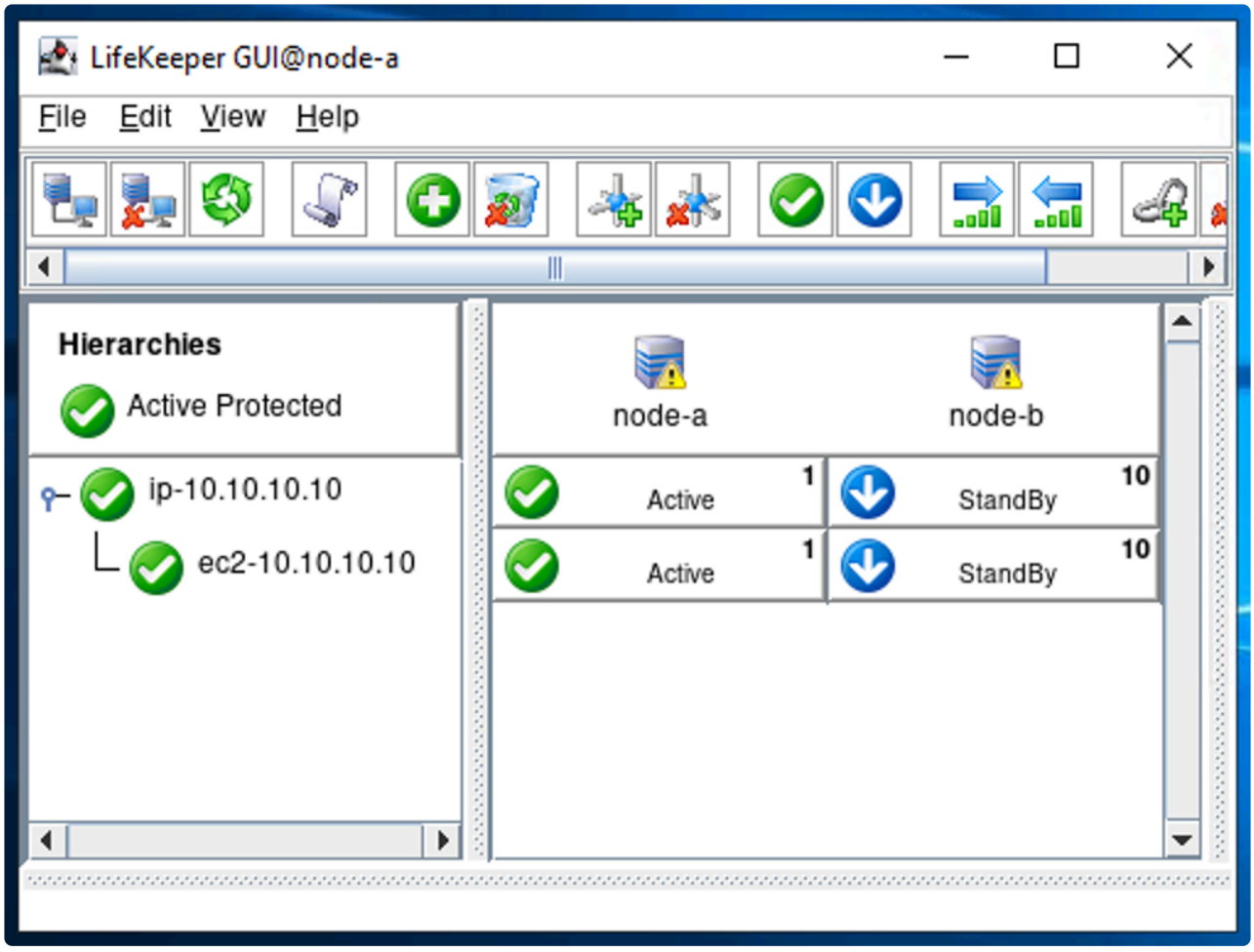
| | |
|---|---|
| Template Priority | 1 |
| Target Priority | 10 |

5. 事前拡張が完了したら、「Extend comm/ec2 Resource」に進みます。「ec2-10.10.10.10」を選択します。



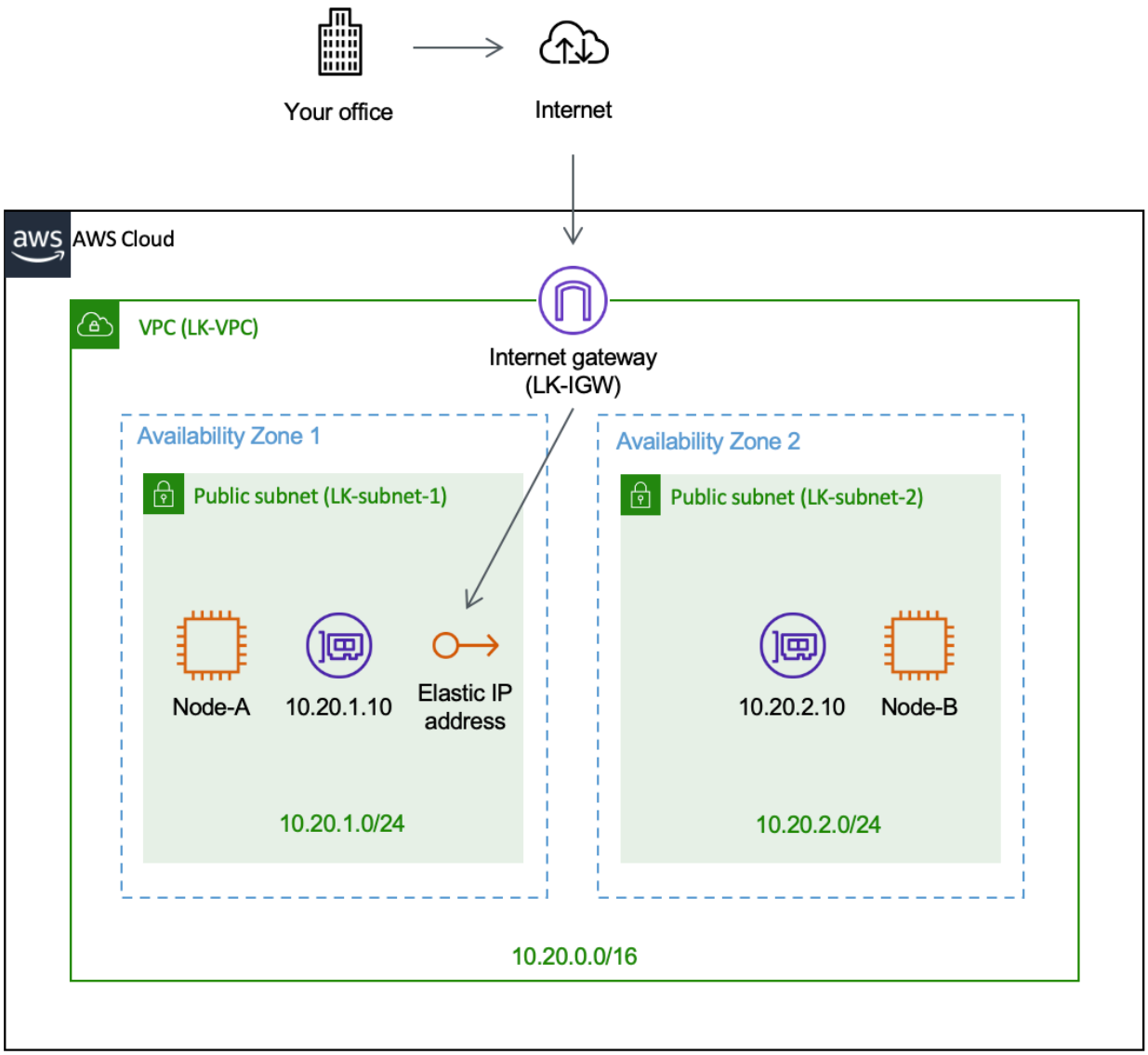

6. 「Hierarchy successfully extended」と表示されたら、リソースの作成は完了です。

7. 以下のように EC2 リソースが作成されます。


LifeKeeper GUI@node-a
File Edit View Help
Hierarchies
Active Protected
ip-10.10.10.10
ec2-10.10.10.10
node-a
node-b
Active 1
StandBy 10
Active 1
StandBy 10

# 11.2.7.2.2. AWS EC2 リソースの作成（Elastic IP シナリオ）

ここでは、アクティブノードを指すエントリーを作成することで、Elastic IP アドレスを使用してノードを切り替える方法について説明します。




## ルートテーブルのエントリーを更新する権限をインスタンスに割り当てる

! 以下の手順を完了するには、AWS CLI を各ノードにインストールする必要があります。

- AWS CLI をインストールする

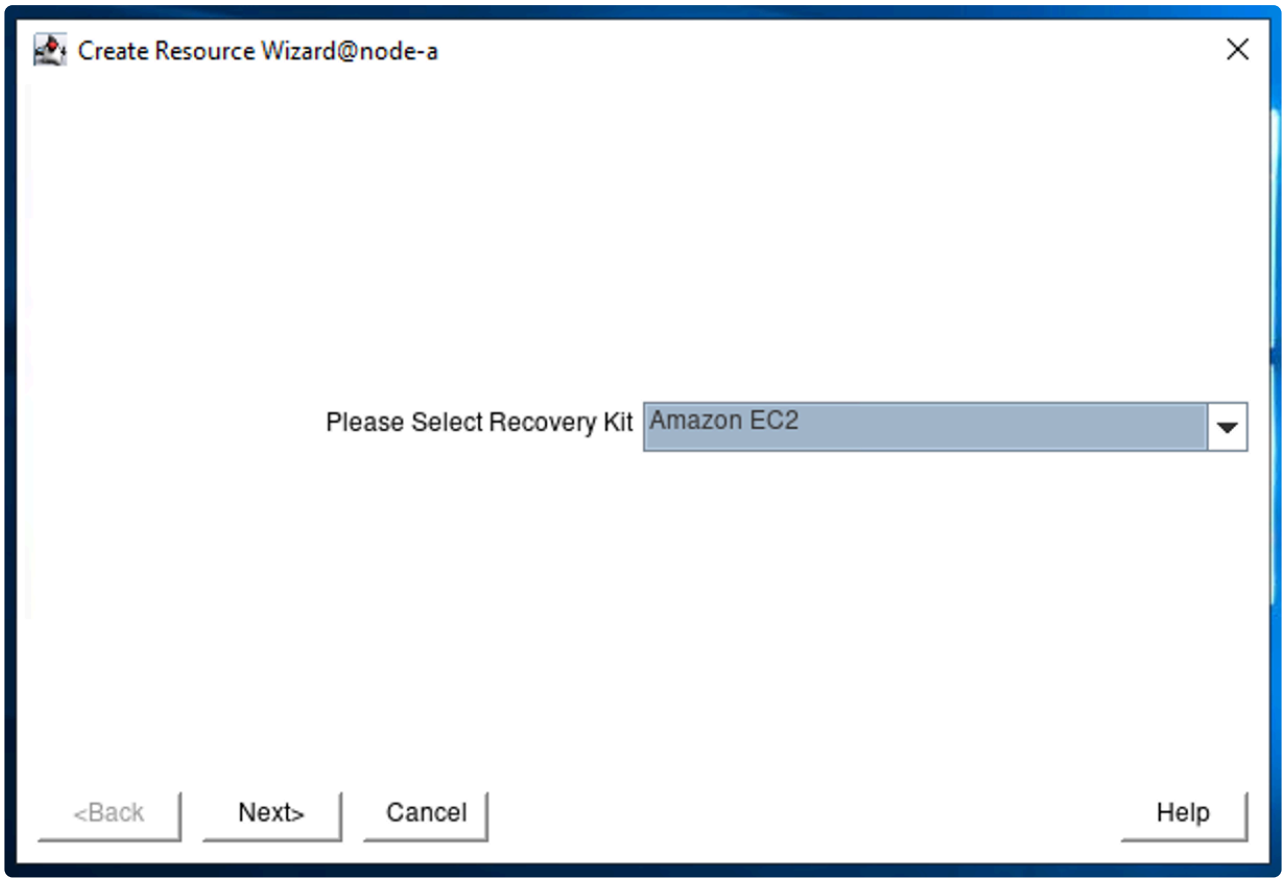
- EC2 Recovery Kit を使用するための権限を割り当てる

また、Elastic IP を取得する必要があります。以下の例では、Elastic IP アドレスは 50.18.115.213 です。なお、この IP アドレスは一例です。

## EC2 リソースの作成（Elastic IP シナリオ）

Elastic IP（フロントエンドクラスター）シナリオで EC2 リソースを作成するには、次の手順を実行します。

1. Recovery Kitに「Amazon EC2」を選択します。




2. 「Create Resource Wizard @ node-a」で、以下の値を指定します。

> **注記:** チェックマーク（✓）が付いている項目については、デフォルト値を確認し、推奨の値を使用してください。

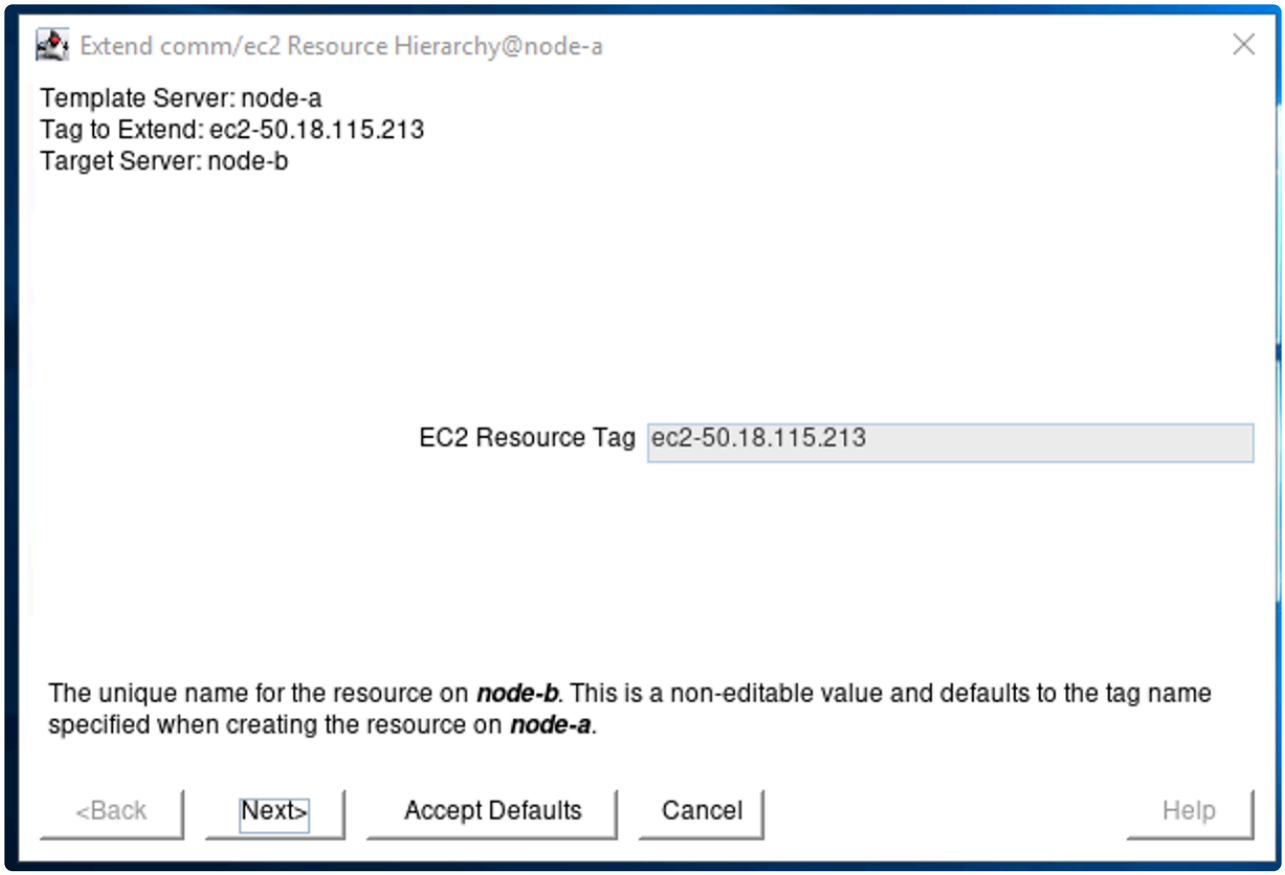
| フィールド | 値 |
|---|---|
| Switchback Type | Intelligent ✓ |
| Server | node-a ✓ |

| | |
|---|---|
| EC2 Resource Type | Elastic IP (Frontend Cluster) |
| Network Interface | eith0 |
| Elastic IP | リストから利用可能なElastic IPを選択します。<br>例 50.18.115.213 |
| EC2 Resource Tag | ec2-50.18.115.213 |

3. 「Pre-Extend Wizard」で、次の値を指定します。

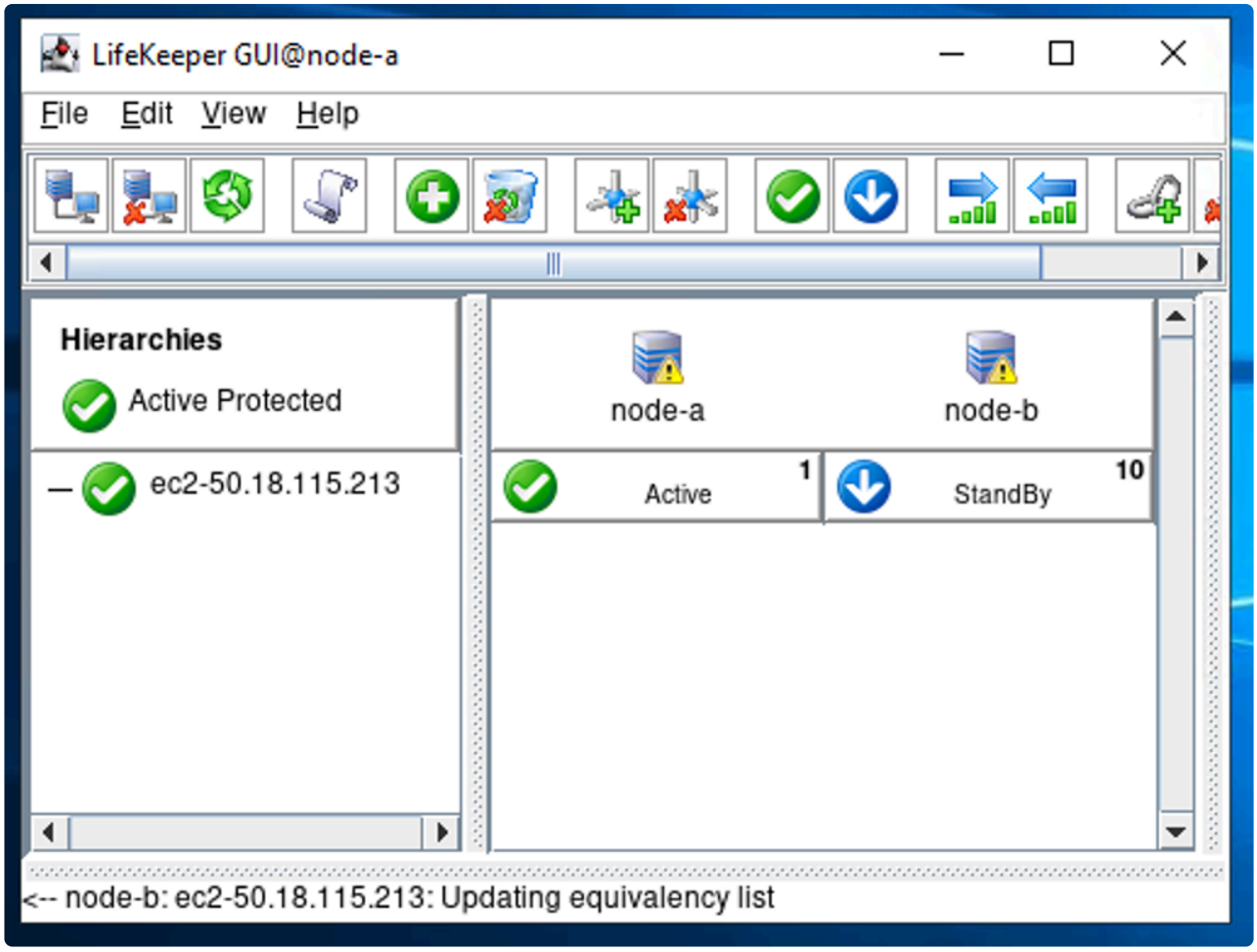
| Field | Value |
|---|---|
| Target Server | node-b |
| Switchback Type | Intelligent |
| Template Priority | 1 |
| Target Priority | 10 |

4. 事前拡張が完了したら、「Extend comm/ec2 Resource」に進みます。「ec2-50.18.115.213」を選択します。



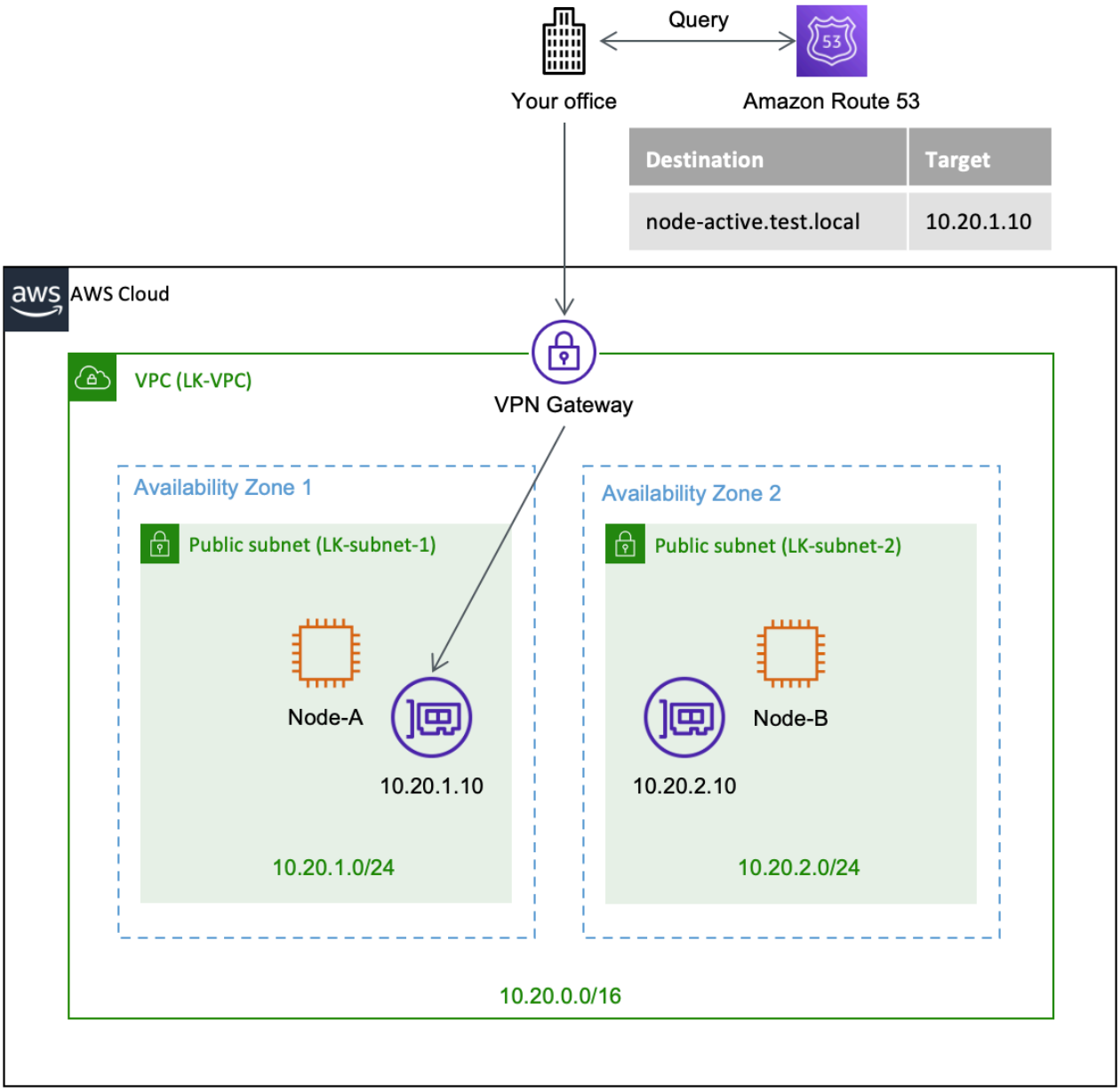
5. 「Hierarchy successfully extended」と表示されたら、リソースの作成は完了です。

6. 以下のようにEC2リソースが作成されます。



# 11.2.7.2.3. AWS Route 53リソースの作成

ここでは、アクティブノードを指す DNS エントリーを作成して、Route 53 を使用してノードを切り替える方法について説明します。




## 作成するRoute 53のレコードセット

| 名前 | タイプ | 値 |
|---|---|---|
| node-a | A-IPv4 Address | 10.20.1.10 |
| node-b | A-IPv4 Address | 10.20.2.10 |
| node-active | A-IPv4 Address | 10.20.1.10 (node-a と同じ) |

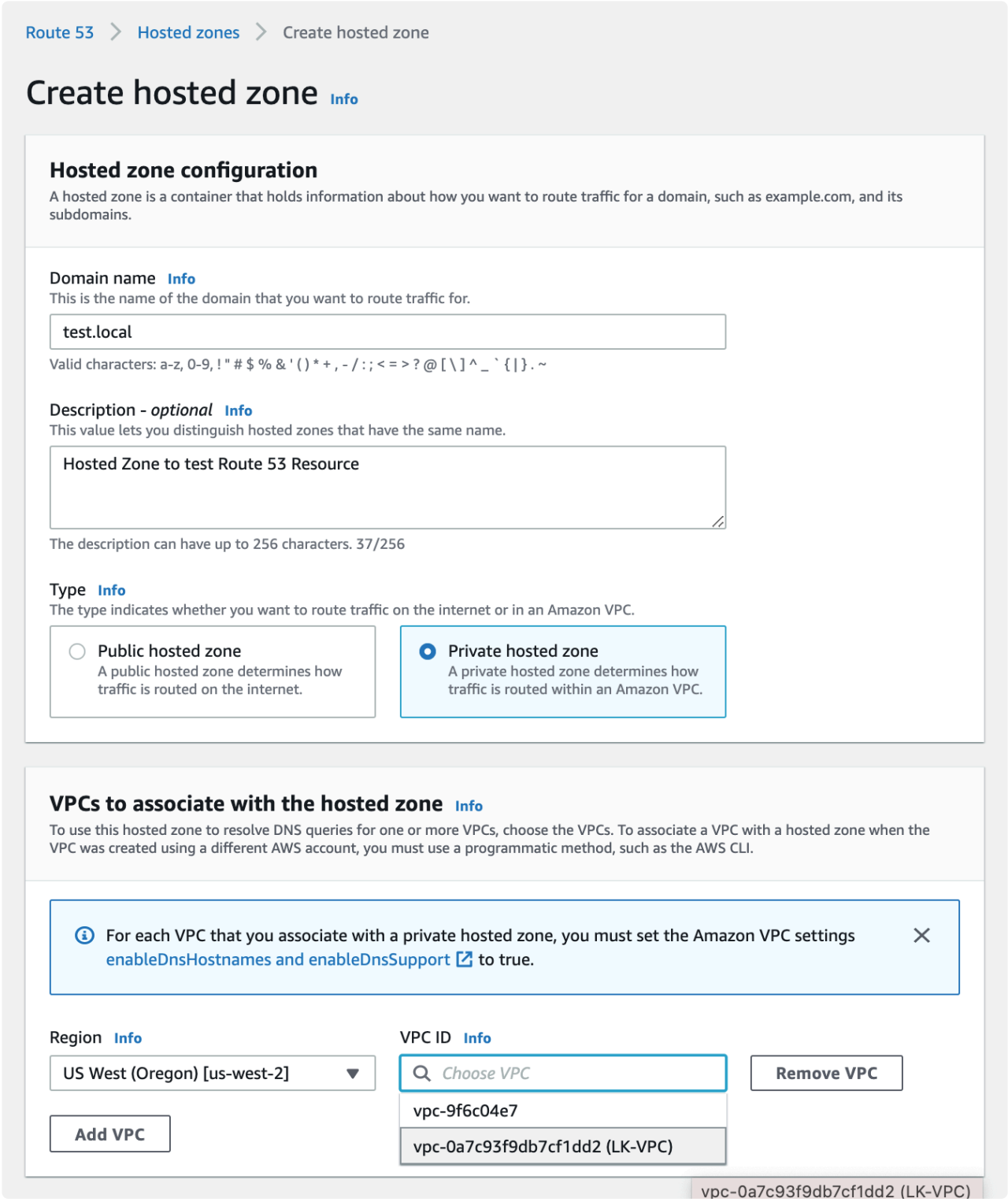
# ホストゾーンを作成する

まず、ホストゾーンが必要です（これはexample.comなど、DNSドメインのようなものです）。

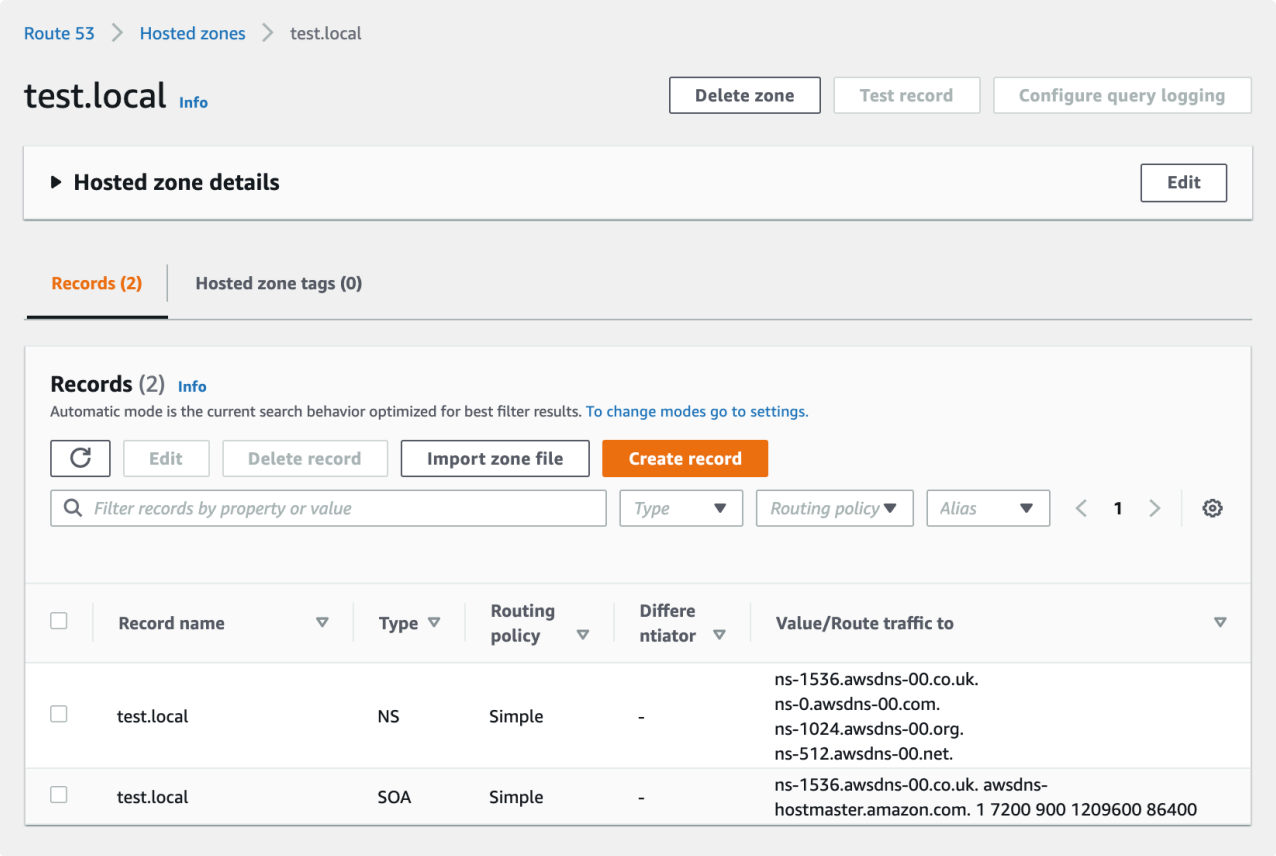
すでにホストゾーンがある場合は、このセクションをスキップして既存のホストゾーンを使用してください。

1. AWSのマネジメントコンソールで、Route 53に移動します。
2. ホストゾーン を選択します。
3. 「ホストゾーンの作成」を選択します。
4. 以下の値を入力します。

| フィールド | 値 |
|---|---|
| ドメイン名 | test.local |
| タイプ | Private hosted zone |
| VPC の関連付け | VPC のリージョンと名前を選択(例: LK-VPC) |



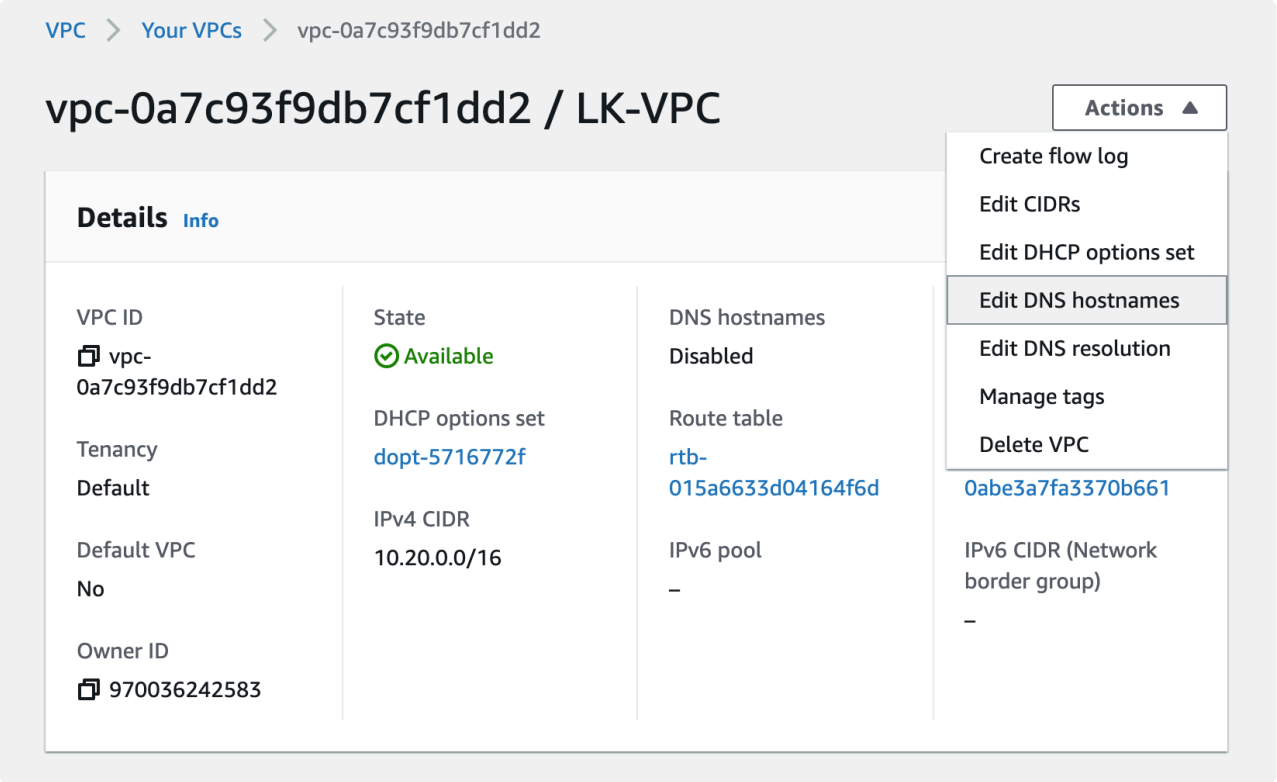

5. これでホストゾーンが作成されました。この時点では、2つのエントリー（NS と SOA）しか作成されていません。




6. 別の VPC から接続するには、他の VPC を関連付ける必要があります。

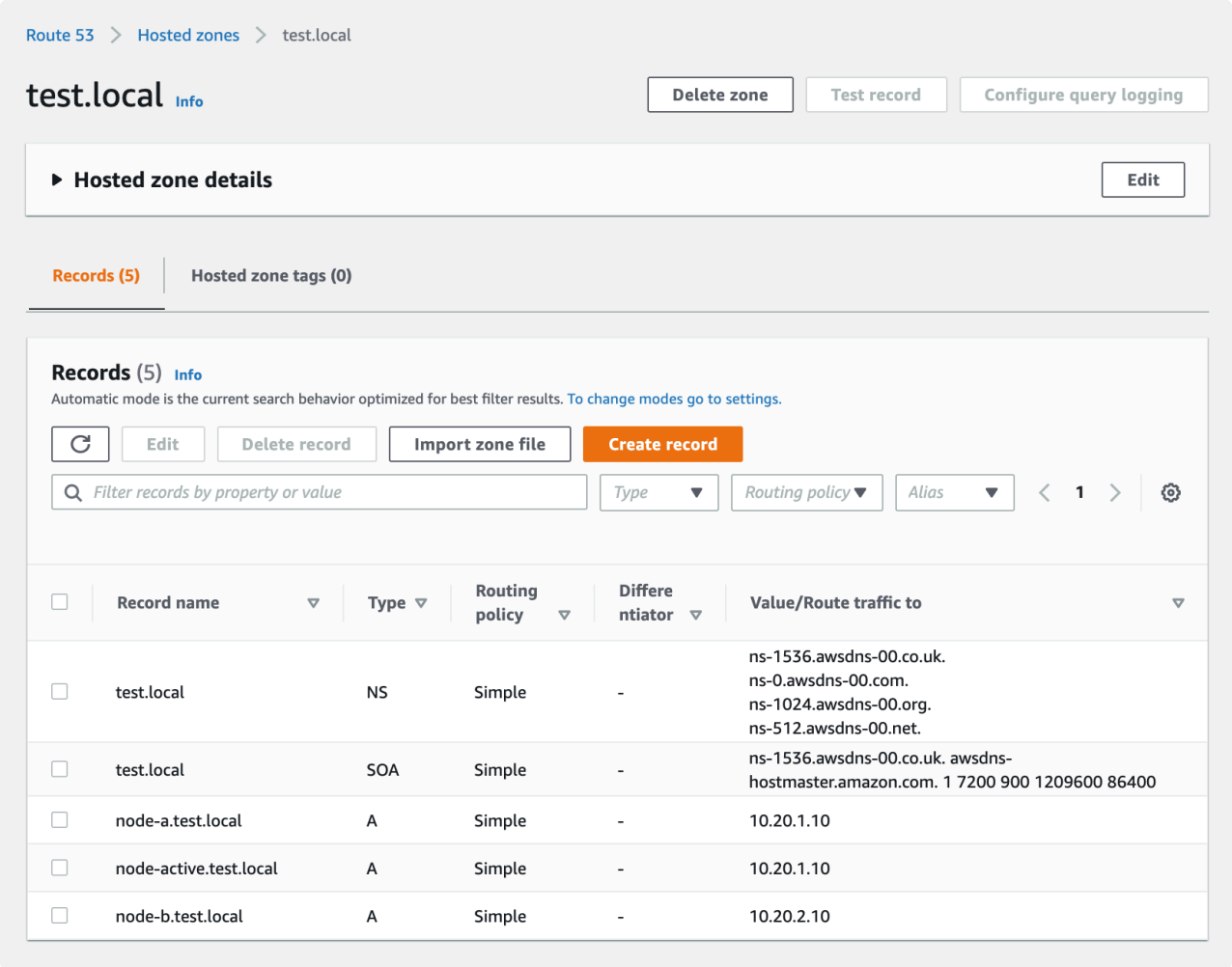
> ! VPC で DNS を使用するには、VPC 設定で DNS ホスト名と DNS 解決を有効にしてください。

VPC の設定に移動し、この2つの値が有効（enabled）になっていることを確認します。




## ホストゾーンにエントリーを作成する

1. 以下に示すようにレコードを作成します。



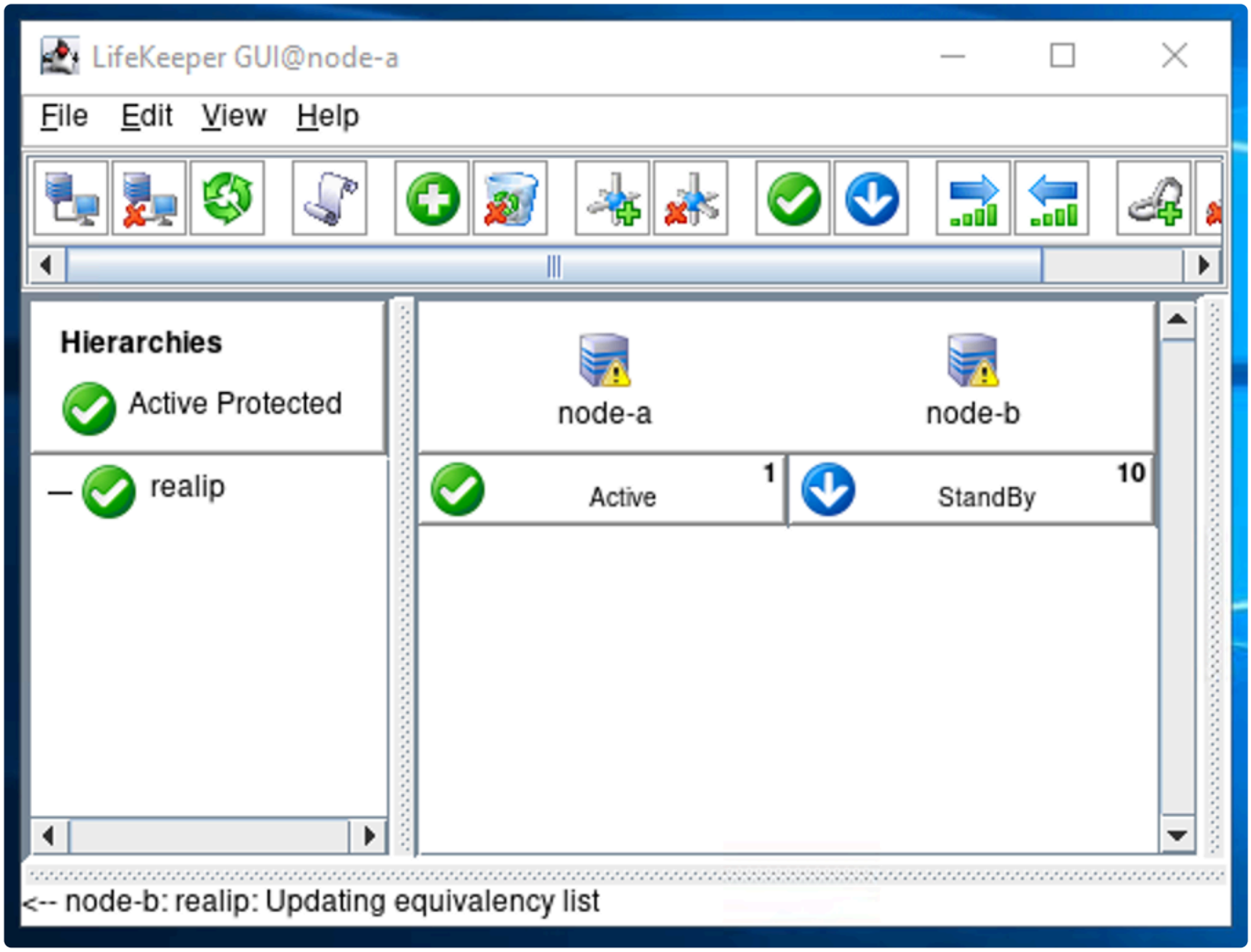

2. ホストを名前で解決できることを確認します。

```
[root@node-a ~]# ping node-a.test.cl
PING node-a.test.cl (10.20.1.10) 56(84) bytes of data.
64 bytes from node-a (10.20.1.10): icmp_seq=1 ttl=64 time=0.019 ms
64 bytes from node-a (10.20.1.10): icmp_seq=2 ttl=64 time=0.022 ms
```

# IPリソースを作成する

これで、IPリソースを作成できます。なお、リソースのIPアドレスはアクティブノードである必要があります (node-active.test.cl の IP アドレス： `10.20.1.10`). 詳細については、 IP リソースの作成 を参照してください。 アクティブノード (`10.20.1.10`) のIPアドレスが選択されると、「IP リソースタグ」フィールドが「realip」に設定され、ノードb（セカンダリー）の IP アドレスは `10.20.2.10` 2 番目のノードの実際の IP アドレス）に設定されています。これらは、1 番目のノードの「realip」idを選択することで自動的に選択されます。

IP リソースが作成されると、LifeKeeper ユーザーインターフェースは以下のように表示されます。




## インスタンスに Route 53 のエントリーを更新する権限を割り当てる

> ! 以下の手順を実施するには、AWS CLI を各ノードにインストールする必要があります。

- AWS CLI をインストールする

EC2 インスタンス上で動作する LifeKeeper が Route 53 のエントリーを更新できるようにするには、インスタンスに DNS 更新を許可するロールが必要です。

- route53:ListHostedZones
- route53:GetChange
- route53:ChangeResourceRecordSets
- route53:ListResourceRecordSets

これを実現するには、以下に示すようなポリシーを作成し（アクセスできるリソースを制限する必要がある場合があることに注意してください）、それをロールに割り当てます。

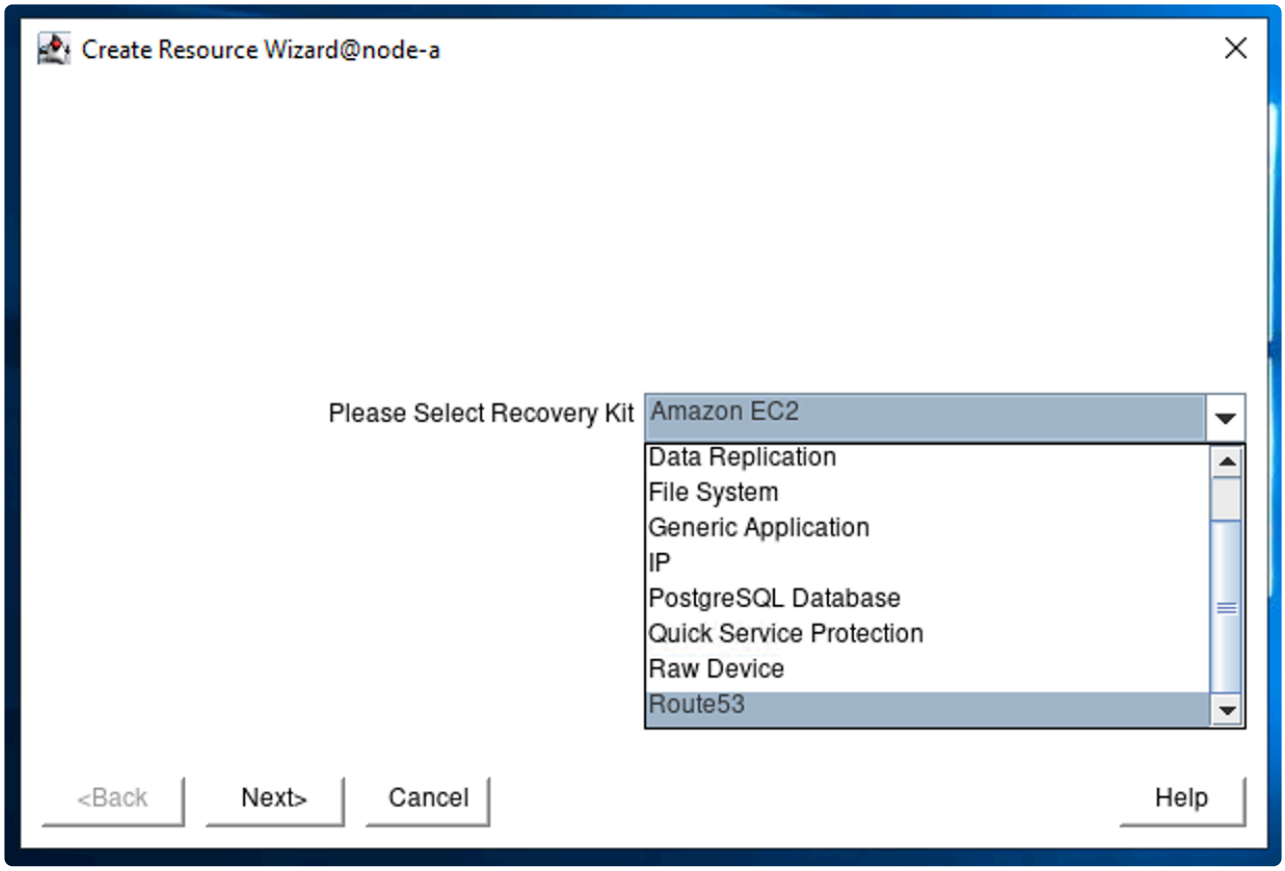
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "route53:GetChange",
                "route53:ChangeResourceRecordSets",
                "route53:ListResourceRecordSets"
            ],
            "Resource": [
                "arn:aws:route53:::hostedzone/*",
                "arn:aws:route53:::change/*"
            ]
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": "route53:ListHostedZones",
            "Resource": "*"
        }
    ]
}
```

ロールを定義したら、それを EC2 インスタンスに割り当てます。

## Route 53 リソースを作成する

Route 53リソースは、IPリソースを作成した後であれば簡単に作成できます。

1. Recovery KitにRoute 53 を選択します。




2. 「Create Resource Wizard @ node-a」で、以下の値を指定します。

> **注記:** チェックマーク ( ✓ ) が付いている項目については、デフォルト値を確認し、推奨の値を使用してください。

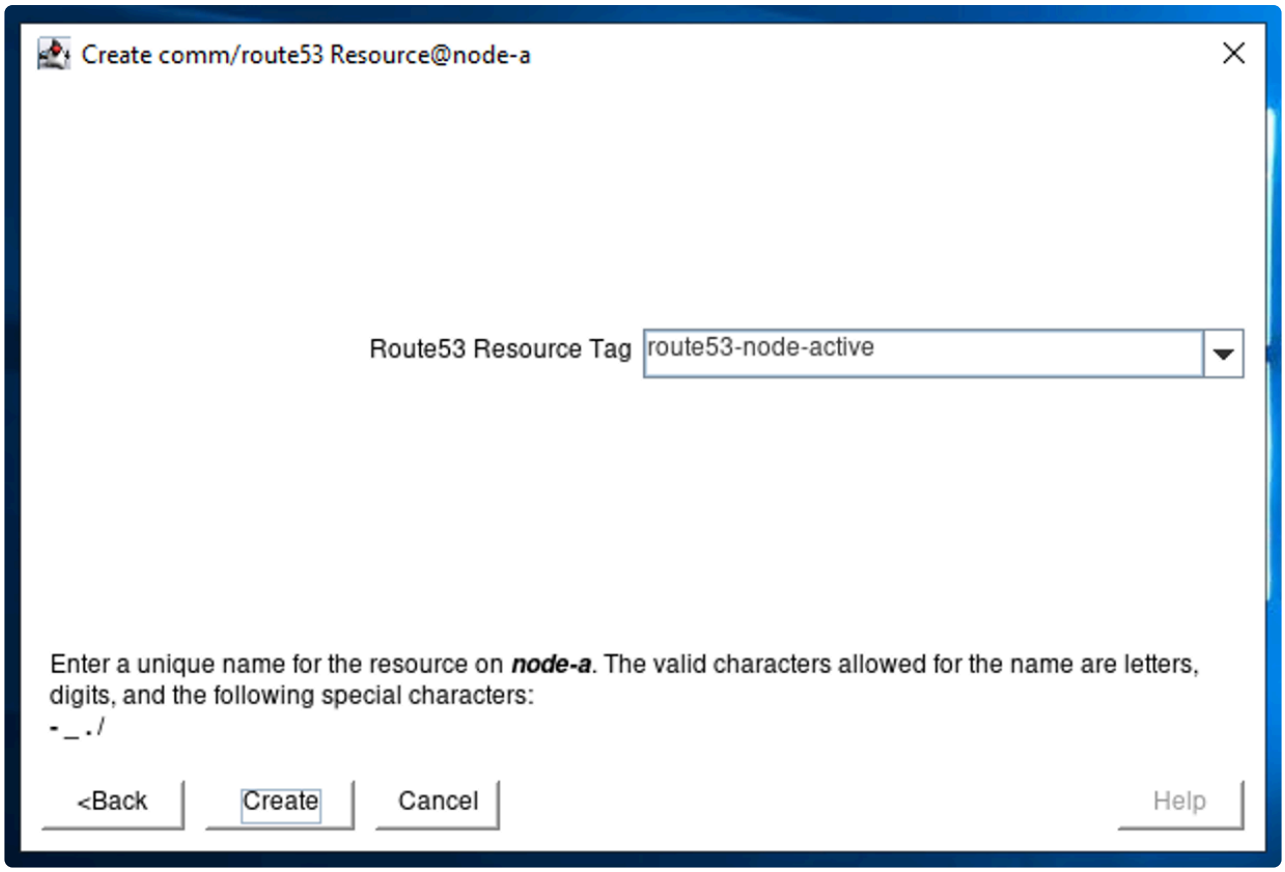
| フィールド | 値 |
|---|---|
| Switchback Type | Intelligent ✓ |
| Server | node-a ✓ |
| Domain Name (Route53 hosted zone) | test.local |
| Host Name | node-active |
| IP Resource | realip ✓ |
| Route53 Resource Tag | reoute53-node-active ✓ |

3. 「Pre-Extend Wizard」で、以下の値を指定します。

| フィールド | 値 |
|---|---|
| Target Server | node-b ✓ |

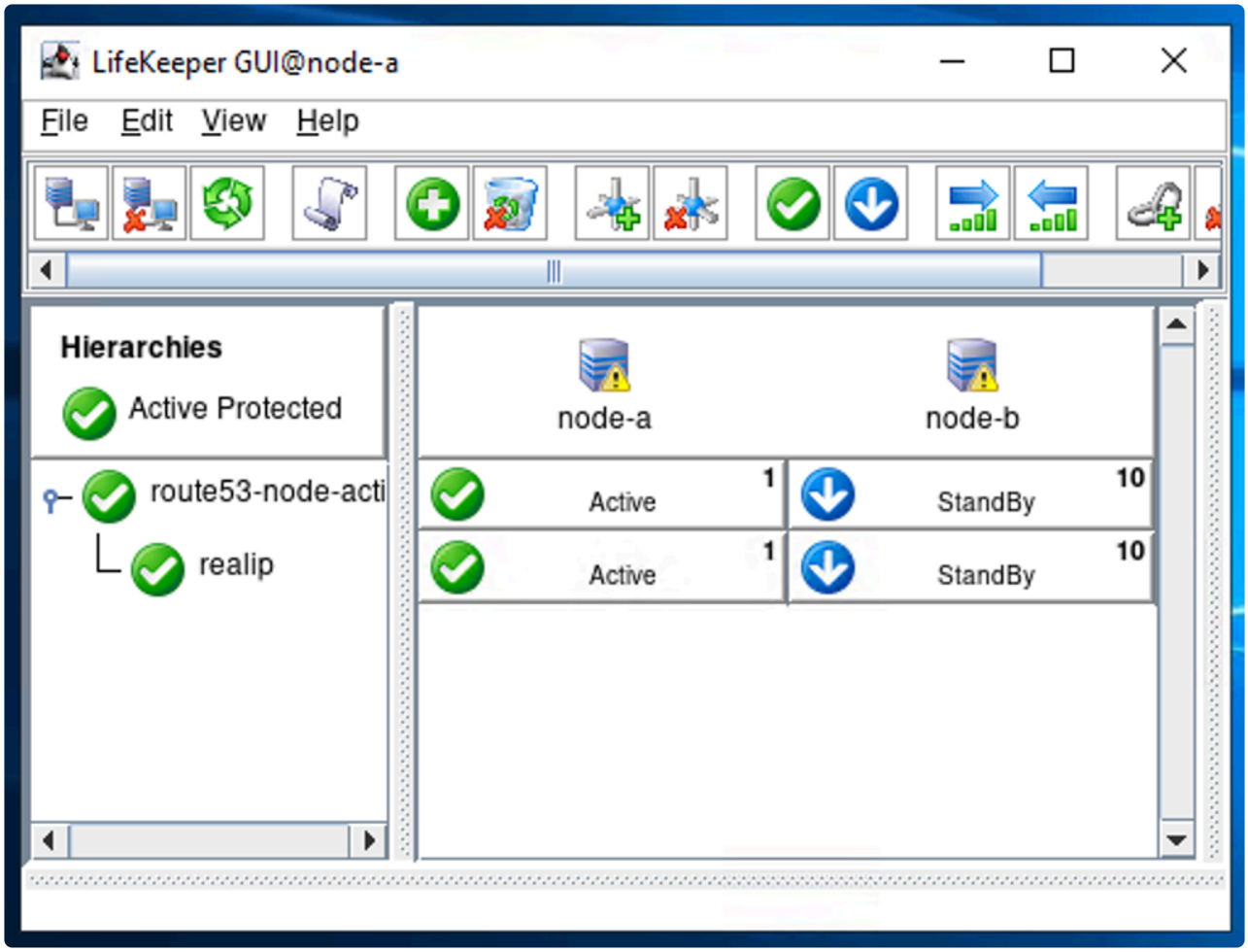
| | |
|---|---|
| Switchback Type | Intelligent |
| Template Priority | 1 |
| Target Priority | 10 |

4. Pre-Extend が完了したら、「Extend comm/route53 Resource」に進みます。「route53-node-active」を選択します。




5. 「Hierarchy successfully extended」と表示されたら、リソースの作成は完了です。

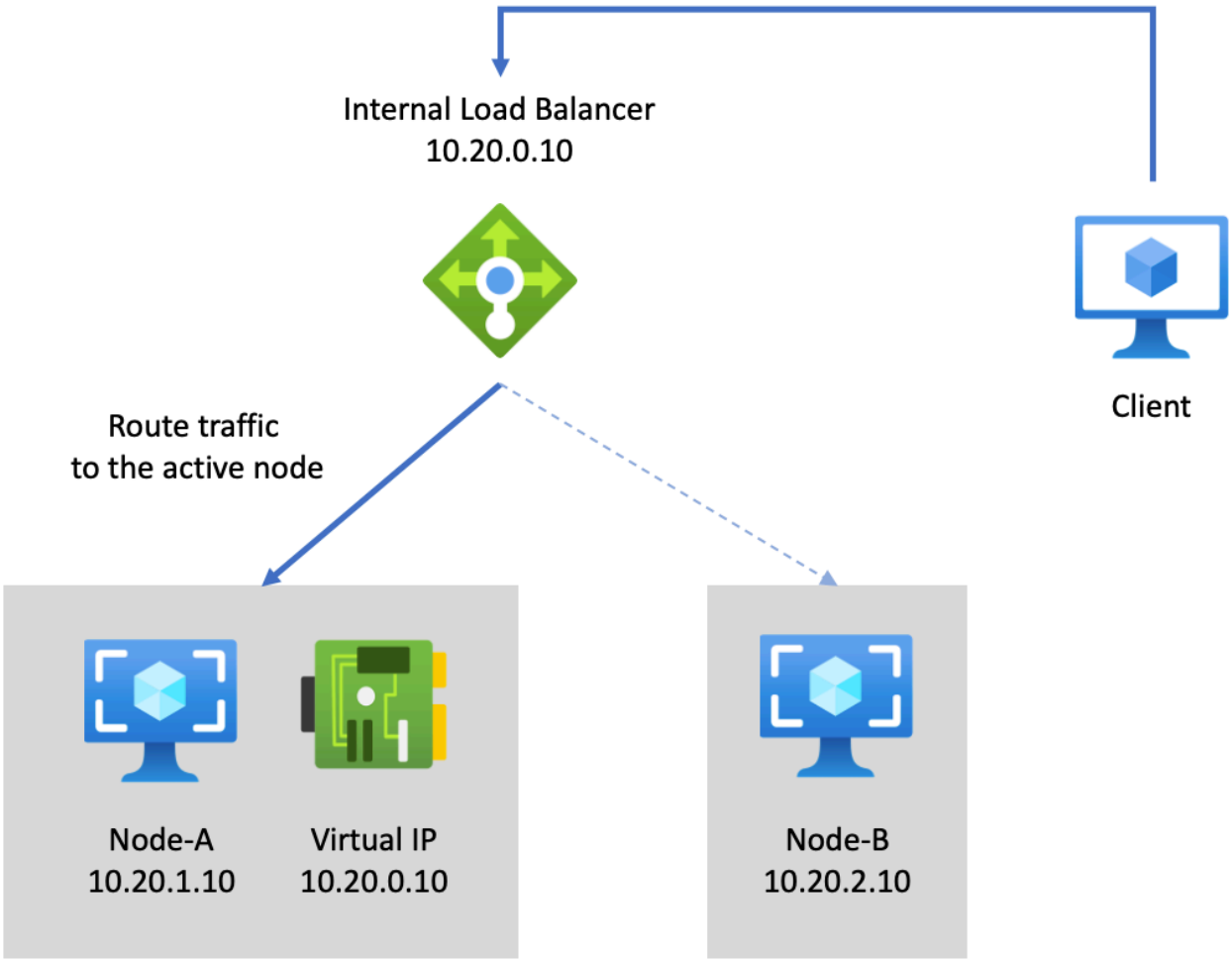
6. これで、下図のように Route53 リソースが作成されました。




realip は route53-node-active の下にあることに注意してください。これは、「route53-node-active」リソースが「realip」リソースに依存することを示しています。

# 11.2.7.2.4. Azure – 内部ロードバランサーの使用

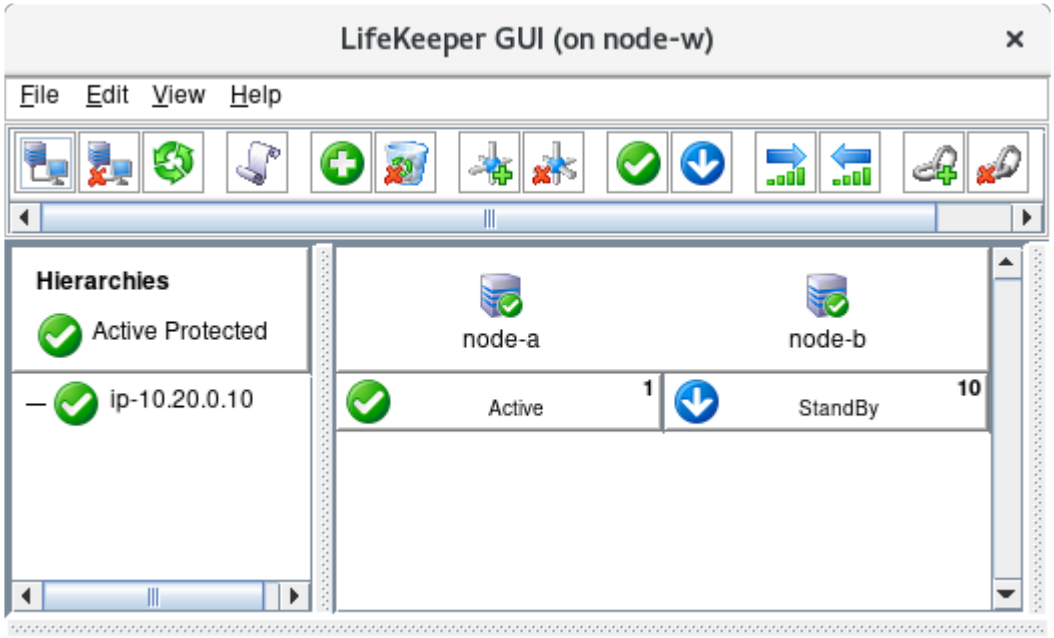
ここでは、アクティブノードを指す内部ロードバランサーを作成して、Azure内部ロードバランサーを使用してノードを切り替える方法について説明します。クライアントは、内部ロードバランサーが提供するフロントエンドIPアドレスに接続します。VMは「ヘルスチェックプローブ」機能によって定期的にチェックされ、内部ロードバランサーはリクエストをアクティブノードにルーティングします。




## IP リソースを作成する

Azure では、クライアントがロードバランサーに接続して、トラフィックをアクティブノードに分散できるようにします。理論上は、IP リソース（仮想IPアドレスを表す）は必要ありません。ただし、このセクションで後述するように、仮想 IP アドレスの作成は必要です。

Azure の場合、仮想 IP アドレスは、これらのノードの前にあるロードバランサーの IP アドレスと同じである必要があります。以下の例では、IP アドレス `10.20.0.10`を使用します。

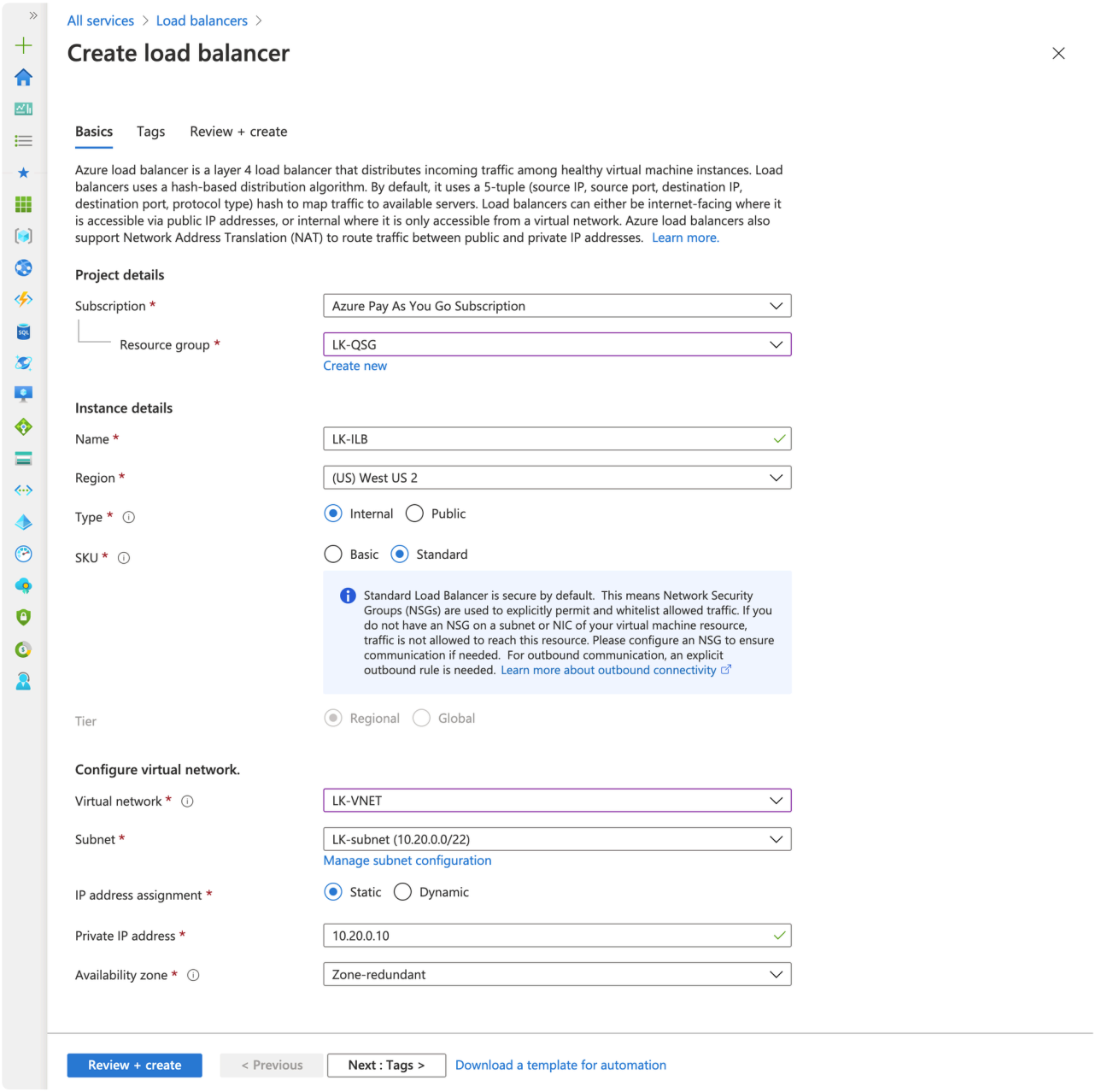



## 内部ロードバランサーを作成する

内部ロードバランサーはトラフィックをアクティブノードに分散し、次の手順で作成できます。ロードバランサーを構成するには、node-a でアプリケーションを起動して、LifeKeeper を介して構成する前に、ロードバランサーが動作していることを確認します。

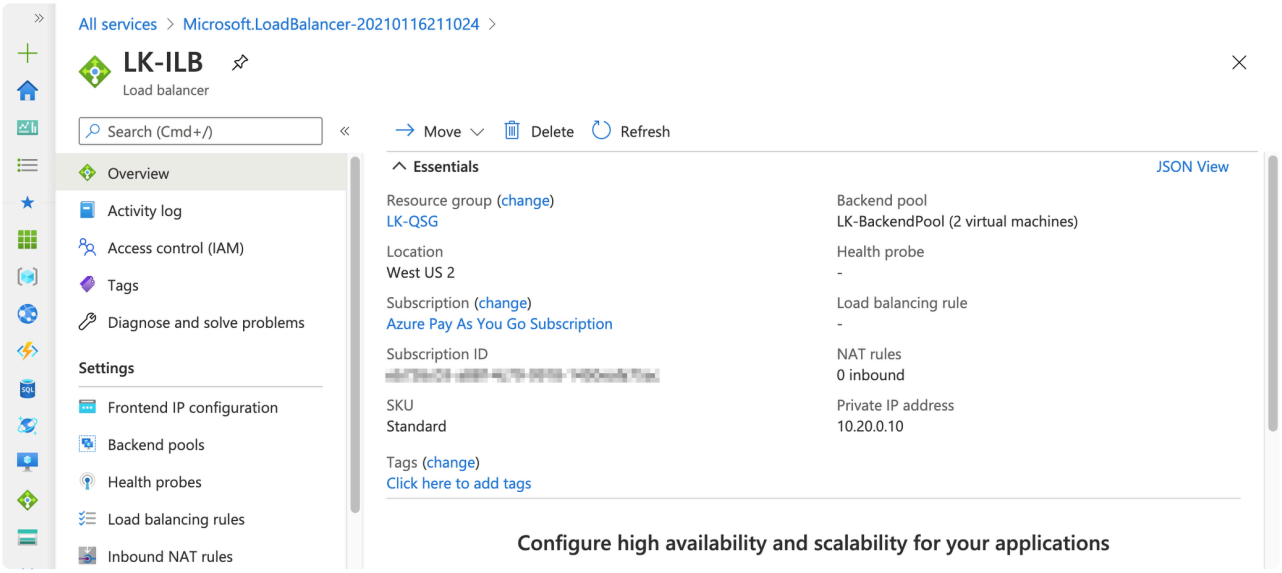
1. Azure ポータルで [ロードバランサー] に移動し、[追加] をクリックして以下のパラメーターを選択し、[Review + create] をクリックします。パラメータを確認すると、ロードバランサーの作成が可能になります。

| 項目 | 値 |
|---|---|
| リソースグループ | LK-QSG |
| 名前 | LK-ILB |
| リージョン | (仮想マシンと同じ) |
| タイプ | Internal |
| SKU | Standard<br>(ワークロードが Availability zone 全体に分散されるため、 Standard を選択します) |
| 仮想ネットワーク | LK-VNET |
| サブネット | LK-subnet |
| IPアドレスの割り当て | Static |
| プライベートIPアドレス | 10.20.0.10 |
| Availability zone | Zone-redundant |

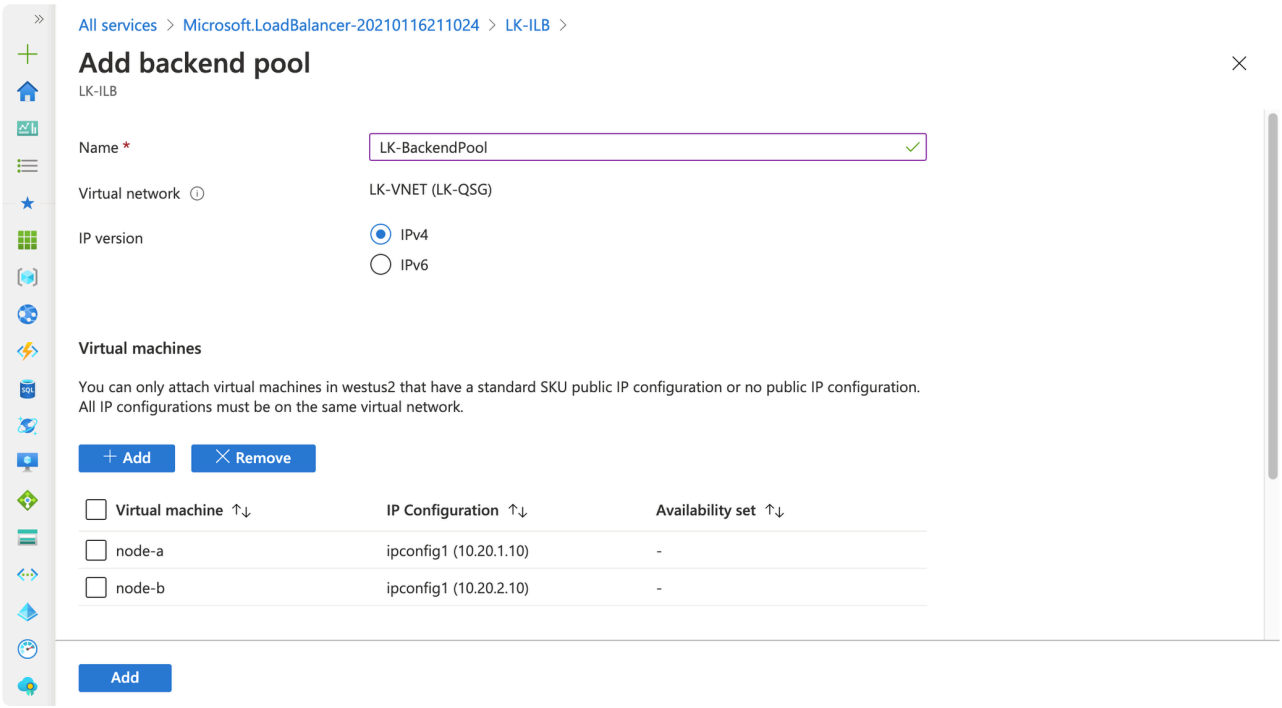



2. ロードバランサーが作成されたら、リソースページ（LK-ILB）に移動します。ロードバランサーの次のセクションを設定します（ロードバランサー設定ページの左側の [設定] セクションにあります）。

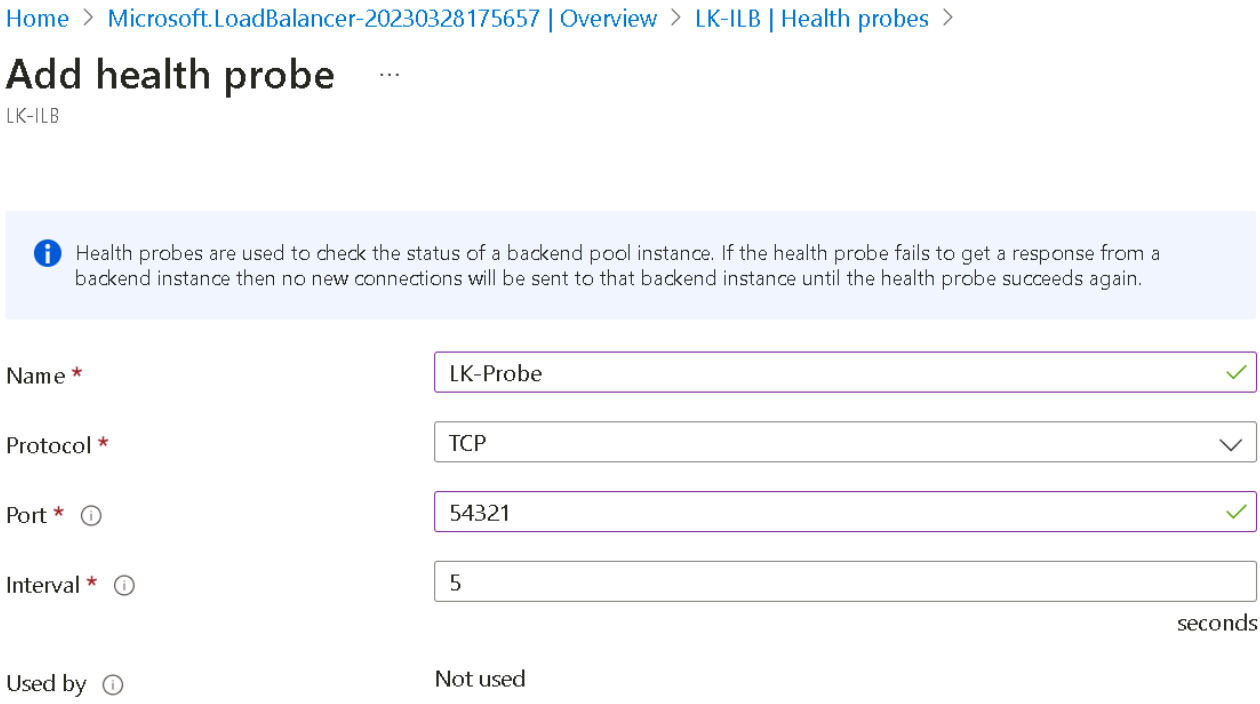
- バックエンドプール (ステップ 3)
- 正常性プローブ (ステップ 4)
- ロードバランシングルール (ステップ 5)




3. [バックエンドプール] ページに移動し、[追加] をクリックします。バックエンドプールページが開いたら、使用可能な仮想マシンのリストからnode-a と node-b を選択し、プールの名前（例：LK-BackendPool）を指定します。

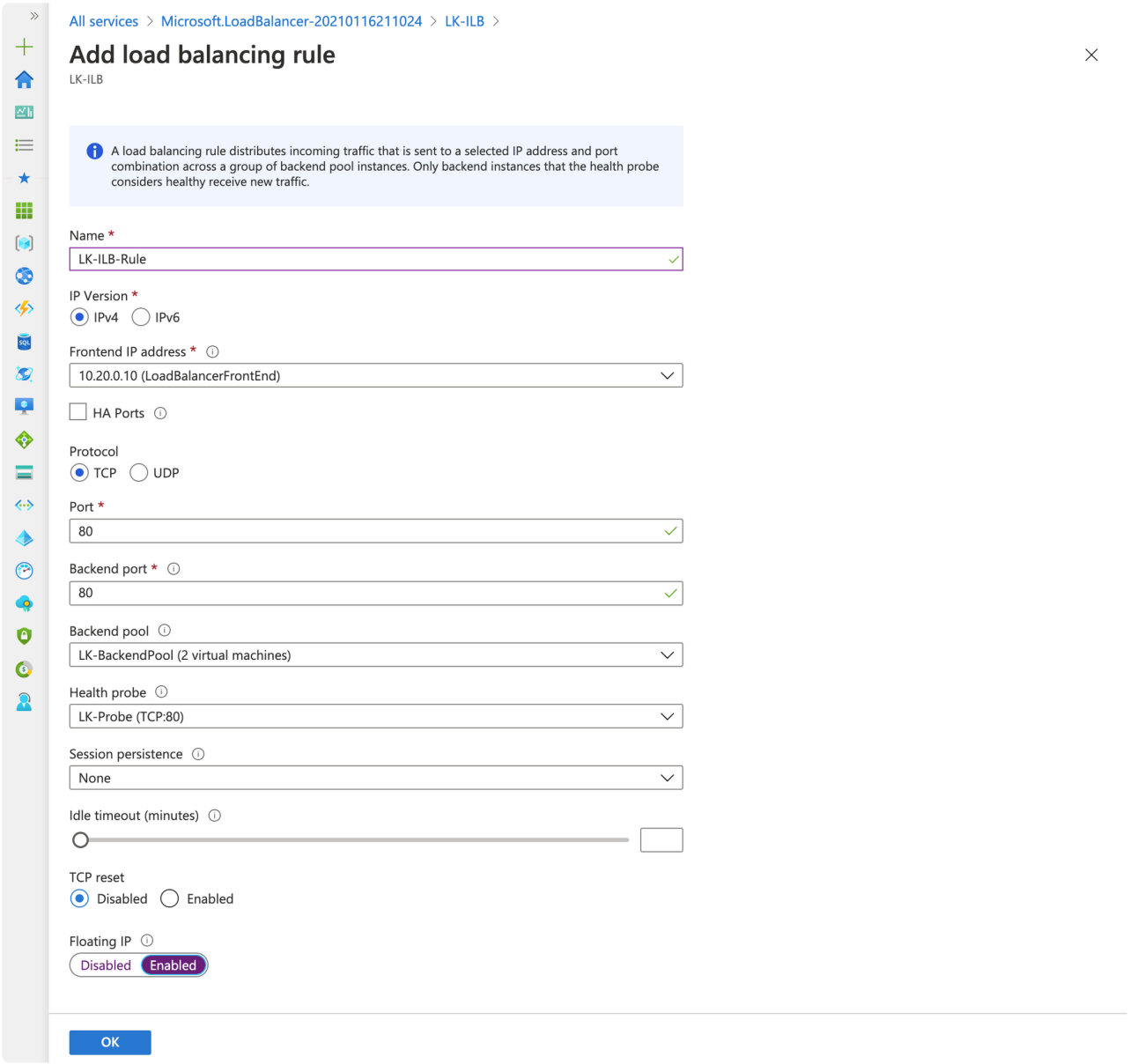



4. [正常性プローブ] ページに移動し、[追加] をクリックします。プローブの名前（例：LK-Probe）を入力し、ポートには LB Health Check リソース が使用するポート（例：54321）を指定します。このポートはアプリケーションが使用するポートとは異なるものを指定してください。




5. [負荷分散規則] ページに移動し、[追加] をクリックして新しい規則を作成します。次の値を入力します。

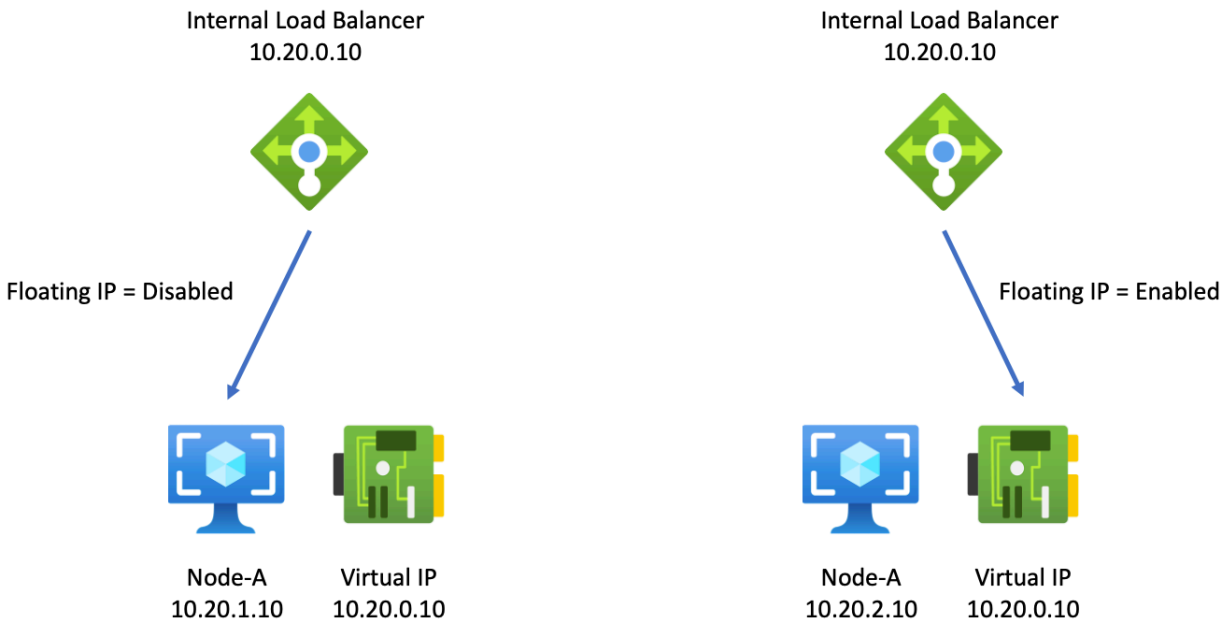
| 項目 | 値 |
|---|---|
| 名前 | LK-ILB-Rule |
| フロントエンド IP アドレス | 10.20.0.10 |
| プロトコル | TCP (アプリが TCP を使用していると想定) |
| ポート番号 # | アプリケーションで使用されるポート番号（例：httpdを保護する場合は80） |
| バックエンドポート | ポートと同じ |
| バックエンドプール | LK-BackendPool |
| 正常性プローブ | LK-Probe |
| セッション永続化 | なし |
| フローティング IP | 有効（無効を選択することもできます）。詳細については、次のセクションを参照してください。 |




# フローティング IP オプションの違い

フローティング IP オプションの違いにより、内部ロードバランサーがパケットを送信する宛先が変わります。

- [無効] が選択されている場合、ロードバランサーはノードのプライマリー IP にパケットを送信します。
- [有効] が選択されている場合、ロードバランサーはパケットを仮想 IP に送信します。

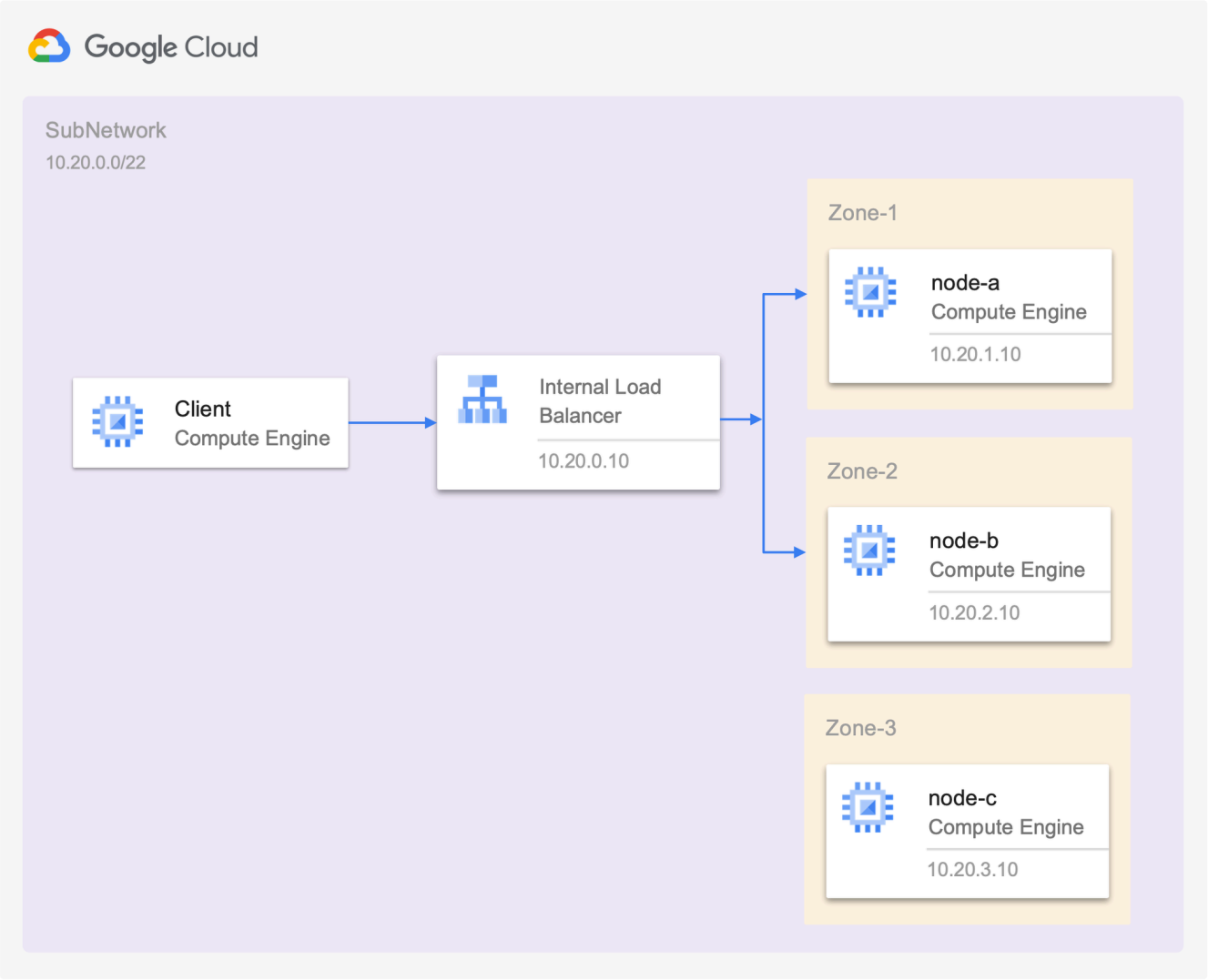



仮想 IP はアクティブノードに接続されているため、一見同じ操作に見えます。ただし、アクティブノードからの応答を仮想 IP から送信する必要がある場合は、「有効」を選択する必要があります。これは、LifeKeeper がこれらのノードを処理するためのより自然な方法です（オンプレミス環境と同じ方法）。アプリケーションがプライマリ IP アドレス（例：`10.20.1.10`）のみをListen する場合は、フローティング IP パラメーターは「無効」に設定する必要があります。

> **注記:** 正常性プローブは、プライマリー IP アドレス（例：`10.20.1.10`）にリクエストを送信することにより、ノードのステータスをチェックします。これは、「フローティング IP」オプションの値に関係なく当てはまります。

# 11.2.7.2.5. Google Cloud – 内部ロードバランサーの使用

ここでは、トラフィックをアクティブノードにルーティングする内部ロードバランサーを作成して、Google Cloud の内部ロードバランサーを使用してノードを切り替える方法について説明します。クライアントは、内部ロードバランサーが提供するフロントエンド IP アドレスに接続します。内部ロードバランサーは、ユーザー定義の「ヘルスチェックプローブ」機能を使用してバックエンドプール内の各 VM の状態を定期的にチェックし、クライアントのリクエストをアクティブノードにルーティングします。



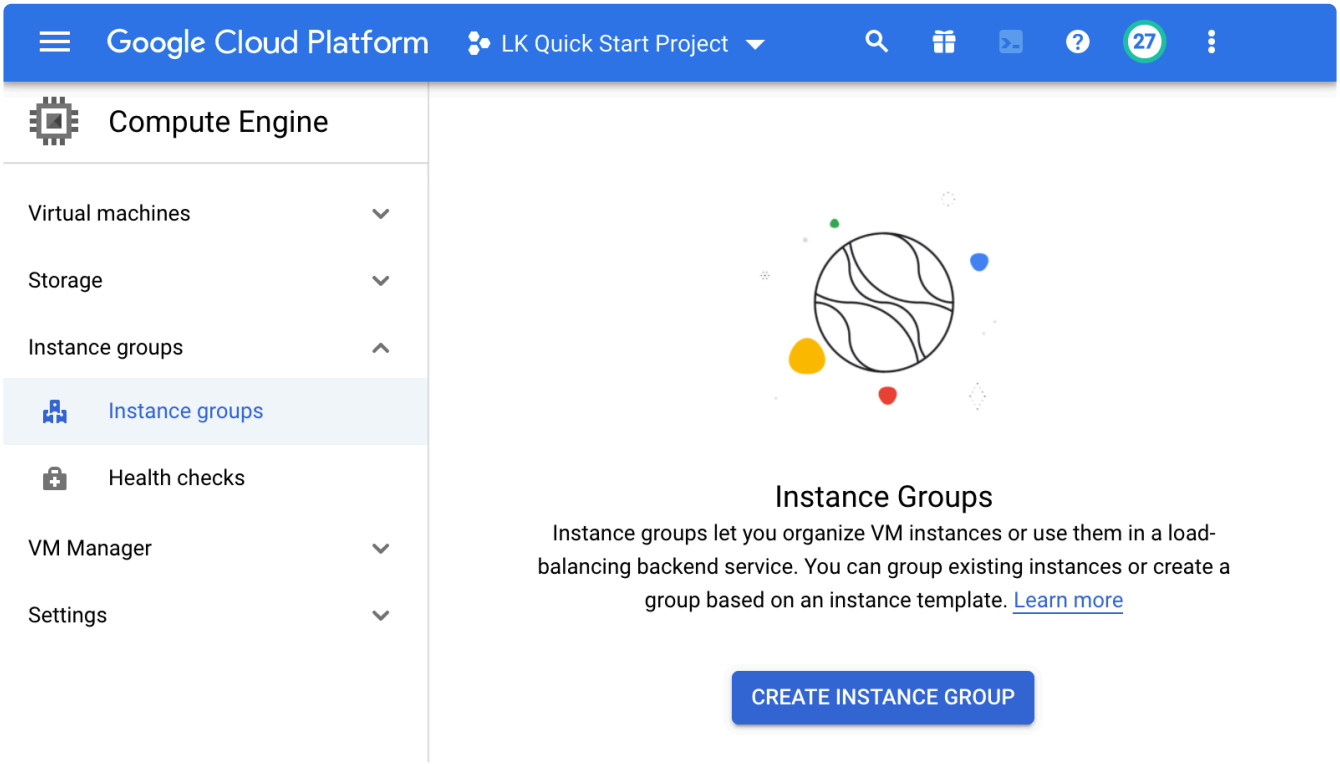

## インスタンスグループを作成する

内部ロードバランサーを使用するには、まず各ノードを含むインスタンスグループを作成します。以下のパラメータを使用してインスタンスグループを作成します。

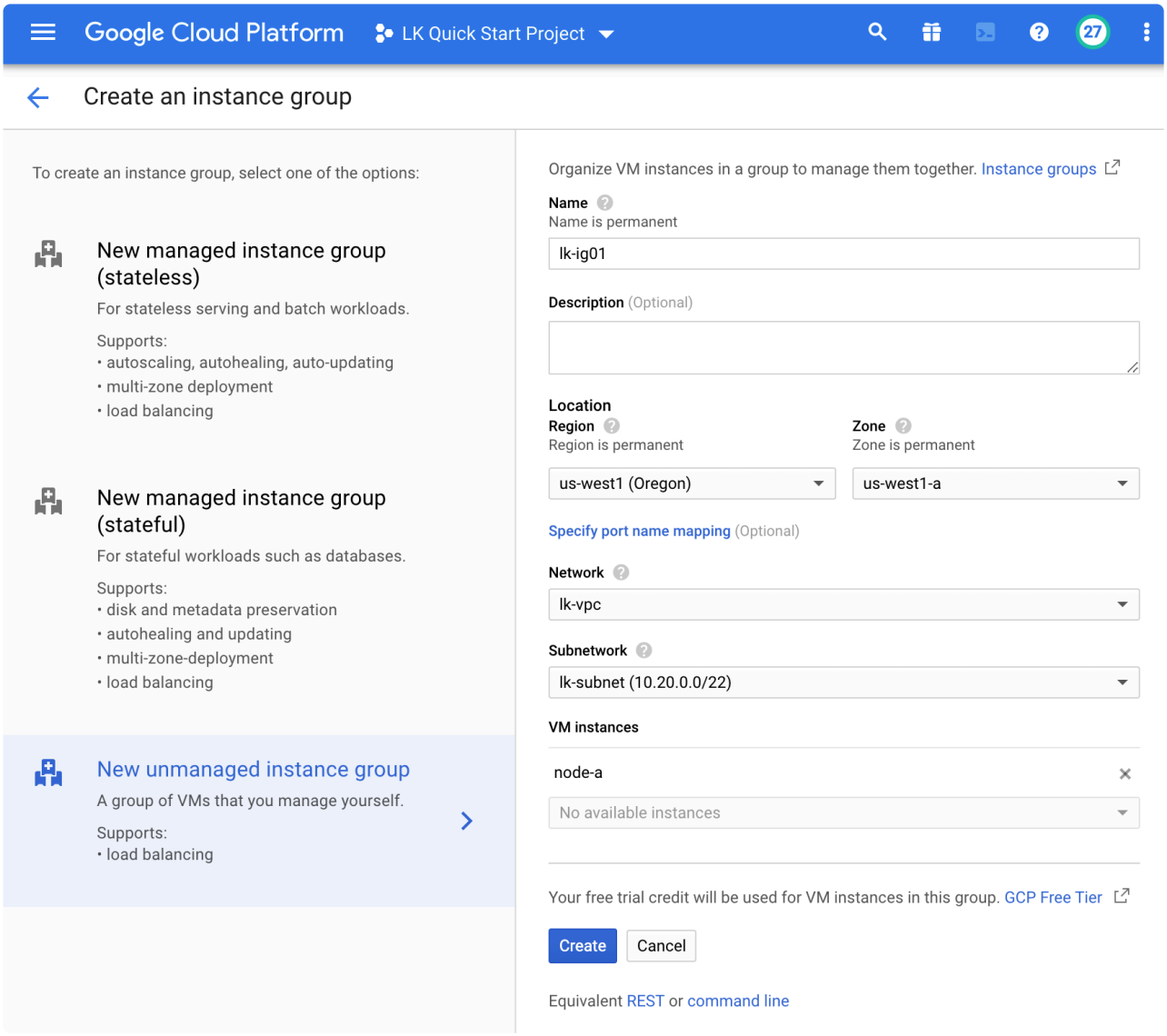
| 名前 | リージョン | ゾーン | ネットワーク | サブネット | VM インスタンス |
|---|---|---|---|---|---|
| lk-ig01 | <Deployment region> | a | lk-vpc | lk-subnet | node-a |

| | | | | | |
|---|---|---|---|---|---|
| lk-ig02 | | b | | | node-b |

1. ナビゲーションメニューから [Compute Engine] > [Instance Group] を選択します。

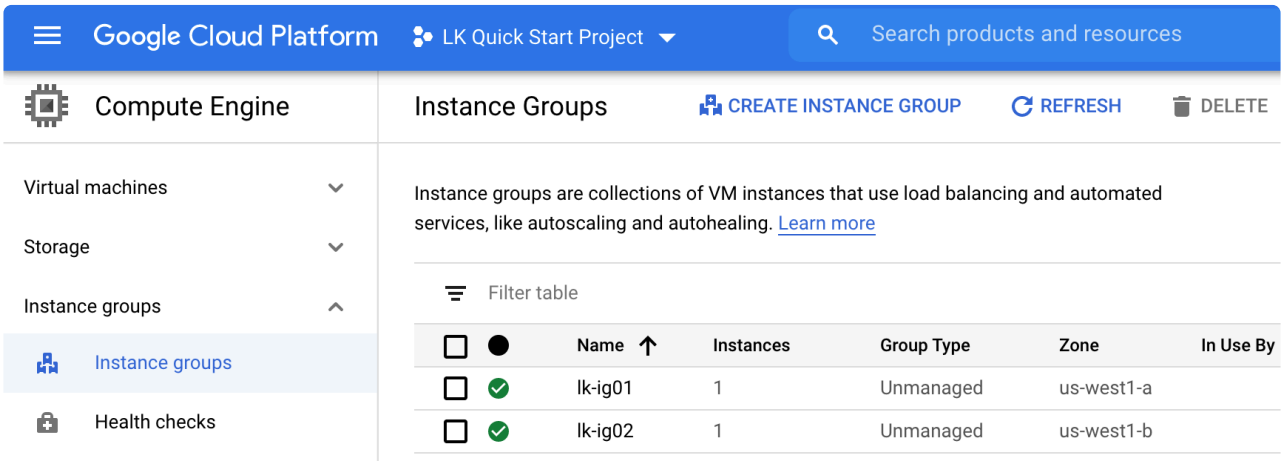



2. 「CREATE INSTANCE GROUP」、「New Unmanaged instance group」を選択し、表にリストされているパラメーターを入力します。




3. 同じ手順に従って `lk-ig02` を作成します。

4. これで、2つのインスタンスグループが作成されました。

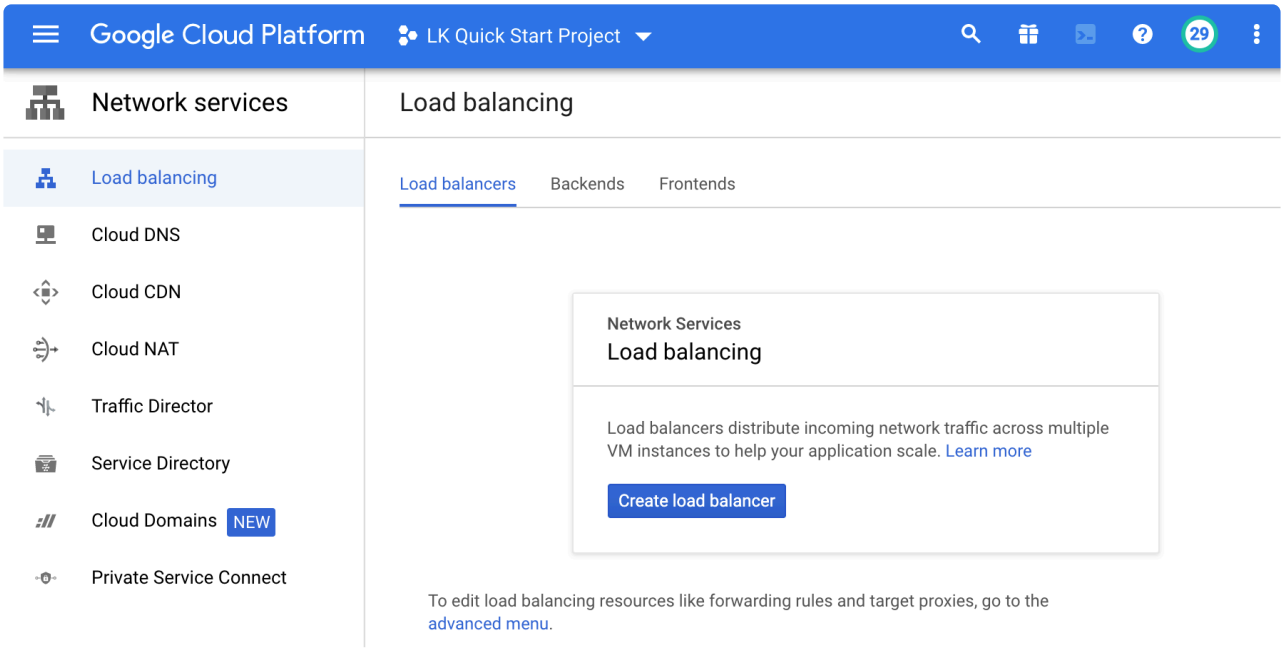



# 内部ロードバランサーを作成する

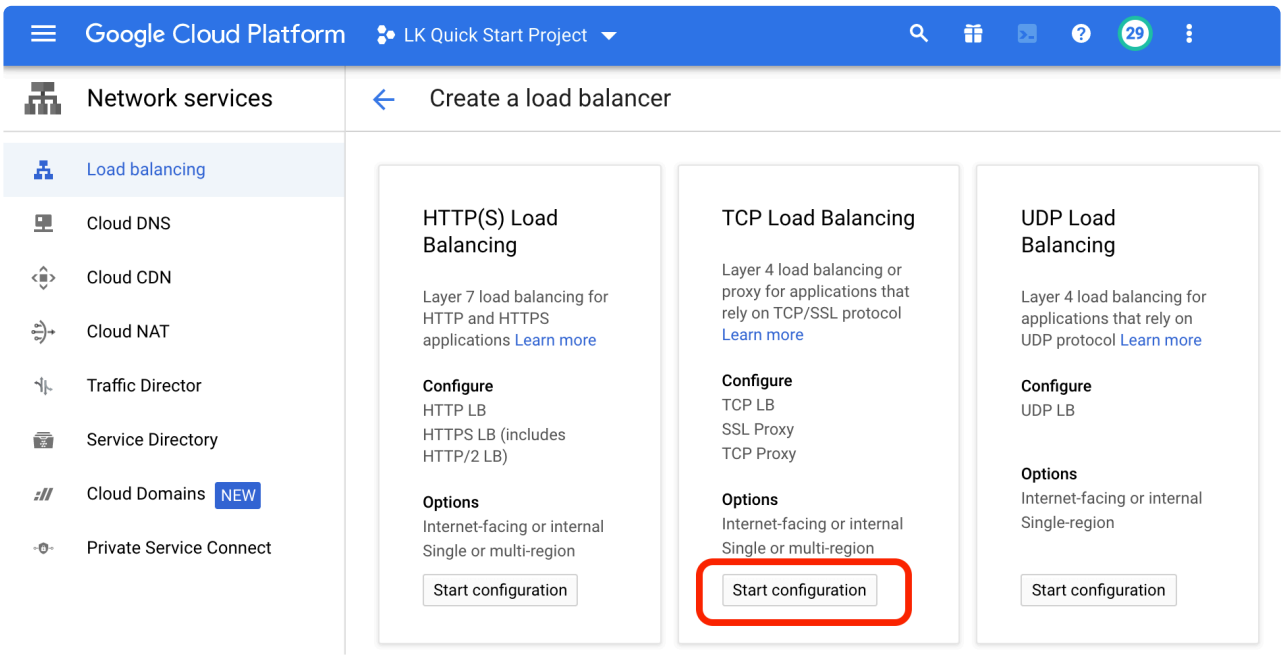
内部ロードバランサーはトラフィックをアクティブノードに分散し、次の手順を使用して作成できます。

ロードバランサーを構成するには、node-aでアプリケーションを起動します（LifeKeeperを使用して構成する前に、ロードバランサーが動作していることを確認します）。

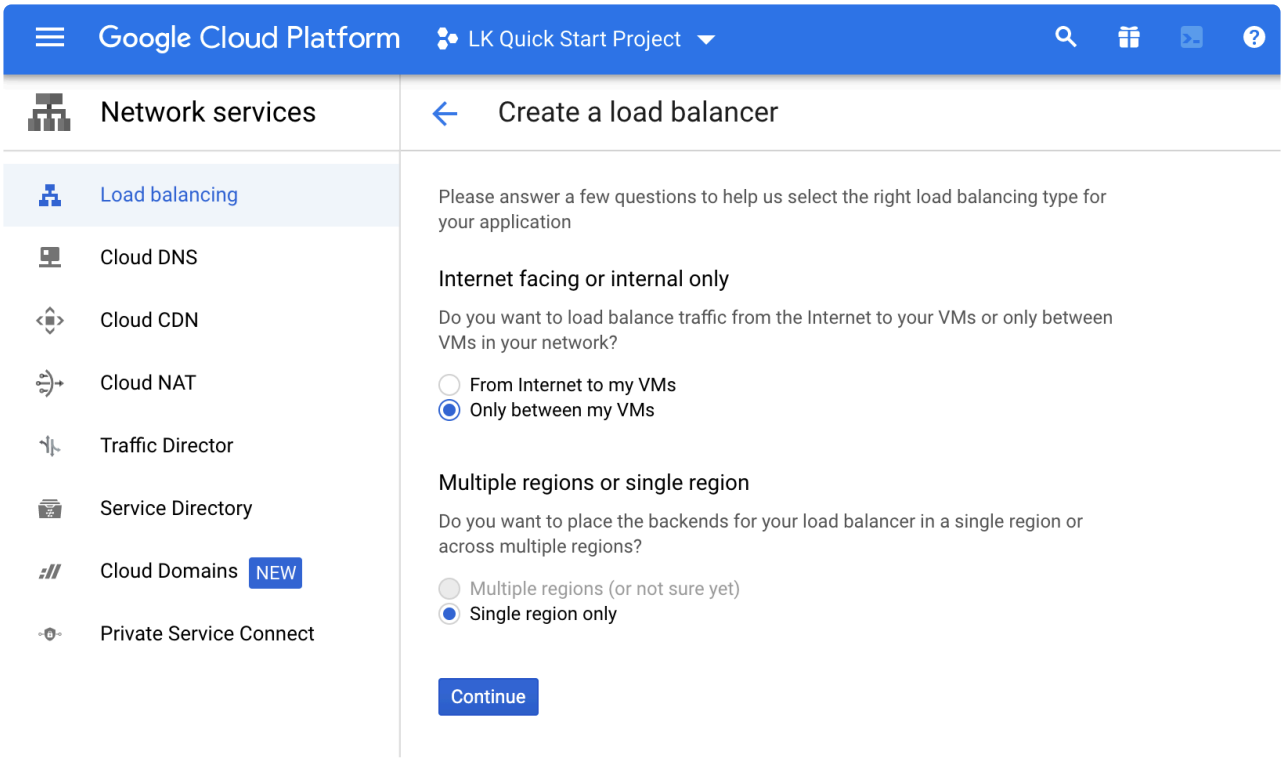
1. ナビゲーションメニューで、[Network Service] > [Load Balancing] を選択します。



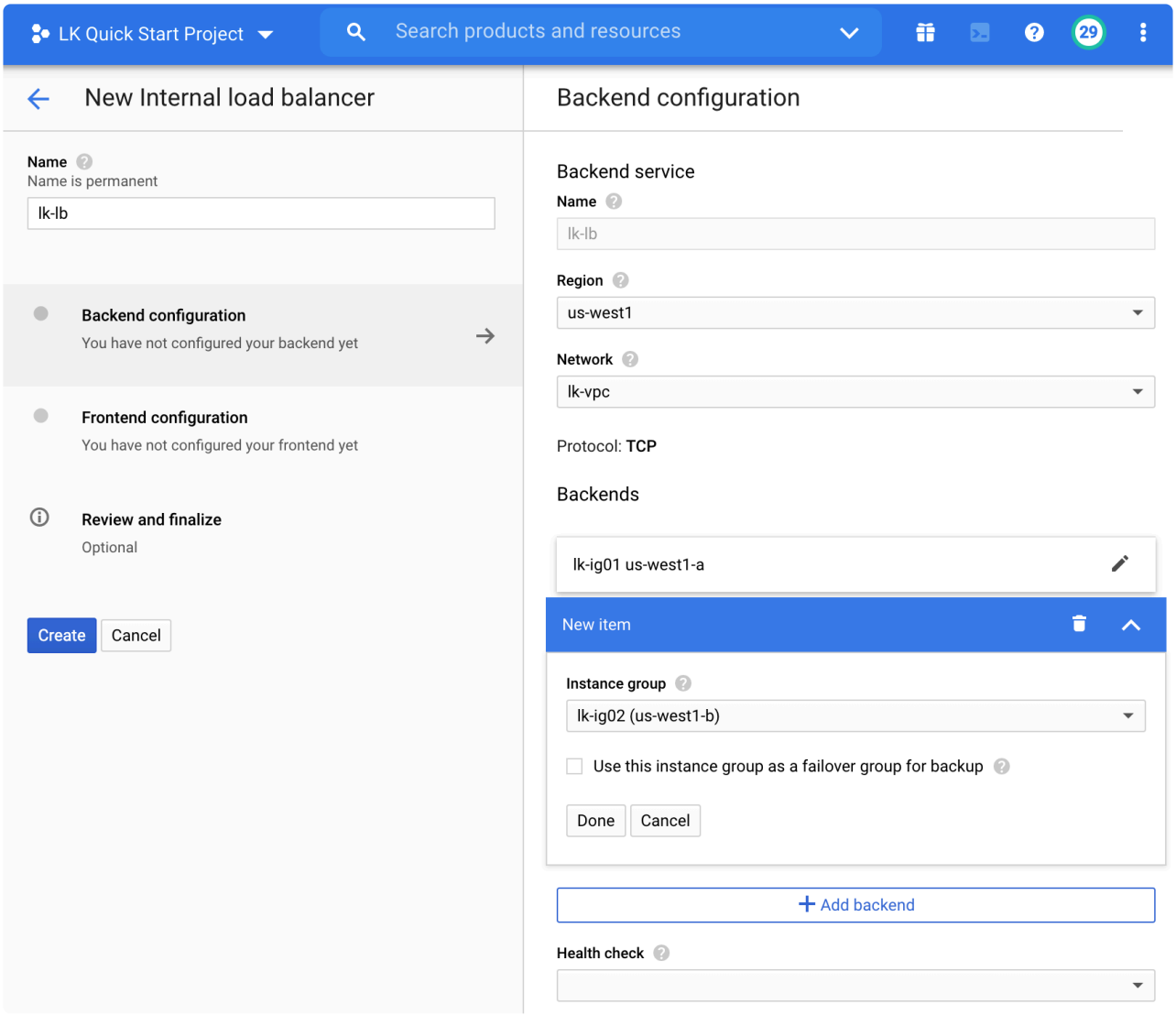

2. 「TCP負荷分散」の下の [構成を開始] をクリックします。



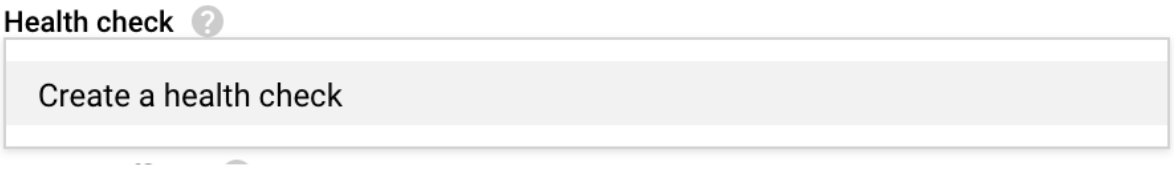

3. 「VM間のみ」（内部ロードバランサーを意味します）を選択し、「シングル リージョンのみ」を選択します。 [続行] をクリックします。



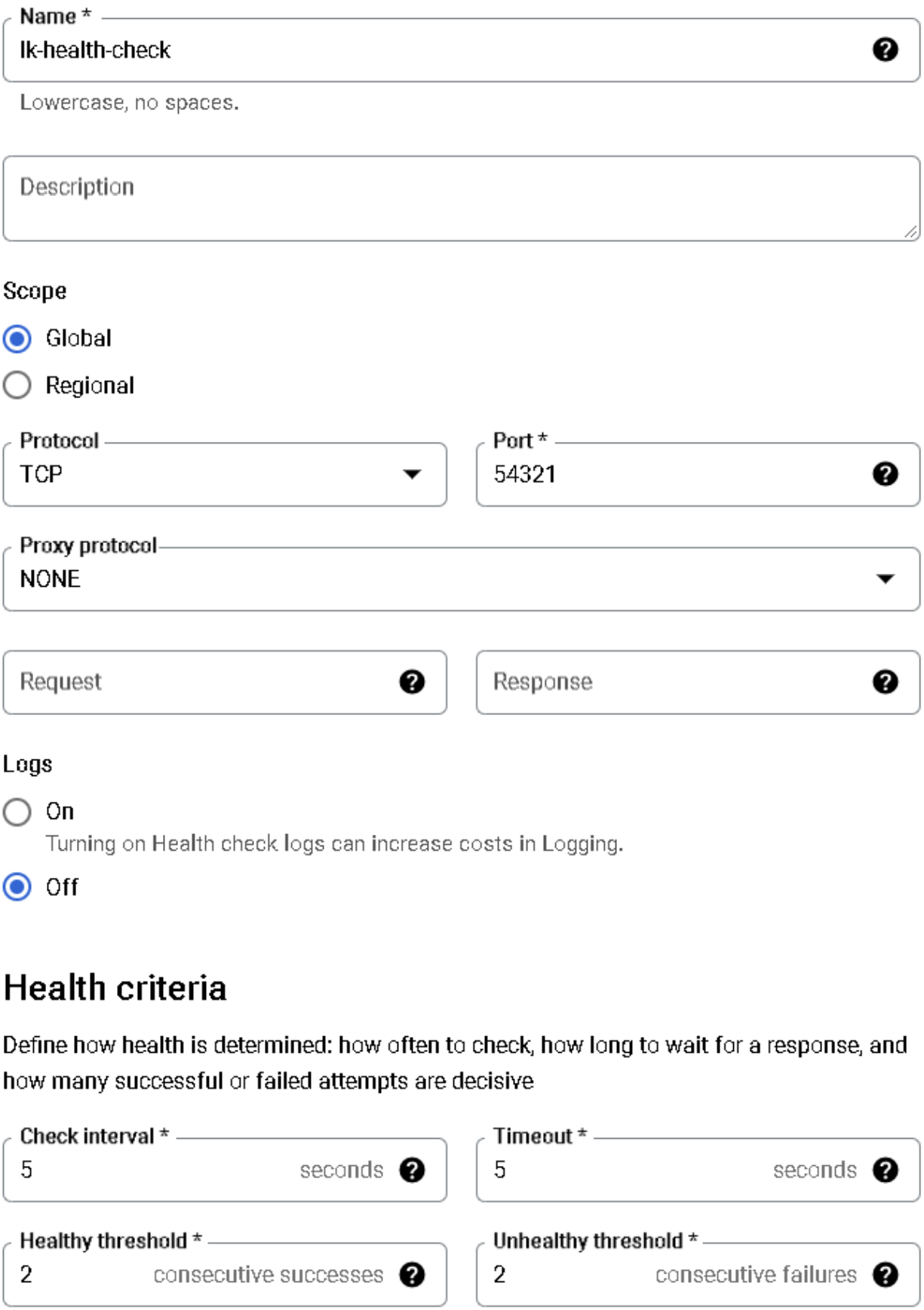

4. 名前をlk-lbと指定し、バックエンドパラメーターを構成します。[リージョン]、[ネットワーク]にlk-vpcを選択し、ドロップダウンメニューから2つのインスタンスグループ（lk-ig01とlk-ig02）を選択して、バックエンドプールに追加します。




5. バックエンドの構成ページの下部にある「ヘルスチェック」に移動します。 [ヘルスチェックの作成] を選択します。

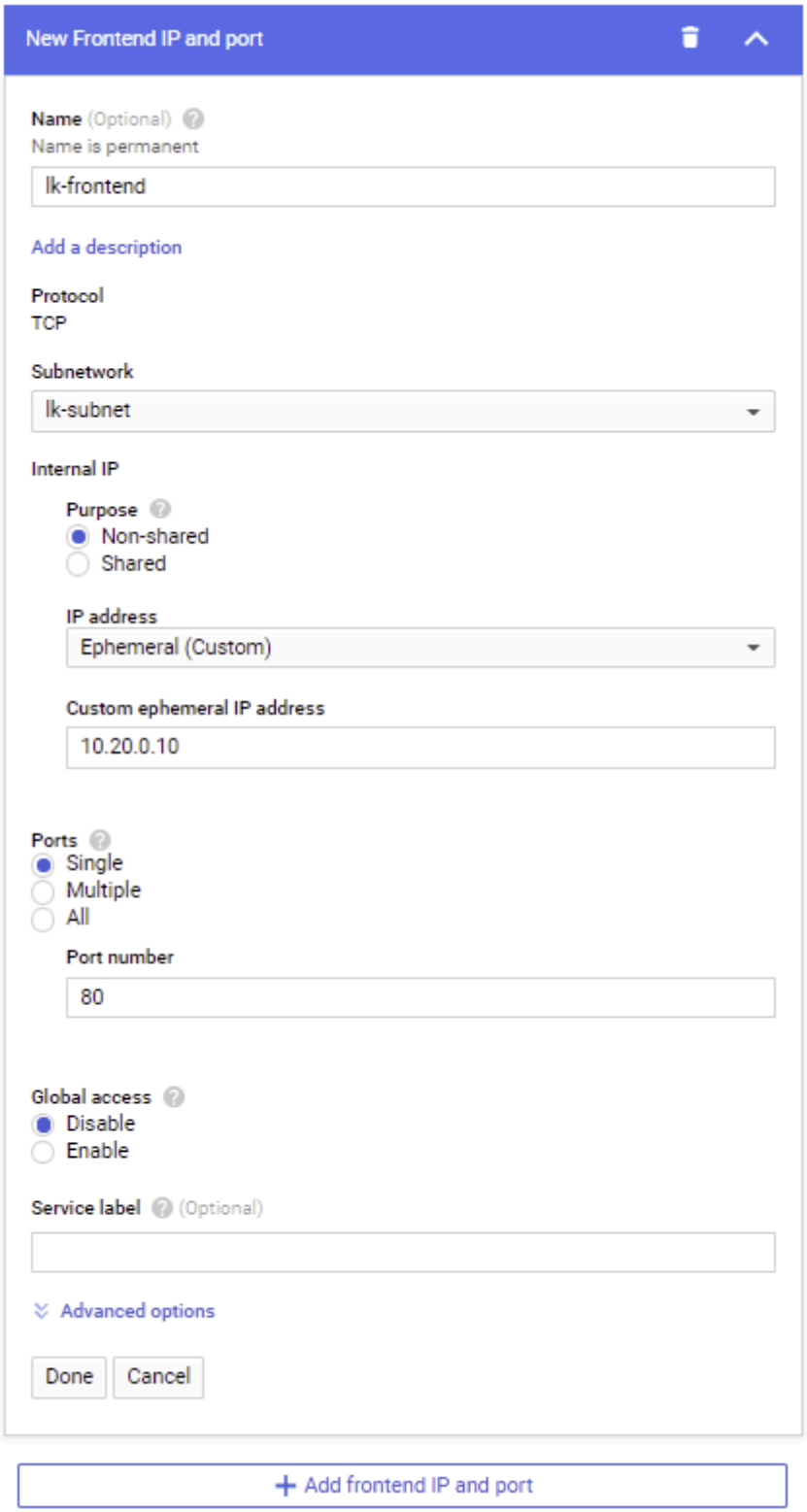



6. 名前は lk-health-check を選択し、[ポート] を選択して、 LB Health Check リソース が使用するポート（例：54321）を指定します。他のパラメーターを確認して、[保存して次へ] をクリックします。

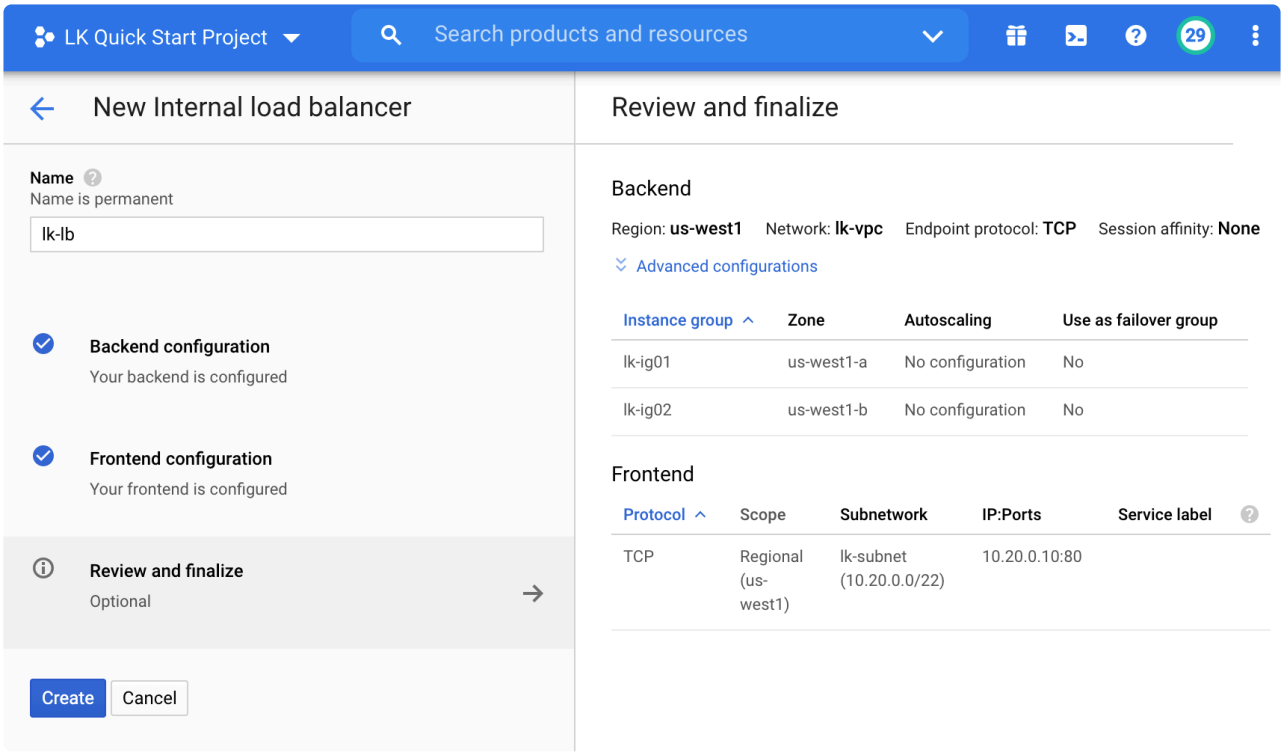



7. 「フロントエンドの設定」タブに移動します。フロントエンドの名前とサブネットワークを指定します。この例では、`lk-frontend` という名前を使用して、`lk-subnet` サブネットワークを選択します。[IP アドレス] で[エフェメラル（カスタム）] を選択し、ロードバランサーのフロントエンドで使用するカスタムエフェメラル IP アドレスを指定します。この例では、IP アドレス`10.20.0.10`

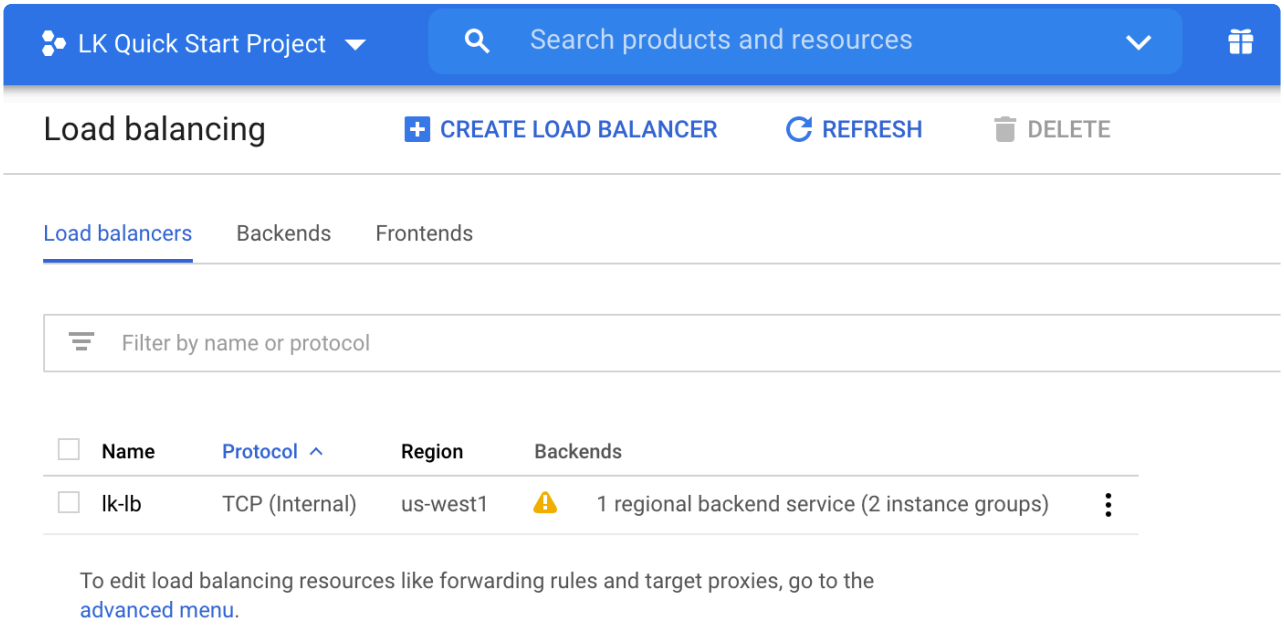
を使用します。 [ポート] で、トラフィックがルーティングされるアプリケーションに基づいて、必要に応じて[単一]、[複数]、または [すべて] を選択し、必要なポートを入力します。この例では、TCP ポート 80 でトラフィックを転送します。




8. 「確認と完了」タブに移動し、[作成] をクリックします。




9. ロードバランサーが作成されると、ステータスが表示されます。アプリケーション（httpdなど）はこれらのノードの1つでのみ実行されるため、 アイコンが表示される場合がありますが、これは想定内です。




10. これで、内部ロードバランサーが設定されました。node-aに保護するアプリケーションをインストールすると、先ほど設定したILBのフロントエンドIP（10.20.0.10）を介してアプリケーションに接続できます。次の例では、ILB を介して現在のターゲットのhttpトラフィックを確認する方法を示しています。

```
[gcp-user@node-c ~]$ curl http://10.20.0.10/
<html>
<body>
Test Page from node-a
</body>
</html>
```

## IP 転送を無効にする

特定の構成では、内部ロードバランサーの異なるバックエンドサーバーでホストされている 2 つのアプリケーションが、ロードバランサー自体のフロントエンド IP を介して通信する必要がある場合があります。これはたとえば、内部ロードバランサを使用して SAP AS ABAP デプロイメントのASCS および ERS インスタンスのフローティング IP フェイルオーバーを管理する場合に発生します。

Google Cloud でのルートベースのロードバランサーの実装の結果として、ロードバランサーのバックエンドサーバーからロードバランサーのフロントエンドIPに送信されるトラフィックは、ロードバランサーのヘルスチェックで正常と見なされるかどうかにかかわらず、デフォルトで、その元となる同じバックエンドサーバーにルーティングされます。 詳細については、 内部 TCP / UDP 負荷分散のトラブルシューティング「**トラフィックが予期したとおりにバックエンド VM に送信されない**」のセクションを参照してください。

以下は、Google Cloud の VM で IP 転送を無効にする手順です。これにより、上記のバックエンド内トラフィックの問題も解決されます。これらの手順は、ロードバランサーのバックエンドターゲットとして追加されるすべての VM で実行する必要があります。

1. etc/default/instance_configs.cfg fileファイルの `[NetworkInterfaces]` セクションで `ip_forwarding = false` を設定します。

```
# vi /etc/default/instance_configs.cfg
# cat /etc/default/instance_configs.cfg
[...]
[NetworkInterfaces]
[...]
ip_forwarding = false
[...]
```

2. 変更が正常に保存されたら、VMを再起動して、変更を有効にします。

IP 転送が無効になっている場合、ロードバランサーのヘルスチェックに応答する LifeKeeper LB Health Check リソースには、ロードバランサーのフロントエンド IP アドレスを保護し、ネットマスク/32（255.255.255.255）を使用する子 IP リソースが必要です。この子 IP リソースがないと、LB Health Check リソースは In-Service になりません。これらのリソースの作成と LB Health Check リソースの依存関係としての追加の詳細については、 ロードバランサーのヘルスチェックに応答する の「**フロントエンド IP リソースの作成**」と「**LB Health Check リソースの依存関係としてのフロントエンド IP リソースの追加**」のセクションを参照してください。

ロードバランサーと対応するLB Health Checkリソース階層の正常な動作をテストする方法の詳細については、 ロードバランサーのヘルスチェックに応答する の「**LB Health Checkリソースのスイッチオーバーとフェイルオーバーのテスト**」のセクションを参照してください。

# 11.2.7.2.6. ロードバランサーのヘルスチェックに応答する

**LB Health Check Kit** は、特定のリソースが現在 In-Service のクラスターノードにロードバランサートラフィックをルーティングするために、LifeKeeper リソース階層の一部として使用できます。これは、リソースが In-Service のクラスターノード上の、ユーザー指定の TCP ポートでリスナーを維持することで使用できます。

## フロントエンド IP リソースの作成

> ✻ **注記:** このセクションは、IP 転送が無効になっている Google Cloud の内部ロードバランサーを使用している場合にのみ実行してください。その他の構成の場合、以下の「LB Health Check リソースの作成」セクションに進んでください。

IP 転送を無効にして Google Cloud ロードバランサーを使用する場合 ( Google Cloud – 内部ロードバランサーの使用) の「IP転送を無効にする」セクションを参照）、ロードバランサーフロントエンドに関連付けられた IP アドレスは、各バックエンドサーバーのローカルネットワークインターフェースにネットワークマスク /32（255.255.255.255）を使用して追加する必要があります。

これらの構成では、ロードバランサーのフロントエンド IP アドレスは、クラウドエージェントプロセスによってネットワークインターフェースに自動的に **追加されることはありません** 。代わりに、各バックエンドサーバーのゲストオペレーティングシステム内に手動で追加する必要があります。これを実現する最も簡単な方法は、ロードバランサーの各フロントエンド IP アドレスに LifeKeeper IP リソースを作成し、次のセクションで作成する LB Health Check リソースの依存関係として追加することです。

IP リソースの作成 の手順に従って、次のパラメーターを使用して、ロードバランサーの各フロントエンド IPアドレスにIPリソースを作成および拡張します。

> ! ネットマスクパラメーターの値として（作成時と拡張時の両方で）255.255.255.255を使用することが重要です。拡張プロセス中に「Accept Defaults」を選択すると、通常、255.255.255.0 がスタンバイノードのネットワークマスクとして誤って使用され、予期せぬネットワークの問題が発生する可能性があるため、注意してください。

| フィールド | 値 |
|---|---|
| **Create Resource ウィザード** | |
| Switchback Type | intelligent |
| Server | node-a |
| IP Resource | <Frontend IP Address> |
| Netmask | 255.255.255.255 |

| | |
|---|---|
| Network Interface | <Network Interface> |
| IP Resource Tag | <Resource Tag> |
| **Pre-Extend** ウィザード | |
| Target Server | node-b |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend comm/ip Resource Hierarchy** ウィザード | |
| IP Resource | <Frontend IP Address> |
| Netmask | 255.255.255.255 |
| Network Interface | <Network Interface> |
| IP Resource Tag | <Resource Tag> |

ロードバランサーのフロントエンド IP アドレスごとに IP リソースが作成されたため、ロードバランサーのヘルスチェックプローブに応答する LB Health Check リソースを作成できます。

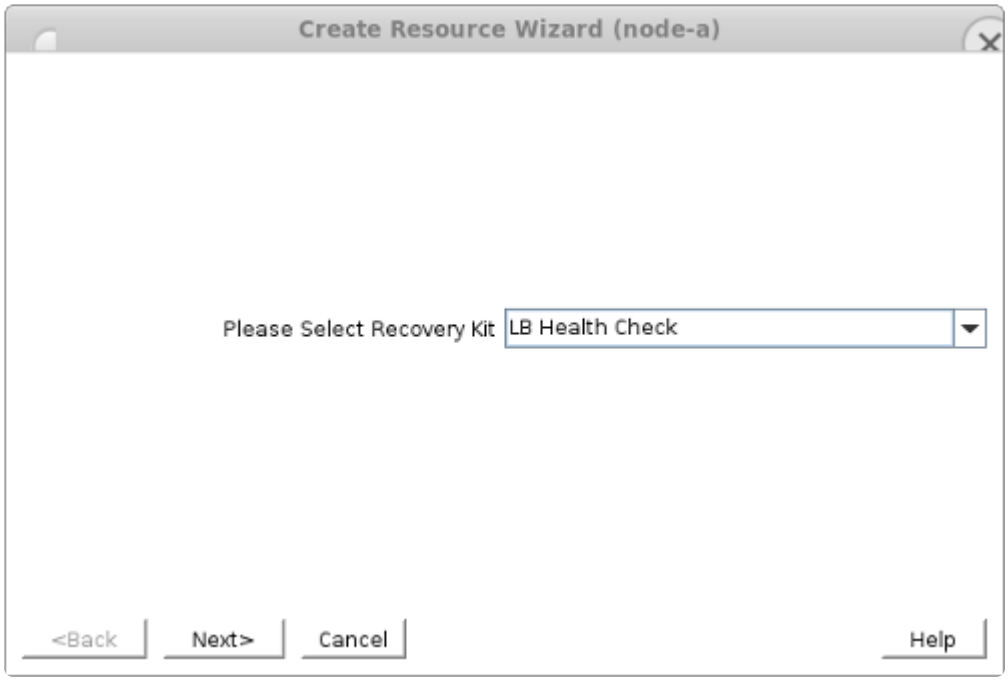
## LB Health Check リソースの作成

この例では、TCP ポート 54321 で Listen するサーバー node-a 上にサンプルの LB Health Check リソースを作成します。

! このリソースが応答しているロードバランサーのヘルスチェックに応じて、サーバー、ポート、またはリソースタグの名前を変更してください。たとえば、各バックエンドサーバーで TCP ポート 41098 をプローブするヘルスチェックを設定した場合、LB Health Check リソースの Reply daemon port には、ここに示す値「54321」ではなく「41098」が含まれます。LB Health Check リソースに使用するポートを選択する場合、1024～65535 の範囲の開いているポートを使用できます。

✻ **注記:** リソースを正常に In-Service にするには、リソースの作成プロセス中にロードバランサーがサーバーにヘルスチェックプローブをアクティブに送信している必要があります。つまり、LB Health Check リソースを作成する前に、ロードバランサーとヘルスチェックが完全に構成されている必要があります。Microsoft Azure および Google Cloud 環境で内部ロードバランサーを作成する方法の詳細については、 クラウド環境でのノードの切り替え のサブセクションを参照してください。

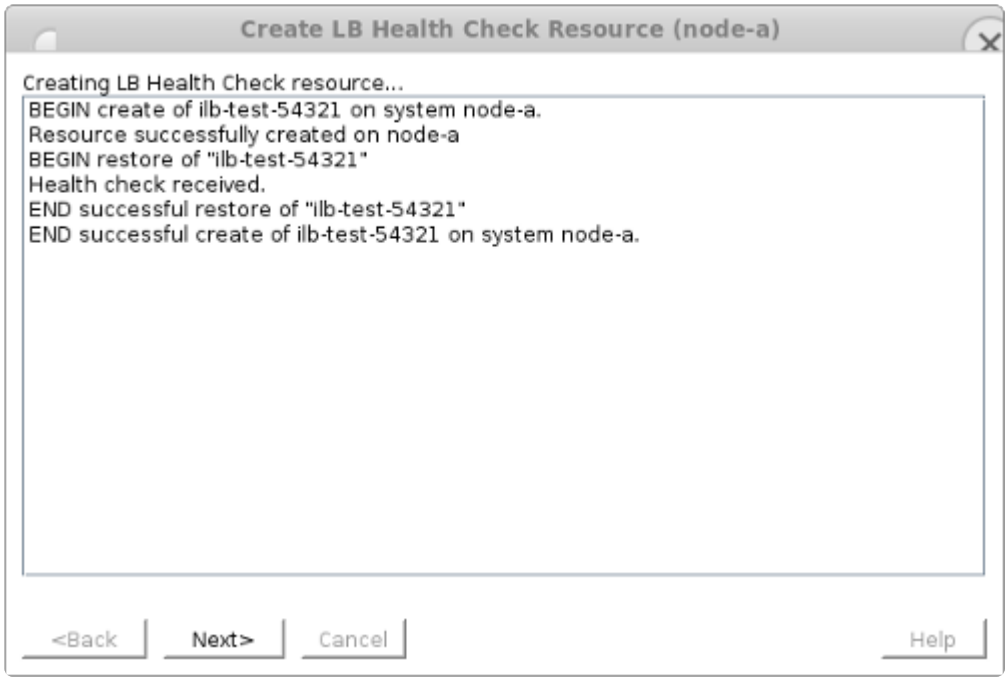
1. LifeKeeper GUIで、 をクリックして Create Resource ウィザードを開きます。「LB Health Check Kit」を選択します。




2. Create Resource ウィザードに次の値を入力し、プロンプトが表示されたら [Create] をクリックします。 アイコンは、デフォルトのオプションが選択されていることを示します。

| フィールド | 値 |
|---|---|
| Switchback Type | Intelligent |
| Server | node-a |
| Reply daemon Port | 54321 |
| Reply daemon message | 空欄のままにします |
| LB Health Check Resource Tag | ilb-test-54321 |

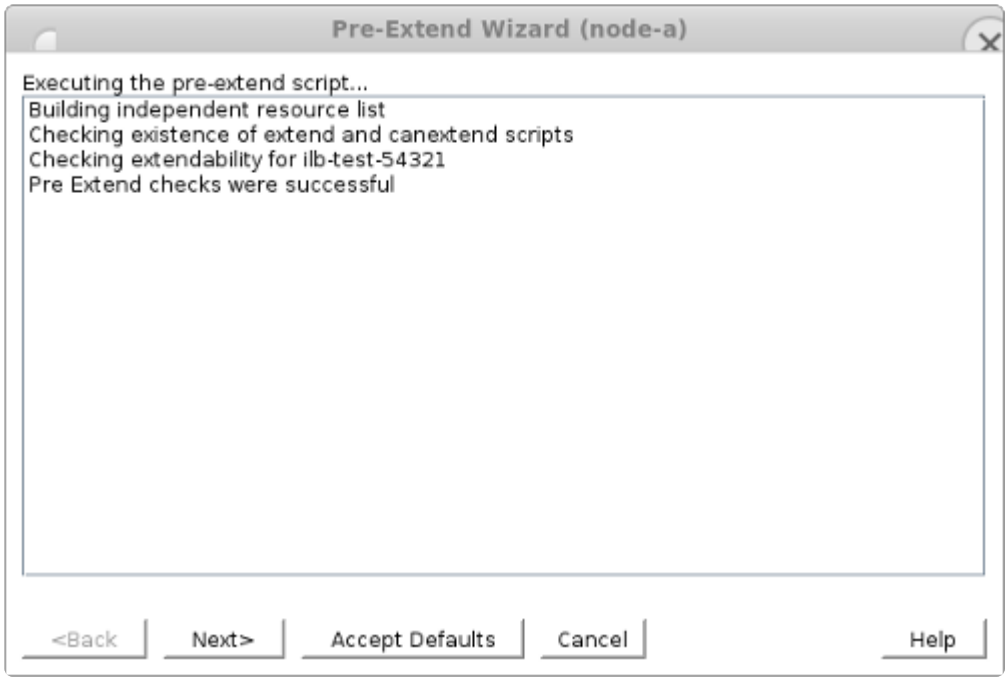
リソースが作成され、正常にサービスが開始されたら、 **[Next>]** をクリックして **Pre-Extend ウィザード** に進みます。




3. **Pre-Extend** ウィザード に次の値を入力します。 アイコンは、デフォルトのオプションが選択されていることを示します。

| フィールド | 値 |
|---|---|
| Target Server | node-b |
| Switchback Type | Intelligent |
| Template Priority | 1 |
| Target Priority | 10 |

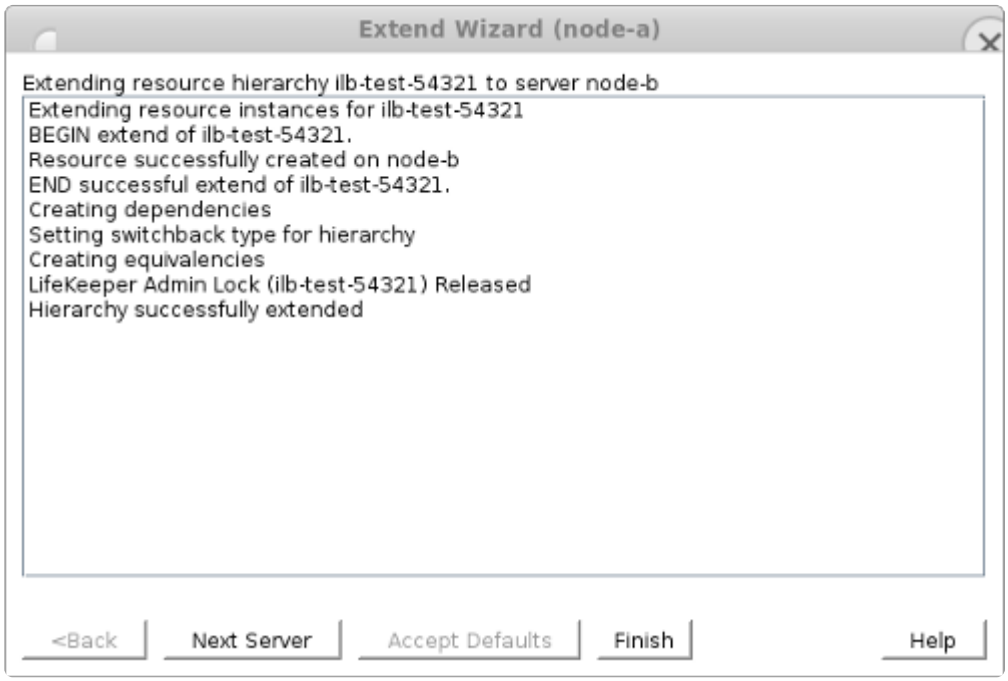
事前拡張チェックに合格したら、 **[Next>]** をクリックして、 **Extend Resource Hierarchy** ウィザードに進みます。




4. **Extend Resource Hierarchy** ウィザードに次の値を入力し、プロンプトが表示されたら **[Extend]** をクリックします。 アイコンは、デフォルトのオプションが選択されていることを示します。

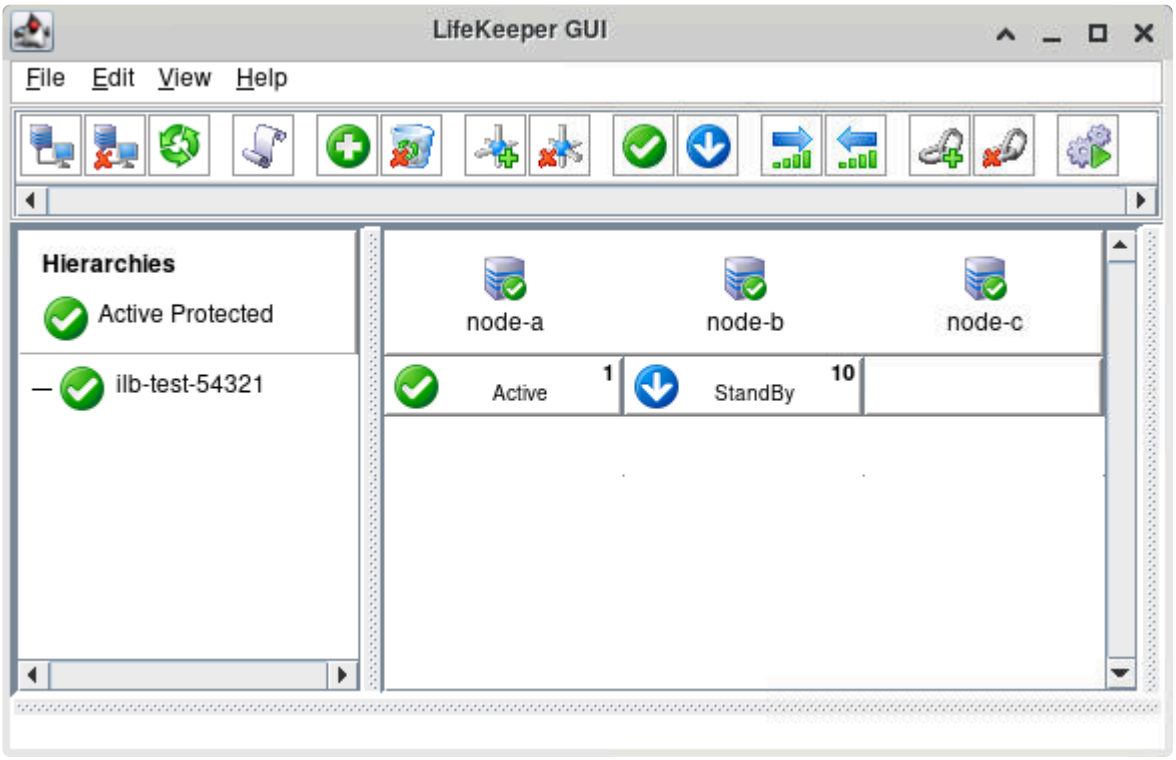
| フィールド | 値 |
|---|---|
| LB Health Check Resource Tag | ilb-test-54321 |

リソースが正常に拡張されたら、 **[Finish]** をクリックします。




5. LifeKeeper GUI に戻ると、新しく作成された **ilb-test-54321** リソースが node-a でアクティブ、node-b でスタンバイであることがわかります。この状態では、ポート 54321 で TCP ヘルスチェッ

クを実行する TCP ロードバランサーは、node-a を正常、node-b を異常として扱い、すべてのロードバランサートラフィックを node-a にルーティングします。このリソースは、保護されたアプリケーションとともにリソース階層に配置されると、ロードバランサーのトラフィックが常に、アプリケーションが現在実行されているサーバーにルーティングされるようにします。



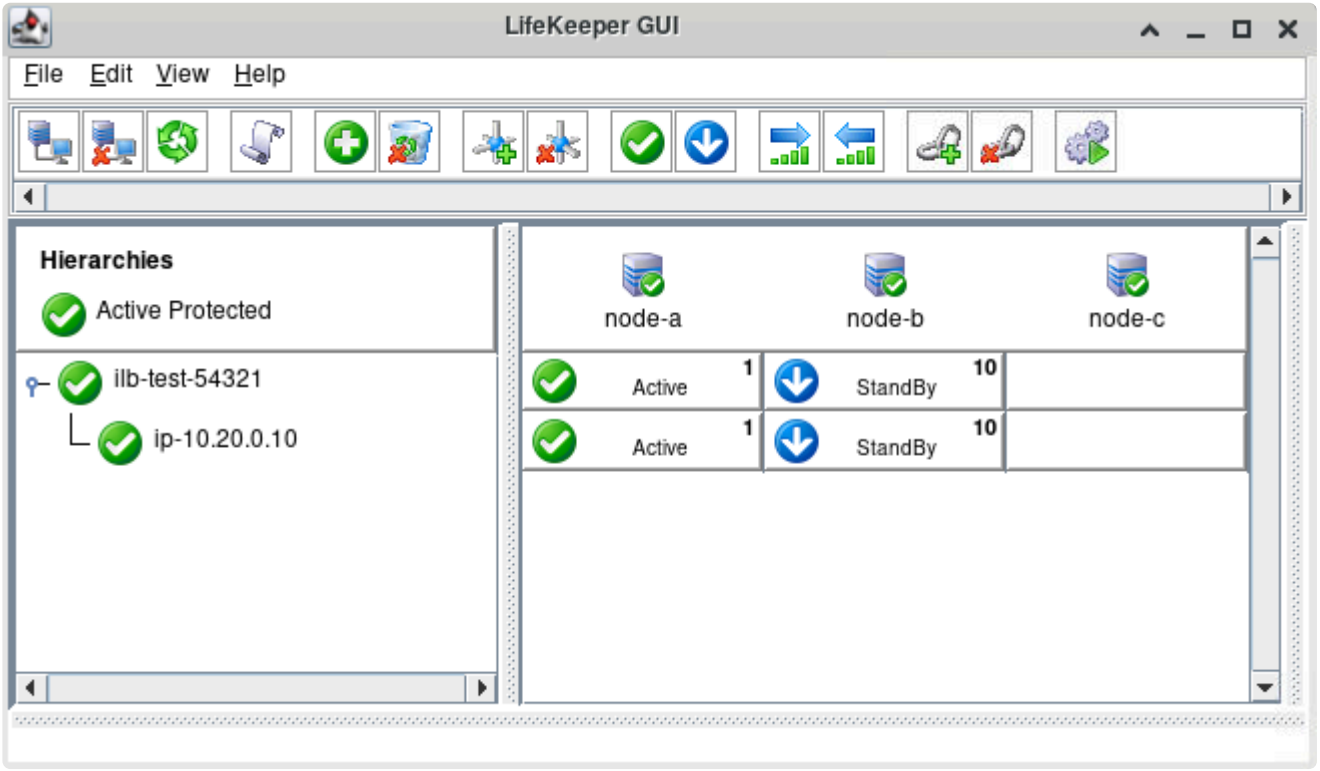

## LB Health Check リソースの依存関係としてフロントエンド IP リソースを追加する

> 注記: このセクションは、IP 転送が無効になっている Google Cloud の内部ロードバランサーを使用している場合にのみ実行してください。その他の構成の場合、以下の「LB Health Check リソースの切り替えとフェイルオーバーのテスト」セクションに進んでください。

IP 転送を無効にして Google Cloud ロードバランサーを使用する場合 ( Google Cloud – 内部ロードバランサーの使用 IP アドレスを保護する IP リソース（ネットワークマスク 255.255.255.255 を使用）は、LB Health Check リソースの依存関係として追加する必要があります。上記の「フロントエンドIPリソースの作成」セクションで作成したIPリソースごとに、次の手順を実行します。

1. **ilb-test-54321** リソースを右クリックし、ドロップダウンメニューから **[Create Dependency...]** を選択します。
2. **子リソースタグ** に、ロードバランサーのフロントエンドIPアドレスを保護するリソースを指定します。
3. **[Next>]** をクリックして続行し、 **[Create Dependency]** をクリックして依存関係を作成します。

IP リソースが依存関係として追加されると、階層は次のようになります




## LB Health Check リソースのスイッチオーバーとフェイルオーバーをテストする

このセクションでは、node-a と node-b をバックエンドターゲットとして使用し、次のプロパティを持つ内部ロードバランサーを作成していると想定します。

- フロントエンド内部 IP: 10.20.0.10
- ポート 54321 での TCP ヘルスチェック

また、前のセクションで作成した **ilb-test-54321** LB Health Check リソースは、現在 node-a でアクティブになっているとします。

便宜上、各サーバーのホスト名を返すだけの一時的な Apache Web サーバーをセットアップします。**node-aとnode-bの両方** で次のコマンドを実行します。SLES サーバーにインストールする場合は、コマンドを適宜調整してください（例：zypper install を使用）。

```
# yum install -y httpd
# systemctl start httpd
# echo $(hostname) > /var/www/html/index.html
```

続行する前に、node-a および node-b の TCP ポート 80 でトラフィックが許可されていることを確認してください。

次に、 **ilb-test-54321** LB Health Checkリソースのスイッチオーバーおよびフェイルオーバー機能をテスト

します。

1. **ilb-test-54321** リソースが node-a でアクティブで node-b でスタンバイの場合、各サーバーで次のコマンドの出力を確認します。

```
[root@node-a ~]# curl http://10.20.0.10
node-a
[root@node-b ~]# curl http://10.20.0.10
node-a
```

> 注記: node-b で curl コマンドを実行すると「node-b」が返され、サーバーが Google Cloud にある場合は、 Google Cloud – 内部ロードバランサーの使用 転送を無効にする」セクションを参照してください。そのセクションの手順が完了したら、ここに戻って期待される動作を確認します。

2. node-a で以下のコマンドを実行します。

   ```
   [root@node-a ~]# while true; do curl http://10.20.0.10; sleep 1; done
   ```

   そして、 **ilb-test-54321** リソースの node-b へのスイッチオーバーを開始します。スイッチオーバーが正常に完了したら、Ctrl-C（SIGINT）を使用して、node-a で実行中のコマンドを終了させます。

   コマンドの出力は次のようになります。

   ```
   …
   node-a
   node-a
   node-a
   [switchover occurs]
   node-b
   node-b
   node-b
   …
   ```

   特に、ロードバランサーは、node-b へのルーティングを開始する前に、node-a へのトラフィックのルーティングを完全に停止する必要があります。スイッチオーバーポイント付近の出力が次のようになっている場合：

   ```
   …
   node-a
   [switchover occurs]
   node-b
   node-a
   node-b
   node-a
   ```

```
node-b
node-a
node-b
node-b
node-b
…
```

ヘルスチェックのプロパティを編集して、バックエンドインスタンスが異常と判定されてロードバランサープールから削除される前に、ヘルスチェックスローブ間の時間を短くしたり、ヘルスチェックスローブの失敗の最小回数を減らしたりする必要がある場合があります。詳しくは、「ロードバランサーのヘルスチェックパラメーターのチューニング」のセクションを参照してください。

3. node-b で **ilb-test-54321** リソースがアクティブになっている状態で、node-a で次のコマンドを実行します。

```
[root@node-a ~]# while true; do curl http://10.20.0.10; sleep 1; done
```

node-b を強制的に再起動して、 **ilb-test-54321** リソースの node-a へのフェイルオーバーを開始します。

```
[root@node-b ~]# echo b > /proc/sysrq-trigger
```

フェイルオーバーが正常に終了したら，Ctrl-C（SIGINT）で node-a 上の実行中のコマンドを終了させます。

node-a でのコマンドの出力は以下のようなものになります。

```
…
node-b
node-b
node-b
[failover occurs]
node-a
node-a
node-a
…
```

この時点で、LB Health Check リソースの動作の基本的な検証は完了です。必要に応じて追加のテストを実行し、スイッチオーバーやフェイルオーバー時の LB Health Check リソースと保護対象のアプリケーション間の相互作用を検証します。LB Health Check リソースの機能テストが終了したら、node-a と node-b の両方で次のコマンドを実行して、一時的な Apache Web サーバーを削除することができます。

```
# systemctl stop httpd
# rm -f /var/www/html/index.html
# yum remove -y httpd
```

# ロードバランサーヘルスチェックパラメーターのチューニング

ロードバランサーのデフォルトのヘルスチェックパラメーターは、ほとんどの一般的な状況で機能するはずですが、望ましいスイッチオーバー動作を実現するために、パラメーターを調整する必要がある場合があります。ユーザーがこれらのパラメーターを調整する必要があるかもしれない典型的な問題は2つあります。

1. 値が低く設定されすぎているため、ロードバランサーが一時的なリソースの制約やネットワークの中断に対して敏感になりすぎている。
2. 値が高く設定されすぎており、スイッチオーバー時にロードバランサーが新しいリソースホストへのトラフィックのルーティングを開始しても、以前のリソースホストは引き続き正常と判定される。

クラウドの負荷分散環境では、一般的に4つの主要なヘルスチェックパラメーターを調整することができます

- **ヘルスチェック間隔** – ヘルスチェックサーバーがヘルスチェックプローブをバックエンドターゲットVMに送信する頻度
- **タイムアウト** – ヘルスチェックサーバーが、ヘルスプローブが失敗したと見なす前に応答の受信を待機する時間。
- **正常のしきい値** – バックエンドターゲットVMが正常と判定されるために、ヘルスチェックプローブが連続して正常な応答を受け取る必要がある回数。
- **異常のしきい値** – バックエンドターゲットVMが異常と判定されるために必要な、連続したヘルスチェックのプローブの失敗回数。

これらのパラメーターの詳細については、 Azure Load Balancer の正常性プローブ および Google Cloud – ヘルスチェックの概要 を参照してください。

これらのパラメーターから、次の値を導出することもできます。

- **正常と判定されるまでの合計時間 = ヘルスチェック間隔 × (正常のしきい値 – 1)** – サーバーとヘルスプローブサーバー間のネットワークレイテンシーが低いと仮定した場合の、正常なサーバーの最初のヘルスチェックプローブ後、ロードバランサーによって正常と判定されるまでの合計時間。
- **異常と判定されるまでの合計時間 = タイムアウト × 異常のしきい値** – 障害が発生したサーバーの最初のヘルスチェックプローブから、ロードバランサーによって異常と判定されるまでの合計時間。

これらのパラメーターの正確な値は各ユーザーの特定の環境によって異なりますが、いくつかの一般的なガイドラインを以下に示します。

タイムアウトまたは異常のしきい値のチューニングが低すぎると、一時的なVMリソースの制約または一時的なネットワークの問題に対して回復力のないロードバランサーになる可能性があります。たとえば、 **タイムアウト = 1秒**や **異常のしきい値 = 1回の失敗** などの極端な値を設定すると、ネットワークが数秒間応答しなくなった場合でも、バックエンドVMが異常と判定されます。これはクラウド環境では珍しいことではありません。ロードバランサーの構成が一時的でマイナーな問題に対してより回復力があるように、タイムアウトを適切な値（例：5秒）にしておくことをお勧めします。

ヘルスチェック間隔と正常のしきい値の組み合わせが高すぎると、ロードバランサーがスイッチオーバーまたはフェイルオーバー後に新しいリソースホストへのトラフィックのルーティングを開始するのに必要

以上に長い時間がかかり、アプリケーションの回復に時間がかかる可能性があります。ほとんどの場合、**正常のしきい値 = 2 回連続成功** を設定するのが適切です。

タイムアウトと異常のしきい値の組み合わせが高すぎると、スイッチオーバー後にロードバランサーが前のリソースホストノードを異常と判断するのに十分な速さで反応せず、ラウンドロビン形式でアクティブサーバーとスタンバイサーバーの両方にロードバランサーのトラフィックがルーティングされる時間が発生することがあります。このような状況を回避するために、リソース階層を何度かスイッチオーバーしてデータを収集し、前のホストで LB Health Check リソースが Out-of-Service になってから新しいホストで In-Service になるまでの最短時間を決定することが推奨されます。
クラスタサーバー間で時刻が同期していると仮定すると、各サーバーの /var/log/lifekeeper.log を調べ、前のホストでの LB Health Check **remove** スクリプトの終了から新しいホストでのLB Health Check **restore** スクリプトの終了までの時間を決定することによって、この時間を求めることができます。この時間（秒）を **Minimum LB Health Check Switchover Time** （最小 LB Health Check スイッチオーバー時間）と呼ぶことにします。そこで、ロードバランサーのヘルスチェックパラメーターを以下のように設定することを推奨します。

> **異常と判定されるまでの合計時間 < 最小 LB Health Checkスイッチオーバー時間 + 正常と判定されるまでの合計時間**
>
> タイムアウトがヘルスチェック間隔と同じになるように選択されている一般的な状況（例えば、両方とも5秒に設定されている）では、推奨は次のようになります。
>
> **異常のしきい値 < ( 最小 LB Health Checkスイッチオーバー時間 / タイムアウト ) + 正常のしきい値 – 1**

例として、ロードバランサーのヘルスチェックパラメーターを、 **ヘルスチェック間隔 = タイムアウト = 5秒** および **正常のしきい値 = 2回連続成功** で設定したとします。リソース階層を使用して繰り返しスイッチオーバーテストからデータを収集することにより、経験的に、 **最小 LB Health Check スイッチオーバー時間 = 20秒** であることがわかります。上記の推奨事項を参考に **異常のしきい値** を選択すると、以下のようになります。

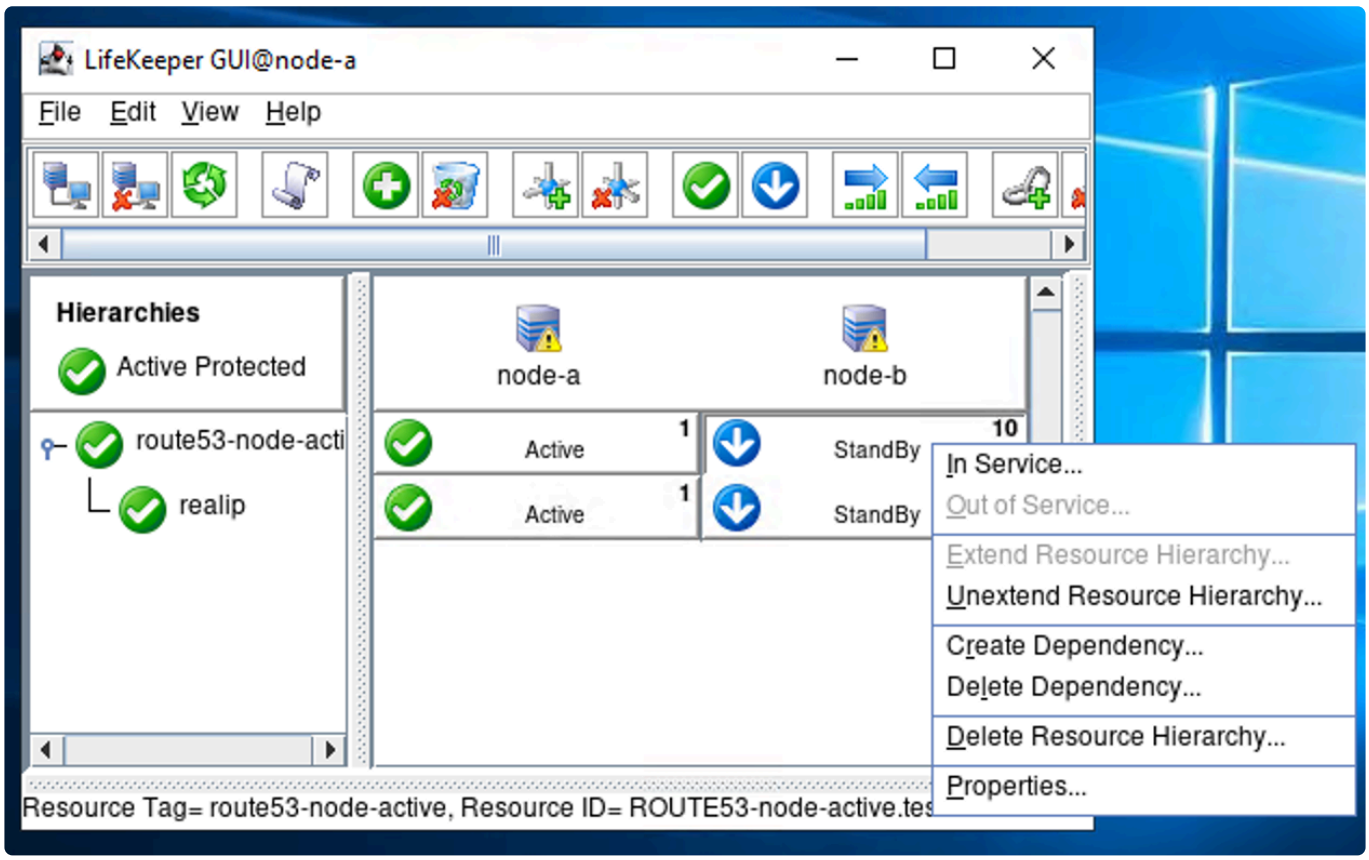
**異常のしきい値** < (20 / 5) + 2 – 1 = 5

このことから、2 ～4 回の連続した失敗を異常のしきい値として選択するのが妥当でしょう。スイッチオーバーテスト中に、ロードバランサーがスイッチオーバー後すぐに前のホストを異常と判定していないことがわかったら、より低い値（2 や3 など）を選択します。もしテスト中に、ロードバランサーが一時的なサーバーやネットワークの問題が原因で、現在のリソースホストを定期的に異常と判定していることがわかった場合は、より高い値（3または4 など）を選択します。この例では、 **異常のしきい値 = 3回連続失敗** を選択するのが妥当かもしれません。

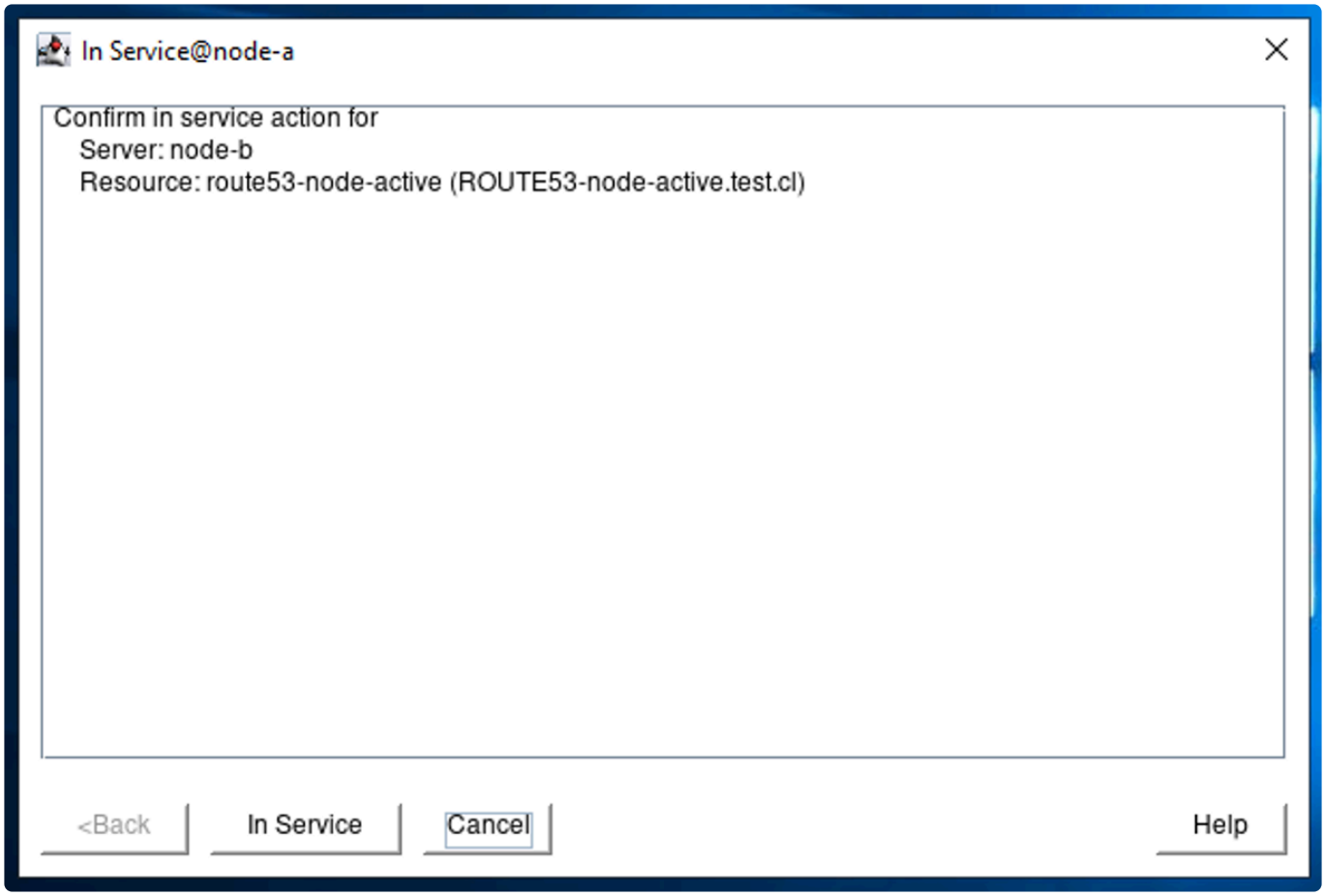
# 11.2.7.3. スタンバイノードに切り替えて、スイッチオーバーが機能していることを確認する

作成時に各クラスターリソースをテストすることをお勧めします。これは、クラスター作成プロセスの早い段階で、構成の誤りやその他の問題を特定するのに役立ちます。

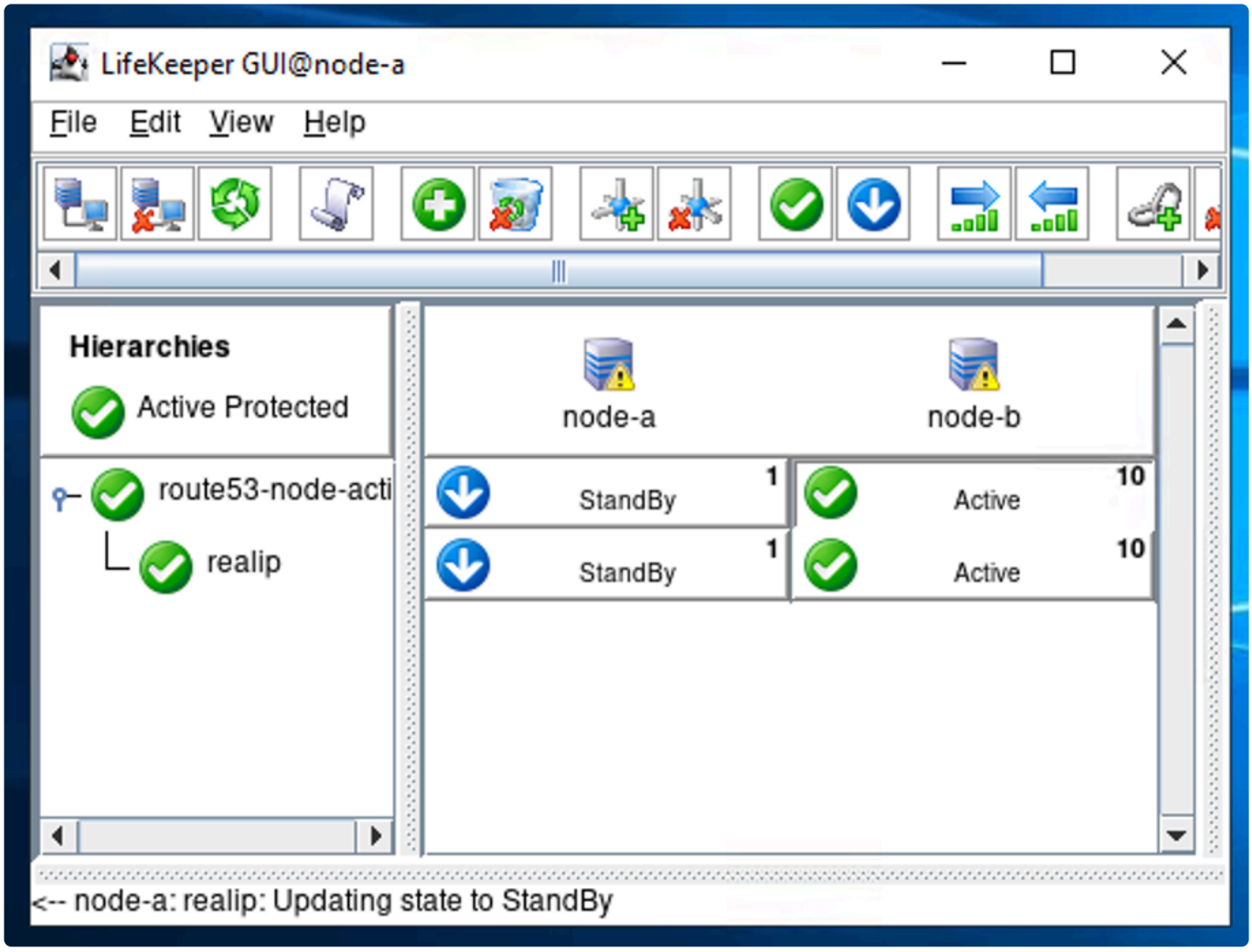
LifeKeeper ユーザーインターフェースで、スタンバイノードのリソースを右クリックし、コンテキストメニューから [In Service...] を選択します。これにより、スタンバイノードへのスイッチオーバーが強制されます。




スイッチオーバーが確認されると、LifeKeeper はリソース（依存リソースを含む）をスタンバイノードにスイッチオーバーします。




スイッチオーバーが正常に完了すると、node-b がアクティブノードとしてリストされます。




スイッチオーバーがうまくいったことを確認したら、node-a にスイッチバックしてください。

# 11.2.7.4. ファイルシステムのデータレプリケーションを作成する方法

ノード間のデータレプリケーションのしくみ はRAID1 デバイスとして機能する NetRAID デバイスを作成します。

このガイドでは、例として次のパラメータを使用しています。DataKeeper がインストールされているローカル環境に応じて、これらのパラメーターを置き換えてください。

| 項目 | 値 |
| --- | --- |
| Disk Device Name for Replication | /dev/xvdb (the second storage device attached)<br><br>**注記:** ストレージデバイス名は、環境によって異なる場合があります。 |
| Mount point | /datakeeper |

1. レプリケーション用のディスクを準備するには、次のいずれかのトピックを参照してください。

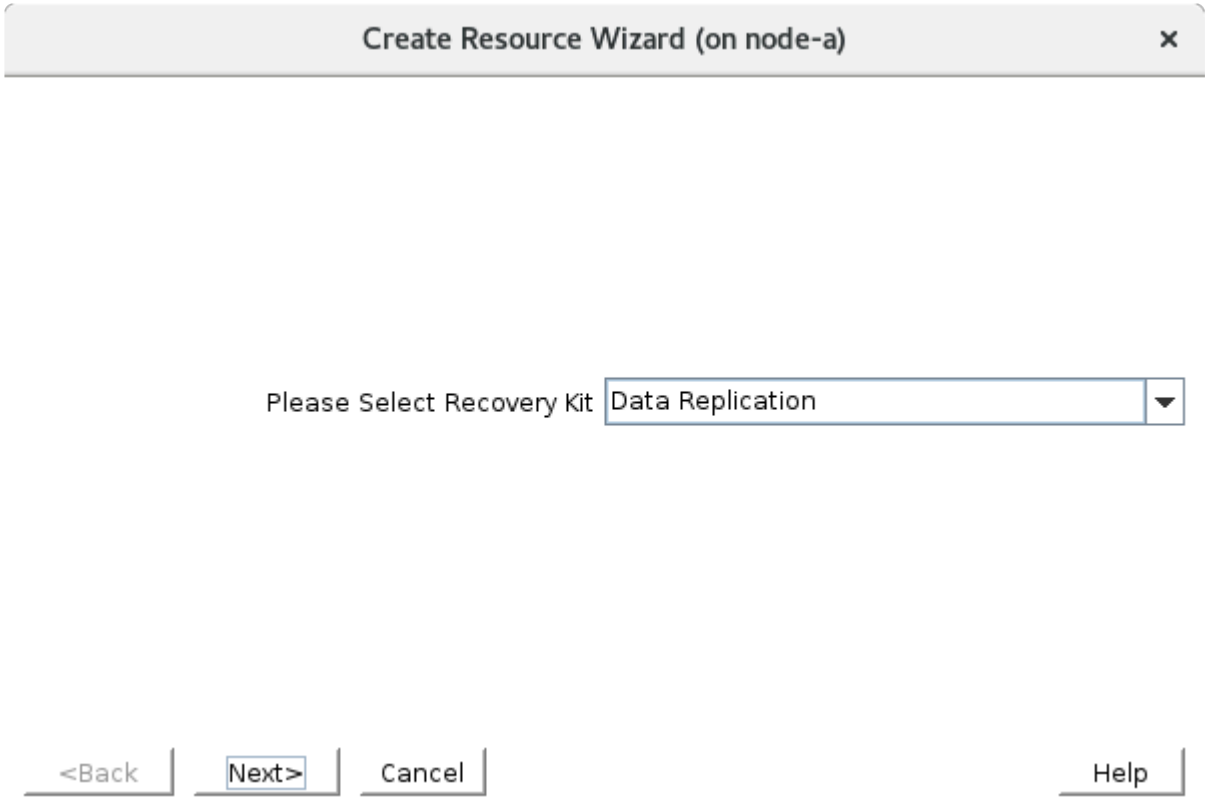
- AWS でレプリケーション用にディスクを準備する方法
- Azure でレプリケーション用にディスクを準備する方法
- Google Cloud でレプリケーション用にディスクを準備する方法

2. ディスクをプライマリノード（node-a）にマウントします。

マウントするストレージデバイスの名前 (`/dev/xvdb1` または `/dev/sdc1`)を必ず確認してください。

```
# mkdir /datakeeper
# mount /dev/xvdb1 /datakeeper
```

3. LifeKeeper ユーザーインターフェースで、新しいリソースを定義します。 を選択して、「Create Resource Wizard @ node-a」を開始します。Recovery Kitに「Data Replication」を選択します。

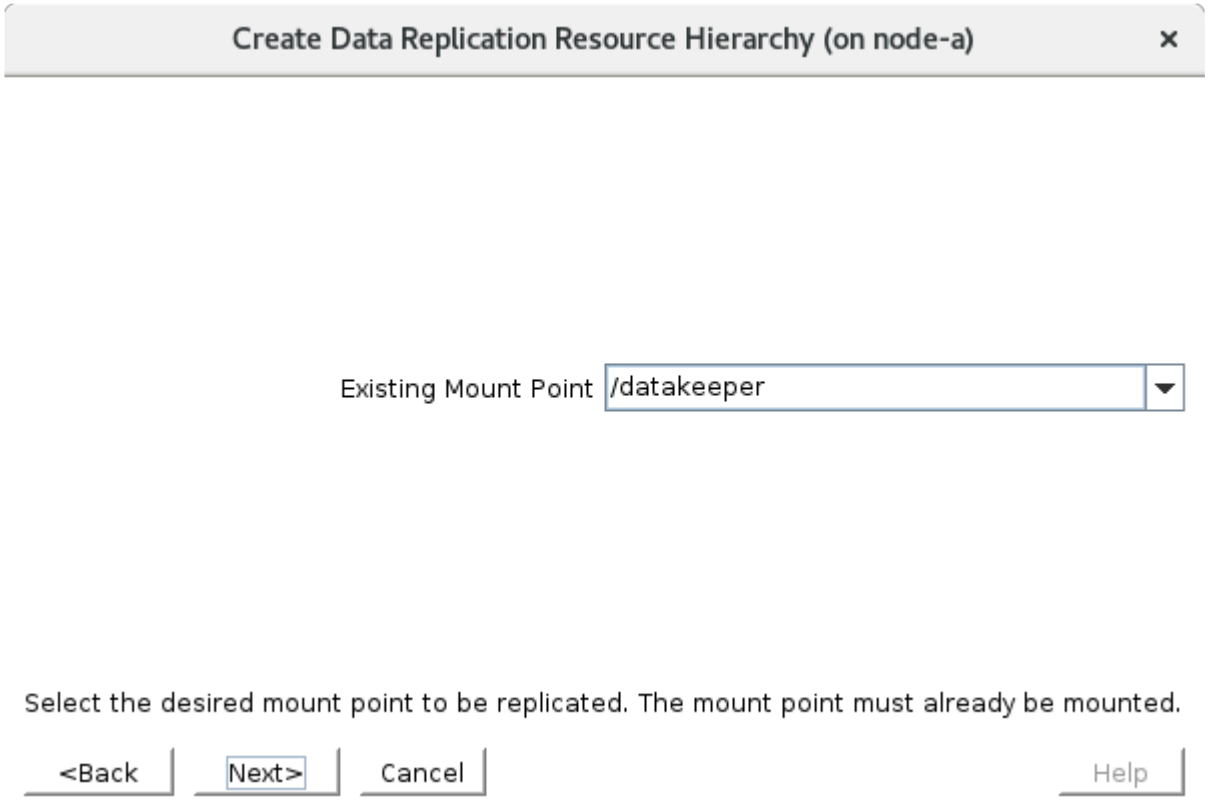



4. 次の値を入力します。デフォルト値を選択します。

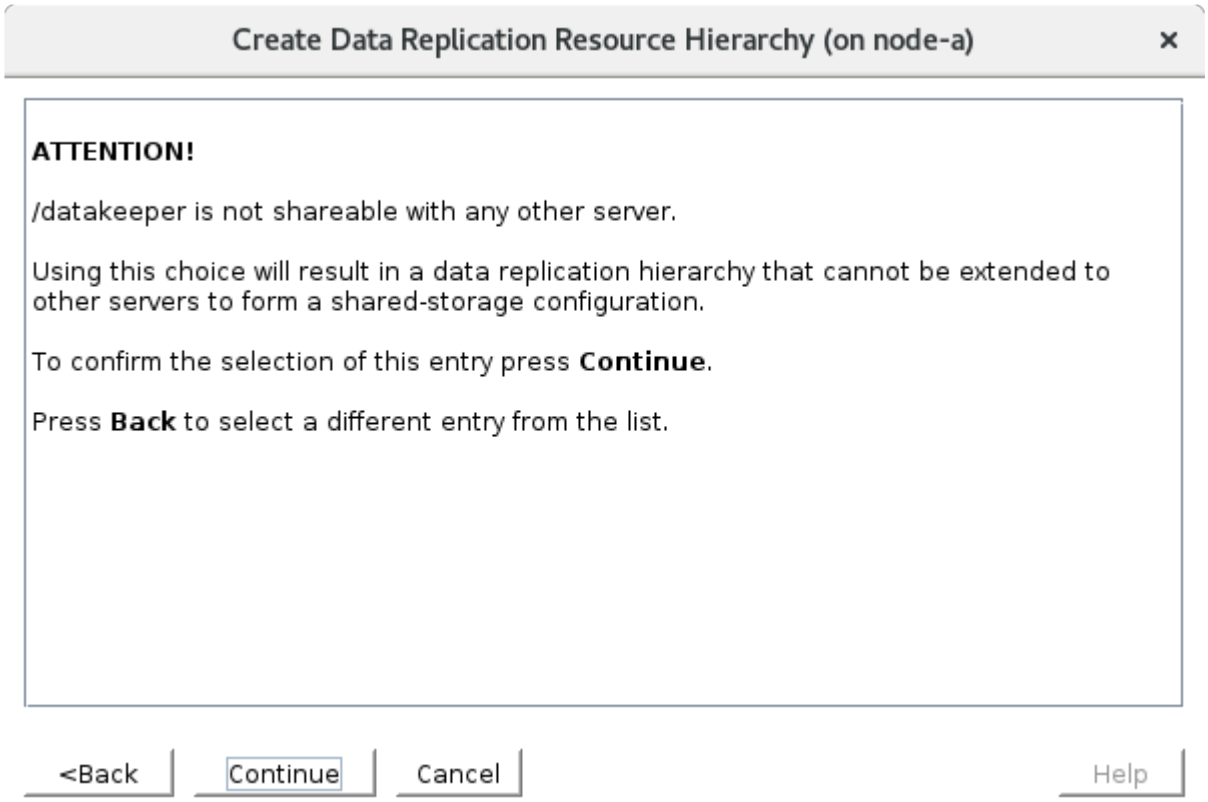
> **注記:** チェックマーク ( ) が付いている項目については、デフォルト値を確認し、推奨の値を使用してください。

| フィールド | 値 |
|---|---|
| Switchback Type | intelligent |
| Server | node-a |
| Hierarchy Type | Replicate Existing Filesystem |

5. マウントポイント /datakeeperを選択します。ウィザードはすでにシステムをスキャンしており、このフィールドの候補として /datakeeper が表示されます。




6. ウィザードで、選択の確認を求められます。[Continue] を選択して次に進みます。

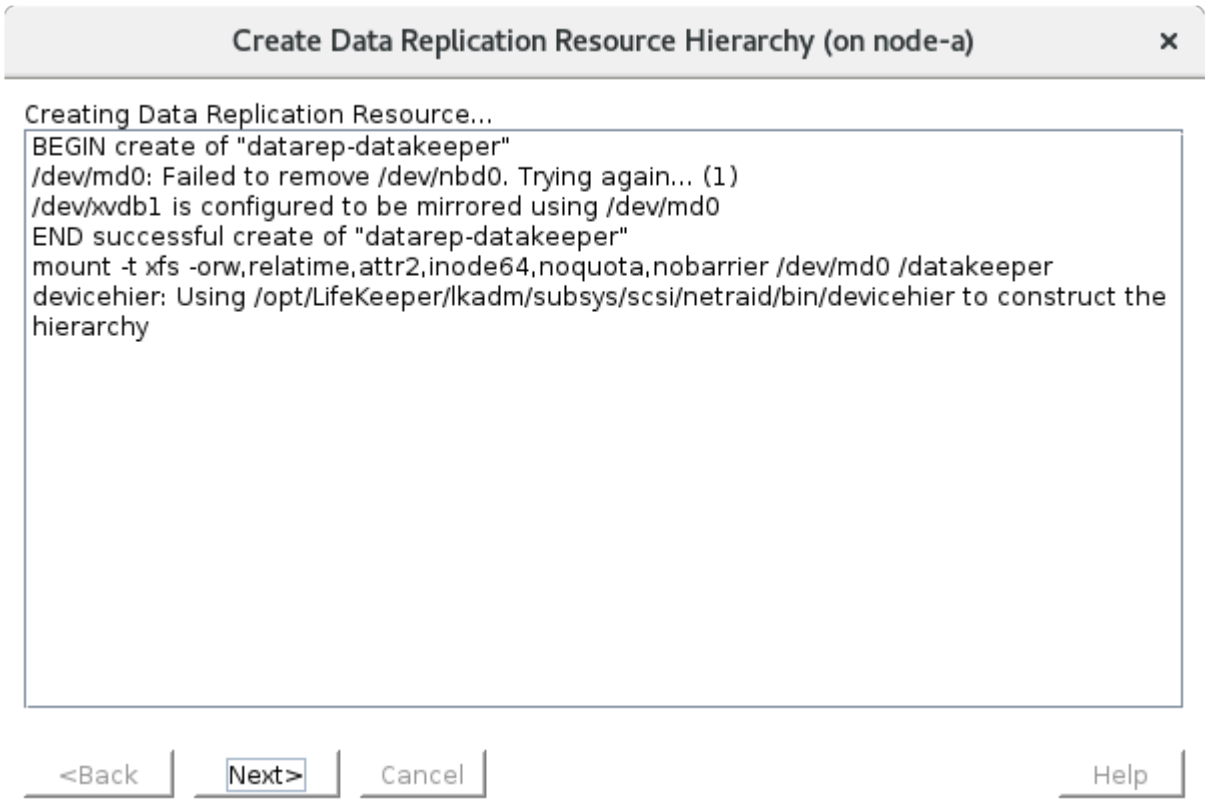



7. フィールドに次の値を入力します。デフォルト値を選択します。

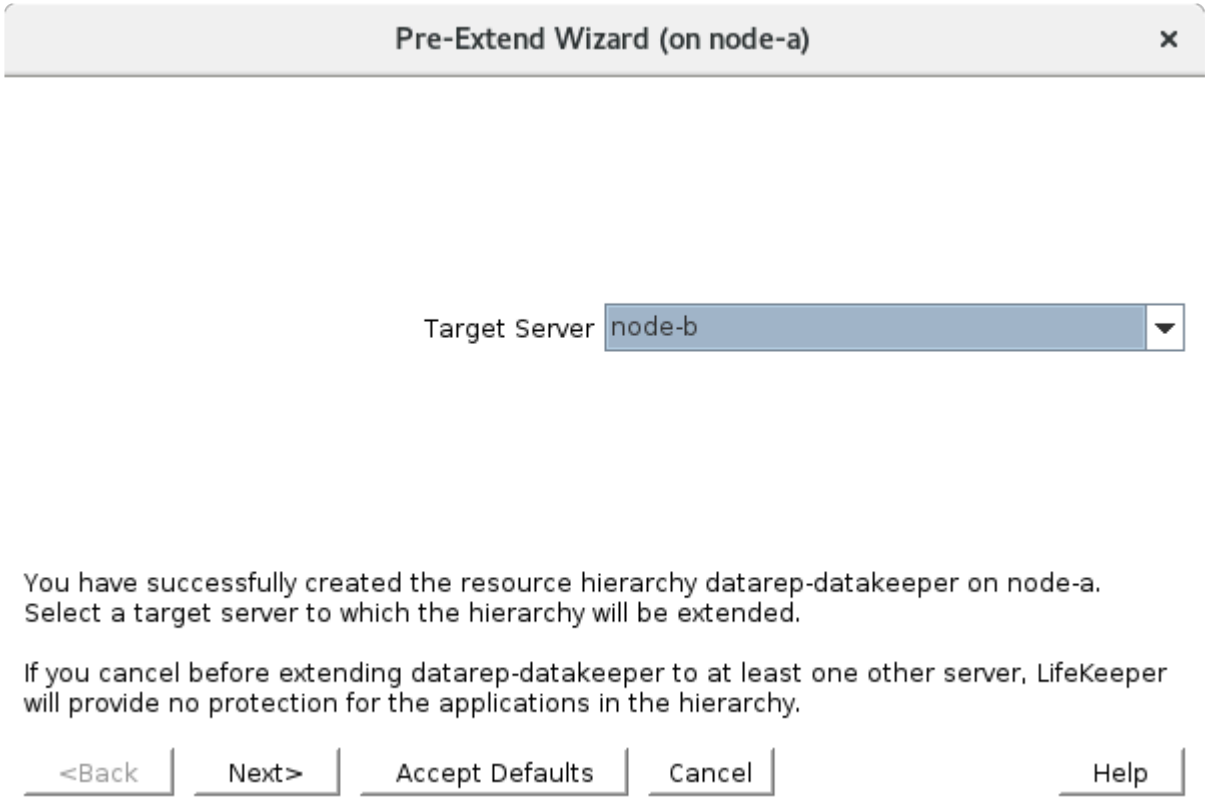
| フィールド | 値 |
|---|---|

| | |
|---|---|
| Data Replication Resource Tag | datarep-datakeeper |
| File System Resource Tag | /datakeeper |
| Bitmap File | /opt/LifeKeeper/bitmap__datakeeper |
| Enable Asynchronous Replication? | no |

8. 「Create Data Replication Resource Hierarchy」ウィザードにこれらの値が表示されます。[Next] を選択して続行します。



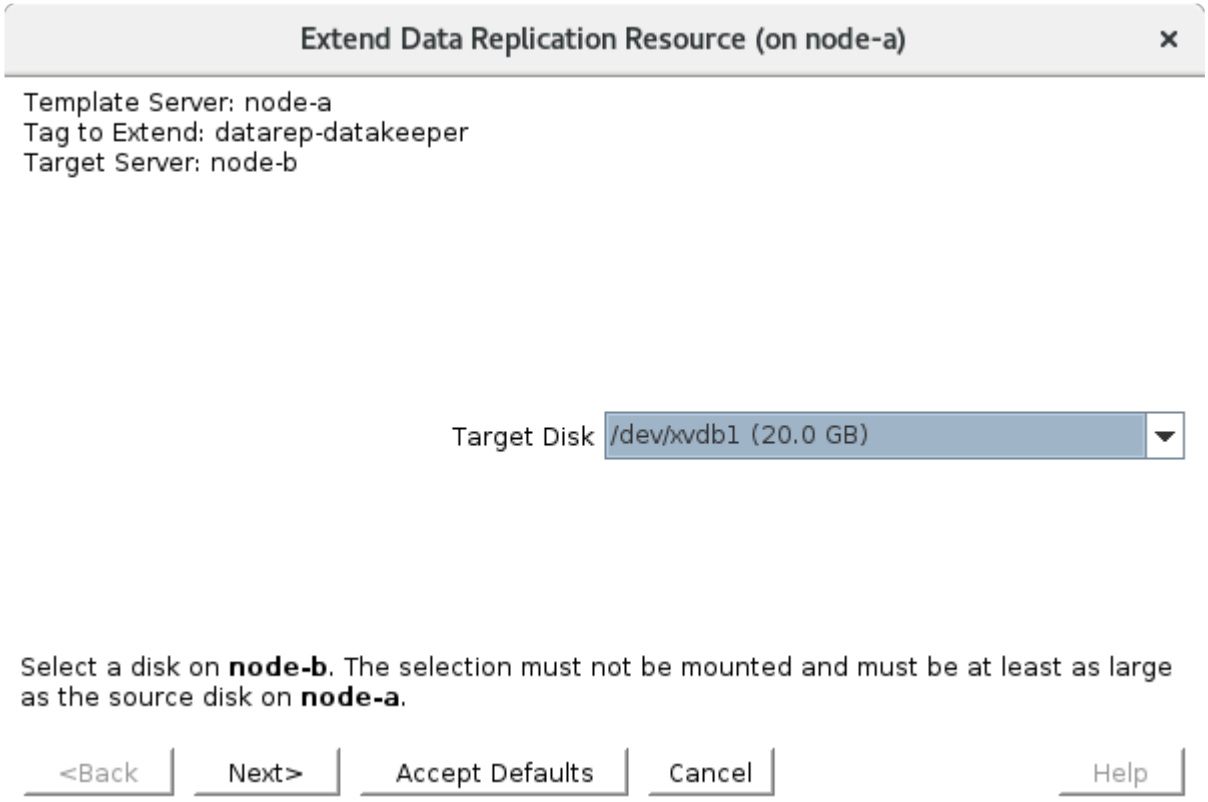

9. 次は「Pre-Extend Wizard」です。デフォルト値を選択します。




| フィールド | 値 |
|---|---|
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |

「Pre-Extend Wizard」のチェックに合格したら、[Next] を選択して続行します。

10. 「Extend gen/filesys Resource Hierarchy」ウィザードでは、まずスタンバイノードとしてディスクを選択します。ドロップダウンリストからディスクを選択します。

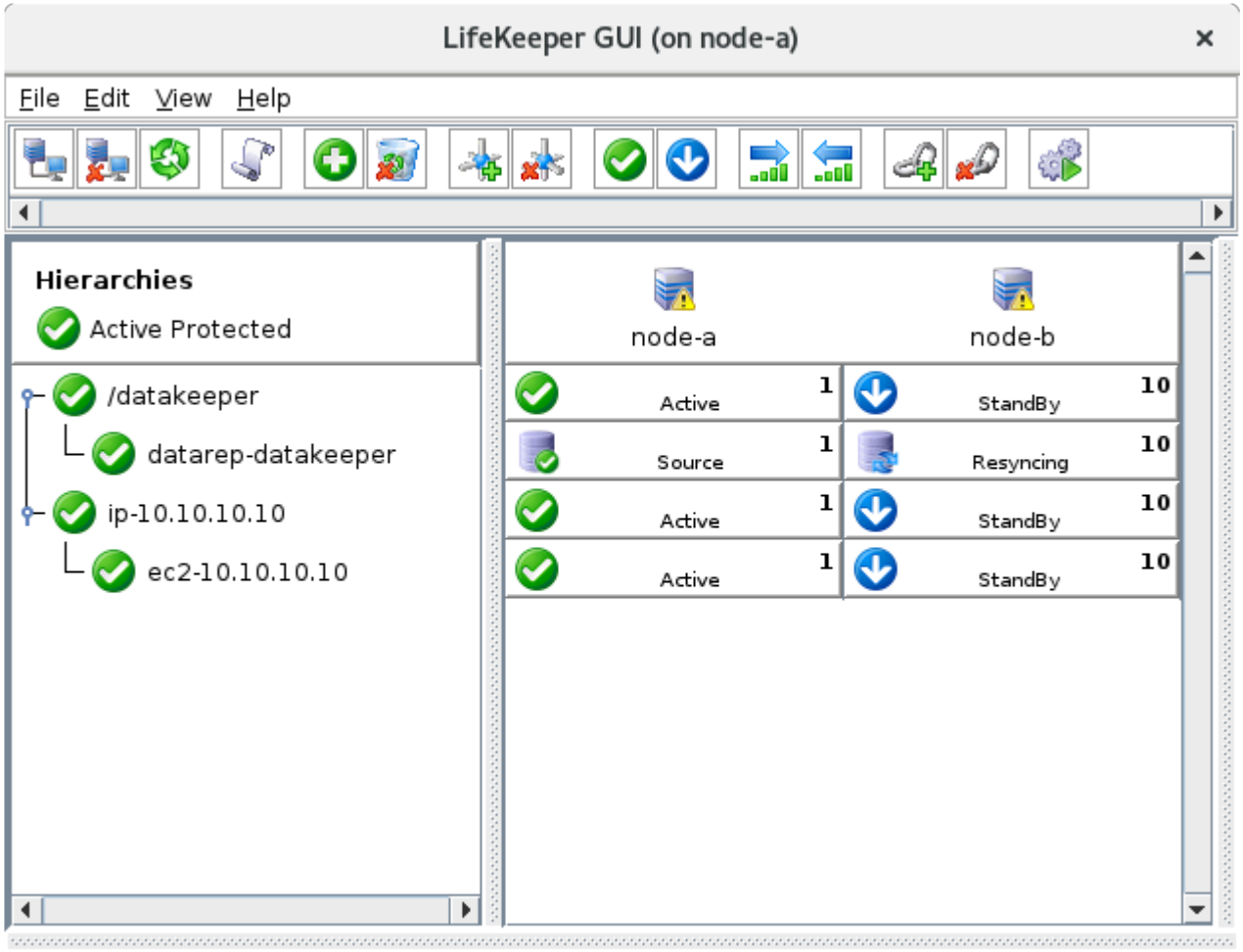



11. 次のページで、デフォルト値を選択します。

| フィールド | 値 |
|---|---|
| Mount point | /datakeeper |
| Root Tag | /datakeeper |
| Target Disk | /dev/xvdb1 (20G) |
| Data Replication Resource Tag | datarep-datakeeper |
| Bitmap File | /opt/LifeKeeper/bitmap__datakeeper |
| Replication Path | 10.20.1.10/10.20.2.10 |

「Extend Wizard」が正常に完了したら、[Finish] を選択します。

これで、データを複製するようにディスクが構成されました。以下に示すように、データの完全な初期同期が必要です。データが同期されると、ラベルが「Target」に変わります。


LifeKeeper GUI (on node-a)
File Edit View Help
Hierarchies
Active Protected
/datakeeper
datarep-datakeeper
ip-10.10.10.10
ec2-10.10.10.10
node-a
node-b
Active 1
StandBy 10
Source 1
Resyncing 10
Active 1
StandBy 10
Active 1
StandBy 10

# 11.2.7.4.1. AWS でレプリケーション用にディスクを準備する方法

次の手順は、各ノードで実行する必要があります。

DataKeeper を使用してディスクのレプリケーションを設定する前に、ディスクをレプリケーション用に準備する必要があります。このチュートリアルでは、各ノードに同じサイズの空のディスクを用意することを想定しています。

このガイドでは、例として次のパラメーターを使用しています。DataKeeper がインストールされているローカル環境に基づいて、これらのパラメーターを置き換える必要があります。

| 項目 | 値 |
|---|---|
| Disk Device Name for Replication | /dev/xvdb (as the second disk) |
| Mount point | /datakeeper |

## /datakeeper ディレクトリーを作成する

```
# mkdir /datakeeper
```

注記：mkdir /datakeeperを実行した後、 所有権を oracle.oinstall に変更 して、Oracle セットアップが実際にデータベースファイルを作成できるようにします。 次のコマンドを実行します ：sudo chown oracle.oinstall /datakeeper

## node-a で使用可能なディスクを確認する

lsblkを使用してマウントするディスクを確認します。以下のスクリーンショットからわかる通り、ストレージデバイス xvdb はまだインスタンスにマウントされていません。

```
[root@node-a media]# lsblk
NAME    MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
xvda    202:0    0    10G  0 disk
├─xvda1 202:1    0     1M  0 part
└─xvda2 202:2    0    10G  0 part /
xvdb    202:16   0     8G  0 disk
loop0     7:0    0 376.3M  0 loop /media
```

# gdisk を使用してパーティションを作成する

gdisk /dev/xvdb を使用して、このディスクにパーティションを作成します。

以下のコマンドを、この順序で入力します。

- n で新しいパーティションを作成します
- <Enter> でデフォルトを選択します（パーティション番号）
- <Enter> でデフォルトを選択します（最初のセクター）
- <Enter> でデフォルトを選択します（最後のセクター）
- <Enter> でデフォルトを選択します（HEXコード）
- w で変更を書き留めます
- y で操作を確定します

```
# gdisk /dev/xvdb
GPT fdisk (gdisk) version 0.8.10

Partition table scan:
  MBR: not present
  BSD: not present
  APM: not present
  GPT: not present

Creating new GPT entries.

Command (? for help): n
Partition number (1-128, default 1):
First sector (34-41943006, default = 2048) or {+-}size{KMGTP}:
Last sector (2048-41943006, default = 41943006) or {+-}size{KMGTP}:
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300):
Changed type of partition to 'Linux filesystem'

Command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING
PARTITIONS!!

Do you want to proceed? (Y/N): Y
OK; writing new GUID partition table (GPT) to /dev/xvdb.
The operation has completed successfully.
```

## mkfs を使用してパーティションにファイルシステムを作成する

```
# mkfs -t xfs /dev/xvdb1
meta-data=/dev/xvdb1             isize=512    agcount=4, agsize=1310655 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=0, sparse=0
data     =                       bsize=4096   blocks=5242619, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

# 11.2.7.4.2. Azure でレプリケーション用にディスクを準備する方法

Azureで作業している場合、2 番目のディスクをインスタンスに接続する方法を、以下のドキュメントを確認してください。

https://docs.microsoft.com/en-us/azure/virtual-machines/linux/attach-disk-portal

## /datakeeper ディレクトリーを作成する

```
# mkdir /datakeeper
```

注記: mkdir /datakeeperを実行した後、所有権を **oracle.oinstallに変更** して、Oracleセットアップが実際にデータベースファイルを作成できるようにします。 **次のコマンドを実行します：**sudo chown oracle.oinstall /datakeeper

## node-a で使用可能なディスクを確認する

lsblk を使用してマウントするディスクを確認します。以下のスクリーンショットからわかる通り、ストレージデバイス sdc はまだインスタンスにマウントされていません。

```
[root@node-a ~]# lsblk
NAME              MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                 8:0    0   64G  0 disk
├─sda1              8:1    0  500M  0 part /boot/efi
├─sda2              8:2    0  500M  0 part /boot
├─sda3              8:3    0    2M  0 part
└─sda4              8:4    0   63G  0 part
  ├─rootvg-tmplv  253:0    0    2G  0 lvm  /tmp
  ├─rootvg-usrlv  253:1    0   10G  0 lvm  /usr
  ├─rootvg-optlv  253:2    0    2G  0 lvm  /opt
  ├─rootvg-homelv 253:3    0    1G  0 lvm  /home
  ├─rootvg-varlv  253:4    0    8G  0 lvm  /var
  └─rootvg-rootlv 253:5    0    2G  0 lvm  /
sdb                 8:16   0    4G  0 disk
└─sdb1              8:17   0    4G  0 part /mnt/resource
sdc                 8:32   0    8G  0 disk
```

上記の Azure のドキュメントに従って、インスタンスにマウントしてください。

parted を使用してパーティションを準備してから、 mkfs を使用してファイルシステムを準備します。

```
[root@node-a ~]# parted /dev/sdc --script mklabel gpt mkpart xfspart xfs 0% 100%
[root@node-a ~]# mkfs.xfs /dev/sdc1
meta-data=/dev/sdc1              isize=512    agcount=4, agsize=524160 blks
         =                       sectsz=4096  attr=2, projid32bit=1
         =                       crc=1        finobt=0, sparse=0
data     =                       bsize=4096   blocks=2096640, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=4096  sunit=1 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
[root@node-a ~]# partprobe /dev/sdc1
```

lsblk を再度実行すると、 /dev/sdc1 をマウントできるようになります。

```
[root@node-a ~]# lsblk
NAME              MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                 8:0    0   64G  0 disk
├─sda1              8:1    0  500M  0 part /boot/efi
├─sda2              8:2    0  500M  0 part /boot
├─sda3              8:3    0    2M  0 part
└─sda4              8:4    0   63G  0 part
  ├─rootvg-tmplv  253:0    0    2G  0 lvm  /tmp
  ├─rootvg-usrlv  253:1    0   10G  0 lvm  /usr
  ├─rootvg-optlv  253:2    0    2G  0 lvm  /opt
  ├─rootvg-homelv 253:3    0    1G  0 lvm  /home
  ├─rootvg-varlv  253:4    0    8G  0 lvm  /var
  └─rootvg-rootlv 253:5    0    2G  0 lvm  /
sdb                 8:16   0    4G  0 disk
└─sdb1              8:17   0    4G  0 part /mnt/resource
sdc                 8:32   0    8G  0 disk
└─sdc1              8:33   0    8G  0 part
```

これで、 LifeKeeper / DataKeeper を使用してデータレプリケーションリソースを作成する準備が整いました。

# 11.2.7.4.3. Google Cloudでレプリケーション用にディスクを準備する方法

## /datakeeper ディレクトリーを作成する

```
# mkdir /datakeeper
```

注記: mkdir /datakeeperを実行した後、 所有権を **oracle.oinstall** に変更 して、Oracle セットアップが実際にデータベースファイルを作成できるようにします。次のコマンドを実行します：sudo chown oracle.oinstall /datakeeper

## node-a で使用可能なディスクを確認する

lsblk を使用してマウントするディスクを確認します。以下のスクリーンショットからわかる通り、ストレージデバイス sdc はまだインスタンスにマウントされていません。

```
[root@node-a ~]# lsblk
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda      8:0    0    20G  0 disk
├─sda1   8:1    0   200M  0 part /boot/efi
└─sda2   8:2    0  19.8G  0 part /
sdb      8:16   0    10G  0 disk
```

parted を使用してパーティションを準備してから、 mkfs を使用してファイルシステムを準備します。

```
[root@node-a ~]# parted /dev/sdb --script mklabel gpt mkpart xfspart xfs 0% 100%
[root@node-a ~]# mkfs.xfs /dev/sdb1
meta-data=/dev/sdb1              isize=512    agcount=4, agsize=655232 blks
         =                       sectsz=4096  attr=2, projid32bit=1
         =                       crc=1        finobt=0, sparse=0
data     =                       bsize=4096   blocks=2620928, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=4096  sunit=1 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
[root@node-a ~]# partprobe /dev/sdb1
```

lsblk を再度実行すると、 /dev/sdc1 をマウントできるようになります。

```
[root@node-a ~]# lsblk
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda      8:0    0    20G  0 disk
├─sda1   8:1    0   200M  0 part /boot/efi
└─sda2   8:2    0  19.8G  0 part /
sdb      8:16   0    10G  0 disk
└─sdb1   8:17   0    10G  0 part
```

これで、 LifeKeeper / DataKeeper を使用してデータレプリケーションリソースを作成する準備が整いました。

# 11.2.7.5. 他のリソース（データベースまたはアプリケーション）を保護する方法

このセクションでは、データベースやアプリケーションなど、さまざまなリソースを作成するための基本的な手順を説明します。（特に断りのない限り）各リソースは独立して作成することができます。

- Oracleリソースの保護
- Quick Service Protectionを使用したMSSQLの保護
- PostgreSQLリソースの保護
- NFSリソースの保護
- SAPリソースの保護
- SAP HANAリソースの保護

このガイドでは、さまざまなパターンを示すために、さまざまなタイプのIPリソースを使用します。以下のような異なるIPリソースタグが見られることがあります。
- ip-10.10.10.10
- ip-10.20.0.10
- ip-10.20.10.100
- realip
（IPリソースの構成方法に基づいて）異なるタグが表示される場合がありますが、これらの違いは各アプリケーションまたはサービス（Oracleなど）の保護方法には影響しません。

# 11.2.7.5.1. Oracleリソースの保護（PDBを除く）

このセクションでは、Oracleリソース（PDBを除く）を保護するための手順を概説します。
Oracleリソースを保護するには、DataKeeperを使用してノード間でデーターを複製する必要があります（共有ストレージまたはSANデバイスが使用されている場合を除く）。
各アイテムの位置の概要は以下の通りです。

| Items | Location |
|---|---|
| Oracle 19c (ORACLE_HOME) | /u01/app/oracle/product/19.3.0/dbhome_1 |
| Database (SID: ORCL) | /datakeeper/oradata/ORCL |

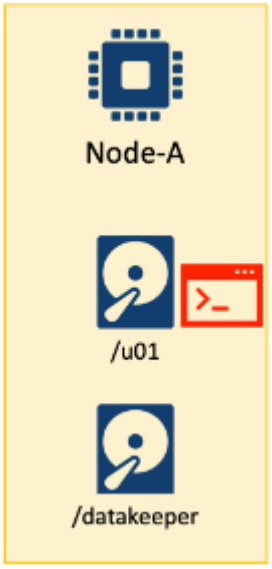
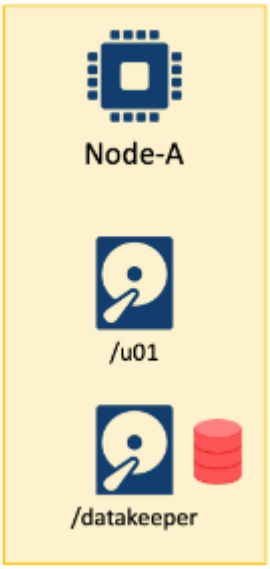
ファイルシステムのデータレプリケーションの作成方法 で説明したように、本ガイドでは `/datakeeper` を使用してノード間でデータのレプリケーションを行います。そのため、`/datakeeper/oradata/ORCL` 上のデータもノード間でレプリケーションされます。

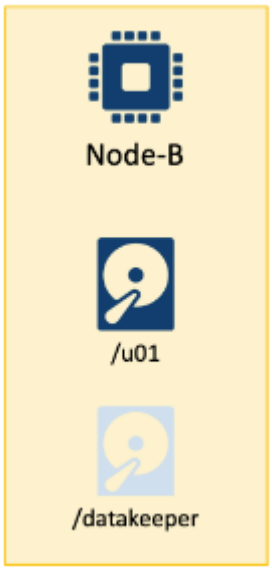
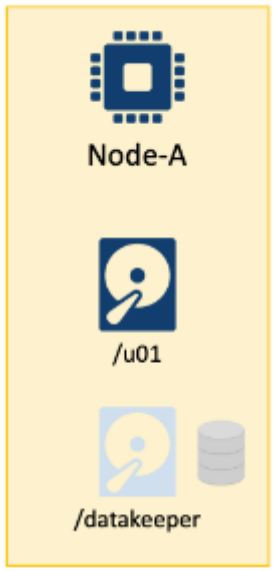
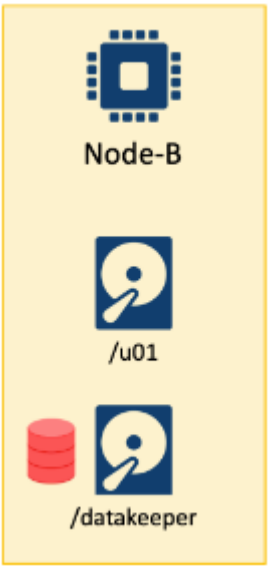
本ガイドでは、以下のデバイスが接続されたコンピューティングリソースを使用しています（Oracleリソースは、評価目的であっても、他のリソースよりも多くのスペースを必要とします）。

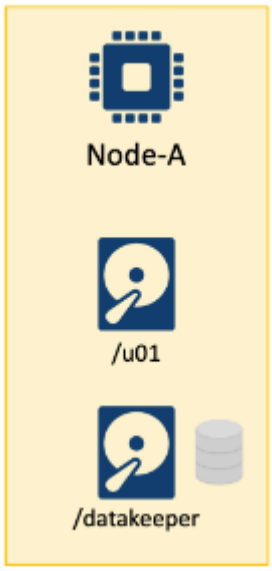
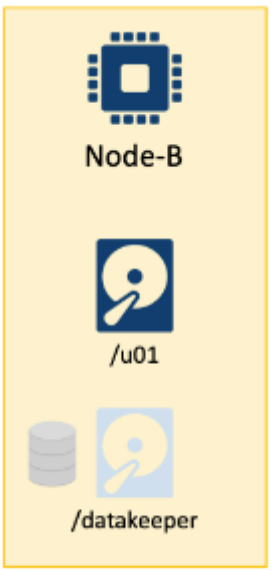
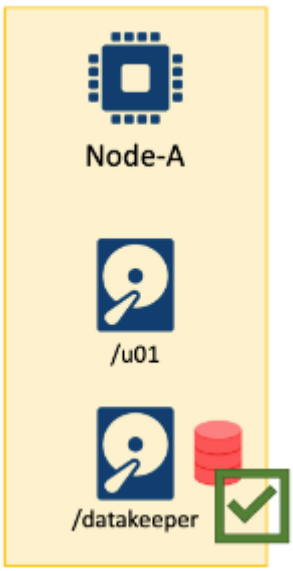
| Resource | Required Size |
|---|---|
| Boot Disk (such as /dev/sda/) | 30 Gb |
| Data Disk (such as /dev/sdb/, configured and mounted at `/datakeeper`) | 20 Gb |
| Memory | 2 Gb |

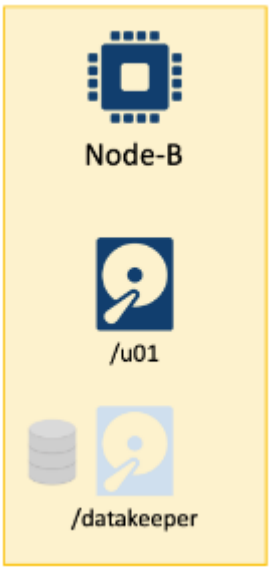
## Oracleリソースの構成

次の表に、Oracleリソースを構成するための一般的な手順の概要を示します。赤い「スタックディスク」の形をしたアイコンは、各ステップの時点でOracleデータベースインスタンスが存在するノードを表します。灰色の「スタックディスク」アイコンは、稼働していないOracleデータベースを表します。
また、灰色の「ストレージ」アイコンは、ストレージ `/datakeeper` がノードで使用できないことを示します。

| | |
|---|---|
| 両方のノードに**Oracle 19c**をインストールする<br><br>システムにOracleをインストールする |  <br> |
| **Node-A**上に**Oracle**データベースを作成する<br><br>データストレージがノードで使用可能かどうかを確認する方法 を確認してください。 データストレージがnode-aで利用可能である必要があります。<br><br>node-aでOracle データーベースを作成する<br><br>node-aでOracleインスタンスを停止する |  |

| | |
|---|---|
| |  |
| **Node-B上にOracleデータベースを作成する**<br><br>node-bへデーターストレージをスイッチオーバーする<br><br>/datakeeper/oradata/ORCL リネームする<br><br>node-bでOracleデーターベース<br><br>node-bのOracle インスタンスを停止する |  <br>  |

| | |
|---|---|
| 両方のノードで**Oracle LISTENER**を構成する<br><br>node-aにデーターストレージをスイッチオーバーする<br><br>両方のノードでOracleリスナーの構成ファイルを更新する |  <br>  |
| **LifeKeeper**を使用して構成を完了する<br><br>• node-aでデータベースとリスナーを起動する<br><br>LifeKeeperでリソースを構成する<br><br>Oracle LISTENERリソースの構成<br><br>Oracleリソースの構成<br><br>Oracleリソースのスイッチオーバーをテストする |  |

| | |
|---|---|
| |  Node-B<br>/u01<br>/datakeeper |

# 11.2.7.5.1.1. Oracleのインストール

両方のノードにOracle 19cをインストールします。

LinuxシステムにOracleをインストールします

- Database Installation Guide for Linux を参照してください。

次の場所にOracle 19cをインストールします。

```
/u01/app/oracle/product/19.3.0/dbhome_1
```

場所は、環境変数 `ORACLE_HOME` で指定する必要があります。Oracleインストーラーを起動したら、インストーラーウィザードページで次の値を指定します。

| ステップ # | 項目 | 値 ( indicates accepting default value) |
|---|---|---|
| 1 | Install Option | Set Up Software Only |
| 2 | Type of database installation | Single Instance database installation |
| 3 | Database Edition | Standard Edition 2 |
| 4 | Oracle Base | `/u01/app/oracle` |
| 5 | Inventory Location | `/u01/app/oraInventory` |
| 6 | Privileges Operating System Groups | |
| 7 | Root Script Execution Configuration | Enter “root” user credential |
| 8 | Perform Prerequisite Checks | Fix if the wizard indicates important issue |
| 9 | Summary | Select “Install” |
| 10 | Install Product | Wait for install |
| 11 | Finish | Install should complete with no errors |

# 11.2.7.5.1.2. Oracleデータベースの作成（PDBを除く）

必ずドキュメントに記載されているノードでこの手順を実行してください。また、そのノードでデータストレージが利用可能かどうかも確認してください。

1. node-aで、以下のようにインストーラを起動します。

```
$ su - oracle
Password:
Last login: Sun Dec 20 04:33:22 UTC 2020 on pts/1
$ cd $ORACLE_HOME
$ ./bin/dbca
```

2. インストーラーが起動したら、次の手順を実行します。

| ステップ # | 項目 | 値 ( indicates accepting default value) |
|---|---|---|
| 1 | Select Operation | Create a database |
| 2 | Creation Mode | Advanced Configuration |
| 3 | Deployment Type | Database Type: Oracle Single Instance Database<br>Template: General Purpose or Transaction Processing |
| 4 | Database Identification | orcl |
| 5 | Storage Option | Storage Type: File System<br>Location: `/datakeeper/oradata` |
| 6 | Fast Recovery Option | Select nothing |
| 7 | Network Configuration | Create a new Listener (Name: LISTENER1) |
| 8 | Configuration Option | Leave as default values |

| | | |
|---|---|---|
| 9 | Management Option | Leave as default values |
| 10 | User Credentials | Select “Use the same administrative password for all accounts” and then Specify Password |
| 11 | Creation Option | Create database |
| 12 | Summary | Confirm the values |
| 13 | Progress Page | Watch for progress |
| 14 | Finish | Confirm the result |

# 11.2.7.5.1.3. Oracleインスタンスを停止する

これらの手順を正しいノードで完了してください。

## データベースをシャットダウンする

データベースをシャットダウンするには、sqlplus を実行し、次のように shutdown immediate コマンドを実行します。

```
[oracle]$ $ORACLE_HOME/bin/sqlplus / AS SYSDBA

SQL*Plus: Release 19.0.0.0.0 - Production on Sun Dec 20 05:50:40 2020
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.


Connected to:
Oracle Database 19c Standard Edition 2 Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
```

## データベースを後で再起動できることを確認する

以下は必須ではありませんが、 startup コマンドでデータベースを再起動してから、 shutdown immediate コマンドでシャットダウンすることをお勧めします。

```
SQL> startup
ORACLE instance started.

Total System Global Area  771748536 bytes
Fixed Size             8901304 bytes
Variable Size                490733568 bytes
Database Buffers     268435456 bytes
Redo Buffers                   3678208 bytes
Database mounted.
Database opened.
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> exit
Disconnected from Oracle Database 19c Standard Edition 2 Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
```

# 11.2.7.5.1.4. /datakeeper/oradata/ORCLの名前を変更する

これまでに、次のタスクが完了しました。

1. Node-Aにデータベースを作成

2. `datarep-datakeeper` リソースをNode-Bにスイッチオーバー

ステップ1では、新しく作成したORCLデータベースに関する情報を `/u01/app/oracle/product/19.3.0/dbhome_1/` に作成しました。
次に、Node-Bに同じORCLデータベースを作成して、ORCLデータベースに関する情報がNode-Bの `/u01/app/oracle/product/19.3.0/dbhome_1/` に格納されるようにします。データベースファイル自体は/datakeeper/oradata/ORCLにあります。

Node-Bで、これらのファイルは次のように表示されます。

```
[ec2-user@node-b ~]$ su - oracle
Password:
Last login: Sun Dec 20 03:58:10 UTC 2020 on pts/0
[oracle@node-b ~]$ ls -al /datakeeper/oradata/ORCL/
total 2477920
drwxr-x--- 2 oracle oinstall       200 Dec 20 04:15 .
drwxrwxr-x 3 oracle oinstall        18 Dec 20 04:11 ..
-rw-r----- 1 oracle oinstall  10600448 Dec 20 05:15 control01.ctl
-rw-r----- 1 oracle oinstall  10600448 Dec 20 05:15 control02.ctl
-rw-r----- 1 oracle oinstall 209715712 Dec 20 05:15 redo01.log
-rw-r----- 1 oracle oinstall 209715712 Dec 20 05:15 redo02.log
-rw-r----- 1 oracle oinstall 209715712 Dec 20 05:15 redo03.log
-rw-r----- 1 oracle oinstall 555753472 Dec 20 05:15 sysaux01.dbf
-rw-r----- 1 oracle oinstall 943726592 Dec 20 05:15 system01.dbf
-rw-r----- 1 oracle oinstall  33562624 Dec 20 04:31 temp01.dbf
-rw-r----- 1 oracle oinstall 356524032 Dec 20 05:15 undotbs01.dbf
-rw-r----- 1 oracle oinstall   5251072 Dec 20 05:15 users01.dbf
```

Node-BのORCLデータベースの作成が正常に完了するようにORCLフォルダの名前を変更します。

```
[oracle@node-b ~]$ cd /datakeeper/oradata/
[oracle@node-b oradata]$ mv ORCL/ ORCL_Backup/
[oracle@node-b oradata]$ ls -al
total 0
drwxrwxr-x 3 oracle oinstall  25 Dec 20 05:26 .
drwxr-xr-x 3 root   root      21 Dec 20 02:24 ..
drwxr-x--- 2 oracle oinstall 200 Dec 20 04:15 ORCL_Backup
```

# 11.2.7.5.1.5. 両方のノードでOracleリスナーの設定ファイルを更新する

以下の手順は、両方のノードで実行する必要があります。

## リスナーのステータスを確認する

次のコマンドを使用して、リスナーの現在のステータスを確認します。

```
$ cd $ORACLE_HOME/bin/
$ ./lsnrctl status LISTENER1
```

## 実行中の場合はリスナーを停止する

```
$ cd $ORACLE_HOME/bin/
$ ./lsnrctl stop LISTENER1
```

## 設定ファイルを編集する

設定ファイルは、以下の場所にあります。
`/u01/app/oracle/product/19.3.0/dbhome_1/network/admin/listener.ora`.

このファイルの内容は、以下に示す例のようになります。

```
# listener.ora Network Configuration File: /u01/app/oracle/product/19.3.0/dbho
me_1/network/admin/listener.ora
# Generated by Oracle configuration tools.
LISTENER1 =
  (DESCRIPTION_LIST =
      (DESCRIPTION =
           (ADDRESS = (PROTOCOL = TCP)(HOST = node-a)(PORT = 1521))
           (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
       )
   )
```

以前に設定されたIPアドレスを確認するには、IPリソースの作成 を参照してください。
また、 SID_LIST_LISTENER1 リソース用の追加セクションが必要です。更新されたファイルは、次の例のようになります。

```
# listener.ora Network Configuration File: /u01/app/oracle/product/19.3.0/dbho
```

```
me_1/network/admin/listener.ora
# Generated by Oracle configuration tools.
SID_LIST_LISTENER1 =
     (SID_LIST =
     (SID_DESC =
             (SID_NAME = orcl)
       )
 )
LISTENER1 =
  (DESCRIPTION_LIST =
      (DESCRIPTION =
           (ADDRESS = (PROTOCOL = TCP)(HOST = 10.10.10.10)(PORT = 1521))
           (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
       )
   )
```

# 11.2.7.5.1.6. Node-aでデータベースとリスナーを起動する

次の手順はNode-aで実行する必要があります。

前の手順で説明したように、データベースを `sqlplus` で起動し、リスナーを`lsnrctl`で起動します。Node-aで両方のプロセスを開始して、対応するリソースが作成されたときにLifeKeeperがこれらのプロセスを認識できるようにします。

## データベースを起動する

```
[oracle@node-a ~]$ $ORACLE_HOME/bin/sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Sun Dec 20 05:58:05 2020
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.

Connected to an idle instance.

SQL> startup
ORACLE instance started.

Total System Global Area  771748536 bytes
Fixed Size          8901304 bytes
Variable Size       490733568 bytes
Database Buffers    268435456 bytes
Redo Buffers          3678208 bytes
Database mounted.
Database opened.
SQL>EXIT
```

## リスナーを起動する

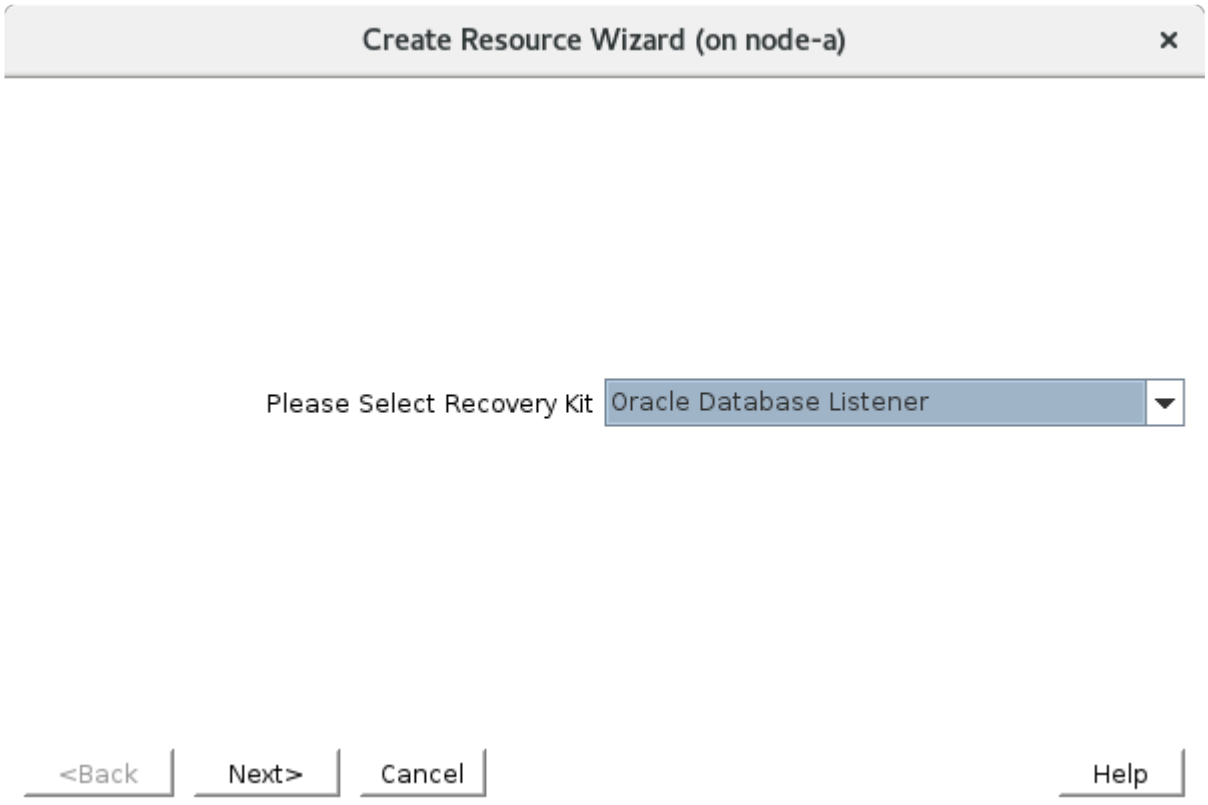
```
$ cd $ORACLE_HOME/bin/
$ ./lsnrctl start LISTENER1
```

# 11.2.7.5.1.7. Oracleリスナーリソースの設定

両方のノードで ”Oracleリスナーの設定ファイルを更新する”:#update-config-file-for-oracle-listener-on-both-nodes で説明しているように、保護するリスナーの名前は `LISTENER1` です。

> **Note**: 9.5.2より前のバージョンを使用している場合は、リスナーのIPアドレスを指定しないでください

1. LifeKeeperユーザーインターフェースで、新しいリソースを定義します。 を選択して、Create Resource Wizard (on node-a) を開始します。Recovery Kitに「Oracle Database Listener」を選択します。

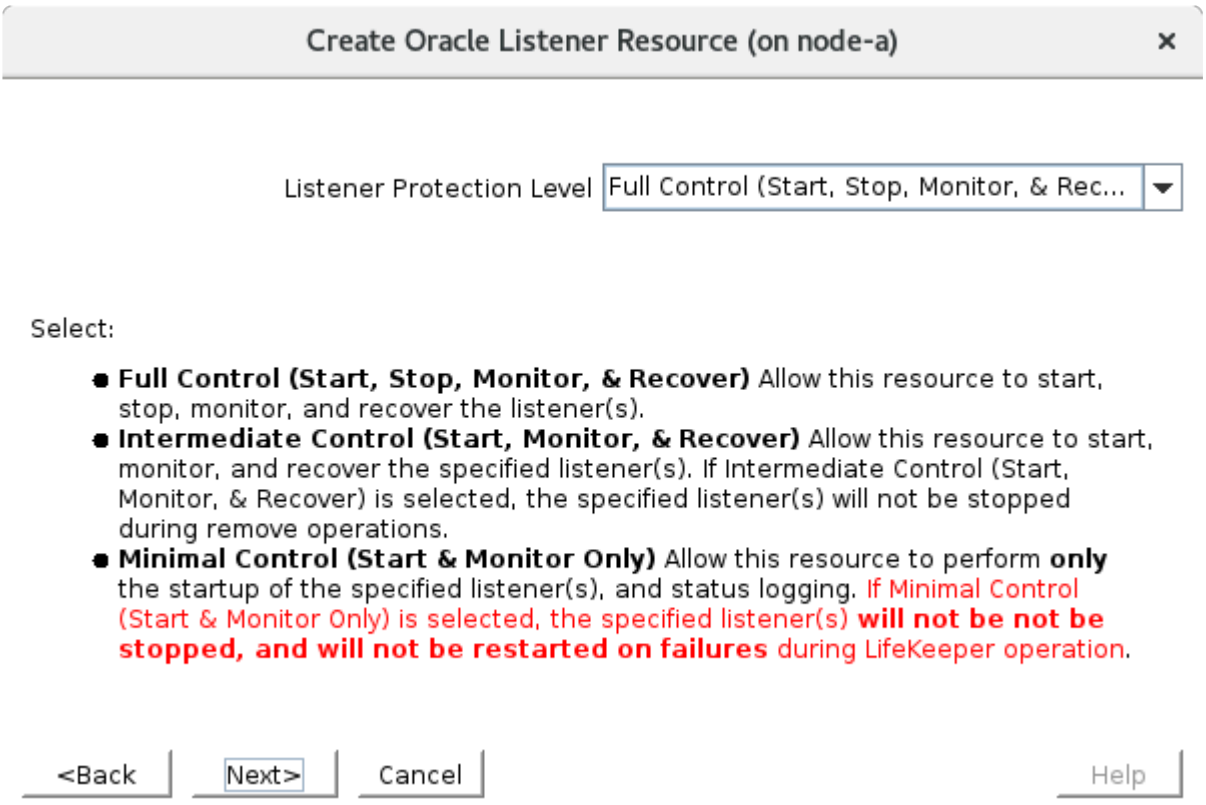



2. 以下の値を入力します。デフォルト値を選択します。

| フィールド | 値 |
|---|---|
| Switchback Type | intelligent |
| Server | node-a |
| Listener Configuration File Path | /u01/app/oracle/product/19.3.0/dbhome_1/network/admin/listener.ora |

| | |
|---|---|
| Listener Name(s) | LISTENER1 |
| Listener Executable(s) | /u01/app/oracle/product/19.3.0/dbhome_1/bin/lsnrctl |

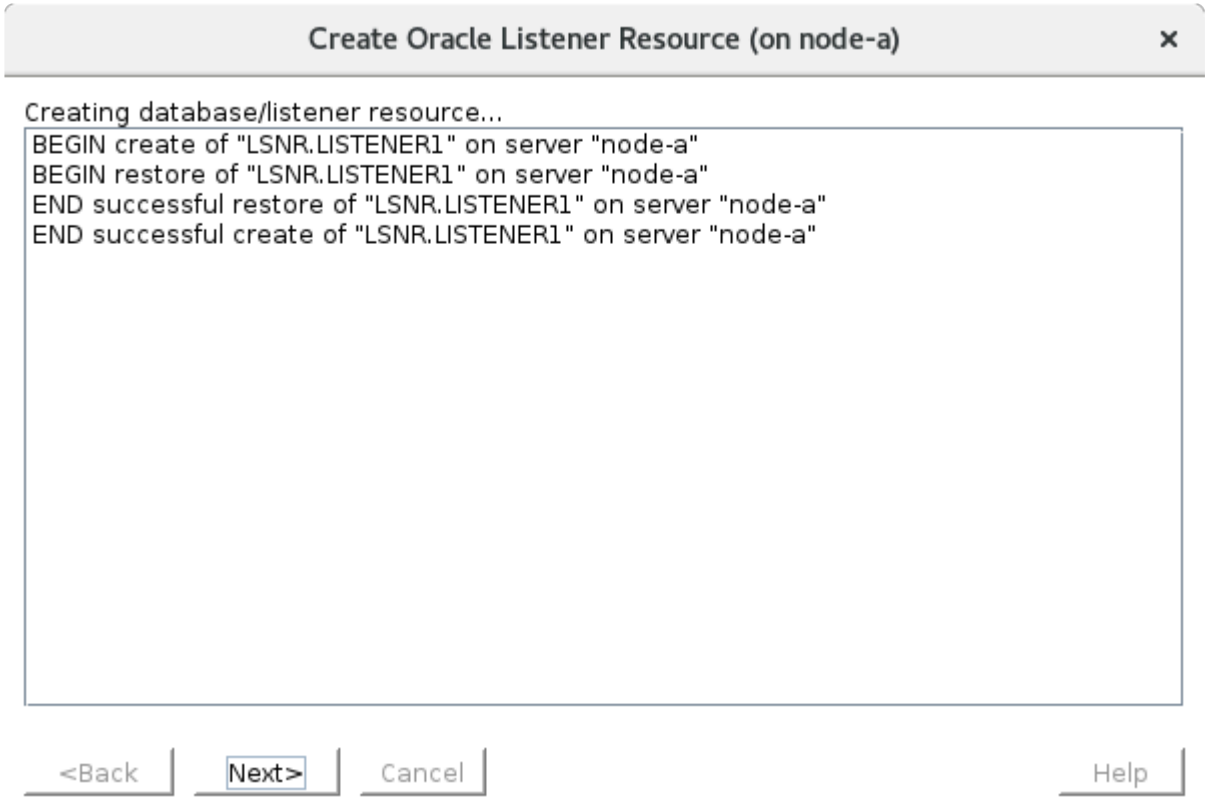
3. Listener Protection Levelを選択します。ウィザードの推奨に従って「Full Control」を選択します。



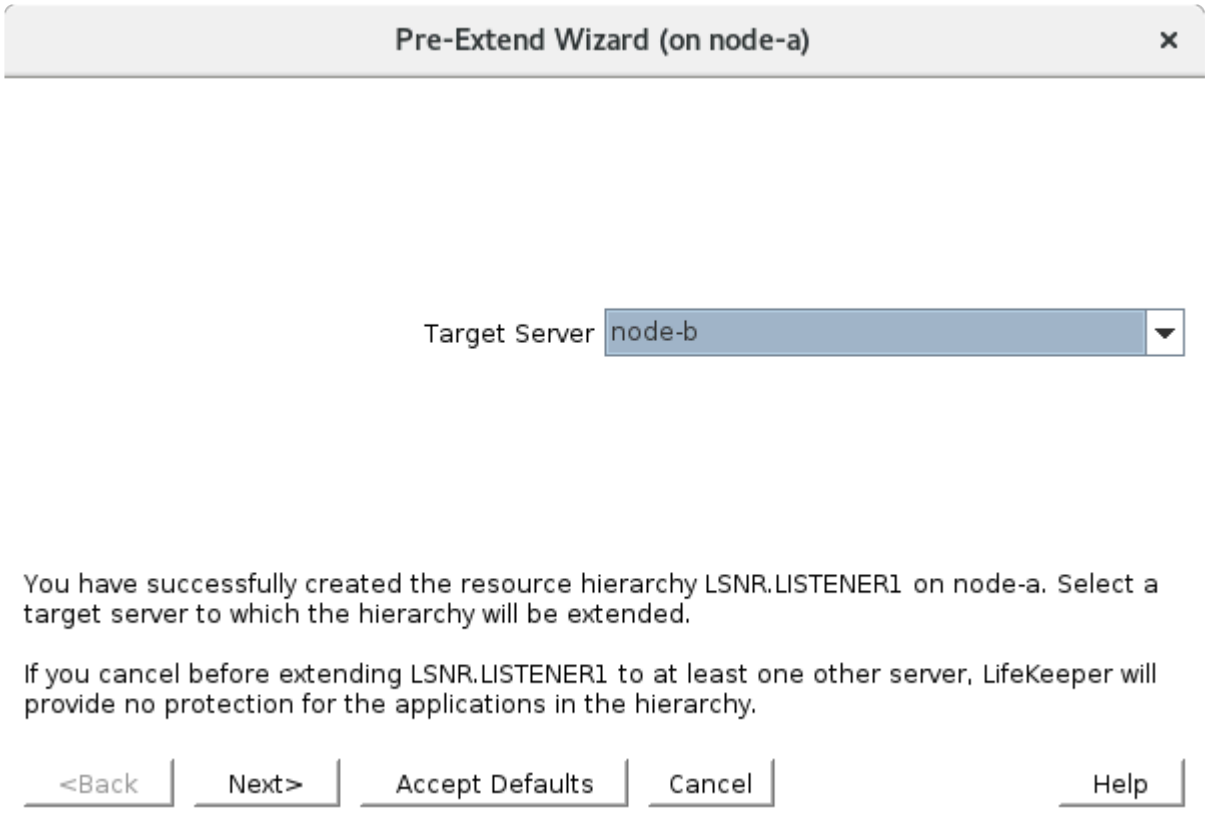

4. 以下の値を入力します。デフォルト値を選択します。

| フィールド | 値 |
|---|---|
| Listener Recovery Level | Standard (On) |
| IP address Name(s) | ip-10.10.10.10 |
| Listener Tag | LSNR.LISTENER1 |

5. ウィザードは、指定されたすべての値を確認します。「Next >」を選択して続行します。



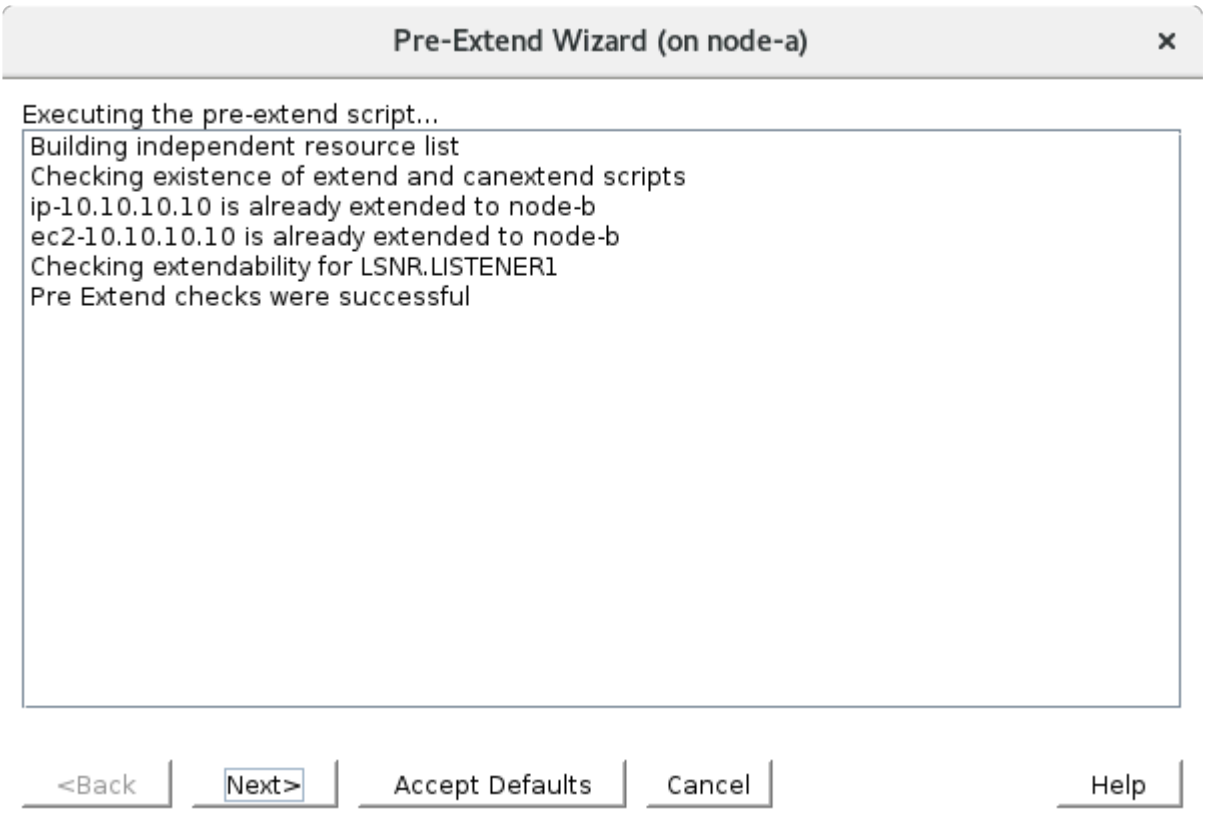

6. 次のステップは「Pre-Extend Wizard」です。デフォルト値を選択します。



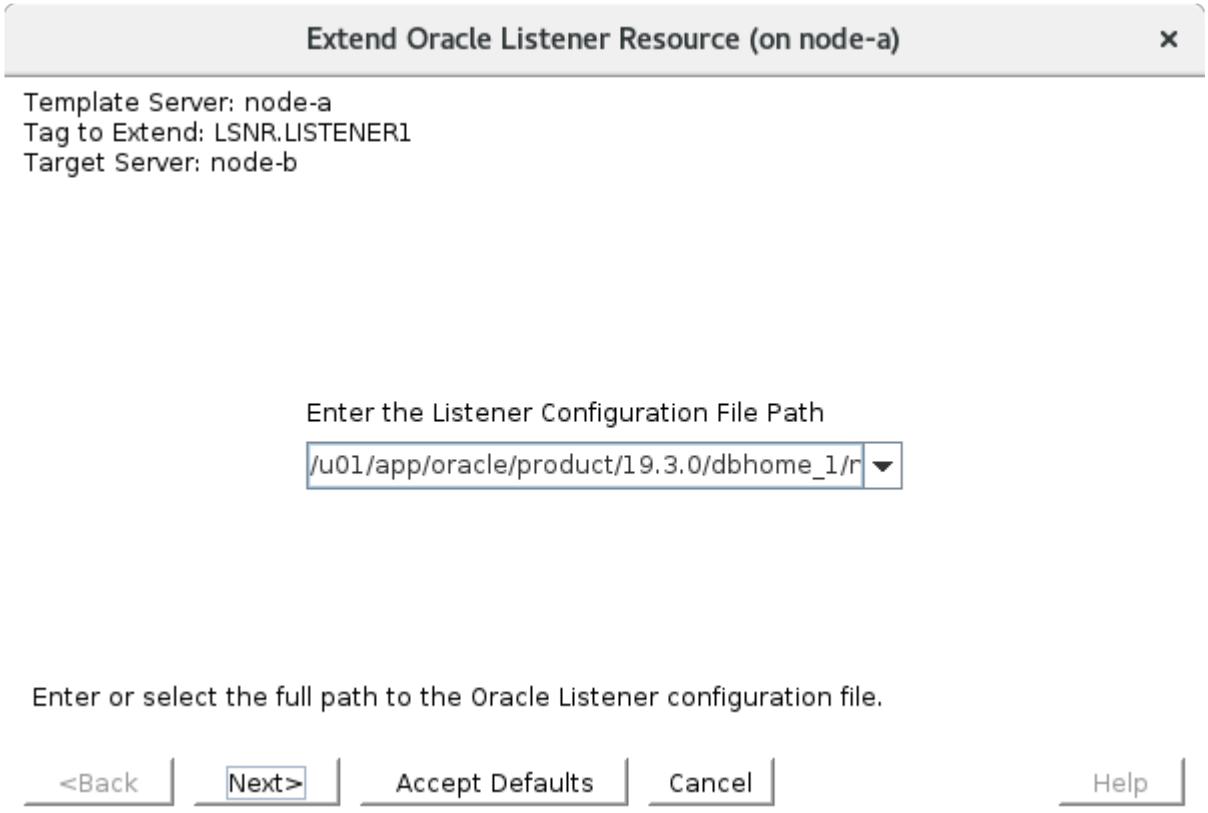

| フィールド | 値 |
| --- | --- |
| Switchback Type | intelligent |

| | |
|---|---|
| Template Priority | 1 |
| Target Priority | 10 |

「Pre-Extend Wizard」のチェックに合格したら、「Next」を選択して続行します。



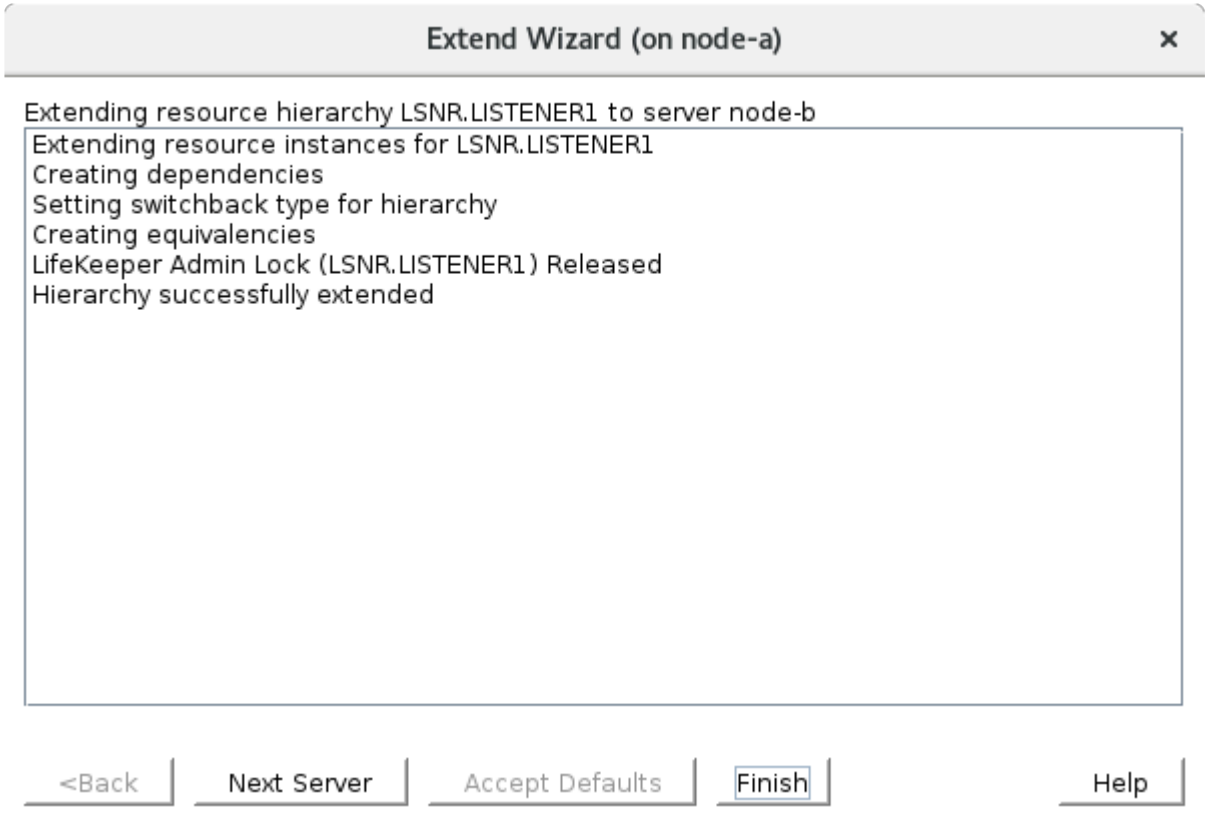

7. 「Extend Oracle Listener Resource」ウィザードでは、まずListener Configuration File Pathを選択します。これは手順2と同様、ウィザードによって検出されたパスを使用します。



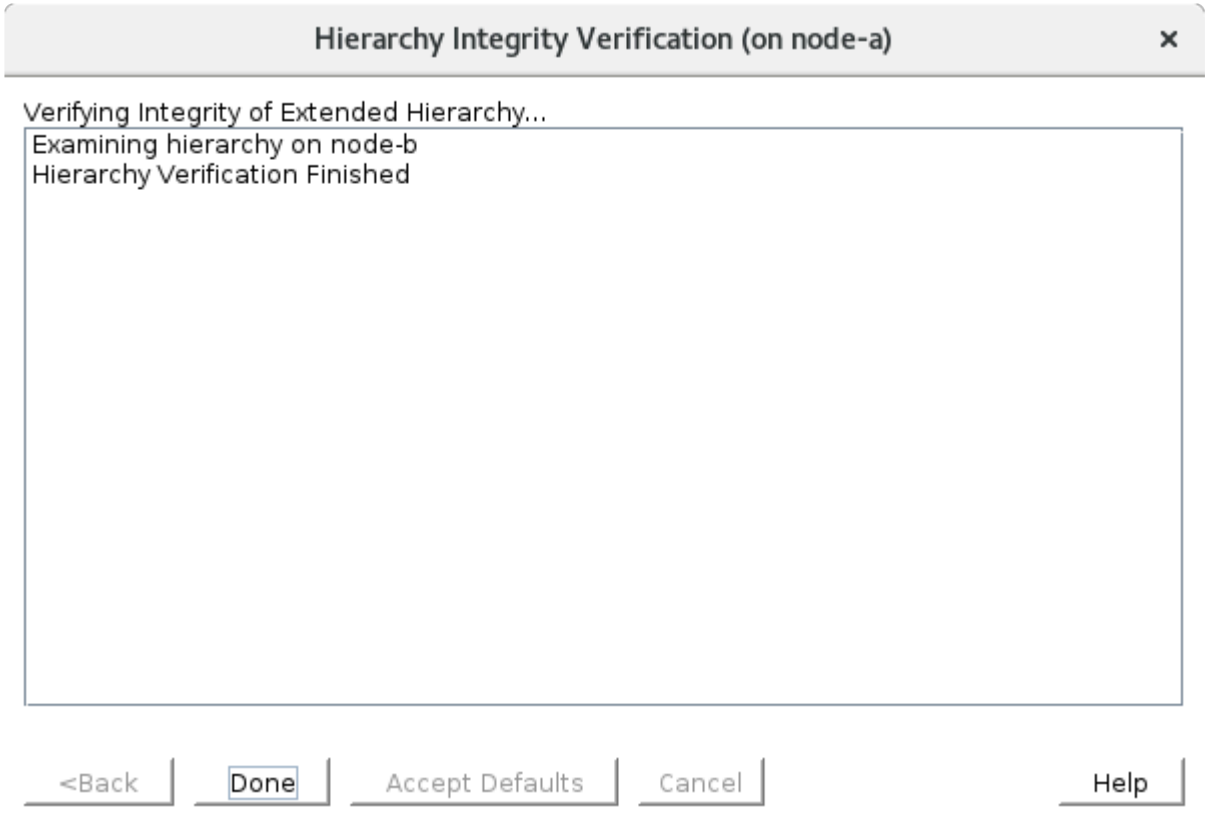

8. 以下の値を入力します。デフォルト値を選択します。

| フィールド | 値 |
|---|---|
| Listener Executable(s) | /u01/app/oracle/product/19.3.0/dbhome_1/bin/lsnrctl |
| Listener Tag | LSNR.LISTENER1 |

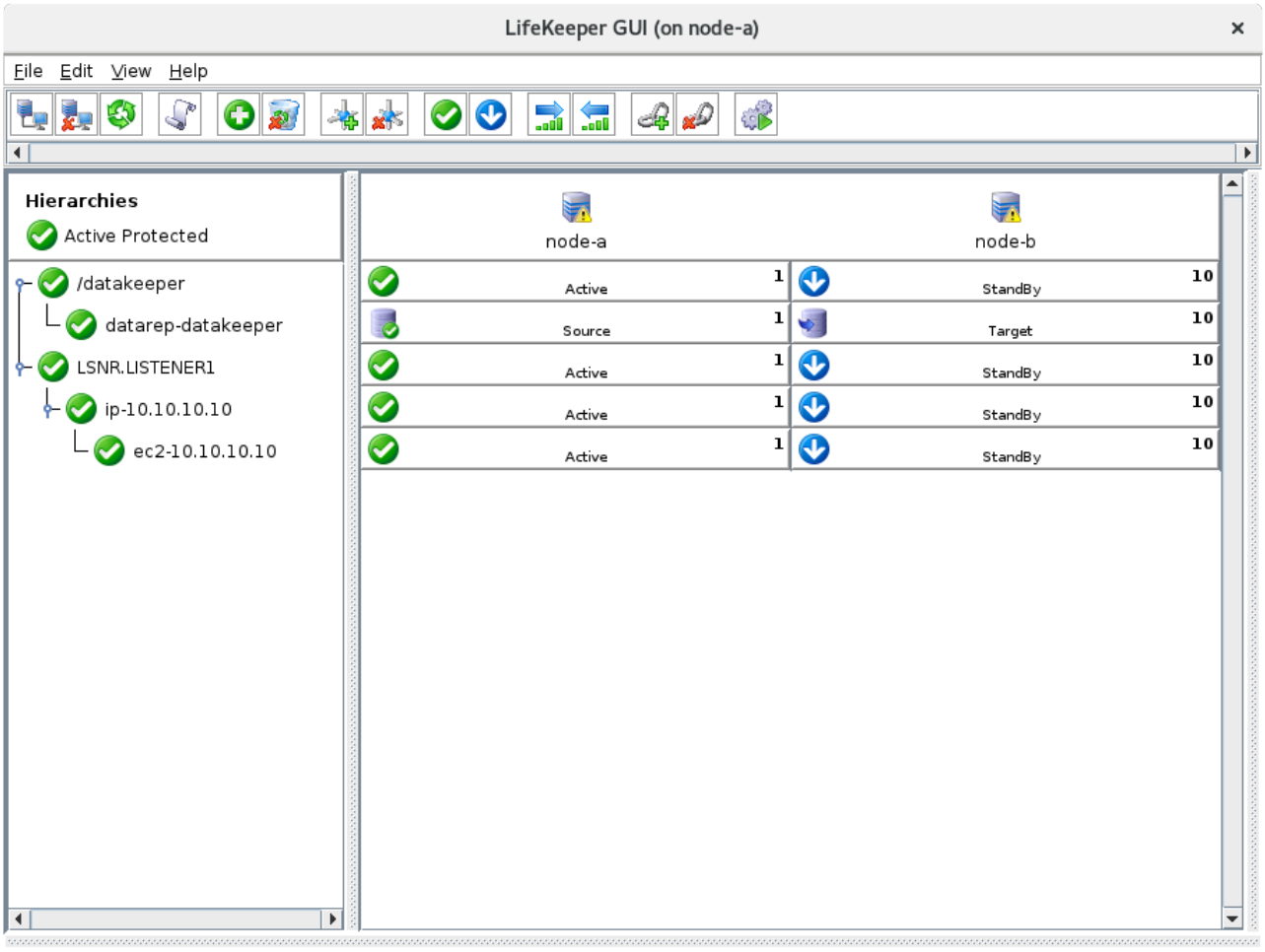
9. 拡張ウィザードは値を確認し、リスナーリソースをNode-Bに拡張します。設定が完了したら、[Finish] を選択します。




10. これで、階層が検証されました。 確認が完了したら、[Done] を選択します。



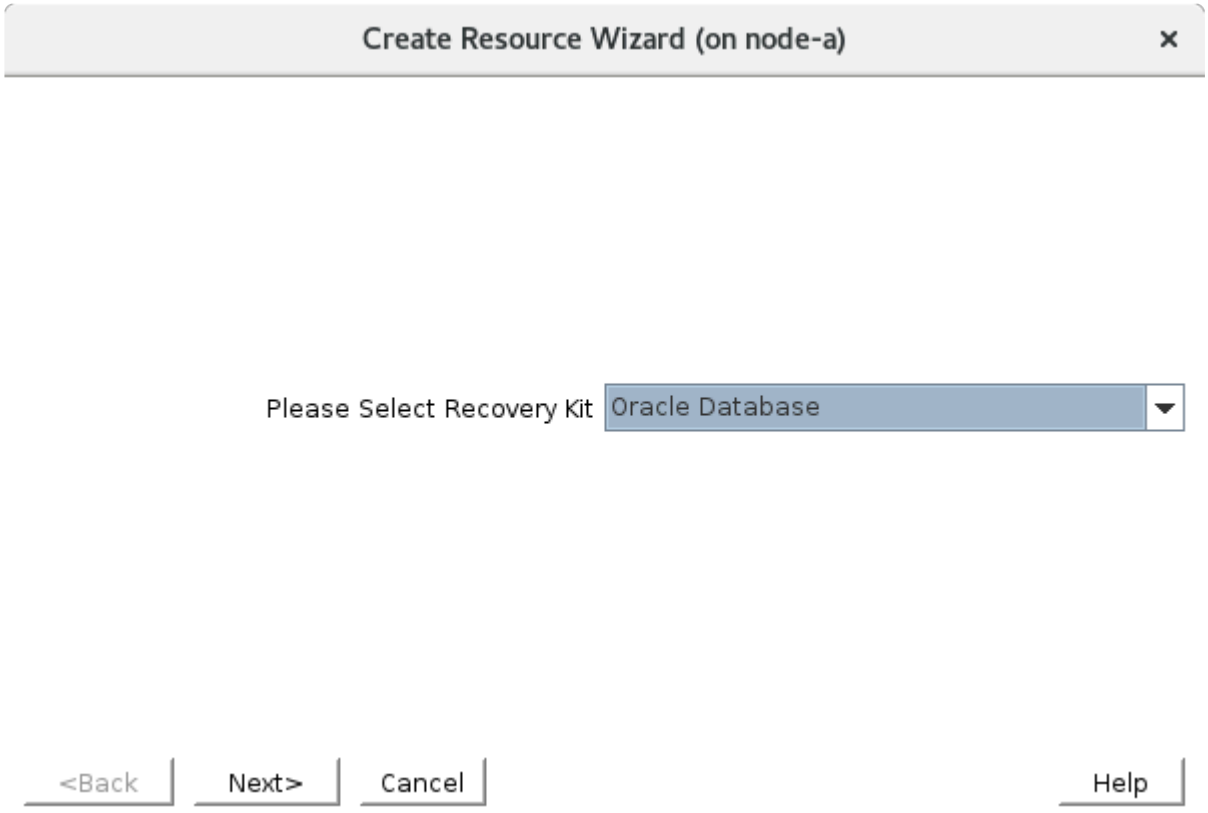

11. Oracleリスナーリソースが定義されました。Oracleリスナーリソースは、以下に示すようにip-10.10.10.10リソースに依存しています。


LifeKeeper GUI (on node-a)
File Edit View Help
Hierarchies
Active Protected
/datakeeper
datarep-datakeeper
LSNR.LISTENER1
ip-10.10.10.10
ec2-10.10.10.10
node-a
node-b
Active 1
StandBy 10
Source 1
Target 10
Active 1
StandBy 10
Active 1
StandBy 10
Active 1
StandBy 10

# 11.2.7.5.1.8. Oracleリソースの設定

Oracleデータベースの作成（PDBを除く） で説明したように、保護するインスタンスの名前はORCL.です。

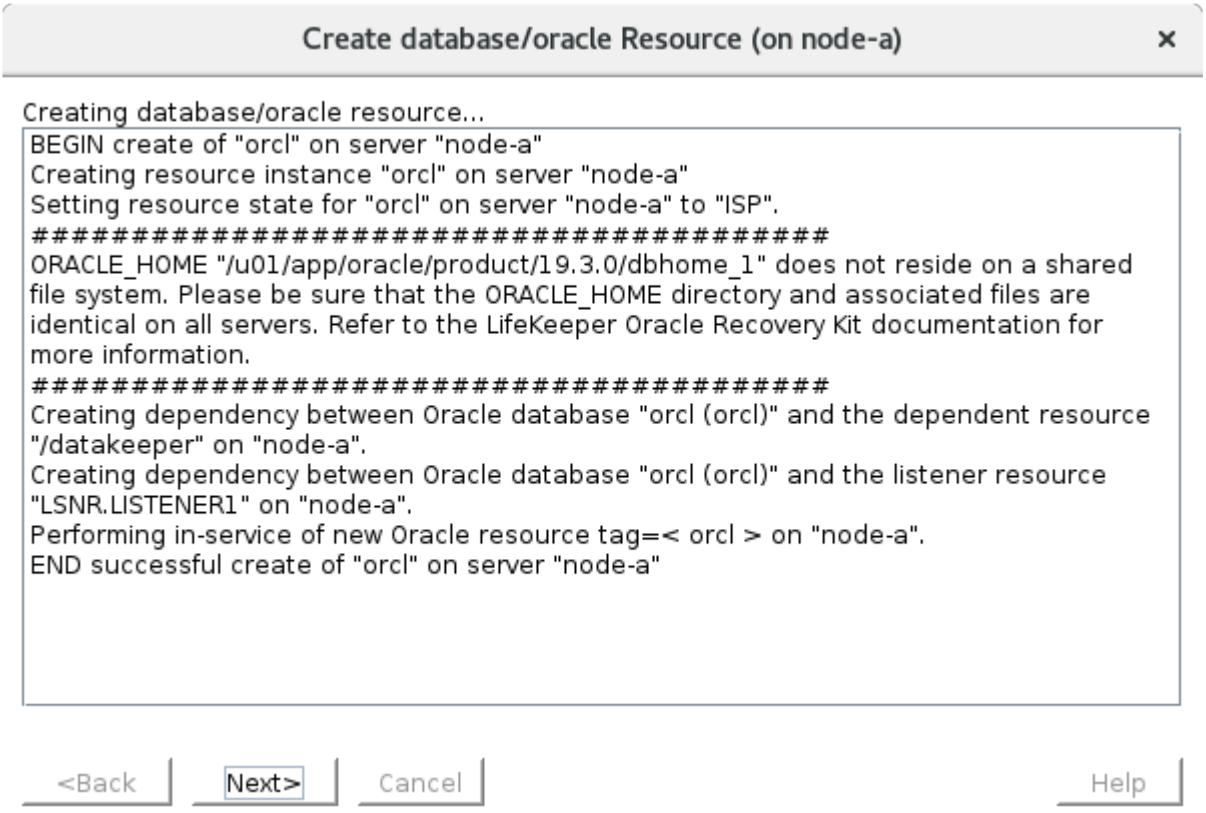
1. LifeKeeperユーザーインターフェースで、新しいリソースを定義します。 を選択して、Create Resource Wizard (on node-a) を開始します。Recovery Kitに「Oracle Database」を選択します。



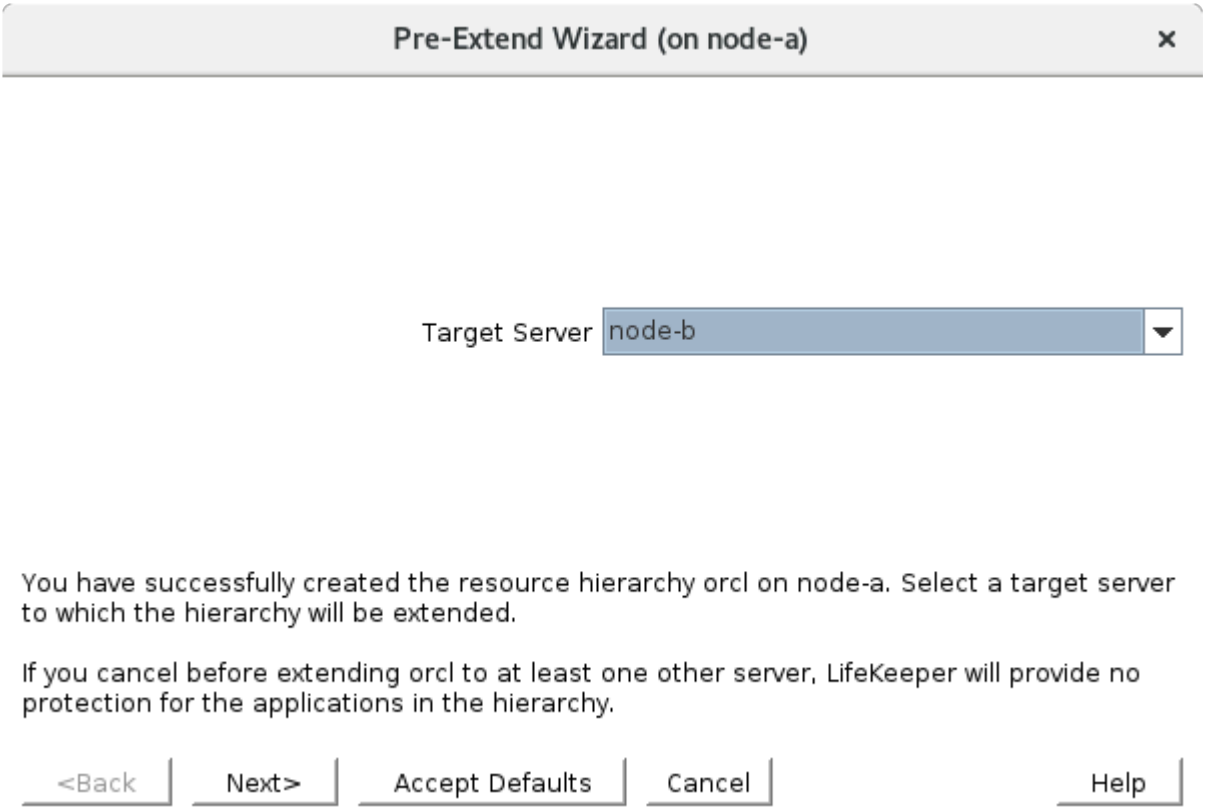

2. 以下の値を入力します。デフォルト値を選択します。

| フィールド | 値 |
|---|---|
| Switchback Type | intelligent |
| Server | node-a |
| Oracle SID | ORCL |
| Username | <Leave blank> |
| Oracle Listener | LSNR.LISTENER1 |
| Database Tag | ORCL |

3. ウィザードは、値を確認します。完了したら、「Next >」を選択して続行します。



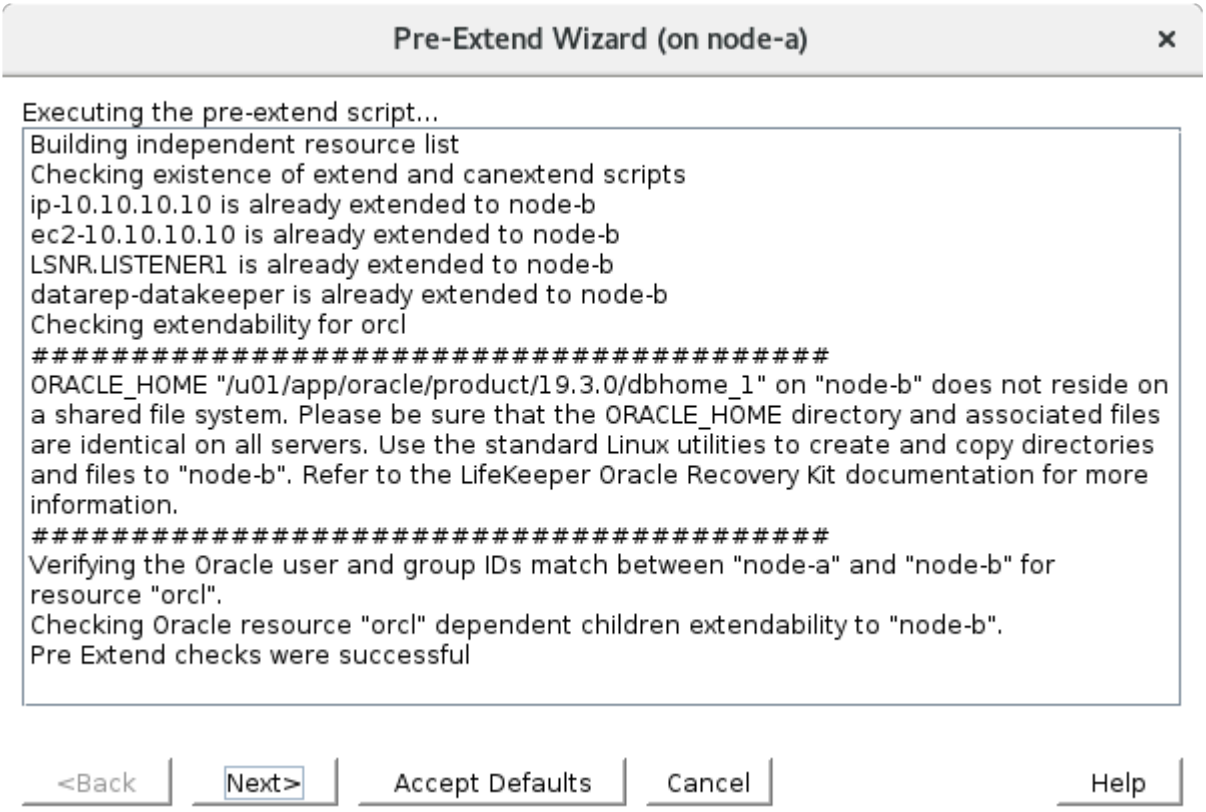

4. 次は「Pre-Extend Wizard」です。デフォルト値を選択します。



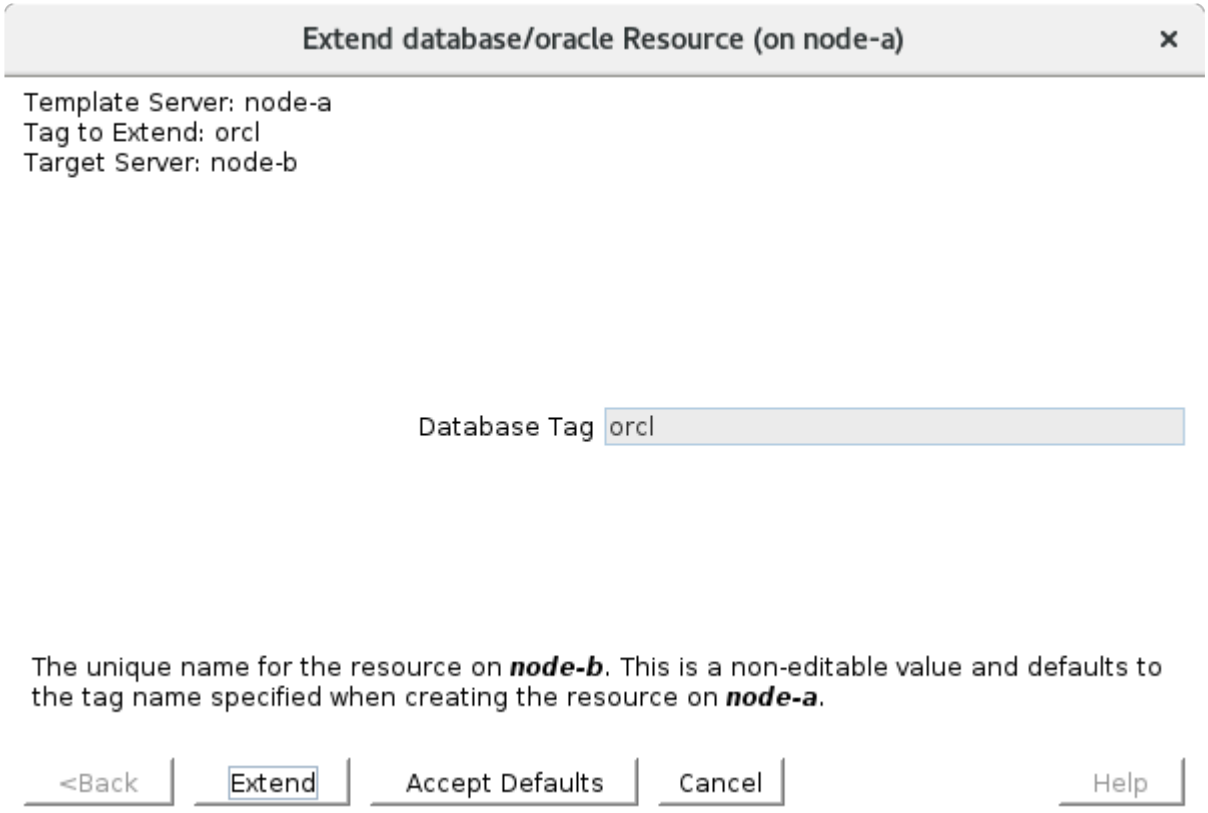

| フィールド | 値 |
| --- | --- |
| Switchback Type | intelligent |

| | |
|---|---|
| Template Priority | 1 |
| Target Priority | 10 |

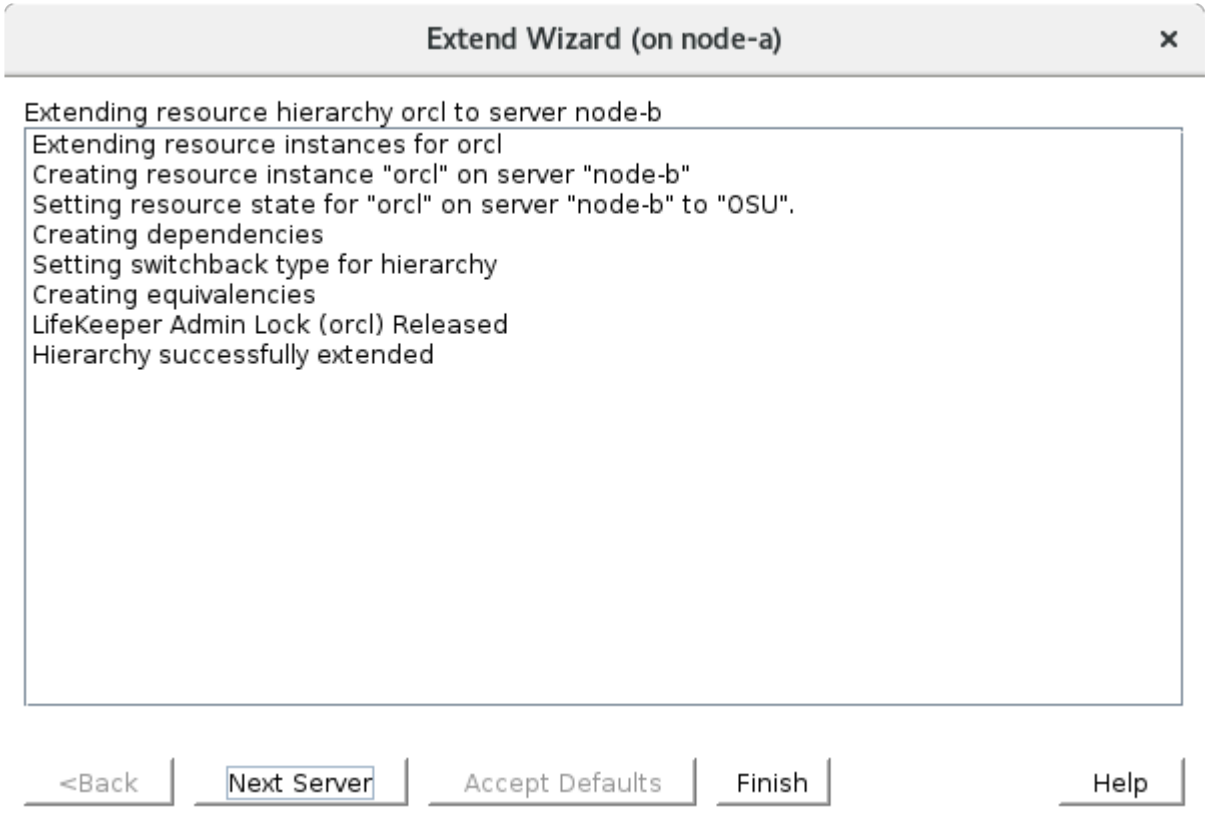
「Pre-Extend Wizard」のチェックに合格したら、[Next] をクリックして続行します。



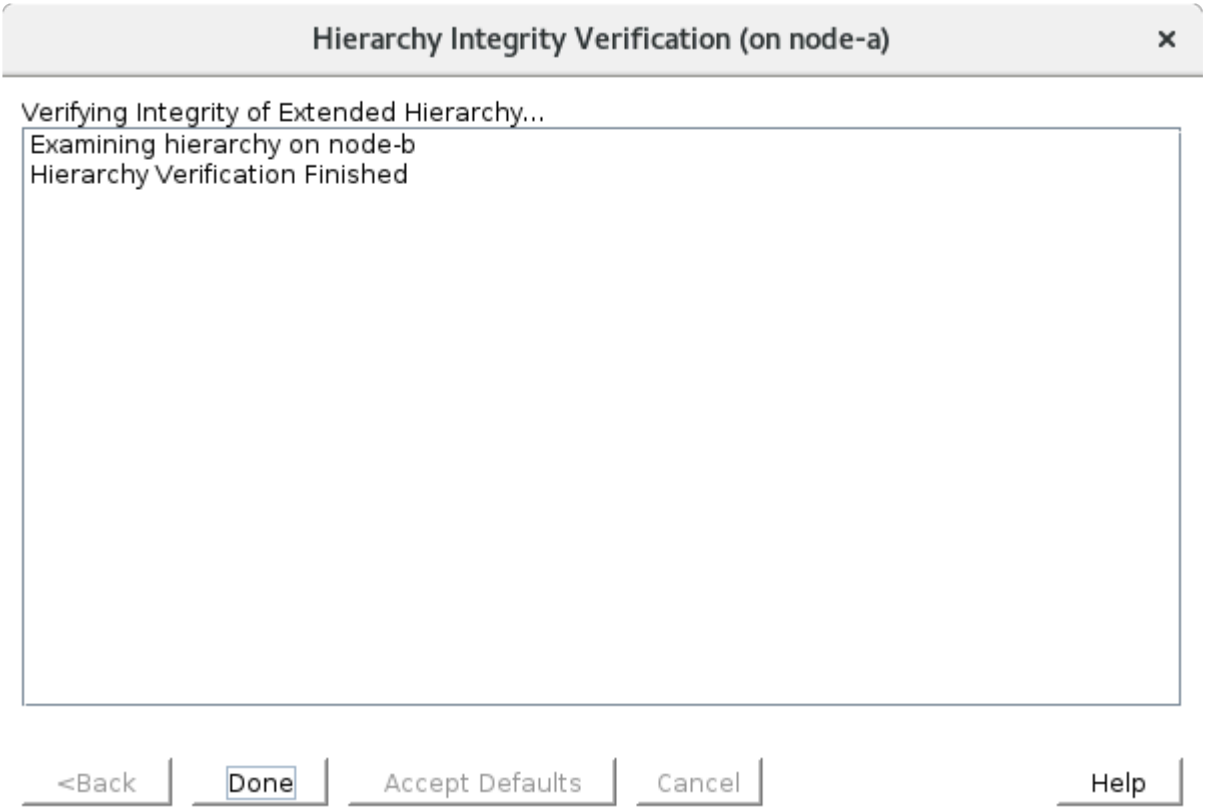

5. 「Extend database/oracle Resource」ウィザードで、まずデータベースタグを選択します。デフォルト値を確認し、[Extend] をクリックします。



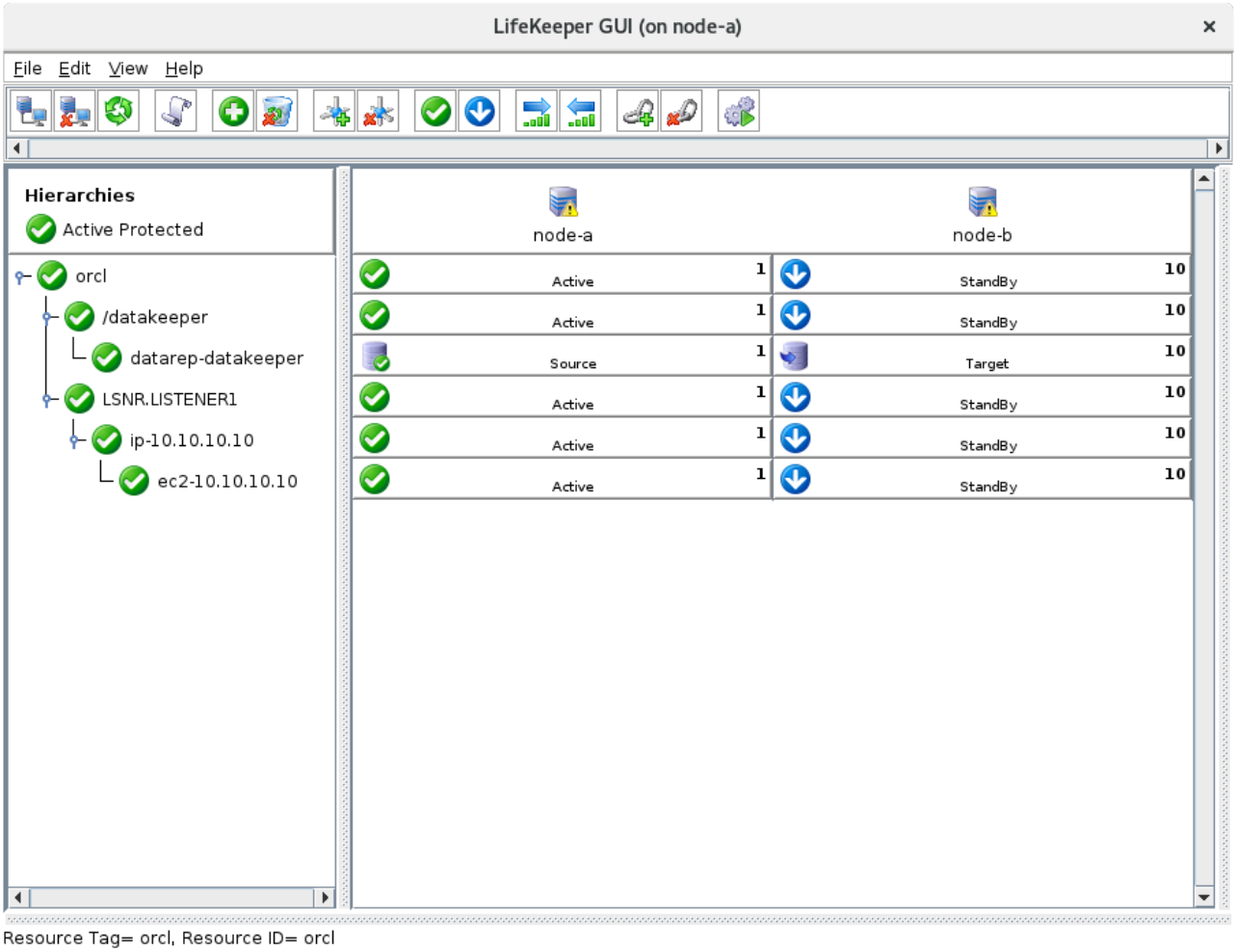

6. Extend WizardでOracleリソースを設定したら、[Finish] をクリックします。




7. これで階層がチェックされて完了しました。[Done] をクリックします。




8. LifeKeeper GUIに戻ります。「orcl」リソースは、以下に示すように、/datakeeperリソースとLSNR.LISTENERに依存します。



# 11.2.7.5.1.9. Oracleリソースのスイッチオーバーをテストする

## コマンドラインツールからステータスを把握する

スイッチオーバーをテストする前に、各ノードのステータスを確認してください。

### ファイルシステムのステータスを把握する
### ステータスを確認する最も簡単な方法は、 dfを使用することです。

```
[oracle@node-b dbhome_1]$ df
Filesystem     1K-blocks     Used Available Use% Mounted on
devtmpfs          916888        0    916888   0% /dev
tmpfs            7340032   450560   6889472   7% /dev/shm
tmpfs             940184    17480    922704   2% /run
tmpfs             940184        0    940184   0% /sys/fs/cgroup
/dev/xvda2      31444972 17247248  14197724  55% /
tmpfs             188040        0    188040   0% /run/user/1000
/dev/md0        20960236  4997824  15962412  24% /datakeeper
tmpfs             188040        0    188040   0% /run/user/54321
```

/datakeeperリソースはノードに接続され、ここに示すようにマウントされます。

### UリスナーとOracleプロセスのステータスを把握する

ps コマンドを使用して、各ノードのステータスを確認できます。プロセスが実行されている場合は、以下のように一覧表示されます。

```
$ ps afx | grep -v grep | grep tnslsnr
13806 ?        Ssl    0:00 /u01/app/oracle/product/19.3.0/dbhome_1/bin/tnslsn
r LISTENER1 -inherit
$ ps afx | grep -v grep | grep ora_pmon
23091 ?        Ss     0:00 ora_pmon_orcl
```

### IPリソースのステータスを把握する

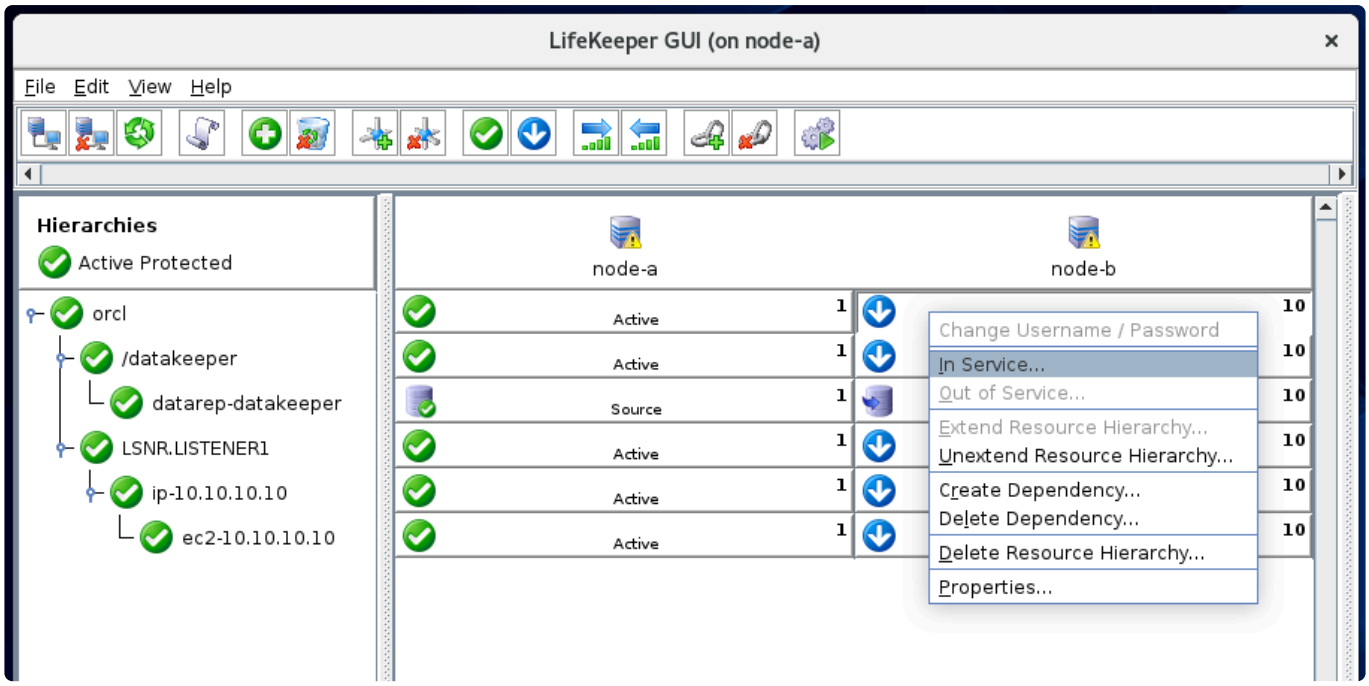
クライアントがアクティブノードに接続する方法 で説明したように、IPリソースがアクティブな場合、仮想IPアドレスは次のようにipコマンドで確認できます（10.10.10.10参照）

```
[oracle@node-b dbhome_1]$ ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 02:a8:0c:57:53:0b brd ff:ff:ff:ff:ff:ff
```

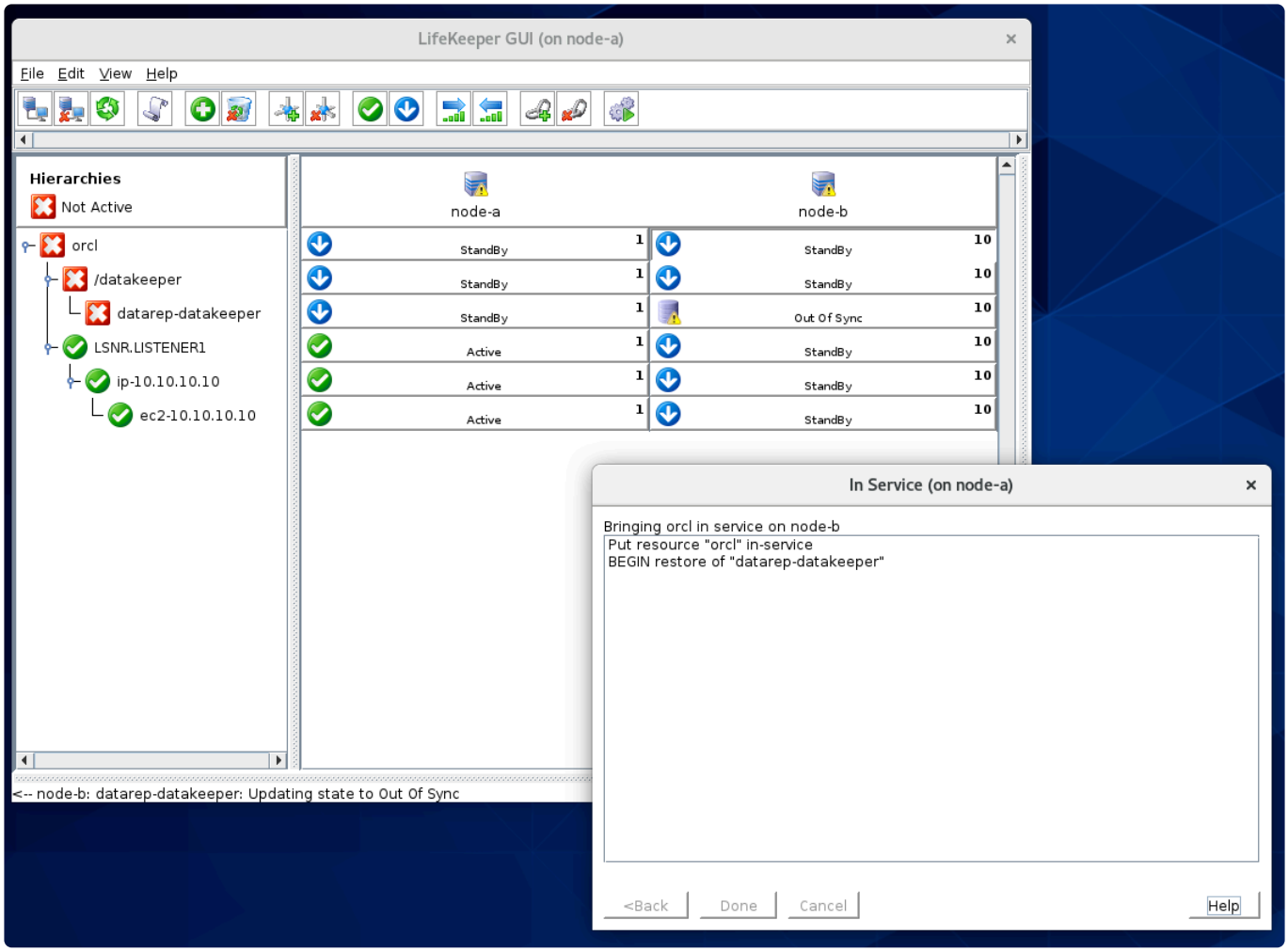
```
    inet 10.20.10.12/24 brd 10.20.10.255 scope global noprefixroute dynamic et
h0
       valid_lft 3585sec preferred_lft 3585sec
    inet 10.10.10.10/32 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::a8:cff:fe57:530b/64 scope link
       valid_lft forever preferred_lft forever
```

## ノード間でリソースを切り替える

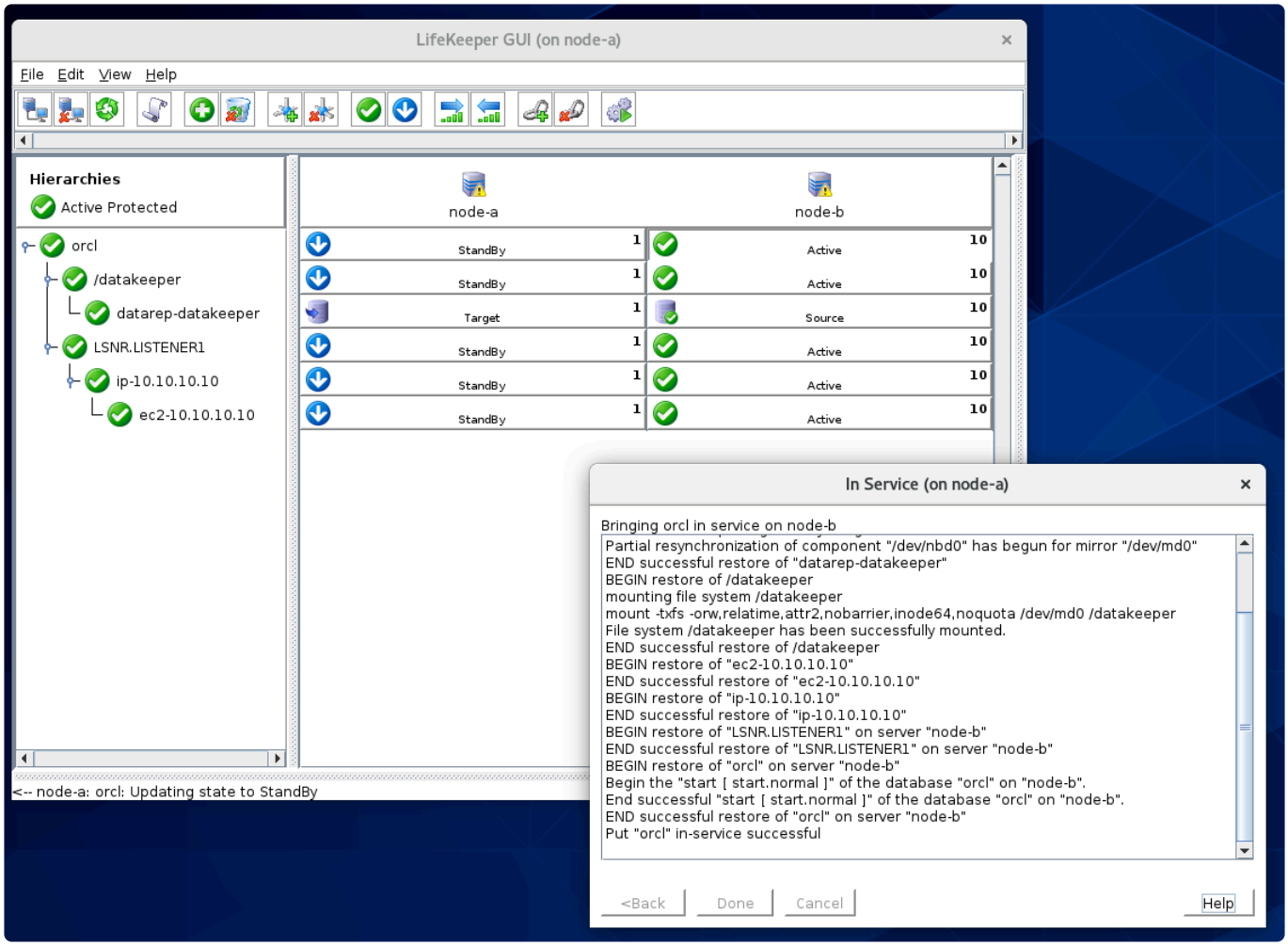
node-b（スタンバイノード）の「orcl」リソースでコンテキストメニューを表示し、「In Service」を選択します。




まず、リソースとすべての子の依存関係がnode-aの「StandBy」に移行します。




node-aですべてのリソースを停止させると、node-bで依存関係と「orcl」リソースが起動します。


LifeKeeper GUI (on node-a)
File Edit View Help
Hierarchies
Active Protected
orcl
/datakeeper
datarep-datakeeper
LSNR.LISTENER1
ip-10.10.10.10
ec2-10.10.10.10
node-a
node-b
StandBy 1 Active 10
StandBy 1 Active 10
Target 1 Source 10
StandBy 1 Active 10
StandBy 1 Active 10
StandBy 1 Active 10
<-- node-a: orcl: Updating state to StandBy
In Service (on node-a)
Bringing orcl in service on node-b
Partial resynchronization of component "/dev/nbd0" has begun for mirror "/dev/md0"
END successful restore of "datarep-datakeeper"
BEGIN restore of /datakeeper
mounting file system /datakeeper
mount -txfs -orw,relatime,attr2,nobarrier,inode64,noquota /dev/md0 /datakeeper
File system /datakeeper has been successfully mounted.
END successful restore of /datakeeper
BEGIN restore of "ec2-10.10.10.10"
END successful restore of "ec2-10.10.10.10"
BEGIN restore of "ip-10.10.10.10"
END successful restore of "ip-10.10.10.10"
BEGIN restore of "LSNR.LISTENER1" on server "node-b"
END successful restore of "LSNR.LISTENER1" on server "node-b"
BEGIN restore of "orcl" on server "node-b"
Begin the "start [ start.normal ]" of the database "orcl" on "node-b".
End successful "start [ start.normal ]" of the database "orcl" on "node-b".
END successful restore of "orcl" on server "node-b"
Put "orcl" in-service successful
<Back
Done
Cancel
Help

# 11.2.7.5.2. Quick Service Protectionを使用したMicrosoft SQL Serverの保護

このセクションでは、Linux環境でMicrosoft SQL Server2017を保護するための手順を概説します

> ✻ 本ドキュメントでは、Microsoft SQL Server 2017(以後MSSQLとする)を例として説明します。Microsoft SQL Server 2012 R2からMicrosoft SQL Server 2019で同じ手順を適用することができます。

LifeKeeperにはMSSQL専用のApplication Recovery Kitはありませんが、LifeKeeperはMSSQLを一般的なサービスとして保護できます。この機能はQuick Service Protection (QSP)と呼ばれ、MSSQLだけでなく、Linux環境で動作する他のサービスも保護することができます。
データベースリソースを保護するには、DataKeeperを使用してノード間でデータを複製する必要があります（共有ストレージまたはSANデバイスを使用している場合を除く）。
次の表は、各コンポーネントの位置の概要を示したものです。

| アイテム | ロケーション |
|---|---|
| Master DataBase Files | /datakeeper/mssql/data |
| Master Log File | /datakeeper/mssql/xlog |

ファイルシステムのデータレプリケーションを作成する方法 で説明したように、本ガイドでは `/datakeeper` を使用してノード間のデータレプリケーションを行います。したがって、 `/datakeeper/mssql/data` のデータもノード間で複製されます。（これは `/datakeeper/mssql/data` でも同様です）

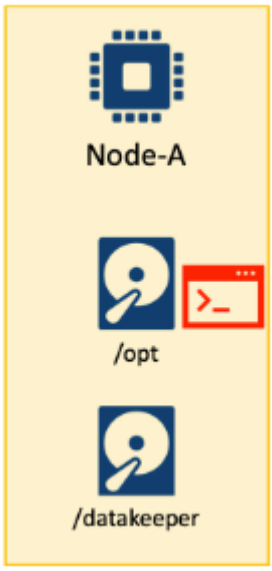
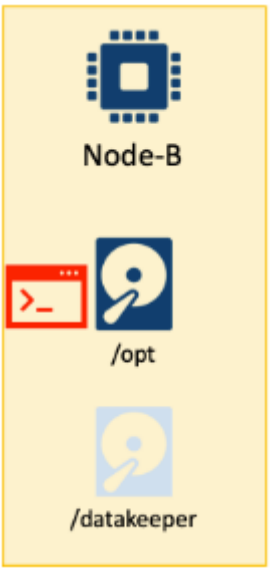
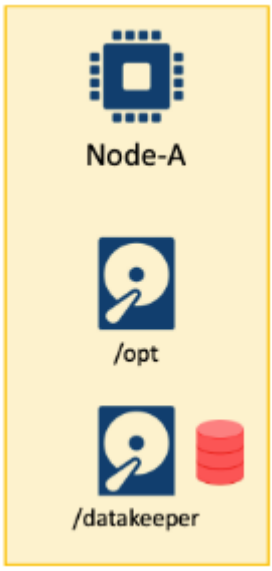
このガイドでは、以下のコンピューティングリソースを使用しています（Microsoft SQL Serverリソースは、評価目的であっても他のリソースよりも多くのメモリを必要とします）ので、ご注意ください。

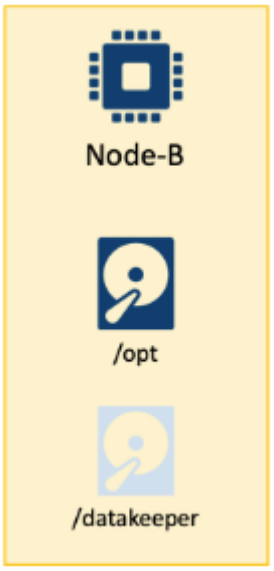
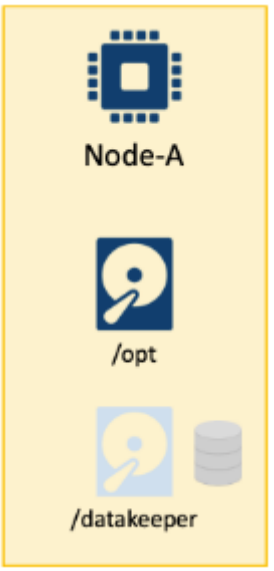
| リソース | 必要なサイズ |
|---|---|
| Memory | 4 GiB |

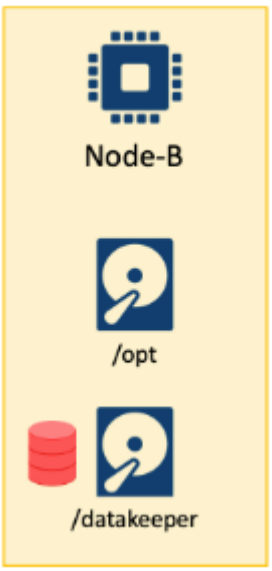
## MSSQLリソースの設定

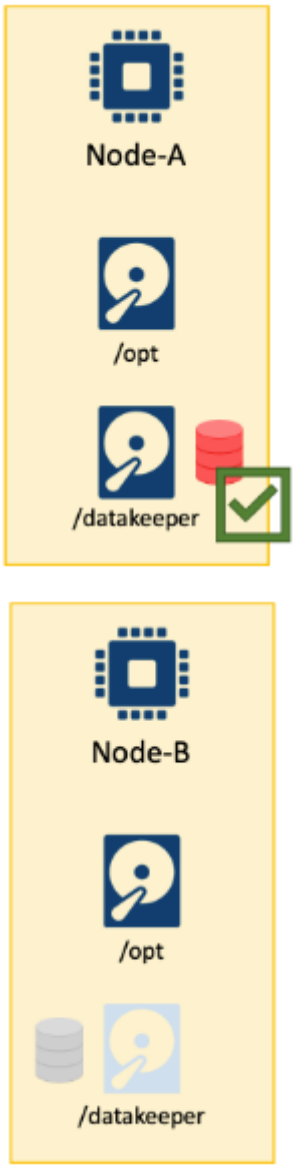
次の表は、MSSQLリソースを構成するための一般的な手順の概要です。赤色の「スタックディスク」のアイコンは、各ステップの時点でデータベースインスタンスを持つノードを示しています。グレーの「スタックディスク」アイコンは、データーベースが稼働していないことを示します。

また、灰色の「ストレージ」アイコンは、複製されたストレージ `/datakeeper` がノードで使用できないことを示します。

| | |
|---|---|
| 両方のノードに**MSSQL2017**をインストールする<br><br>Install MSSQL 2017 on a System |  <br>  |
| **Node-A**でデータベースを作成する<br><br>データストレージがノードで使用可能かどうかを確認する方法 を確認してください。データストレージはNode-aで利用可能である必要があります。<br><br>マスターデータベースとログファイルをnode-a上の複製されたストレージに再配置する |  |

<table>
<tr>
<td></td>
<td> 
</td>
</tr>
<tr>
<td>Node-Bにデータベースを作成する<br><br>node-bにデータストレージをスイッチオーバーする<br><br>/datakeeper/MSSQLの下のフォルダーの名前を変更する<br><br>マスターデータベースとログファイルをNode-bの複製されたストレージに再配置する</td>
<td> 


</td>
</tr>
</table>

| | |
|---|---|
| **LifeKeeper**を使用して設定を完了する<br><br>node-aへデーターストレージをスイッチバックする<br><br>LifeKeeperでリソースを構成する<br><br>両方のノードでLocalRecoveryパラメーターをカスタマイズする<br><br>リソース間の依存関係を更新する |  |

# 11.2.7.5.2.1. Microsoft SQL Server 2017のインストール

両方のノードにMSSQLをインストールします。

1. MicrosoftのSQL Server on Linux のインストール ガイド guide を参照してください。このガイドには、MSSQLサーバーだけでなく、異なるバージョンのオペレーティングシステムに関する説明も含まれています。

2. Microsoft SQL Server 2017 Red Hat リポジトリの設定ファイルをダウンロードします。

```
# curl -o /etc/yum.repos.d/mssql-server.repo
https://packages.microsoft.com/config/rhel/8/mssql-server-2017.repo
```

3. 以下のコマンドを実行し、SQL Serverをインストールします。

```
1 # yum install -y mssql-server
```

4. MSSQLサーバーを設定します。

```
1 # /opt/mssql/bin/mssql-conf setup
```

5. これで、サーバーのインストールは完了です。以下のコマンドでMSSQLサーバーの状態を確認します。

```
1 # systemctl status mssql-server
```

6. MSSQLサーバーを起動する前にまだいくつかの手順があります。これでMSSQLサーバーは正常にインストールされました。両ノードでこれらの手順を実行します。

# 11.2.7.5.2.2. マスターデーターベースとログファイルを複製されたストレージに再配置する

前述したように、データーは複製されたファイルシステム上に保存する必要があります。そのため、各コンポーネントには以下の場所を使用します。

| アイテム | 場所 |
| --- | --- |
| Master DataBase Files | /datakeeper/mssql/data |
| Master Log File | /datakeeper/mssql/xlog |

Node-aおよびNode-bの*両方*で以下の手順を実行する必要があります。

その間に行う必要のある追加の手順がいくつかあります。Quick Service Protectionを使用したMSSQLの保護

まず、node-aにディレクトリを作成します。

```
# mkdir -p /datakeeper/mssql/data
# mkdir -p /datakeeper/mssql/xlog
# chown mssql -R /datakeeper/mssql/
# chgrp mssql -R /datakeeper/mssql/
```

次のように、/opt/mssql/bin/mssql-conf ツールを使用して、マスターデーターベースを新しく作成されたディレクトリに再配置します。

```
# /opt/mssql/bin/mssql-conf set filelocation.masterlogfile /datakeeper/mssql/xlog/mastlog.ldf
SQL Server needs to be restarted in order to apply this setting. Please run
'systemctl restart mssql-server.service'.

# /opt/mssql/bin/mssql-conf set filelocation.masterdatafile /datakeeper/mssql/data/master.mdf
SQL Server needs to be restarted in order to apply this setting. Please run
'systemctl restart mssql-server.service'.
```

MSSQLサーバーを起動します。

```
# systemctl restart mssql-server.service
# systemctl status mssql-server.service
● mssql-server.service - Microsoft SQL Server Database Engine
   Loaded: loaded (/usr/lib/systemd/system/mssql-server.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2020-12-30 18:55:54 UTC; 6s ago
     Docs: https://docs.microsoft.com/en-us/sql/linux
 Main PID: 15506 (sqlservr)
   CGroup: /system.slice/mssql-server.service
           ├─15506 /opt/mssql/bin/sqlservr
           └─15508 /opt/mssql/bin/sqlservr
```

データーベースが起動すると、先ほど指定したフォルダーに作成されたいくつかのファイルが表示されます。

```
# ls -al /datakeeper/mssql/data
total 51136
drwxr-xr-x 2 mssql mssql      137 Dec 30 18:56 .
drwxr-xr-x 6 mssql mssql       62 Dec 30 18:48 ..
-rw-rw---- 1 mssql mssql  4194304 Dec 30 18:56 master.mdf
-rw-rw---- 1 mssql mssql  8388608 Dec 30 18:56 modellog.ldf
-rw-rw---- 1 mssql mssql  8388608 Dec 30 18:56 model.mdf
-rw-rw---- 1 mssql mssql 14090240 Dec 30 18:56 msdbdata.mdf
-rw-rw---- 1 mssql mssql   524288 Dec 30 18:56 msdblog.ldf
-rw-rw---- 1 mssql mssql  8388608 Dec 30 18:56 tempdb.mdf
-rw-rw---- 1 mssql mssql  8388608 Dec 30 18:56 templog.ldf
# ls -al /datakeeper/mssql/xlog
total 2048
drwxr-xr-x 2 mssql mssql      25 Dec 30 18:55 .
drwxr-xr-x 6 mssql mssql      62 Dec 30 18:48 ..
-rw-rw---- 1 mssql mssql 2097152 Dec 30 18:56 mastlog.ldf
```

これで、インスタンスを停止できます。

```
# systemctl stop mssql-server.service
```

# 11.2.7.5.2.3. /dataKeeper/MSSQLの下のフォルダーの名前を変更する

これまでに、以下のタスクが完了しました。

1. node-aにデーターベースを作成
2. `datarep-datakeeper` リソースをnode-bにスイッチオーバー

ステップ1で、新しく作成されたSQLデータベースに関する情報が `/var/opt/mssql/mssql.conf` に作成されました。

次に、node-bに同じmssqlデータベースを作成して、mssqlデータベースに関する情報がnode-bの `/var/opt/mssql/mssql.conf` に格納されるようにします。データベースファイル自体は `/datakeeper/mssql/` に格納されています。

node-bで、これらのファイルは次のように表示されます。

```
[root@node-b ~]# cd /datakeeper/mssql/
[root@node-b mssql]# ls -al data
total 51136
drwxr-xr-x 2 mssql mssql      137 Dec 30 18:38 .
drwxr-xr-x 6 mssql mssql       62 Dec 30 18:36 ..
-rw-rw---- 1 mssql mssql  4194304 Dec 30 18:38 master.mdf
-rw-rw---- 1 mssql mssql  8388608 Dec 30 18:38 modellog.ldf
-rw-rw---- 1 mssql mssql  8388608 Dec 30 18:38 model.mdf
-rw-rw---- 1 mssql mssql 14090240 Dec 30 18:38 msdbdata.mdf
-rw-rw---- 1 mssql mssql   524288 Dec 30 18:38 msdblog.ldf
-rw-rw---- 1 mssql mssql  8388608 Dec 30 18:38 tempdb.mdf
-rw-rw---- 1 mssql mssql  8388608 Dec 30 18:38 templog.ldf
[root@node-b mssql]# ls -al xlog_bak/
total 2048
drwxr-xr-x 2 mssql mssql      25 Dec 30 18:38 .
drwxr-xr-x 6 mssql mssql      62 Dec 30 18:36 ..
-rw-rw---- 1 mssql mssql 2097152 Dec 30 18:38 mastlog.ldf
```

node-bでデータベースを正常に作成できるように、これらのフォルダーの名前を変更します。

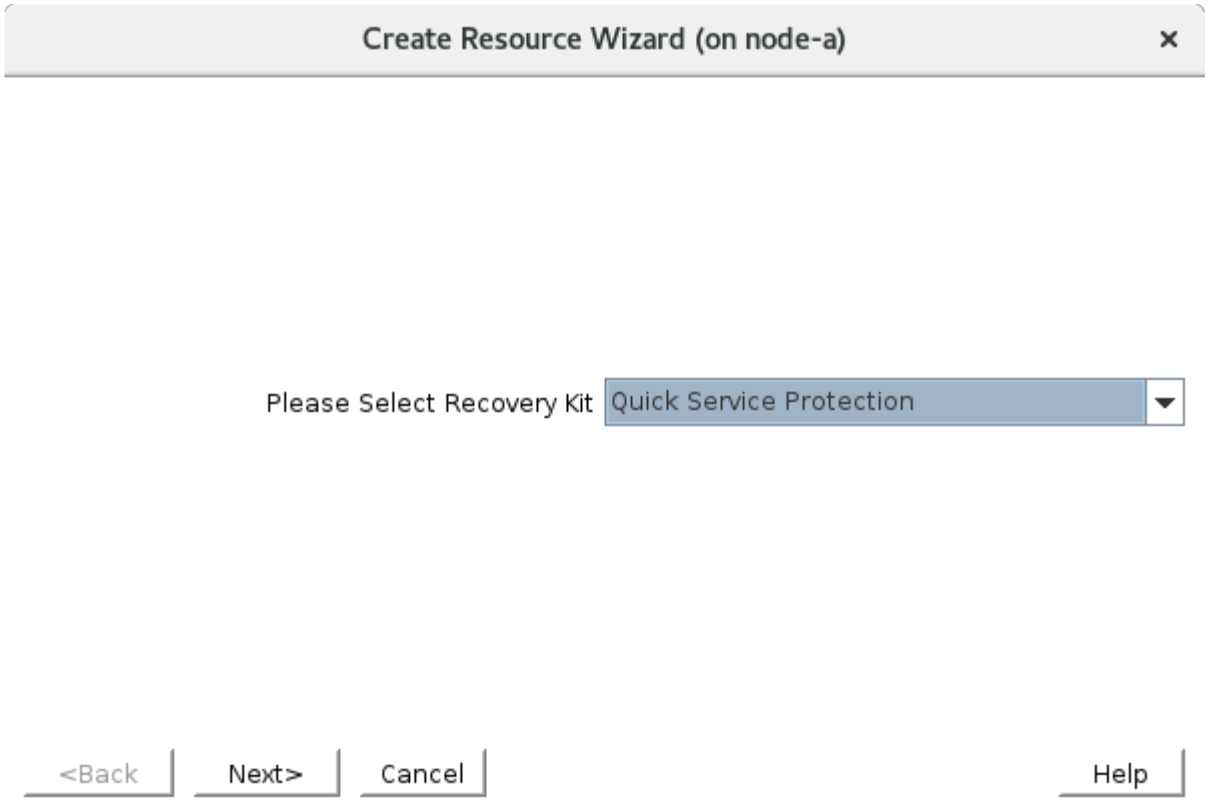
```
[root@node-b mssql]# mv data data_bak
[root@node-b mssql]# mv xlog xlog_bak
[root@node-b mssql]# ls -al
total 0
drwxr-xr-x 4 mssql mssql  38 Dec 30 18:48 .
drwxr-xr-x 5 root  root   57 Dec 30 18:36 ..
drwxr-xr-x 2 mssql mssql 137 Dec 30 18:38 data_bak
drwxr-xr-x 2 mssql mssql  25 Dec 30 18:38 xlog_bak
```

# 11.2.7.5.2.4. MSSQLリソースの設定

マスターデータベースが複製されたストレージに再配置されたので、LifeKeeperを使用してMSSQLサーバーリソースを保護する準備が整いました。LifeKeeperで設定する前に、node-aでMSSQLを起動します

```
[root@node-a ~]# systemctl start mssql-server.service
[root@node-a ~]# systemctl status mssql-server.service
● mssql-server.service - Microsoft SQL Server Database Engine
   Loaded: loaded (/usr/lib/systemd/system/mssql-server.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2020-12-30 18:59:02 UTC; 1s ago
     Docs: https://docs.microsoft.com/en-us/sql/linux
 Main PID: 17146 (sqlservr)
   CGroup: /system.slice/mssql-server.service
           ├─17146 /opt/mssql/bin/sqlservr
           └─17148 /opt/mssql/bin/sqlservr
```

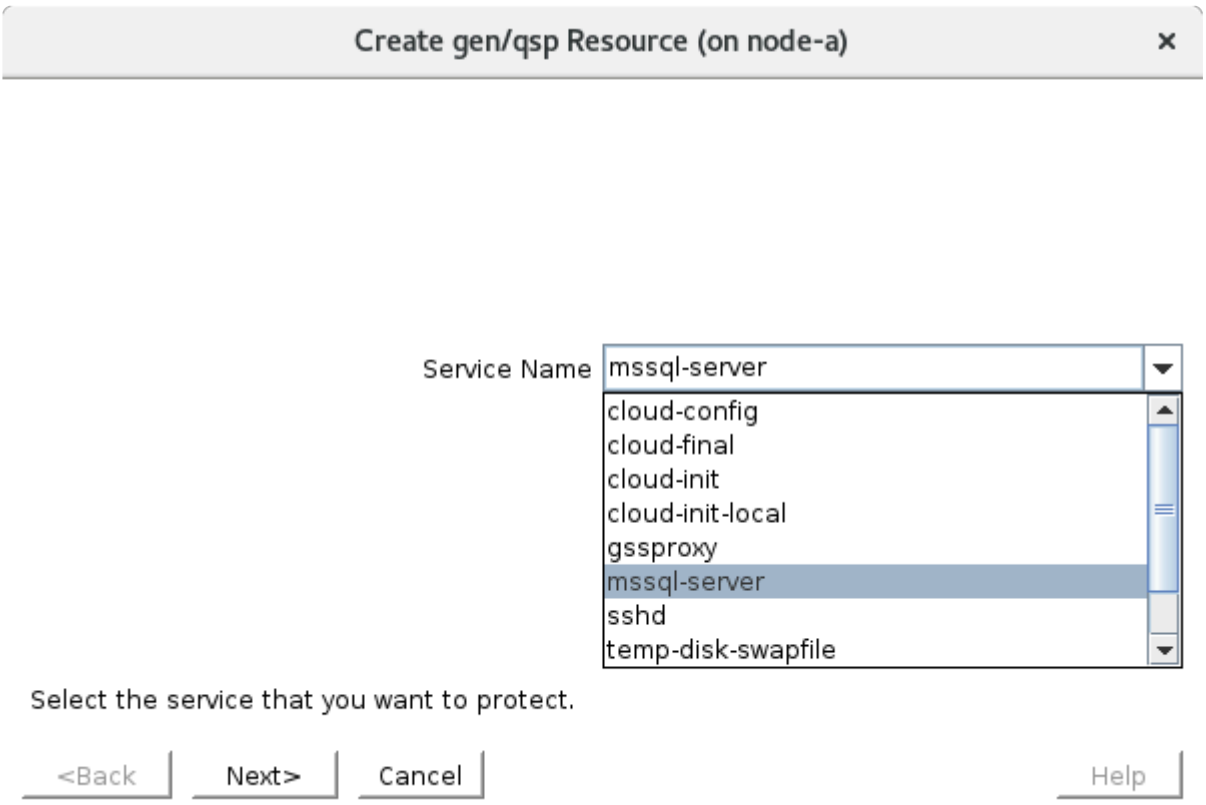
1. LifeKeeperユーザーインターフェースで、新しいリソースを定義します。 を選択して、Create Resource Wizard (on node-a) を開始します。Recovery Kitに「Quick Service Protection」を選択します。




2. 以下の値を入力します。デフォルト値を選択します。

| フィールド | 値 |
|---|---|
| Switchback Type | intelligent |
| Server | node-a |

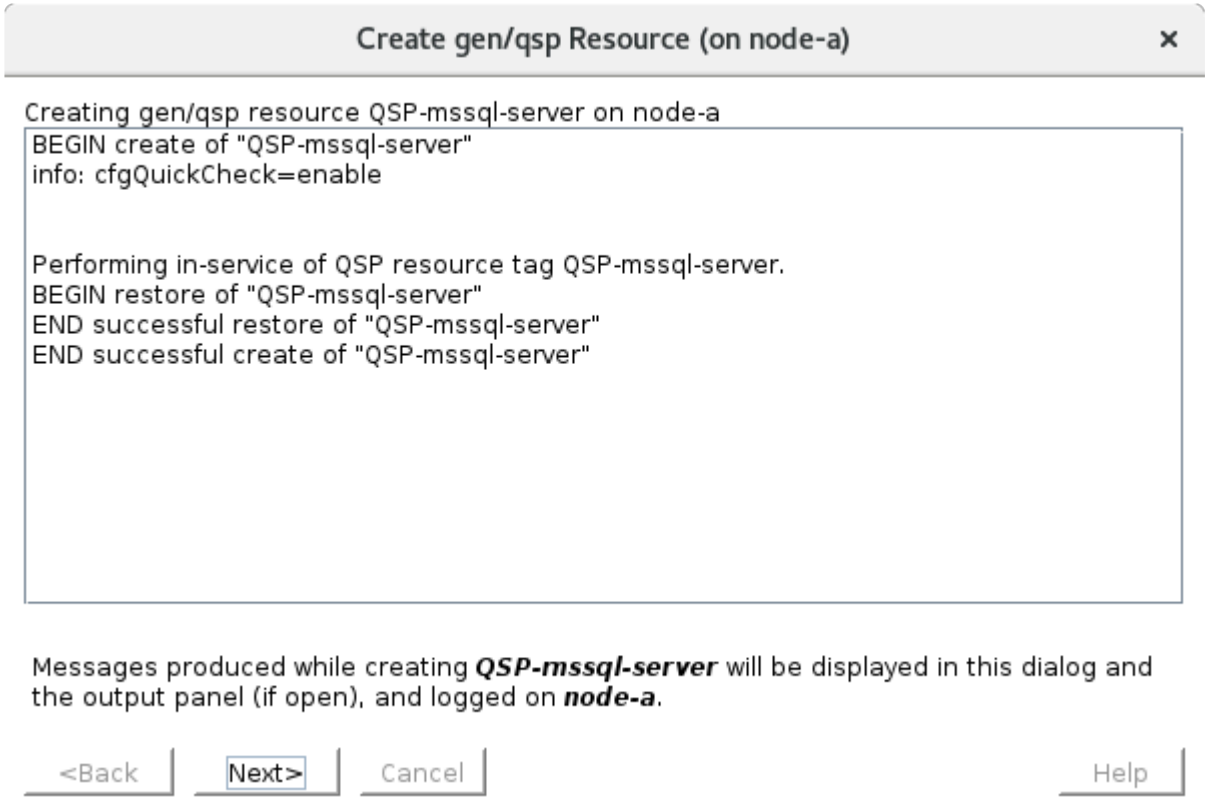
3. [Service Name] にドロップダウンリストから [mssql-server] を選択します。



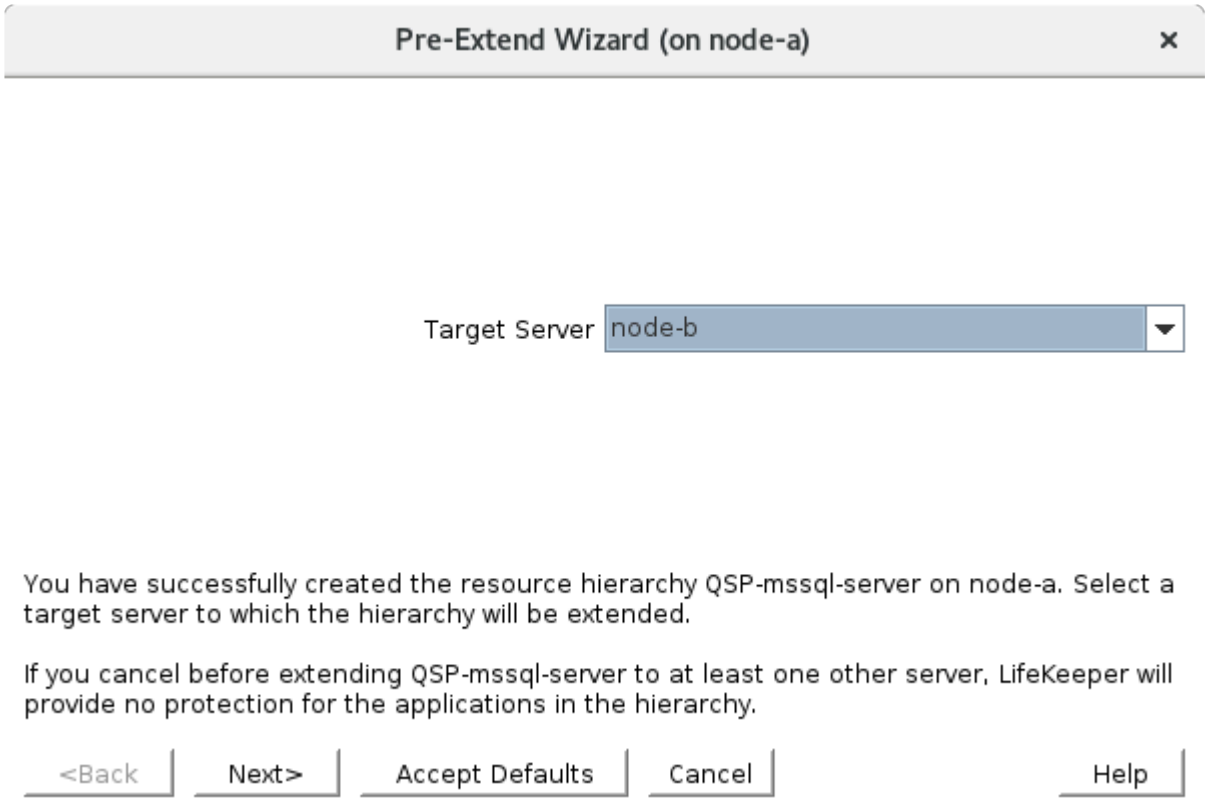

4. 以下の値を入力します。デフォルト値を選択します。

| フィールド | 値 |
|---|---|
| Enable or Disable Monitoring | enable |
| Resource Tag | QSP-mssql-server |

5. ウィザードは、値を確認します。確認が完了したら、「Next >」を選択して続行します。



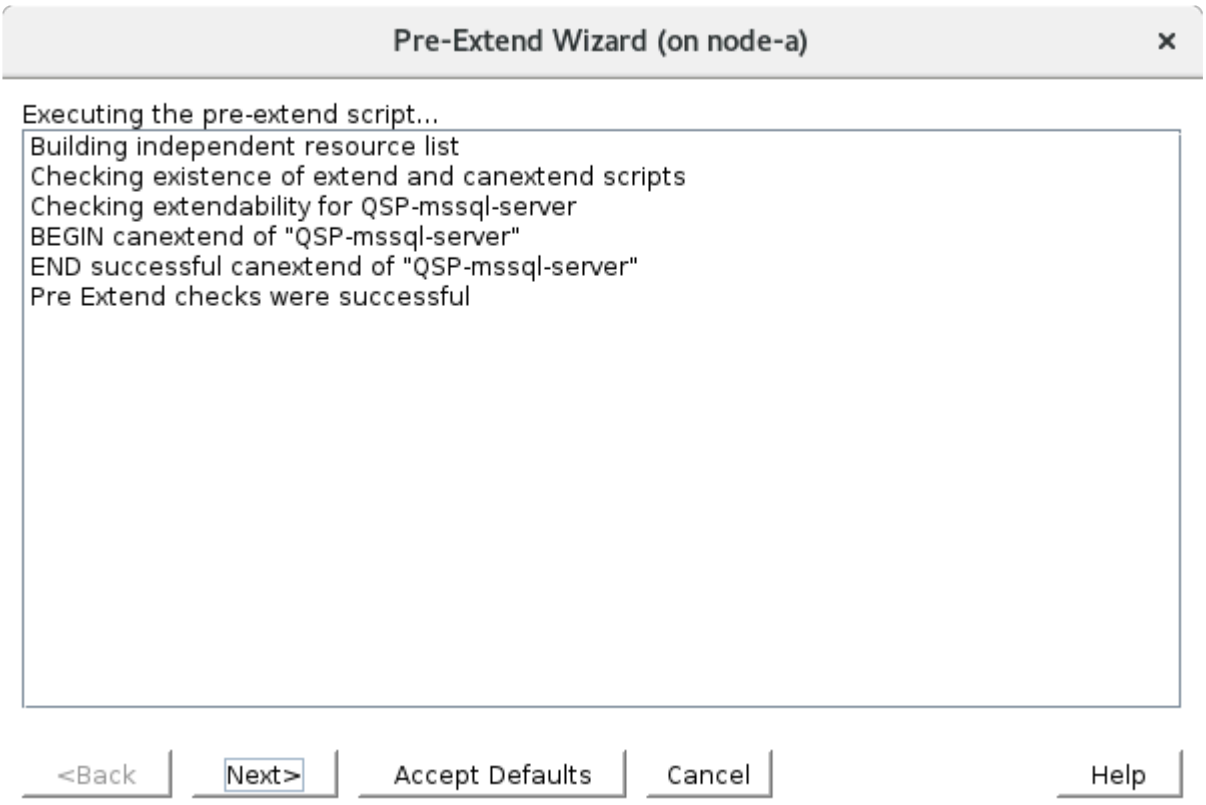

6. 次のステップは「Pre-Extend Wizard」です。デフォルト値を選択します。



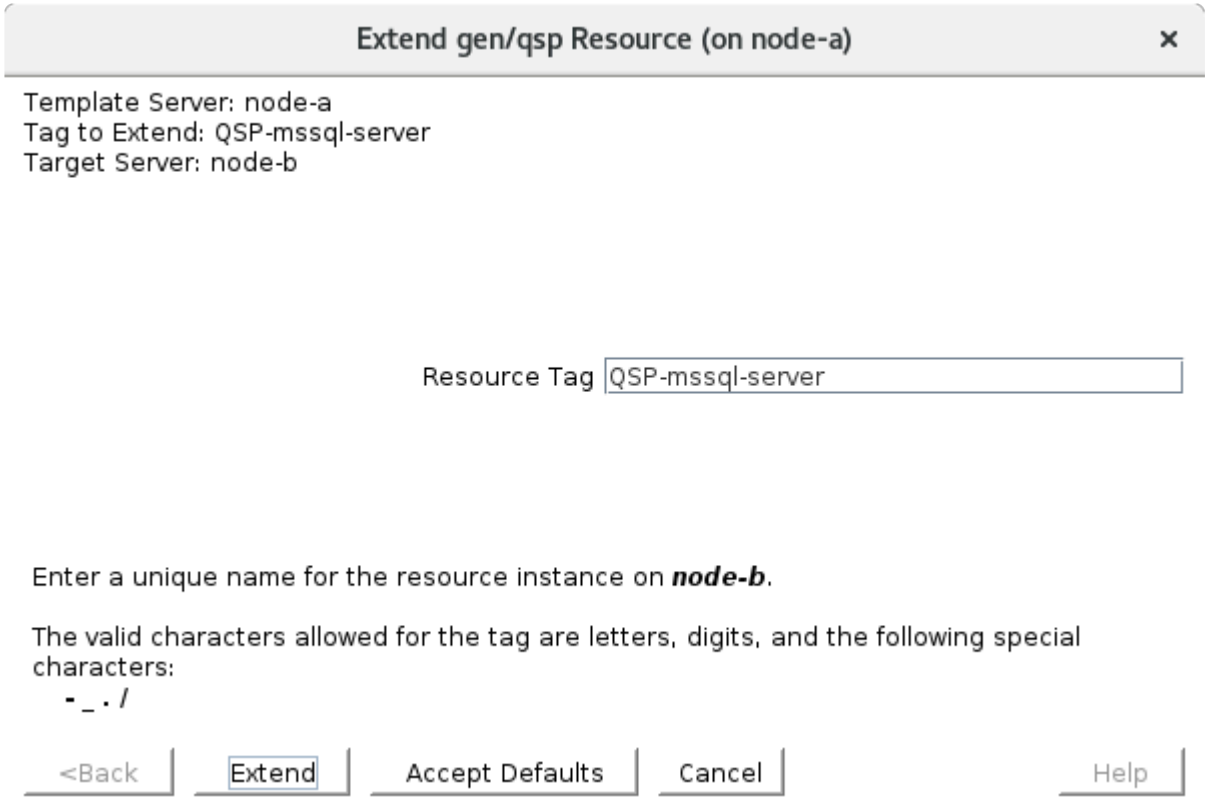

| フィールド | 値 |
| --- | --- |
| Switchback Type | intelligent |

| | |
|---|---|
| Template Priority | 1 |
| Target Priority | 10 |

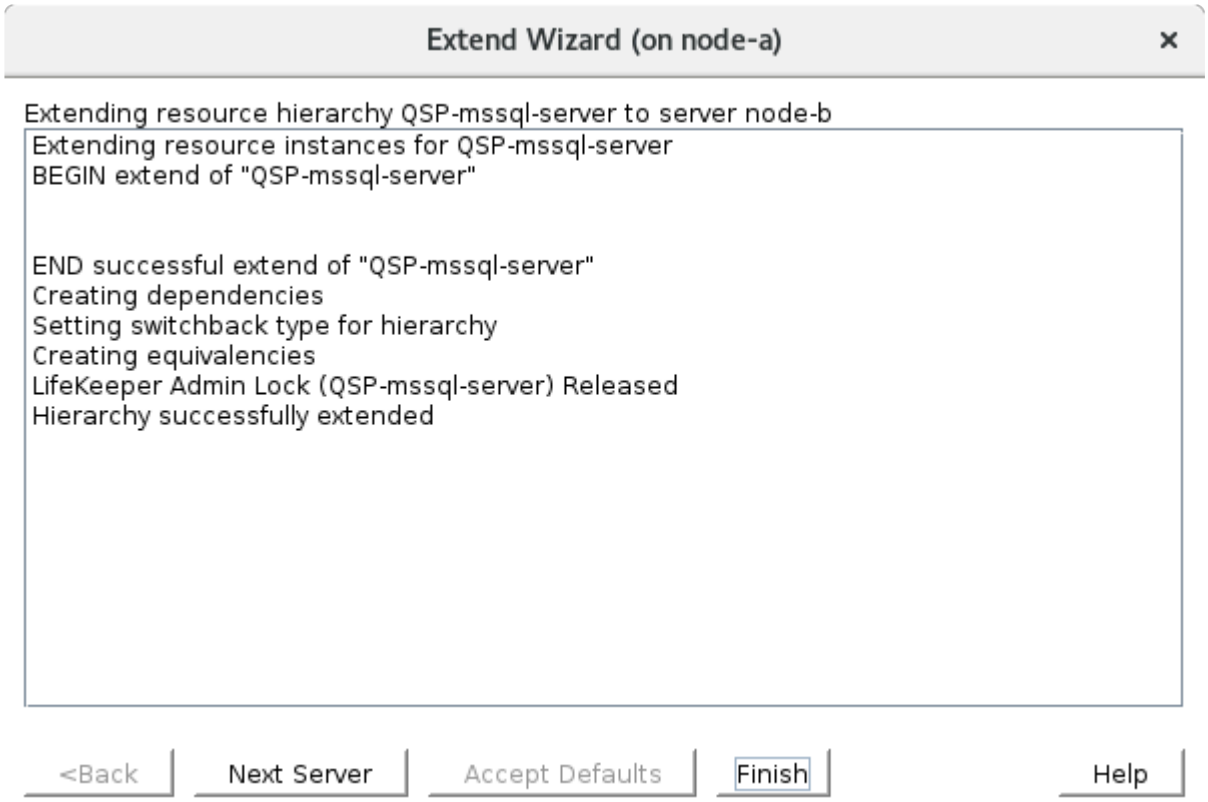
「Pre-Extend Wizard」のチェックが完了して合格したら、「Next」を選択して続行します。



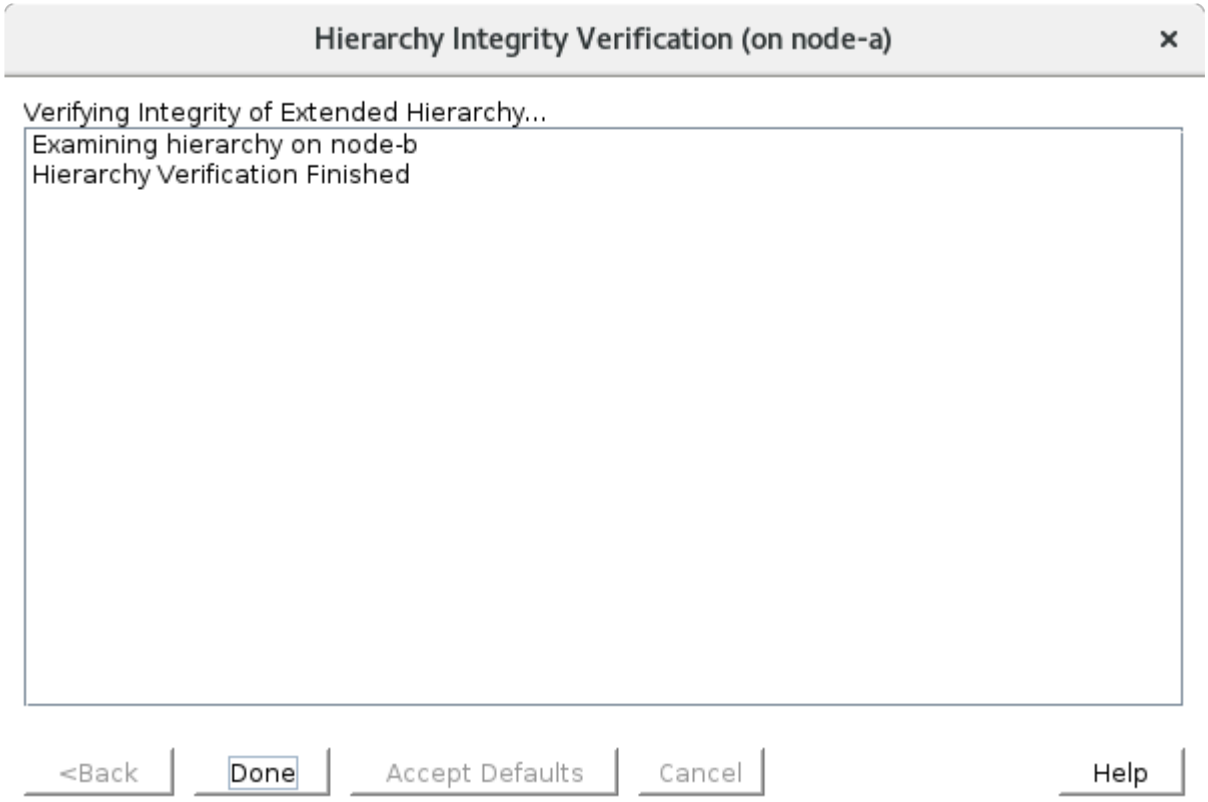

7. 「Extend gen/qsp Resource」ウィザードでは、まずDatabase Tagを選択します。デフォルト値を確認し、[Extend] をクリックします。



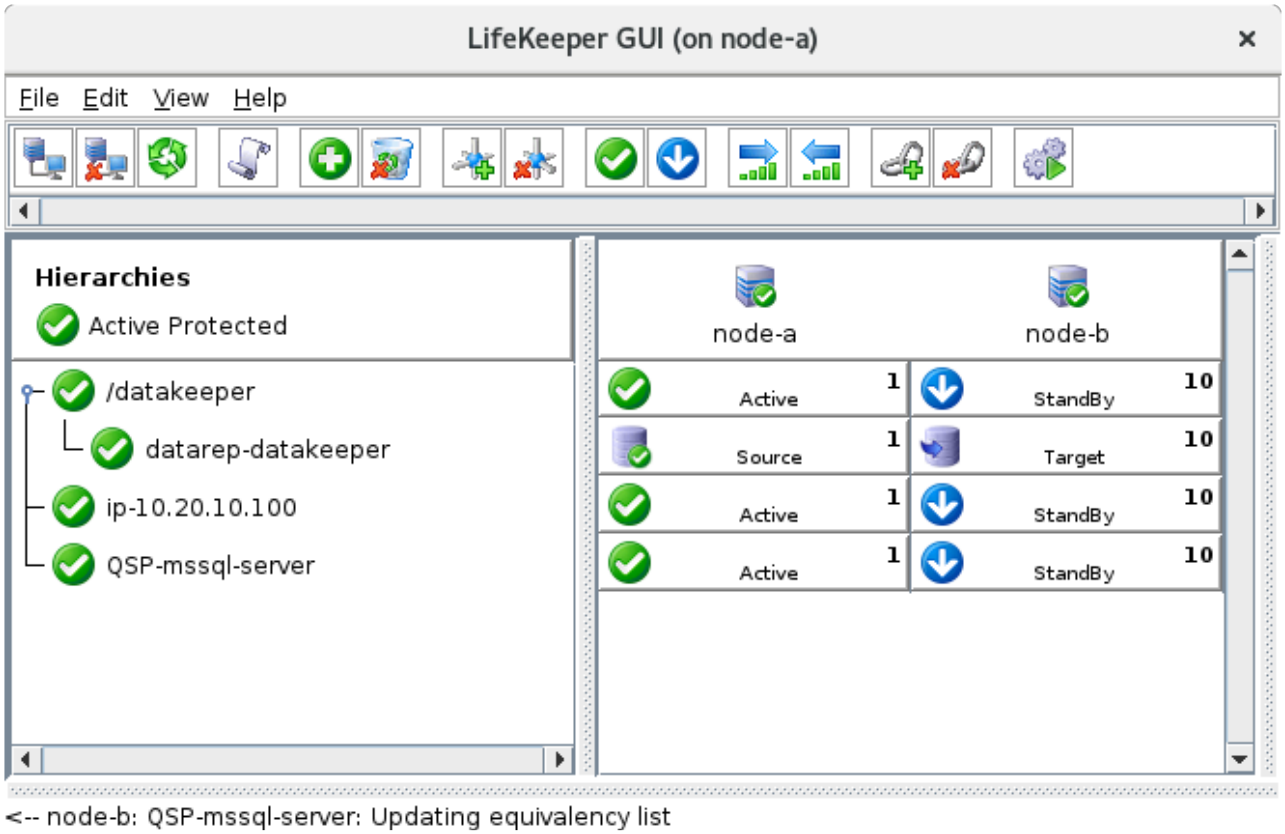

8. Extend Wizardでgen/qspリソースを構成したら、[Finish] をクリックして完了します。




9. これで、階層が正常に作成および検証されました。[Done] をクリックして完了します。




10. LifeKeeper GUIに戻ります。「QSP-mssql-server」リソースは次のように定義されています。



# 11.2.7.5.2.5. 両方のノードでLocalRecoveryパラメーターをカスタマイズする

Quick Service Protection Wizardにより、ほとんどの設定が完了しました。しかし、まだいくつかのパラメーターをカスタマイズする必要があります。

> node-aとnode-bの両方でこのコマンドを実行します。

まず、次のように lkpolicy ツールを使用していくつかのポリシーパラメーターを確認します。

```
# /opt/LifeKeeper/bin/lkpolicy -g -v
Server-level Policies:
    Failover: On
    LocalRecovery: On
Resource-level Policies:
    QSP-mssql-server:
        Failover: On
        LocalRecovery: On    <--- This needs to be updated
    /datakeeper:
        Failover: On
        LocalRecovery: On
    datarep-datakeeper:
        Failover: On
        LocalRecovery: On
    ip-10.20.10.100:
        Failover: On
        LocalRecovery: On
```

QSP-mssql-serverの LocalRecovery パラメーターは現在オンに設定されており、オフに変更する必要があります。次のコマンドで変更します。

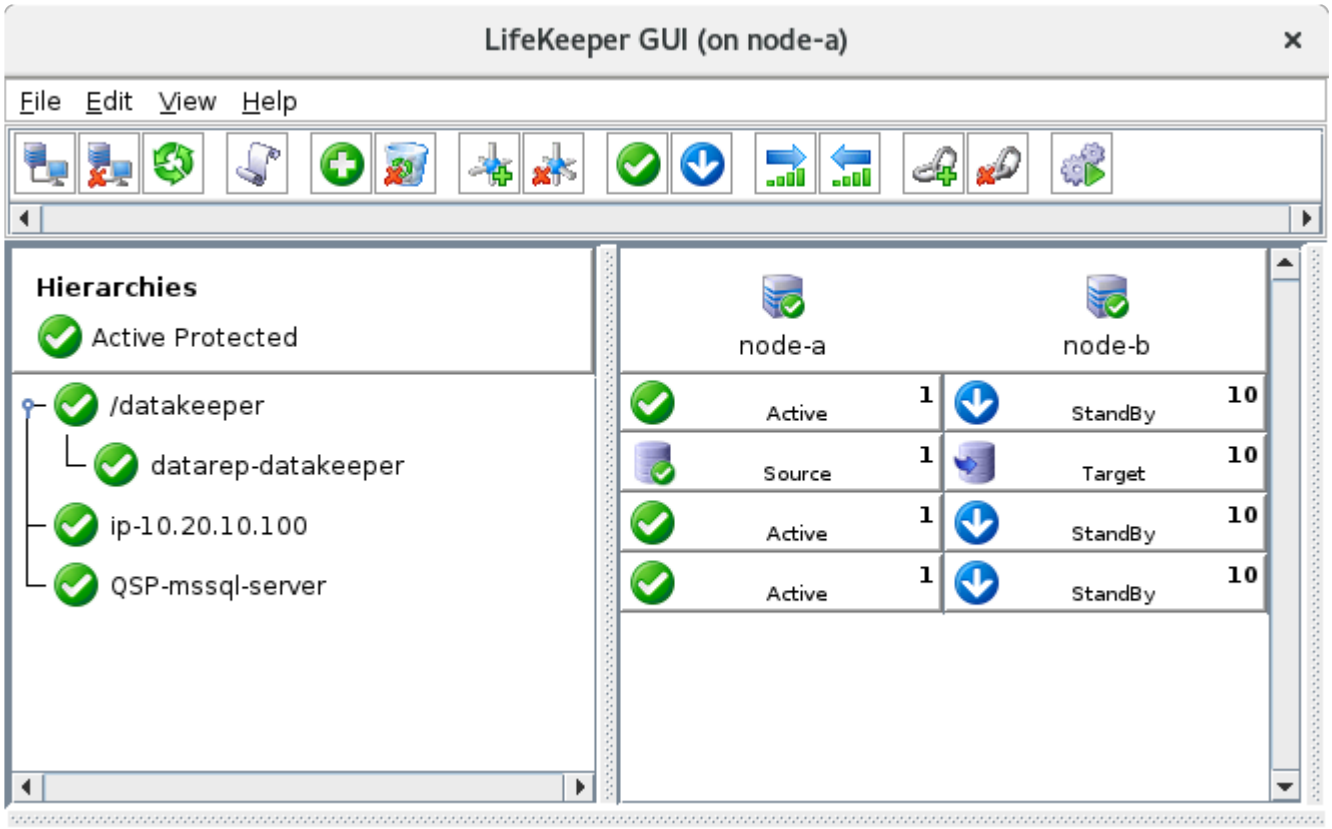
```
# /opt/LifeKeeper/bin/lkpolicy -s LocalRecovery -E tag="QSP-mssql-server"
Policy Disabled
```

値が更新されたら、次のように確認します。

```
# /opt/LifeKeeper/bin/lkpolicy -g -v
Server-level Policies:
    Failover: On
    LocalRecovery: On
Resource-level Policies:
    QSP-mssql-server:
        Failover: On
        LocalRecovery: Off
    /datakeeper:
        Failover: On
        LocalRecovery: On
    datarep-datakeeper:
        Failover: On
        LocalRecovery: On
    ip-10.20.10.100:
        Failover: On
        LocalRecovery: On
```

# 11.2.7.5.2.6. リソース間の依存関係を更新する

この時点で、LifeKeeper GUIには、以下に示すようにMSSQL関連のリソースが表示されます。

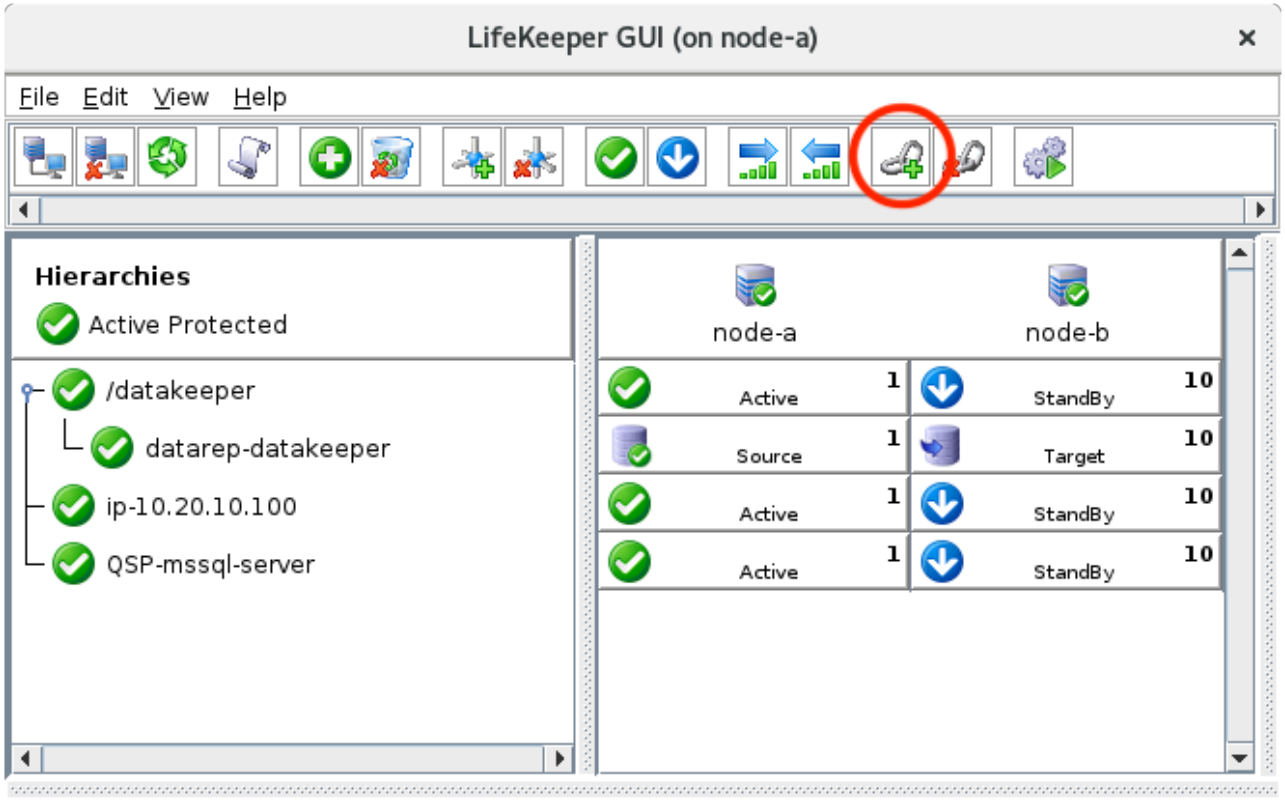



QSP-mssql-serverリソースは、次のリソースに依存しています。

- ip-10.20.10.100（データベースに接続するために、接続する仮想IPが必要です）
- /datakeeper（データベースを動作させるために、基盤となるファイルシステムをマウントする必要があります）

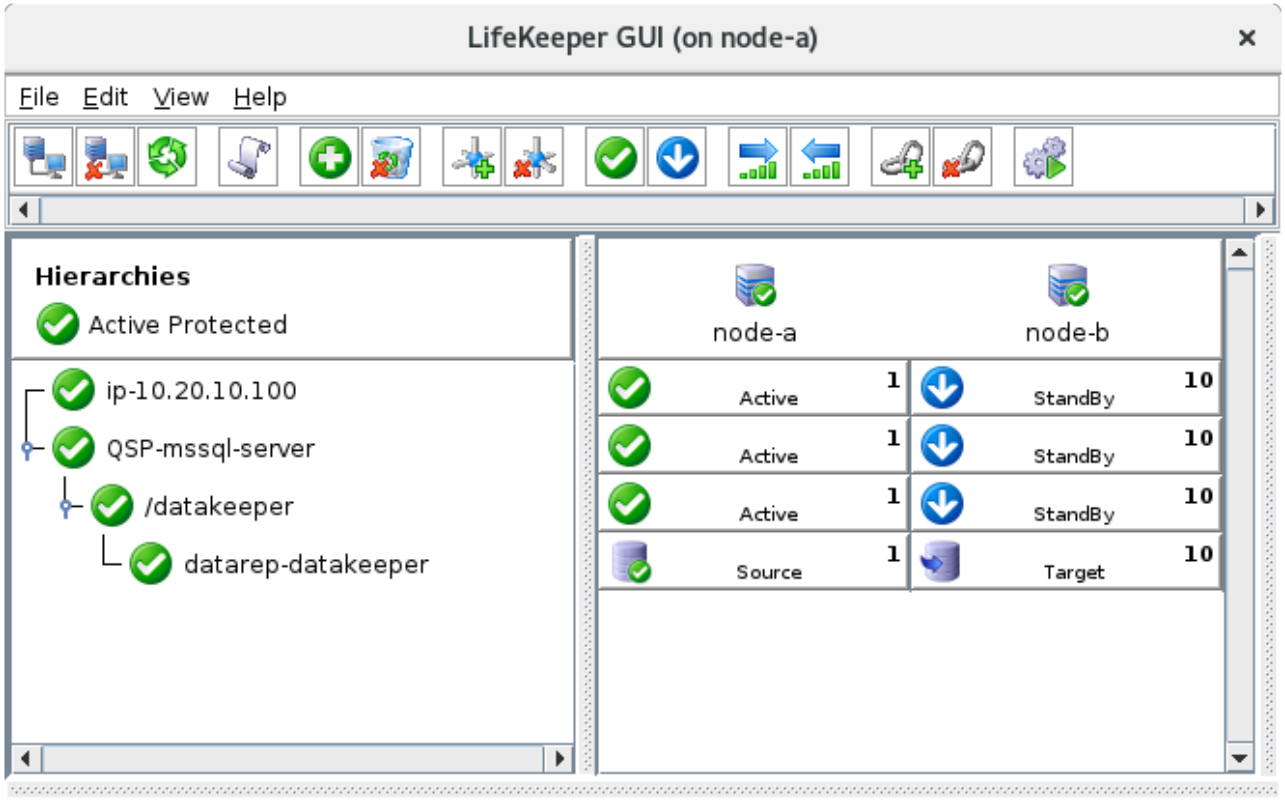
ここでは、リソース間の依存関係を定義する方法について説明します。

1. 「Create Dependency」を選択します。

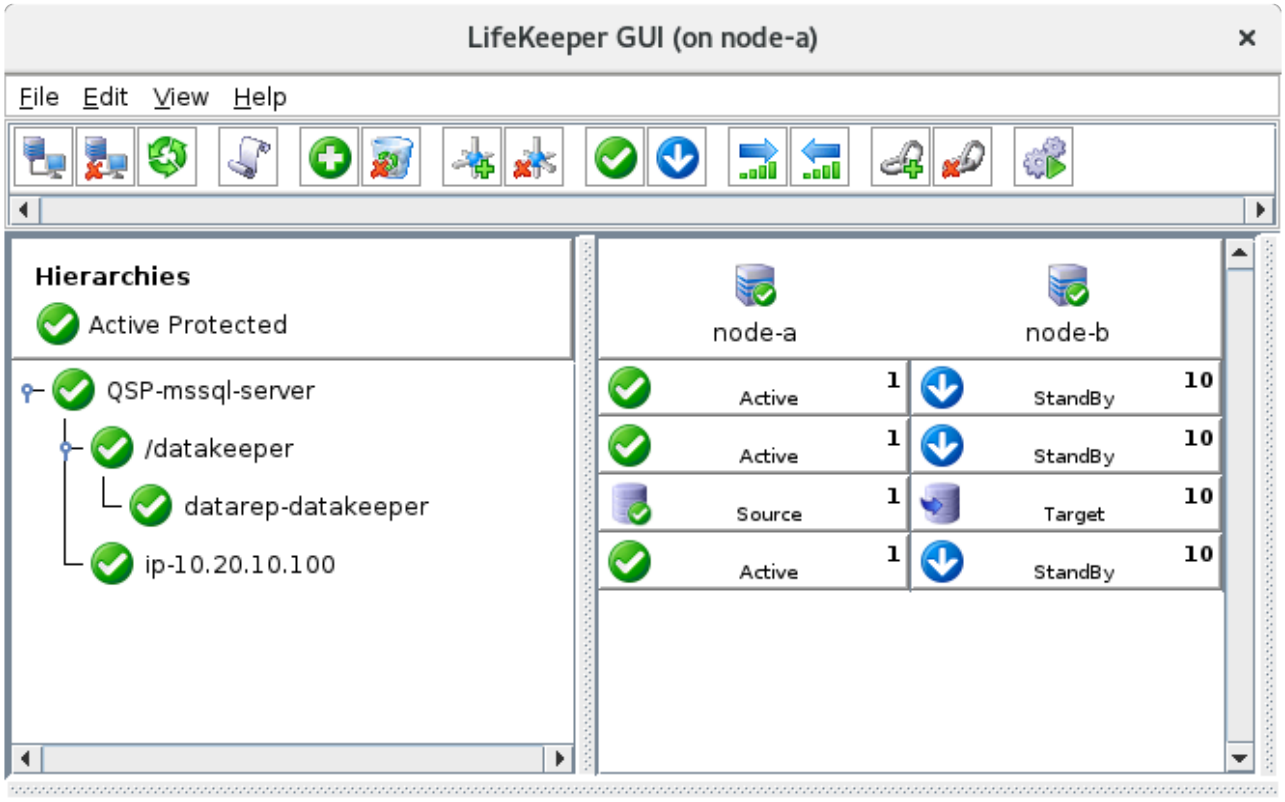



2. サーバーに「node-a」を選択します。
3. 親リソースタグに「QSP-mssql-server」を選択します。
4. 子リソースタグに「datakeeper」を選択します。

   依存関係が定義されると、LifeKeeper GUIは次のように関係を表示します。




5. 「ip-10.20.10.100」でこの手順を繰り返します。これで、依存関係が定義されました。




<-- node-b: ip-10.20.10.100: Adding parent: QSP-mssql-server

# 11.2.7.5.3. PostgreSQLリソースの保護

! ここでは、Postgresデータを格納する場所として、 datakeeper/pgsql/data/ を使用します。以下の手順では、このディレクトリは複製されたストレージ上に作成されます。これ以外のディレクトリ構造を作成する場合は、以下で説明する残りの手順全体で、環境の違いを考慮してください。

Postgres 12を設定する前に、 “LinuxノードへのPostgres12のインストール” の指示に従ってください。

✻ PostgreSQLをプライマリーノードとスタンバイノードの両方にインストールしたら、*プライマリーノードでのみ*次の手順を実行します。

## プライマリーノードにデータを保存するためのディレクトリを作成する

```
[root@node-a]# mkdir -m 700 -p /datakeeper/pgsql/data
[root@node-a]# chown postgres:postgres -R /datakeeper/pgsql
```

## プライマリーノードでPostgresデータベースサーバーを初期化する

この例では、 initdb は /usr/pgsql-12/bin/initdb にあります。 initdb の場所を使用して、ここでパラメーターを作成します。

```
[root@node-a ~]# su postgres -c "/usr/pgsql-12/bin/initdb -D /datakeeper/pgsql/data"
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.

(snip)

Success. You can now start the database server using:

    /usr/pgsql-12/bin/pg_ctl -D /datakeeper/pgsql/data -l logfile start
```

✻ 以下の手順で、使用する場所を確認します。

| アイテム | 場所 |
| --- | --- |

| | |
|---|---|
| Location of Postgres binaries | /usr/pgsql-12/bin/ |
| Data | /datakeeper/pgsql/data |

## プライマリーノードでPostgresプロセスを開始する

```
[root@node-a ~]# su postgres -c "/usr/pgsql-12/bin/pg_ctl start -D /datakeepe
r/pgsql/data"
waiting for server to start....2020-12-30 03:40:20.925 UTC [6346] LOG:  starti
ng PostgreSQL 12.5 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 2015062
3 (Red Hat 4.8.5-39), 64-bit
2020-12-30 03:40:20.926 UTC [6346] LOG:  listening on IPv6 address "::1", por
t 5432
2020-12-30 03:40:20.926 UTC [6346] LOG:  listening on IPv4 address "12
7.0.0.1", port 5432
2020-12-30 03:40:20.932 UTC [6346] LOG:  listening on Unix socket "/var/run/po
stgresql/.s.PGSQL.5432"
2020-12-30 03:40:20.946 UTC [6346] LOG:  listening on Unix socket "/tmp/.s.PGS
QL.5432"
2020-12-30 03:40:20.960 UTC [6346] LOG:  redirecting log output to logging col
lector process
2020-12-30 03:40:20.960 UTC [6346] HINT:  Future log output will appear in dir
ectory "log".
 done
server started
```

以下の手順で、使用する以下の場所を確認します。

| アイテム | 場所 |
|---|---|
| Unix socket | /tmp/.s.PGSQL.5432 |

## LifeKeeperでPostgreSQLリソースを保護する

1. LifeKeeperユーザーインターフェースで、 を選択します。
2. Create Resource Wizard (on node-a) が表示されます。Recovery Kitに「PostgreSQL Database」を選択します。
3. 次のパラメーターを選択します。

| アイテム | 場所 |
|---|---|

| | |
|---|---|
| Switchback Type | intelligent |
| Server | node-a |
| PostgreSQL Executable Location | `/usr/pgsql-12/bin/` |
| PostgreSQL Client Executable Location | /usr/pgsql-12/bin/psql |
| PostgreSQL Administration Executable Location | /usr/pgsql-12/bin/pg_ctl |
| PostgreSQL Data Directory | `/datakeeper/pgsql/data` |
| PostgreSQL Port | 5432 |
| PostgreSQL Socket Path | `/tmp/.s.PGSQL.5432` |
| Enter Database Administrator User | postgres |
| PostgreSQL Logfile | /tmp/pgsql-5432.lk.log |
| PostgreSQL Database Tag | pgsql-5432 |

ウィザードはこれらの値をチェックします。node-a上に「pgsql-5432」が正常に作成されたら、次のステップに進みます。

4. Pre-Extend Wizard @ node-aで、以下の値を選択します。

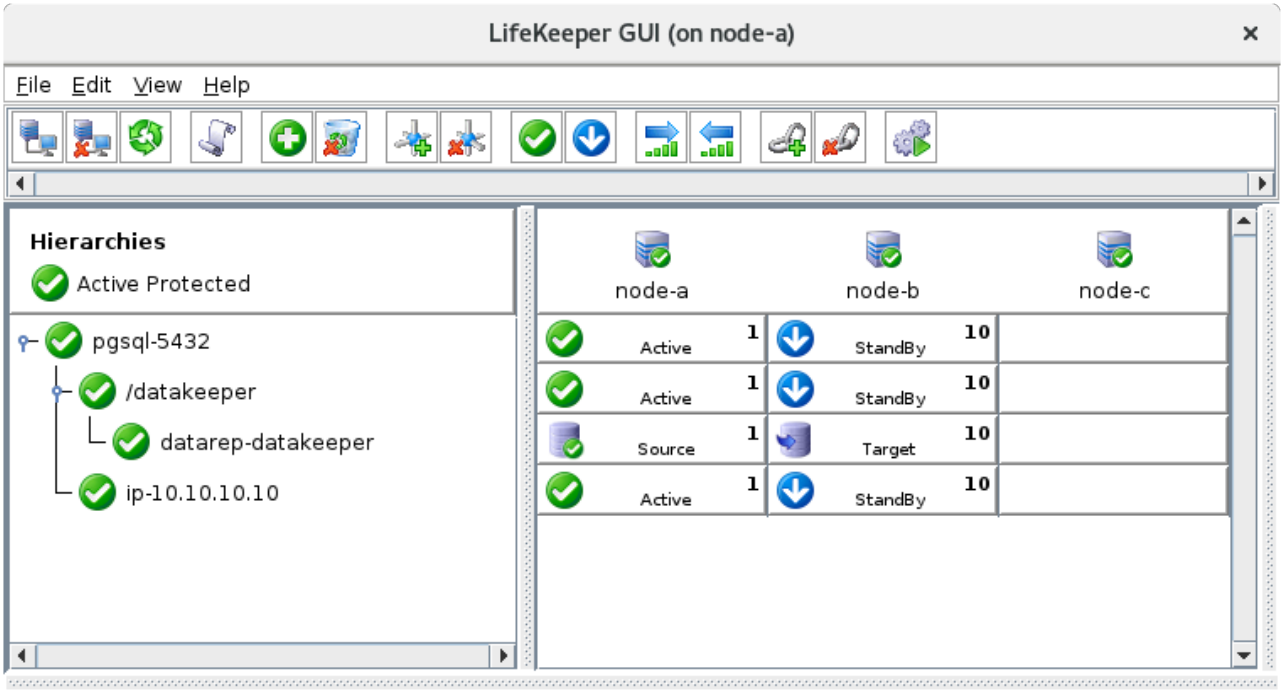
| アイテム | 場所 |
|---|---|
| Target Server | node-b |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |

ウィザードはこれらの値をチェックします。Pre-Extendのチェックが完了したら、次のステップに進みます。

5. Extend database/pgsql Resource @ node-aウィザードで、以下の値を選択します。

| フィールド | 値 |
|---|---|
| PostgreSQL Executable Location | `/usr/pgsql-12/bin/` |
| PostgreSQL Database Tag | pgsql-5432 |

これで、PostgreSQLリソースがLifeKeeperで定義されました。ウィザードは、PostgreSQLリソースと複製されたディスク（/datakeeper）間の依存関係を自動的に定義します。




<-- node-b: ip-10.10.10.10: Adding parent: pgsql-5432

# 11.2.7.5.3.1. LinuxノードにPostgreSQL 12をインストールする

このセクションでは、LinuxノードにPostgreSQL 12をインストールするために必要な基本的な手順の概要を説明します。

PostgreSQLはアクティブノードとスタンバイノードの両方にインストールする必要があります。

- PostgreSQL のインストール方法は、https://www.postgresql.org/download/ に掲載されています。

Linuxのディストリビューション、バージョン、アーキテクチャを選択することで、カスタマイズされたインストール手順が提供されます。製品をインストールする直前までは、その指示に従ってください。
`initdb` はnode-aとnode-bで呼び方が異なるので、その手順については親ページで説明します。
RedHat Enterprise Linux 8またはその互換OSの場合は、次のセクションを実行します。

```
# Install the repository RPM:
# dnf install -y https://download.postgresql.org/pub/repos/yum/reporpms/EL-8-x86_64/pgdg-redhat-repo-latest.noarch.rpm
# Disable the built-in PostgreSQL module:
# dnf -qy module disable postgresql
# Install PostgreSQL:
# dnf install -y postgresql12-server
```

initdbを実行する前のステップで止めてください。

## Postgresユーザーにパスワードを設定する

```
1 # passwd postgres
```

## データを格納するディレクトリを/datakeeper/pgsql/data下に変更する

```
# cp -p /var/lib/pgsql/.bash_profile /var/lib/pgsql/.bash_profile.org
# vi /var/lib/pgsql/.bash_profile
# cat /var/lib/pgsql/.bash_profile
[ -f /etc/profile ] && source /etc/profile
PGDATA=/datakeeper/pgsql/data
export PGDATA
# If you want to customize your settings,
# Use the file below. This is not overridden
# by the RPMS.
[ -f /var/lib/pgsql/.pgsql_profile ] && source /var/lib/pgsql/.pgsql_profile
```

# 11.2.7.5.3.2. NFSリソースの保護

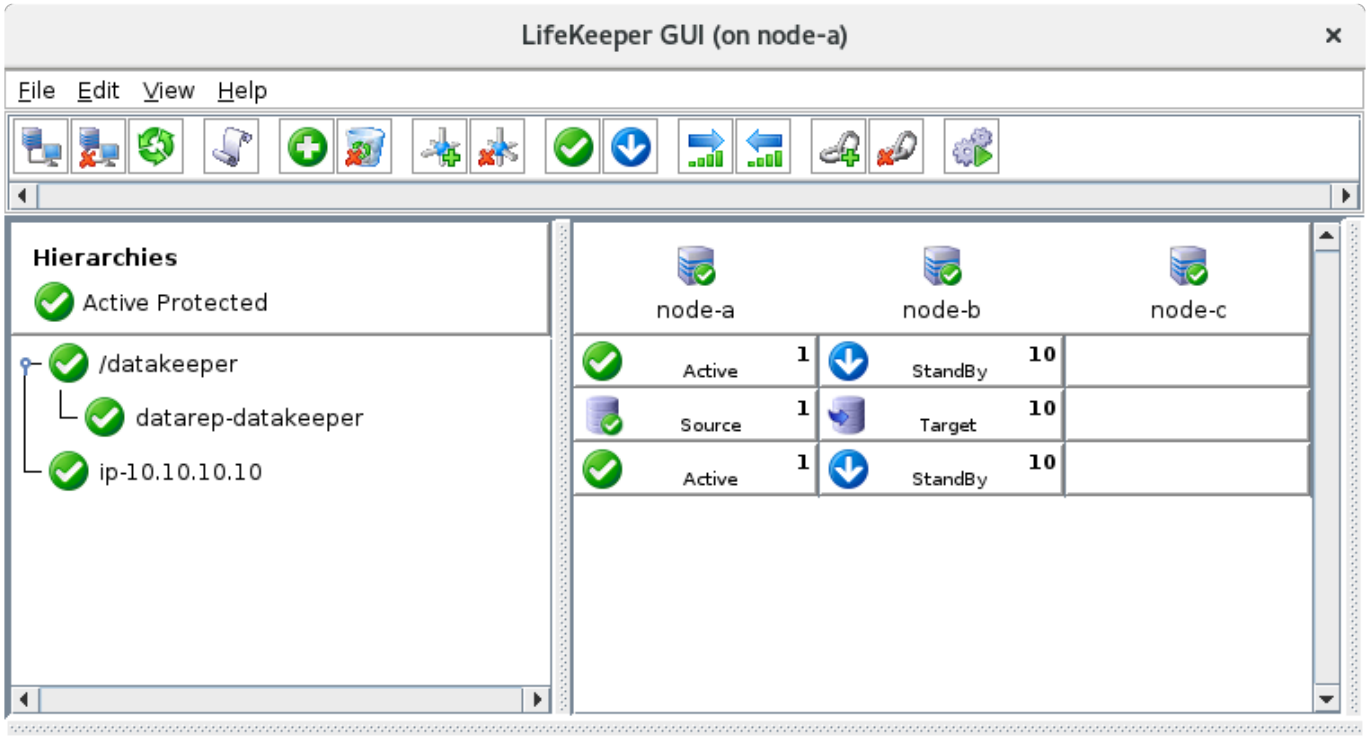
このセクションでは、NFSサーバーのクラスターを保護するために必要な手順について説明します。

## はじめる前に

このセクションでは、クラスターに次の2つのリソースが構成されていることを前提としています。

- “IP リソース”:#creating-an-ip-resource
- `/datakeeper“ でのデーターレプリケーションリソース”:#how-to-create-data-replication-of-a-file-system`

NFSソフトウェアをインストールする前のLifeKeeper GUIは以下のような状態です。




<-- node-b: Updating app/res type: scsi i2o

## NFSサーバーソフトウェアのインストール

NFSサーバーとクライアントユーティリティを両方のノードにインストールします。RHEL 8.xでは、これらは `nfs-utils` パッケージによって提供され、SLES 15.xでは、`nfs-kernel-server` パッケージとその依存関係によって提供されます。

たとえば、RHEL 8.xでは、node-aとnode-bの両方に `nfs-utils` をインストールします。

```
# yum -y install nfs-utils
(snip)
Complete!
```

## NFSサーバーの設定

node-aとnode-bの両方で必要なNFS関連サービスを開始し、それらを有効にして、起動時に自動的に開始されるようにします。

```
# systemctl enable --now rpcbind nfs-server
```

> 以下の手順は、node-aで*のみ*行ってください。

`/etc/exports` を編集して、NFSエクスポートを定義します。ここでは、DataKeeperによって保護されている共有ストレージ（ `/datakeeper` ）を使用します。 `/datakeeper/nfs/data` を作成し、エクスポート先として使用します。 `/etc/exports` ファイルを編集すると、以下のようになります。

```
[root@node-a ~]# mkdir -p /datakeeper/nfs/data
[root@node-a ~]# vi /etc/exports
[root@node-a ~]# cat /etc/exports
/datakeeper/nfs/data 10.20.0.0/22(rw,no_root_squash)
```

node-aで `/etc/exports` に追加された共有ファイルシステムをエクスポートします。

```
[root@node-a ~]# exportfs -rav
```

## クライアントからNFSサーバーへのアクセスを確認する

> 次の手順は、クライアントマシンで実行する必要があります。

クライアントマシン（node-cなど）で、次の手順を実行してNFSソフトウェアをインストールします。node-aからローカルフォルダーにNFSエクスポートをマウントします。（例： `/local/nfsclient` にマウント）

```
[root@node-c ~]# yum -y install nfs-utils
(snip)
Complete!
[root@node-c /]# mkdir -p /local/nfsclient
[root@node-c ~]# mount -t nfs -o nfsvers=4 node-a:/datakeeper/nfs/data /local/nfsclient
```

これで、NFSサーバーのコンテンツが表示されます。NFSサーバーから切断（アンマウント）して、NFSクラスターを構成します。

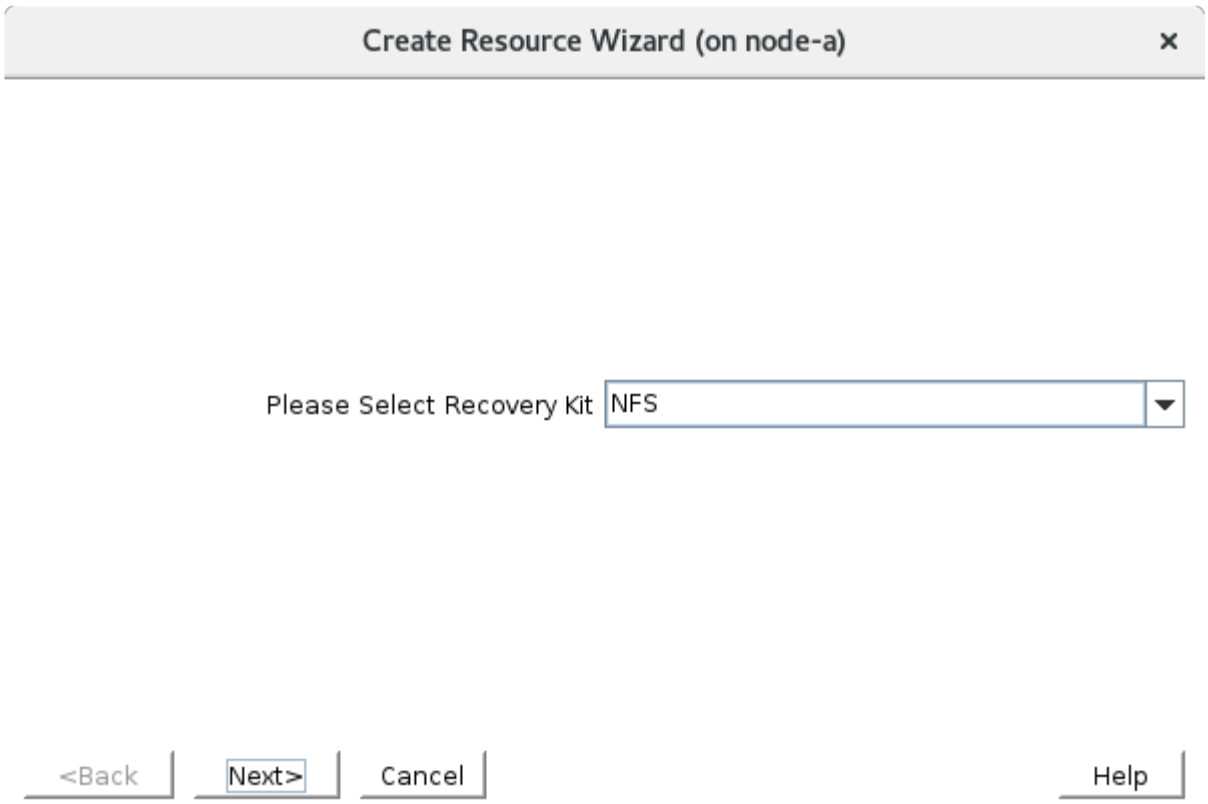
```
[root@node-c ~]# umount /local/nfsclient
```

## LifeKeeperを使用してNFSリソースを保護する

node-aで次の手順を実行してください。

1. LifeKeeperユーザーインターフェースで、 を選択します。

2. Create Resource Wizard (on node-a) が表示されます。Recovery Kitに「NFS Recovery Kit」を選択します。




3. 以下のパラメーターを選択します。

| アイテム | ロケーション |
|---|---|
| Switchback Type | intelligent |
| Server | node-a |
| Export Point | /datakeeper/nfs/data |
| IP Tag | ip-10.10.10.10 |
| NFS Tag | nfs-/datakeeper/nfs/data |

ウィザードはこれらの値をチェックします。node-aで「nfs-/datakeeper/nfs/data」が正常に作成されたら、次の手順に進みます。

4. Pre-Extend Wizard@node-aで次の値を選択します。

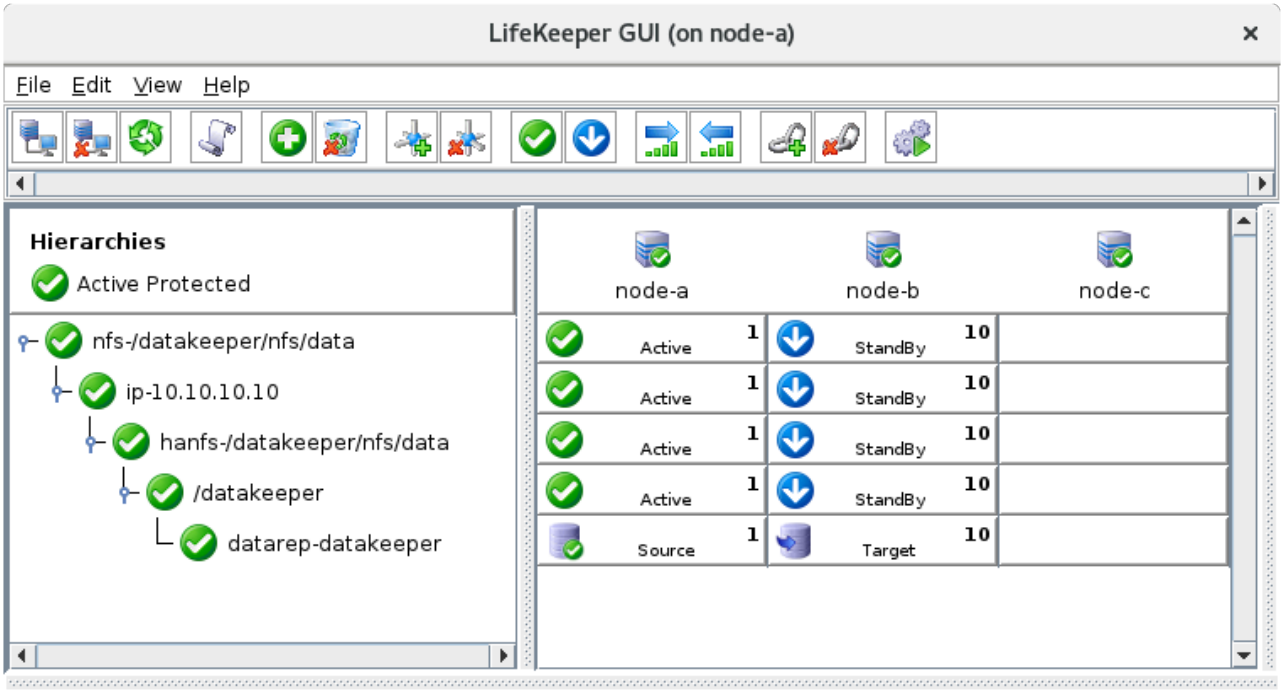
| フィールド | 値 |
|---|---|
| Target Server | node-b |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |

ウィザードはこれらの値をチェックします。Pre-Extend Checkが完了したら、次の手順に進みます。

5. Extend gen/nfs Resource Hierarchy @ node-aウィザードで、次の値を選択します。

| フィールド | 値 |
|---|---|
| NFS Tag | nfs-/datakeeper/nfs/data |

NFSリソース階層がLifeKeeperで定義されました。ウィザードは、NFSリソースと必要なリソース（IPリソースおよび /datakeeper リソース）間の依存関係を自動的に定義します。




<-- node-w: Updating server state to: alive

# 11.2.7.5.4. Protecting MySQL Resources

## Objective

This document is intended to aid you in installing, configuring and using the LifeKeeper for Linux evaluation product to make MySQL highly available. If MySQL is not already installed, please allocate some time to install it on your servers. Once this task has been completed, you may install and configure LifeKeeper for Linux.

There are five phases in this process:

- Prepare to Install
- Configure Storage
- Install, Configure and Start MySQL
- Install LifeKeeper for Linux – MySQL Cluster
- Configure the Cluster

# 11.2.7.5.4.1. Prepare to Install

## Hardware Requirements

### Primary and Secondary Servers

- Systems must meet the minimum requirements for the Linux distribution to be used during the evaluation.
- 512MB RAM minimum; 1GB RAM recommended.
- 2GB of available hard disk space recommended.
- Multiple Network Interface Cards (NIC's) are recommended.
- Configure one or more additional partitions to be used for data replication. On the primary server, these will become the source partitions. On the secondary server(s), these will become the target partitions. In this evaluation example we will be replicating MySQL data (which will be a partition mounted at /var/lib/mysql)
- For replicated partitions, a target partition's size must equal to or larger than the size of its source partition.
- The system ( / ) and boot (/boot) partitions are not eligible for replication.

**Note:** You may use more than one partition for replicated data, allowing for separation of multiple database files and/or log directories for performance reasons.

### Client

This system is not required but is recommended for testing the cluster environment.

- A standard linux terminal running the MySQL client can be used to test the configuration.

## Software Requirements

### Primary Server and Secondary Server

Please refer to the Support Matrix for a full list of supported Operating Systems

- Current patches / security updates are recommended
- Satisfied dependencies; especially if the Linux installation package selection was base/minimal you will need to refer to the dependencies documentation at Linux Dependencies
- It is recommended that IPtables is disabled

- # /etc/init.d/iptables off
- # chkconfig iptables off
- See Running LifeKeeper With a Firewall for information regarding the ports LifeKeeper for Linux uses.

> Note: Several OS commands and database specific commands will require administrator (root) permissions.

> Note: The use of iptables or firewalld will depend on the version of the OS installed.

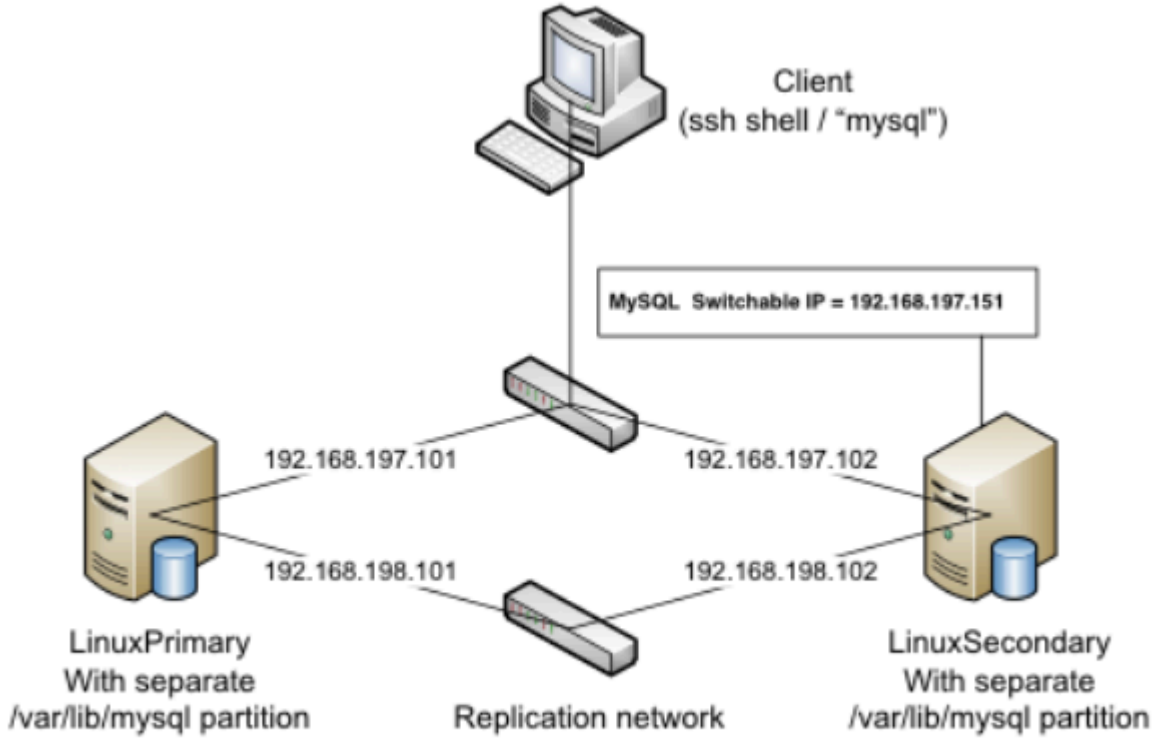
- Disable SELinux :
  - Edit /etc/selinux/config
  - Set SELINUX=disabled (note: permissive mode is also acceptable)
- Check the configuration of your /etc/hosts file
  - localhost.localdomain and localhost are the only entries that can be on 127.0.0.1
  - Create a separate entry for your hostname with a static address
- GUI Authentication with PAM
  - LifeKeeper for Linux now leverages the Pluggable Authentication Module (PAM) provided in the Linux Standard Base (LSB).
  - Users are identified and authenticated against the system's PAM configuration. Privilege levels are determined from group membership as provided through PAM.
  - In order to access the GUI, a user must be a member in one of the three LifeKeeper groups: lkadmin, lkoper or lkguest.
  - See the following URL for more information on this topic:
    - See Configuring GUI Users for more information.

# Network Requirements

For your evaluation, we recommend configuring your machines similarly to the following example.

LinuxPrimary and LinuxSecondary are multi-homed, between two LAN segments (the second NIC in each server could even be connected via a cross-over cable if a second physical network is not available). The second NIC is optional in this configuration, but highly recommended in production environments to avoid a single point of failure.

In this example evaluation scenario we will be leveraging local, replicated storage.




**Network Configuration Example**

## Primary Server and Secondary Servers

- Configure the Host file with entries for all LifeKeeper protected servers. This is typically /etc/hosts.

  Example:
  192.168.197.101 LinuxPrimary
  192.168.197.102 LinuxSecondary

- See your Network Administrator to obtain an unused IP Address to be used as the switchable IP Address. This switchable IP Address will be created later in the configuration process.
- Public Network connection(s) configured with:
  - Static IP address

    - Correct subnet mask
    - Correct gateway address
    - Correct DNS server address(es)

- Private Network connection(s) configured with:
    - Static IP address (on a different subnet from the public network)
    - Correct network mask
    - No gateway IP address
    - No DNS server addresses

## Client

Must be able to communicate on the same subnet/network as the servers Public interface addresses. In our example, this is the 192.168.197.0/24 network.

# 11.2.7.5.4.2. Configure Storage

## Before You Begin

### Ensure the following:

- You have an extra disk/partition on both servers that can be used for data replication. A target volume's size must be equal to or larger than the size of its source disk/partition.
- If /datakeeper/lib/mysql does not exist, create the directory.
- Refer to this Known Issue if you run into an odd sector issue while configuring the partition for MySQL.

## Partition local storage for use with SIOS Data Replication

### Primary Server

On your Primary server, perform the following actions:

1. Identify an existing free, unused disk partition to use as our mysql/mariadb data directory. Alternatively, create a new partition. Use the "gdisk" utility to partition your disk appropriately. In this example /dev/sdb is an unused disk where we will create a single partition

   a. gdisk /dev/sdb

   b. Press "n" to create a new partition

   c.This example uses a new disk, so we will use all default values (Partition 1, entire disk and Linux filesystem partition type) Hit Enter four times to confirm these parameters.

   d. Press "w" to write the partition table

   e. Press "Y" to confirm to overwrite existing partitions

**Example**

[root@LinuxPrimary ~]# **gdisk /dev/sdb**

GPT fdisk (gdisk) version 0.8.10

Partition table scan:
MBR: protective
BSD: not present
APM: not present
GPT: present

Found valid GPT with protective MBR; using GPT.

Command (? for help): **n**
Partition number (1-128, default 1): **<enter>**
First sector (34-2047, default = 34) or {+-}size{KMGTP}: **<enter>**
Last sector (34-2047, default = 2047) or {+-}size{KMGTP}: **<enter>**
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300): **<enter>**
Changed type of partition to 'Linux filesystem'

Command (? for help): **w**

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING PARTITIONS!!

Do you want to proceed? (Y/N): **Y**
OK; writing new GUID partition table (GPT) to /dev/sdb.
Warning: The kernel is still using the old partition table.
The new table will be used at the next reboot.
The operation has completed successfully.

[root@LinuxPrimary ~]#

2. Format the newly created disk partition

   # mkfs.ext3 /dev/sdb1

3. Mount the partition temporarily at /mnt

   # mount /dev/sdb1 /mnt

4. Move any existing data from /datakeeper/lib/mysql/ into this new disk partition (assumes a default MySQL configuration)

   # cd /datakeeper/lib/mysql
   # mv * /mnt

5. Remount /dev/sdb1 at /datakeeper/lib/mysql

   # cd /root
   # umount /mnt
   # mount /dev/sdb1 /datakeeper/lib/mysql

6. Note: there is no need to add this partition to /etc/fstab. Lifekeeper will take care of mounting this automatically.

**Result:**

```
[root@LinuxPrimary ~]# df /datakeeper/lib/mysql
Filesystem 1K-blocks Used Available Use% Mounted on
/dev/sdb1 253855 11083 229666 5% /datakeeper/lib/mysql<
```

## Secondary Server

7. On your Secondary server, only perform Step #1 above, where you partition the disk. The size of the Target disk/partition needs to be the same size, or greater, than our Source disk/partition.

# 11.2.7.5.4.3. Install, Configure and Start MySQL

> Note: Several OS commands and database specific commands will require administrator (root) permissions.

> Note: Individual setup steps may vary due to OS.

## Primary Server

On your Primary server, perform the following actions:

1. Install both the “mariadb” and “mariadb-server” rpm packages if they do not exist on your system.

   Apply any required dependencies as well

   ```
   # yum install mariadb mariadb-server
   ```

2. Verify that your local disk partition is still mounted at /datakeeper/lib/mysql via the “df” command

3. If this is a fresh MySQL install, initialize a sample MySQL database:

   ```
   # /usr/bin/mysql_install_db --datadir=/datakeeper/lib/mysql --user=mysql
   ```

4. Ensure that all files in your MySQL data directory (/datakeeper/lib/mysql) have correct permissions and ownership

   ```
   # chown -R mysql:mysql /datakeeper/lib/mysql
   ```

   ```
   # chmod 755 /datakeeper/lib/mysql
   ```

5. Finally, manually start the MySQL daemon from the command line.

> **Do Not** start it via the “service” command, or the /etc/init.d/ scripts

```
# mysqld_safe --user=mysql --socket=/datakeeper/lib/mysql/mysql.sock --port=3306 --datadir=/datakeeper/lib/mysql &
```

6. Verify MySQL is running by connecting with the mysql client:

   To specify the socket run the following command.

mysql --socket=/datakeeper/lib/mysql/mysql.sock

[root@LinuxPrimary mysql]# **mysql**

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MariaDB connection id is 1

Server version: 5.5.68-MariaDB MariaDB Server

Copyright © 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)] > **exit**

Bye

[root@LinuxPrimary mysql]#

7. Perform the following steps.

   a. Launch mysql, using mysql, with the following command: mysql mysql --socket=/datakeeper/lib/mysql/mysql.sock

   b. Inside of the program ran: update user set Password=PASSWORD ('<new password>') where User='root'

   c. Then ran: flush privileges;

8. Verify your new password:

   # mysql mysql --socket=/datakeeper/lib/mysql -u root -p

   (Enter new password.)

   #exit

9. Create a MySQL configuration file. We will place this in the same shared directory (/datakeeper/lib/mysql/my.cnf)

   # vi /datakeeper/lib/mysql/my.cnf

**Example**

# cat /datakeeper/lib/mysql/my.cnf

[mysqld]

```
datadir=/datakeeper/lib/mysql

socket=/datakeeper/lib/mysql/mysql.sock

pid-file=/datakeeper/lib/mysql/mysqld.pid

user=root

port=3306

# Default to using old password format for compatibility with mysql 3.x

# clients (those using the mysqlclient10 compatibility package).

old_passwords=1

# Disabling symbolic-links is recommended to prevent assorted security risks;

# to do so, uncomment this line:

# symbolic-links=0

[mysqld_safe]

log-error=/datakeeper/log/mysqld.log

pid-file=/datakeeper/run/mysqld/mysqld.pid

[client]

user=root

password=new password
```

10. Delete the original MySQL configuration file, located in /etc

```
# rm /etc/my.cnf
```

# Secondary Server

On your Secondary Server:

1. Install both the “mariadb” and “mariadb-server” rpm packages if they do not exist on your system.

   Apply any required dependencies as well

```
# yum install mariadb mariadb-server
```

2. Delete the original MySQL configuration file, located in /etc

   ```
   # rm /etc/my.cnf
   ```

3. There is no need to perform any of the additional steps taken on the Primary Server.

# 11.2.7.5.4.4. Install LifeKeeper for Linux – MySQL Cluster

For ease of installation, SIOS has provided the LifeKeeper for Linux with an installation script. Towards the end of the script, the desired Application Recovery Kits (ARKs) should be selected for installation. The software will be installed to the following locations:

| SPS for Linux Component | Install Location |
|---|---|
| LifeKeeper Software | /opt/LifeKeeper |
| LifeKeeper Config File | /etc/default/LifeKeeper |

Perform the following actions on **both** Primary and Secondary server.

## Download Software

1. Open the LifeKeeper for Linux evaluation email you received from SIOS.
2. Download the LifeKeeper for Linux Software from the link provided in your email. It is generally easiest to use “wget” to recursively download all files. Example:

   a. # cd /root

   b. # wget -r <URL>

   c. After you have successfully downloaded the software you will have downloaded contents similar to the follow directory listing:

   [root@LinuxPrimary ~]# ls -l <directory> total 63680

   -rw-r—r— 1 root root 23163 May 30 14:03 EULA.pdf

   -rw-r—r— 1 root root 536 May 30 14:03 readme.txt

   -rw-r—r— 1 root root 65179648 May 30 14:03 sps.img
3. Download your Evaluation license key from the link specified in your evaluation email. Save the license file to an easy to remember location on both servers.

## Run the LifeKeeper Installer Script

1. Loopback mount the sps.img file previously downloaded, which is an ISO9660 image file. Run the “setup” script inside:

   a. # mount -o loop sps.img /mnt

b. # cd /mnt

c. # ./setup

2. During this procedure, you will hit Enter in most cases to accept default values and continue to the next screen. Note the following exceptions:

   a. On the screen titled “High AvailabilityNFS” you may select “n” as in this particular eval guide we will not be creating a highly available NFS server cluster configuration.

   b. If you have plans to create a highly available NFS service, adjust your response accordingly.

3. Towards the end of the setup script, you can choose to install a trial license key now, or later. We will install the license key in the next step, so you can safely select “n” at this point.

4. In the final screen of the “setup” select the DataKeeper from the list displayed on the screen.

5. The following RPMs should be be installed:

   a. steeleye-lkSQL-<version>.noarch.rpm

   b. steeleye-lkDR-<version>.noarch.rpm

6. Un-mount the Distribution Enabling disk image:

   a. # cd /root

   b. # umount /mnt

# Install the Evaluation License Keys

The last phase of the setup process installs the licensing keys. You must install the evaluation license key file (“.lic”) that you downloaded with your evaluation software before starting the LifeKeeper for Linux.

1. To install your trial license key, run the “lkkeyins” command on both Primary and Secondary Server. This command is located at /opt/LifeKeeper/bin/lkkeyins . Example:

   # /opt/LifeKeeper/bin/lkkeyins <path_to_license/<filename>.lic

2. Validate your license keys were installed via the /opt/LifeKeeper/bin/lklicmgr command

   # /opt/LifeKeeper/bin/lklicmgr
   License File: 20101230.lic

| Product | Type | Expiry |
|---|---|---|
| LifeKeeper for Linux | Eval | 27 Mar 2013 (87 days) |

| | | |
|---|---|---|
| SIOS Data Replication ARK | Eval | 27 Mar 2013 (87 days) |
| MySQL Recovery Kit | Eval | 27 Mar 2013 (87 days) |

# Start the LifeKeeper for Linux

1. Start

```
# /opt/LifeKeeper/bin/lkstart
```

# 11.2.7.5.4.5. Configure the Cluster

✻ Before you begin to create the MySQL Resource Hierarchy, refer to Install LifeKeeper for Linux – MySQL Cluster and Configuration Considerations for MySQL and make sure you have completed the necessary configuration tasks.

## Create the MySQL Resource Hierarchy

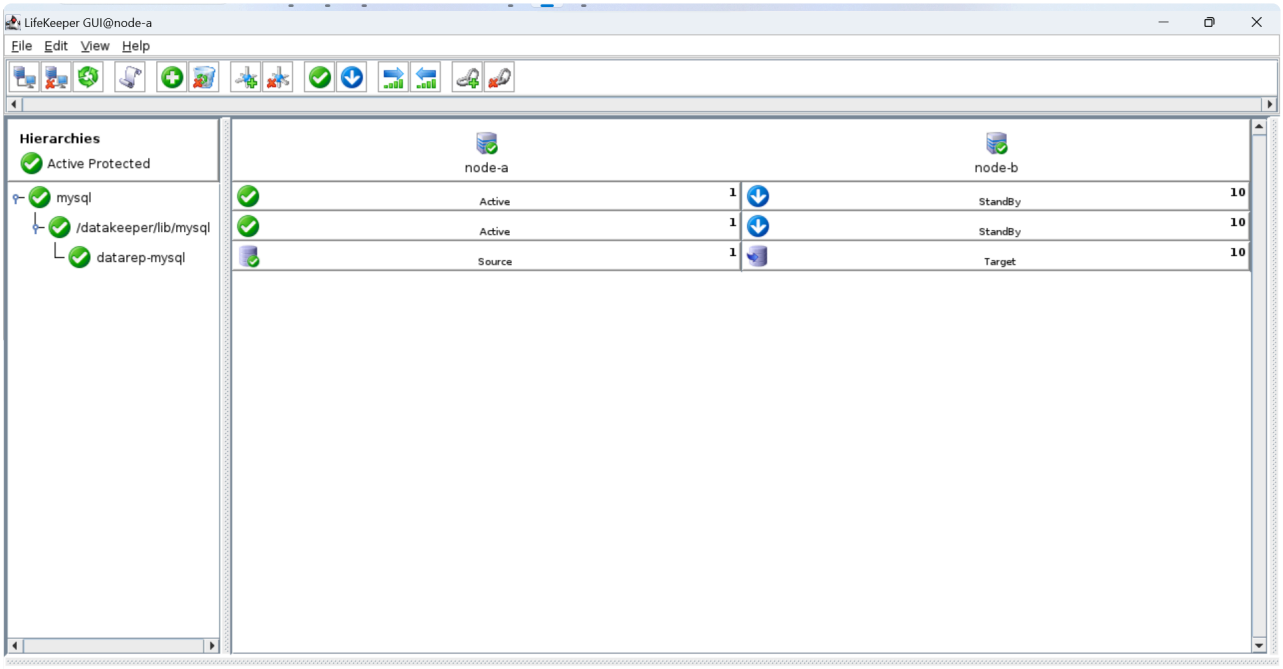
Create a MySQL resource to protect the MySQL database and make it high available between cluster nodes.

✻ **Important** At this point, MySQL needs to be already running on the Primary Server. It should NOT be running on the Secondary Server. See section titled “Install, Configure, and Start MySQL” above to review the process to configure and start MySQL as needed.

1. From the LifeKeeper GUI toolbar, click Create Resource Hierarchy.
2. Select **MySQL Database** and click **Next**.
3. Proceed Through the Resource Creation wizard, providing the following values

| Field | Recommended Entries or Notes |
|---|---|
| Switchback Type | Intelligent |
| Server | LinuxPrimary (Primary Cluster Node) |
| Location of my.cnf | Enter “/datakeeper/lib/mysql”. Note that earlier in the MySQL configuration process we created a my.cnf file in this directory |
| Location of MySQL | Leave as default (/usr/bin) since we are using a standard MySQL install/configuration in this example |
| Database tag | Leave as default |

4. Select Create to define the MySQL resource hierarchy on the Primary Server
   Note: LifeKeeper will automatically identify that the MySQL resource has a dependency on the FileSystem (Data Replication) resource (/datakeeper/lib/mysql). The Filesystem Resource will appear underneath the MySQL resource in the GUI
5. Click Next to Extend the File System Resource to the Secondary Server
6. In the Extend Wizard, select “Accept Defaults”
7. As a result the MySQL resource is now protected on both cluster nodes. Click Finish to exit the Extend wizard.
8. Your resource hierarchy should look as follows:


LifeKeeper GUI@node-a
File Edit View Help
Hierarchies
Active Protected
mysql
/datakeeper/lib/mysql
datarep-mysql
node-a
node-b
Active 1
StandBy 10
Active 1
StandBy 10
Source 1
Target 10

# 11.2.7.5.5. SAPリソースの保護

このセクションでは、SAP S/4HANA 1909 AS ABAP環境をデプロイし、LifeKeeper SAP Recovery Kitを使用してABAP SAP Central Services (ASCS) と対応するEnqueue Replication Server (ERS) インスタンスを保護します。LifeKeeperを使用して冗長化されたプライマリーアプリケーションサーバー（PAS）インスタンスを保護することは可能ですが、ここでは代わりに2つのアベイラビリティゾーンにまたがる1つのPASインスタンスと1つの追加アプリケーションサーバー（AAS）インスタンスからなる外部アプリケーションサーバープールを作成する方法を取ります。

PASとAASインスタンスはLifeKeeperによって保護されないため、このガイドでは高可用性Central Services（ASCS/ERS）クラスターのインストールと設定のみを詳細に説明します。またこのセクションでは、高可用性データベースクラスター（例：SAP HANAを使用する場合など）の構成については説明していません。インストールされているデータベースに関連するガイドのセクションを参照して、計画中のSAP 環境に適合するように調整してください。

作成されるセントラルサービスクラスターノードの詳細を以下に示します。このガイドで説明するインスタンスのプロビジョニングは、評価でアプリケーションを実行するために必要な最低限の仕様を示しており、一般的に生産的なワークロードを処理するには十分ではありません。本番環境でVMをプロビジョニングする際のベストプラクティスについては、SAPおよびクラウドプロバイダーが提供するドキュメント（Hardware Requirements for SAP System Hostsなど

デプロイメントで使用するオペレーティングシステムと構成は、SAP、SIOS、およびクラウドプロバイダーによってサポートされている必要があります。SAP NetWeaverおよびS/4HANAのさまざまなバージョンでサポートされているオペレーティングシステムを確認するには、SAPの Product Availability Matrix (PAM) を参照してください。.

## 評価デプロイメントのためのクラスターアーキテクチャ

- node-a (プライマリーASCSインスタンスホスト、スタンバイERSインスタンスホスト、プライマリーNFSサーバー)
  - プライベートIP: 10.20.1.10
  - vCPU's: 2
  - メモリー: 4GB
  - SAP NFS共有マウント済み
  - SAPインスタンスASCS10とERS20がSID「SPS」の下にインストールされている
  - LifeKeeper for LinuxのQuorum/Witness機能を有効にし、以下のRecovery Kitがインストールされている
    - SAP Recovery Kit
    - SIOS DataKeeper
    - NFS Recovery Kit
    - [AWS] EC2 Recovery Kit
    - [Azure, Google Cloud] Generic Application Recovery Kit for Load Balancer Health Checks (GenLB Recovery Kit)

- node-b （プライマリーERSインスタンスホスト、スタンバイASCSインスタンスホスト、スタンバイNFSサーバー)
  - プライベートIP: 10.20.2.10

    - vCPU’s: 2
    - メモリー: 4GB
    - NFS共有マウント済み
    - SAPインスタンスASCS10とERS20がSID「SPS」の下にインストールされている
    - LifeKeeper for LinuxのQuorum/Witness機能を有効にし、以下のRecovery Kitがインストールされている
        - SAP Recovery Kit
        - SIOS DataKeeper
        - NFS Recovery Kit
        - [AWS] EC2 Recovery Kit
        - [Azure, Google Cloud] Generic Application Recovery Kit for Load Balancer Health Checks (GenLB Recovery Kit)

- node-c (Quorum/Witness node)
    - Private IP: 10.20.3.10
    - vCPU’s: 1
    - Memory: 2GB
    - LifeKeeper for LinuxのQuorum/Witness機能を有効にし、以下のRecovery Kitがインストールされている

- node-d (プライマリーアプリケーションサーバーホスト)
    - プライベートIP: 10.20.1.5
    - vCPU’s: 2
    - メモリー: 4GB
    - SAP NFS共有マウント済み
    - SAP PASインスタンスD01がSID「SPS」の下にインストールされている

- node-e (追加のアプリケーションサーバーホスト)
    - プライベートIP: 10.20.2.5
    - vCPU’s: 2
    - メモリー: 4GB
    - SAP NFS共有マウント済み
    - SAP AASインスタンスD02がSID「SPS」の下にインストールされている

- ASCS10仮想ホスト名：sps-ascs

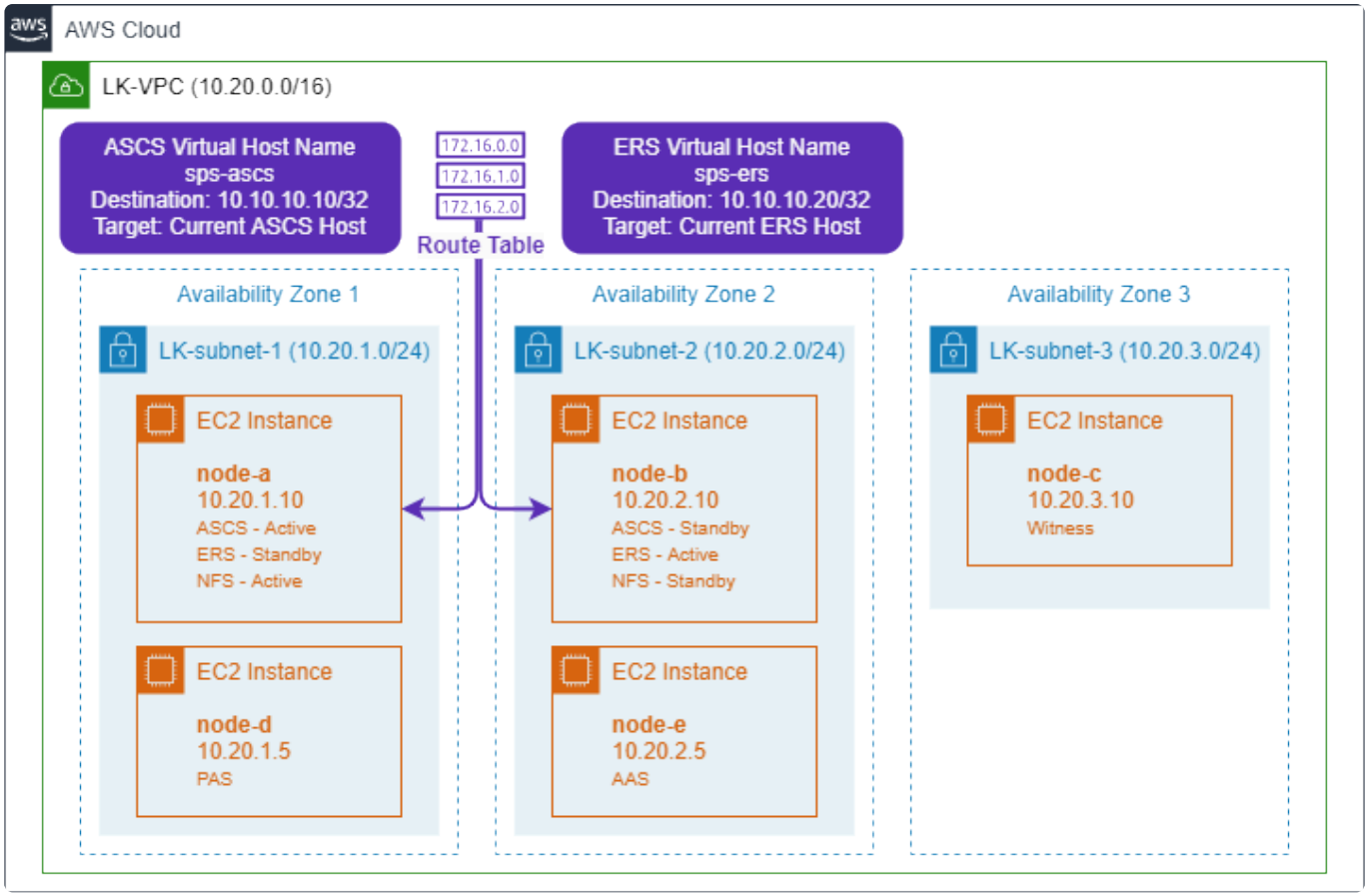
- ERS20仮想ホスト名：sps-ers

- NFS共有ファイルシステム:
    - /sapmnt/SPS
    - /usr/sap/trans

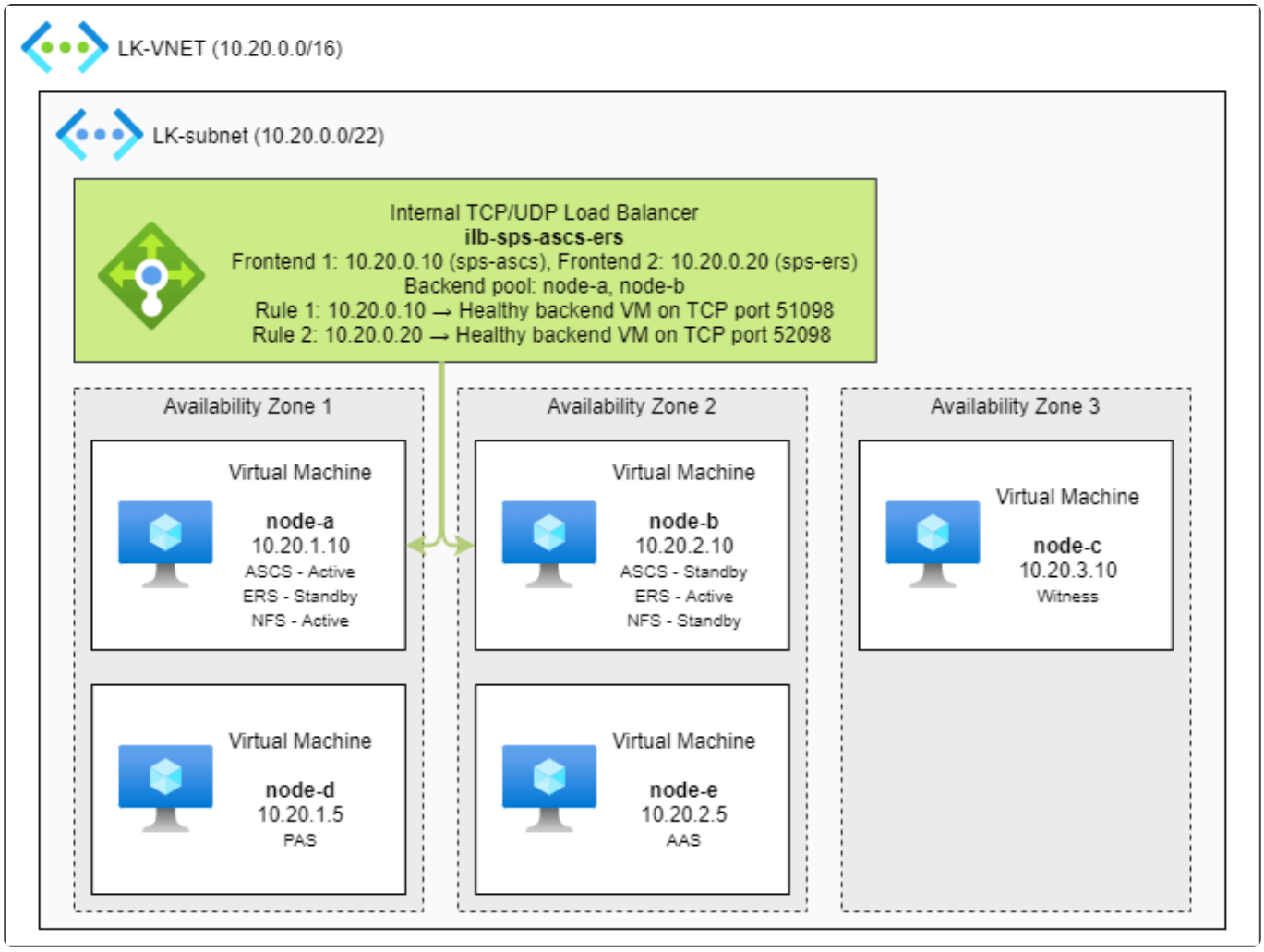
- SIOS DataKeeperレプリケーションファイルシステム:
    - /usr/sap/SPS/ASCS10
    - /usr/sap/SPS/ERS20

# クラウド固有のアーキテクチャ図

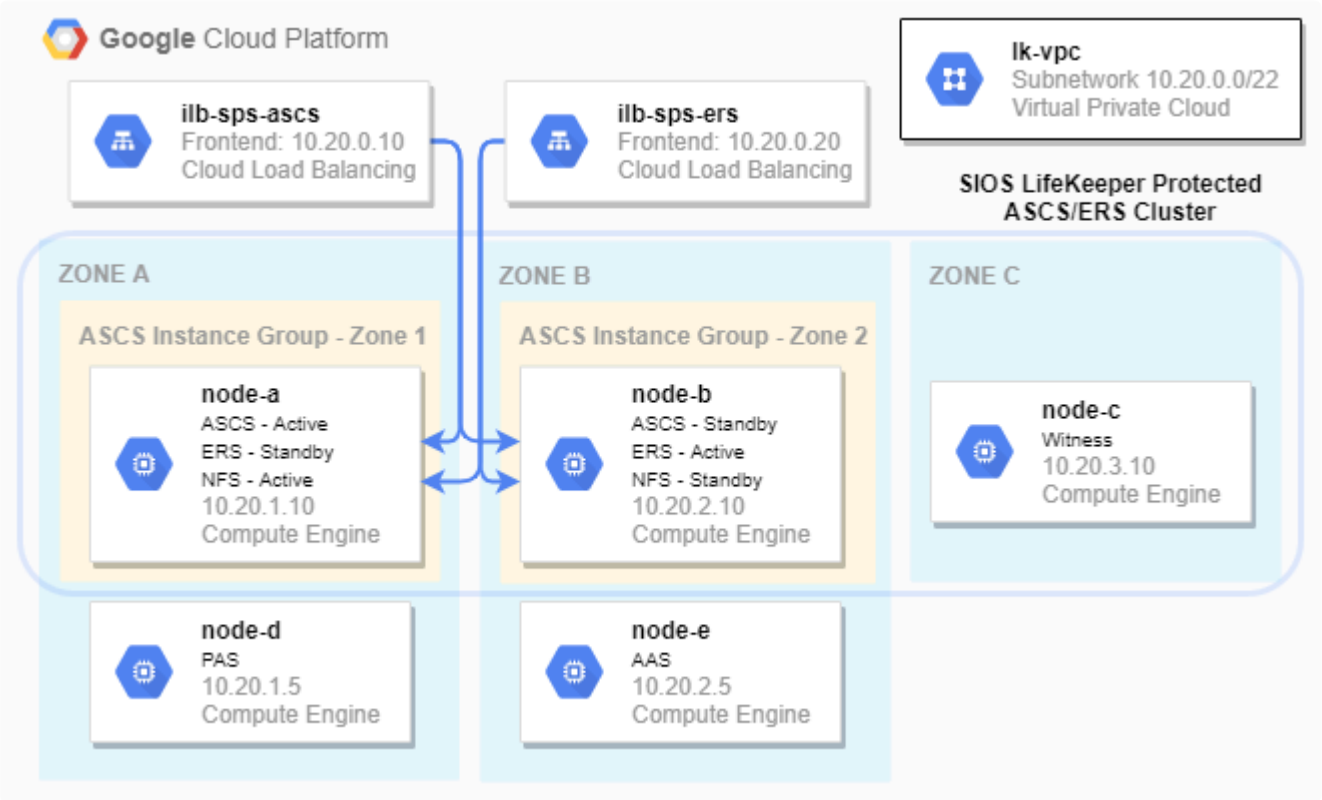
## アマゾンウェブサービス（AWS）上のSIOSで保護されたASCS/ERSクラスター



## Microsoft Azure上のSIOSで保護されたASCS/ERSクラスター



## Google Cloud上のSIOSで保護されたASCS/ERSクラスター




# 前提条件

このガイドに従うには、次の前提条件がすでに完了している必要があります。

1. 3つのVMインスタンス（node-a, node-b, node-c）が、このガイドで前述したネットワーク規約を使用して作成されている。ファイアーウォールルールは、ノード間通信とSSH接続を許可するように配置されている。詳細については、ネットワークコンポーネントの設定とインスタンスの作成 を参照してください。

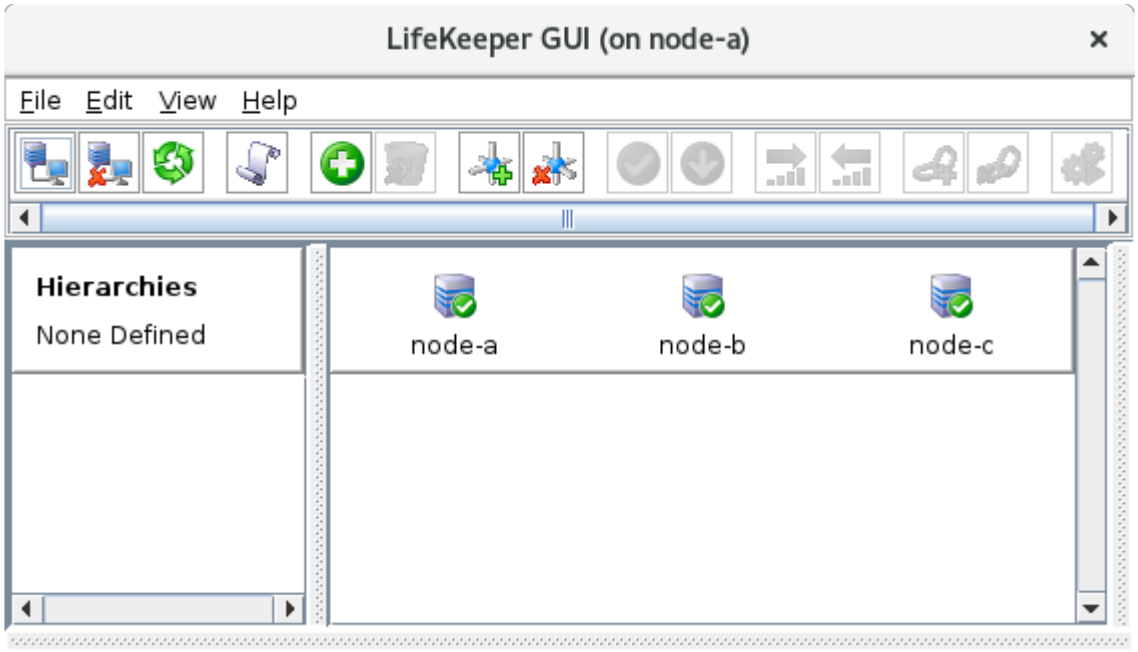
2. 3つのVMインスタンスはすべて、LifeKeeper for Linuxを実行するように構成されている。特に、SELinuxは無効、ファイアーウォールは無効、各ノードの/etc/hostsファイルには各ノードのホスト名をそのプライベートIPに解決するためのエントリーが含まれ、rootユーザーは各ノードで同じパスワードを使用している。詳細については LifeKeeper for Linuxを実行するためのLinuxノードの設定 を参照してください。

3. LifeKeeper for Linuxは、Quorum/Witness機能を有効にして3ノードすべてにインストールされており、node-aとnode-bには必要なすべての追加のRecovery Kitがインストールされている。node-c（Witnessノード）には、LifeKeeperのコアインストール以外に追加のRecovery Kitは必要ありません。各ノードに必要なすべてのSIOSライセンスがインストールされている。詳細については、LifeKeeper for Linux のインストールを参照してください。

> **注意:** 必要なRecovery Kitはnode-a とnode-b にのみインストールされているので、このガイドの中で LifeKeeper GUI から実行するすべての手順はnode-a またはnode-b のいずれかで実行する必要があります。

4. LifeKeeperのコミュニケーションパスは、クラスターノードの全てのペア間で定義されている。3つの双方向コミュニケーションパス（node-a ↔ node-b,、node-a ↔ node-c、 node-b ↔ node-c）を作成する必要があることに注意してください。詳細については、ログインと基本設定タスクを参照してください。

これらの作業をすべて完了すると、LifeKeeperのGUIは以下の画像のようになります。



# 11.2.7.5.5.1. ASCSおよびERSの仮想IPを作成する

SAP AS ABAPリリース 7.53 (ABAP Platform 1809) とEnqueue Standalone Framework v2 (ENSAv2) の導入により、ASCSとERSインスタンスは関連する仮想ホスト名を通じて相互に通信します。仮想ホスト名sps-ascsとsps-ersは、それぞれASCS10とERS20インスタンスの現在のホストに解決するように設定されます。

これらの仮想IPアドレスの推奨される実装は、クラウドプラットフォームによって異なります。お使いのクラウドプラットフォームに対応するセクションに記載されている手順に従ってください。

- AWS – ASCSおよびERSの仮想IPを作成する
- Azure – ASCSおよびERSの内部ロードバランサーを作成する
- Google Cloud – ASCSおよびERSの内部ロードバランサーを作成する

# 11.2.7.5.5.1.1. AWS – ASCSおよびERSの仮想IPを作成する

ASCSおよびERSインスタンスに関連付けられた仮想IPアドレスを保護するためのリソースを作成する前に、node-aおよびnode-bで次の手順を完了する必要があります。

- AWS CLIをインストールする
- EC2 Recovery Kitを使用するためのアクセス許可を割り当てる
- PINGブロードキャストを無効にする
- AWS – 送信元/送信先チェックを無効にする

次に、AWS EC2リソースの作成（ルートテーブルシナリオ）のプロセスに従って、ASCSおよびERSインスタンスに関連付けられた仮想IPアドレスのリソース階層を作成します。

EC2ルートテーブルのリソースタイプで使用されるIPアドレスはVPC CIDR（LK-VPCの場合は10.20.0.0/16）の外部に配置する必要があるため、次の仮想ホスト名とIPアドレスを使用します。

| インスタンス | 仮想ホスト名 | IP アドレス |
|---|---|---|
| ASCS10 | sps-ascs | 10.10.10.10 |
| ERS20 | sps-ers | 10.10.10.20 |

## ASCSの仮想IPリソース階層を作成する

1. IPリソースの作成 で説明した手順に従い、以下のパラメータを使用してLifeKeeper IPリソース（*ip-sps-ascs*）を作成および拡張し、node-aおよびnode-b上のASCS仮想IPアドレスを保護します。IPリソースはnode-aで作成され、node-bに拡張されることに注意してください。また、作成されたリソースはWitnessノードであるnode-cに拡張されてはならないことにも注意してください。
   ✓ アイコンは、デフォルトのオプションが選択されていることを示しています。

| フィールド | 値 |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent ✓ |
| Server | node-a |
| IP Resource | 10.10.10.10 |
| Netmask | 255.255.255.0 ✓ |
| Network Interface | eth0 ✓ |
| IP Resource Tag | ip-sps-ascs |
| **Pre-Extend Wizard** | |

| | |
|---|---|
| Target Server | node-b |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend comm/ip Resource Hierarchy Wizard** | |
| IP Resource | 10.10.10.10 |
| Netmask | 255.255.255.0 |
| Network Interface | eth0 |
| IP Resource Tag | ip-sps-ascs |

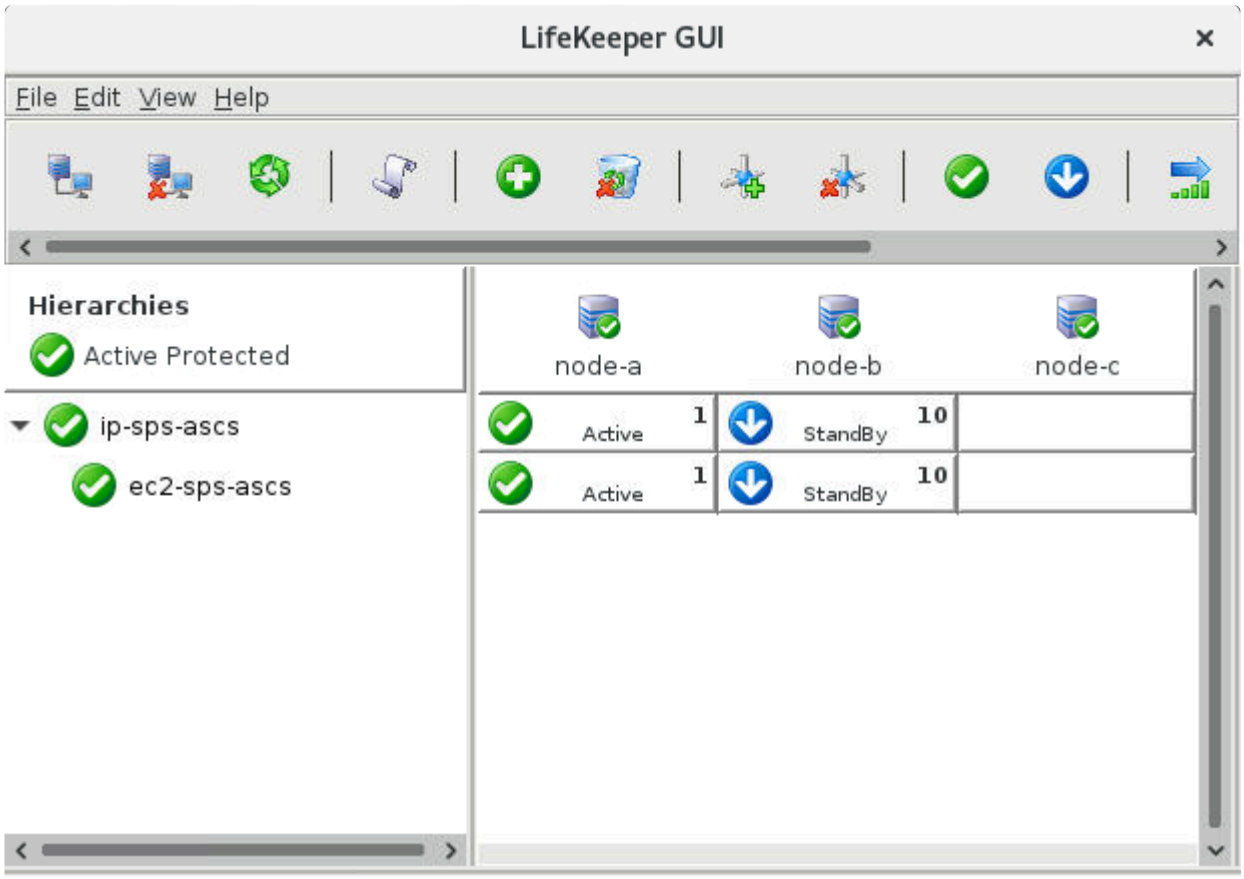
2. AWS EC2リソースの作成（ルートテーブルシナリオ） で説明している手順に従って、以下のパラメーターを使用してLifeKeeper EC2リソース（ **ec2-sps-ascs** ）を作成および拡張し、IPリソース切り替え時にAWSでルートテーブルのバックエンド操作を管理します。EC2リソースは、node-aで作成され、node-bに拡張されていることに注意してください。また、作成されたリソースはWitnessノードであるnode-cに拡張されてはならないことにも注意してください。 アイコンは、デフォルトのオプションが選択されていることを示しています。

| フィールド | 値 |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent |
| Server | node-a |
| EC2 Resource type | RouteTable (Backend cluster) |
| IP Resource | ip-sps-ascs |
| EC2 Resource Tag | ec2-sps-ascs |
| **Pre-Extend Wizard** | |
| Target Server | node-b |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend comm/ec2 Resource Hierarchy Wizard** | |

| | |
|---|---|
| EC2 Resource Tag | ec2-sps-ascs |

> 注意： EC2リソースの作成または拡張に失敗した場合は、（i）トラフィックを10.10.10.10/32からNode-Aに転送するルートがAWSのLK-VPCのルートテーブルに存在すること、（ii）Node-AとNode-Bで送信元/送信先のチェックが無効になっていること、（iii）依存IPリソースip-sps-ascsの優先度の値がそれぞれNode-Aで1、Node-Bで10であることを確認してください。

IPリソース階層が正常に作成されると、LifeKeeper GUIは次の画像のようになります。




## ERS仮想IPリソース階層を作成する

1. IPリソースの作成 で説明している手順に従って、以下のパラメーターを使用してLifeKeeper IPリソース（ip-sps-ers）を作成および拡張し、node-aおよびnode-bのERS仮想IPアドレスを保護します。IPリソースは、node-bで作成され、node-aに拡張されていることに注意してください。また、作成されたリソースはWitnessノードであるnode-cに拡張されてはならないことにも注意してください。 アイコンは、デフォルトのオプションが選択されていることを示しています。

| フィールド | 値 |
|---|---|
| **Create Resource Wizard** | |

| | |
|---|---|
| Switchback Type | intelligent |
| Server | node-b |
| IP Resource | 10.10.10.20 |
| Netmask | 255.255.255.0 |
| Network Interface | eth0 |
| IP Resource Tag | ip-sps-ers |
| **Pre-Extend Wizard** | |
| Target Server | node-a |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend comm/ip Resource Hierarchy Wizard** | |
| IP Resource | 10.10.10.20 |
| Netmask | 255.255.255.0 |
| Network Interface | eth0 |
| IP Resource Tag | ip-sps-ers |

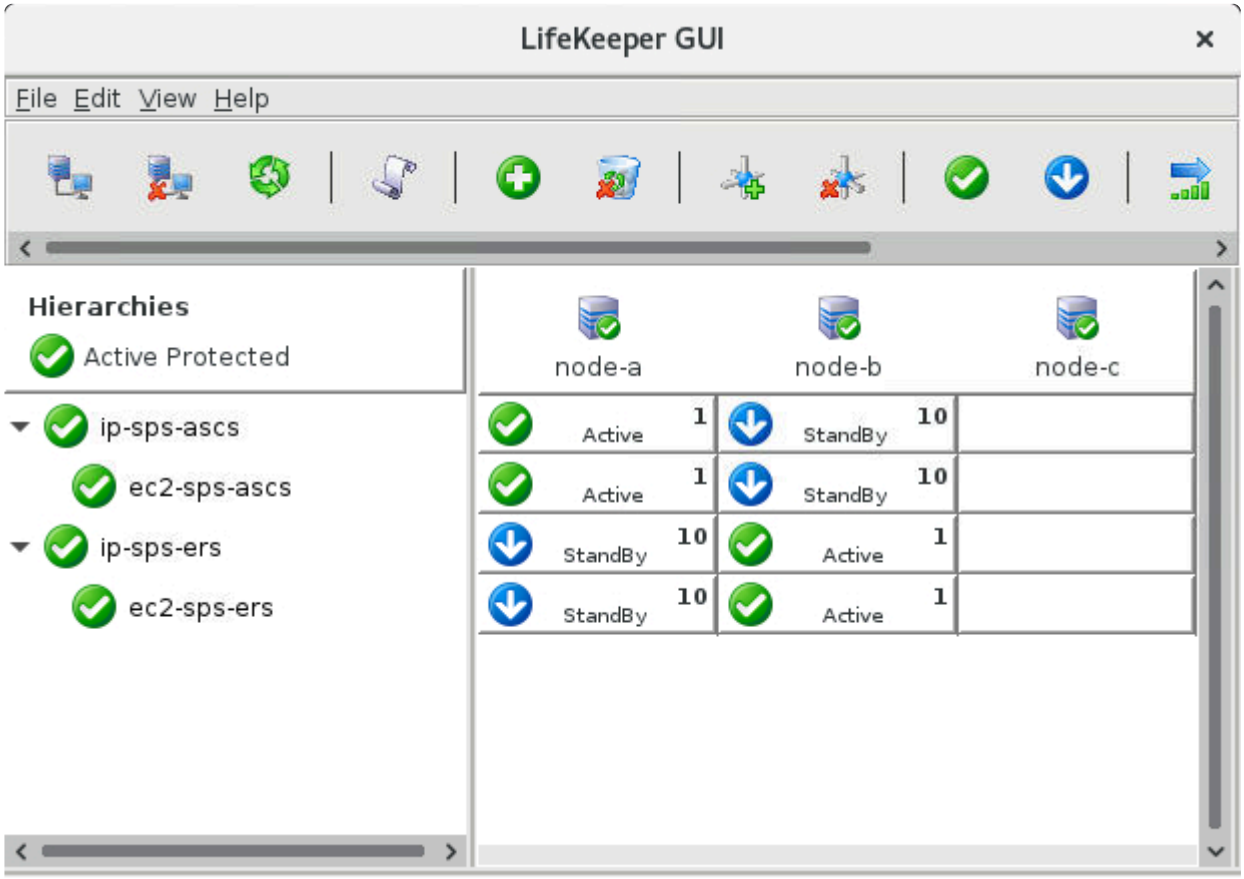
2. AWS EC2リソースの作成（ルートテーブルシナリオ） EC2リソース（ **ec2-sps-ers** ）を作成および拡張し、IPリソース切り替え時にAWSでルートテーブルのバックエンド操作を管理します。EC2リソースは、node-bで作成され、node-aに拡張されていることに注意してください。また、作成されたリソースはWitnessノードであるnode-cに拡張されてはならないことにも注意してください。
アイコンは、デフォルトのオプションが選択されていることを示しています。

| フィールド | 値 |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent |
| Server | node-b |
| EC2 Resource type | RouteTable (Backend cluster) |
| IP Resource | ip-sps-ers |

| | |
|---|---|
| EC2 Resource Tag | ec2-sps-ers |
| **Pre-Extend Wizard** | |
| Target Server | node-a |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend comm/ec2 Resource Hierarchy Wizard** | |
| EC2 Resource Tag | ec2-sps-ers |

*注意*：EC2リソースの作成または拡張に失敗した場合は、（i）トラフィックを10.10.10.10/32からNode-Bに転送するルートがAWSのLK-VPCのルートテーブルに存在すること、（ii）Node-AとNode-Bで送信元/送信先のチェックが無効になっていること、（iii）依存IPリソースip-sps-ersの優先度の値がそれぞれNode-Aで10、Node-Bで1であることを確認してください。

IPリソース階層が正常に作成されると、LifeKeeper GUIは次の画像のようになります。



# /etc/hostsにエントリーを追加する

Node-aおよびNode-bの/etc/hostsに以下のエントリーを追加し、仮想ホスト名を対応するIPアドレスに解決できるようにします。この仮想ホスト名は共有ファイルシステム /export/sapmnt/SPS および /export/usr/sap/trans にアクセスするために使用されるので、sps-ascsのエントリーは node-dおよびnode-e（PAS および AAS ホスト）にも追加する必要があります。

```
10.10.10.10    sps-ascs
10.10.10.20    sps-ers
```

# 11.2.7.5.5.1.2. Azure – ASCSおよびERSの内部ロードバランサーを作成する

Microsoft Azureでは、TCP内部ロードバランサーを使用して、ASCSおよびERSの仮想IPのフェイルオーバーを容易にします。ロードバランサーは2つのフロントエンドで構成され、それぞれが動作するサブネットからIPアドレスを割り当てられます。この例では、各仮想ホストに対応するロードバランサーフロントエンドに、次のIPアドレスを使用します。

| Instance | Virtual Host Name | Internal TCP Load Balancer | Frontend IP Address |
|---|---|---|---|
| ASCS10 | sps-ascs | ilb-sps-ascs-ers | 10.20.0.10 |
| ERS20 | sps-ers | ilb-sps-ascs-ers | 10.20.0.20 |

Azure – 内部ロードバランサーの使用 と ロードバランサーヘルスチェックへの応答 に記載されている手順に従って、以下のAzureおよびLifeKeeperのリソースを作成し、設定します。

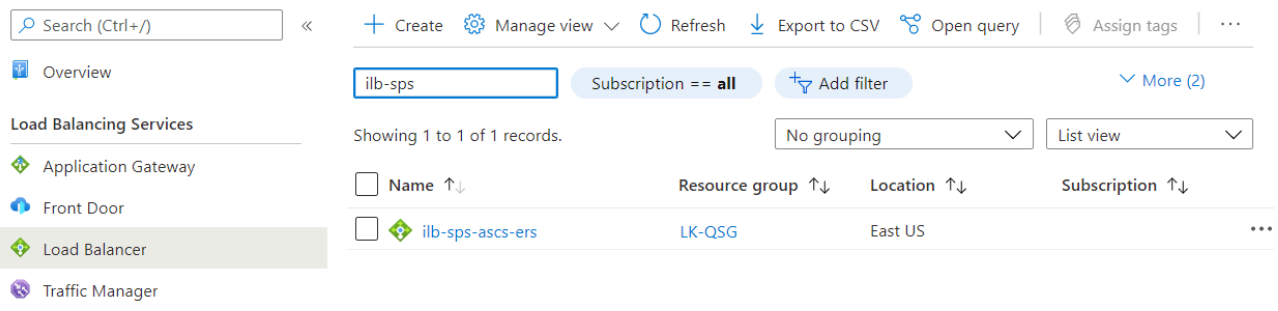
1. 次のプロパティを使用して、Azureで単一のロードバランサー **ilb-sps-ascs-ers** を作成します。

| Name | ilb-sps-ascs-ers |
|---|---|
| **Create Load Balancer** | |
| Resource Group | LK-QSG |
| Name | ilb-sps-ascs-ers |
| Region | （仮想マシンと同じ） |
| Type | Internal |
| SKU | Standard（ワークロードが可用性ゾーンに分散されるため、Standardを選択します） |
| Tier | Regional |
| **Frontend IP 1 Configuration** | |
| Name | SPSASCSFrontEnd |
| Virtual Network | LK-VNET |
| Subnet | LK-subnet (10.20.0.0/22) |
| Assignment | Static |
| IP address | 10.20.0.10 |
| Availability zone | Zone-redundant |
| **Frontend IP 2 Configuration** | |
| Name | SPSERSFrontEnd |

| | |
|---|---|
| Virtual Network | LK-VNET |
| Subnet | LK-subnet (10.20.0.0/22) |
| Assignment | Static |
| IP address | 10.20.0.20 |
| Availability zone | Zone-redundant |
| **Backend Pool** | |
| Name | backend-sps-ascs-ers |
| Backend Pool Configuration | NIC |
| IP Version | IPv4 |
| Virtual machines | node-a, node-b |
| **Health Probe #1** | |
| Name | probe-sps-ascs |
| Protocol | TCP |
| Port | 51098（後で作成する、対応するLifeKeeper LB Health Checkリソースと一致する必要があります） |
| Interval | 5 seconds |
| Unhealthy threshold | 2 consecutive failures |
| **Health Probe #2** | |
| Name | probe-sps-ers |
| Protocol | TCP |
| Port | 52098（後で作成する、対応するLifeKeeper LB Health Checkリソースと一致する必要があります） |
| Interval | 5 seconds |
| Unhealthy threshold | 2 consecutive failures |
| **Load Balancing Rule #1** | |
| Name | ilb-rule-sps-ascs |
| IP Version | IPv4 |
| Frontend IP address | SPSASCSFrontEnd (10.20.0.10) |
| HA Ports | Click (評価の展開を簡略化するため、すべてのポートへの転送を許可します) |
| Backend pool | backend-sps-ascs-ers |

| | |
|---|---|
| Health probe | probe-sps-ascs (TCP:51098) |
| Session persistence | None |
| Idle timeout | 4 minutes |
| TCP reset | Disabled |
| Floating IP | Enabled |
| **Load Balancing Rule #2** | |
| Name | ilb-rule-sps-ers |
| IP Version | IPv4 |
| Frontend IP address | SPSERSFrontEnd (10.20.0.20) |
| HA Ports | Click (評価の展開を簡略化するため、すべてのポートへの転送を許可します) |
| Backend pool | backend-sps-ascs-ers |
| Health probe | probe-sps-ers (TCP:52098) |
| Session persistence | None |
| Idle timeout | 4 minutes |
| TCP reset | Disabled |
| Floating IP | Enabled |

作成後、Microsoft Azureコンソールの **Load Balancing Services** → **Load Balancer** ページに、新しく作成されたロードバランサーが表示されます。



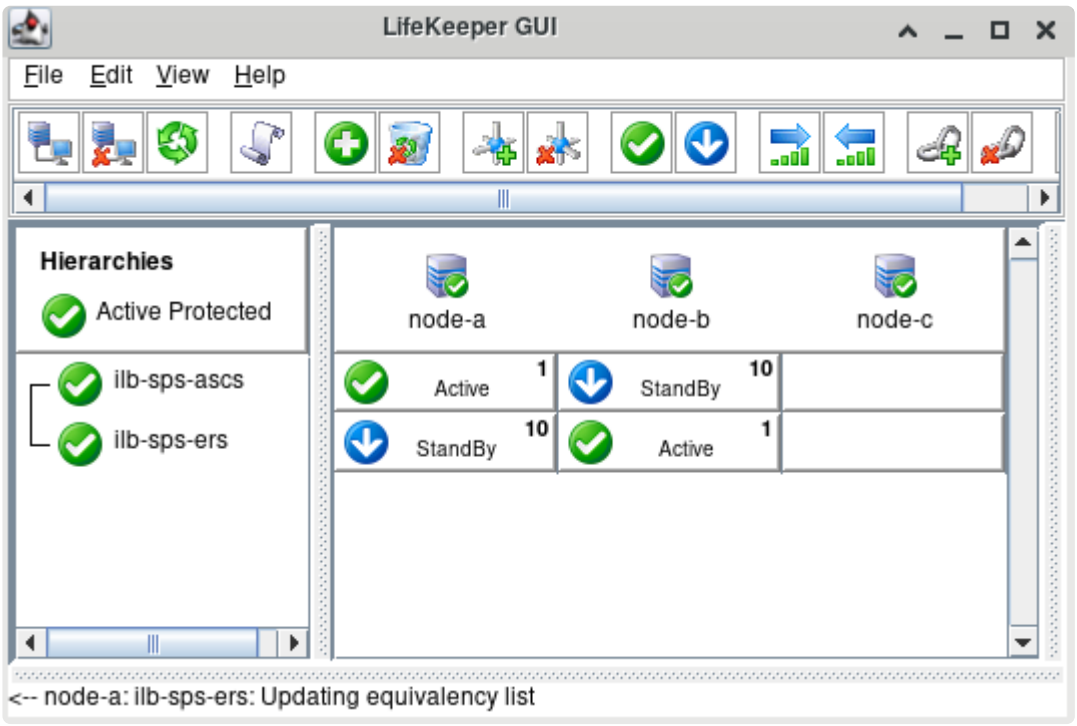

2. ロードバランサーヘルスチェックへの応答 の手順に従って、LB Health Checkをインストールし、以下のプロパティで2つのLifeKeeper LB Health Checkリソース、 **ilb-sps-ascs** と **ilb-sps-ers** を作成します。 **ilb-sps-ascs** はnode-aで作成され、node-bに拡張されますが、ilb-sps-ers はnode-bで作成され、node-aに拡張されることに注意してください。 アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |

| | |
|---|---|
| Switchback Type | intelligent |
| Server | node-a |
| Reply daemon Port | 51098 |
| Reply daemon message | None (Empty) |
| LB Health Check Resource Tag | ilb-sps-ascs |
| **Pre-Extend Wizard** | |
| Target Server | node-b |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend LB Health Check Resource Hierarchy Wizard** | |
| LB Health Check Resource Tag | ilb-sps-ascs |

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent |
| Server | node-b |
| Reply daemon Port | 52098 |
| Reply daemon message | None (Empty) |
| LB Health Check Resource Tag | ilb-sps-ers |
| **Pre-Extend Wizard** | |
| Target Server | node-a |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend LB Health Check Resource Hierarchy Wizard** | |
| LB Health Check Resource Tag | ilb-sps-ers |

リソースは、作成と拡張に成功すると、LifeKeeper GUI のリソースペインに表示されます。




3. IPリソースの作成 の手順に従って、次のパラメーターを使用してLifeKeeper IPリソース (**ip-sps-ascs**)を作成して拡張し、node-aおよびnode-bのASCS仮想IPアドレスを保護します。IPリソースはnode-aで作成され、node-bに拡張されていることに注意してください。また、作成されたリソースは、Witnessノードであるnode-cに拡張してはならないことに注意してください。 アイコンは、デフォルトのオプションが選択されていることを示しています。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent |
| Server | node-a |
| IP Resource | 10.20.0.10 |
| Netmask | 255.255.252.0 |
| Network Interface | eth0 |
| IP Resource Tag | ip-sps-ascs |
| **Pre-Extend Wizard** | |
| Target Server | node-b |
| Switchback Type | intelligent |
| Template Priority | 1 |

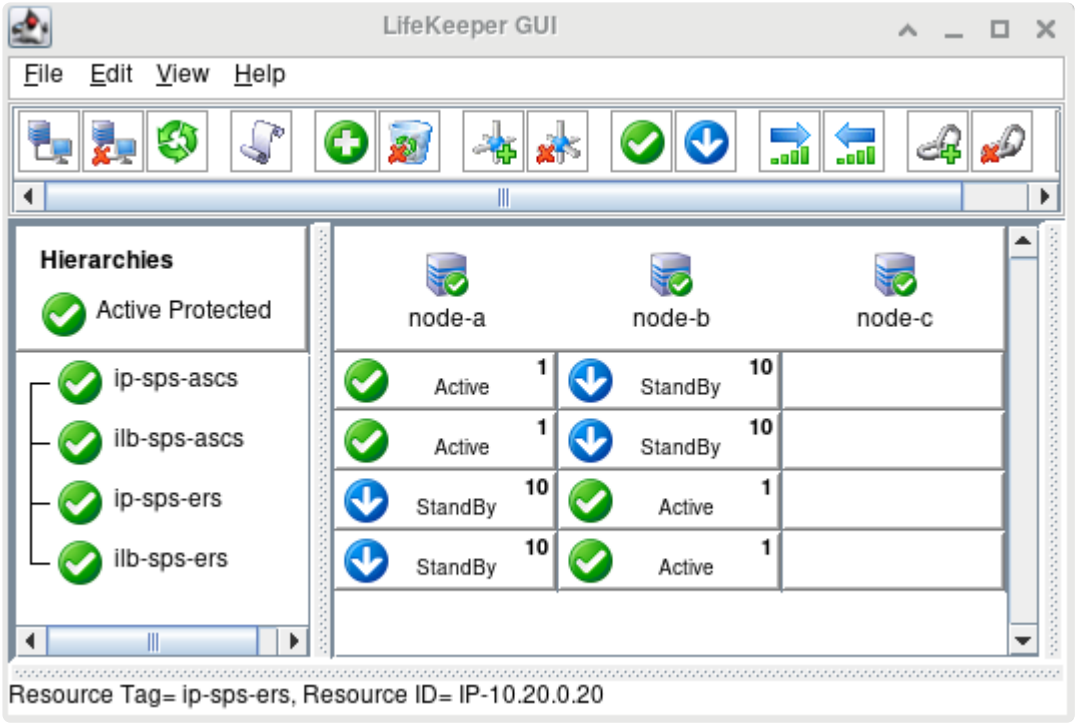
| | |
|---|---|
| Target Priority | 10 ✓ |
| **Extend comm/ip Resource Hierarchy Wizard** | |
| IP Resource | 10.20.0.10 ✓ |
| Netmask | 255.255.252.0 ✓ |
| Network Interface | eth0 ✓ |
| IP Resource Tag | ip-sps-ascs ✓ |

4. IPリソースの作成 の手順に従って、次のパラメーターを使用してLifeKeeper IPリソース（**ip-sps-ers**）を作成して拡張し、node-aおよびnode-bのERS仮想IPアドレスを保護します。IPリソースはnode-bで作成され、node-aに拡張されていることに注意してください。また、作成されたリソースは、Witnessノードであるnode-cに拡張してはならないことに注意してください。✓ アイコンは、デフォルトのオプションが選択されていることを示しています。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent ✓ |
| Server | node-b |
| IP Resource | 10.20.0.20 |
| Netmask | 255.255.252.0 ✓ |
| Network Interface | eth0 ✓ |
| IP Resource Tag | ip-sps-ers |
| **Pre-Extend Wizard** | |
| Target Server | node-a |
| Switchback Type | intelligent ✓ |
| Template Priority | 1 ✓ |
| Target Priority | 10 ✓ |
| **Extend comm/ip Resource Hierarchy Wizard** | |
| IP Resource | 10.20.0.20 ✓ |
| Netmask | 255.255.252.0 ✓ |

| | |
|---|---|
| Network Interface | eth0 |
| IP Resource Tag | ip-sps-ers |

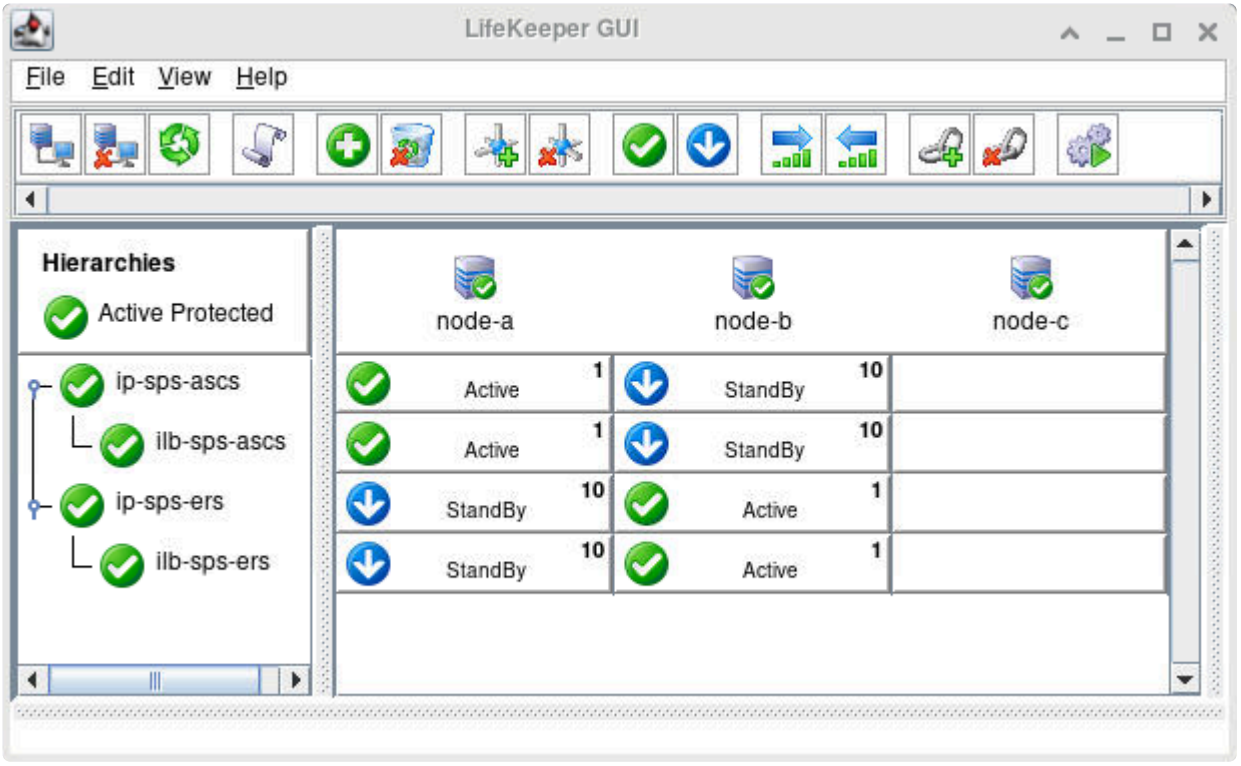
IPリソースが正常に作成されると、LifeKeeper GUIは次の画像のようになります。




5. node-aの **ip-sps-ascs** リソースを右クリックし、 **[Create Dependency...]** をクリックします。子リソースタグとして **ilb-sps-ascs** を指定し、 *[Create Dependency] * をクリックします。

6. node-bの **ip-sps-ers** リソースを右クリックし、[Create Dependency...] をクリックします。子リソースタグとして **ilb-sps-ers** を指定し、*[Create Dependency]* をクリックします。

   依存関係が正常に作成されると、LifeKeeper GUIは次の画像のようになります。




7. node-aおよびnode-bの/etc/hostsに以下のエントリーを追加し、各仮想ホスト名を対応するロードバランサーのフロントエンドIPアドレスに解決できるようにします。この仮想ホスト名は、共有ファイルシステムの /export/sapmnt/SPS および /export/usr/sap/trans へのアクセスに使用されるため、sps-ascsのエントリーをnode-dおよびnode-e (PAS および AAS ホスト) 上にも追加する必要があります。

```
10.20.0.10 sps-ascs
10.20.0.20 sps-ers
```

8. ロードバランサーのヘルスチェックに応答する の **LB Health Checkリソースの切り替えとフェイルオーバーのテスト** セクションで説明しているように、LB Health Checkリソース階層のスイッチオーバーとフェイルオーバーをテストします。検出された問題を修正するか、必要に応じてロードバランサーのヘルスチェックのパラメーターを調整して、正常に動作するようにします。

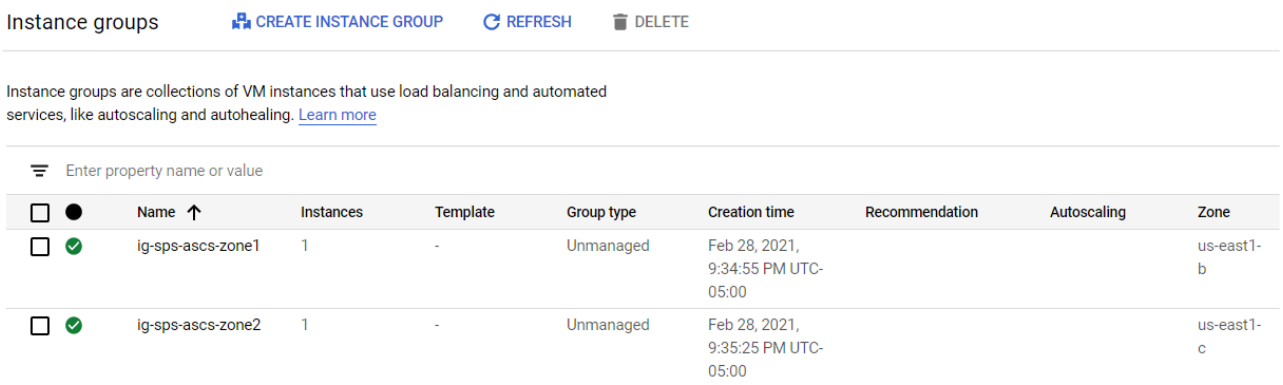
# 11.2.7.5.5.1.3. Google Cloud – ASCSおよびERSの内部ロードバランサーを作成する

Google Cloudでは、ASCSとERSの仮想IPのフェイルオーバーを容易にするために、TCP内部ロードバランサーが使用されています。各ロードバランサーのフロントエンドには、それが動作しているサブネットからエフェメラルIPが割り当てられます。この例では、各仮想ホストに対応するロードバランサーに以下のIPアドレスを使用します。

| Instance | Virtual Host Name | Internal TCP Load Balancer | Frontend IP Address |
|---|---|---|---|
| ASCS10 | sps-ascs | ilb-sps-ascs | 10.20.0.10 |
| ERS20 | sps-ers | ilb-sps-ers | 10.20.0.20 |

Google Cloud – 内部ロードバランサーを使用する および ロードバランサーヘルスチェックに応答する に記載されている手順に従って、次のGoogle CloudおよびLifeKeeperリソースを作成して設定します

1. node-aを含む **ig-sps-ascs-zone1** と、node-bを含む **ig-sps-ascs-zone2** という2つのアンマネージドインスタンスグループを作成します。




2. 次のプロパティを使用して、2つのTCP内部ロードバランサー **ilb-sps-ascs** と **ilb-sps-ers** を作成します。 アイコンは、デフォルトのオプションが選択されていることを示します。

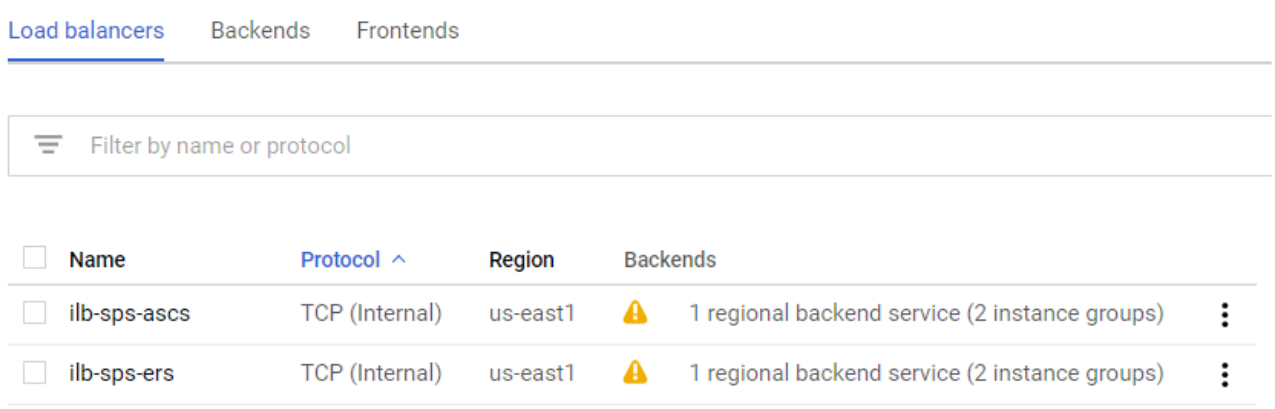
| Name | ilb-sps-ascs |
|---|---|
| **Backend configuration** | |
| Region | <Deployment region> (e.g., us-east1) |
| Network | lk-vpc |
| Backends | ig-sps-ascs-zone1<br>ig-sps-ascs-zone2 |
| **Health Check** | |
| Name | hc-sps-ascs |

| | |
|---|---|
| Protocol | TCP |
| Port | 51098 （後で作成する、対応するLB Health Checkリソースと一致する必要があります） |
| Proxy protocol | NONE |
| Request | 空欄のままにします |
| Response | 空欄のままにします |
| Check interval | 5 seconds |
| Timeout | 5 seconds |
| Healthy threshold | 2 consecutive successes |
| Unhealthy threshold | 2 consecutive failures |
| Connection draining timeout | 15 seconds |
| **Frontend configuration** | |
| Name | fe-sps-ascs |
| Subnetwork | lk-subnet |
| Purpose | Non-shared |
| IP address | Ephemeral (Custom) |
| Custom ephemeral IP address | 10.20.0.10 |
| Ports | All |
| Global access | Disable |
| Service label | Leave empty |

| **Name** | **ilb-sps-ers** |
|---|---|
| **Backend configuration** | |
| Region | <Deployment region> (e.g., us-west1) |
| Network | lk-vpc |
| Backends | ig-sps-ascs-zone1<br>ig-sps-ascs-zone2 |

| **Health Check** | |
|---|---|
| Name | hc-sps-ers |
| Protocol | TCP |
| Port | 52098 （後で作成する、対応するLB Health Checkリソースと一致する必要があります） |
| Proxy protocol | NONE |
| Request | 空欄のままにします |
| Response | 空欄のままにします |
| Check interval | 5 seconds |
| Timeout | 5 seconds |
| Healthy threshold | 2 consecutive successes |
| Unhealthy threshold | 2 consecutive failures |
| Connection draining timeout | 15 seconds |
| **Frontend configuration** | |
| Name | fe-sps-ers |
| Subnetwork | lk-subnet |
| Purpose | Non-shared |
| IP address | Ephemeral (Custom) |
| Custom ephemeral IP address | 10.20.0.20 |
| Ports | All |
| Global access | Disable |
| Service label | Leave empty |

作成すると、Google Cloud Console の Network services → Load balancing ページに、2つのロードバランサーが表示されます。




3. Google Cloud – 内部ロードバランサーを使用する の「**IP転送を無効にする**」セクションで説明している手順に従って、node-aとnode-bがロードバランサーのフロントエンドIPアドレスを介して通信できるようにします。変更を有効にするには、node-aとnode-bを再起動します。

4. IPリソースの作成 で説明している手順に従って、以下のパラメータを使用して、node-aとnode-b上のASCSとERS仮想IPアドレスを保護するために、ネットマスク255.255.255.255の2つのLifeKeeper IPリソース（**ip-sps-ascs** と **ip-sps-ers**）を作成し、拡張します。 **ip-sps-ascs** はnode-aで作成され、node-bに拡張されますが、 **ip-sps-ers** はnode-bで作成され、node-aに拡張されることに注意してください。また、作成されたリソースはWitnessノードであるnode-cに拡張してはいけないことに注意してください。 ✔ アイコンは、デフォルトのオプションが選択されていることを示します。

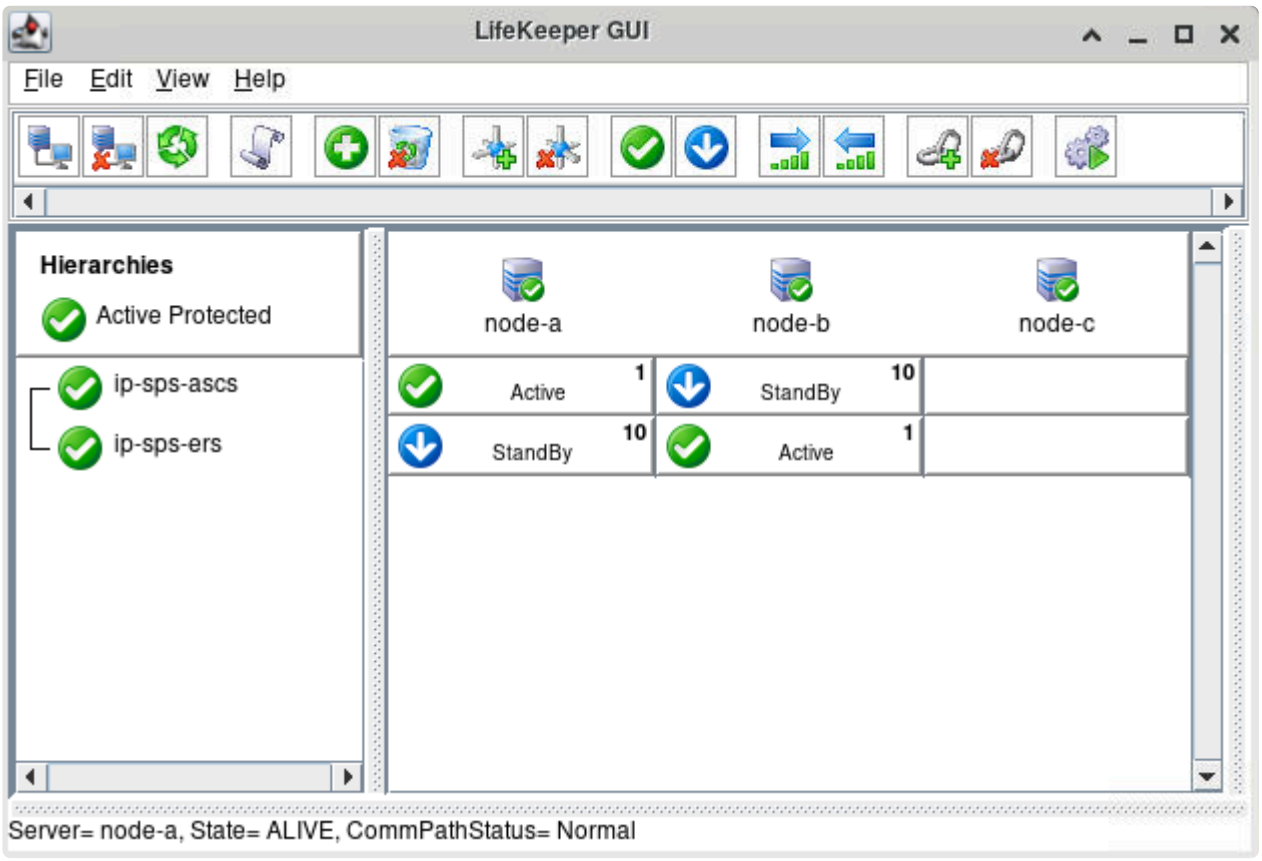
> ! ネットマスクパラメータの値として（作成時と拡張時の両方で）255.255.255.255を使用することが重要です。拡張プロセス中に「デフォルトを受け入れる（Accept Defaults）」を選択しないように注意してください。選択すると、255.255.255.0がスタンバイノードのネットワークマスクとして誤って使用され、予期せぬネットワークの問題が発生する可能性があります。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent ✔ |
| Server | node-a |
| IP Resource | 10.20.0.10 |
| Netmask | 255.255.255.255 |
| Network Interface | eth0 ✔ |
| IP Resource Tag | ip-sps-ascs |
| **Pre-Extend Wizard** | |
| Target Server | node-b |

| | |
|---|---|
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend comm/ip Resource Hierarchy Wizard** | |
| IP Resource | 10.20.0.10 |
| Netmask | 255.255.255.255 |
| Network Interface | eth0 |
| IP Resource Tag | ip-sps-ascs |

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent |
| Server | node-b |
| IP Resource | 10.20.0.20 |
| Netmask | 255.255.255.255 |
| Network Interface | eth0 |
| IP Resource Tag | ip-sps-ers |
| **Pre-Extend Wizard** | |
| Target Server | node-a |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend comm/ip Resource Hierarchy Wizard** | |
| IP Resource | 10.20.0.20 |
| Netmask | 255.255.255.255 |
| Network Interface | eth0 |
| IP Resource Tag | ip-sps-ers |

IPリソースが正常に作成されると、LifeKeeper GUIは次の画像のようになります。



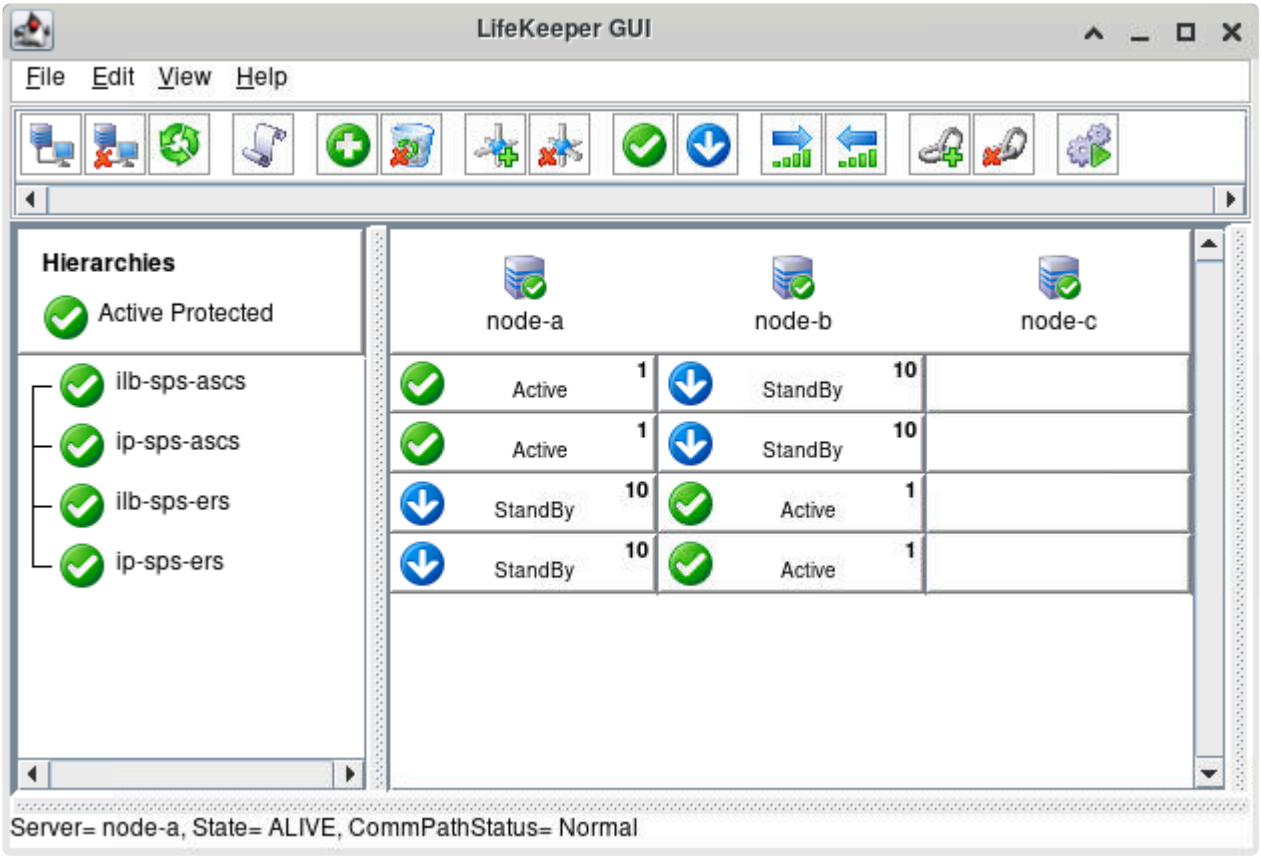

5. ロードバランサーのヘルスチェックに応答する の手順に従って、ロードバランサーヘルスチェック用のLB Health Checkをインストールし、以下のプロパティを使用して2つのLB Health Checkリソース **ilb-sps-ascs** と **ilb-sps-ers** を作成します。 **ilb-sps-ascs** はnode-aで作成され、node-bに拡張されますが、 **ilb-sps-ers** はnode-bで作成され、node-aに拡張されることに注意してください。
アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent |
| Server | node-a |
| Reply daemon Port | 51098 |
| Reply daemon message | None (Empty) |
| LB Health Check Resource Tag | ilb-sps-ascs |
| **Pre-Extend Wizard** | |
| Target Server | node-b |
| Switchback Type | intelligent |

| | |
|---|---|
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend LB Health Check Resource Hierarchy Wizard** | |
| LB Health Check Resource Tag | ilb-sps-ascs |

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent |
| Server | node-b |
| Reply daemon Port | 52098 |
| Reply daemon message | None (Emply) |
| LB Health Check Resource Tag | ilb-sps-ers |
| **Pre-Extend Wizard** | |
| Target Server | node-a |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend LB Health Check Resource Hierarchy Wizard** | |
| LB Health Check Resource Tag | ilb-sps-ers |

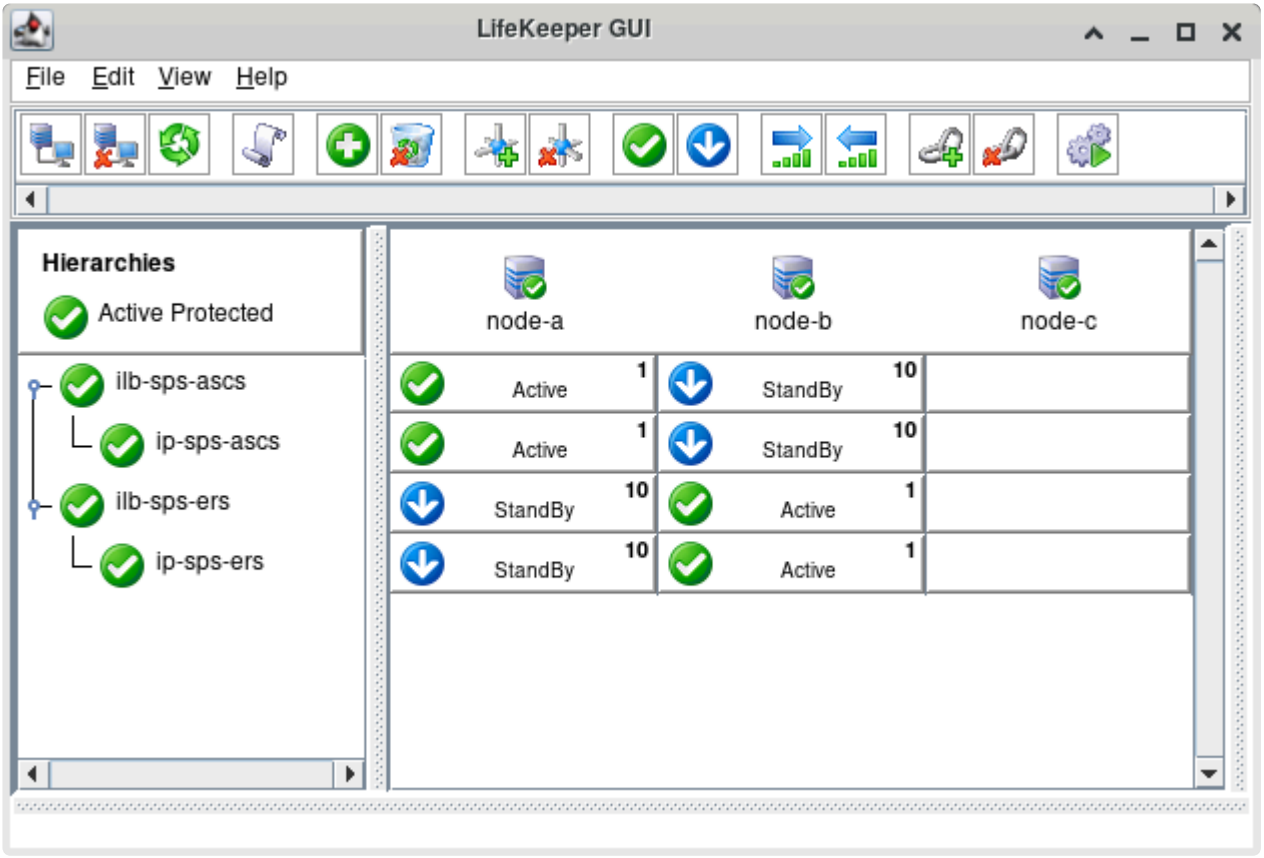
リソースが正常に作成･拡張されると、LifeKeeper GUIリソースペインにリソースが表示されます。




6. node-aの **ip-sps-ascs** リソースを右クリックし、 **[Create Dependency...]** をクリックします。子リソースタグとして **ilb-sps-ascs** を指定し、 **[Create Dependency]** をクリックします

7. node-bの **ip-sps-ers** リソースを右クリックし、 **[Create Dependency...]** をクリックします。子リソースタグとして **ilb-sps-ers** を指定し、 **[Create Dependency]** をクリックします。

   両方の依存関係が作成されると、LifeKeeper GUIは次の画像のようになります。




8. 次のエントリーをnode-aおよびnode-bの/etc/ hostsに追加して、各仮想ホスト名を対応するロードバランサーのフロントエンドIPアドレスに解決できるようにします。

```
10.20.0.10 sps-ascs
10.20.0.20 sps-ers
```

9. ロードバランサーのヘルスチェックに応答する の **LB Health Check**リソースのスイッチオーバーとフェイルオーバーのテスト セクションに記載されているように、GenLBリソースのスイッチオーバーとフェイルオーバーをテストします。見つかった問題を修正したり、ロードバランサーのヘルスチェックのパラメーターを必要に応じて調整したりして、正常に動作するようにします。

# 11.2.7.5.5.2. SAP ファイルシステムを作成する

**IMPORTANT**: 業界のベストプラクティスに沿って、SIOS は、SAP システムをセットアップする際に、SAP サービスと同じノードに NFS サーバー/クライアントをインストールすること（これは「ループバック構成」とも呼ばれます）を推奨しなくなりました。 代わりに、次のような構成を代替として使用することを推奨する方向に進みます。
• EFS in AWS
• Azure Premium File Shares with NFS
• Google NFS Share
• NFS as separate cluster
今後、「ループバック構成」での新規インストールはサポートいたしません。 既存のお客様には、SIOS ソフトウェア内の問題であれば、可能な範囲でサポートし、別の構成に移行することをお勧めします。 次回のリリースでは、既存のコンフィギュレーションはサポートされません。

このセクションでは、SAPインスタンスをホストする各ノードにマウントされる、高可用性 NFS 共有ファイルシステムを作成します。高可用性 SAP AS ABAP 環境においてシステム間で共有される一般的なファイルシステムには、以下のものがあります。

- /sapmnt/<SID>
- /usr/sap/trans
- /usr/sap/<SID>/ASCS<InstNum>
- /usr/sap/<SID>/ERS<InstNum>

詳細については、高可用性システムのファイルシステムの設定 を参照してください。

**Note:** インスタンス固有のファイルシステム（/usr/sap//ASCS、/usr/sap//ERS）は、対応するSAPインスタンスが稼働しているシステムにマウントすればよいので、NFS を介した共有ではなく SIOS DataKeeper によるレプリケーションが可能です。

**Note:** これらの共有ファイルシステムは、可用性の高い ASABAP デプロイメントをサポートします。その他のデータベース関連ファイルシステムは、選択したバックエンドデータベースプラットフォームによって異なります。

NFS 共有をエクスポートするための推奨の実装は、クラウドプラットフォームによって異なります。お使いのクラウドプラットフォームに対応するセクションに記載されている手順に従ってください。

- AWS/Azure – SAP 共有ファイルシステムおよびレプリケーションファイルシステムの作成
- Google Cloud – SAP 共有ファイルシステムおよびレプリケーションファイルシステムの作成

# 11.2.7.5.5.2.1. AWS/Azure – SAP共有ファイルシステムおよびレプリケーションファイルシステムの作成

> ! NFS 共有をエクスポートするサーバー（またはクラスター） が、ASCS + ERS を実行しているサーバーとは別のサーバーグループである場合、NFS 共有を使用した ASCS + ERS がサポートされます。

> ✻ 注意： このセクションは、AWSおよびMicrosoft Azureでのデプロイメントの場合の手順です。Google Cloudにデプロイする場合は、Google Cloud – SAP共有ファイルシステムの作成 に記載されている手順に従ってください。

このセクションでは、SAPインスタンスをホストする各ノード上にマウントされる高可用性NFS共有ファイルシステムを作成します。また、node-aとnode-bの間でASCS10とERS20インスタンスのデータを複製するために、SIOS DataKeeperミラーを作成します。高可用性SAP AS ABAP環境において、システム間で共有または複製される典型的なファイルシステムには、以下のものがあります。

- /sapmnt/<SID>
- /usr/sap/trans
- /usr/sap/<SID>/ASCS<InstNum>
- /usr/sap/<SID>/ERS<InstNum>

詳細については、 Setting up File Systems for a High-Availability System を参照してください。

> ✻ 注意： このガイドでは、すべてのリソースがクラスター内で自己完結するように、ファイル共有とレプリケーションのメカニズムを実装することを選択しましたが、他の構成も可能です。たとえば、NFS共有ファイルシステムは、可用性の高い外部NFSサーバークラスターでホストしたり、Amazon Elastic File System（EFS）などのクラウドネイティブNFSソリューションでホストすることもできます。

1. 可用性の高い SAP インストールをサポートするために、node-aとnode-bに4つの追加ディスクを作成しアタッチします。この例で使用されているデバイス名は、お使いの環境によって異なる場合があります（例：/dev/sdbの代わりに/dev/xvdb）ので、適切なデバイス名を使用するようにセクションに記載されているコマンドを適宜調整してください。これらのディスクサイズは、評価目的にのみ使用されていることに注意してください。本番環境でリソースをプロビジョニングする場合は、SAP、クラウドプロバイダー、およびサードパーティのストレージプロバイダーの関連ドキュメントを参照してください。

| Device Name | Minimum Size | Mount Point |
|---|---|---|
| /dev/sdb | 8GB | /export/sapmnt/SPS |

| | | |
|---|---|---|
| /dev/sdc | 8GB | /export/usr/sap/trans |
| /dev/sdd | 8GB | /usr/sap/SPS/ASCS10 |
| /dev/sde | 4GB | /usr/sap/SPS/ERS20 |

✻ 注意： SAP のインストールをサポートするためにストレージを接続する場合は、他の構成を使用できます。 たとえば、LVM を介して複数の論理パーティションで分割された単一の物理ボリュームを使用すると便利な場合があります。

2. node-a と node-b の両方で以下のコマンドを実行し、xfs ファイルシステムで/dev/sdb1パーティションを作成します。

```
[root@node-a ~]# parted /dev/sdb --script mklabel gpt mkpart xfspart xfs 0% 10
0%

[root@node-a ~]# mkfs.xfs /dev/sdb1
meta-data=/dev/sdb1    isize=512    agcount=4, agsize=1310592 blks
         =             sectsz=4096 attr=2, projid32bit=1
         =             crc=1       finobt=1, sparse=1, rmapbt=0
         =             reflink=1
data     =             bsize=4096  blocks=5242368, imaxpct=25
         =             sunit=0     swidth=0 blks
naming   =version 2    bsize=4096  ascii-ci=0, ftype=1
log      =internal log bsize=4096  blocks=2560, version=2
         =             sectsz=4096 sunit=1 blks, lazy-count=1
realtime =none         extsz=4096  blocks=0, rtextents=0

[root@node-a ~]# partprobe /dev/sdb1
```

node-a、node-bの両方で、/dev/sdc、/dev/sdd、/dev/sdeに対して、このコマンドを繰り返します。

3. Node-a と node-b の両方で次のコマンドを実行して、共有および複製された SAP ファイル システムのマウントポイントを作成します。

```
[root@node-a ~]# mkdir -p /export/{usr/sap/trans,sapmnt/SPS} /sapmnt/SPS /usr/
sap/{trans,SPS/{ASCS10,ERS20,SYS}}
```

## DataKeeper のデータレプリケーションリソース階層を作成する

1. ファイルシステムのデータレプリケーションの作成方法 で説明した手順に従って、以下のパラメーターを使用して DataKeeper データレプリケーションリソース（ **datarep-/export/sapmnt/SPS** ）を作成･拡張し、node-a と node-b 間で /export/sapmnt/SPS ディレクトリーの内容をミラーリング

します。DataKeeper リソースは node-a で作成され、node-b に拡張されることに注意してください。 また、作成されたリソースは、Witness ノードである node-c に拡張してはならないことに注意してください。 アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent |
| Server | node-a |
| Hierarchy Type | Replicate New Filesystem |
| Source Disk | /dev/sdb1 (8.0 GB) |
| New Mount Point | /export/sapmnt/SPS |
| New Filesystem Type | xfs |
| Data Replication Resource Tag | datarep-/export/sapmnt/SPS |
| File System Resource Tag | /export/sapmnt/SPS |
| Bitmap File | /opt/LifeKeeper/bitmap__export_sapmnt_SPS |
| Enable Asynchronous Replication? | no |
| **Pre-Extend Wizard** | |
| Target Server | node-b |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend scsi/netraid Resource Hierarchy Wizard** | |
| Mount Point | /export/sapmnt/SPS |
| Root Tag | /export/sapmnt/SPS |
| Target Disk | /dev/sdb1 (8.0 GB) |
| Data Replication Resource Tag | datarep-/export/sapmnt/SPS |
| Bitmap File | /opt/LifeKeeper/bitmap__export_sapmnt_SPS |
| Replication Path | 10.20.1.10/10.20.2.10 |

2. ファイルシステムのデータレプリケーションの作成方法 で説明している手順に従って、以下のパラメーターを使用してDataKeeperデータレプリケーションリソース（datarep-/export/usr/sap/trans）を作成･拡張し、node-a と node-b 間で/export/usr/sap/transディレクトリの内容をミラーリングします。 DataKeeper リソースは node-a で作成され、node-b に拡張されることに注意してください。 また、作成されたリソースは、Witnessノードである node-c に拡張してはならないことに注意してください。アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent |
| Server | node-a |
| Hierarchy Type | Replicate New Filesystem |
| Source Disk | /dev/sdc1 (8.0 GB) |
| New Mount Point | /export/usr/sap/trans |
| New Filesystem Type | xfs |
| Data Replication Resource Tag | datarep-/export/usr/sap/trans |
| File System Resource Tag | /export/usr/sap/trans |
| Bitmap File | /opt/LifeKeeper/bitmap__export_usr_sap_trans |
| Enable Asynchronous Replication? | no |
| **Pre-Extend Wizard** | |
| Target Server | node-b |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend scsi/netraid Resource Hierarchy Wizard** | |
| Mount Point | /export/usr/sap/trans |
| Root Tag | /export/usr/sap/trans |
| Target Disk | /dev/sdc1 (8.0 GB) |
| Data Replication Resource Tag | datarep-/export/usr/sap/trans |
| Bitmap File | /opt/LifeKeeper/bitmap__export_usr_sap_trans |

| | |
|---|---|
| Replication Path | 10.20.1.10/10.20.2.10 |

3. ファイルシステムのデータレプリケーションの作成方法 で説明している手順に従って、以下のパラメーターを使用してDataKeeperデータレプリケーションリソース（datarep-/usr/sap/SPS/ASS10）を作成･拡張し、node-a と node-b 間で /usr/sap/SPS/ASS10 ディレクトリーの内容をミラーリングします。DataKeeper リソースは node-a で作成され、node-b に拡張されることに注意してください。 また、作成されたリソースは、Witness ノードである node-c に拡張してはならないことに注意してください。 アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent |
| Server | node-a |
| Hierarchy Type | Replicate New Filesystem |
| Source Disk | /dev/sdd1 (8.0 GB) |
| New Mount Point | /usr/sap/SPS/ASCS10 |
| New Filesystem Type | xfs |
| Data Replication Resource Tag | datarep-/usr/sap/SPS/ASCS10 |
| File System Resource Tag | /usr/sap/SPS/ASCS10 |
| Bitmap File | /opt/LifeKeeper/bitmap__usr_sap_SPS_ASCS10 |
| Enable Asynchronous Replication? | no |
| **Pre-Extend Wizard** | |
| Target Server | node-b |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend scsi/netraid Resource Hierarchy Wizard** | |
| Mount Point | /usr/sap/SPS/ASCS10 |
| Root Tag | /usr/sap/SPS/ASCS10 |
| Target Disk | /dev/sdd1 (8.0 GB) |

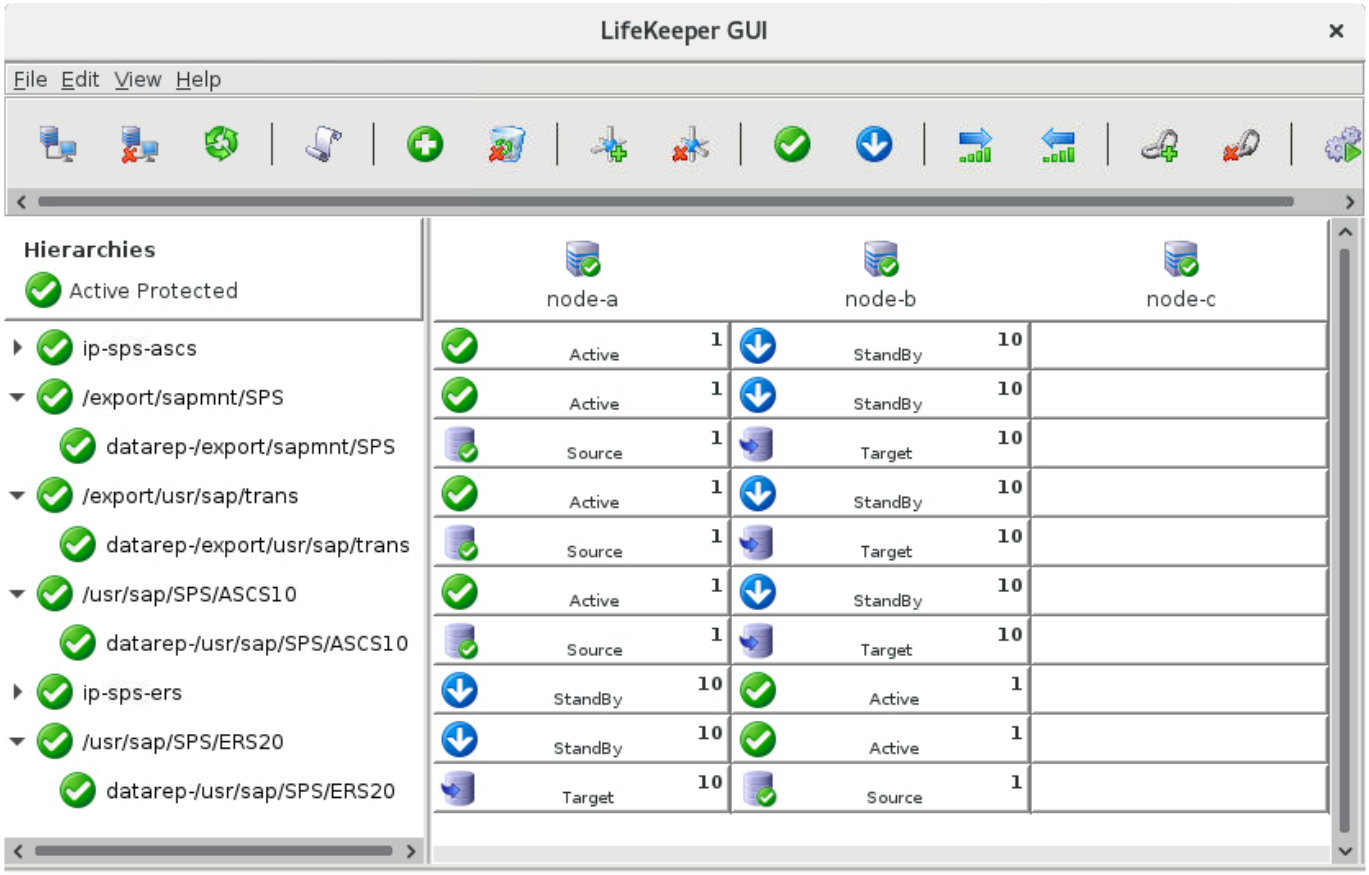
| | |
|---|---|
| Data Replication Resource Tag | datarep-/usr/sap/SPS/ASCS10 |
| Bitmap File | /opt/LifeKeeper/bitmap__usr_sap_SPS_ASCS10 |
| Replication Path | 10.20.1.10/10.20.2.10 |

4. ファイルシステムのデータレプリケーションの作成方法 で説明している手順に従って、以下のパラメーターを使用して DataKeeper データレプリケーションリソース（datarep-/usr/sap/SPS/ERS20）を作成･拡張し、node-a と node-b 間で /usr/sap/SPS/ERS20 ディレクトリーの内容をミラーリングします。 DataKeeper リソースがは node-b で作成され、node-a に拡張されることに注意してください。 また、作成されたリソースは、Witness ノードである node-c に拡張してはならないことに注意してください。アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent |
| Server | node-b |
| Hierarchy Type | Replicate New Filesystem |
| Source Disk | /dev/sde1 (4.0 GB) |
| New Mount Point | /usr/sap/SPS/ERS20 |
| New Filesystem Type | xfs |
| Data Replication Resource Tag | datarep-/usr/sap/SPS/ERS20 |
| File System Resource Tag | /usr/sap/SPS/ERS20 |
| Bitmap File | /opt/LifeKeeper/bitmap__usr_sap_SPS_ERS20 |
| Enable Asynchronous Replication? | no |
| **Pre-Extend Wizard** | |
| Target Server | node-a |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend scsi/netraid Resource Hierarchy Wizard** | |
| Mount Point | /usr/sap/SPS/ERS20 |

| | |
|---|---|
| Root Tag | /usr/sap/SPS/ERS20 |
| Target Disk | /dev/sde1 (4.0 GB) |
| Data Replication Resource Tag | datarep-/usr/sap/SPS/ERS20 |
| Bitmap File | /opt/LifeKeeper/bitmap__usr_sap_SPS_ERS20 |
| Replication Path | 10.20.2.10/10.20.1.10 |

すべての DataKeeper データレプリケーションリソース階層が作成されると、LifeKeeper GUI は以下の画像のようになります。




# NFS リソース階層を作成する

NFS リソースの保護 で説明しているように、node-a と node-b の両方に NFS サーバーとクライアントユーティリティがインストールされ、両方のノードで `rpcbind` と `nfs-server` サービスが起動時に実行できるようにされていることを確認してください。

1. RedHat Enterprise Linux 8.x にデプロイする場合は、/etc/nfs.conf の [nfsd] セクションに「udp=y」という行を追加し、 nfs-server サービスを再起動して、UDP による NFSv3 を有効にします。この作業は、node-a と node-b の両方で行う必要があります。

```
# vi /etc/nfs.conf
```

```
# cat /etc/nfs.conf
<snip>
[nfsd]
<snip>
udp=y
<snip>
# systemctl restart nfs-server
```

2. node-a の /etc/export sに以下のエントリーを追加します。

```
/export/sapmnt/SPS *(rw,sync,no_root_squash)
/export/usr/sap/trans *(rw,sync,no_root_squash)
```

> 注意： ここでは、簡単にするためにすべてのクライアントシステムがこれらの共有ファイルシステムをマウントできるようにしていますが、SAP 環境内の必要なサーバーだけにアクセスを制限するために、これらのエントリーを変更することもできます。

3. node-a で以下のコマンドを実行し、共有ファイルシステムをエクスポートします。

```
[root@node-a ~]# exportfs -rav
exporting *:/export/usr/sap/trans
exporting *:/export/sapmnt/SPS
```

4. 以下のコマンドを実行して、node-a、node-b の両方、および PAS と AAS インスタンスをホストするサーバーから、共有ファイルシステムを表示できることを確認します。

```
# showmount -e sps-ascs
Export list for sps-ascs:
/export/usr/sap/trans *
/export/sapmnt/SPS    *
```

5. NFSリソースの保護 NFS リソース（ **nfs-/export/sapmnt/SPS** ）を作成および拡張し、共有の /export/sapmnt/SPS ファイルシステムを保護します。 NFS リソースは node-a に作成され、node-b に拡張されていることに注意してください。 また、作成されたリソースは、Witness ノードである node-c に拡張してはならないことに注意してください。 ✓ アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent ✓ |
| Server | node-a |

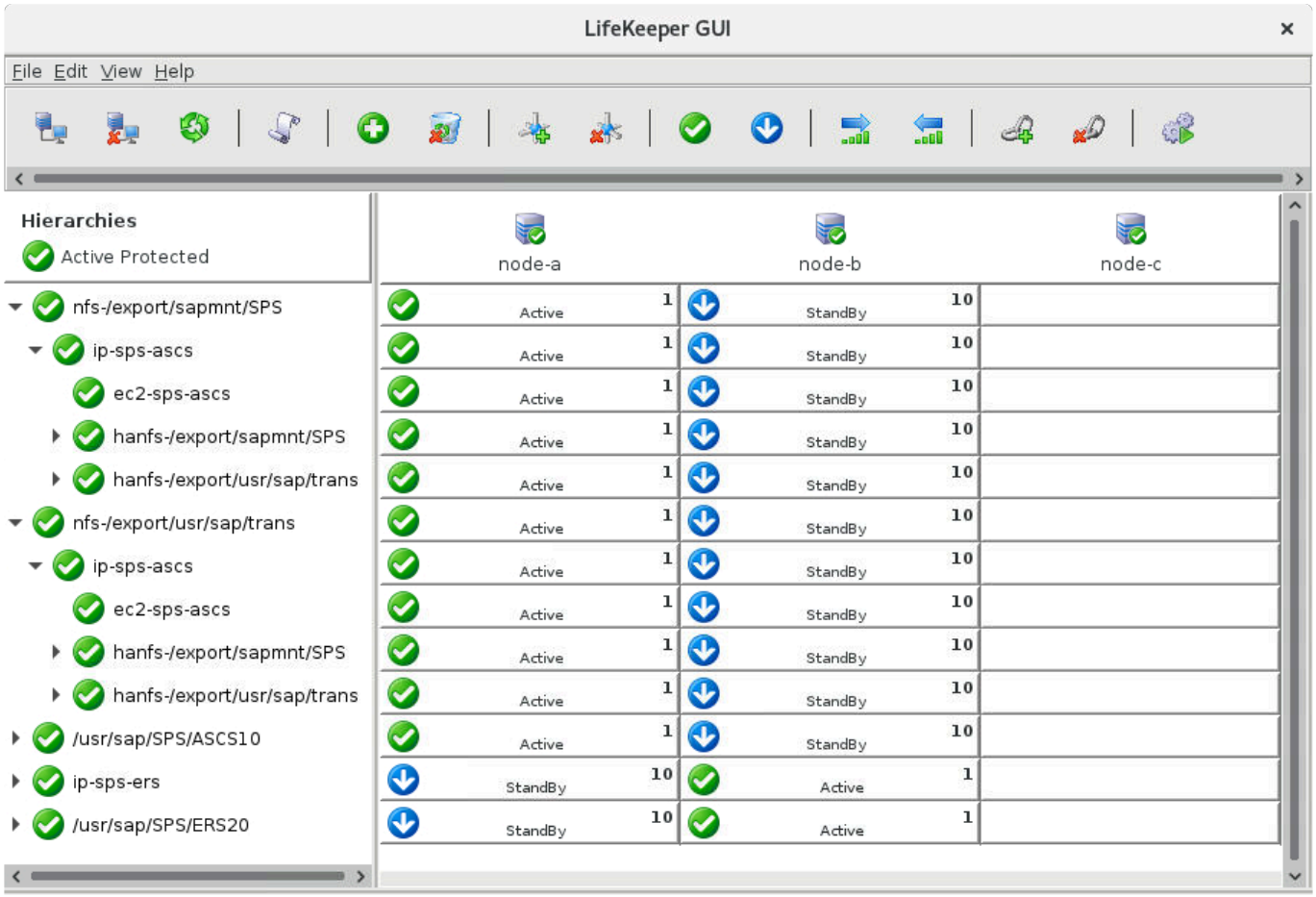
| | |
|---|---|
| Export Point | /export/sapmnt/SPS |
| IP Tag | ip-sps-ascs |
| NFS Tag | nfs-/export/sapmnt/SPS ✓ |
| **Pre-Extend Wizard** | |
| Target Server | node-b |
| Switchback Type | intelligent ✓ |
| Template Priority | 1 ✓ |
| Target Priority | 10 ✓ |
| **Extend gen/nfs Resource Hierarchy Wizard** | |
| NFS Tag | nfs-/export/sapmnt/SPS ✓ |

6. NFSリソースの保護 で説明している手順に従って、次のパラメータを使用してNFSリソース（ **nfs-/export/usr/sap/trans** ）を作成および拡張し、共有ファイルシステム/export/usr/sap/transを保護します。 NFSリソースはnode-aに作成され、node-bに拡張されていることに注意してください。また、作成されたリソースは、Witnessノードであるnode-cに拡張してはならないことに注意してください。 ✓ アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent ✓ |
| Server | node-a |
| Export Point | /export/usr/sap/trans |
| IP Tag | ip-sps-ascs |
| NFS Tag | nfs-/usr/sap/trans ✓ |
| **Pre-Extend Wizard** | |
| Target Server | node-b |
| Switchback Type | intelligent ✓ |
| Template Priority | 1 ✓ |
| Target Priority | 10 ✓ |
| **Extend gen/nfs Resource Hierarchy Wizard** | |

| NFS Tag | nfs-/export/usr/sap/trans |
|---|---|

NFS リソース階層が正常に作成されると、LifeKeeper GUI は次の図のようになります。 ASCS および ERS インスタンスの仮想ホスト名を保護するために使用されるリソースの名前は、ASCS および ERS の仮想 IP を作成する のどのサブセクションに従ったかによって異なる場合があることに注意してください。




## /etc/fstab にマウントエントリーを追加する

1. node-a、node-b、node-d、node-e（ASCS、ERS、PAS、AASのインスタンスホスト）で以下のコマンドを実行し、ブート中に/sapmnt/SPSにsps-ascs:/export/sapmnt/SPSをマウントします。

```
# echo "sps-ascs:/export/sapmnt/SPS    /sapmnt/SPS    nfs    nfsvers=3,proto=udp,rw,sync,bg    0 0" >> /etc/fstab
# mount /sapmnt/SPS
```

2. node-dとnode-e（PASとAASのインスタンスホスト）で以下のコマンドを実行し、ブート中にsps-ascs:/export/usr/sap/transを/usr/sap/transにマウントします。

```
# echo "sps-ascs:/export/usr/sap/trans    /usr/sap/trans    nfs    nfsvers=3,proto=udp,rw,sync,bg    0 0" >> /etc/fstab
```

```
# mount /usr/sap/trans
```

## /etc/default/LifeKeeper を変更する

SAP Recovery Kit が共有ファイルシステム上のファイルへのアクセスを必要とする管理タスクを実行する前に、 sps-ascs:/export/sapmnt/SPS NFS共有の可用性を監視できるように、node-a と node-b の /etc/default/LifeKeeper に「SAP_NFS_CHECK_DIRS=/sapmnt/SPS」という行を追加します。

**重要：** Amazon Elastic File System（EFS）サービス経由でファイルシステムを共有する場合は、SAP_NFS_CHECK_DIRS調整可能パラメータを使用しないでください。EFSサービスは、ローカルNFSサーバーと同じ方法でヘルスPingに応答しないためです。

**注意：** ファイルシステムの共有にNFSv4やTCPを使用している場合、切り替えやフェイルオーバー時に問題や予期せぬハングが発生する場合があります。ただし、ファイル共有の設定にこれらのいずれかまたは両方を使用する必要がある場合は、/etc/default/LifeKeeper のパラメーターを変更する必要があります。NFSv4 を使用する場合、node-a と node-b の両方で /etc/default/LifeKeeper に NFS_VERSION=4という調整可能な値を設定します。TCP（NFSv3またはNFSv4のいずれか）を使用する場合、node-a と node-b の両方で調整可能な値 /etc/default/LifeKeeper に NFS_RPC_PROTOCOL=tcp を設定します。たとえば、TCP 経由で NFSv4 を使用してファイルシステムを共有する場合、node-a と node-b の両方で /etc/default/LifeKeeper に調整可能な値 NFS_VERSION=4 と NFS_RPC_PROTOCOL=tcp の両方を設定する必要があります。

これで、SAP インスタンスをインストールするための共有ファイルシステムの準備が整いました。

# 11.2.7.5.5.2.2. Google Cloud – SAP 共有ファイルシステムおよびレプリケーションファイルシステムの作成

! NFS 共有をエクスポートするサーバー（またはクラスター） が、ASCS + ERS を実行しているサーバーとは別のサーバーグループである場合、NFS 共有を使用した ASCS + ERS がサポートされます。

✻ 注意： このセクションは、Google Cloud でのデプロイの場合の手順です。AWS または Microsoft Azure にデプロイする場合は、AWS/Azure – SAP共有ファイルシステムおよびレプリケーションファイルシステムの作成 に記載されている手順に従ってください。

このセクションでは、SAP インスタンスをホストする各ノードにマウントされる、高可用性 NFS 共有ファイルシステムを作成します。また、SIOS DataKeeper ミラーを作成して、node-a と node-b の間で ASCS10 および ERS20 インスタンスのデータを複製します。高可用性の SAP AS ABAP 環境において、システム間で共有または複製される典型的なファイルシステムには、次のようなものがあります。

- /sapmnt/<SID>
- /usr/sap/trans
- /usr/sap/<SID>/ASCS<InstNum>
- /usr/sap/<SID>/ERS<InstNum>

詳細については、高可用性システムのファイルシステムの設定 を参照してください。

✻ 注意： このガイドでは、すべてのリソースがクラスター内で自己完結するように、ファイル共有とレプリケーションのメカニズムを実装することを選択しましたが、他の構成も可能です。たとえば、NFS 共有ファイルシステムは、可用性の高い外部 NFS サーバークラスターでホストしたり、NetApp Cloud Volumes Service などのクラウドネイティブ NFS ソリューションでホストすることができます。

1. 高可用性 SAP のインストールをサポートするために、node-a と node-b に4つの追加ディスクを作成し、アタッチします。この例で使用されているデバイス名は、お使いの環境によって異なる場合があります（例：/dev/sdbではなく/dev/xvdb）ので、適切なデバイス名を使用するように、このセクションで示されるコマンドを適宜調節してください。これらのディスクサイズは、評価目的にのみ使用されることに注意してください。本番環境でリソースをプロビジョニングする場合は、SAP、クラウドプロバイダー、およびサードパーティストレージプロバイダーの関連ドキュメントを参照してください。

| Device Name | Minimum Size | Mount Point |
|---|---|---|
| /dev/sdb | 10GB | /export/sapmnt/SPS |

| | | |
|---|---|---|
| /dev/sdc | 10GB | /export/usr/sap/trans |
| /dev/sdd | 10GB | /usr/sap/SPS/ASCS10 |
| /dev/sde | 10GB | /usr/sap/SPS/ERS20 |

✻ 注意： SAP のインストールをサポートするためにストレージを接続する場合は、他の構成を使用できます。 たとえば、LVM を介して複数の論理パーティションで分割された単一の物理ボリュームを使用すると便利な場合があります。

2. node-a と node-b の両方で以下のコマンドを実行し、xfs ファイルシステムで /dev/sdb1 パーティションを作成します。

```
[root@node-a ~]# parted /dev/sdb --script mklabel gpt mkpart xfspart xfs 0% 10
0%

[root@node-a ~]# mkfs.xfs /dev/sdb1
meta-data=/dev/sdb1    isize=512    agcount=4, agsize=1310592 blks
         =             sectsz=4096 attr=2, projid32bit=1
         =             crc=1       finobt=1, sparse=1, rmapbt=0
         =             reflink=1
data     =             bsize=4096  blocks=5242368, imaxpct=25
         =             sunit=0     swidth=0 blks
naming   =version 2    bsize=4096  ascii-ci=0, ftype=1
log      =internal log bsize=4096  blocks=2560, version=2
         =             sectsz=4096 sunit=1 blks, lazy-count=1
realtime =none         extsz=4096  blocks=0, rtextents=0

[root@node-a ~]# partprobe /dev/sdb1
```

Node-a、Node-b の両方で、/dev/sdc、/dev/sdd、/dev/sdeに対して、このコマンドを繰り返します

3. Node-a と node-b の両方で次のコマンドを実行して、共有および複製された SA Pファイルシステムのマウントポイントを作成します。

```
[root@node-a ~]# mkdir -p /export/{usr/sap/trans,sapmnt/SPS} /sapmnt/SPS /usr/
sap/{trans,SPS/{ASCS10,ERS20,SYS}}
```

## DataKeeper のデータレプリケーションリソース階層を作成する

1. “ファイルシステムのデータレプリケーションの作成方法”#how-to-create-data-replication-of-a-file-system で説明した手順に従って、以下のパラメーターを使用して DataKeeper データレプリケーションリソース（ **datarep-/export/sapmnt/SPS** ）を作成･拡張し、node-a と node-b 間で /export/sapmnt/SPS ディレクトリーの内容をミラーリングします。DataKeeper リソースは node-a で作成

され、node-b に拡張されることに注意してください。また、作成されたリソースは、Witness ノードである node-c に拡張してはならないことに注意してください。 アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent |
| Server | node-a |
| Hierarchy Type | Replicate New Filesystem |
| Source Disk | /dev/sdb1 (10.0 GB) |
| New Mount Point | /export/sapmnt/SPS |
| New Filesystem Type | xfs |
| Data Replication Resource Tag | datarep-/export/sapmnt/SPS |
| File System Resource Tag | /export/sapmnt/SPS |
| Bitmap File | /opt/LifeKeeper/bitmap__export_sapmnt_SPS |
| Enable Asynchronous Replication? | no |
| **Pre-Extend Wizard** | |
| Target Server | node-b |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend scsi/netraid Resource Hierarchy Wizard** | |
| Mount Point | /export/sapmnt/SPS |
| Root Tag | /export/sapmnt/SPS |
| Target Disk | /dev/sdb1 (10.0 GB) |
| Data Replication Resource Tag | datarep-/export/sapmnt/SPS |
| Bitmap File | /opt/LifeKeeper/bitmap__export_sapmnt_SPS |
| Replication Path | 10.20.1.10/10.20.2.10 |

2. ファイルシステムのデータレプリケーションの作成方法 で説明している手順に従って、以下のパラメーターを使用して DataKeeper データレプリケーションリソース（ **datarep-/export/usr/sap/trans** ）を作成･拡張し、node-a と node-b 間で /export/usr/sap/trans ディレクトリの内容をミラーリングします。DataKeeper リソースは node-a で作成され、node-b に拡張されることに注意してください。また、作成されたリソースは、Witness ノードである node-c に拡張してはならないことに注意してください。 アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent |
| Server | node-a |
| Hierarchy Type | Replicate New Filesystem |
| Source Disk | /dev/sdc1 (10.0 GB) |
| New Mount Point | /export/usr/sap/trans |
| New Filesystem Type | xfs |
| Data Replication Resource Tag | datarep-/export/usr/sap/trans |
| File System Resource Tag | /export/usr/sap/trans |
| Bitmap File | /opt/LifeKeeper/bitmap__export_usr_sap_trans |
| Enable Asynchronous Replication? | no |
| **Pre-Extend Wizard** | |
| Target Server | node-b |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend scsi/netraid Resource Hierarchy Wizard** | |
| Mount Point | /export/usr/sap/trans |
| Root Tag | /export/usr/sap/trans |
| Target Disk | /dev/sdc1 (10.0 GB) |
| Data Replication Resource Tag | datarep-/export/usr/sap/trans |

| | |
|---|---|
| Bitmap File | /opt/LifeKeeper/bitmap__export_usr_sap_trans |
| Replication Path | 10.20.1.10/10.20.2.10 |

3. ファイルシステムのデータレプリケーションの作成方法 で説明している手順に従って、以下のパラメーターを使用して DataKeeper データレプリケーションリソース（datarep-/usr/sap/SPS/ASS10）を作成･拡張し、node-a と node-b 間で /usr/sap/SPS/ASS10 ディレクトリーの内容をミラーリングします。DataKeeper リソースは node-a で作成され、node-b に拡張されることに注意してください。また、作成されたリソースは、Witness ノードである node-c に拡張してはならないことに注意してください。 アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent |
| Server | node-a |
| Hierarchy Type | Replicate New Filesystem |
| Source Disk | /dev/sdd1 (10.0 GB) |
| New Mount Point | /usr/sap/SPS/ASCS10 |
| New Filesystem Type | xfs |
| Data Replication Resource Tag | datarep-/usr/sap/SPS/ASCS10 |
| File System Resource Tag | /usr/sap/SPS/ASCS10 |
| Bitmap File | /opt/LifeKeeper/bitmap__usr_sap_SPS_ASCS10 |
| Enable Asynchronous Replication? | no |
| **Pre-Extend Wizard** | |
| Target Server | node-b |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend scsi/netraid Resource Hierarchy Wizard** | |
| Mount Point | /usr/sap/SPS/ASCS10 |

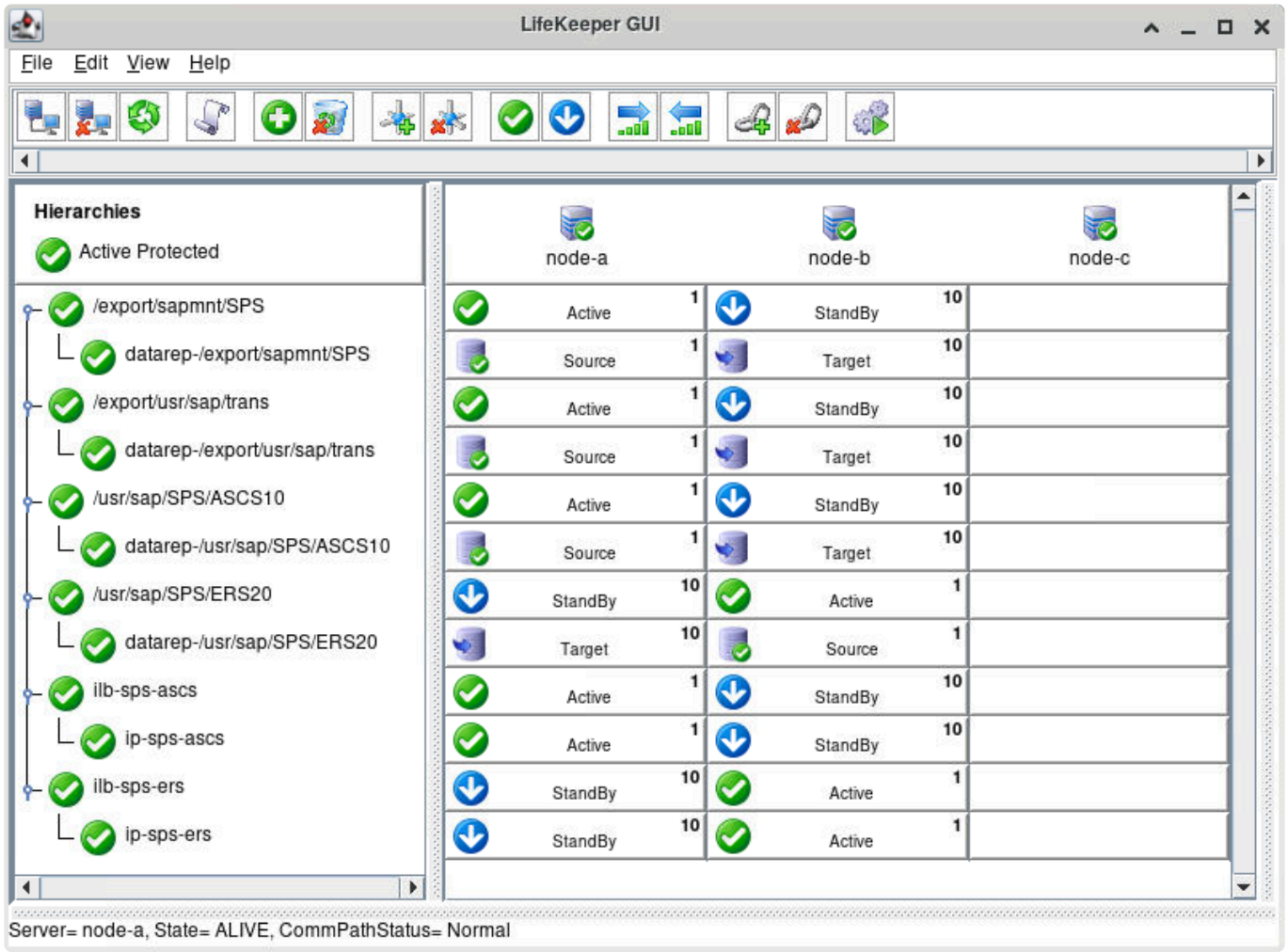
| | |
|---|---|
| Root Tag | /usr/sap/SPS/ASCS10 |
| Target Disk | /dev/sdd1 (10.0 GB) |
| Data Replication Resource Tag | datarep-/usr/sap/SPS/ASCS10 |
| Bitmap File | /opt/LifeKeeper/bitmap__usr_sap_SPS_ASCS10 |
| Replication Path | 10.20.1.10/10.20.2.10 |

4. ファイルシステムのデータレプリケーションの作成方法 で説明している手順に従って、以下のパラメーターを使用して DataKeeper データレプリケーションリソース（ **datarep-/usr/sap/SPS/ERS20** ）を作成･拡張し、node-a と node-b 間で /usr/sap/SPS/ERS20 ディレクトリーの内容をミラーリングします。 DataKeeper リソースがは node-b で作成され、node-a に拡張されることに注意してください。 また、作成されたリソースは、Witness ノードである node-c に拡張してはならないことに注意してください。アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent |
| Server | node-b |
| Hierarchy Type | Replicate New Filesystem |
| Source Disk | /dev/sde1 (10.0 GB) |
| New Mount Point | /usr/sap/SPS/ERS20 |
| New Filesystem Type | xfs |
| Data Replication Resource Tag | datarep-/usr/sap/SPS/ERS20 |
| File System Resource Tag | /usr/sap/SPS/ERS20 |
| Bitmap File | /opt/LifeKeeper/bitmap__usr_sap_SPS_ERS20 |
| Enable Asynchronous Replication? | no |
| **Pre-Extend Wizard** | |
| Target Server | node-a |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |

| Extend scsi/netraid Resource Hierarchy Wizard | |
|---|---|
| Mount Point | /usr/sap/SPS/ERS20 ✓ |
| Root Tag | /usr/sap/SPS/ERS20 ✓ |
| Target Disk | /dev/sde1 (10.0 GB) |
| Data Replication Resource Tag | datarep-/usr/sap/SPS/ERS20 ✓ |
| Bitmap File | /opt/LifeKeeper/bitmap__usr_sap_SPS_ERS20 ✓ |
| Replication Path | 10.20.2.10/10.20.1.10 |

すべての DataKeeper データレプリケーションリソース階層が作成されると、LifeKeeper GUI は以下の画像のようになります。




# NFS リソース階層を作成する

NFSリソースの保護 で説明しているように、node-a と node-b の両方に NFS サーバーとクライアントユーティリティがインストールされ、両方のノードで rpcbind と rpcbind

サービスが起動時に実行できるようになっていることを確認してください。

1. node-aの/etc/exportsに以下のエントリーを追加します。

```
/export/sapmnt/SPS *(rw,sync,no_root_squash)
/export/usr/sap/trans *(rw,sync,no_root_squash)
```

**注意：** ここでは、簡単にするためにすべてのクライアントシステムがこれらの共有ファイルシステムをマウントできるようにしていますが、SAP 環境内の必要なサーバーだけにアクセスを制限するために、これらのエントリーを変更することもできます。

2. node-aで以下のコマンドを実行し、共有ファイルシステムをエクスポートします。

```
[root@node-a ~]# exportfs -rav
exporting *:/export/usr/sap/trans
exporting *:/export/sapmnt/SPS
```

3. 以下のコマンドを実行して、node-a、node-bの両方、およびPASとAASインスタンスをホストするサーバーから、共有ファイルシステムを表示できることを確認します。

```
# showmount -e sps-ascs
Export list for sps-ascs:
/export/usr/sap/trans *
/export/sapmnt/SPS    *
```

4. NFSリソースの保護 で説明している手順に従って、次のパラメーターを使用してNFSリソース（**nfs-/export/sapmnt/SPS**）を作成および拡張し、共有の /export/sapmnt/SPS ファイルシステムを保護します。 NFSリソースは node-a に作成され、node-b に拡張されていることに注意してください。また、作成されたリソースは、Witness ノードである node-c に拡張してはならないことに注意してください。 ✓ アイコンは、デフォルトのオプションが選択されていることを示します。

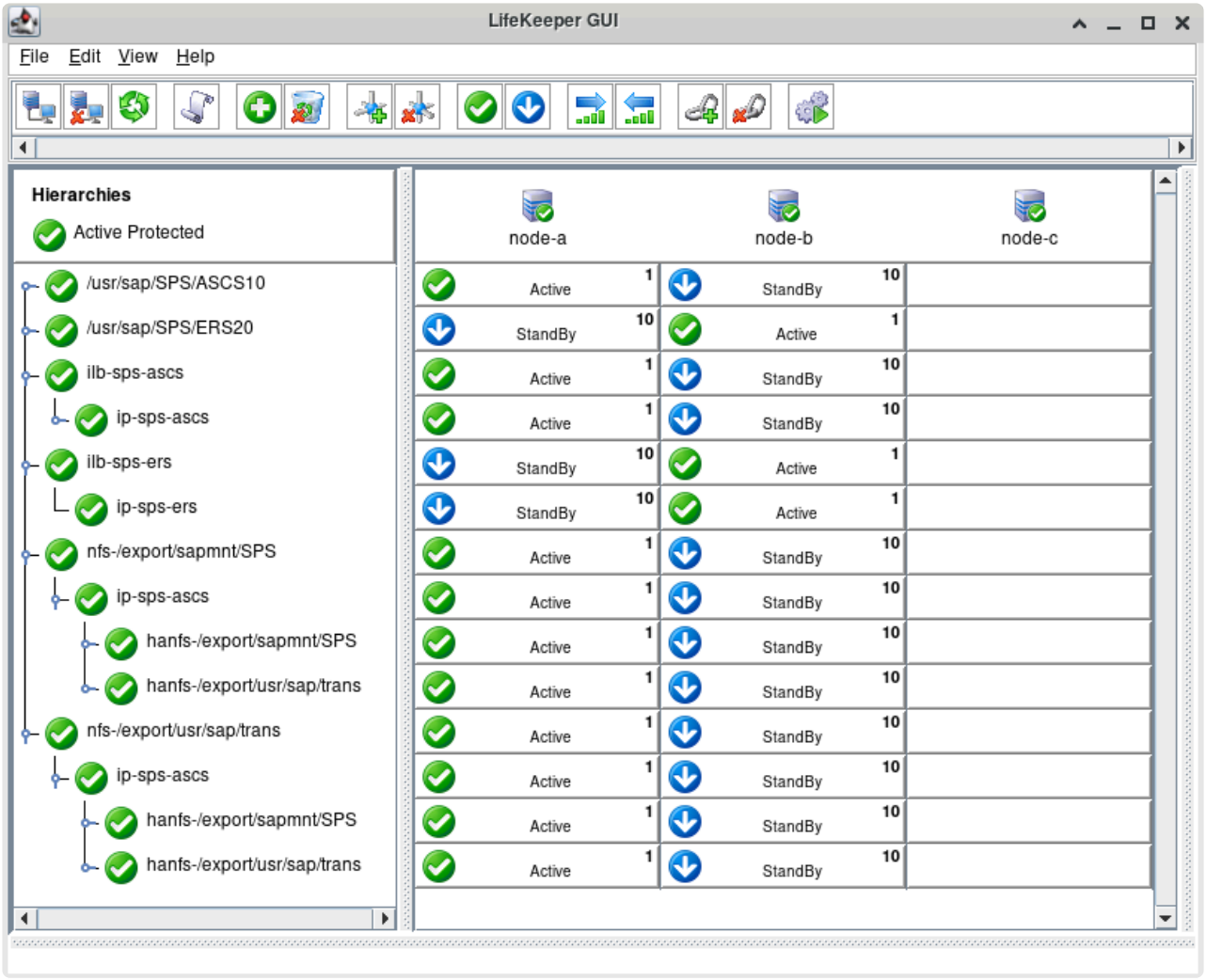
| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent ✓ |
| Server | node-a |
| Export Point | /export/sapmnt/SPS |
| IP Tag | ip-sps-ascs |
| NFS Tag | nfs-/export/sapmnt/SPS ✓ |

| **Pre-Extend Wizard** | |
|---|---|
| Target Server | node-b |
| Switchback Type | intelligent ✓ |
| Template Priority | 1 ✓ |
| Target Priority | 10 ✓ |
| **Extend gen/nfs Resource Hierarchy Wizard** | |
| NFS Tag | nfs-/export/sapmnt/SPS ✓ |

5. NFSリソースの保護 で説明している手順に従って、次のパラメーターを使用してNFS リソース（ **nfs-/export/usr/sap/trans** ）を作成および拡張し、共有ファイルシステム /export/usr/sap/trans を保護します。 NFS リソースは node-a に作成され、node-b に拡張されていることに注意してください。 また、作成されたリソースは、Witness ノードである node-c に拡張してはならないことに注意してください。 ✓ アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent ✓ |
| Server | node-a |
| Export Point | /export/usr/sap/trans |
| IP Tag | ip-sps-ascs |
| NFS Tag | nfs-/usr/sap/trans ✓ |
| **Pre-Extend Wizard** | |
| Target Server | node-b |
| Switchback Type | intelligent ✓ |
| Template Priority | 1 ✓ |
| Target Priority | 10 ✓ |
| **Extend gen/nfs Resource Hierarchy Wizard** | |
| NFS Tag | nfs-/export/usr/sap/trans ✓ |

NFS リソース階層が正常に作成されると、LifeKeeper GUI は次の図のようになります。



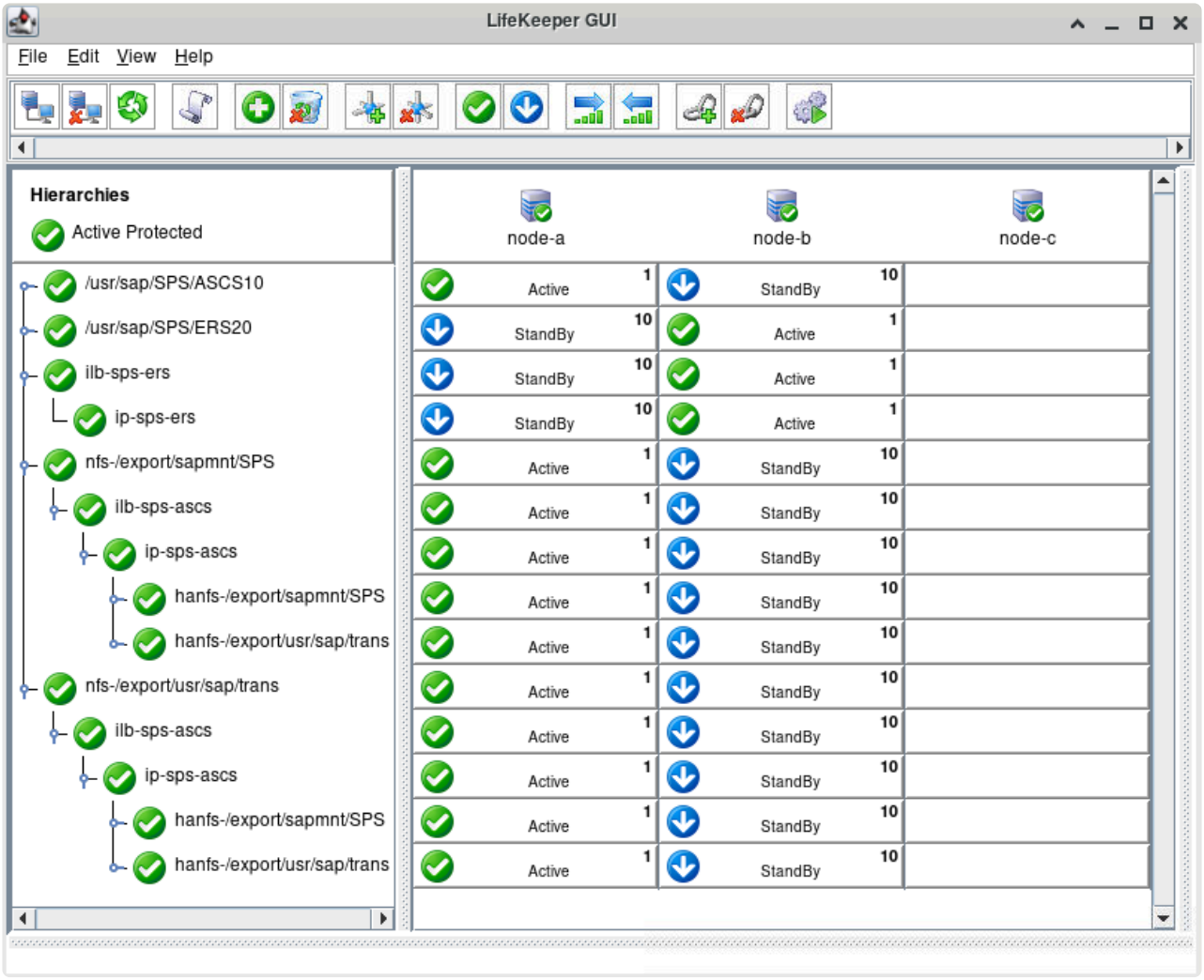

6. **nfs-/export/sapmnt/SPS** リソースを右クリックし、ドロップダウンメニューから **[Delete Dependency...]** を選択します。Child Resource Tag に **[ip-sps-ascs]** を選択します。 **[Next>]** をクリックして続行し、 **[Delete Dependency]** をクリックして依存関係を削除します。

7. **nfs-/export/sapmnt/SPS** リソースを右クリックし、ドロップダウンメニューから **[Create Dependency...]** を選択します。Child Resource Tag に **[ilb-sps-ascs]** を選択します。 **[Next>]** をクリックして続行し、 **[Create Dependency]** をクリックして依存関係を作成します。

8. **nfs-/export/usr/sap/trans** リソースを右クリックし、ドロップダウンメニューから **[Delete Dependency...]** を選択します。Child Resource Tag に **[ip-sps-ascs]** を選択します。*[Next>]* をクリックして続行し、*[Delete Dependency]* をクリックして依存関係を削除します。

9. **nfs-/export/usr/sap/trans** リソースを右クリックし、ドロップダウンメニューから **[Create Dependency...]** を選択します。Child Resource Tag に **[ilb-sps-ascs]** を選択します。 **[Next>]** をクリックして続行し 、 **[Create Dependency]** をクリックして依存関係を作成します。

これらの手順が完了すると、LifeKeeper GUI は以下の画像のようになります。




## /etc/fstab にマウントエントリーを追加する

1. node-a、node-b、node-d、node-e（ASCS、ERS、PAS、AASのインスタンスホスト）で以下のコマンドを実行し、ブート中に/sapmnt/SPSにsps-ascs:/export/sapmnt/SPSをマウントします。

```
# echo "sps-ascs:/export/sapmnt/SPS    /sapmnt/SPS    nfs    nfsvers=3,proto=t
cp,rw,sync,bg    0 0" >> /etc/fstab
# mount /sapmnt/SPS
```

2. node-dとnode-e（PASとAASのインスタンスホスト）で以下のコマンドを実行し、ブート中にsps-ascs:/export/usr/sap/transを/usr/sap/transにマウントします。

```
# echo "sps-ascs:/export/usr/sap/trans    /usr/sap/trans    nfs    nfsvers=3,p
roto=tcp,rw,sync,bg    0 0" >> /etc/fstab
# mount /usr/sap/trans
```

## /etc/default/LifeKeeper を変更する

SAP Recovery Kit が共有ファイルシステム上のファイルへのアクセスを必要とする管理タスクを実行する前に、 sps-ascs:/export/sapmnt/SPS NFS共有の可用性を監視できるように、node-a と node-b の /etc/

default/LifeKeeper に「SAP_NFS_CHECK_DIRS=/sapmnt/SPS」という行を追加します。また、「NFS_RPC_PROTOCOL=tcp」という行をnode-aとnode-b の /etc/default/LifeKeeper に追加して、pingnfsユーティリティがNFS共有の可用性を確認する際にTCPプロトコルを使用するようにします。

```
# vi /etc/default/LifeKeeper
# grep 'SAP_NFS_CHECK_DIRS=\|NFS_RPC_PROTOCOL=' /etc/default/LifeKeeper
SAP_NFS_CHECK_DIRS=/sapmnt/SPS
NFS_RPC_PROTOCOL=tcp
```

これで、SAP インスタンスをインストールするための共有ファイルシステムの準備が整いました。

# 11.2.7.5.5.3. SAPインスタンスのインストール

以下の手順で、node-aおよびnode-bにASCS10およびERS20インスタンスをインストールします。

## インストールの準備

1. node-aおよびnode-bにおいて、SAPシステムの要件を満たすために、利用可能なスワップ領域を1MiBより大きくします。この例では、1GiBのスワップファイルを作成して、利用可能なスワップ領域を増やします。

> **Note:** 本番環境では、より多くのスワップ領域が必要になる可能性があります。詳細については、SAP Note 1597355 (Swap space recommendation for Linux)を参照してください。

```
# mkdir /swap
# dd if=/dev/zero of=/swap/swapfile.img bs=1024 count=1M
# chmod 600 /swap/swapfile.img
# mkswap /swap/swapfile.img
# echo "/swap/swapfile.img    swap    swap    sw    0 0" >> /etc/fstab
# swapon /swap/swapfile.img
```

2. tcshパッケージがnode-aおよびnode-bの両方にインストールされていることを確認します。SLESサーバーにインストールする場合は、このコマンドを変更します（例：`zypper install` を使用する）。

```
# yum install -y tcsh
```

3. node-aおよびnode-bで、SAPのインストールファイルを格納する/sap-installディレクトリを作成します。

```
# mkdir /sap-install
```

> **Note:** 選択したインスタンスタイプのルートファイルシステムに、必要なSAPインストールファイルを抽出後に保存するための十分なディスクスペースがない場合、SAPソフトウェアのインストールを完了するために、各インスタンスに一時ディスクをアタッチし、それを/sap-installにマウントすることができます。

4. SAPCAR、SAP Software Provisioning Manager 2.0（SWPM20SPxx_x-xxxx.SAR） および必要なすべてのパッケージ（SAPEXE_xx-xxxx.SAR、SAPHOSTAGENTxx_xx-xxxx.SAR など） を両ノードの /sap-install ディレクトリにダウンロードします。SAPCARが実行可能であることを確認します。

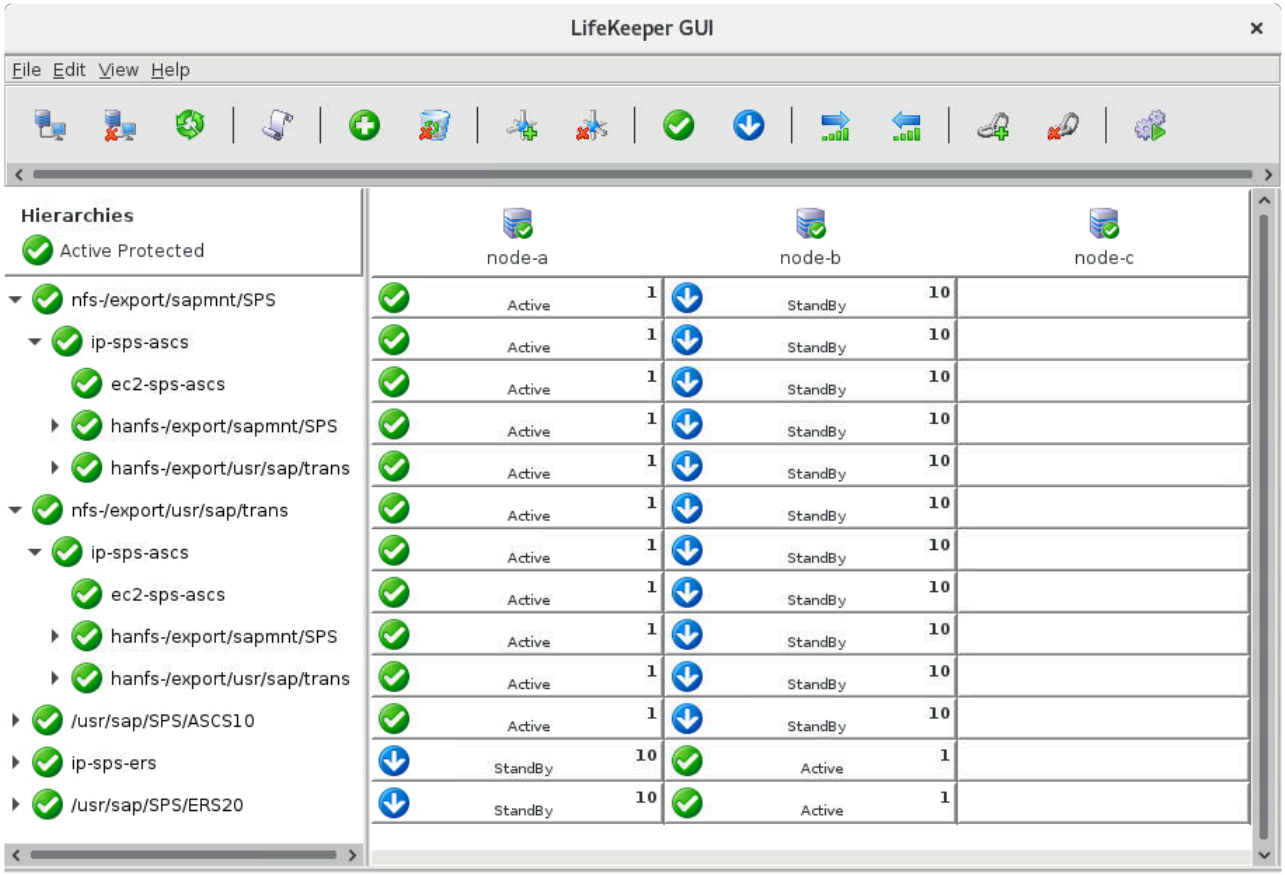
```
# chmod +x /sap-install/SAPCAR
```

5. SAPCARを使用してSWPM2.0用のファイルを展開します。

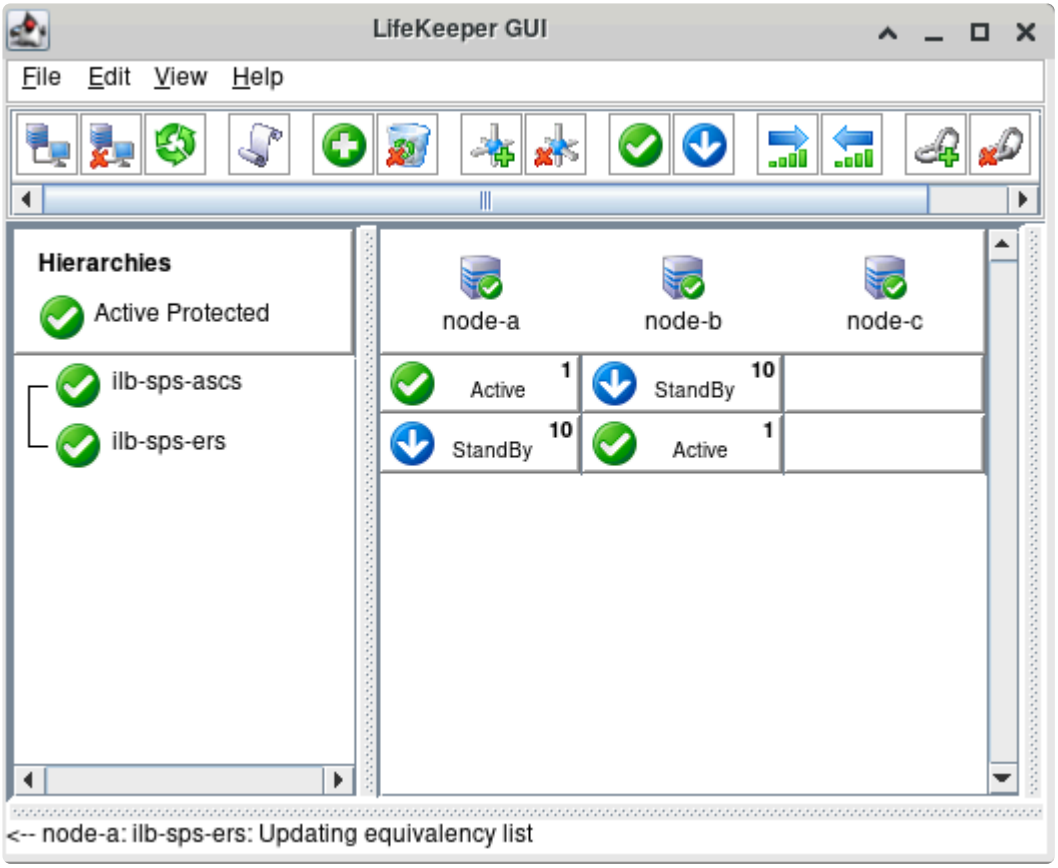
```
# cd /sap-install
# ./SAPCAR -xvf SWPM20SPxx_x-xxxx.SAR
```

SAPインスタンスをインストールする前に、SAPファイルシステムの作成 の該当するクラウド固有のサブセクションで説明されているすべての共有および複製ファイルシステムがnode-aとnode-bのいずれかまたは両方に正しくマウントされていること、ASCSおよびERSインスタンスの仮想ホスト名フェイルオーバーを行うLifeKeeperリソースがそれぞれnode-aおよびnode-bでIn-Serviceであることを確認します。

AWS または Microsoft Azure にデプロイする場合、LifeKeeper GUI は次の図のようになります。




Google Cloud にデプロイする場合、LifeKeeper GUI は次の画像のようになります。




## node-AにASCSインスタンスをインストールする

以下の手順を実行して、ASCS10インスタンスをnode-aにインストールします。

1. リモートシステムからインストールする場合、node-aのTCPポート4237（sapinstグラフィカルインターフェースを公開するポート）のインバウンドトラフィックを許可するファイアーウォールルールを追加します。

2. node-aで以下のコマンドを実行し、ASCSインスタンスに関連付けられた仮想ホスト名でsapinstを起動します。

```
[root@node-a ~]# /sap-install/sapinst SAPINST_USE_HOSTNAME=sps-ascs
```

**Note:** SAPINST_USE_HOSTNAME パラメーターの値として使用される仮想ホスト名は、ASCS および ERS 仮想 IP の作成 のセクションで作成および保護された ASCS インスタンス仮想 IP に解決する必要があります。

**Note:** SAPINST_REMOTE_ACCESS_USER= パラメーターを使用して、root 以外のユーザが sapinst グラフィカル インターフェイスにログインできるようにすることができます。

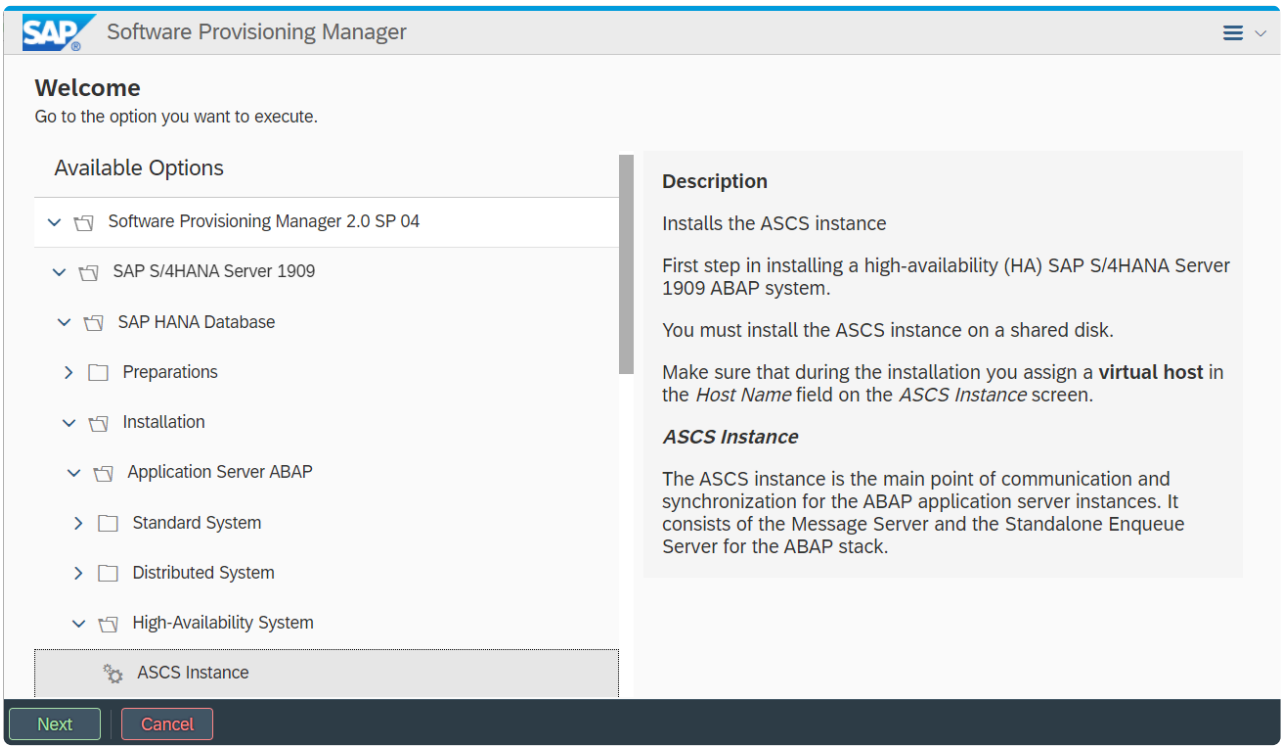
**Note:** dt_socket トランスポートライブラリーがないために sapinst がエラーを表示する場

合は、`yum -y install libnsl` を使用して libnsl パッケージをインストールしてください。

3. sapinst グラフィカルインターフェースにアクセスするには、Webブラウザを使用して https://:4237/sapinst/docs/index.html に移動します。

4. **Software Provisioning Manager 2.0 SP XX → SAP S/4HANA Server 1909 → SAP HANA Database → Installation → Application Server ABAP → High-Availability System → ASCS Instance** の順に選択し、 **Next** をクリックします。

**Note:** このガイドに記載されているスクリーンショットは、Software Provisioning Manager 2.0 SP 04 固有のものであり、他のバージョンの Software Provisioning Manager を使用する場合はインターフェースが異なる場合があります。




5. ASCSインスタンスをインストールする際に、以下のパラメータを指定します。 アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| SAP System ID (SAPSID) | SPS |
| SAP Mount Directory | /sapmnt |
| Password for All Users | SAP User Password |
| Password of SAP System Administrator | <SAP Admin Password> |

| | |
|---|---|
| User ID | 空欄のままにします |
| Group ID of sapsys | 空欄のままにします |
| Specify path to SAPEXE.SAR | /sap-install |
| Specify path to SAPHOSTAGENT.SAR | /sap-install |
| ASCS Instance Number | 10 |
| ASCS Instance Host Name | sps-ascs |
| ABAP Message Server Port | 3610 |
| Internal ABAP Message Server Port | 3910 |
| Yes, clean up operating system users | Click |

6. パラメーターを確認し、インストールを続行します。

**Note:** SAP ファイルシステムの権限が不十分なためにインストールに失敗した場合は、以下の **SAPファイルシステムの権限の修正** セクションの説明に従って権限を修正し、インストールを再試行してください。

7. Node-aで以下のコマンドを実行し、SAPのインストールファイルをクリーンアップします。

```
[root@node-a ~]# rm -rf /root/.sapinst
```

## node-BでERSインスタンスのインストールを準備する

次の手順を実行して、node-b で ERS インスタンスのインストールの準備をします。特にこの手順では、必要なユーザーとグループを作成し、SAP Host Agent をインストールして、必要なバイナリを node-b の /usr/sap/SPS/SYS ディレクトリにインストールします。

1. リモート システムからインストールする場合は、ファイアーウォールルールを追加して、node-b の TCP ポート 4237（sapinst グラフィカルインターフェースを公開するポート）でインバウンドトラフィックを許可します。

2. node-bで以下のコマンドを実行し、sapinstを起動します。

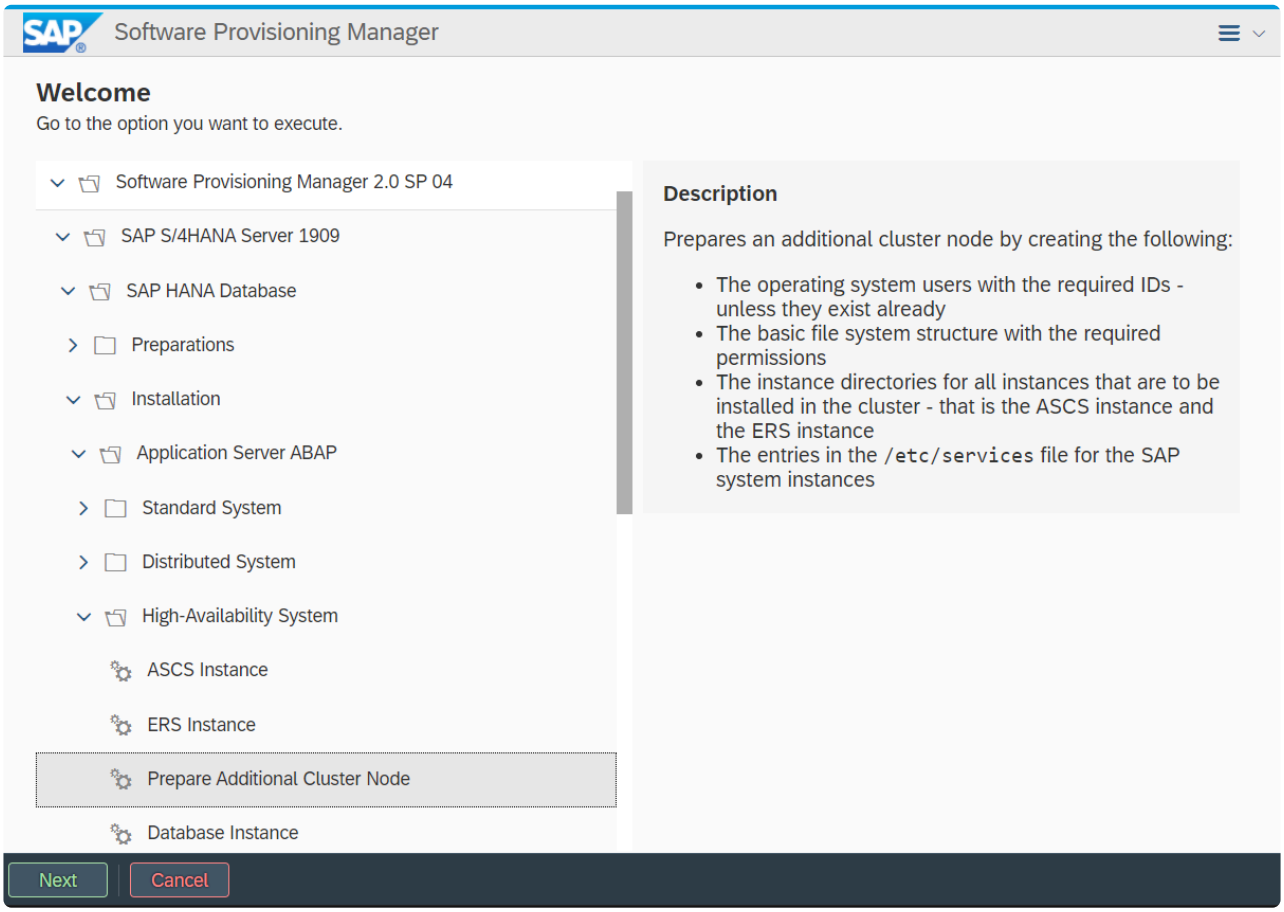
```
[root@node-b ~]# /sap-install/sapinst
```

**Note:** `SAPINST_REMOTE_ACCESS_USER=` パラメーターを使用して、root 以外のユーザが sapinst グラフィカル インターフェイスにログインできるようにすることができます。

**Note:** dt_socket トランスポートライブラリがないために sapinst がエラーを表示する場合は、`yum install -y libnsl` を使用して libnsl パッケージをインストールしてください。

3. sapinst グラフィカルインターフェースにアクセスするには、Web ブラウザーを使用して https://:4237/sapinst/docs/index.html に移動します。

4. **Software Provisioning Manager 2.0 SP XX → SAP S/4HANA Server 1909 → SAP HANA Database → Installation → Application Server ABAP → High-Availability System → Prepare Additional Cluster Node** の順に選択し、 **Next** をクリックします。




5. ERSインスタンスをインストールするためにnode-bで準備をする際、以下のパラメーターを指定します。

| Field | Value |
|---|---|
| Profile Directory | /sapmnt/SPS/profile |
| Password of SAP System Administrator | <SAP Admin Password> |
| Specify path to SAPHOSTAGENT.SAR | /sap-install |
| Yes, clean up operating system users | Click |

6. パラメーターを確認し、インストールを続行します。

**Note:** SAP ファイルシステムの権限が不十分なためにインストールに失敗した場合は、以下の「**SAP ファイルシステムの権限の修正**」セクションの説明に従って権限を修正し、インストールを再試行してください。

7. node-bで以下のコマンドを実行し、SAPのインストールファイルをクリーンアップします。

```
[root@node-b ~]# rm -rf /root/.sapinst
```

## SAP ファイルシステムの権限の修正

node-aとnode-bの両方で以下のコマンドを実行し、SAPファイルシステムに正しい権限を設定します。

```
# chown -R spsadm /export/sapmnt/SPS /sapmnt/SPS /usr/sap/SPS
# chgrp -R sapsys /export/{usr/sap,sapmnt} /sapmnt /usr/sap
```

## node-BにERSインスタンスをインストールする

以下の手順を実行して、ERS20インスタンスをnode-bにインストールします。

1. リモートシステムからインストールする場合、node-bのTCPポート4237（sapinstグラフィカルインターフェースを公開するポート）のインバウンドトラフィックを許可するファイアーウォールルールを追加します。

2. node-bで以下のコマンドを実行し、ERSインスタンスに関連付けられた仮想ホスト名でsapinstを開始します。

```
[root@node-b ~]# /sap-install/sapinst SAPINST_USE_HOSTNAME=sps-ers
```

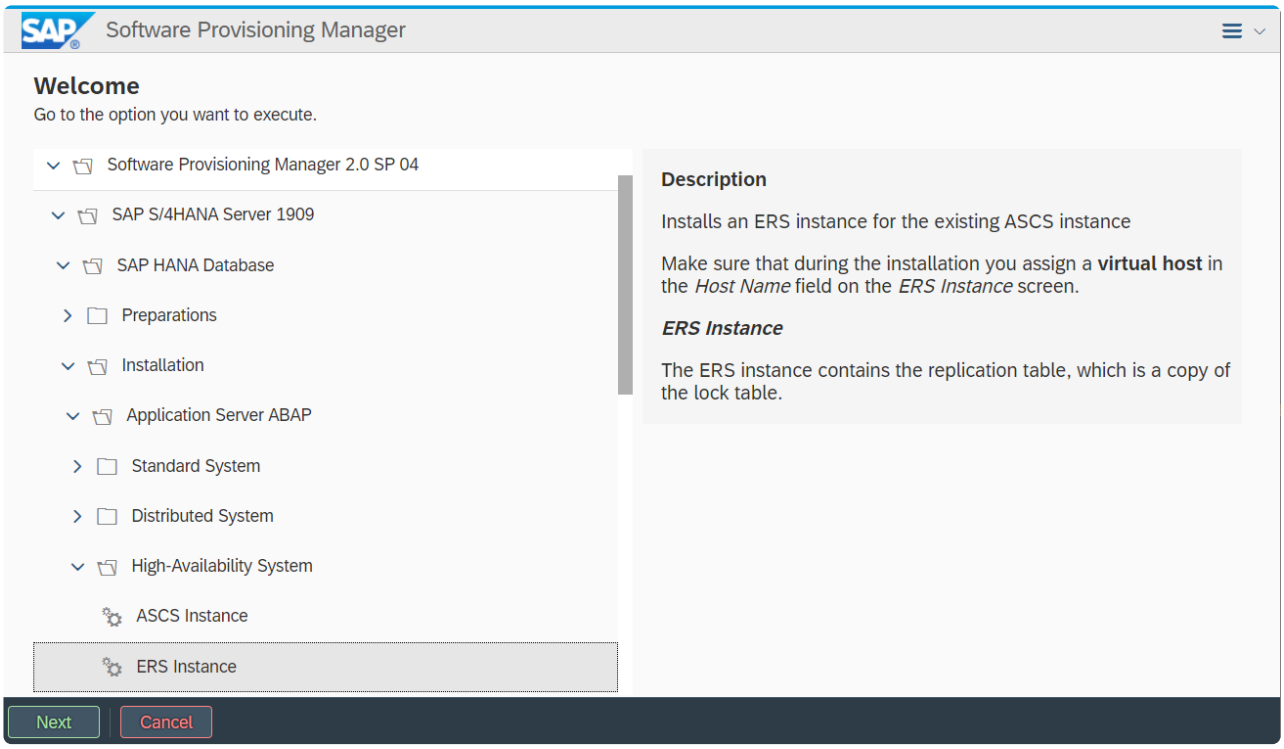
**Note:** SAPINST_USE_HOSTNAME パラメーターの値として使用される仮想ホスト名は、セクション ASCS および ERS の仮想 IP を作成する で作成および保護された ERS インスタンスの仮想 IP に解決される必要があります。

**Note:** SAPINST_REMOTE_ACCESS_USER= パラメーターを使用して、root 以外のユーザが sapinst グラフィカル インターフェイスにログインできるようにすることができます。

**Note:** dt_socket トランスポートライブラリがないために sapinst がエラーを表示する場合は、yum install -y libnsl を使用して libnsl パッケージをインストールしてください。

3. sapinst グラフィカルインターフェースにアクセスするには、Web ブラウザから https://:4237/sapinst/docs/index.html に移動します。

4. **Software Provisioning Manager 2.0 SP XX → SAP S/4HANA Server 1909 → SAP HANA Database → Installation → Application Server ABAP → High-Availability System → ERS Instance** の順に選択し、*Next* をクリックします。




5. ERS インスタンスをインストールする際に、次のパラメーターを指定します。 ✅ アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| Profile Directory | /sapmnt/SPS/profile ✅ |
| Password of SAP System Administrator (spsadm) | <spsadm User Password> |
| Specify path to SAPHOSTAGENT.SAR | /sap-install |
| Password of SAP System Administrator (sapadm) | <spsadm User Password> |
| User ID | 空欄のままにします ✅ |
| Group ID of sapsys | デフォルト値を使用 ✅ |
| Name of the ASCS Instance to be Replicated | ASCS10 ✅ |
| Number of the ASCS Instance to be Replicated | 10 ✅ |
| Number of the ERS Instance | 20 ✅ |

| | |
|---|---|
| ERS Instance Host | sps-ers |
| Yes, clean up operating system users | Click |

6. パラメーターを確認し、インストールを続行します。

> **Note:** SAP ファイルシステムの権限が不十分なためにインストールに失敗した場合は、以下の「**SAP ファイルシステムの権限の修正**」セクションの説明に従って権限を修正し、インストールを再試行してください。

7. node-bで以下のコマンドを実行し、SAPのインストールファイルをクリーンアップします。

```
[root@node-b ~]# rm -rf /root/.sapinst
```

## 両方のノードで /usr/sap/sapservices を変更する

node-aにはASCS10インスタンスのみ、node-bにはERS20インスタンスのみをインストールしたため、/usr/sap/sapservicesファイルは各ノードで不完全な状態になっています。ASCS10とERS20の両方のインスタンスのエントリーが含まれるように、node-aとnode-bの両方でこのファイルを変更します。

```
# vi /usr/sap/sapservices
# cat /usr/sap/sapservices
#!/bin/sh
LD_LIBRARY_PATH=/usr/sap/SPS/ASCS10/exe:$LD_LIBRARY_PATH; export LD_LIBRARY_PA
TH; /usr/sap/SPS/ASCS10/exe/sapstartsrv pf=/usr/sap/SPS/SYS/profile/SPS_ASCS1
0_sps-ascs -D -u spsadm
LD_LIBRARY_PATH=/usr/sap/SPS/ERS20/exe:$LD_LIBRARY_PATH; export LD_LIBRARY_PAT
H; /usr/sap/SPS/ERS20/exe/sapstartsrv pf=/usr/sap/SPS/SYS/profile/SPS_ERS20_sp
s-ers -D -u spsadm
```

## ASCSインスタンスを再起動してエンキューレプリケーションを有効にする

Node-bにERSインスタンスがインストールされたので、エンキューレプリケーションを有効にするためにnode-aでASCSインスタンスとそれに対応するSAP Start Serviceプロセスを再起動する必要があります。

1. SPS SAPインストールのデフォルトプロファイル（/usr/sap/SPS/SYS/profile/DEFAULT.PFL）に次の行が表示されることを確認します。存在しない場合は、追加します。

```
enq/replicatorhost = sps-ers
enq/replicatorinst = 20
```

2. node-aで以下のコマンドを実行してASCSインスタンスを再起動し、エンキューレプリケーションが有効になっていることを確認します。

```
[root@node-a ~]# su - spsadm -c "sapcontrol -nr 10 -function StopWait 60 2"
15.03.2021 00:29:23
Stop
OK
[root@node-a ~]# su - spsadm -c "sapcontrol -nr 10 -function RestartService SP
S"
15.03.2021 00:29:44
RestartService
OK
[root@node-a ~]# su - spsadm -c "sapcontrol -nr 10 -function StartWait 60 2"
15.03.2021 00:30:07
Start
OK
[root@node-a ~]# su - spsadm -c "sapcontrol -nr 10 -function EnqGetStatistic
| grep replication_state"
replication_state: GREEN
```

## ASCSおよびERSのインストールが成功したことを確認する

以下のコマンドを実行し、node-aでASCS10、node-bでERS20のインスタンスが正常に実行されていることを確認します。

1. 
```
[root@node-a ~]# su - spsadm -c "sapcontrol -nr 10 -function GetProcess
List"
01.01.2021 00:00:00
GetProcessList
OK
name, description, dispstatus, textstatus, starttime, elapsedtime, pid
msg_server, MessageServer, GREEN, Running, 2021 01 01 00:00:00, 0:00:0
0, 8420
enq_server, Enqueue Server 2, GREEN, Running, 2021 01 01 00:00:00, 0:0
0:00, 8428
```

2. 
```
[root@node-b ~]# su - spsadm -c "sapcontrol -nr 20 -function GetProcess
List"
01.01.2021 00:00:00
GetProcessList
OK
name, description, dispstatus, textstatus, starttime, elapsedtime, pid
enq_replicator, Enqueue Replicator 2, GREEN, Running, 2021 01 01 00:0
0:00, 0:00:00, 25263
```

# PASとAASのインスタンスをインストールする

データベース（SAP HANAなど）をインストールして高可用性の構成が完了したら、SWPM 2.0 を使用して、2つのアベイラビリティゾーンにまたがる2つの外部アプリケーションサーバーノードに、プライマリーアプリケーションサーバー（PAS）と追加のアプリケーションサーバー（AAS）インスタンスを1台ずつインストールします。
アプリケーションサーバーの冗長性は AAS インスタンスの存在によって提供されるため、アプリケーションサーバーインスタンスに LifeKeeper の保護は必要ありません。インストール前に、/sapmnt/SPS および /usr/sap/trans ファイルシステム用に作成した NFS 共有を PAS および AAS インスタンスの両方のホストにマウントする必要があることに注意してください。PASとAASインスタンスのインストールに関する詳細については、SAPのドキュメントを参照してください。

# 11.2.7.5.5.4. LifeKeeper SAP リソースを作成する

ASCS10 と ERS20 インスタンスを保護する LifeKeeper リソースを作成するために必要な手順は、クラウドプラットフォームによって異なります。お使いのクラウドプラットフォームに対応するセクションに記載されている手順に従ってください。

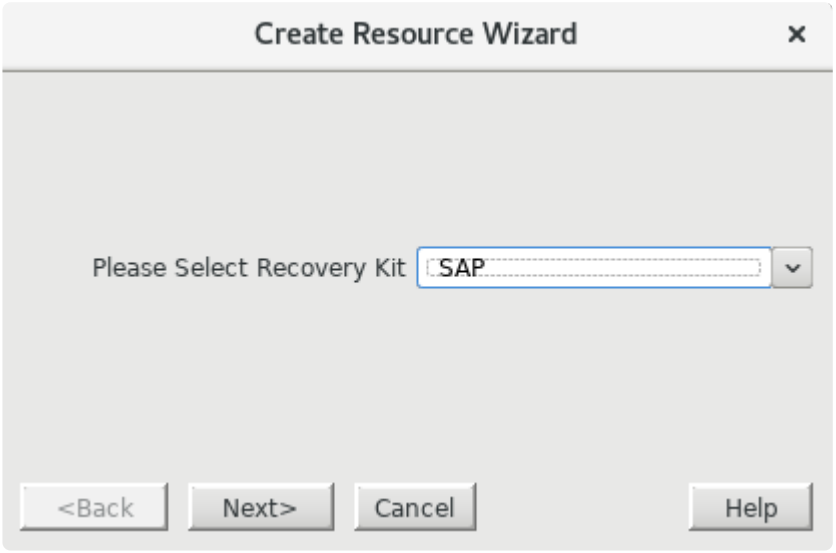
- AWS/Azure – LifeKeeper SAP リソースを作成する
- Google Cloud – LifeKeeper SAP リソースを作成する

# 11.2.7.5.5.4.1. AWS/Azure – LifeKeeper SAP リソースを作成する

**Note:** このセクションは、AWS と Microsoft Azure にデプロイメントする場合の手順です。Google Cloud へのデプロイについては、Google Cloud – LifeKeeper SAPリソースを作成する で説明している手順に従ってください。

## ASCS リソース階層を作成する

1. LifeKeeper GUI で アイコンをクリックし、 **Create Resource Wizard** を開きます。「SAP」Recovery Kitを選択します。

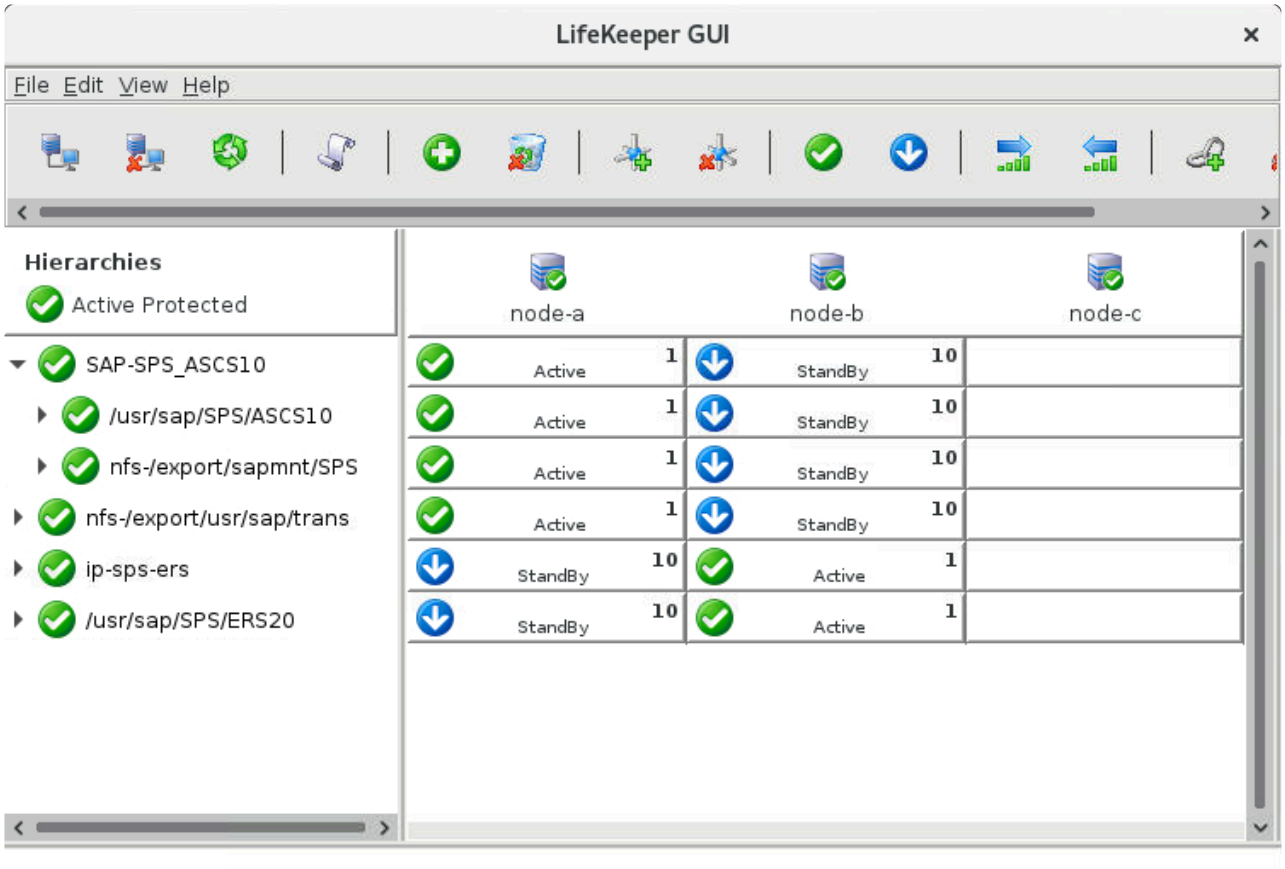



2. 次の値を入力して LifeKeeper リソース (**SAP-SPS_ASCS10**) を作成および拡張し、node-a および node-b の ASCS10 インスタンスを保護します。リソースは node-a で作成され、node-b に拡張されることに注意してください。 また、作成されたリソースは、Witness ノードである node-c に拡張してはならないことに注意してください。 アイコンは、デフォルトのオプションが選択されていることを示します。

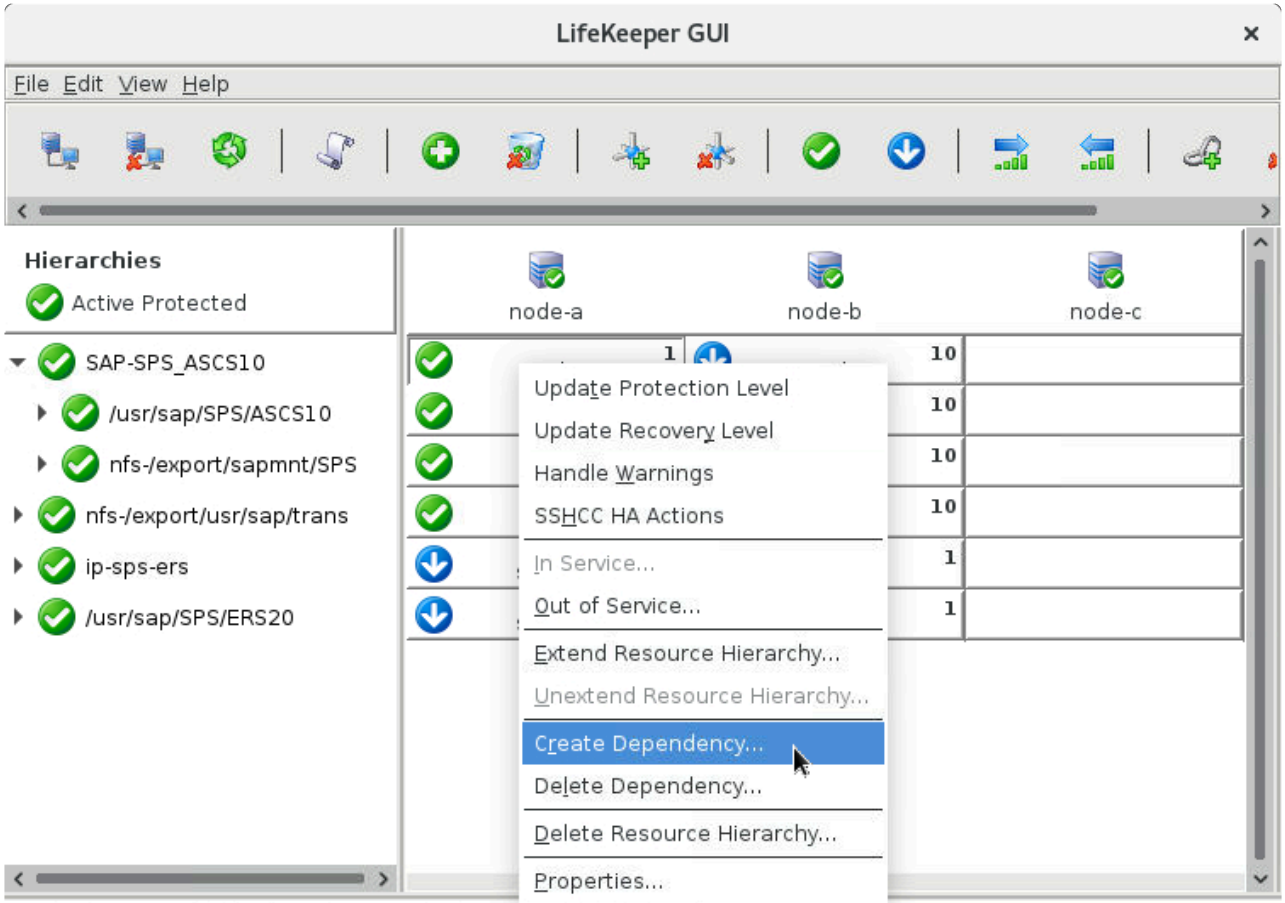
| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent |
| Server | node-a |
| SAP SID | SPS |
| SAP Instance for SPS | ASCS10 |

| | |
|---|---|
| IP child resource | ip-sps-ascs |
| Automate dependent filesystem creation | no |
| Dependent filesystem resource | /usr/sap/SPS/ASCS10,nfs-/export/sapmnt/SPS |
| SAP Tag | SAP-SPS_ASCS10 |
| **Pre-Extend Wizard** | |
| Target Server | node-b |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend gen/app Resource Hierarchy Wizard** | |
| Root Tag | SAP-SPS_ASCS10 |

**SAP-SPS_ASCS10** リソースが正常に作成され、node-b に拡張されると、LifeKeeper リソースパネルは以下のようになります。

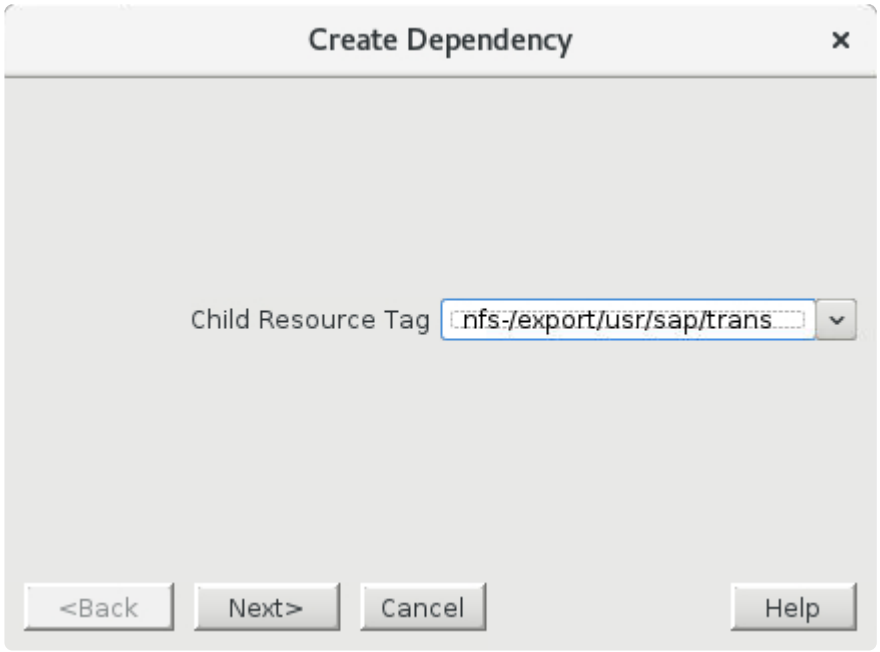



3. **SAP-SPS_ASCS10** リソースを右クリックし、ドロップダウンメニューから **[Create Dependency...]** を選択します。

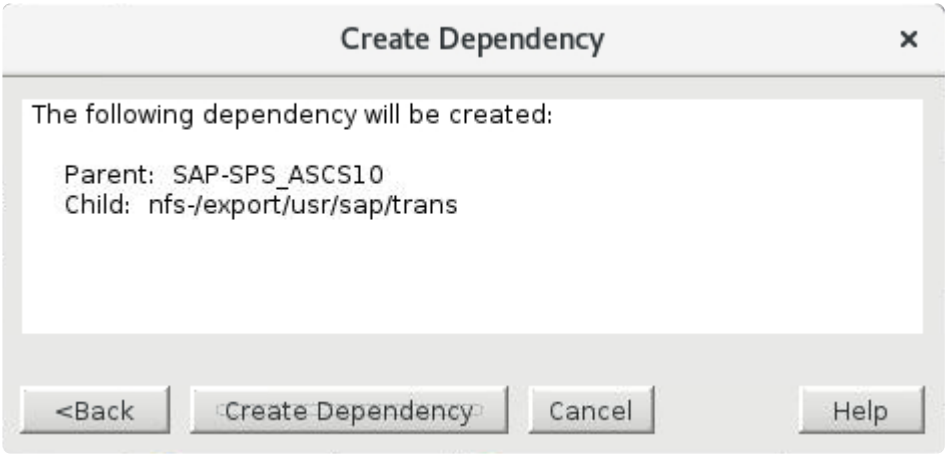


Resource Tag= SAP-SPS_ASCS10, Resource ID= SAPID-SPS-ASCS10

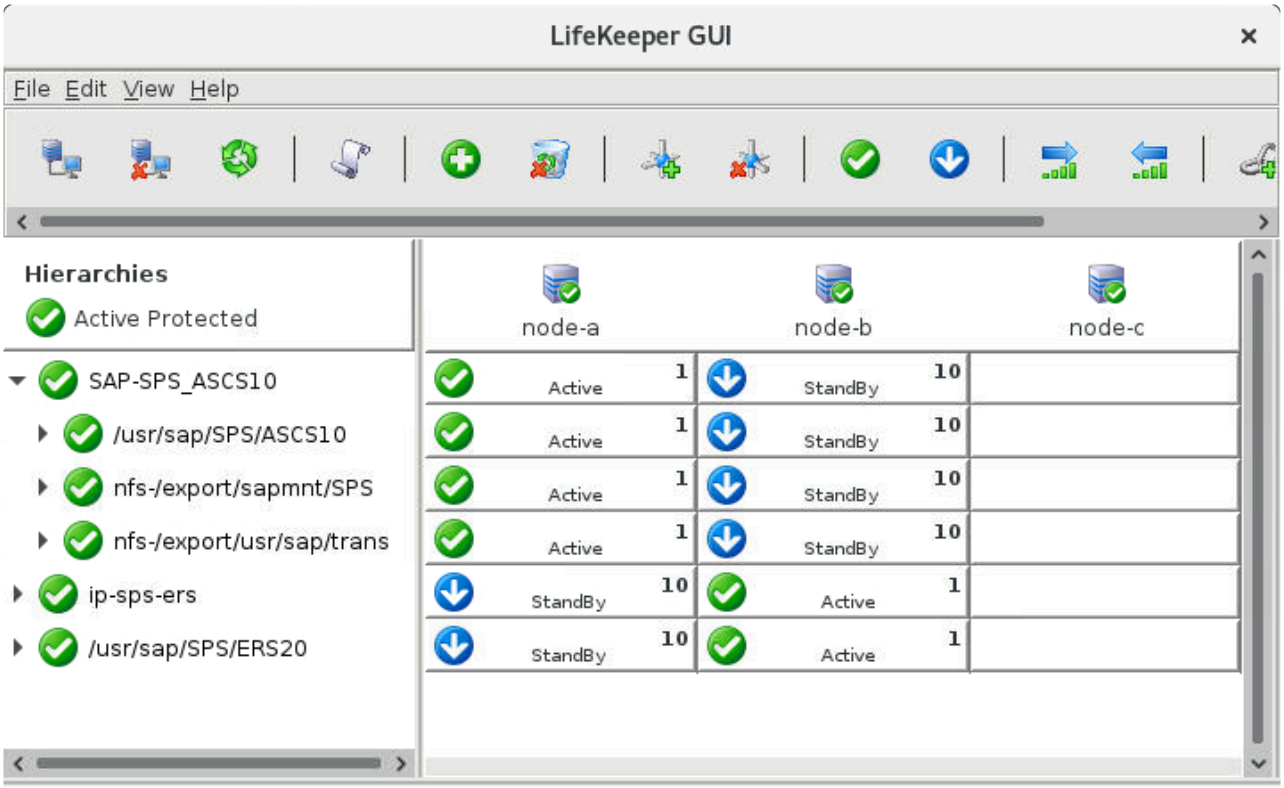
**Child Resource Tag** には、 **nfs-/export/usr/sap/trans** リソースを指定します。




**Next>** をクリックして続行し、 **[Create Dependency]** をクリックして依存関係を作成します。

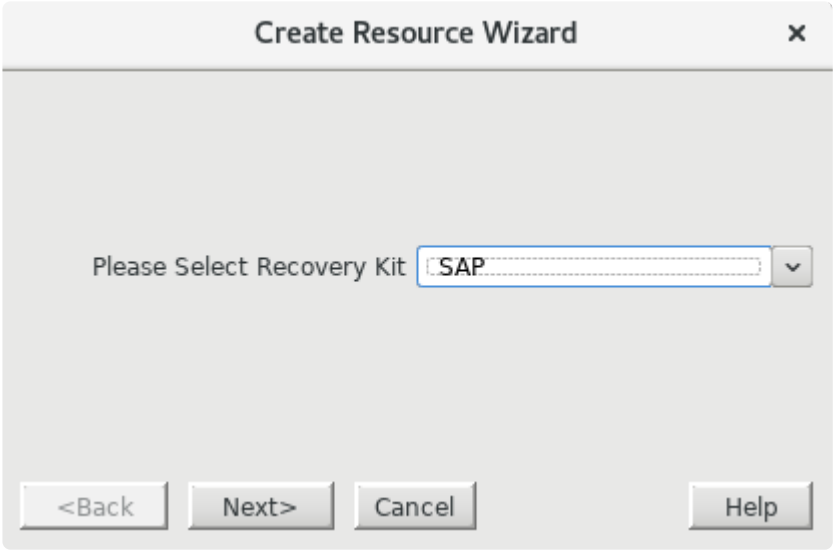



作成された階層は以下のようになります。




# ERS リソース階層を作成する

1. LifeKeeper GUI で アイコンをクリックして、 **Create Resource Wizard** を開きます。「SAP」Recovery Kit を選択します。

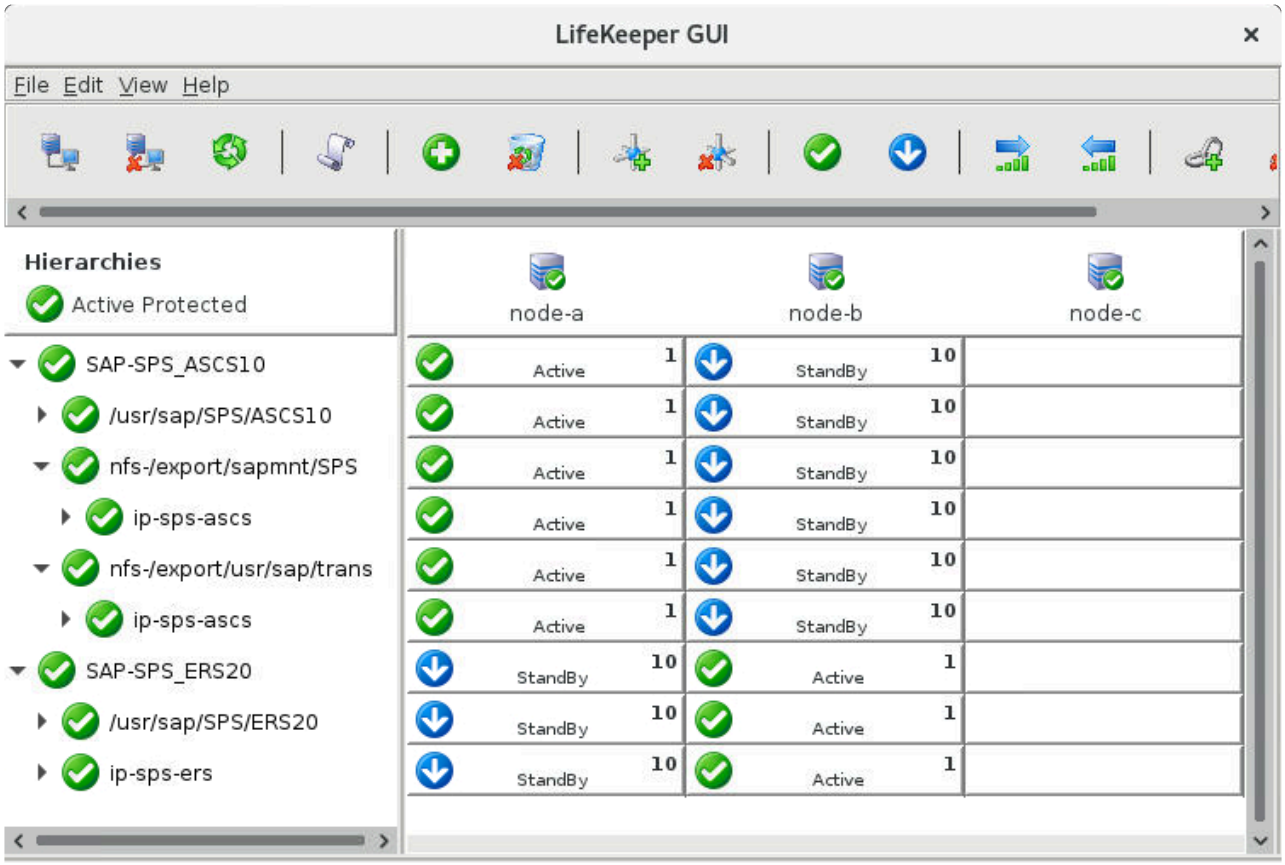



2. 以下の値を入力して LifeKeeper リソース (**SAP-SPS_ERS20**) を作成して拡張し、node-a および node-b の ERS20 インスタンスを保護します。このリソースは node-b で作成され、node-a に拡張されることに注意してください。また、作成されたリソースは、Witness ノードである node-c に拡張してはならないことに注意してください。 アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent |
| Server | node-b |
| SAP SID | SPS |
| SAP Instance for SPS | ERS20 |
| IP child resource | ip-sps-ers |
| Dependent filesystem resource | /usr/sap/SPS/ERS20 |
| SAP Tag | SAP-SPS_ERS20 |
| **Pre-Extend Wizard** | |
| Target Server | node-a |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend gen/app Resource Hierarchy Wizard** | |

| Root Tag | SAP-SPS_ERS20 |
|---|---|

**SAP-SPS_ERS20** リソースが正常に作成され、node-a に拡張されると、LifeKeeper リソースパネルは以下のようになります。



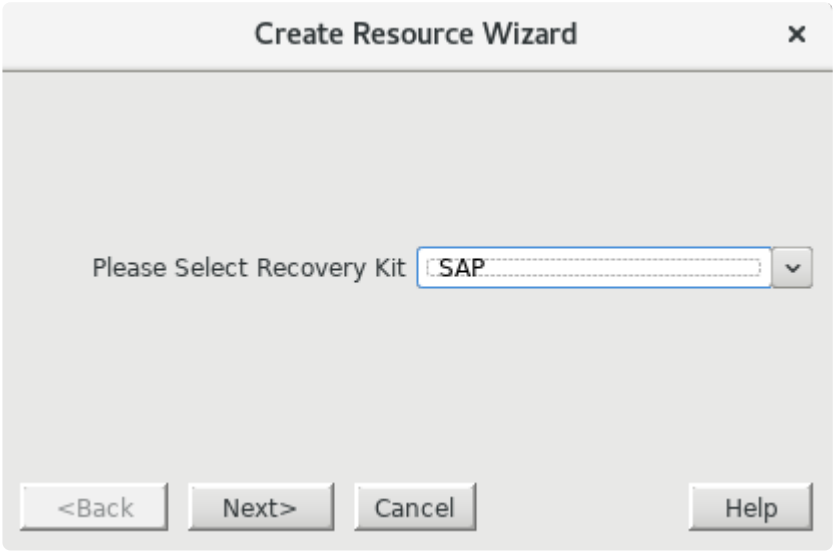

これでASCS10とERS20のインスタンスがLifeKeeperによって正常に保護されました。

# 11.2.7.5.5.4.2. Google Cloud – LifeKeeper SAP リソースを作成する

**Note:** このセクションは、Google Cloud 上へのデプロイメントの場合の手順を説明しています。AWS または Microsoft Azure の場合は、AWS/Azure – LifeKeeper SAP リソースを作成する に記載されている手順に従ってください。

## ASCS リソース階層を作成する

1. LifeKeeper GUI で アイコンをクリックして、*Create Resource Wizard* を開きます。「SAP」Recovery Kit を選択します。

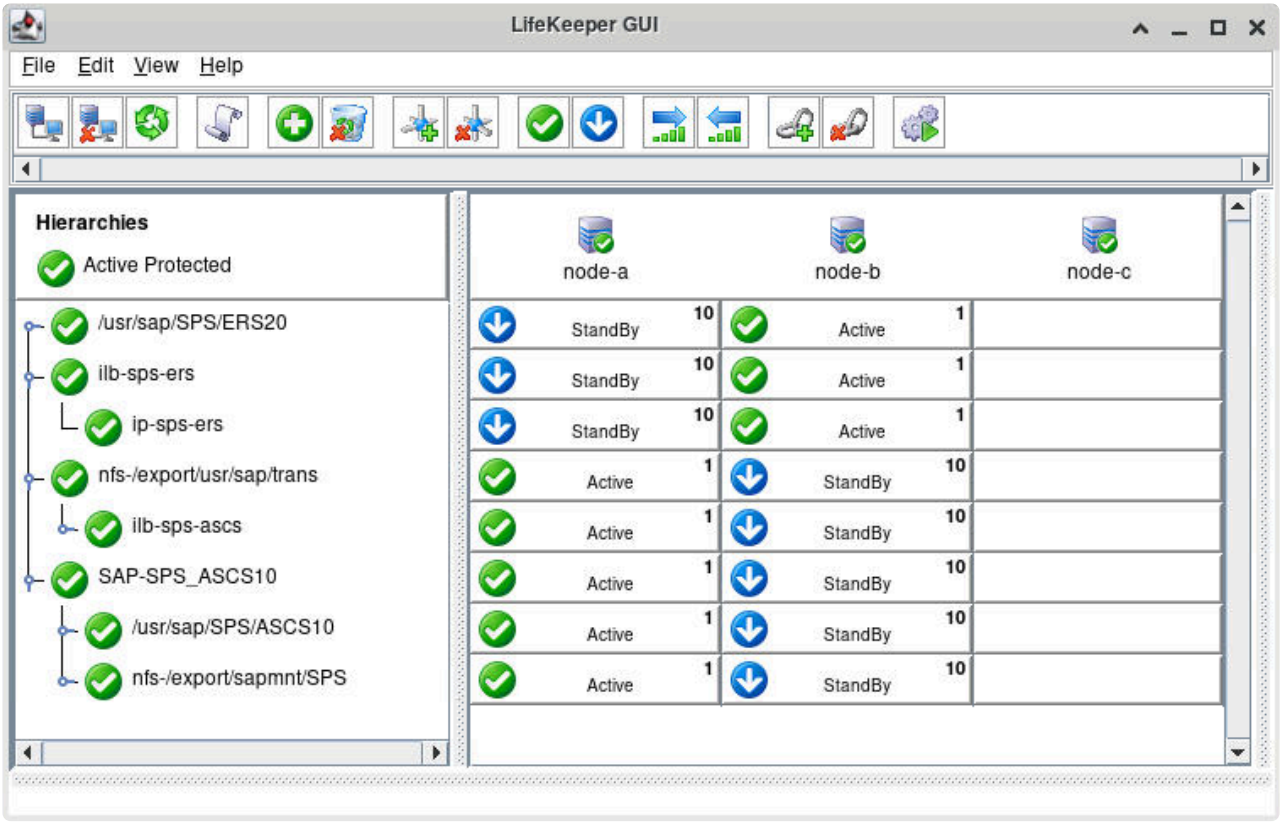



2. 次の値を入力して LifeKeeper リソース (**SAP-SPS_ASCS10**) を作成および拡張し、node-a および node-b の ASCS10 インスタンスを保護します。リソースは node-a で作成され、node-b に拡張されることに注意してください。 また、作成されたリソースは、Witness ノードである node-c に拡張してはならないことに注意してください。 アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent |
| Server | node-a |
| SAP SID | SPS |
| SAP Instance for SPS | ASCS10 |

| | |
|---|---|
| IP child resource | none |
| Automate dependent filesystem creation | no |
| Dependent filesystem resource | /usr/sap/SPS/ASCS10,nfs-/export/sapmnt/SPS |
| SAP Tag | SAP-SPS_ASCS10 |
| **Pre-Extend Wizard** | |
| Target Server | node-b |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend gen/app Resource Hierarchy Wizard** | |
| Root Tag | SAP-SPS_ASCS10 |

**SAP-SPS_ASCS10** リソースが正常に作成され、node-b に拡張されると、LifeKeeper リソースパネルは以下のようになります。

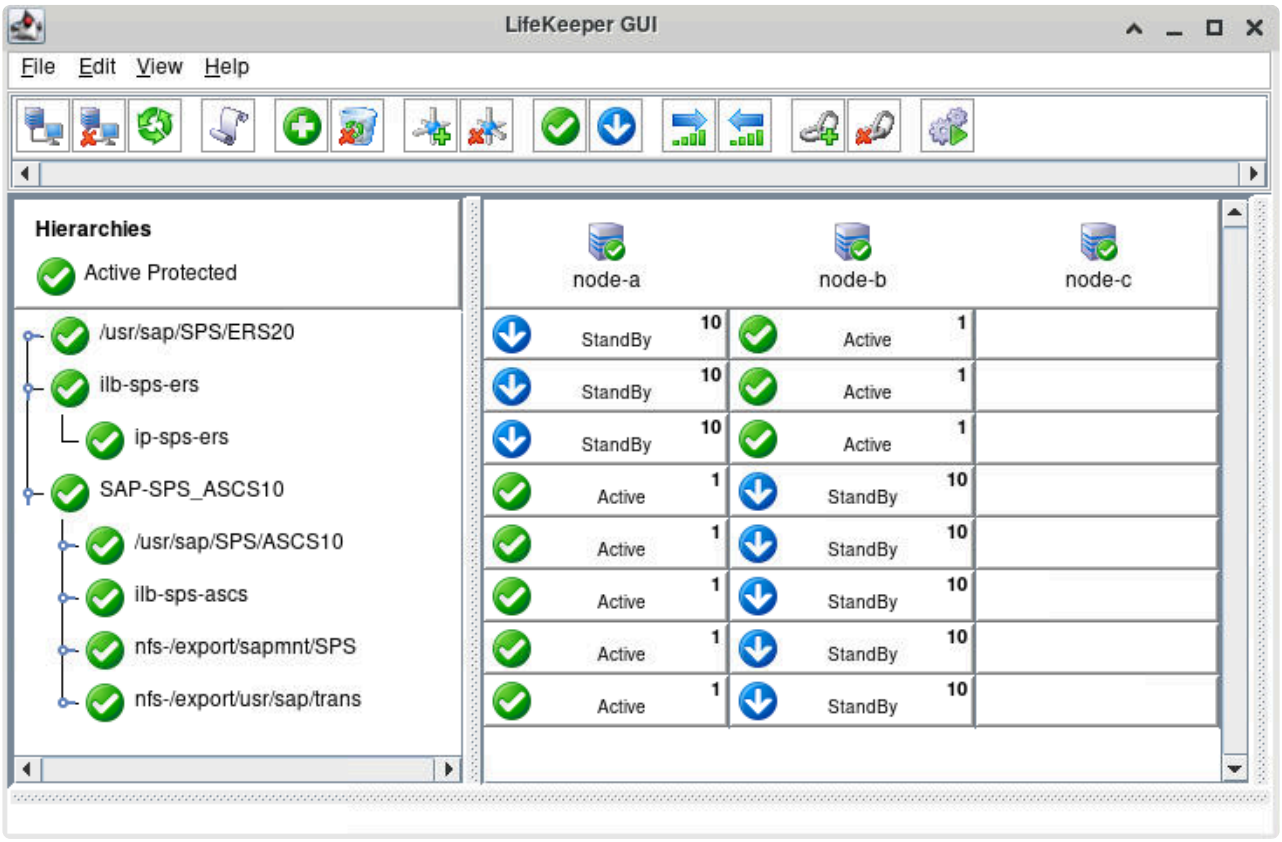



3. **SAP-SPS_ASCS10** リソースを右クリックし、ドロップダウンメニューから **[Create Dependency...]** を選択します。 **Child Resource Tag** には、*nfs-/export/usr/sap/trans* を選択します。 **Next>** をクリックして続行し、 **[Create Dependency]** をクリックして依存関係を作成しま

す。

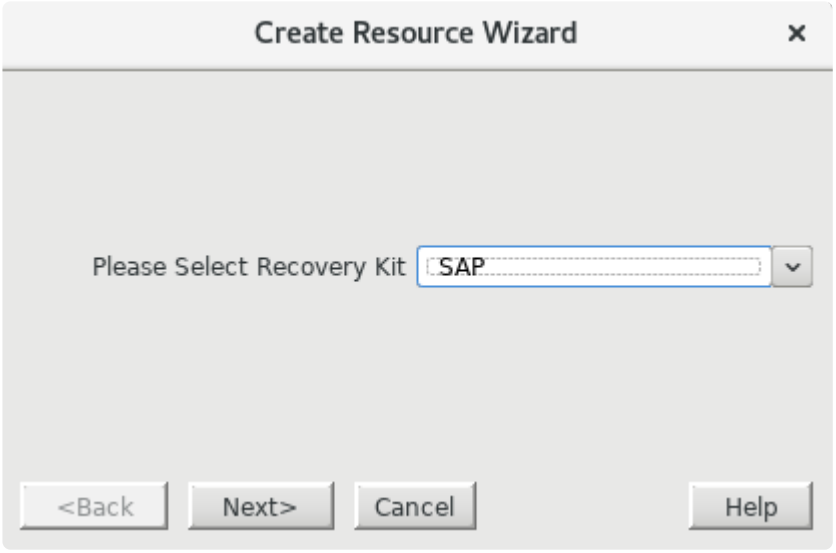
4. **SAP-SPS_ASCS10** リソースを右クリックし、ドロップダウンメニューから **[Create Dependency...]** を選択します。 **Child Resource Tag** には、 **ilb-sps-ascs** を選択します。 **Next>** をクリックして続行し、*[Create Dependency]* をクリックして依存関係を作成します。

   作成された階層は以下のようになります。




## ERS リソース階層を作成する

1. LifeKeeper GUI でアイコンをクリックして、 **Create Resource Wizard** を開きます。「SAP」Recovery Kit を選択します。

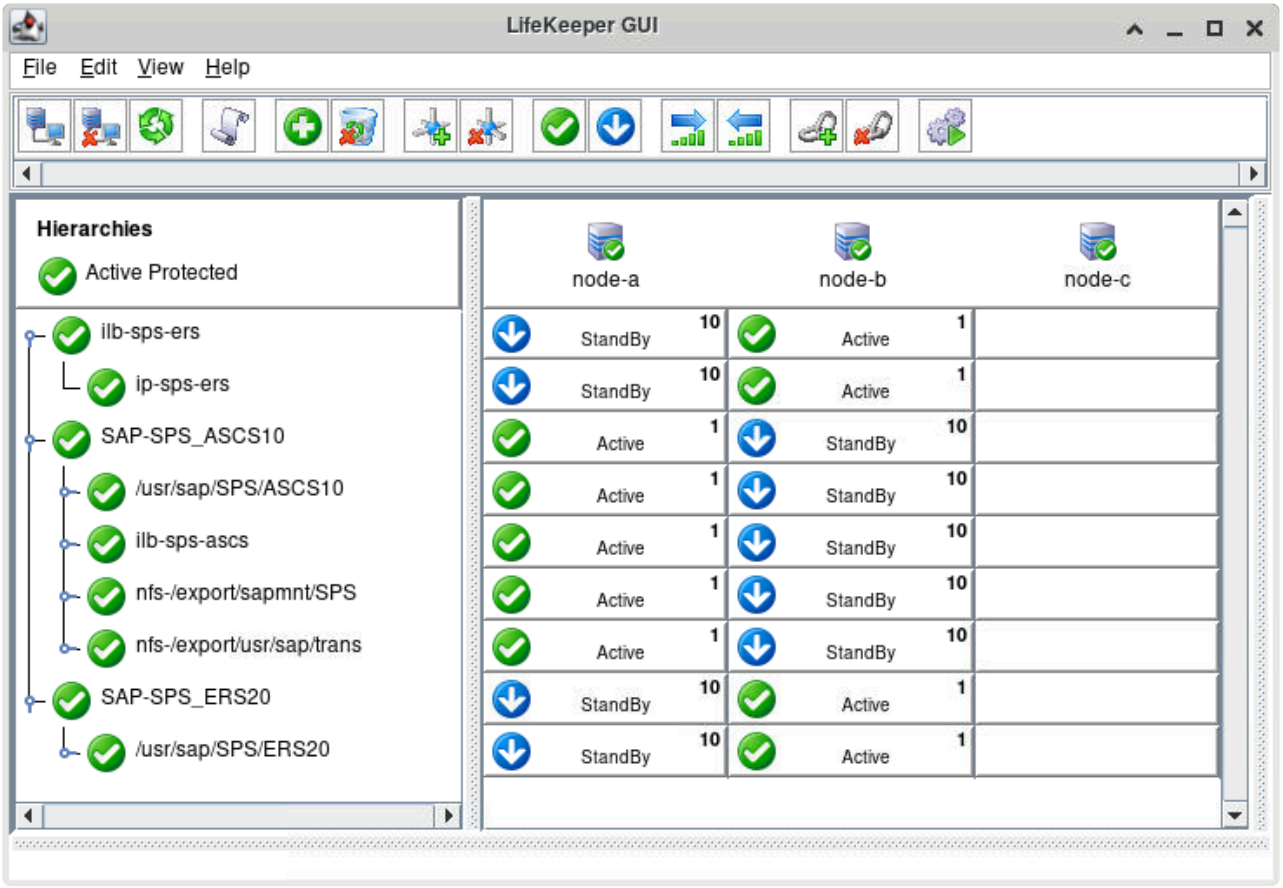



2. 以下の値を入力して LifeKeeper リソース (**SAP-SPS_ERS20**) を作成して拡張し、node-a および node-b の ERS20 インスタンスを保護します。このリソースは node-b で作成され、node-a に拡張されることに注意してください。また、作成されたリソースは、Witness ノードである node-c に拡張してはならないことに注意してください。 ✓ アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent ✓ |
| Server | node-b |
| SAP SID | SPS ✓ |
| SAP Instance for SPS | ERS20 ✓ |
| IP child resource | none ✓ |
| Automate dependent filesystem creation | no ✓ |
| Dependent filesystem resource | /usr/sap/SPS/ERS20 |
| SAP Tag | SAP-SPS_ERS20 ✓ |
| **Pre-Extend Wizard** | |
| Target Server | node-a |
| Switchback Type | intelligent ✓ |
| Template Priority | 1 ✓ |
| Target Priority | 10 ✓ |

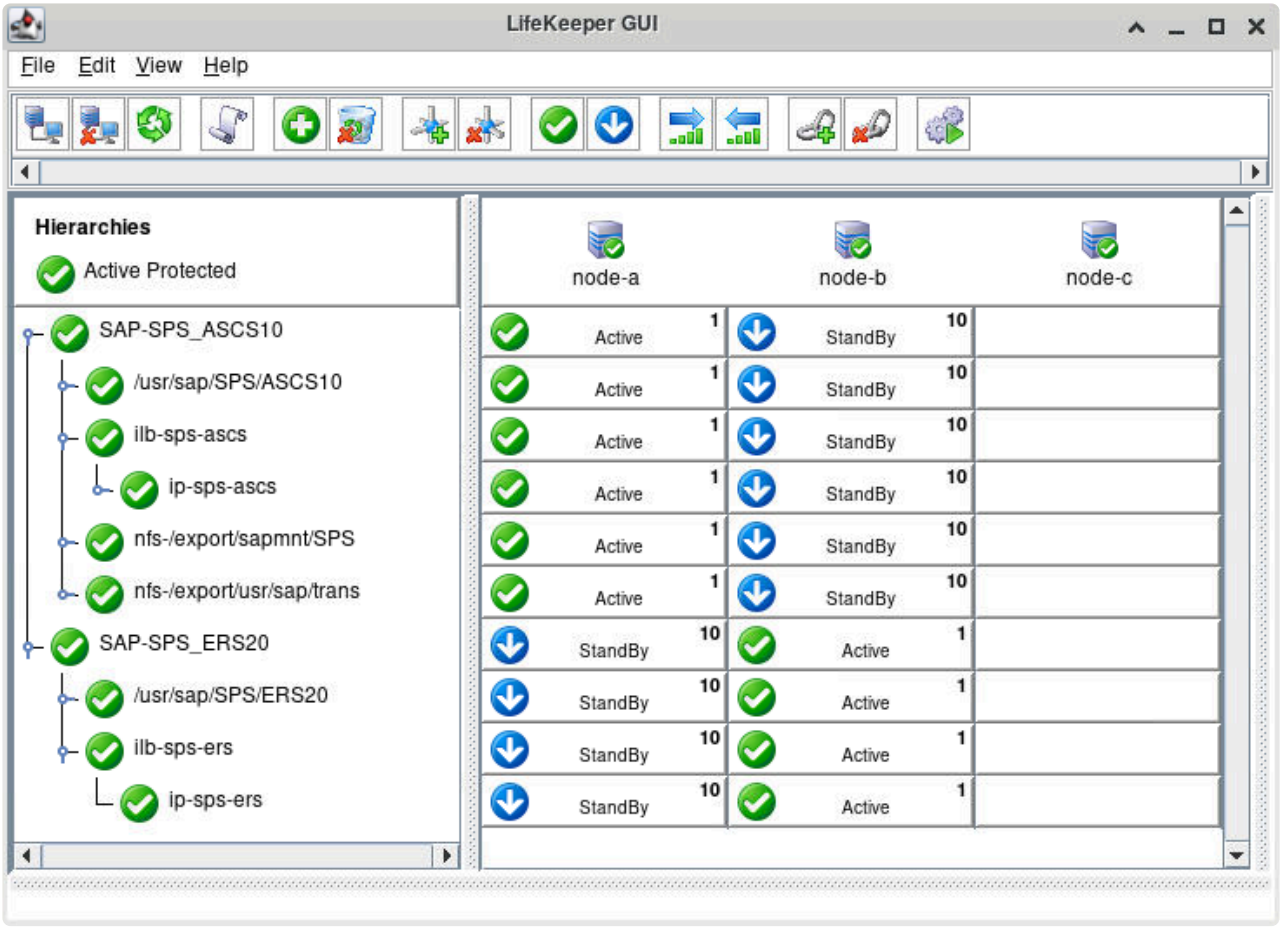
| Extend gen/app Resource Hierarchy Wizard | |
|---|---|
| Root Tag | SAP-SPS_ERS20 ✓ |

**SAP-SPS_ERS20** リソースが正常に作成され、node-a に拡張されると、LifeKeeper リソースパネルは以下のようになります。




3. **SAP-SPS_ERS20** リソースを右クリックし、ドロップダウン メニューから **[Create Dependency...]** を選択します。 **Child Resource Tag** には、 **ilb-sps-ers** を選択します。*Next>* をクリックして続行し、 **[Create Dependency]** をクリックして依存関係を作成します。

   作成された階層は以下のようになります。



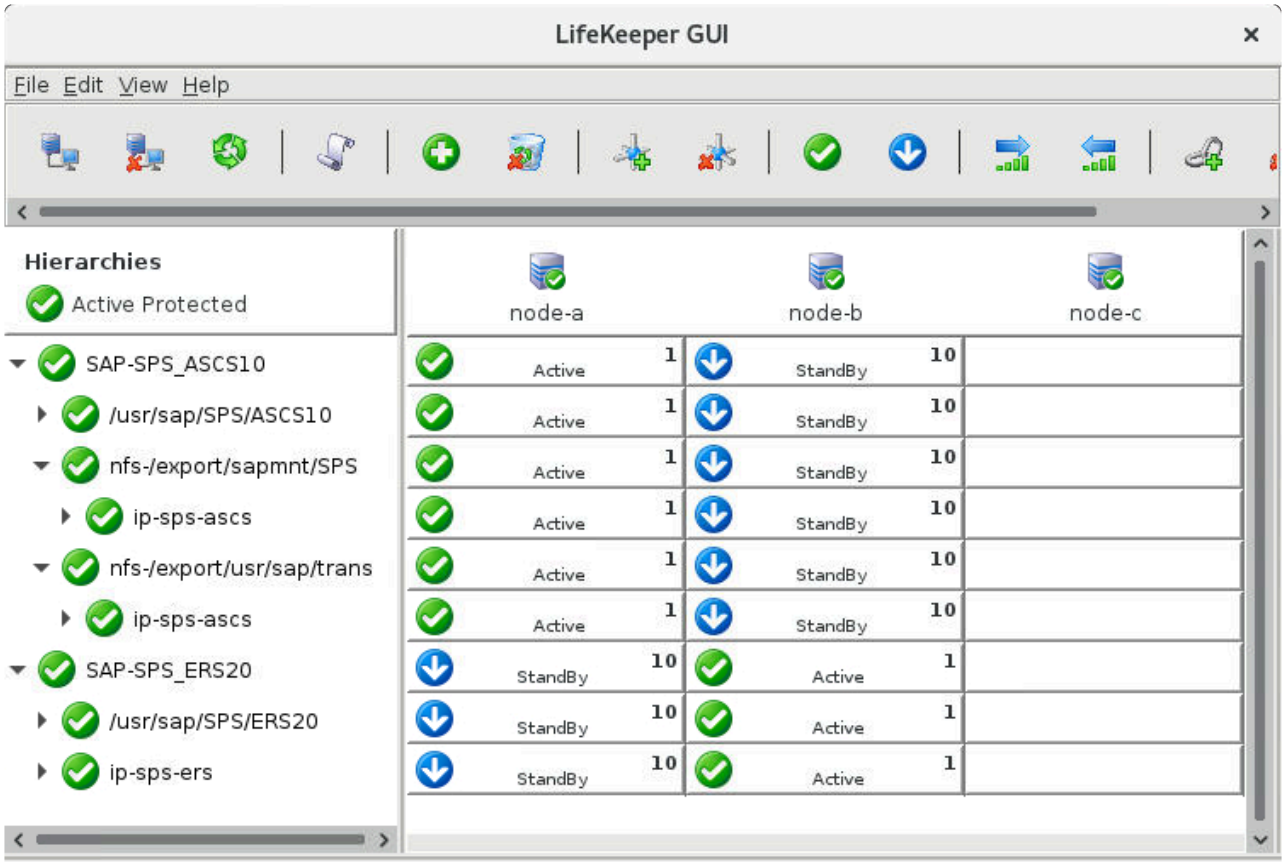

これでASCS10とERS20のインスタンスがLifeKeeperによって正常に保護されました。

# 11.2.7.5.5.5. スイッチオーバーとフェイルオーバーのテスト

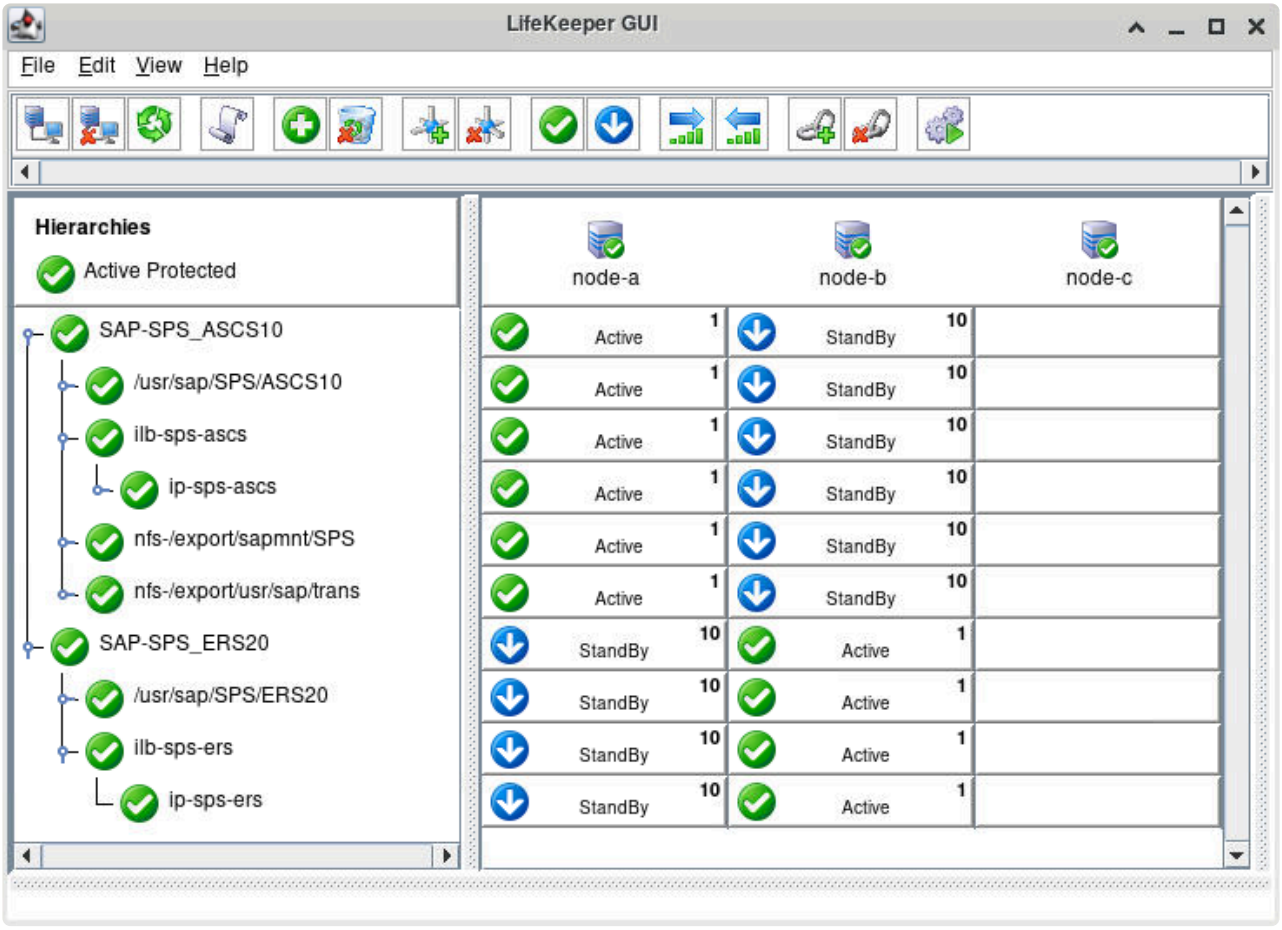
このセクションでは、 **SAP-SPS_ASCS10** と **SAP-SPS_ERS20** のリソース階層で、スイッチオーバーとフェイルオーバー時に期待される動作を検証するための基本的なテストを実行します。ASCS10 インスタンスのエンキューサーバプロセスが、スイッチオーバーまたはフェイルオーバー後に ERS20 インスタンスからエンキューロックテーブルを正常に回復できることをテストすることが重要です。

1. **SAP-SPS_ASCS10** のリソースの状態が、node-a でアクティブ、node-b で スタンバイとなっていること、および **SAP-SPS_ERS20** のリソースの状態が、node-b でアクティブ、node-a でスタンバイとなっていることを確認します。

   AWS または Azure では、LifeKeeper の GUI は以下の画像のようになります。




   Google Cloudでは、LifeKeeper の GUI は以下の画像のようになります。




2. 以下のコマンドを実行し、node-aでASCS10、node-bでERS20のインスタンスが正常に動作していることを確認します。

```
[root@node-a ~]# su - spsadm -c "sapcontrol -nr 10 -function GetProcessList"
04.03.2021 20:24:12
GetProcessList
OK
name, description, dispstatus, textstatus, starttime, elapsedtime, pid
msg_server, MessageServer, GREEN, Running, 2020 12 21 16:53:00, 1755:31:12, 11
497
enq_server, Enqueue Server 2, GREEN, Running, 2020 12 21 16:53:00, 1755:31:1
2, 11498

[root@node-b ~]# su - spsadm -c "sapcontrol -nr 20 -function GetProcessList"
04.03.2021 20:24:22
GetProcessList
OK
name, description, dispstatus, textstatus, starttime, elapsedtime, pid
enq_replicator, Enqueue Replicator 2, GREEN, Running, 2021 02 22 16:55:17, 24
3:29:05, 30028
```

3. node-a で次のコマンドを実行して、0-99のラベルが付いた100個の排他ロック (累積なし) を、エンキューサーバーが保持するロック テーブルに書き込みます。

```
[root@node-a ~]# su - spsadm -c "enq_admin --set_locks=100:X:DIAG::TAB:%u p
f=/usr/sap/SPS/SYS/profile/SPS_ASCS10_sps-ascs"
Enqueue Server 2

2021-03-04 20:32:16; OK; 'Set Locks'; Response=41496 usec
============================================================
```

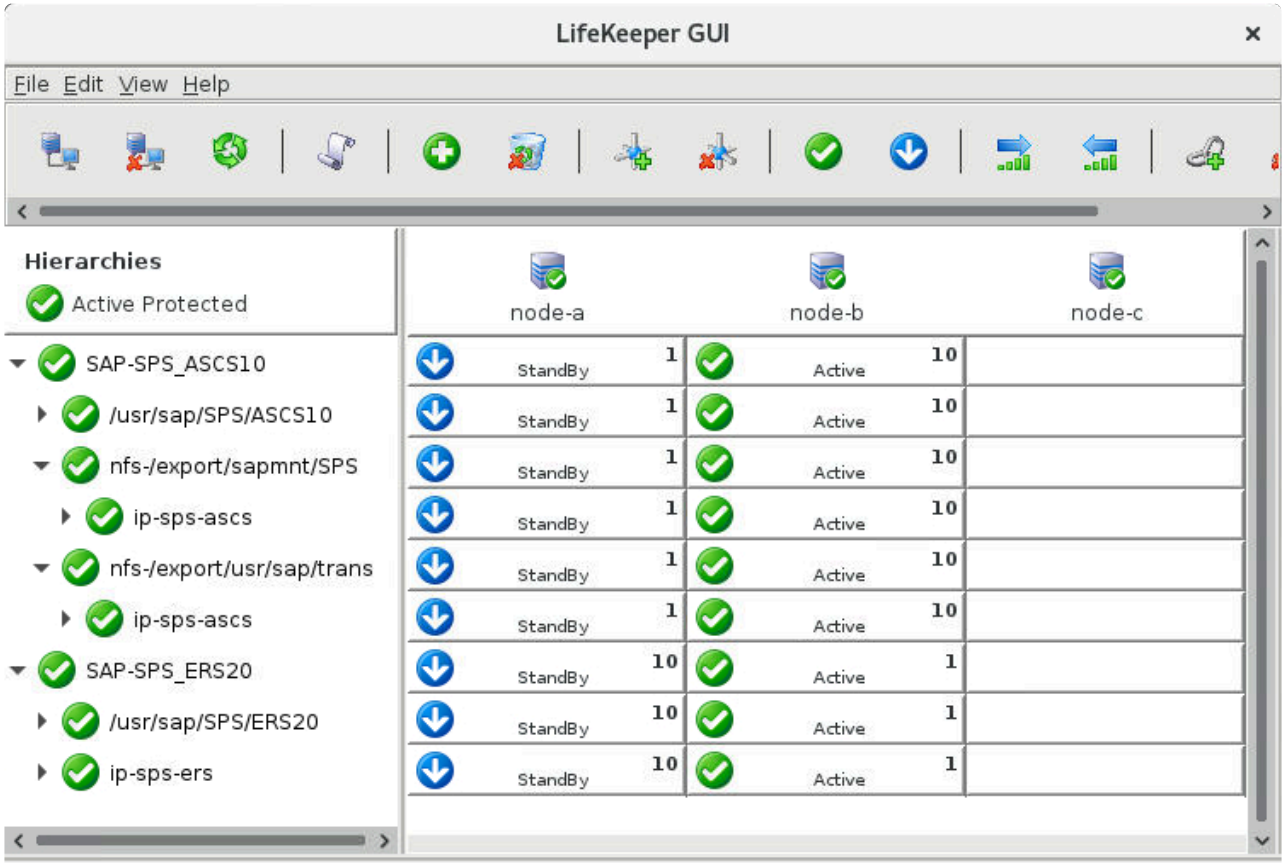
次のコマンドを実行して、node-a のロック テーブルにロックが正常に格納され、node-b のエンキューレプリケーションサーバーに複製されたことを確認します。

```
[root@node-a ~]# su - spsadm -c "sapcontrol -nr 10 -function EnqGetStatistic"
| grep locks_now
locks_now: 100

[root@node-b ~]# su - spsadm -c "sapcontrol -nr 20 -function EnqGetStatistic"
| grep locks_now
locks_now: 100
```

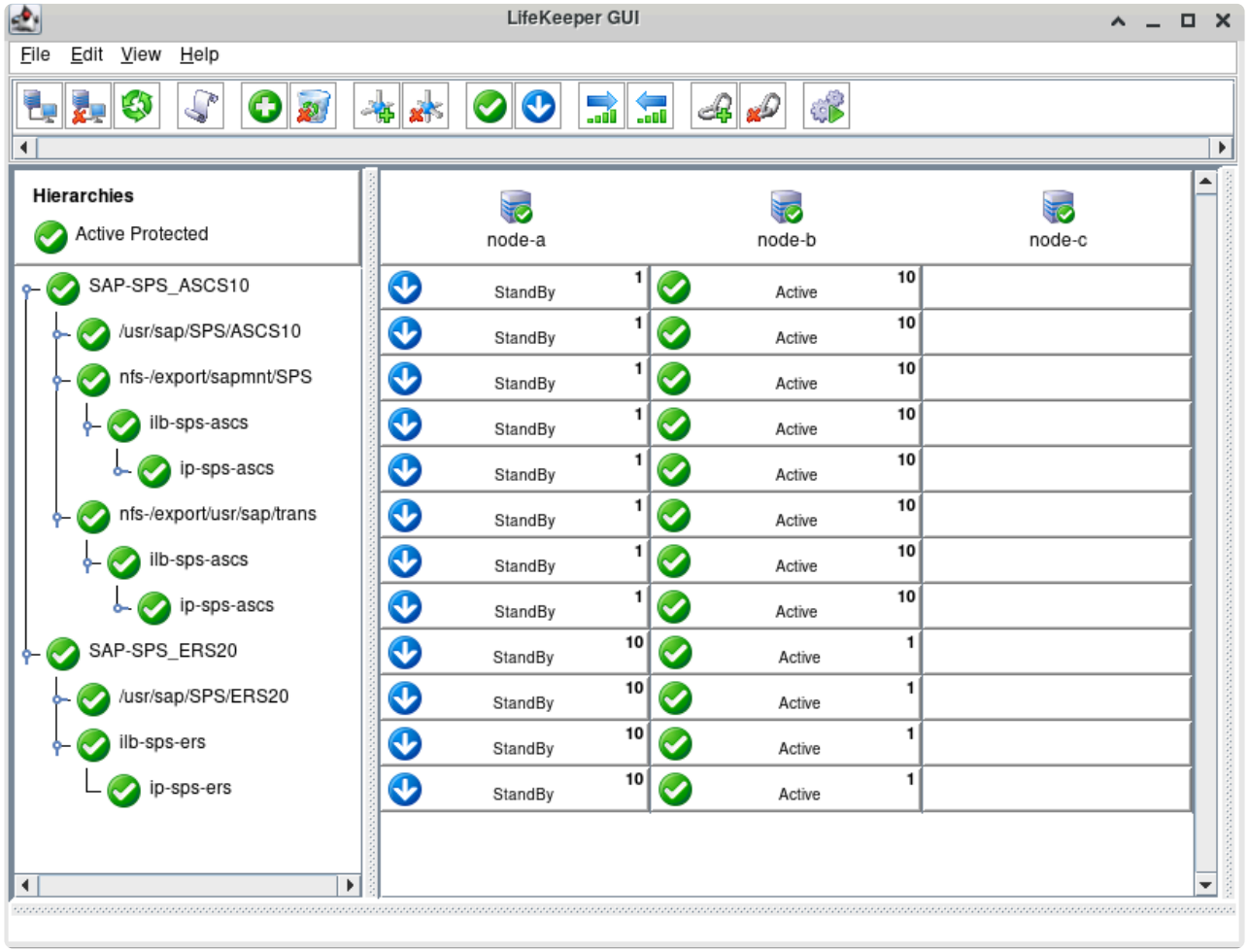
**IMPORTANT** Google Cloud のデプロイで node-b のエンキューレプリケーションサーバーにロックが正常に複製されない場合は、 Google Cloud – 内部ロードバランサーを使用する のセクション「**IP 転送を無効にする**」で説明しているように、node-a と node-b の両方で IP 転送が無効になっていることを確認してください。
この構成では、ネットワークマスク 255.255.255.255 を使用して、対応するロードバランサーのフロントエンド IP アドレスを保護する依存 IP リソースが各 GenLB リソースにあることも確認してください。Google Cloud上でこれらの設定を行わないと、エンキューサーバーとエンキューレプリケーションサーバーが異なるクラスタノードで動作している場合、対応する内部ロードバランサーのフロントエンドIPアドレスを通じて互いに通信することができなくなります。

4. node-b で **SAP-SPS_ASCS10** リソースを右クリックし、*In-Service...* 操作を選択して、ASCS リソース階層のスイッチオーバーを実行します。 **In Service** をクリックして、スイッチオーバーを開始します。スイッチオーバーが完了すると、 **SAP-SPS_ASCS10** と **SAP-SPS_ERS20** の両方のリソースがnode-b でIn-Serviceになります。

   AWS または Azure では、LifeKeeper の GUI は以下の画像のようになります。

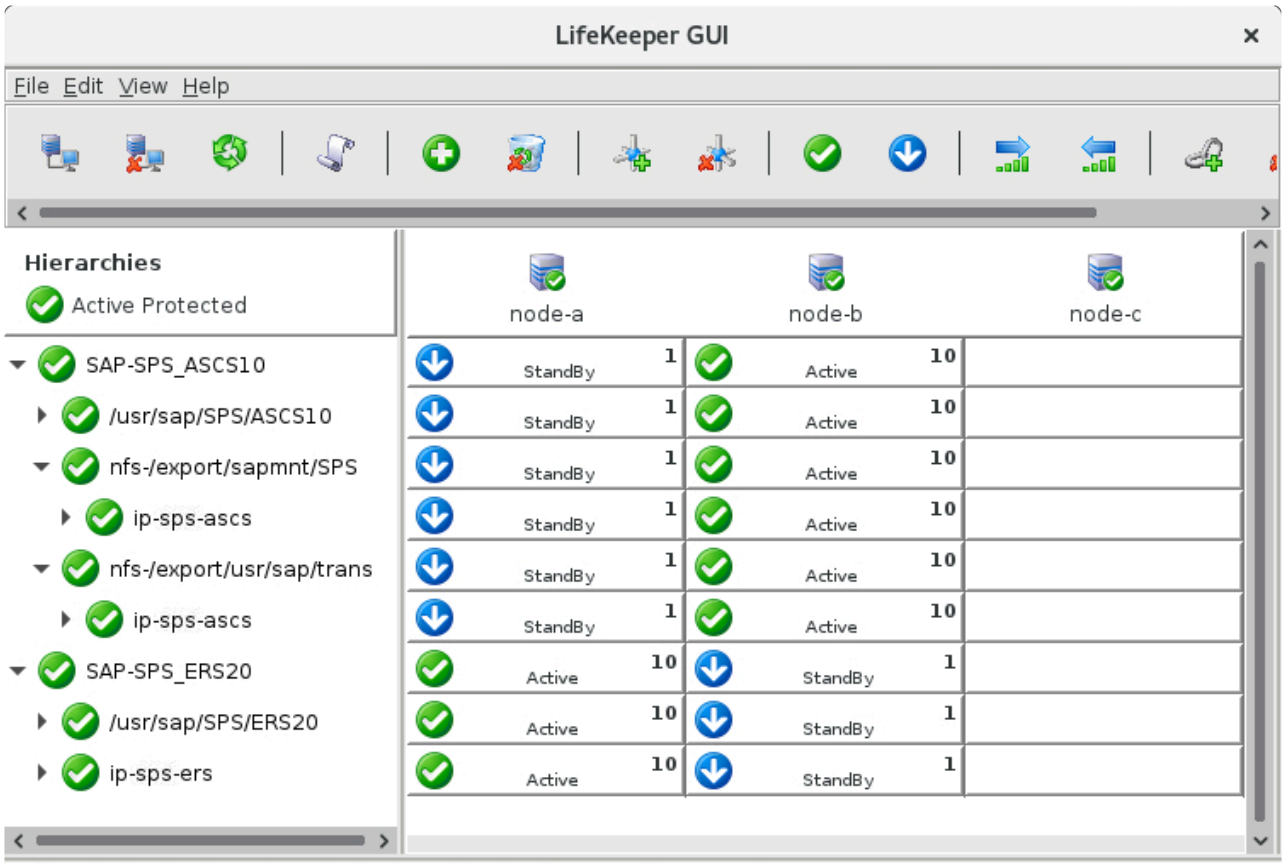



Google Cloudでは、LifeKeeper の GUI は以下の画像のようになります。

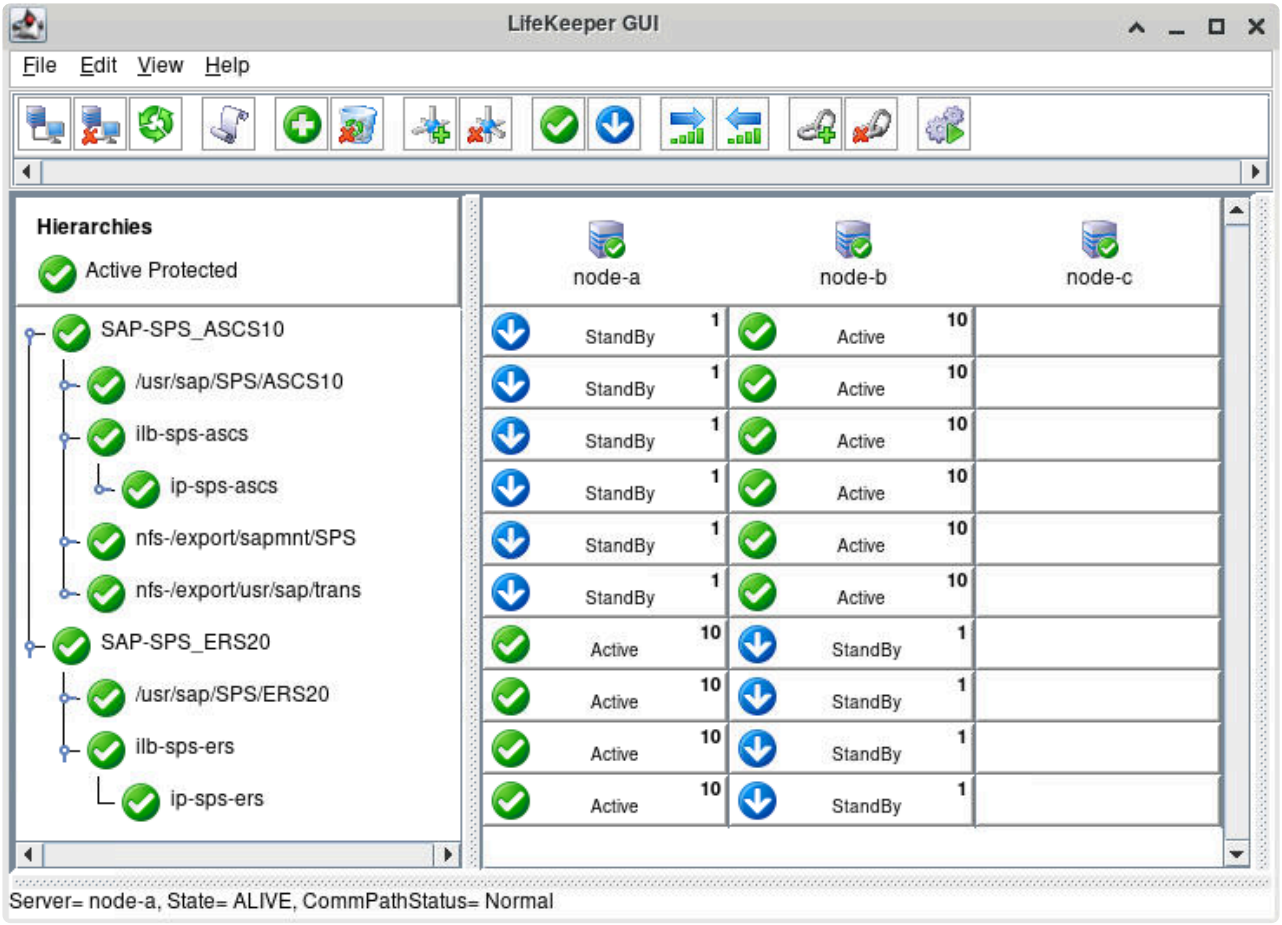


ASCS リソース階層が node-b 上で正常に In-Service になり、エンキューサーバープロセスがエンキューレプリケーションサーバープロセスからバックアップエンキューロックテーブルのコピーを取得すると、LifeKeeperは自動的に **SAP-SPS_ERS20** リソースを node-a に再配置し、クラスターノード間でロックテーブルの冗長性を提供します。このプロセスの完了には数分かかる場合があります。

AWS または Azure では、LifeKeeper の GUI は以下の画像のようになります。




Google Cloudでは、LifeKeeper の GUI は以下の画像のようになります。




5. LifeKeeper が **SAP-SPS_ERS20** リソースを node- a に再配置したら、次のコマンドを実行して、ASCS10 および ERS20 インスタンスがそれぞれ node-b および node- a で正常に実行されていること、両方ともステップ3で書き込んだ100個のロックをまだ保持していることを検証します。

```
[root@node-a ~]# su - spsadm -c "sapcontrol -nr 20 -function GetProcessList"
04.03.2021 20:58:57
GetProcessList
OK
name, description, dispstatus, textstatus, starttime, elapsedtime, pid
enq_replicator, Enqueue Replicator 2, GREEN, Running, 2021 03 04 20:57:34, 0:0
1:23, 21967

[root@node-a ~]# su - spsadm -c "sapcontrol -nr 20 -function EnqGetStatistic"
| grep locks_now
locks_now: 100

[root@node-b ~]# su - spsadm -c "sapcontrol -nr 10 -function GetProcessList"
04.03.2021 20:56:56
GetProcessList
OK
name, description, dispstatus, textstatus, starttime, elapsedtime, pid
msg_server, MessageServer, GREEN, Running, 2021 03 04 20:54:47, 0:02:09, 17074
enq_server, Enqueue Server 2, GREEN, Running, 2021 03 04 20:54:47, 0:02:09, 17
075
```

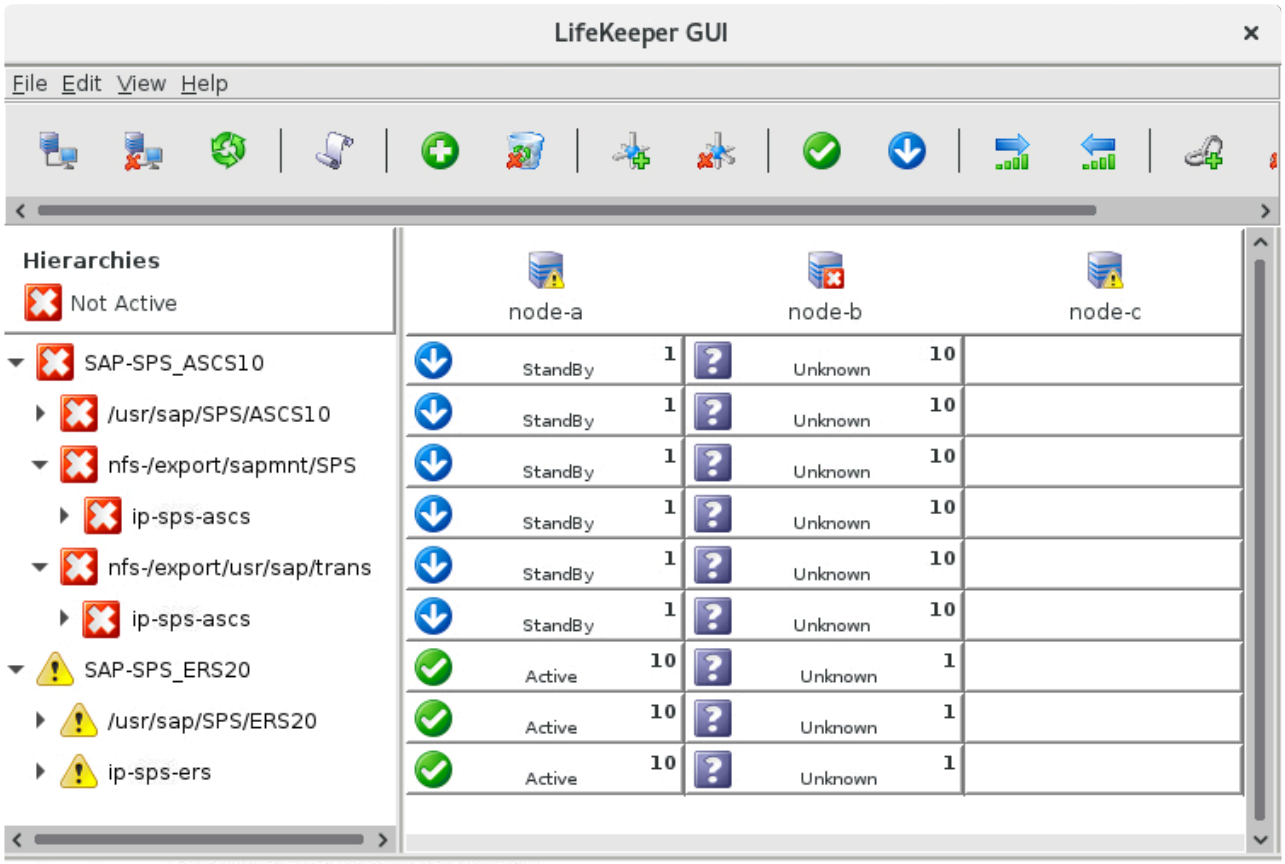
```
[root@node-b ~]# su - spsadm -c "sapcontrol -nr 10 -function EnqGetStatistic"
| grep locks_now
locks_now: 100
```

6. 次のコマンドを実行して、node-b を強制的に再起動します。

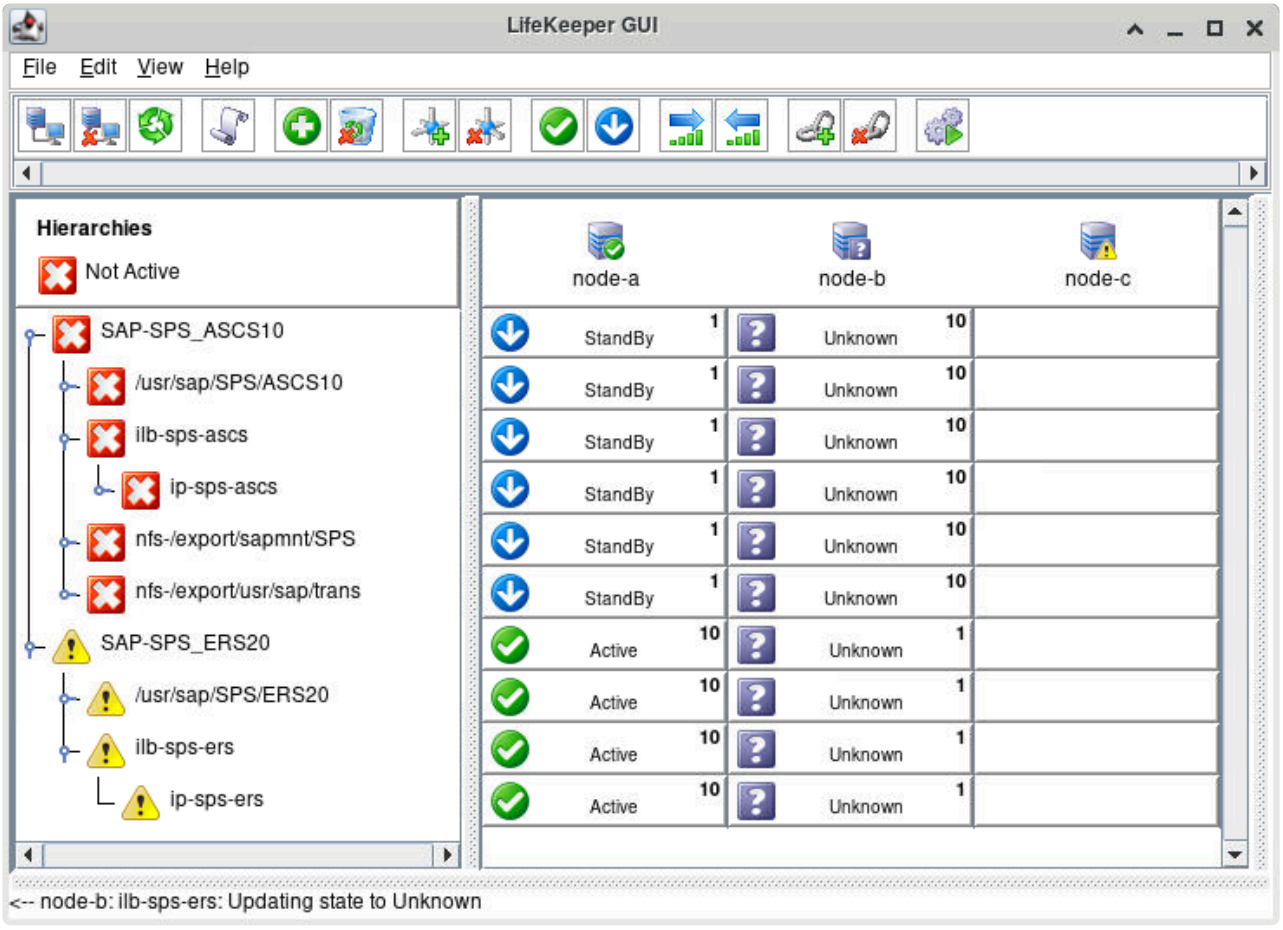
```
[root@node-b ~]# echo b > /proc/sysrq-trigger
```

LifeKeeper が node-b の電源がオフになっていることを検出すると、LifeKeeper GUI で node-b のステータスが「Unknown」に更新されます。

AWS または Azure では、LifeKeeper の GUI は以下の画像のようになります。

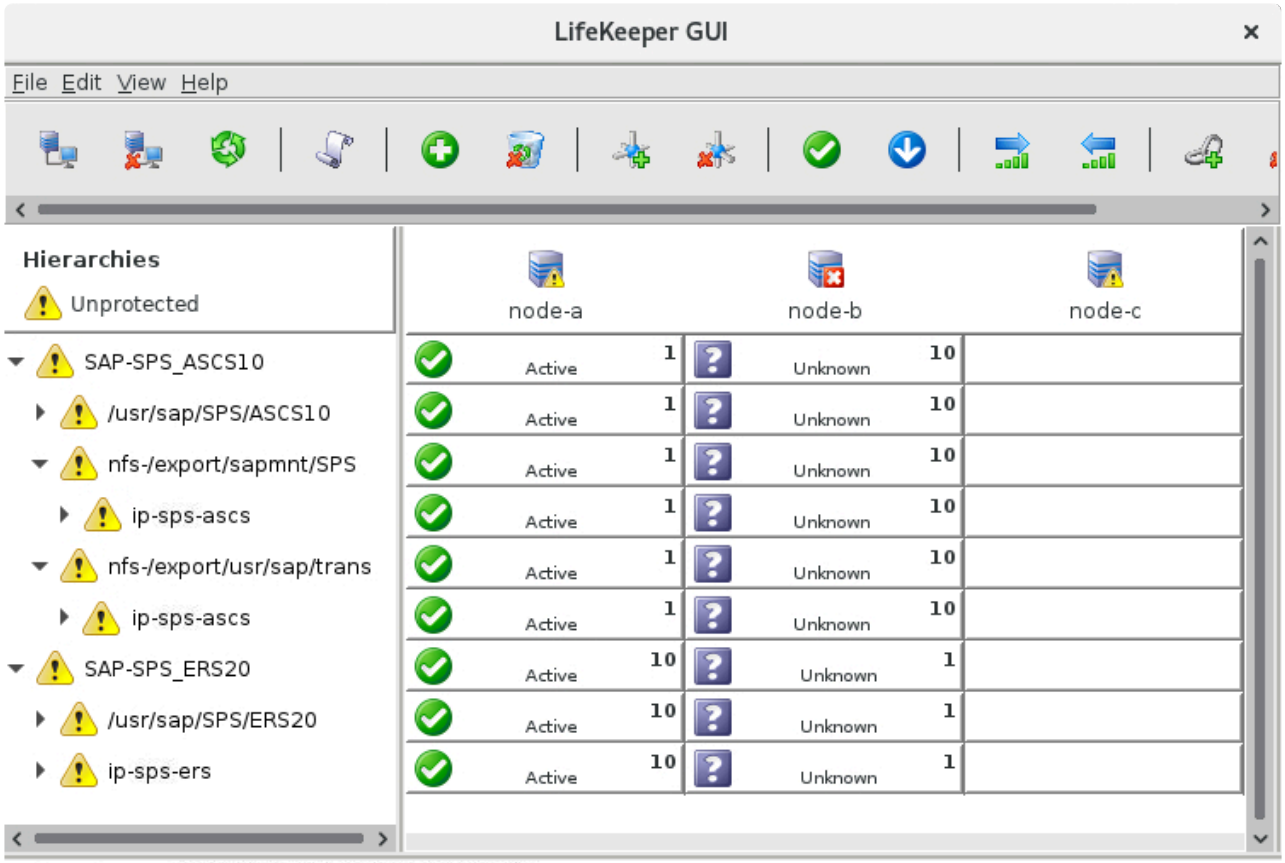



Google Cloudでは、LifeKeeper の GUI は以下の画像のようになります。

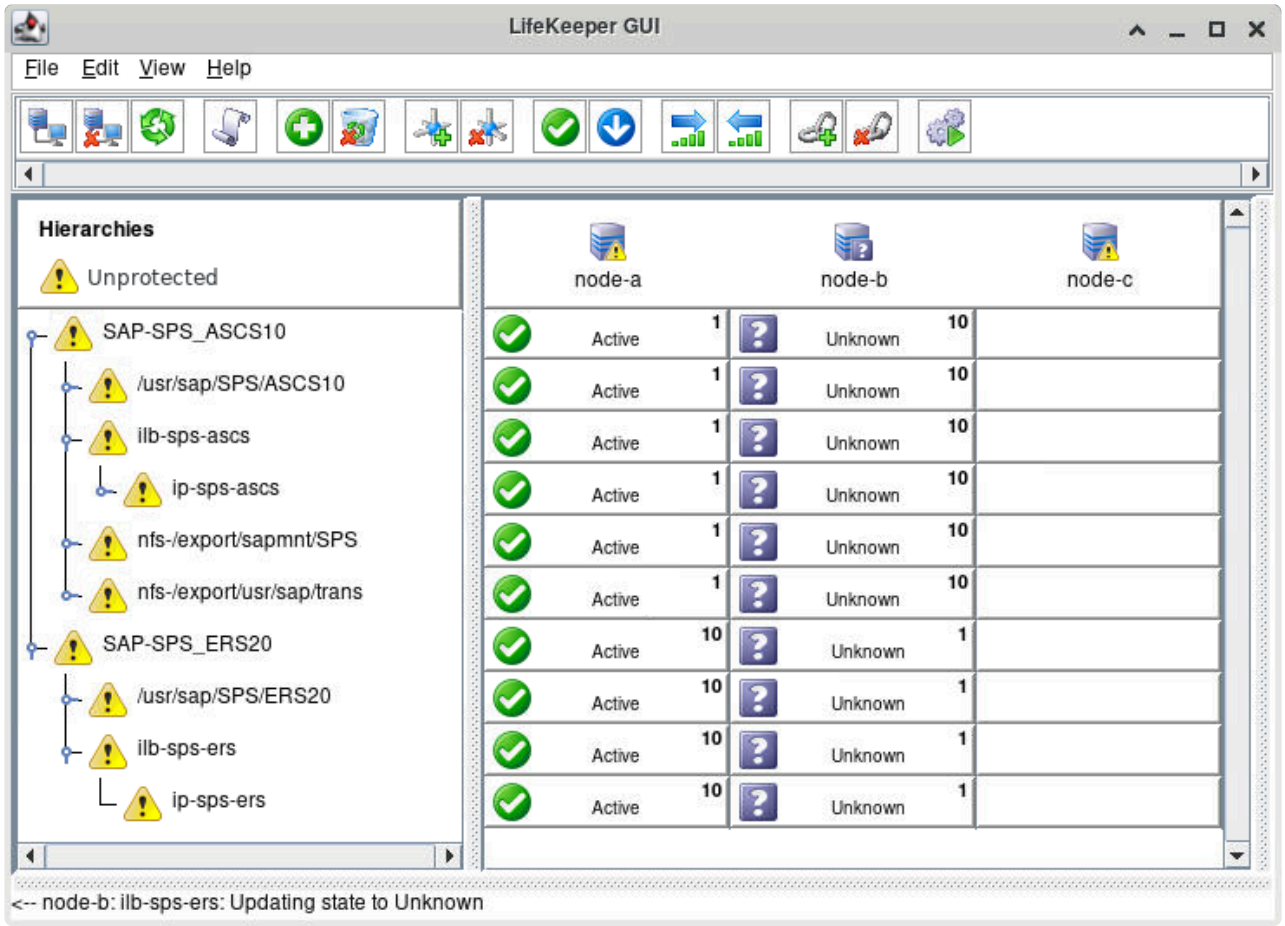



この時点で、LifeKeeper は **SAP-SPS_ASCS10** リソース階層の node-a への自動フェイルオーバーを開始します。 **SAP-SPS_ASCS10** と **SAP-SPS_ERS20** のリソース階層は、node-b がオンラインに戻るまで、両方とも node-a で Active になります。

AWS または Azure では、LifeKeeper の GUI は以下の画像のようになります。

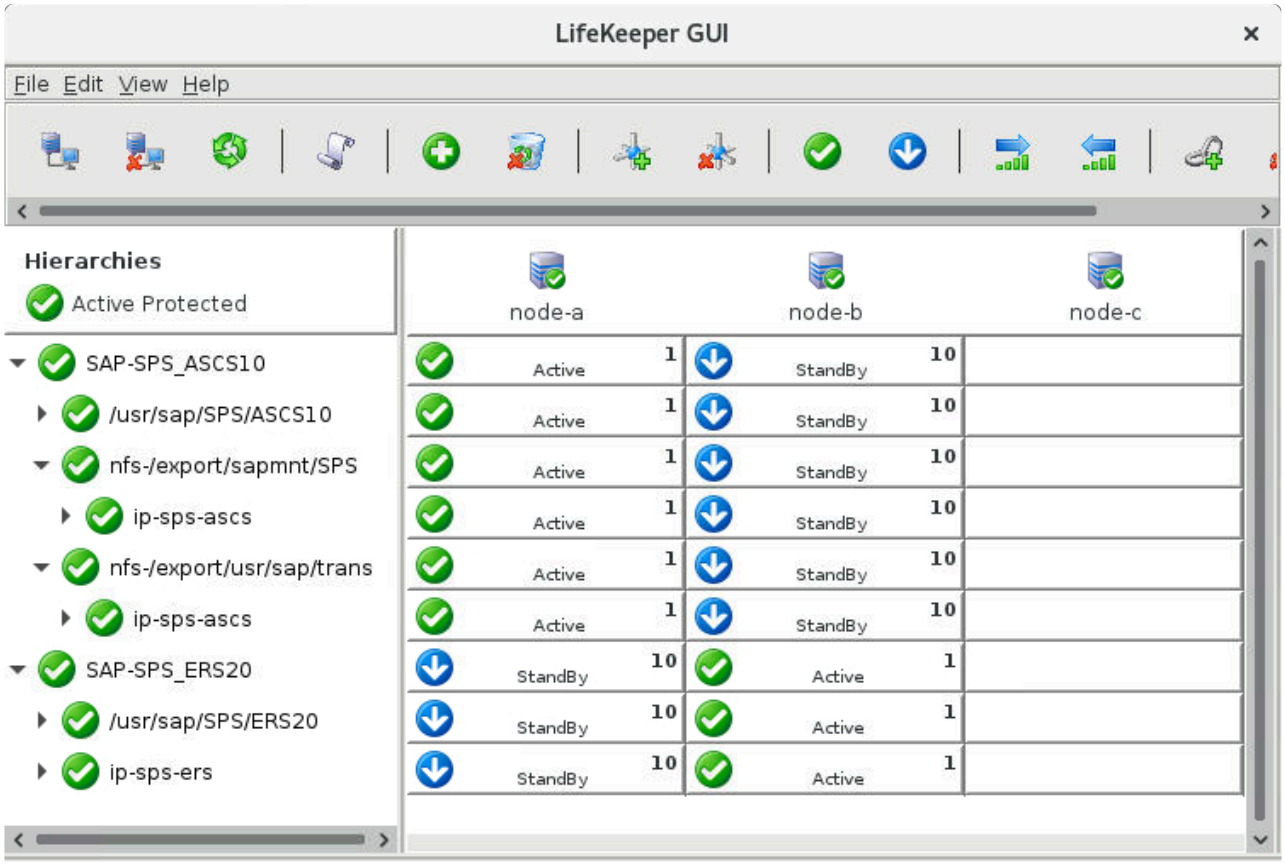



Google Cloudでは、LifeKeeper の GUI は以下の画像のようになります。

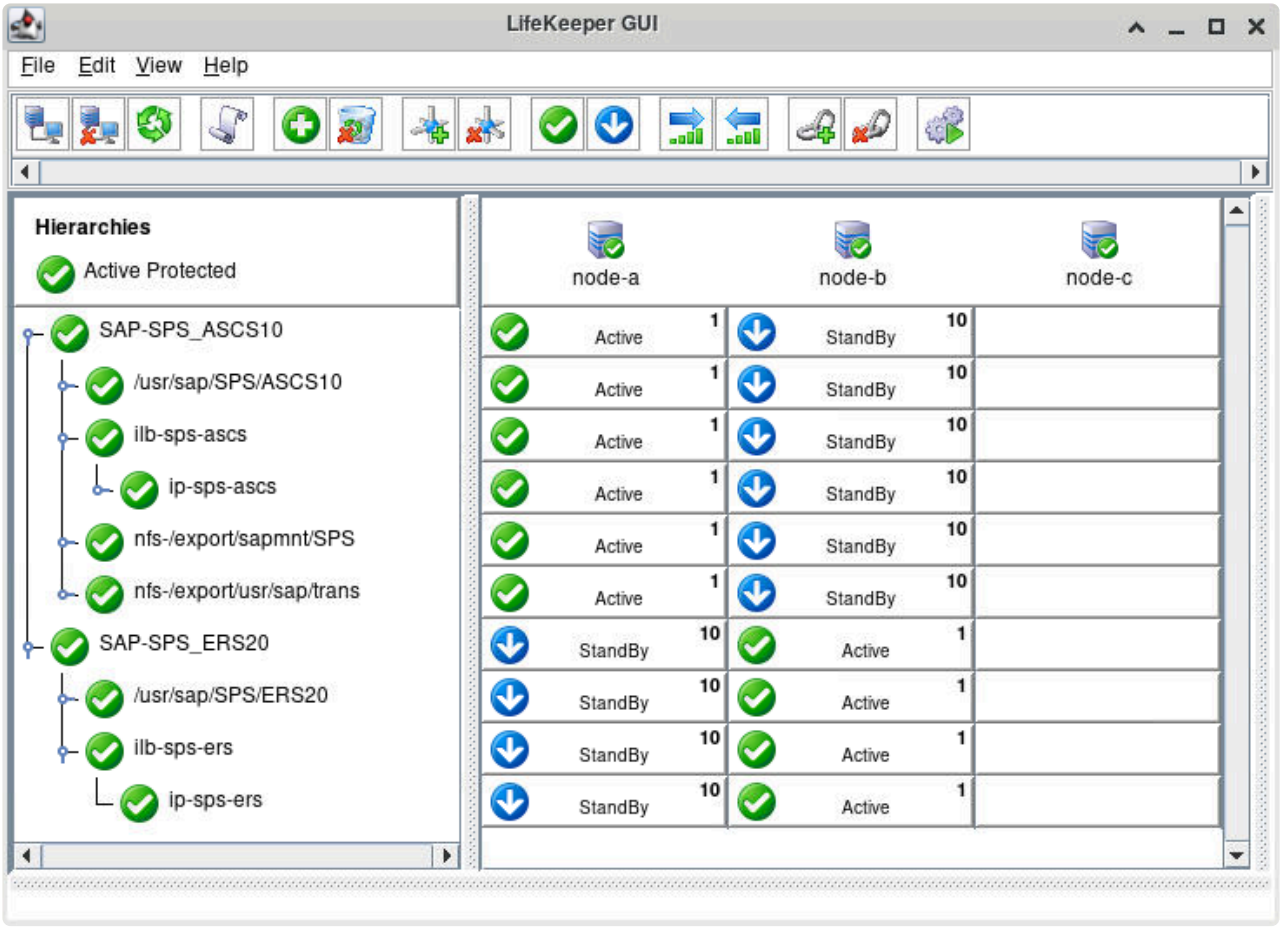


node-b がオンラインに戻ると、LifeKeeper は自動的に **SAP-SPS_ERS20** リソース階層をnode-b に再配置します。このプロセスが完了するまでに数分かかる場合があります。このプロセスが完了すると、 **SAP-SPS_ASCS10** と **SAP-SPS_ERS20** のリソース階層が、node-a と node-b でそれぞれ In-Service に戻ります。

AWS または Azure では、LifeKeeper の GUI は以下の画像のようになります。




Google Cloudでは、LifeKeeper の GUI は以下の画像のようになります。




7. 手順2、3で指定した sapcontrol コマンドを再度実行し、各ノードで期待される状態を確認します。

8. node-aで以下のコマンドを実行し、手順3で書き込んだロック100個を解除します。

```
[root@node-a ~]# su - spsadm -c "enq_admin --release_locks=100:X:DIAG::TAB:%u
pf=/usr/sap/SPS/SYS/profile/SPS_ASCS10_sps-ascs"
Enqueue Server 2

2021-03-04 21:10:22; OK; 'Release Locks'; Response=36883 usec
============================================================
```

これで、ASCS および ERS リソース階層の基本的なスイッチオーバーとフェイルオーバー機能の検証は終了です。

# 11.2.7.5.6. SAP HANA リソースの保護

このセクションでは、高可用性 SAP HANA 2.0 SPS04 クラスターをデプロイします。この構成では、SAP HANA システムレプリケーションを使用してアクティブなデータベースホストからスタンバイデータベース ホストにデータを複製し、LifeKeeper SAP HANA Recovery Kit を使用して HDB データベース インスタンスの監視、復旧、フェイルオーバーを処理します。

作成される SAP HANA クラスター ノードの詳細を以下に示します。このガイドで説明しているインスタンスのプロビジョニングは、評価でアプリケーションを実行するために **最低限** 必要な仕様を示しており、一般的に生産的なワークロードを処理するには十分ではありません。本番環境で VM をプロビジョニングする際のベスト プラクティスについては、SAP およびクラウド プロバイダーが提供するドキュメント（Hardware Requirements for SAP System Hosts など）を参照してください。

デプロイで使用するオペレーティングシステムと構成は、SAP、SIOS、およびお客様のクラウドプロバイダーによってサポートされている必要があります。SAP HANA のさまざまなバージョンでサポートされているオペレーティングシステムを確認するには、SAP の Product Availability Matrix（PAM）を参照してください。

## 評価デプロイメントのためのクラスターアーキテクチャ

- node-a (プライマリーデータベースホスト)
  - Private IP: 10.20.1.10
  - vCPU's: 8
  - Memory: 32GB
  - SAP HANA 2.0 SPS04 データベース インスタンス HDB00 が SID「SPS」の下にインストールされている
  - SAP HANA システムレプリケーションパラメーター：
    - Site name: SiteA
    - Replication mode: primary
    - Operation mode: primary
  - LifeKeeper for Linux が Quorum/Witness 機能を有効にしてインストールされ、以下の Recovery Kit がインストールされている
    - SAP HANA Recovery Kit
    - [AWS] Amazon EC2 Recovery Kit
    - [Azure, Google Cloud] Generic Application Recovery Kit for Load Balancer Health Checks (GenLB Recovery Kit)

- node-b (スタンバイデータベースホスト)
  - Private IP: 10.20.2.10
  - vCPU's: 8
  - Memory: 32GB
  - SAP HANA 2.0 SPS04 データベース インスタンス HDB00 が SID「SPS」の下にインストールされている
  - SAP HANA システムレプリケーションパラメーター：
    - Site name: SiteB
    - Replication mode: sync
    - Operation mode: logreplay

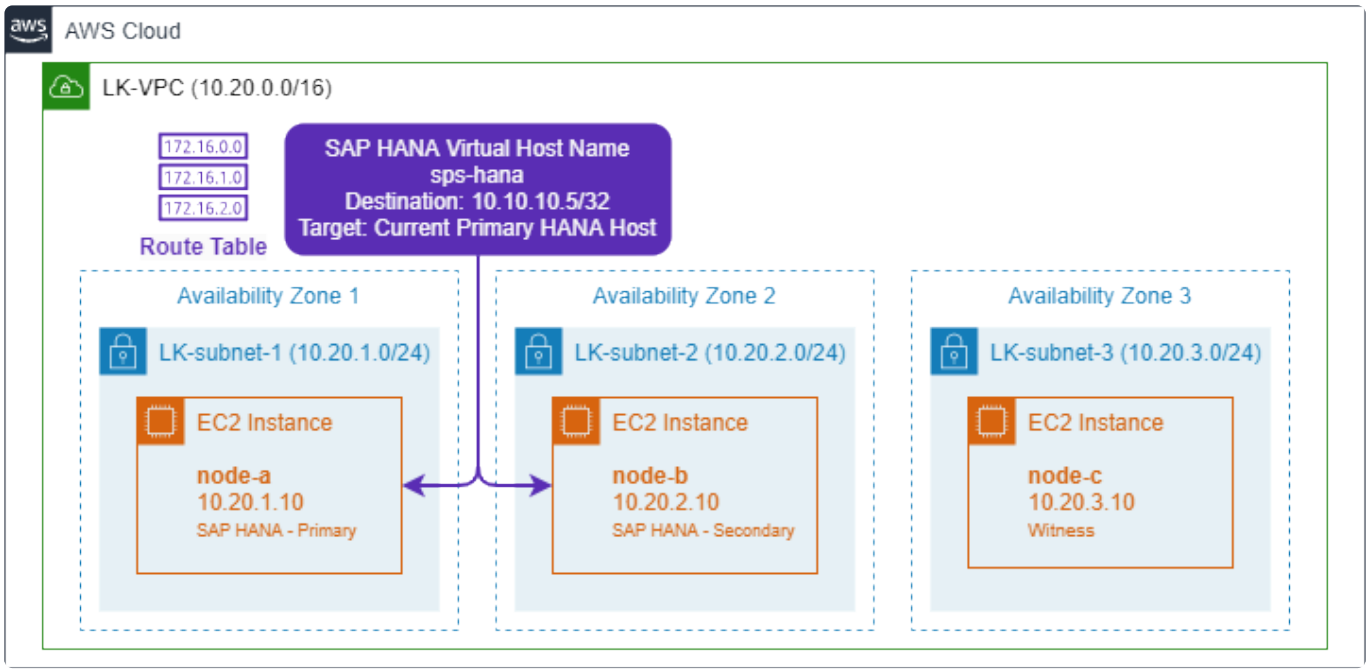
    - LifeKeeper for Linux が Quorum/Witness 機能を有効にしてインストールされ、以下の Recovery Kit がインストールされている:
        - SAP HANA Recovery Kit
        - [AWS] Amazon EC2 Recovery Kit
        - [Azure, Google Cloud] Generic Application Recovery Kit for Load Balancer Health Checks (GenLB Recovery Kit)

- node-c (Quorum/Witness ノード)
    - Private IP: 10.20.3.10
    - vCPU's: 1
    - Memory: 2GB
    - LifeKeeper for Linuxがquorum/witness機能を有効にしてインストールされています

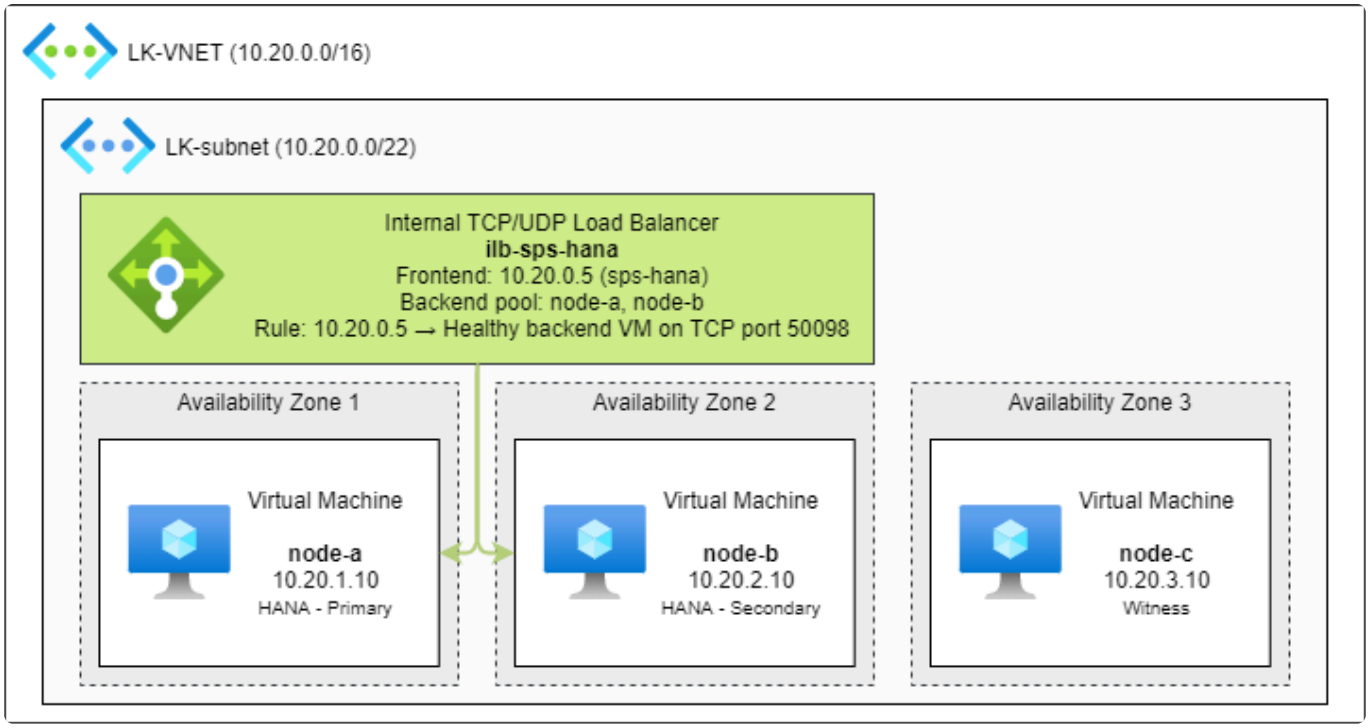
- SAP HANA virtual hostname: sps-hana

# クラウド固有のアーキテクチャ図

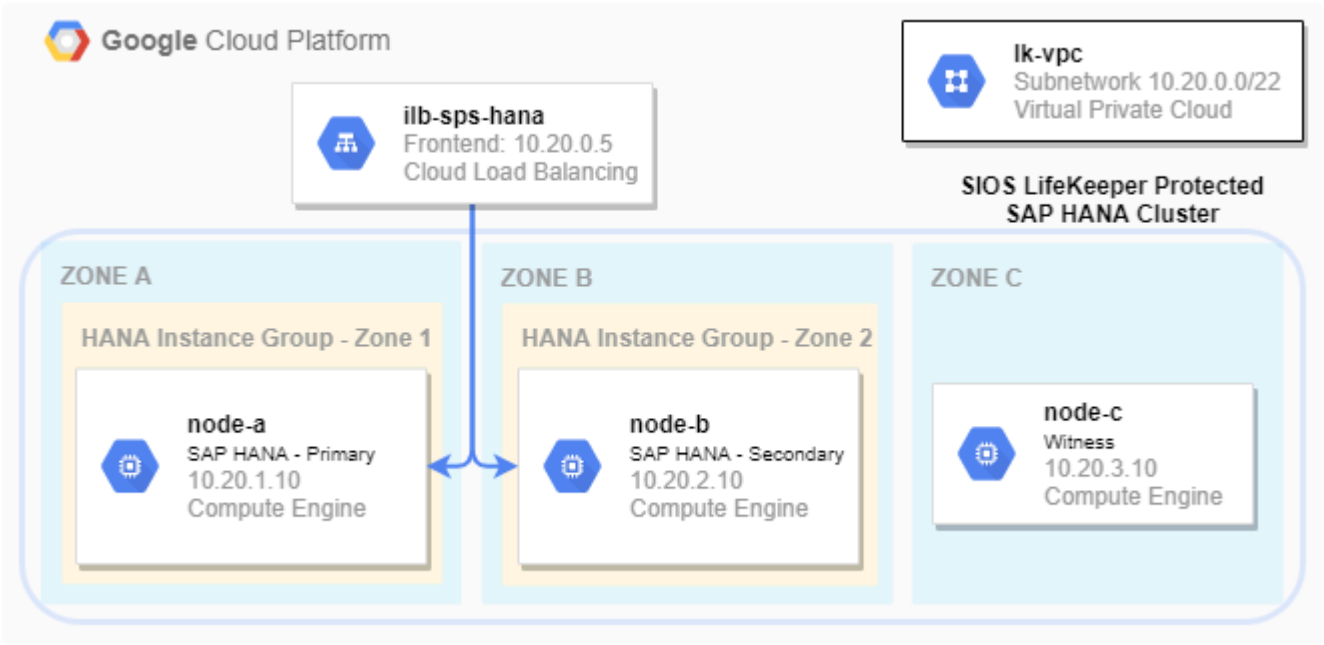
## アマゾン ウェブ サービス（AWS）上の SIOS で保護された SAP HANA クラスター



## Microsoft Azure 上の SIOS で保護された SAP HANA クラスター




## Google Cloud 上の SIOS で保護された SAP HANA クラスター




# 前提条件

このガイドを使用するには、次の前提条件がすでに完了している必要があります。

1. 3 つの VM インスタンス (node-a、node-b、および node-c) が、このガイドで前述したネットワーク規則を使用して作成されている。ノード間通信と SSH 接続を許可するファイアーウォール ルールが設定されている。詳細については、 ネットワーク コンポーネントの構成とインスタンスの作成 を参照してください。

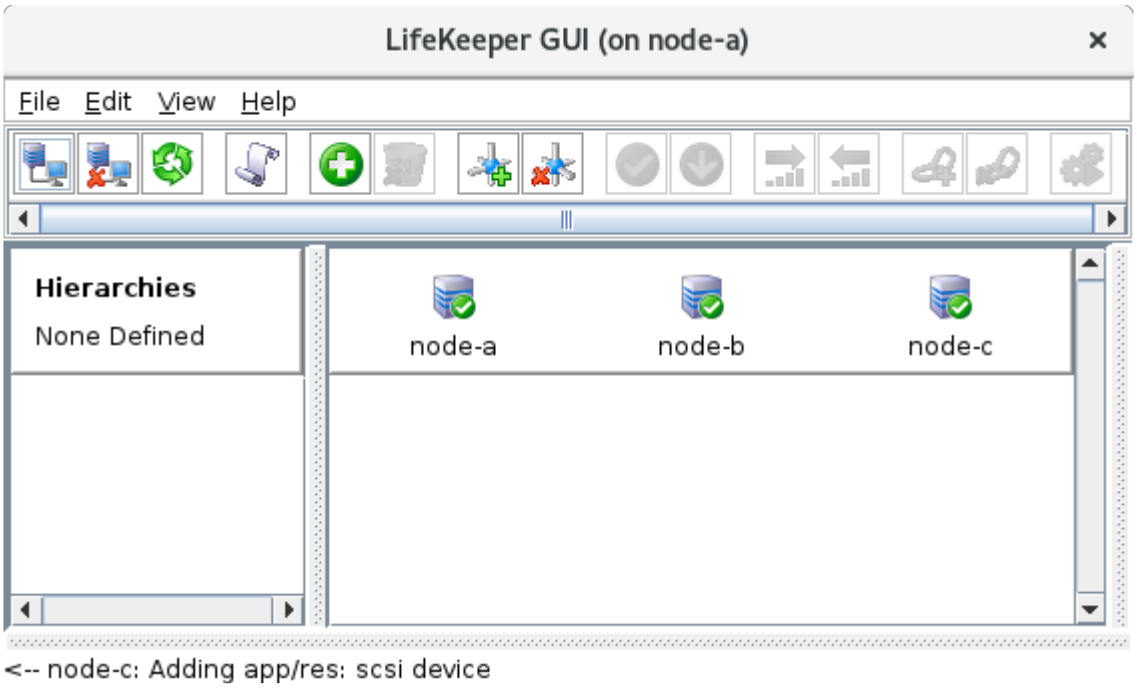
2. 3つの VM インスタンスはすべて、LifeKeeper for Linuxを実行するように設定されている。具体的には、SELinux は無効、ファイアーウォールは無効、各ノードの /etc/hosts ファイルには各ノードのホスト名をそのプライベート IP に解決するためのエントリーが含まれ、root ユーザは各ノードで同じパスワードを持っている。詳細については、 LifeKeeper for Linuxを実行するためのLinuxノードの設定 を参照してください。

3. LifeKeeper for Linux が3つのノードすべてにインストールされ、Quorum/Witness 機能が有効になっており、必要なすべての Recovery Kit トがインストールされている。Node-c（Witnessノード）には、LifeKeeper のコアインストール以外の追加のRecovery Kit は不要。各ノードに必要なすべての SIOS ライセンスがインストールされている。詳細については、 LifeKeeper for Linux をインストールする を参照してください。

> **重要:** 必要な Recovery Kit は node-a と node-b にのみインストールされるため、LifeKeeper GUI を介して実行されるこのガイドのすべての手順は、node-a または node-b のいずれかで実行する必要があります。

4. LifeKeeper のコミュニケーションパスが、クラスターノードのすべてのペア間で定義されている。なお、この場合、 **3つ** の双方向通信路（node-a ↔ node-b、node-a ↔ node-c、node-b ↔ node-c)）を作成する必要があります。詳細については、 ログインと基本設定タスク を参照してください。

これらの作業をすべて完了すると、LifeKeeper の GUI は以下の画像のようになります。



# 11.2.7.5.6.1. SAP HANA プライマリーデータベースの仮想 IP を作成する

仮想ホスト名 sps-hana は、現在のプライマリデータベースホストに解決するように構成されます

仮想 IP アドレスの推奨される実装は、クラウドプラットフォームによって異なります。 該当するセクションに記載されている手順に従ってください。

- AWS – SAP HANA 仮想 IP を作成する
- Azure – SAP HANAプライマリデータベースのロードバランサーの作成
- Google Cloud – SAP HANAプライマリデータベースのロードバランサーの作成

# 11.2.7.5.6.1.1. AWS – SAP HANA 仮想 IP を作成する

SAP HANA データベースに関連付けられた仮想 IP アドレスを保護するリソースを作成する前に、node-a と node-b で次の手順を完了する必要があります。

- AWS CLI をインストールする
- EC2 Recovery Kit を使用する権限を割り当てる
- PING ブロードキャストを無効にする
- AWS – 送信元/送信先チェックを無効にする

ここでは、AWS EC2 リソースの作成（RouteTableシナリオ） で説明した手順で、SAP HANA データベースに関連付けられた仮想 IP アドレスのリソース階層を作成します。

EC2 の RouteTable リソース タイプで使用される IP アドレスは、VPC CIDR (LK-VPC の場合は 10.20.0.0/16) の外にある必要があるため、次の仮想ホスト名と IP アドレスを使用します。

| Instance | Virtual Host Name | IP Address |
|---|---|---|
| HDB00 | sps-hana | 10.10.10.5 |

## SAP HANA 仮想 IP リソース階層を作成する

1. IP リソースの作成 で説明している手順に従って、次のパラメータを使用して LifeKeeper IP リソース (ip-sps-hana) を作成および拡張し、node-a および node-b の SAP HANA 仮想 IP アドレスを保護します。IP リソースは node-a で作成され、node-b に拡張されることに注意してください。 また、作成されたリソースは Witness ノードである node-c に拡張してはならないことに注意してください。 アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent |
| Server | node-a |
| IP Resource | 10.10.10.5 |
| Netmask | 255.255.255.0 |
| Network Interface | eth0 |
| IP Resource Tag | ip-sps-hana |
| **Pre-Extend Wizard** | |
| Target Server | node-b |

| | |
|---|---|
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend comm/ip Resource Hierarchy Wizard** | |
| IP Resource | 10.10.10.5 |
| Netmask | 255.255.255.0 |
| Network Interface | eth0 |
| IP Resource Tag | ip-sps-hana |

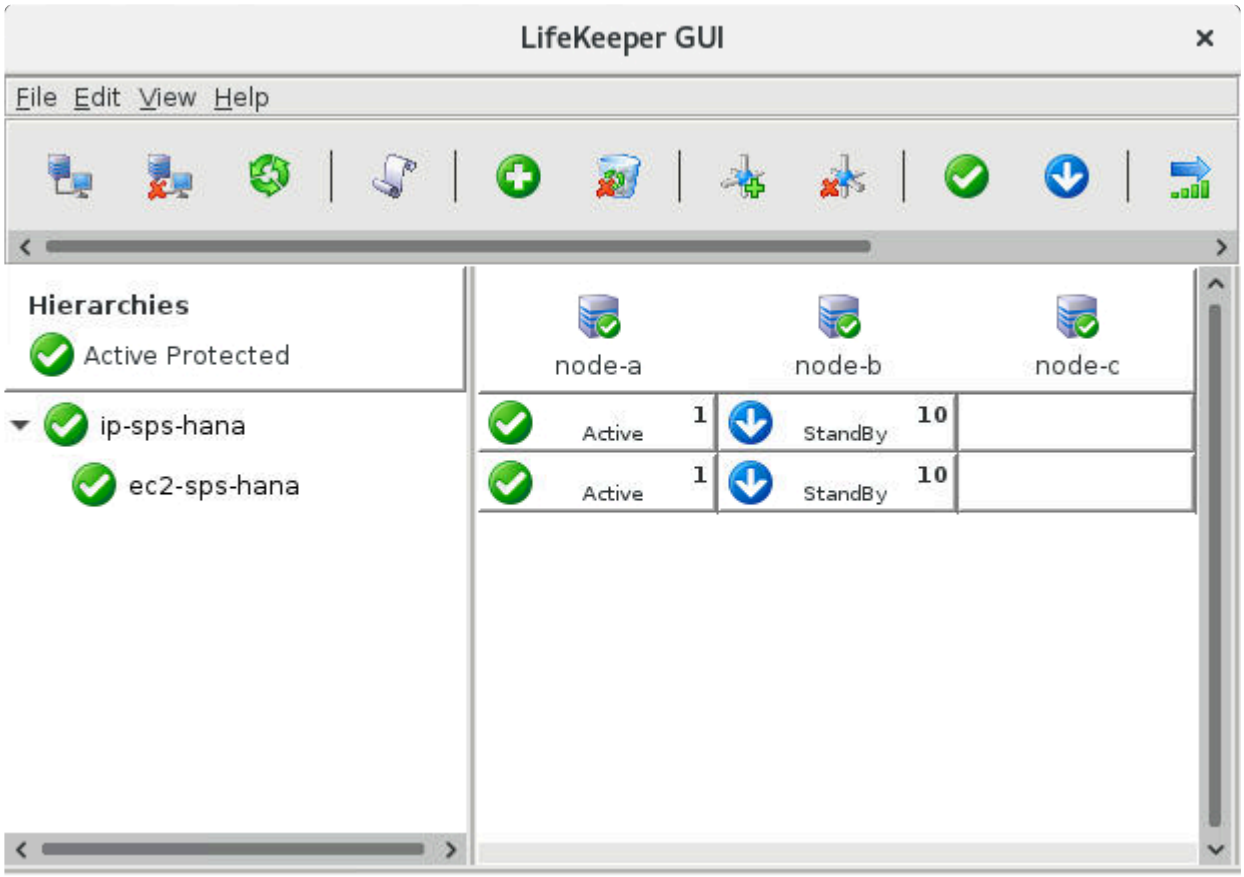
2. AWS EC2 リソースの作成 で説明している手順に従って、次のパラメータを使用して LifeKeeper EC2 リソース (**ec2-sps-hana**) を作成および拡張し、IP リソースがスイッチオーバーされた際に AWS でルートテーブルのバックエンド操作を管理します。 切り替えました。EC2 リソースは node-a で作成され、node-b に拡張されることに注意してください。また、作成されたリソースは Witness ノードである node-c に拡張してはならないことに注意してください。 アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent |
| Server | node-a |
| EC2 Resource type | RouteTable (Backend cluster) |
| IP Resource | ip-sps-hana |
| EC2 Resource Tag | ec2-sps-hana |
| **Pre-Extend Wizard** | |
| Target Server | node-b |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend comm/ec2 Resource Hierarchy Wizard** | |

| EC2 Resource Tag | ec2-sps-hana |
|---|---|

**Note:** EC2リソースの作成･拡張に失敗した場合は、(i)AWS の LK-VPC のルートテーブルに10.10.10.5/32から Node-A への経路が存在すること、 (ii)Node-A と Node-B で送信元/送信先チェックが無効になっていること、(iii)依存する IP リソース ip-sps-hana の優先順位がそれぞれ Node-A が1、Node-B が10になっていることを確認してください。

IP リソース階層が正常に作成されると、LifeKeeper GUI は次の図のようになります。




## /etc/hosts にエントリーを追加する

node-a と node-b の /etc/hosts に次のエントリーを追加して、仮想ホスト名を対応する IP アドレスに解決できるようにします。

```
10.10.10.5    sps-hana
```

# 11.2.7.5.6.1.2. Azure – SAP HANA プライマリーデータベースロードバランサーを作成する

- Microsoft Azure では、プライマリー HANA データベースホストに関連付けられた仮想 IP のフェイルオーバーを容易にするために、TCP 内部ロードバランサーを使用します。ロードバランサーのフロントエンドには、それが動作するサブネットから IP アドレスが割り当てられます。この例では、仮想ホスト名に対応するロードバランサーに次の IP アドレスを使用します。

| Instance | Virtual Host Name | Internal TCP Load Balancer | Frontend IP Address |
|---|---|---|---|
| HDB00 | sps-hana | ilb-sps-hana | 10.20.0.5 |

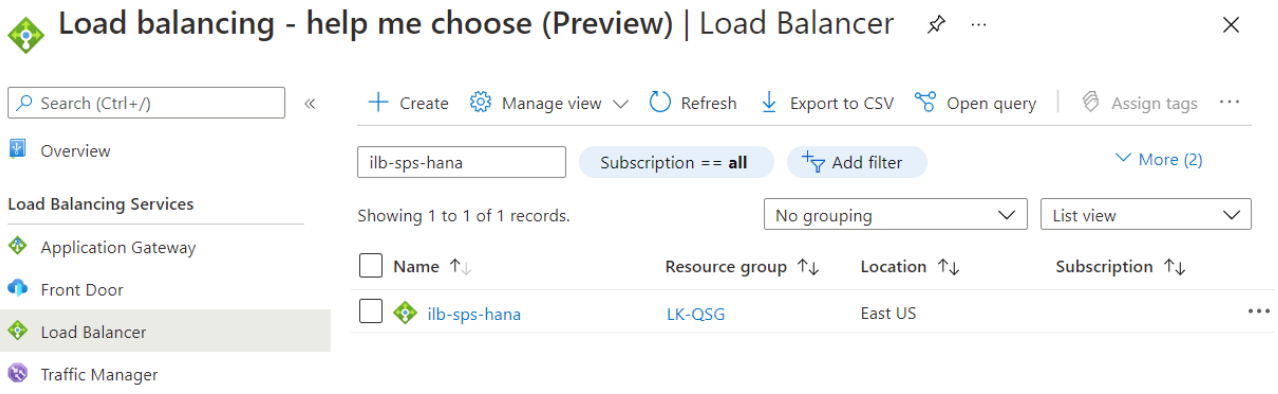
Azure – 内部ロードバランサーを使用する と ロードバランサーのヘルス チェックに応答する の手順に従って、以下の Azure と LifeKeeper のリソースを作成し、設定します。

1. 次のプロパティを使用して、Azure でロードバランサー **ilb-sps-hana** を作成します。

| Name | ilb-sps-ascs-ers |
|---|---|
| **Create Load Balancer** | |
| Resource Group | LK-QSG |
| Name | ilb-sps-hana |
| Region | (仮想マシンと同じ) |
| Type | Internal |
| SKU | Standard (ワークロードが可用性ゾーン全体に分散されるため、[Standard] を選択します) |
| Tier | Regional |
| **Frontend IP Configuration** | |
| Name | SPSHANAFrontEnd |
| Virtual Network | LK-VNET |
| Subnet | LK-subnet (10.20.0.0/22) |
| Assignment | Static |
| IP address | 10.20.0.5 |
| Availability zone | Zone-redundant |
| **Backend Pool** | |
| Name | backend-sps-hana |
| Backend Pool Configuration | NIC |

| | |
|---|---|
| IP Version | IPv4 |
| Virtual machines | node-a, node-b |
| **Health Probe** | |
| Name | probe-sps-hana |
| Protocol | TCP |
| Port | 50098 (これは、後で作成される対応するLB Health Check リソースと一致する必要があります) |
| Interval | 5 seconds |
| Unhealthy threshold | 2 consecutive failures |
| **Load Balancing Rule** | |
| Name | ilb-rule-sps-hana |
| IP Version | IPv4 |
| Frontend IP address | SPSHANAFrontEnd (10.20.0.5) |
| HA Ports | Click (評価でのデプロイを簡素化するため、すべてのポートへの転送を許可します) |
| Backend pool | backend-sps-hana |
| Health probe | probe-sps-hana (TCP:50098) |
| Session persistence | None |
| Idle timeout | 4 minutes |
| TCP reset | Disabled |
| Floating IP | Enabled |

作成すると、Microsoft Azure コンソールの **Load Balancing Services** → **Load Balancer** ページに、新しく作成されたロードバランサーが表示されます。

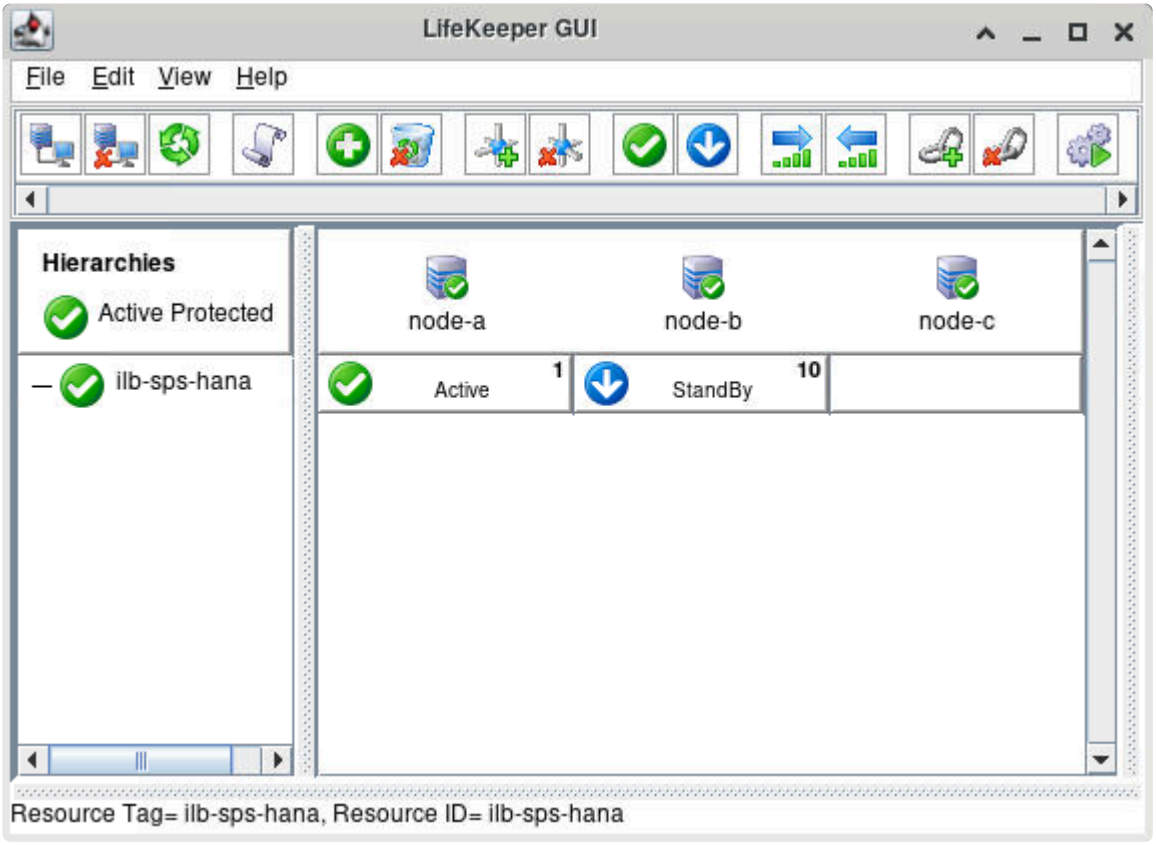



2. 以下の ロードバランサーのヘルスチェックに応答する 手順に従って、LB Health Check をインストールし、以下のプロパティを使用してLB Health Check リソース、 ilb-sps-hana を作成します。

*ilb-sps-hana* は node-a で作成され、node-b に拡張されることに注意してください。 アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent |
| Server | node-a |
| Reply daemon Port | 50098 |
| Reply daemon message | 空欄のままにします |
| LB Health Check Resource Tag | ilb-sps-hana |
| **Pre-Extend Wizard** | |
| Target Server | node-b |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend LB Health Check Resource Hierarchy Wizard** | |
| LB Health Check Resource Tag | ilb-sps-hana |

リソースが作成され、正常に拡張されると、LifeKeeper GUI リソースペインにリソースが表示されます。

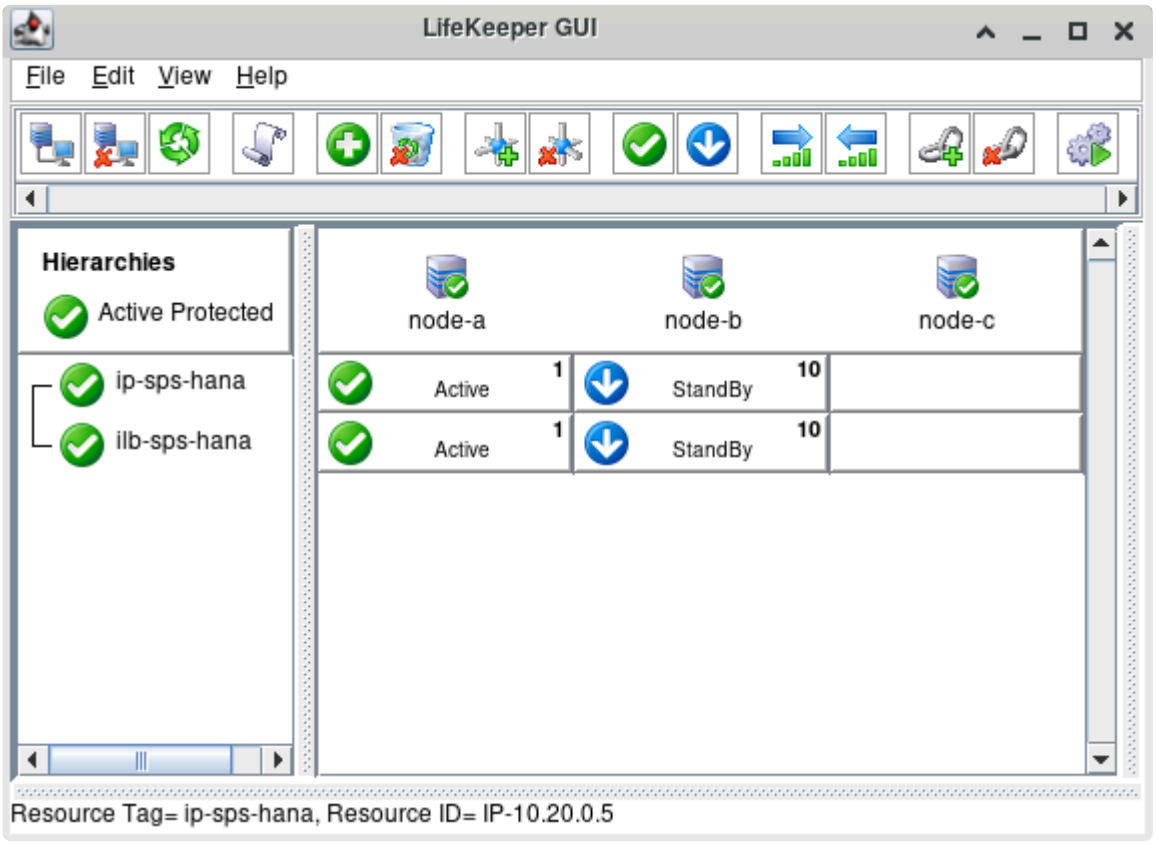



3. IP リソースの作成 で説明している手順に従って、次のパラメータを使用して LifeKeeper IP リソース (**ip-sps-hana**) を作成および拡張し、node-a および node-b で HANA データベースの仮想 IP アドレスを保護します。IP リソースは node-a で作成され、node-b に拡張されることに注意してください。 また、作成されたリソースは Witness ノードである node-c に拡張してはならないことに注意してください。 アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent |
| Server | node-a |
| IP Resource | 10.20.0.5 |
| Netmask | 255.255.252.0 |
| Network Interface | eth0 |
| IP Resource Tag | ip-sps-hana |
| **Pre-Extend Wizard** | |
| Target Server | node-b |
| Switchback Type | intelligent |

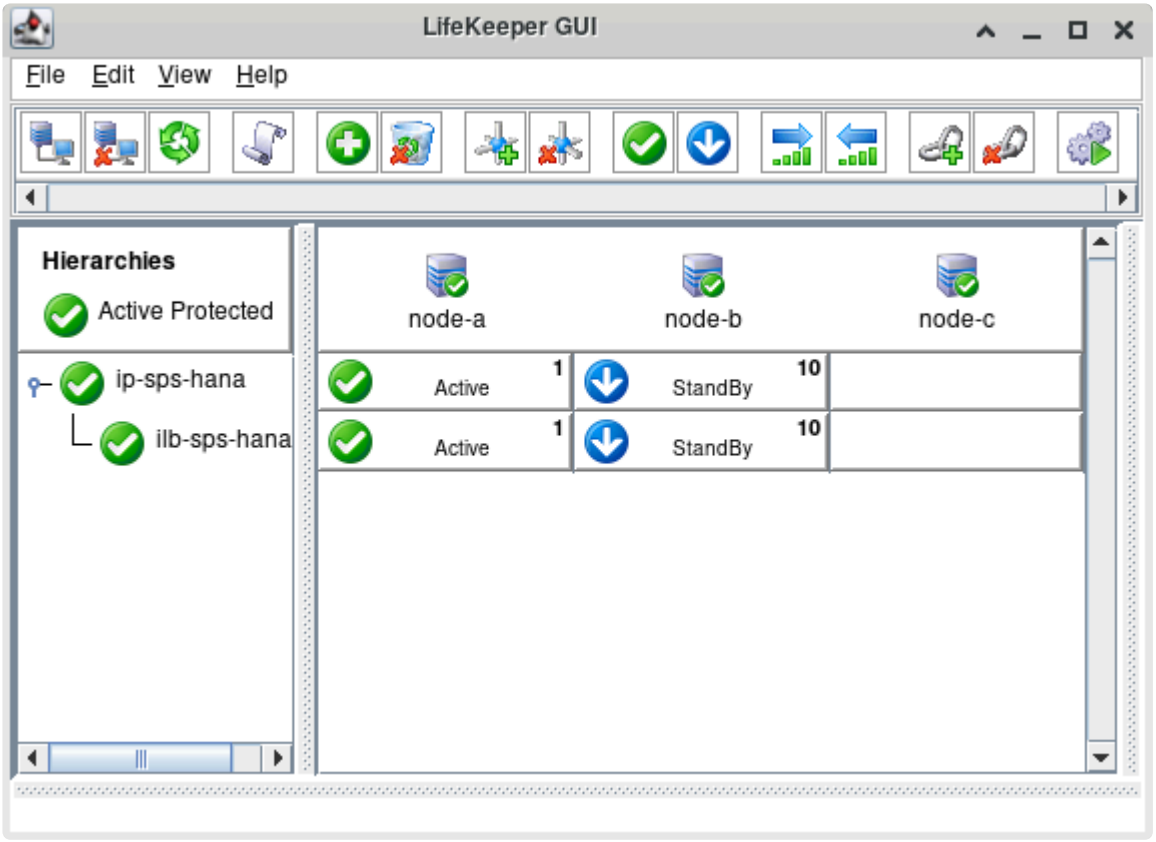
| | |
|---|---|
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend comm/ip Resource Hierarchy Wizard** | |
| IP Resource | 10.20.0.5 |
| Netmask | 255.255.252.0 |
| Network Interface | eth0 |
| IP Resource Tag | ip-sps-hana |

IP リソースが正常に作成されると、LifeKeeper GUI は次の図のようになります。




4. node-aの **ip-sps-hana** リソースを右クリックし、 **[Create Dependency...]** をクリックします。 子リソースタグに **ilb-sps-hana** を指定し、 **[Create Dependency]** をクリックします。

   依存関係が正常に作成されると、LifeKeeper GUI は次の図のようになります。




5. 次のエントリーを node-a と node-b の /etc/hosts に追加して、仮想ホスト名を対応するロードバランサーのフロントエンド IP アドレスに解決できるようにします。

```
10.20.0.5 sps-hana
```

6. ロードバランサーのヘルスチェックに応答する の「**LB Health Check リソースのスイッチオーバーとフェイルオーバーのテスト**」セクションの説明に従って、LB Health Check リソース階層のスイッチオーバーとフェイルオーバーをテストします。見つかった問題を修正したり、ロードバランサーのヘルスチェックのパラメーターを必要に応じて調整したりして、正常に動作するようにします。

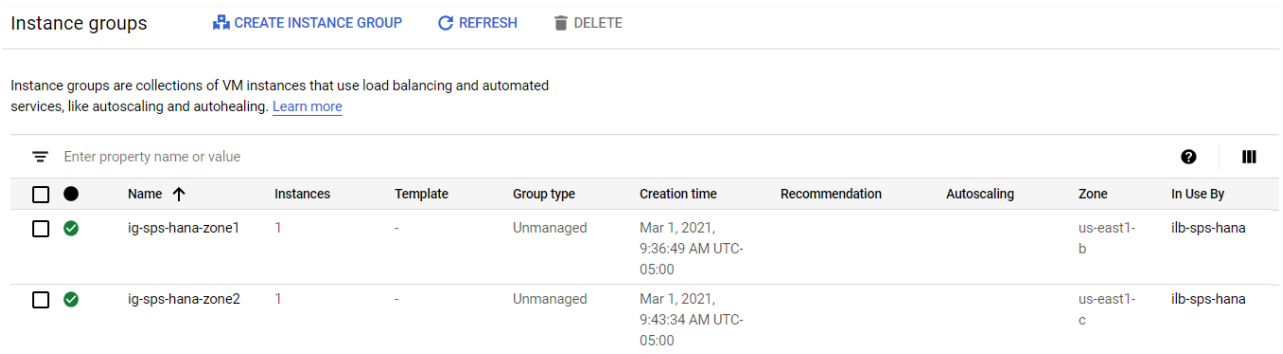
# 11.2.7.5.6.1.3. Google Cloud – SAP HANA プライマリーデータベースロードバランサーを作成する

Google Cloud では、SAP HANA の仮想IPのフェイルオーバーを容易にするために、TCP 内部ロードバランサーが使用されます。ロードバランサーのフロントエンドには、それが動作するサブネットからエフェメラル IP が割り当てられます。この例では、sps-hana 仮想ホスト名に対応するロードバランサーに次の IP アドレスを使用します。

| Instance | Virtual Host Name | Internal TCP Load Balancer | Frontend IP Address |
|---|---|---|---|
| HDB00 | sps-hana | ilb-sps-hana | 10.20.0.5 |

Google Cloud – 内部ロードバランサーを使用する と ロードバランサーのヘルス チェックに応答する の手順に従い、以下の Google Cloud および LifeKeeper リソースを作成し、設定します

1. Node-aを含む **ig-sps-hana-zone1** と node-b を含む **ig-sps-hana-zone2** という2つのアンマネージドインスタンスグループを作成します。

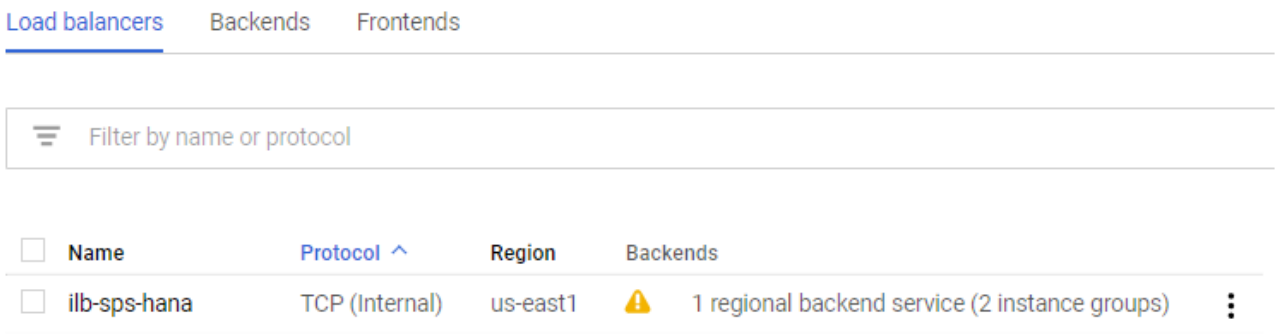



2. 次のプロパティを使用して、TCP 内部ロードバランサー **ilb-sps-hana** を作成します。 アイコンは、デフォルトのオプションが選択されていることを示します。

| Name | ilb-sps-hana |
|---|---|
| **Backend configuration** | |
| Region | <Deployment region> (e.g., us-east1) |
| Network | lk-vpc |
| Backends | ig-sps-hana-zone1<br>ig-sps-hana-zone2 |
| **Health Check** | |
| Name | hc-sps-hana |
| Protocol | TCP |

| | |
|---|---|
| Port | 50098 (これは、後で作成される対応する LB Health Check リソースと一致する必要があります) |
| Proxy protocol | NONE |
| Request | 空欄のままにします |
| Response | 空欄のままにします |
| Check interval | 5 seconds |
| Timeout | 5 seconds |
| Healthy threshold | 2 consecutive successes |
| Unhealthy threshold | 2 consecutive failures |
| Connection draining timeout | 10 seconds |
| **Frontend configuration** | |
| Name | fe-sps-hana |
| Subnetwork | lk-subnet |
| Purpose | Non-shared |
| IP address | Ephemeral (Custom) |
| Custom ephemeral IP address | 10.20.0.5 |
| Ports | All |
| Global access | Disable |
| Service label | 空欄のままにします |

作成すると、Google Cloud Console の **Network services → Load balancing** ページにロードバランサーが表示されます。



3. ”Google Cloud – 内部ロードバランサーを使用する”:#gcp-using-an-internal-load-balancer の「**IP 転送を無効にする**」セクションで説明している手順に従って、node-a と node-b がロードバランサーのフロントエンド IP アドレスを介して通信できるようにします。変更を反映するために、node-aとnode-bを再起動します。

   ロードバランサーのフロントエンド IP アドレスを使ってプライマリーデータベースに接続するためには、このステップは厳密には必要ではありませんが、“スイッチオーバーとフェールオーバーのテスト”#test-switchover-and-failover-sap-hana セクションで説明するように、クラスター内からデータベース接続をテストできるようになります。

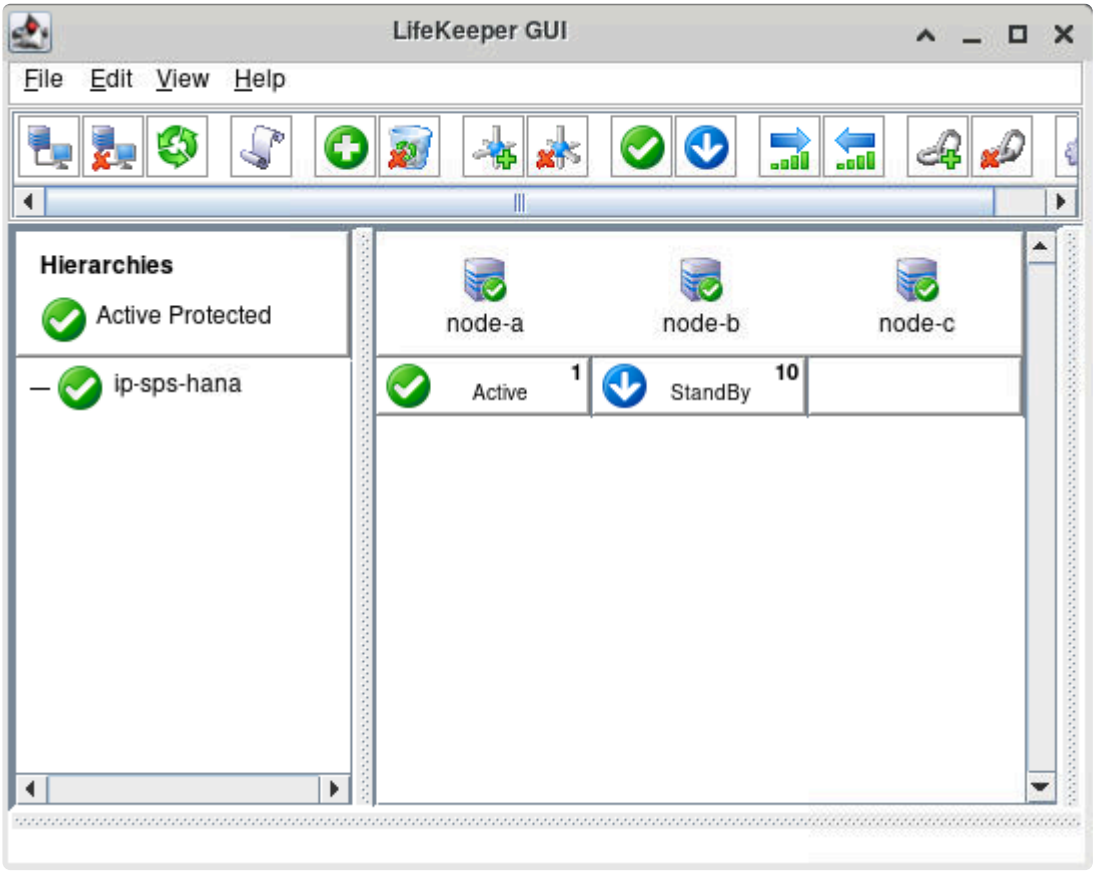
4. IPリソースの作成 で説明している手順に従って、以下のパラメーターを使用して、ネットマスク255.255.255.255の LifeKeeper IP リソース（ip-sps-hana）を作成および拡張し、node-a および node-b 上の SAP HANA 仮想IPアドレスを保護します。*ip-sps-hana* は node-a で作成され、node-b に拡張されることに注意してください。また、この IP リソースは、Witness ノードである node-c に拡張しないでください。 ✓ アイコンは、デフォルトのオプションが選択されていることを示します。

> **IMPORTANT** Netmask パラメーターの値として、(作成時と拡張時の両方で) 255.255.255.255 を使用することが重要です。拡張プロセス中に「Accept Defaults」を選択しないように注意してください。これにより、通常、255.255.255.0 がスタンバイノードのネットワークマスクとして誤って使用され、予期しないネットワークの問題が発生する可能性があります。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent ✓ |
| Server | node-a |
| IP Resource | 10.20.0.5 |
| Netmask | 255.255.255.255 |
| Network Interface | eth0 ✓ |
| IP Resource Tag | ip-sps-hana |
| **Pre-Extend Wizard** | |
| Target Server | node-b |
| Switchback Type | intelligent ✓ |
| Template Priority | 1 ✓ |
| Target Priority | 10 ✓ |
| **Extend comm/ip Resource Hierarchy Wizard** | |

| | |
|---|---|
| IP Resource | 10.20.0.5 |
| Netmask | 255.255.255.255 |
| Network Interface | eth0 |
| IP Resource Tag | ip-sps-hana |

IP リソースが正常に作成されると、LifeKeeper GUI は次の図のようになります。

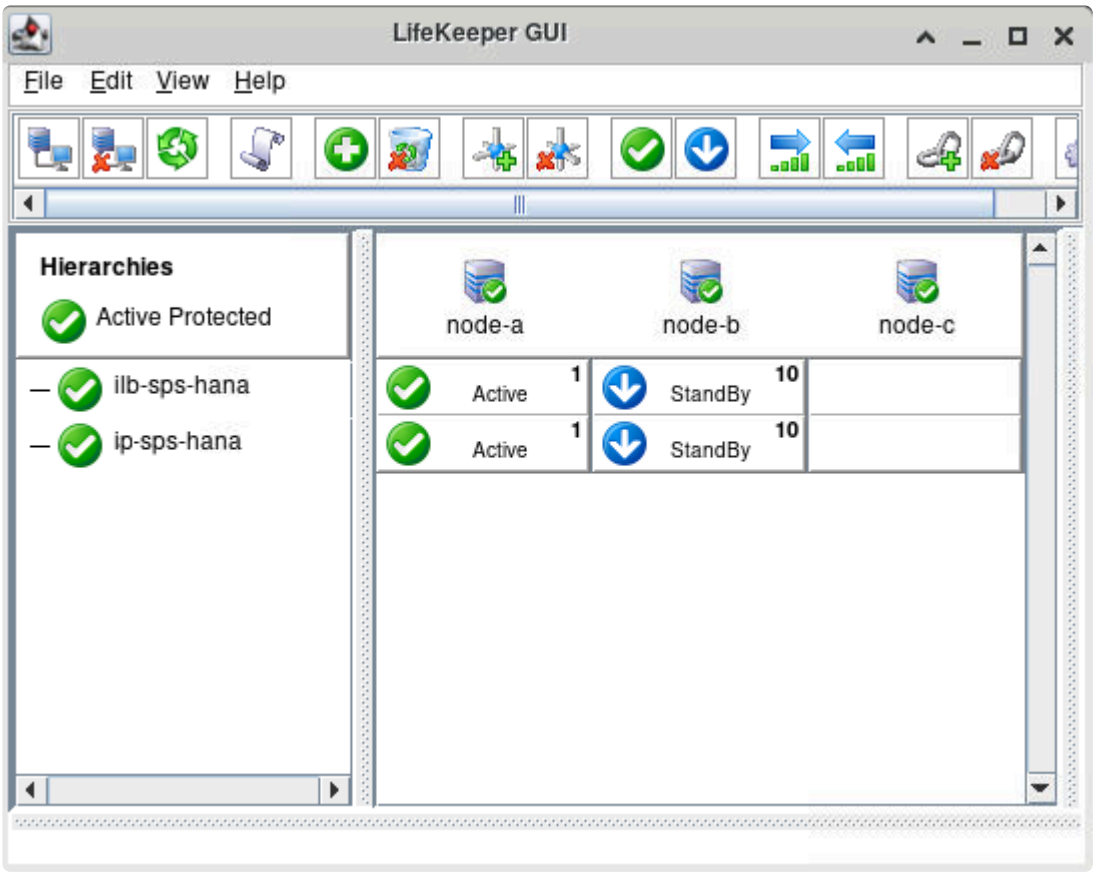



5. ロードバランサーのヘルス チェックに応答する の手順に従って、以下のプロパティを使用してLB Health Checkリソース、 **ilb-sps-hana** を作成します。 アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent |
| Server | node-a |
| Reply daemon Port | 50098 |
| Reply daemon message | None (Empty) |

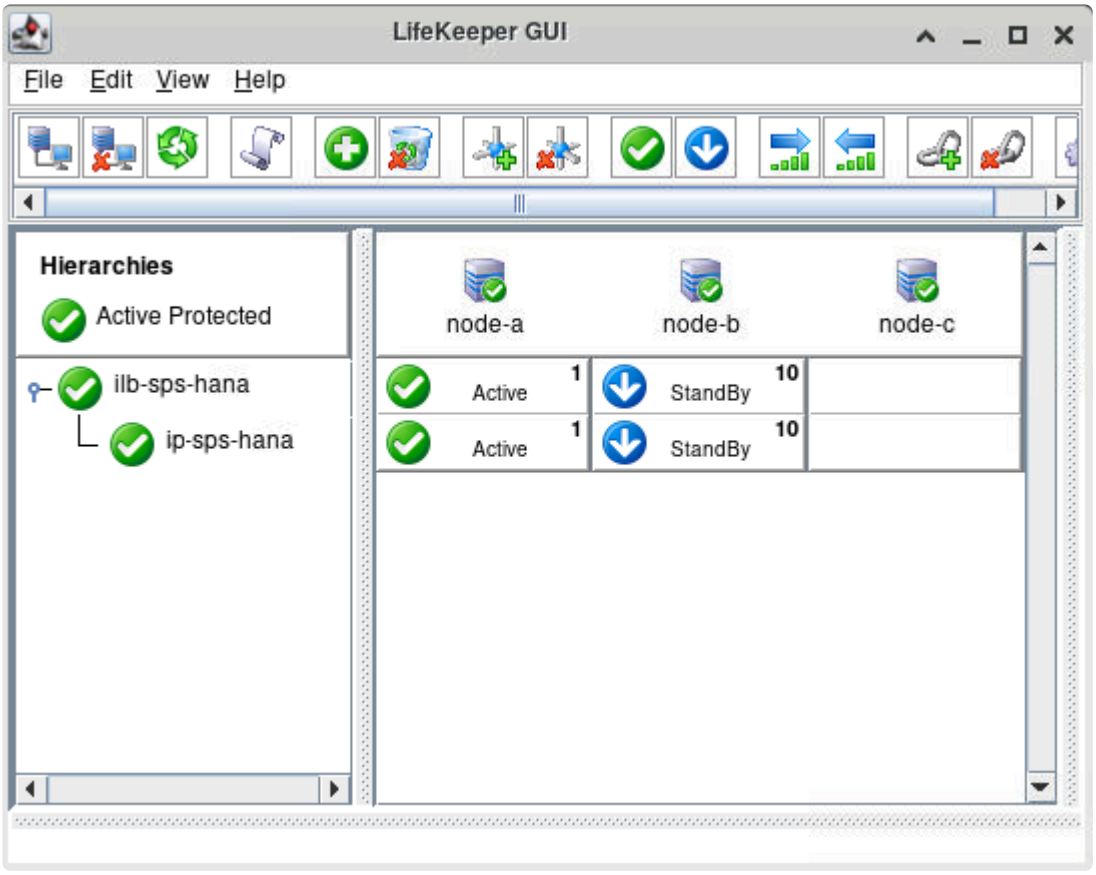
| | |
|---|---|
| LB Health Check Resource Tag | ilb-sps-hana |
| **Pre-Extend Wizard** | |
| Target Server | node-b |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend LB Health Check Resource Hierarchy Wizard** | |
| LB Health Check Resource Tag | ilb-sps-hana |

リソースが作成され、正常に拡張されると、LifeKeeper GUI リソースペインにリソースが表示されます。




6. node-a の **ilb-sps-hana** リソースを右クリックし、**[Create Dependency...]** をクリックします。子リソースタグに **ip-sps-hana** を指定し、**[Create Dependency]** をクリックします。

   依存関係が作成されると、LifeKeeper GUI は次の図のようになります。




7. 次のエントリーを node-a と node-b の /etc/hosts に追加して、仮想ホスト名を対応するロードバランサーのフロントエンド IP アドレスに解決できるようにします。

```
10.20.0.5 sps-hana
```

8. ロードバランサーのヘルス チェックに応答する の「**LB Health Check リソースのスイッチオーバーとフェイルオーバーのテスト**」セクションの説明に従って、 LB Health Check リソースのスイッチオーバーとフェイルオーバーをテストします。見つかった問題を修正したり、ロードバランサーのヘルスチェックのパラメーターを調整したりして、正常に動作するようにします。

# 11.2.7.5.6.2. SAP HANA ファイルシステムのディスクをアタッチする

1. SAP HANA のインストールをサポートするために、3つの追加ディスクを作成して node-a と node-b に接続します。 この例で使用されているデバイス名は、お使いの環境によって異なる場合があります（例：/dev/sdb ではなく /dev/xvdb など）。これらのディスクサイズは、評価目的にのみ使用されることに注意してください。本番環境でリソースをプロビジョニングする場合は、SAP、クラウドプロバイダー、およびサードパーティのストレージプロバイダーの関連ドキュメントを参照してください。

| Device Name | Minimum Size | Mount Point |
|---|---|---|
| /dev/sdb | 20GB | /hana/log |
| /dev/sdc | 30GB | /hana/data |
| /dev/sdd | 60GB | /hana/shared |

**Note:** SAP HANA のインストールをサポートするためにストレージを接続する場合は、他の構成を使用できます。たとえば、LVM を介して複数の論理パーティションで分割された単一の物理ボリュームを使用すると便利な場合があります。

2. node-a と node-b の両方で次のコマンドを実行して、xfs ファイル システムで /dev/sdb1 パーティションを作成します。

```
[root@node-a ~]# parted /dev/sdb --script mklabel gpt mkpart xfspart xfs 0% 10
0%

[root@node-a ~]# mkfs.xfs /dev/sdb1
meta-data=/dev/sdb1    isize=512    agcount=4, agsize=1310592 blks
         =             sectsz=4096 attr=2, projid32bit=1
         =             crc=1       finobt=1, sparse=1, rmapbt=0
         =             reflink=1
data     =             bsize=4096  blocks=5242368, imaxpct=25
         =             sunit=0     swidth=0 blks
naming   =version 2    bsize=4096  ascii-ci=0, ftype=1
log      =internal log bsize=4096  blocks=2560, version=2
         =             sectsz=4096 sunit=1 blks, lazy-count=1
realtime =none         extsz=4096  blocks=0, rtextents=0

[root@node-a ~]# partprobe /dev/sdb1
```

node-a と node-b の両方で、/dev/sdc と /dev/sdd に対してこれらのコマンドを繰り返します。

3. 次のコマンドを実行して、SAP HANA ファイル システムのマウント ポイントを作成します。

```
[root@node-a ~]# mkdir -p /hana/{log,data,shared}
```

4. blkid コマンドを使用して、各パーティションの UUID を表示します。

```
[root@node-a ~]# blkid | sort
/dev/sdb1: UUID="f058a8de-4e17-4c1d-97a3-a2672aff58dd" TYPE="xfs" PARTLABEL="x
fspart" PARTUUID="f780893d-bd61-4beb-89df-fb23ea04f16e"
/dev/sdc1: UUID="c0ab7218-ec24-4a6f-b883-37e505ec806c" TYPE="xfs" PARTLABEL="x
fspart" PARTUUID="34c47333-7174-468c-b8f0-a3491cd0fe0c"
/dev/sdd1: UUID="057f4f63-8fb3-486b-82b5-49fbbd4914cc" TYPE="xfs" PARTLABEL="x
fspart" PARTUUID="0c9cc7d3-9622-4b3f-8949-e744a02af466"
```

5. 手順3で取得した UUID を参照して、node-a と node-b の /etc/fstab に以下のエントリーを追加し、起動時にファイルシステムがマウントされるようにします。この例のサンプル UUID を、お使いのシステム上の適切なパーティションの UUID に置き換えます。

```
UUID=f058a8de-4e17-4c1d-97a3-a2672aff58dd       /hana/log      xfs     default
s        0 0
UUID=c0ab7218-ec24-4a6f-b883-37e505ec806c       /hana/data       xfs     defau
lts        0 0
UUID=057f4f63-8fb3-486b-82b5-49fbbd4914cc       /hana/shared    xfs     defaul
ts        0 0
```

6. node-a と node-b で次のコマンドを実行して、SAP HANA ファイル システムを両方のノードにマウントします。

```
mount /hana/log
mount /hana/data
mount /hana/shared
```

これで、SAP HANA をインストールするためのファイルシステムの準備が完了しました。

# 11.2.7.5.6.3. SAP HANA のインストールとシステムレプリケーションの設定

## node-a に SAP HANA をインストールする

1. SAP HANA インストール ファイルを含む /sap-install ディレクトリを node-a に作成します。

```
mkdir /sap-install
```

> **Note:** 選択したインスタンスタイプの root ファイルシステムに、必要な SAP HANA インストールファイルを一度解凍して保存するのに十分なディスクスペースがない場合、SAP HANA ソフトウェアのインストールを完了するために、各インスタンスに一時ディスクを接続し、それを /sap-install にマウントすることができます。

2. SAP HANA インストールファイルを node-a の /sap-install にダウンロードします。

3. SAP HANA のインストールファイルを展開します。この例では、SAP HANA 2.0 SPS04 Platform Edition をインストールします。

```
[root@node-a ~]# unzip /sap-install/hana-2-platform-edition-sp4.zip -d /sap-in
stall/
```

4. hdblcm 実行ファイルを実行して、インストールを開始します。

```
[root@node-a ~]# /sap-install/DATA_UNITS/HDB_LCM_LINUX_X86_64/hdblcm
```

5. 次のパラメーターを入力して、SAP HANA データベース サーバーの基本的なインストールを完了します。デプロイに必要な場合は、インストール中にその他のオプションのコンポーネントを選択します。 アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| Enter selected action index | 1 (install) |
| Components to install | 2 (HDB Server) |
| Enter Installation Path | /hana/shared |
| Enter Local Host Name | node-a |
| Do you want to add hosts to the system? (y/n) | n |
| Enter SAP HANA System ID | SPS |

| | |
|---|---|
| Enter Instance Number | 00 |
| Enter Local Host Worker Group | default |
| Select Usage / Enter Index | 4 (custom) |
| Enter Location of Data Volumes | /hana/data/SPS |
| Enter Location of Log Volumes | /hana/log/SPS |
| Restrict maximum memory allocation? | n |
| Enter Certificate Host Name for Host 'node-a' | node-a |
| Enter System Administrator (spsadm) Password | <spsadm User Password> |
| Confirm System Administrator (spsadm) Password | <spsadm User Password> |
| Enter System Administrator Home Directory | /usr/sap/SPS/home |
| Enter System Administrator Login Shell | /bin/sh |
| Enter System Administrator User ID | 1001 (or default value) |
| Enter ID of User Group (sapsys) | 79 (or default value) |
| Enter System Database User (SYSTEM) Password | <SYSTEM DB User Password> |
| Confirm System Database User (SYSTEM) Password | <SYSTEM DB User Password> |
| Restart system after machine reboot? | n |
| Do you want to continue? | y |

**IMPORTANT**: SAP HANA のインストールに失敗した場合は、必要なすべての依存パッケージ (例： libatomic / libatomic1、compat-sap-c++、libtool-ltdl / libltdl7 など) がインストールされていることを確認し、インストールを再試行してください。

6. インストールが完了したら、node-a で次のコマンドを実行して、HDB00 インスタンスが正常に実行されていることを確認します。

```
[root@node-a ~]# su - spsadm -c "sapcontrol -nr 00 -function GetProcessList"
04.03.2021 23:42:13
GetProcessList
OK
name, description, dispstatus, textstatus, starttime, elapsedtime, pid
hdbdaemon, HDB Daemon, GREEN, Running, 2021 03 04 23:38:10, 0:04:03, 23804
```

```
hdbcompileserver, HDB Compileserver, GREEN, Running, 2021 03 04 23:38:41, 0:0
3:32, 24044
hdbnameserver, HDB Nameserver, GREEN, Running, 2021 03 04 23:38:11, 0:04:02, 2
3822
hdbpreprocessor, HDB Preprocessor, GREEN, Running, 2021 03 04 23:38:41, 0:03:3
2, 24047
hdbwebdispatcher, HDB Web Dispatcher, GREEN, Running, 2021 03 04 23:39:24, 0:0
2:49, 24454
hdbindexserver, HDB Indexserver-SPS, GREEN, Running, 2021 03 04 23:38:41, 0:0
3:32, 24093
hdbxsengine, HDB XSEngine-SPS, GREEN, Running, 2021 03 04 23:38:41, 0:03:32, 2
4096
```

## node-b に SAP HANA をインストールする

1. 「node-a」を「node-b」に置き換えて、前セクションのインストール手順を node-b で繰り返します。

> **IMPORTANT:** パラメーター（spsadm ユーザーのユーザー ID と sapsys グループのグループ ID を含む）は、両方のインストールで全く同じである必要があります。

2. インストールが完了したら、node-b で次のコマンドを実行して、HDB00 インスタンスが正常に実行されていることを確認します。

```
[root@node-b ~]# su - spsadm -c "sapcontrol -nr 00 -function GetProcessList"
04.03.2021 23:44:09
GetProcessList
OK
name, description, dispstatus, textstatus, starttime, elapsedtime, pid
hdbdaemon, HDB Daemon, GREEN, Running, 2021 03 04 23:38:16, 0:05:53, 10785
hdbcompileserver, HDB Compileserver, GREEN, Running, 2021 03 04 23:38:46, 0:0
5:23, 11111
hdbnameserver, HDB Nameserver, GREEN, Running, 2021 03 04 23:38:16, 0:05:53, 1
0804
hdbpreprocessor, HDB Preprocessor, GREEN, Running, 2021 03 04 23:38:46, 0:05:2
3, 11114
hdbwebdispatcher, HDB Web Dispatcher, GREEN, Running, 2021 03 04 23:39:29, 0:0
4:40, 11438
hdbindexserver, HDB Indexserver-SPS, GREEN, Running, 2021 03 04 23:38:47, 0:0
5:22, 11159
hdbxsengine, HDB XSEngine-SPS, GREEN, Running, 2021 03 04 23:38:47, 0:05:22, 1
1162
```

## 両方のノードで SYSTEMDB および SPS データベースをバッ

## クアップする

node-a と node-b の両方で次のコマンドを実行して、SYSTEMDB データベースと SPS テナントデータベースのバックアップを作成し、 を SYSTEM データベースユーザーのパスワードに置き換えます。

```
# su - spsadm -c "hdbsql -i 00 -u SYSTEM -p <SYSTEM User Password> -d SystemD
B \"BACKUP DATA USING FILE ('/hana/shared/SPS/HDB00')\""
0 rows affected (overall time 7390.598 msec; server time 7388.081 msec)

# su - spsadm -c "hdbsql -i 00 -u SYSTEM -p <SYSTEM User Password> -d SPS \"BA
CKUP DATA USING FILE ('/hana/shared/SPS/HDB00')\""
0 rows affected (overall time 5104.380 msec; server time 5102.842 msec)
```

# PKI ssfs KEY および DAT ファイルをコピーする

1. 次のコマンドを実行して、node-b の HDB00 インスタンスを停止します。

```
[root@node-b ~]# su - spsadm -c "sapcontrol -nr 00 -function StopSystem HDB"
04.03.2021 23:55:10
StopSystem
OK
```

2. node-b で HDB00 インスタンスが正常に停止したことを確認します。

```
[root@node-b ~]# su - spsadm -c "sapcontrol -nr 00 -function GetProcessList"
04.03.2021 23:55:51
GetProcessList
OK
name, description, dispstatus, textstatus, starttime, elapsedtime, pid
hdbdaemon, HDB Daemon, GRAY, Stopped, , , 10785
```

3. 次のコマンドを実行して、/usr/sap/SPS/SYS/global/security/rsecssfs/data/SSFS_SPS.DAT および /usr/sap/SPS/SYS/global/security/rsecssfs/key/SSFS_SPS.KEY ファイルを node-a から node-b にコピーします。

```
[root@node-a ~]# scp -r /usr/sap/SPS/SYS/global/security/rsecssfs/ root@node-
b:/usr/sap/SPS/SYS/global/security/
SSFS_SPS.DAT                  100% 2960     2.8MB/s   00:00
SSFS_SPS.KEY                  100%  187   190.9KB/s   00:00
```

# SAP HANA システム レプリケーションを設定する

1. node-a で次のコマンドを実行して、サイト名 SiteA を使用したシステムレプリケーションを有効にします。.

```
[root@node-a ~]# su - spsadm -c "hdbnsutil -sr_enable --name=SiteA"
nameserver is active, proceeding ...
successfully enabled system as system replication source site
done.
```

2. 次のコマンドを実行して、サイト名 SiteB を使用して node-b をセカンダリーシステムレプリケーションサイトとして登録します。

```
[root@node-b ~]# su - spsadm -c "hdbnsutil -sr_register --remoteHost=node-a --
remoteInstance=00 --replicationMode=sync --operationMode=logreplay --name=Site
B"
adding site ...
nameserver node-a:30001 not responding.
collecting information ...
updating local ini files ...
done.
```

3. 次のコマンドを実行して、node-b で HDB00 インスタンスを起動します。

```
[root@node-b ~]# su - spsadm -c "sapcontrol -nr 00 -function StartSystem HDB"
05.03.2021 00:11:51
StartSystem
OK
```

4. 次のコマンドを実行して、node-b で HDB00 インスタンスが正常に実行されていること、および node-a と node-b でレプリケーションモードとオペレーションモードがそれぞれ primary/primary と sync/logreplay であることを確認します。

```
[root@node-b ~]# su - spsadm -c "sapcontrol -nr 00 -function GetProcessList"
05.03.2021 00:13:36
GetProcessList
OK
name, description, dispstatus, textstatus, starttime, elapsedtime, pid
hdbdaemon, HDB Daemon, GREEN, Running, 2021 03 05 00:11:52, 0:01:44, 15816
hdbcompileserver, HDB Compileserver, GREEN, Running, 2021 03 05 00:11:57, 0:0
1:39, 15940
hdbindexserver, HDB Indexserver-SPS, GREEN, Running, 2021 03 05 00:11:57, 0:0
1:39, 15997
hdbnameserver, HDB Nameserver, GREEN, Running, 2021 03 05 00:11:52, 0:01:44, 1
5835
hdbpreprocessor, HDB Preprocessor, GREEN, Running, 2021 03 05 00:11:57, 0:01:3
9, 15943
hdbwebdispatcher, HDB Web Dispatcher, GREEN, Running, 2021 03 05 00:12:02, 0:0
1:34, 16217
hdbxsengine, HDB XSEngine-SPS, GREEN, Running, 2021 03 05 00:11:57, 0:01:39, 1
```

```
6000


[root@node-b ~]# su - spsadm -c "hdbnsutil -sr_state"
System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~

online: true

mode: sync
operation mode: logreplay
site id: 2
site name: SiteB

is source system: false
is secondary/consumer system: true
has secondaries/consumers attached: false
is a takeover active: false
active primary site: 1

primary masters: node-a

Host Mappings:
~~~~~~~~~~~~~~

node-b -> [SiteB] node-b
node-b -> [SiteA] node-a

Site Mappings:
~~~~~~~~~~~~~~
SiteA (primary/primary)
    |---SiteB (sync/logreplay)

Tier of SiteA: 1
Tier of SiteB: 2

Replication mode of SiteA: primary
Replication mode of SiteB: sync

Operation mode of SiteA: primary
Operation mode of SiteB: logreplay

Mapping: SiteA -> SiteB
done.
```

HDB00 インスタンスが node-a と node-b で実行され、両方のノードが SAP HANA システムレプリケーションに登録されたので、LifeKeeper で SAP HANA リソースを作成する準備が整いました。

# 11.2.7.5.6.4. LifeKeeper SAP HANA リソースを作成する

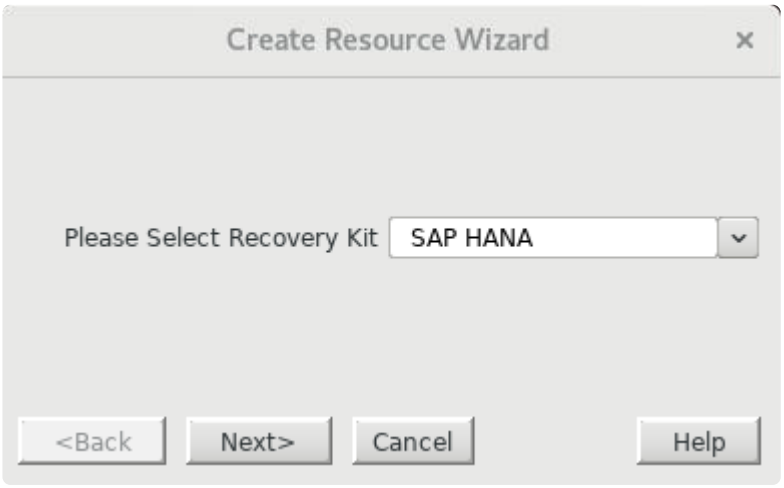
HDB00 インスタンスを保護する LifeKeeper リソースを作成するために必要な手順は、クラウドプラットフォームによって異なります。お使いのクラウドプラットフォームに対応するセクションに記載されている手順に従ってください。

- AWS/Azure – LifeKeeper SAP HANA リソースを作成する
- Google Cloud – LifeKeeper SAP HANA リソースを作成する

# 11.2.7.5.6.4.1. AWS/Azure – LifeKeeper SAP HANA リソースを作成する

**Note:** このセクションは、AWS または Microsoft Azure へのデプロイメントの場合に実施してください。Google Cloud へのデプロイについては、 Google Cloud – LifeKeeper SAP HANA リソースを作成する を参照してください。

1. LifeKeeper GUI で アイコンをクリックして、 **Create Resource Wizard** を開きます。「SAP HANA」Recovery Kit を選択します。

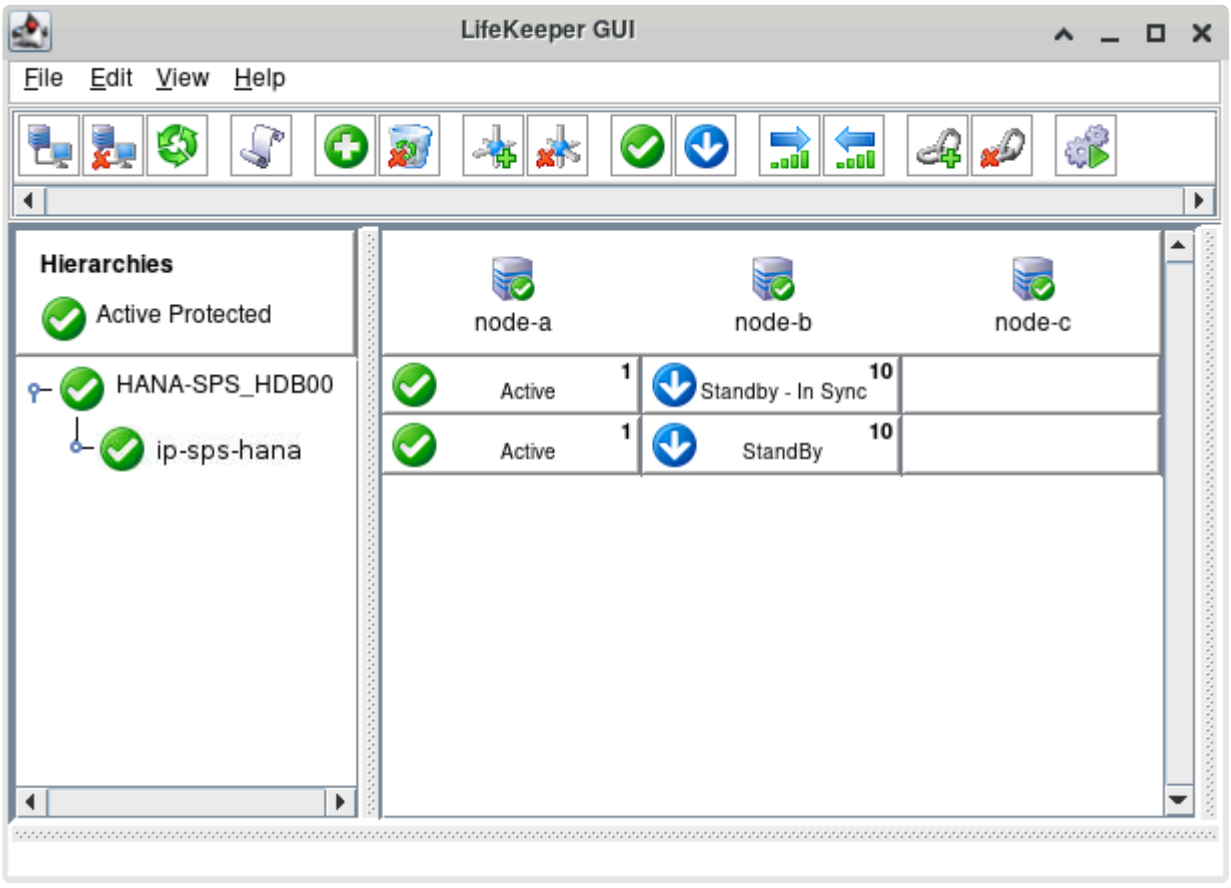



2. 以下の値を入力して LifeKeeper リソース（**HANA-SPS_HDB00**）を作成および拡張し、node-a および node-b の HDB00 データベースインスタンスを保護します。 作成されたリソースは、Witness ノードである node-c に拡張しないでください。 アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent |
| Server | node-a |
| SAP HANA SID | SPS |
| SAP HANA Instance for SPS | HDB00 |
| IP child resource | ip-sps-hana |
| Enable/Disable Local Recovery | Enable |
| HANA Tag | HANA-SPS_HDB00 |

| **Pre-Extend Wizard** | |
|---|---|
| Target Server | node-b |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend database/hana Resource Hierarchy Wizard** | |
| Root Tag | HANA-SPS_HDB00 |
| Enable/Disable Local Recovery | Enable |

**HANA-SPS_HDB00** リソースが正常に作成され、node-b に拡張されると、LifeKeeper リソースパネルは次のようになります。

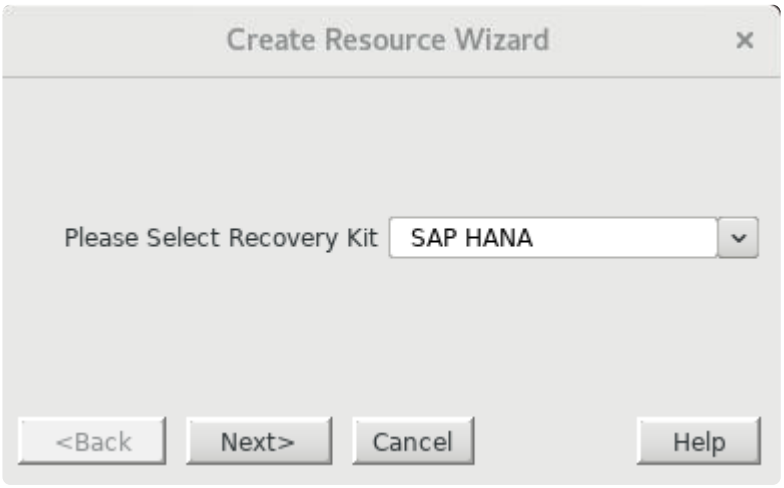



SAP HANA データベースインスタンス HDB00 は、LifeKeeper によって正常に保護されました。

# 11.2.7.5.6.4.2. Google Cloud – LifeKeeper SAP HANA リソースを作成する

**Note:** このセクションは、Google Cloud でのデプロイの場合の手順です。AWS または Microsoft Azure でのデプロイについては、AWS/Azure – LifeKeeper SAP HANA リソースを作成する を参照してください。

1. LifeKeeper GUI で アイコンをクリックして、 **Create Resource Wizard** を開きます。「SAP HANA」Recovery Kit を選択します。

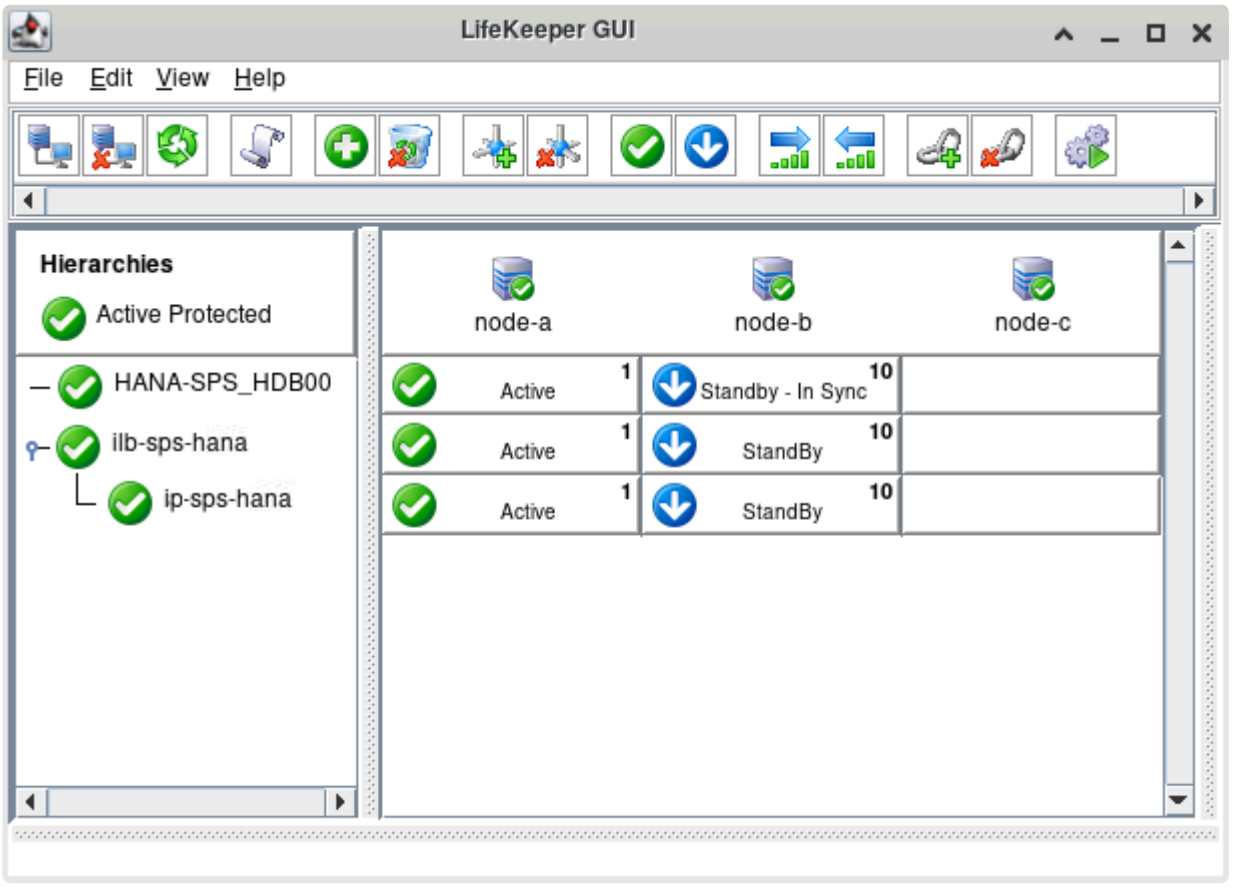



2. 以下の値を入力して LifeKeeper リソース（**HANA-SPS_HDB00**）を作成および拡張し、node-a および node-b の HDB00 データベースインスタンスを保護します。 作成されたリソースは、Witness ノードである node-c に拡張しないでください。 アイコンは、デフォルトのオプションが選択されていることを示します。

| Field | Value |
|---|---|
| **Create Resource Wizard** | |
| Switchback Type | intelligent |
| Server | node-a |
| SAP HANA SID | SPS |
| SAP HANA Instance for SPS | HDB00 |
| IP child resource | none |
| Enable/Disable Local Recovery | Enable |
| HANA Tag | HANA-SPS_HDB00 |

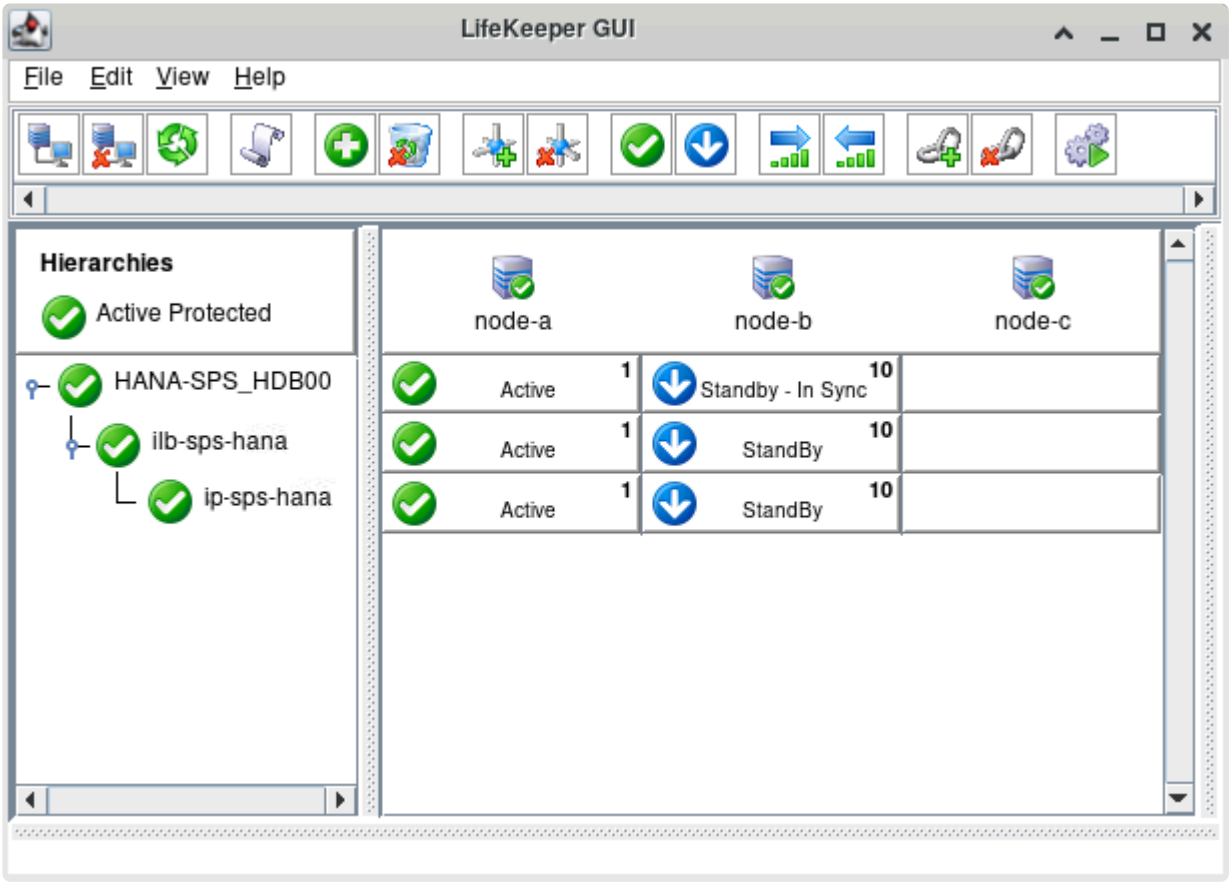
| **Pre-Extend Wizard** | |
|---|---|
| Target Server | node-b |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |
| **Extend gen/app Resource Hierarchy Wizard** | |
| Root Tag | HANA-SPS_HDB00 |
| Enable/Disable Local Recovery | Enable |

**HANA-SPS_HDB00** リソースが正常に作成され、node-b に拡張されると、LifeKeeper リソースパネルは次のようになります。




3. node-a の **HANA-SPS_HDB00** リソースを右クリックし、**[Create Dependency...]** をクリックします。子リソースタグに **ilb-sps-hana** を指定し、**[Create Dependency]** をクリックします。

   依存関係が作成されると、LifeKeeper GUI は次の図のようになります。

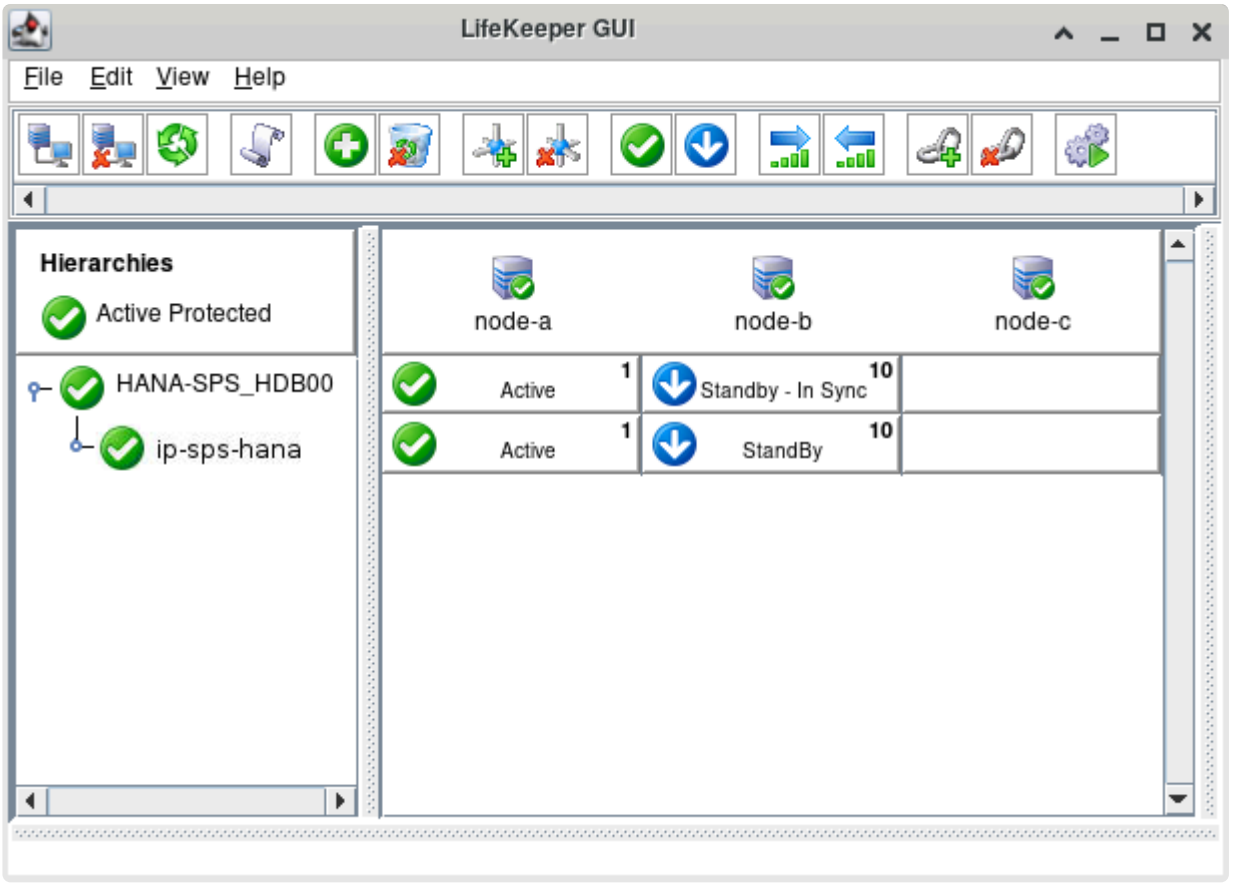



SAP HANA データベースインスタンス HDB00 は、LifeKeeper によって正常に保護されました。

# 11.2.7.5.6.5. スイッチオーバーとフェイルオーバーのテスト

このセクションでは、基本的なテストを実行して、スイッチオーバーとフェイルオーバーでの **HANA-SPS_HDB00** リソース階層の予想される動作を確認します。

1. **HANA-SPS_HDB00** リソースの状態が現在、node-a で「Active」、node-b で「Standby – In-Sync」であることを確認します。




2. Node-a と node-b で以下のコマンドを実行し、両ノードで HDB00 インスタンスが正常に動作していること、node-a と node-b がともに SAP HANA システムレプリケーションに適切に登録されていること、仮想ホスト名 sps-hana を介してプライマリーデータベースに接続できることを確認します。レプリケーションモードとオペレーションモードは、node-a で primary/primary、node-b で sync/logreplay となります。

**Note**: Google Cloud のデプロイでは、仮想ホスト名 sps-hana を使用して node-b から node-a のプライマリーデータベースインスタンスに接続できるようにするために、Google Cloud – 内部ロードバランサーを使用する の「**IP 転送を無効にする**」セクションに記載されている手順を実行する必要があります。

```
[root@node-a ~]# su - spsadm -c "sapcontrol -nr 00 -function GetProcessList"
04.03.2021 21:31:50
```

```
GetProcessList
OK
name, description, dispstatus, textstatus, starttime, elapsedtime, pid
hdbdaemon, HDB Daemon, GREEN, Running, 2021 03 04 19:44:27, 1:47:23, 19236
hdbcompileserver, HDB Compileserver, GREEN, Running, 2021 03 04 19:45:46, 1:4
6:04, 19618
hdbindexserver, HDB Indexserver-SPS, GREEN, Running, 2021 03 04 19:45:47, 1:4
6:03, 19666
hdbnameserver, HDB Nameserver, GREEN, Running, 2021 03 04 19:44:27, 1:47:23, 1
9254
hdbpreprocessor, HDB Preprocessor, GREEN, Running, 2021 03 04 19:45:46, 1:46:0
4, 19621
hdbwebdispatcher, HDB Web Dispatcher, GREEN, Running, 2021 03 04 19:46:15, 1:4
5:35, 20241
hdbxsengine, HDB XSEngine-SPS, GREEN, Running, 2021 03 04 19:45:47, 1:46:03, 1
9669
```

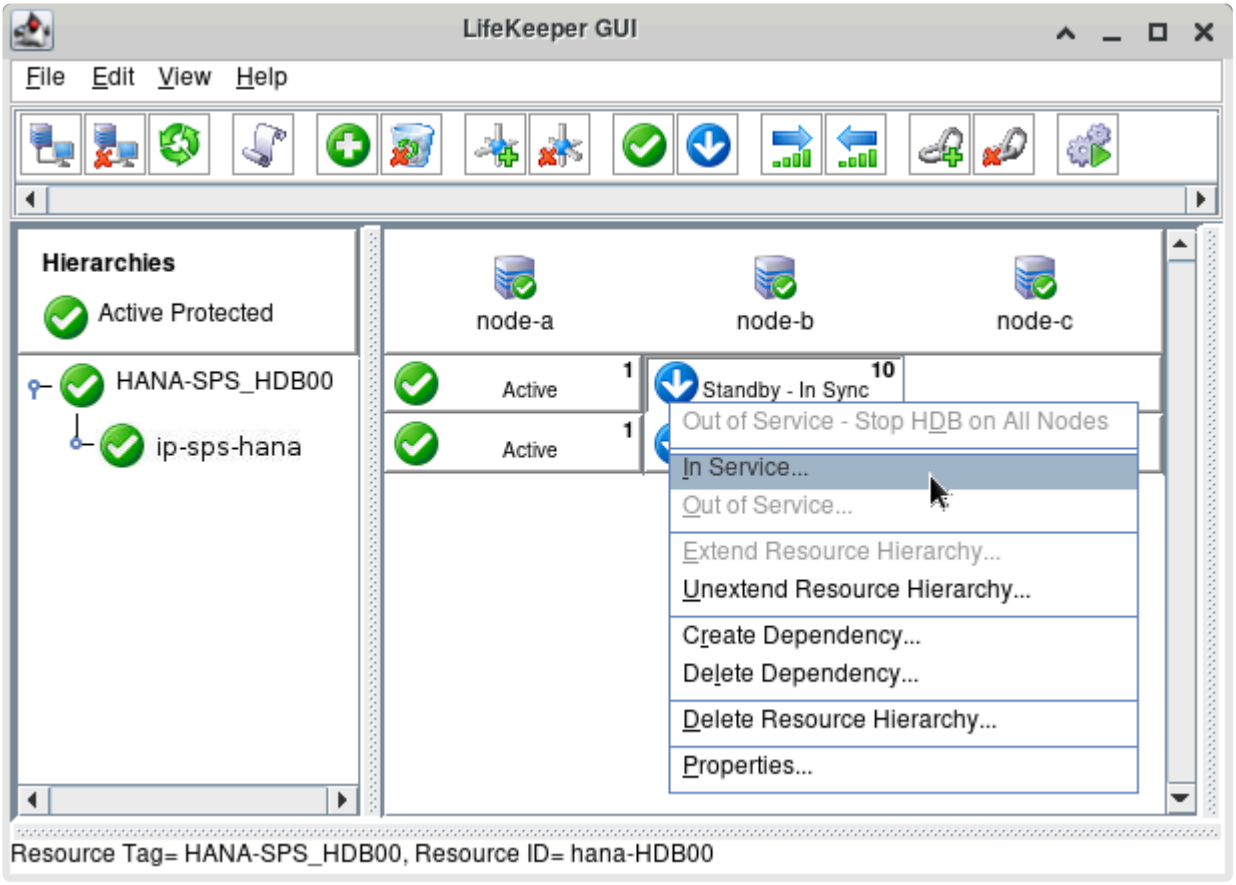
```
[root@node-a ~]# su - spsadm -c "hdbnsutil -sr_state"
System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~

online: true

mode: primary
operation mode: primary
site id: 1
site name: SiteA

is source system: true
is secondary/consumer system: false
has secondaries/consumers attached: true
is a takeover active: false

Host Mappings:
~~~~~~~~~~~~~~

node-a -> [SiteB] node-b
node-a -> [SiteA] node-a

Site Mappings:
~~~~~~~~~~~~~~
SiteA (primary/primary)
    |---SiteB (sync/logreplay)

Tier of SiteA: 1
Tier of SiteB: 2
```

```
Replication mode of SiteA: primary
Replication mode of SiteB: sync

Operation mode of SiteA: primary
Operation mode of SiteB: logreplay

Mapping: SiteA -> SiteB
done.
```

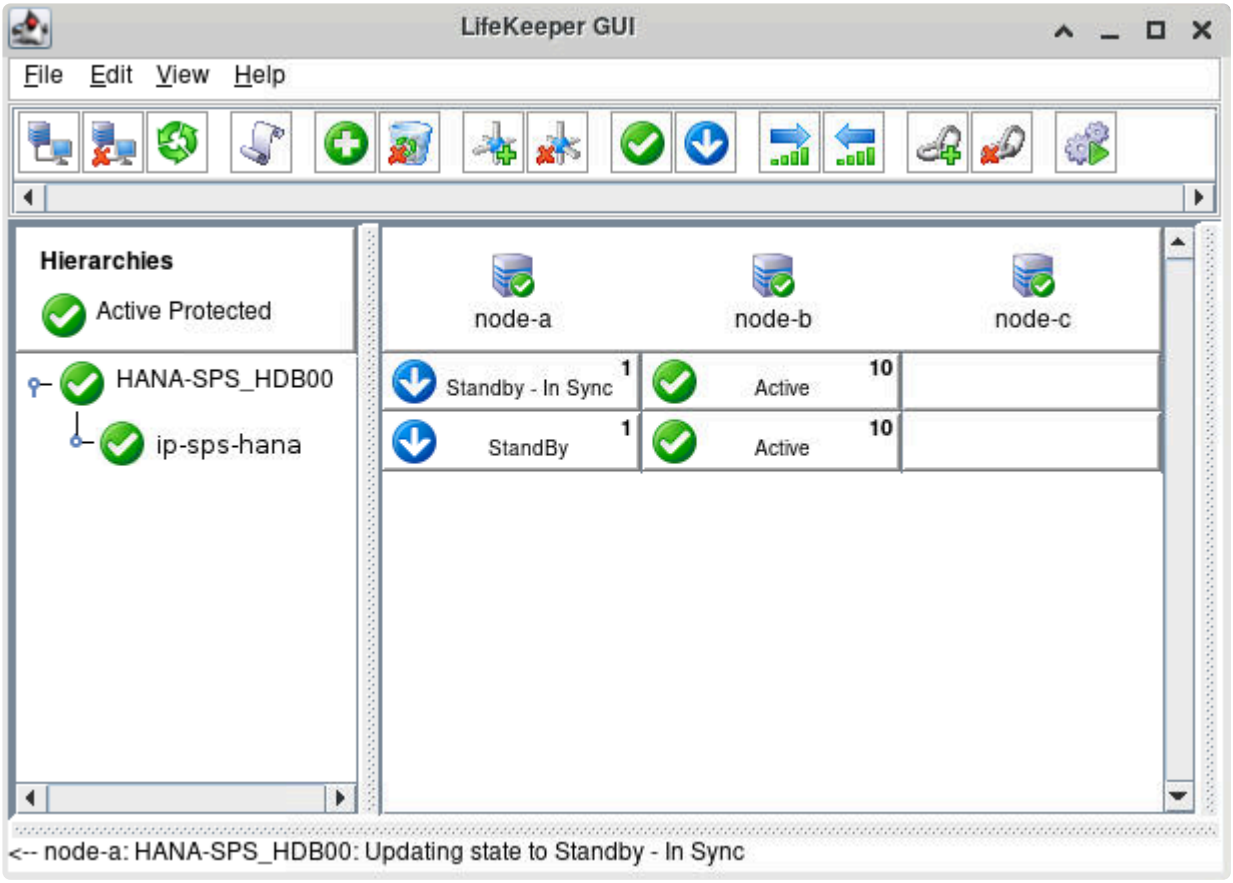
```
[root@node-b ~]# su - spsadm -c "sapcontrol -nr 00 -function GetProcessList"
04.03.2021 21:38:10
GetProcessList
OK
name, description, dispstatus, textstatus, starttime, elapsedtime, pid
hdbdaemon, HDB Daemon, GREEN, Running, 2021 03 04 19:48:50, 1:49:20, 13383
hdbcompileserver, HDB Compileserver, GREEN, Running, 2021 03 04 19:49:00, 1:4
9:10, 13747
hdbindexserver, HDB Indexserver-SPS, GREEN, Running, 2021 03 04 19:49:00, 1:4
9:10, 13794
hdbnameserver, HDB Nameserver, GREEN, Running, 2021 03 04 19:48:51, 1:49:19, 1
3401
hdbpreprocessor, HDB Preprocessor, GREEN, Running, 2021 03 04 19:49:00, 1:49:1
0, 13750
hdbwebdispatcher, HDB Web Dispatcher, GREEN, Running, 2021 03 04 19:49:42, 1:4
8:28, 15167
hdbxsengine, HDB XSEngine-SPS, GREEN, Running, 2021 03 04 19:49:00, 1:49:10, 1
3797
```

```
[root@node-b ~]# su - spsadm -c "hdbsql -n sps-hana -i 00 -u SYSTEM -p <SYSTE
M user password> -d SPS '\s'"
host           : sps-hana:30013
sid            : SPS
dbname         : SPS
user           : SYSTEM
kernel version: 2.00.046.00.1581325702
SQLDBC version:        libSQLDBCHDB 2.04.182.1579711187
autocommit     : ON
locale         : en_US.UTF-8
input encoding: UTF8
sql port       : node-a:30015
```

3. node-b の **HANA-SPS_HDB00** リソースを右クリックし、 **[In-Service...]** を選択して、SAP HANA リソース階層のスイッチオーバーを実行します。 **[In Service]** をクリックして、スイッチオーバーを開始します。




スイッチオーバーが完了し、リソースの状態が node-a で「Standby – In Sync」に、node-b で「Active」に切り替わるまで待ちます。なお、スタンバイ データベース インスタンスの稼働後、リソースの状態が「Standby – HDB Running」から「Standby – In-Sync」に移行するまでに最大2分かかる場合があることに注意してください。




4. Node-a と node-b で以下のコマンドを実行し、両ノードで HDB00 インスタンスが正常に動作していること、node-a と node-b がともに SAP HANA システムレプリケーションに適切に登録されていること、仮想ホスト名 sps-hana を介してプライマリーデータベースに接続できることを検証します。レプリケーションモードが sync/logreplay に、オペレーションモードが primary/primary にそれぞれ変更されました。

```
[root@node-a ~]# su - spsadm -c "sapcontrol -nr 00 -function GetProcessList"
04.03.2021 21:50:20
GetProcessList
OK
name, description, dispstatus, textstatus, starttime, elapsedtime, pid
hdbdaemon, HDB Daemon, GREEN, Running, 2021 03 04 21:43:51, 0:06:29, 156845
hdbcompileserver, HDB Compileserver, GREEN, Running, 2021 03 04 21:43:58, 0:0
6:22, 156970
hdbindexserver, HDB Indexserver-SPS, GREEN, Running, 2021 03 04 21:43:58, 0:0
6:22, 157101
hdbnameserver, HDB Nameserver, GREEN, Running, 2021 03 04 21:43:52, 0:06:28, 1
56864
hdbpreprocessor, HDB Preprocessor, GREEN, Running, 2021 03 04 21:43:58, 0:06:2
2, 156973
hdbwebdispatcher, HDB Web Dispatcher, GREEN, Running, 2021 03 04 21:44:36, 0:0
5:44, 157429
hdbxsengine, HDB XSEngine-SPS, GREEN, Running, 2021 03 04 21:43:58, 0:06:22, 1
57104
```

```
[root@node-a ~]# su - spsadm -c "hdbsql -n sps-hana -i 00 -u SYSTEM -p <SYSTE
M user password> -d SPS '\s'"
host           : sps-hana:30013
sid            : SPS
dbname         : SPS
user           : SYSTEM
kernel version: 2.00.046.00.1581325702
SQLDBC version:        libSQLDBCHDB 2.04.182.1579711187
autocommit     : ON
locale         : en_US.UTF-8
input encoding: UTF8
sql port       : node-b:30015
```

```
[root@node-b ~]# su - spsadm -c "sapcontrol -nr 00 -function GetProcessList"
04.03.2021 21:52:36
GetProcessList
OK
name, description, dispstatus, textstatus, starttime, elapsedtime, pid
hdbdaemon, HDB Daemon, GREEN, Running, 2021 03 04 19:48:50, 2:03:46, 13383
hdbcompileserver, HDB Compileserver, GREEN, Running, 2021 03 04 19:49:00, 2:0
3:36, 13747
hdbindexserver, HDB Indexserver-SPS, GREEN, Running, 2021 03 04 19:49:00, 2:0
3:36, 13794
hdbnameserver, HDB Nameserver, GREEN, Running, 2021 03 04 19:48:51, 2:03:45, 1
3401
hdbpreprocessor, HDB Preprocessor, GREEN, Running, 2021 03 04 19:49:00, 2:03:3
6, 13750
hdbwebdispatcher, HDB Web Dispatcher, GREEN, Running, 2021 03 04 19:49:42, 2:0
2:54, 15167
hdbxsengine, HDB XSEngine-SPS, GREEN, Running, 2021 03 04 19:49:00, 2:03:36, 1
3797
```

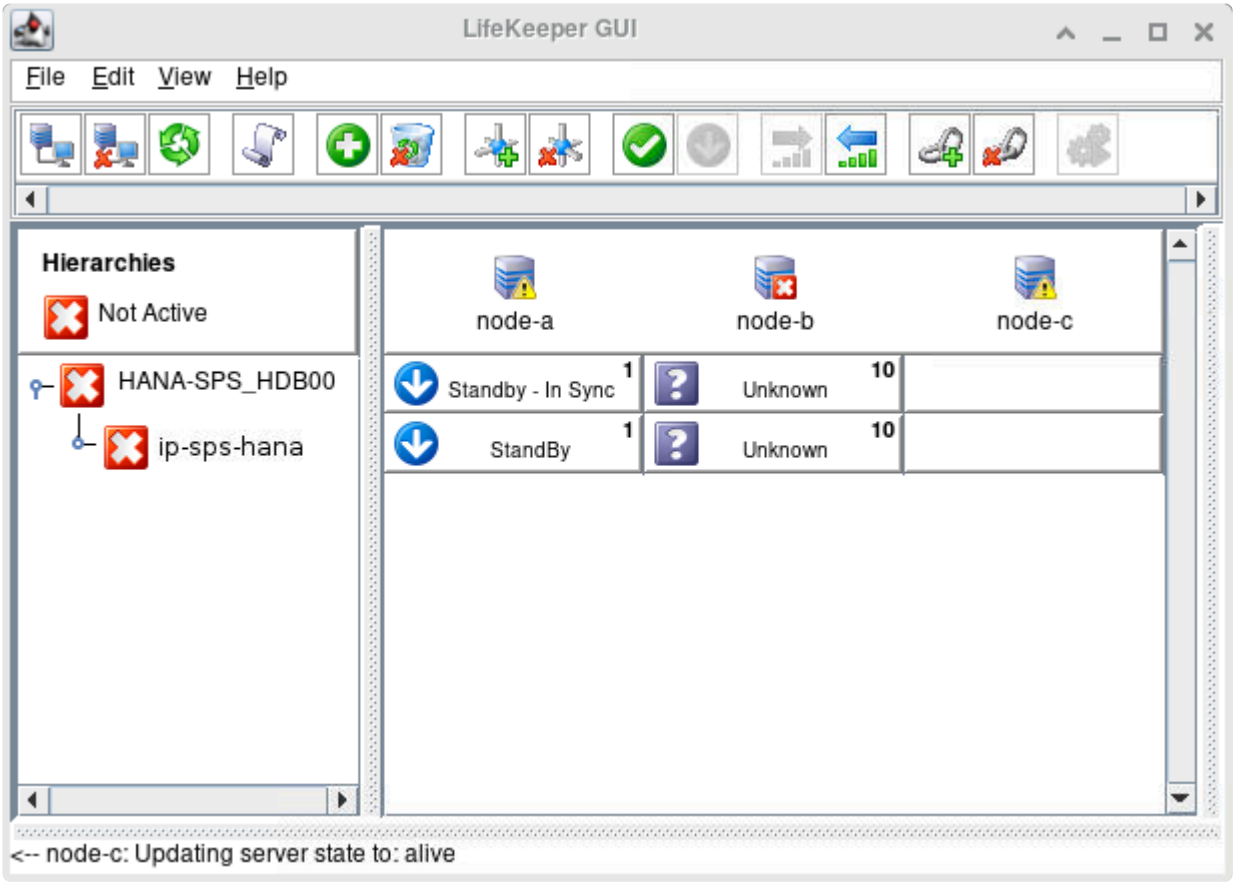
```
[root@node-b ~]# su - spsadm -c "hdbnsutil -sr_state"
System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~

online: true

mode: primary
operation mode: primary
site id: 2
site name: SiteB

is source system: true
is secondary/consumer system: false
```

```
has secondaries/consumers attached: true
is a takeover active: false

Host Mappings:
~~~~~~~~~~~~~~

node-b -> [SiteB] node-b
node-b -> [SiteA] node-a

Site Mappings:
~~~~~~~~~~~~~~
SiteB (primary/primary)
    |---SiteA (sync/logreplay)

Tier of SiteB: 1
Tier of SiteA: 2

Replication mode of SiteB: primary
Replication mode of SiteA: sync

Operation mode of SiteB: primary
Operation mode of SiteA: logreplay

Mapping: SiteB -> SiteA
done.
```

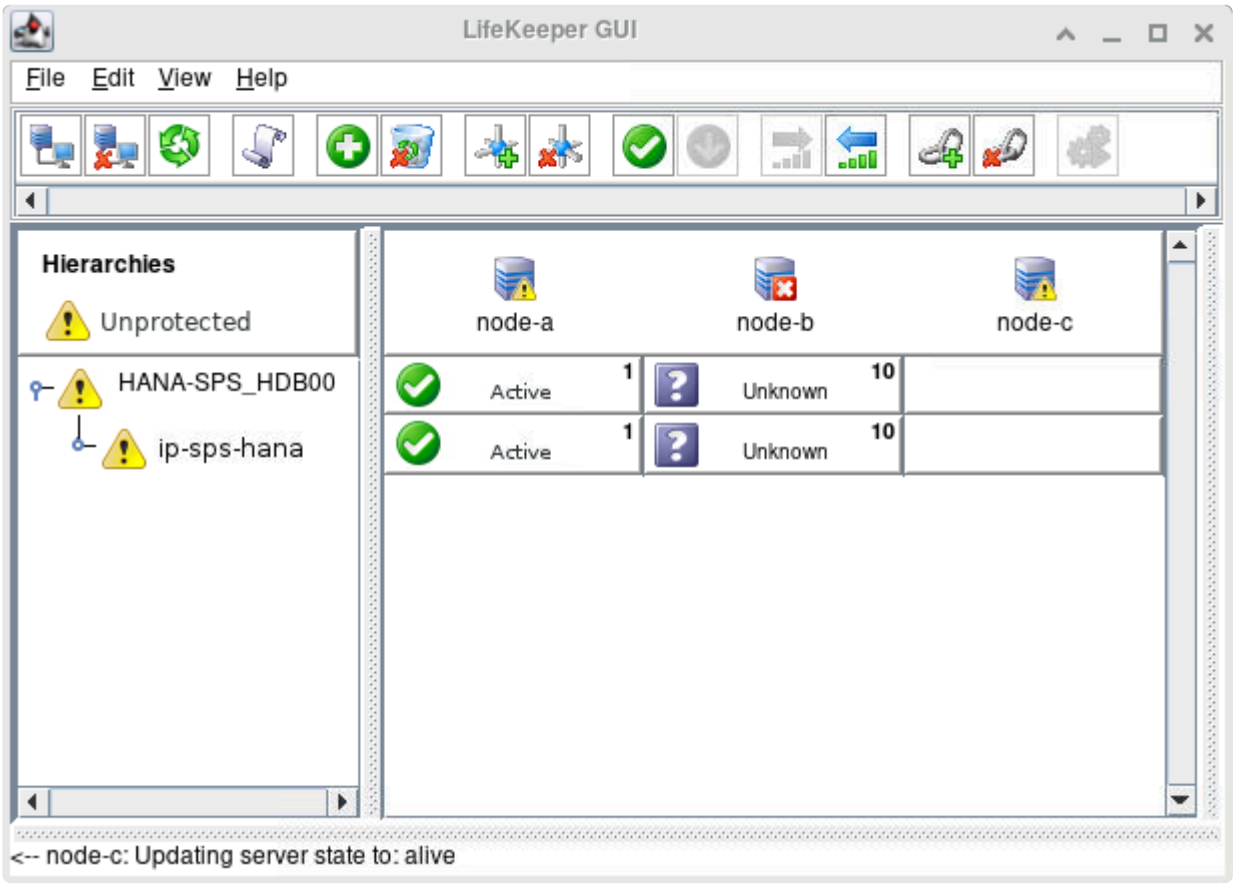
5. 次のコマンドを実行して、node-b を強制的に再起動します。

```
[root@node-b ~]# echo b > /proc/sysrq-trigger
```

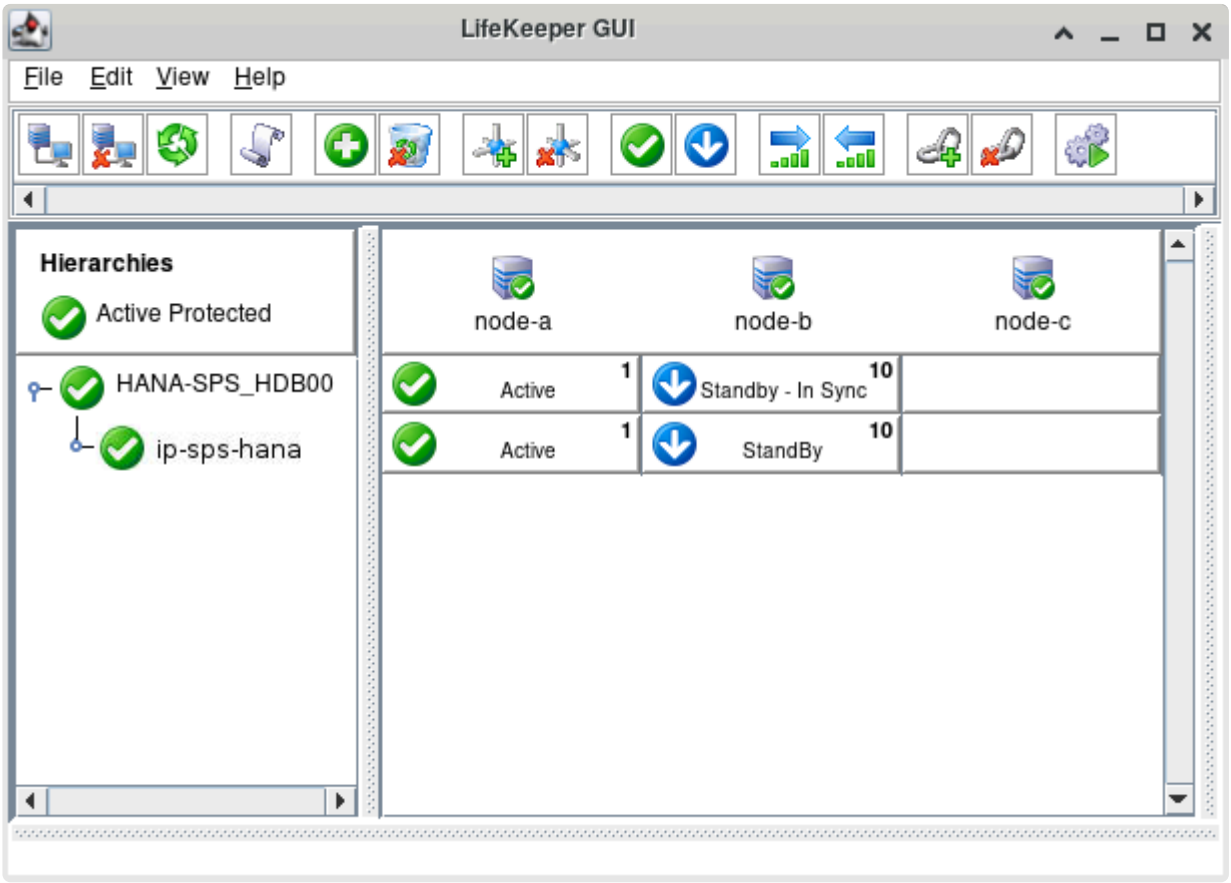
Node-b の電源がオフになっていることを LifeKeeper が検出すると、LifeKeeper GUI で node-b のステータスが「Unknown」に更新されます。




この時点で、LifeKeeper は **HANA-SPS_HDB00** リソース階層のnode-a へのフェイルオーバーを自動的に開始します。



Node-b がオンラインに戻ると、LifeKeeper はそれを SAP HANA システムレプリケーションのセカンダリーレプリケーションサイトとして自動的に再登録し、セカンダリーデータベースインスタンスを再起動します。このプロセスが完了するまでに数分かかる場合があります。このプロセスが完了すると、リソースの状態は node-a で「Active」に戻り、node-b で「Standby – In-Sync」に戻ります。




6. ステップ2のコマンドを再度実行し、両ノードでデータベースが期待通りに動作していることを確認します。

これで、SAP HANA リソース階層の基本的なスイッチオーバーおよびフェイルオーバー機能の確認が完了しました。

# 11.2.7.6. 共通タスク

以下のトピックには、異なるリソースに共通するタスクが含まれています。

- ノードでデータストレージが利用可能かどうかを確認する方法
- 他ノードへのデータストレージのスイッチオーバー

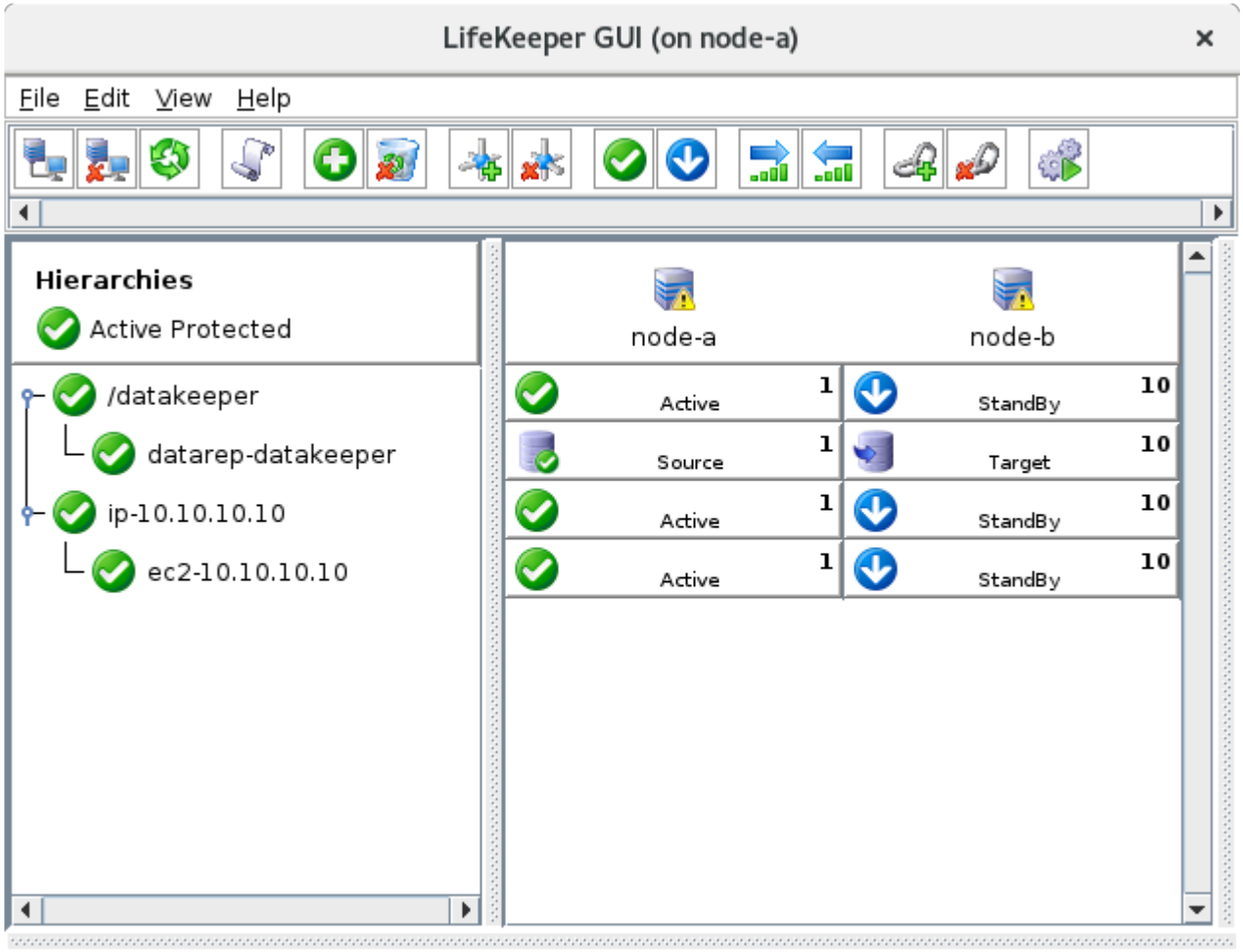
# 11.2.7.6.1. ノードでデータストレージが利用可能かどうかを確認する方法

このステップでは、ノード間でデータストレージを切り替えることになります。ここでは、あるノードでデータストレージ（本ガイドでは/datakeeperに配置）が利用可能かどうかを確認するための2つのオプションを紹介します。

- `df` コマンドの出力を表示します。

```
$ df
Filesystem     1K-blocks     Used Available Use% Mounted on
devtmpfs          916888        0    916888   0% /dev
tmpfs            7340032        0   7340032   0% /dev/shm
tmpfs             940184    17424    922760   2% /run
tmpfs             940184        0    940184   0% /sys/fs/cgroup
/dev/xvda2      31444972 17183608  14261364  55% /
tmpfs             188040        0    188040   0% /run/user/1000
/dev/md0        20960236  2510944  18449292  12% /datakeeper
```

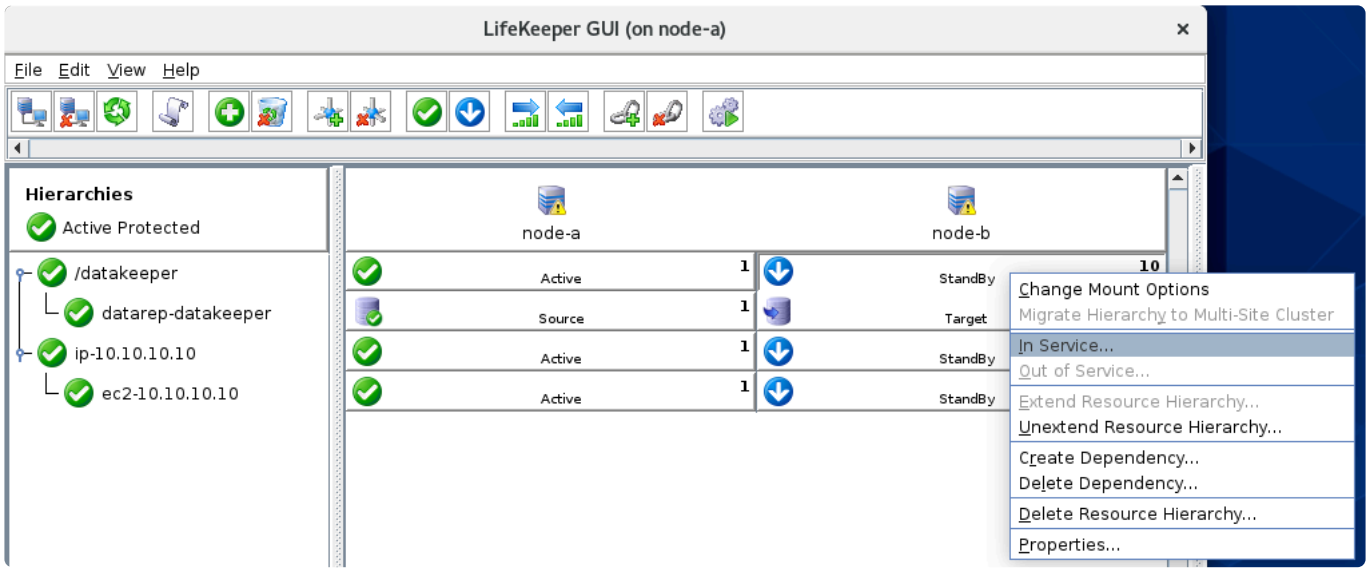
- LifeKeeper GUI で `datarep-datakeeper` リソースのステータスを確認します。ノードのステータスは「Source」、もう一方のノードは「Target」と表示されているはずです。



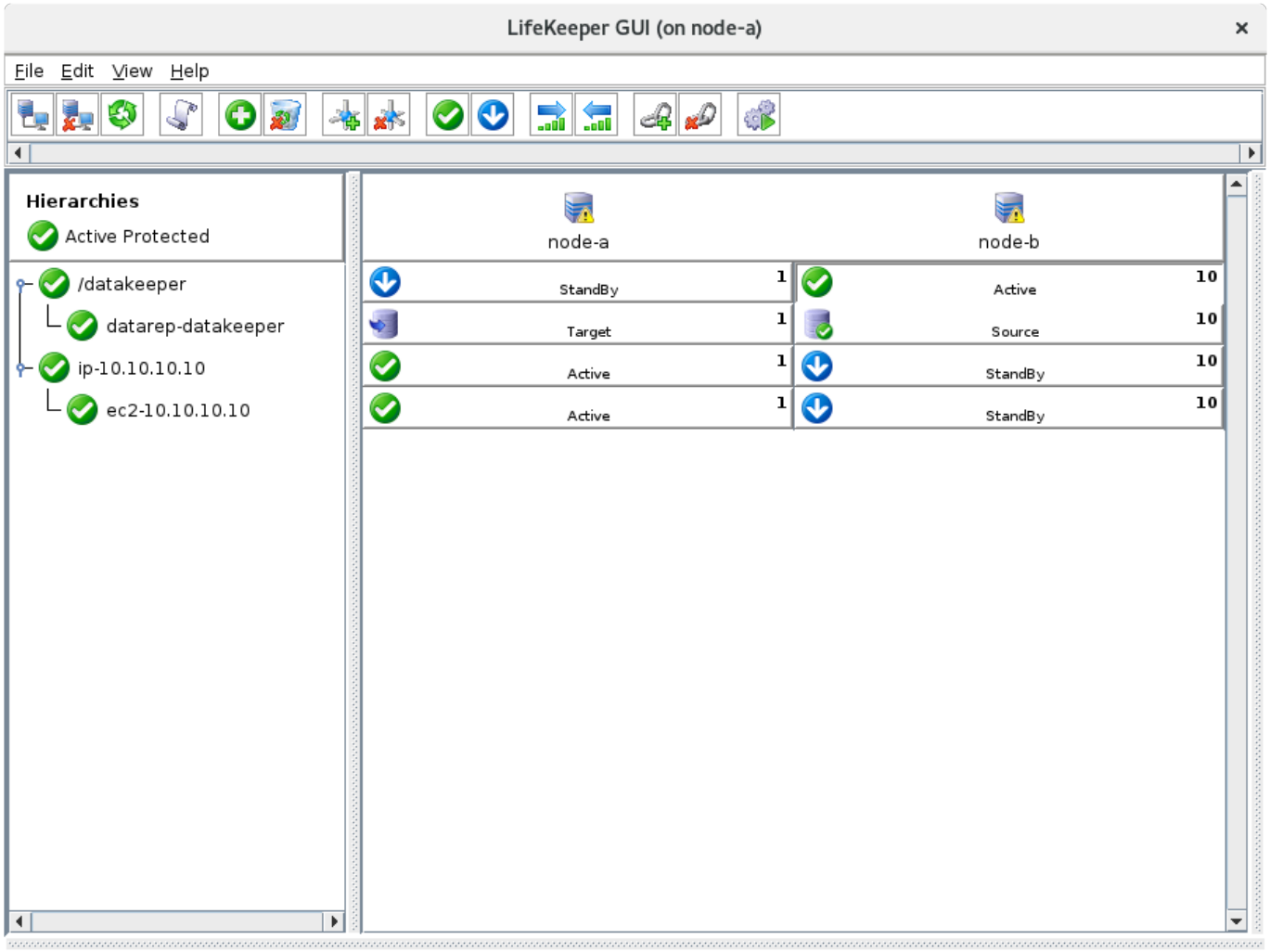

<-- node-b: datarep-datakeeper: Updating state to Target

# 11.2.7.6.2. 他ノードへのデータストレージのスイッチオーバー

LifeKeeper GUI を使用してディスクを他のノードにスイッチオーバーするには、スタンバイ ノードで /datakeeper リソースを選択し、 **[In Service...]** を選択します。




もう一方のノードで `datarep-datakeeper` リソースがアクティブになると、ソースラベルとターゲットラベルが切り替わります。




<-- node-a: datarep-datakeeper: Updating state to Target

# 12. Quick Start Guides

AWS Direct Connect Quick Start Guide

Microsoft Azure Quick Azure Guide

LifeKeeper for Linux in the AWS Cloud (SAP)

Connecting to a LifeKeeper Cluster using AWS Transit Gateway Quick Start Guide

AWS VPC Peering Connections Quick Start Guide

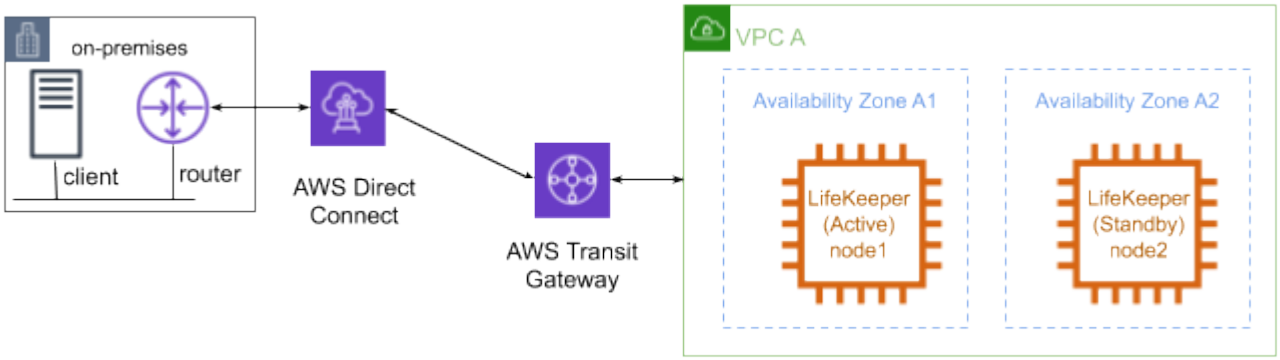
PostgreSQL Cluster with Shared Storage (iSCSI)

Apache/MySQL Cluster Using Both Shared and Replicated Storage

# 12.1. AWS Direct Connect クイックスタートガイド

## 目的

AWS Transit Gateway のリリースにより、AWS Direct Connect を利用したオンプレミス環境（下図の On-premises）から AWS Transit Gateway を経由して、下図の VPC A 内に配置された HA クラスターノードへ接続する構成で、Recovery Kit for EC2 のルートテーブルシナリオが利用可能になりました。




本資料は、LifeKeeper for Linux で VPC 外から接続する構成を構築するための要件や基本操作を解説するものです。

なお、本資料は LifeKeeper や Amazon Web Service(以下 AWS)の基本的な設定や操作、技術的な詳細情報を解説するものではありません。 本構成の前提となる LifeKeeper や AWS に関する用語･操作･技術情報等につきましては、関連のマニュアルやユーザーサイト等であらかじめご確認ください。

> 注記:「Amazon Web Services」、「Powered by Amazon Web Services」のロゴ、「AWS」、「Amazon EC2」、「EC2」、「Amazon Elastic ComputeCloud」、「Amazon Virtual Private Cloud」、「AWS Direct Connect」、「AWS Identity and Access Management」、「AWS Transit Gateway」および「Amazon VPC」は、米国その他の国における Amazon.com, Inc. またはその関連会社の商標です。

# 12.1.1. 利用のための要件

本構成を利用するためには、環境を準備する段階で満たすべきいくつかの要件があります。以下に AWS 環境とその上に作成するインスタンスに関する要件をまとめます。

## AWS 環境の要件

サービスを提供するための基盤となる環境を AWS 上に作成します。本構成を利用するための要件は以下の通りです。

### Amazon Virtual Private Cloud (VPC)

- VPC を AWS 内に設定する必要があります。
- プライマリ用インスタンスを配置するサブネットとスタンバイ用インスタンスを配置するサブネットを、それぞれ異なる Availability Zone(AZ) に作成する必要があります。

### Amazon Elastic Compute Cloud (EC2)

- インスタンスが 2つ以上必要です。
- プライマリ用インスタンスとスタンバイ用インスタンスがそれぞれ異なる AZ で起動するように構成する必要があります。
- インスタンスは、Elastic Network Interface (ENI) に接続されます。
- インスタンスは、LifeKeeper のインストール要件を満たす必要があります。
- AWS Command Line Interface (AWS CLI) を全ての EC2 インスタンスにインストールする必要があります。インストール方法は、「AWS Command Line Interface のインストール 」を参照してください。
- https を使用して Amazon EC2 Web サービスエンドポイントの URL (EC2 URL) に、および http を使用して Amazon EC2 メタデータの URL (http://169.254.169.254/) にアクセス可能でなければなりません。

### AWS Identity and Access Management (IAM)

LifeKeeper が AWS を操作するために、以下のアクセス権を持った IAM ユーザーもしくは IAM ロールが必要です。EC2 インスタンスの root ユーザーからアクセスできるように EC2 の IAM ロール を設定するか、AWS CLI の設定 を適切に行ってください。

- ec2:DisassociateAddress
- ec2:DescribeAddresses
- ec2:AssociateAddress

- ec2:DescribeRouteTables
- ec2:ReplaceRoute

### AWS Transit Gateway

- クラスターノードを配置した VPC と、クライアントを配置したオンプレミス環境とは、Virtual Private Gateway で接続するのではなく、AWS Transit Gateway を経由して接続する必要があります。
- AWS Transit Gateway を作成する際に Default route table association と Default route table propagation を有効にしてください。
- VPC との接続は、Transit Gateway Attachment を作成することで行ってください。
- AWS Direct Connect との接続は Direct Connect Gateway の Gateway association の設定で、作成した AWS Transit Gateway を選択することで行なってください。 この際に、 Allowed prefixes には、クラスターノードを配置した VPC のネットワークアドレスと、仮想 IP アドレスの両方を設定してください。

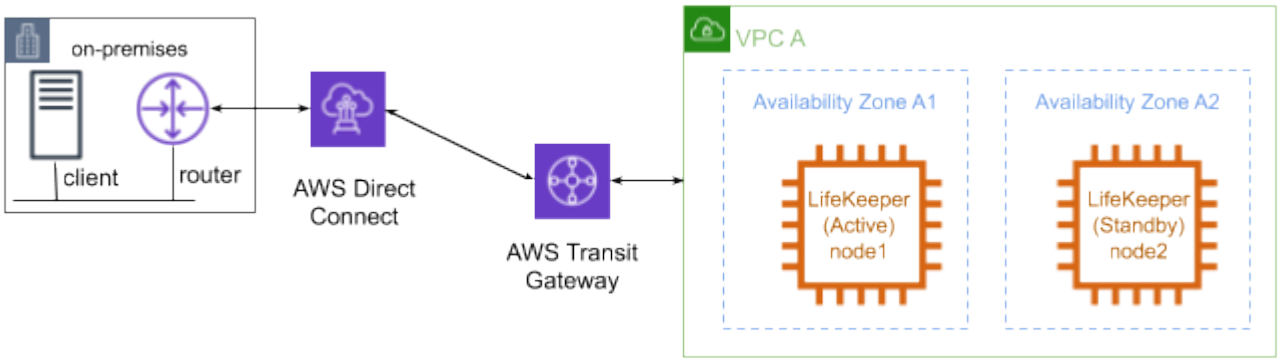
## LifeKeeper ソフトウェアの要件

各サーバーに同じバージョンの LifeKeeper ソフトウェアとパッチをインストールする必要があります。本構成で必要な Application Recovery Kit (ARK) は以下の通りです。具体的な LifeKeeper の要件については、SPS for Linux テクニカルドキュメンテーション および SPS for Linux リリースノート を参照してください。

- LifeKeeper IP Recovery Kit
- LifeKeeper Recovery Kit for EC2

# 12.1.2. AWS Direct Connect 構築手順

以下の構成図の様な環境を構築するための一般的な手順を解説します。



# 12.1.2.1. AWS Direct Connect 準備

「AWS Direct Connect の要件」を満たす環境を構築してください。それぞれのインスタンスに LifeKeeper をインストールして、node1/node2 間にコミュニケーションパスを作成してください。

# 12.1.2.2. Direct Connect リソースの作成

## IP リソース作成

仮想 IP リソースを作成します。IP リソースのアドレスは、VPC で管理している CIDR ブロック外である必要があります。

## EC2 リソース作成

EC2リソースを作成します。リソース作成時に要求される IP リソースは、上述の「**IP リソース作成**」で作成したリソースを指定してください。リソース作成時に要求される EC2 リソースタイプは Route Table (Backend Cluster) を指定してください。

## 保護するサービスのリソース作成

保護するサービスのためのリソースを作成してください。リソース作成に IP リソースが要求される場合は、上述の「**IP リソース作成**」で作成したリソースを指定してください。親リソースが保護するサービスのリソース、子リソースが EC2 リソースとなるようにリソースの依存関係を設定してください。

# 12.1.2.3. ルートテーブルの設定

ルートテーブルを下記のように設定してください。

- クラスターノードを配置した VPC もしくはサブネットのルートテーブルに、オンプレミス環境のネットワークへの経路情報を追加してください。

| 送信先 | ターゲット |
|---|---|
| オンプレミス環境のネットワークアドレス | 作成した Transit Gateway |

- Transit Gateway のルートテーブルに、仮想 IP アドレスへの経路情報を追加してください。

| 送信先 | ターゲット |
|---|---|
| 仮想 IP アドレス | クラスターノードを配置した VPC |

- オンプレミス環境のクライアントやルーターの経路情報を適切に設定し、クラスターノードを配置した VPC のネットワークアドレスや仮想 IP アドレス宛のパケットの送信先が Direct Connect となるようにしてください。

設定が完了したら、クライアントからクラスタサーバーのプライベートアドレスや仮想 IP アドレスにアクセスできることを確認してください。

# 12.1.3. 本構成における設定および運用上の留意点

## LifeKeeper I-O フェンシングの利用を検討してください

AWS 環境では共有ディスク環境が構成できないため、スプリットブレイン問題を防ぐために SCSI リザーブ方式を採ることができません。これを踏まえて、本構成ではより安全に運用できるように、LifeKeeper の I/O フェンシング機能の Quorum / Witness Server もしくは STONITH の利用をご検討ください。

特に、Quorum モードの TCP_REMOTE 設定を利用すれば、別途 Quorum サーバを立てずに I/Oフェンシング機能を実装できるため、クラウド環境においては利用しやすいと考えられます。利用方法については以下の URL をご確認ください。

- Quorom/Witness
- STONITH

## 既知の問題とトラブルシューティング

LifeKeeper for Linux v9.4.1 リリース時点での情報はありません。

# 12.2. Microsoft Azure クイックスタートガイド

本書は、SIOS Protection Suite for Linux（以下、LifeKeeper）を Microsoft Azure（以下、Azure）リソースマネージャー環境においてご利用頂く際の参考として頂くために弊社で実際に行った作業手順をベースとしてクイックスタートガイドとしてまとめたものです。本ガイドでは、Azure VM 上のクラスター環境の例として、以下の構成を作成する手順をご案内します。

- LifeKeeper クラスター（2ノードクラスター） ※マルチ NIC 構成
- DataKeeper レプリケーションによるデーターミラー型の共有データ領域
- IPリソース、Oracleリソース、LB Health Check リソース

> **免責条項:** 本ガイドでは Microsoft Azure 環境における一般的な SIOS Protection Suite for Linux 設定の概略を述べています。そのため、本ガイドで使用している設定値は、説明に必要な項目を抜粋して記載しています。また、記載のない項目は基本的にデフォルト値または任意の値を前提としています。実際には、より多くの Azure インフラストラクチャーオプションを利用した構成が可能です。本ガイドに記載の設定値は、実際の要件に合わせて、適宜置き換えてご利用ください。

## 将来における互換性

本文書に記載されております内容は、2021 年 8 月時点の LifeKeeper for Linux、および Microsoft Azure の仕様に基づいています。（スクリーンショットは2021年4月時点のものです）将来における Azure 及び LifeKeeper の仕様変更について保証するものではありません。必要に応じて最新のドキュメンテーションを参照し、適切な設定を行ってください。

## 用語について

本書で使用する用語について説明します。

- **内部負荷分散（Internal Load Balancing または ILB）** – Azure クラウドサービスまたは地域が限定された仮想ネットワークの内側に存在する仮想マシン間に負荷分散を実現する機能です。
- **[IP リソースが保護する VIP]** – LifeKeeper で作成した仮想IPアドレスです。
- **[ILB が設定する VIP]** – Azureで作成した Internal Load Balancing の仮想 IP アドレスです。
- **[Azure のプライベート IP アドレス]** – Azure の Virtual Network で使用するプライベートアドレスです。

# 12.2.1. Microsoft Azure の概要

Azure は、Microsoft が提供するパブリッククラウドサービスです。ハードウェア、ネットワーク、ディスクなどのストレージ、Windows Server や Linux などのサーバー OS、Web サーバーや RDBMS などのミドルウェア、グループウェアやサーバー・アプリケーション、そして .Net Framework などのアプリケーション実行環境を、ユーザーがオンデマンドでネットワークを経由して利用可能な IT の利用形態です。

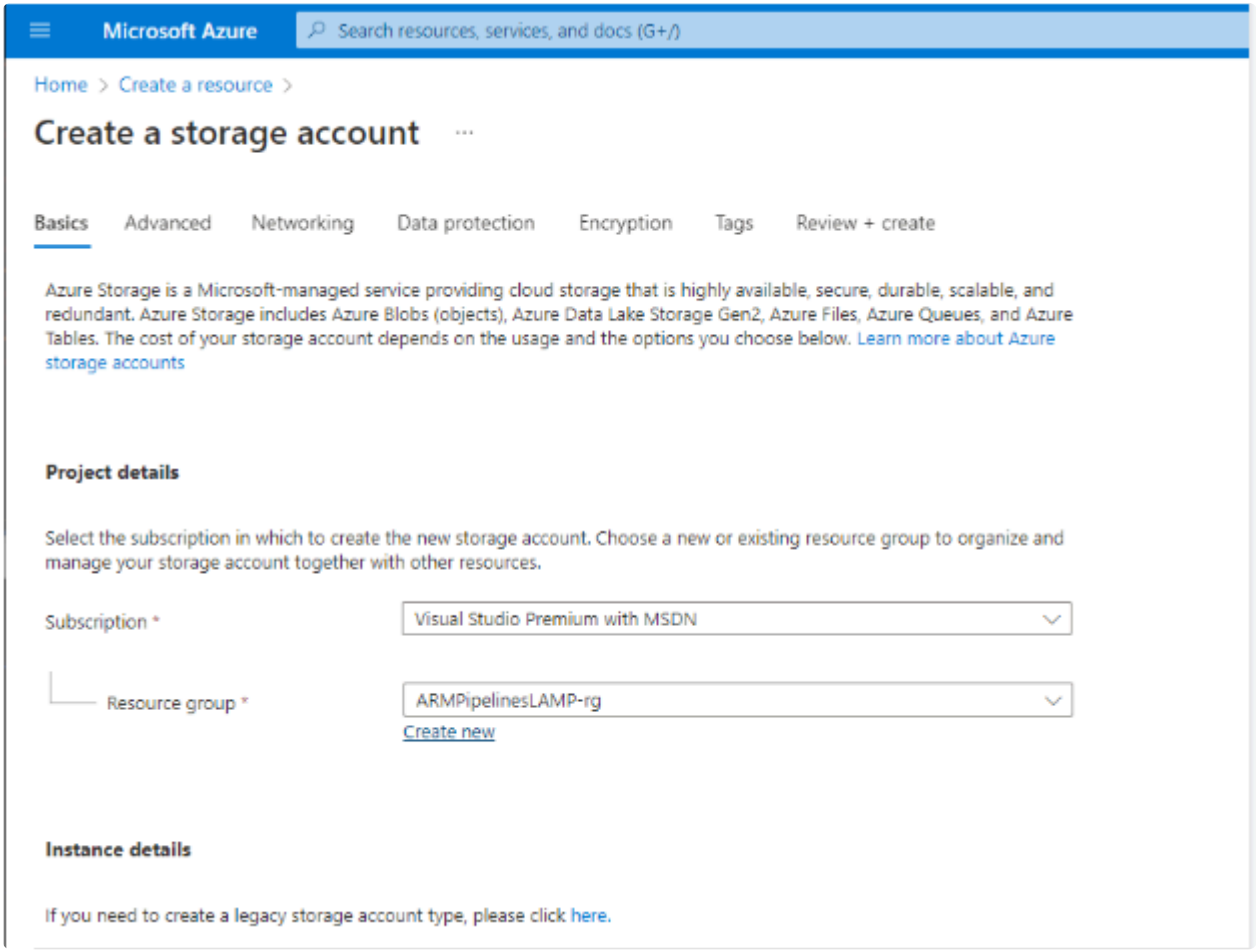
Microsoft が管理する世界中のデータセンターに、インターネット経由でサーバー・アプリケーションを簡単に作成し、展開、管理が行えます。サーバーやインフラを所有することなく、必要な時に必要な量のメモリやディスクといったコンピューティングリソースを利用することが出来ます。料金体系はリソース毎に分かれており、時間単位の従量課金での利用となります。Azure では、Web ブラウザベースの管理ポータルを提供しており、ユーザーは直感的に操作する事が可能です。

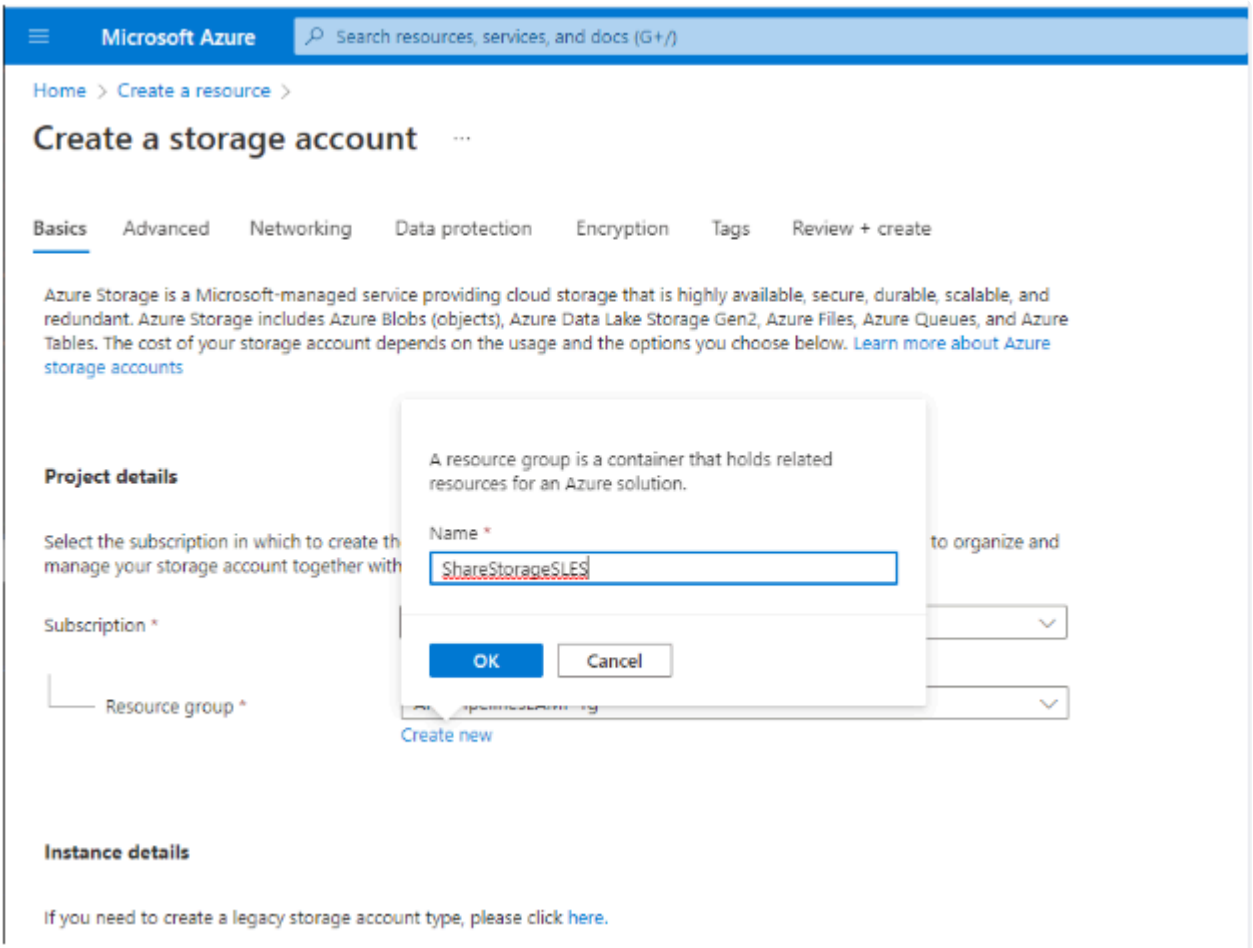
Azure では、仮想マシン(VM)、Web サイト、モバイルサービス、クラウドサービスの 4 つの形態でサービスを提供していますが、本ドキュメントでは、仮想マシンに対して LifeKeeper を導入し、可用性を高める手法について紹介いたします。

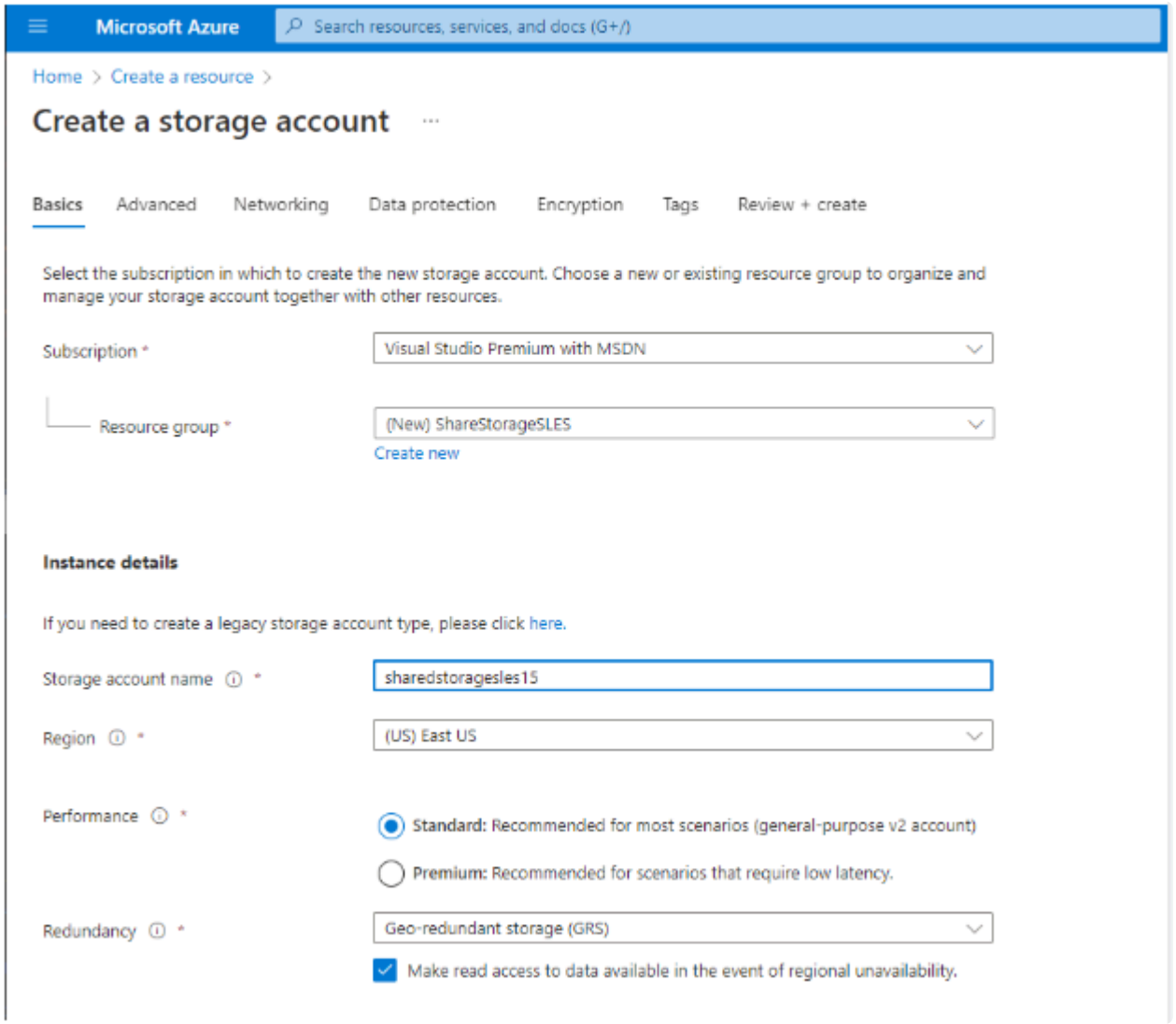
# 12.2.2. Azure Shared Storageのセットアップ方法

このドキュメントでは新しい仮想マシン（VM）に共有ディスクを作成し、アタッチする方法を説明します。既存のVMを使用する場合は、ステップ3「Create and attach a new disk」に進み、ステップ4で既存のVMに追加してください。新しいVMを追加する代わりに、既存のVMを編集します。ステップ7では“SCSI-3 Persistent Reservations (SCSI3) Recovery Kit” が必要です。必要に応じてインストーラーで更新して追加してください。

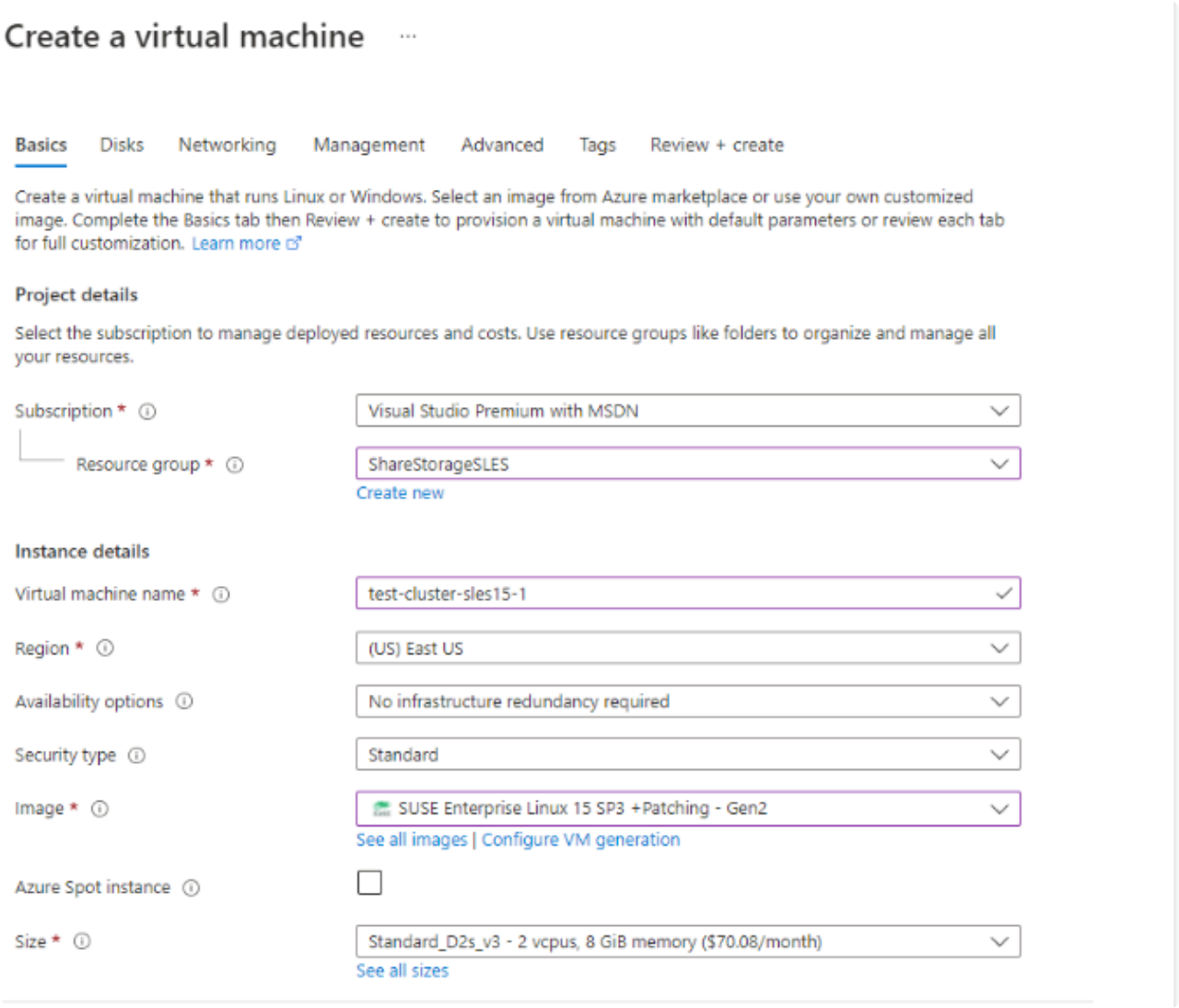
1. ストレージのアカウントを作成する。
    a. 新規リソースグループの作成



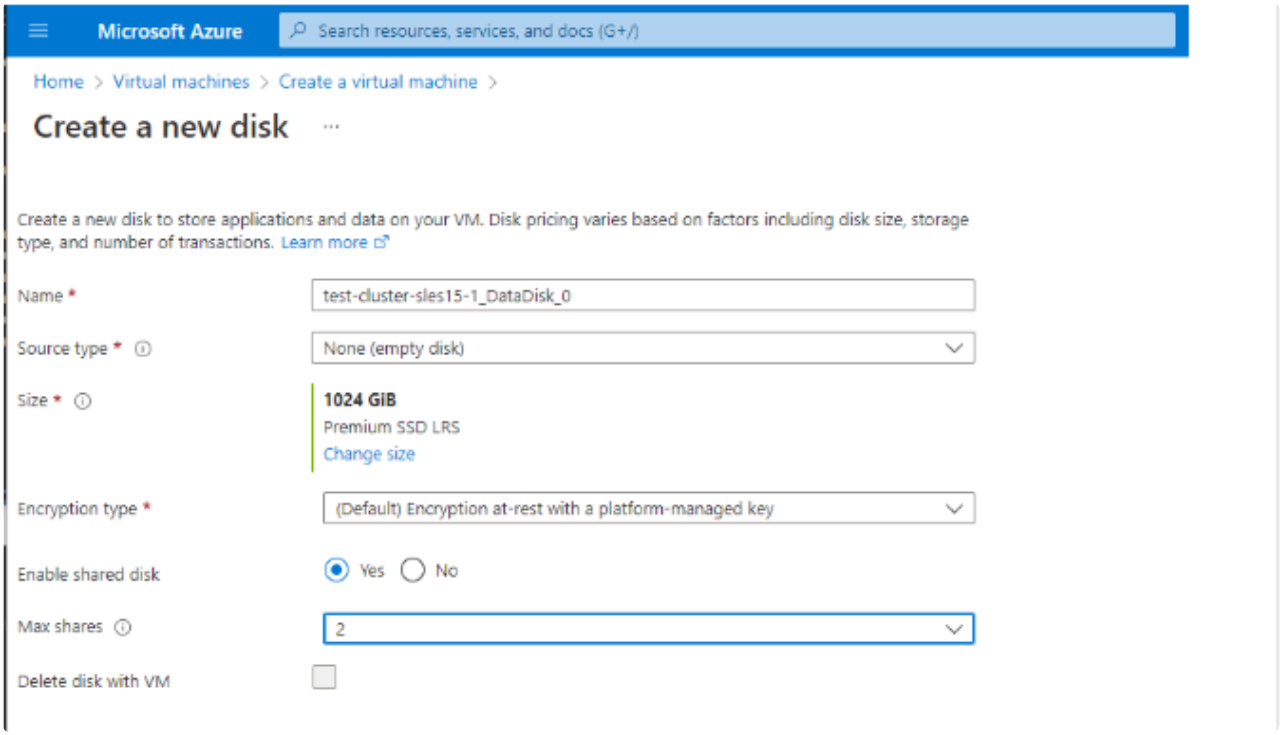

Microsoft Azure
Search resources, services, and docs (G+/)
Home > Create a resource >
Create a storage account
Basics
Advanced
Networking
Data protection
Encryption
Tags
Review + create
Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. Learn more about Azure storage accounts
Project details
Select the subscription in which to create th
to organize and
manage your storage account together with
Subscription *
Resource group *
Create new
A resource group is a container that holds related resources for an Azure solution.
Name *
ShareStorageSLES
OK
Cancel
Instance details
If you need to create a legacy storage account type, please click here.



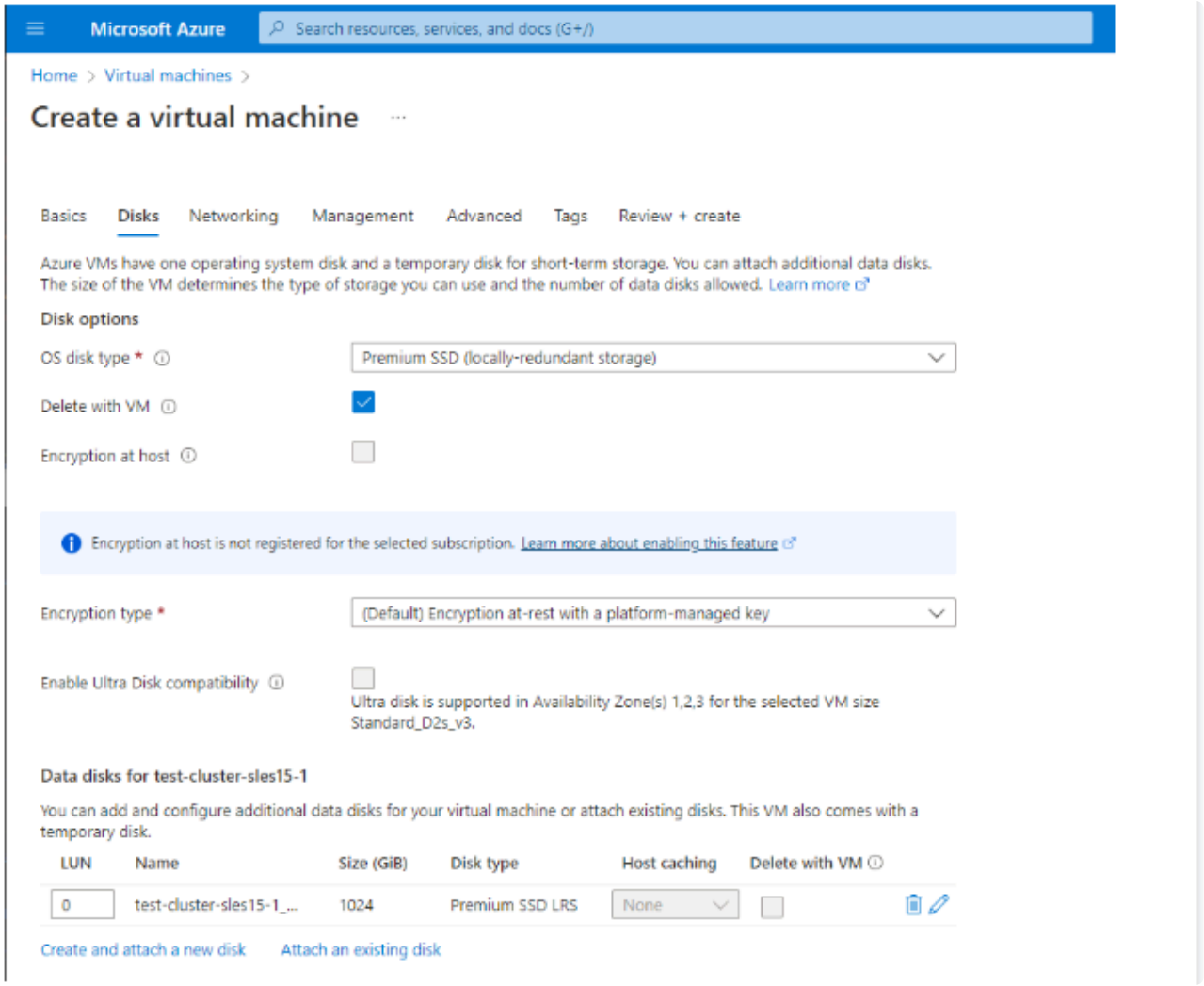

2. リソースグループにVMを作成します。（例では **ShareStorageSLES** というリソースグループに **test-cluster-sles15-1** というVMを作成しています）




3. **Create and attach a new disk** を選択して、共有する新しいディスクを作成します。

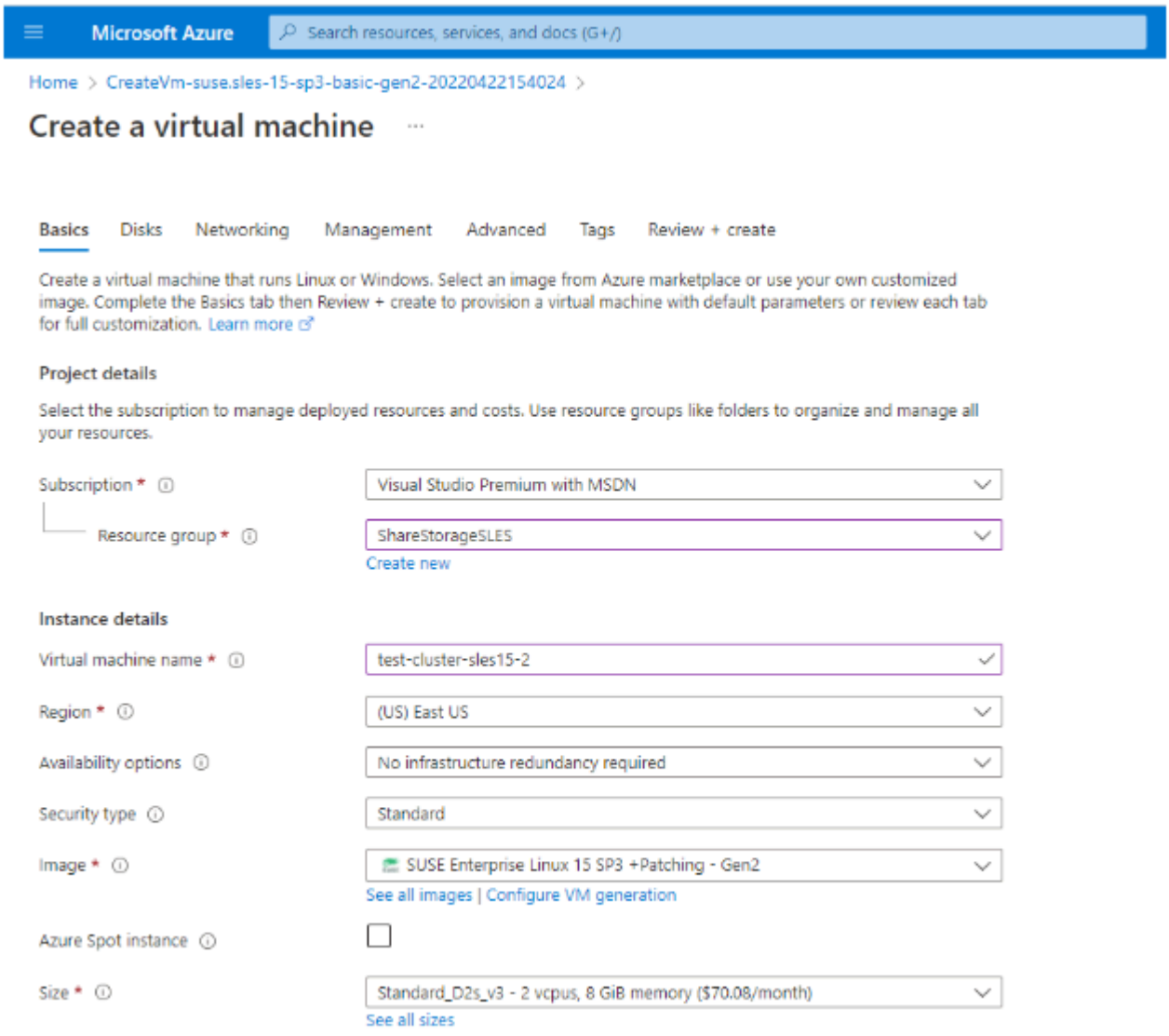


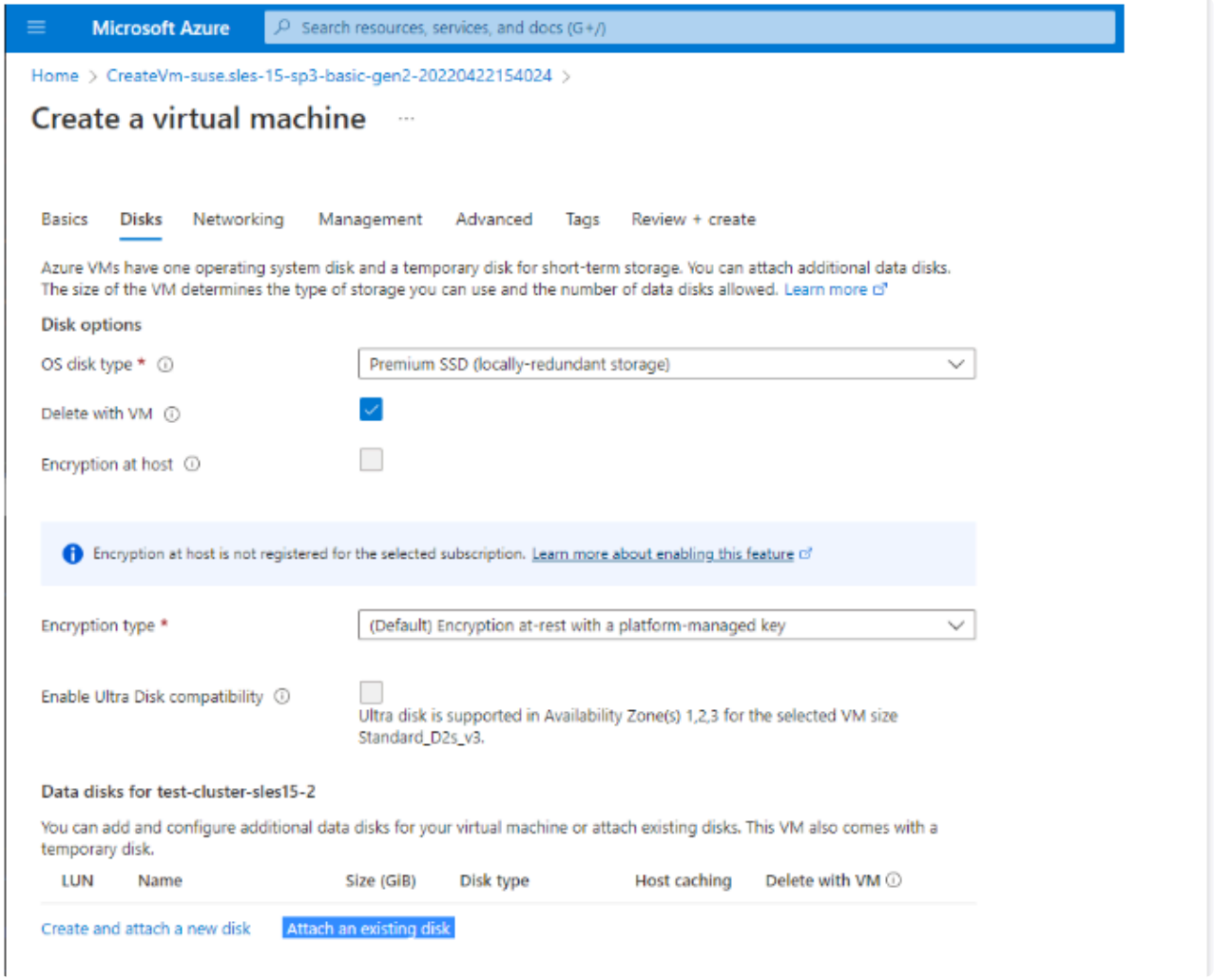
**Enable shared disk** を選択し **max shares** を指定します（2ノードクラスタのため **max share**は**2** を指定します 。）



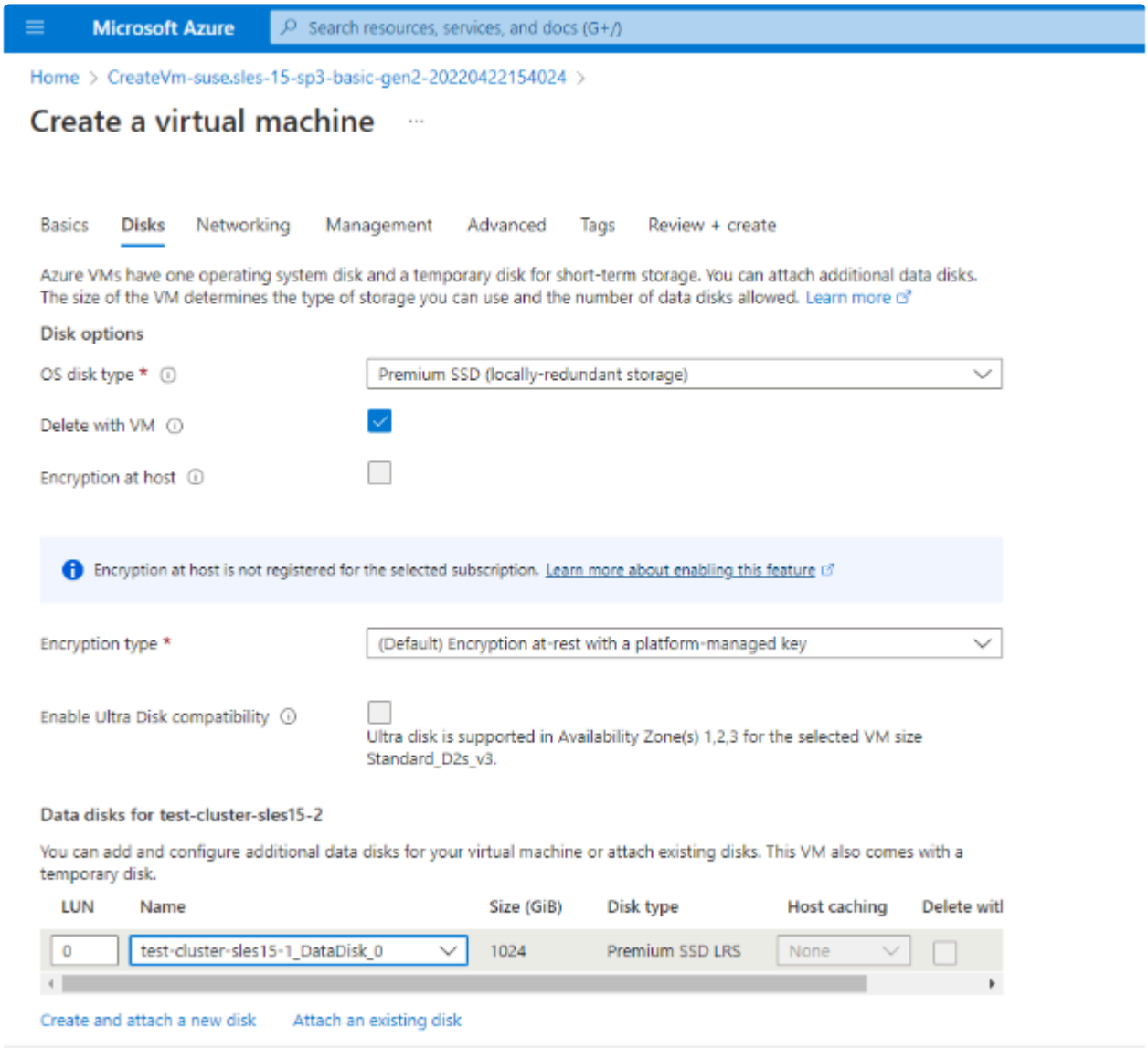

4. 同じリソースグループの同じゾーン に2つ目のVM(例では **test-cluster-sles15-2** )を作成します。Create and attach a new diskを選択するのではなく、 **Attach an existing disk**を選択します 。

> *注意：ホストキャッシュが「None」に設定されていることを確認します。デフォルトでは、「Read-Only」に設定されています。


Microsoft Azure
Search resources, services, and docs (G+/)
Home > CreateVm-suse.sles-15-sp3-basic-gen2-20220422154024 >
Create a virtual machine
Basics Disks Networking Management Advanced Tags Review + create
Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image. Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization. Learn more
Project details
Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.
Subscription * Visual Studio Premium with MSDN
Resource group * ShareStorageSLES
Create new
Instance details
Virtual machine name * test-cluster-sles15-2
Region * (US) East US
Availability options No infrastructure redundancy required
Security type Standard
Image * SUSE Enterprise Linux 15 SP3 +Patching - Gen2
See all images | Configure VM generation
Azure Spot instance
Size * Standard_D2s_v3 - 2 vcpus, 8 GiB memory ($70.08/month)
See all sizes




ステップ3で作成したディスクを選択してください。




5. プライマリノード(例では“test-cluster-sles15-1”)にSSHでログインし、その後ディスクのパーティション分割とフォーマットを行います。

```
ewilliams@dell-5579-ew:~$ ssh -i /home/ewilliams/test-cluster-ew.pem azureuser@20.232.194.160
The authenticity of host '20.232.194.160 (20.232.194.160)' can't be established.
ECDSA key fingerprint is SHA256:xX0p3mxrgx3vILpraiAktWCnDYgS6ZU5XnuCPxnxKzI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '20.232.194.160' (ECDSA) to the list of known hosts.
SUSE Linux Enterprise Server 15 SP3 x86_64 (64-bit)

As "root" (sudo or sudo -i) use the:
  - zypper command for package management
  - yast command for configuration management

If you are using extensions consider to enable the auto-update feature
of the extension agent and restarting the service. As root execute:
  - sed -i s/AutoUpdate.Enabled=n/AutoUpdate.Enabled=y/ /etc/waagent.conf
  - rcwaagent restart

Management and Config: https://www.suse.com/suse-in-the-cloud-basics
Documentation: https://www.suse.com/documentation/sles-15/
Community: https://community.suse.com/

Have a lot of fun...
azureuser@test-cluster-sles15-1:~>
```

6. LifeKeeper本体と仮ライセンスキーのダウンロード･転送します。

```
[ewilliams@fenric ~]$ scp -i /home/ewilliams/test-cluster-ew.pem /home/lk_linux/CD_IMAGES/9.6.2/18/sps/sps.img azureuser@20.232.194.160:/tmp
The authenticity of host '20.232.194.160 (20.232.194.160)' can't be established.
RSA key fingerprint is 8e:27:4e:20:b9:53:ba:12:b9:51:7a:ad:1e:9e:a2:48.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '20.232.194.160' (RSA) to the list of known hosts.
sps.img                                                    100%  336MB  11.2MB/s   00:30
[ewilliams@fenric ~]$ scp -i /home/ewilliams/test-cluster-ew.pem sps.lics azureuser@20.232.194.160:/tmp
sps.lics                                                   100% 6162     6.0KB/s   00:00
[ewilliams@fenric ~]$
```

7. LifeKeeperをインストールします。
   a. SCSI3リカバリーキットが **自動的に** 選択されることに注意してください。

```
[ ] LifeKeeper Device Mapper (DM) Multipath Recovery Kit
[ ] DataKeeper for Linux
[ ] LifeKeeper Hitachi Dynamic Link Manager (HDLM) Recovery Kit
[ ] LifeKeeper LVM Recovery Kit
[ ] LifeKeeper NAS Recovery Kit
[ ] LifeKeeper PowerPath Recovery Kit
[*] LifeKeeper SCSI-3 Persistent Reservations (SCSI3) Recovery Kit
[ ] LifeKeeper NEC iStorage StoragePathSavior (SPS) Recovery Kit
```

8. コミュニケーションパスを設定します。

```
test-cluster-sles15-1:~ # lkcli commpath create --laddr 10.10.0.4 --raddr 10.10.0.5 --dest test-cluster-sles15-2
Performing commpath 'test-cluster-sles15-2:10.10.0.4/10.10.0.5' create...
Commpath 'test-cluster-sles15-2:10.10.0.4/10.10.0.5' created successful.
```

```
test-cluster-sles15-2:~ # lkcli commpath create --raddr 10.10.0.4 --laddr 10.10.0.5 --dest test-cluster-sles15-1
Performing commpath 'test-cluster-sles15-1:10.10.0.5/10.10.0.4' create...
Commpath 'test-cluster-sles15-1:10.10.0.5/10.10.0.4' created successful.
```

9. 仮想マシン(例ではtest-cluster-sles15-1)で共有ディスクにパーティションを作成してください。

```
test-cluster-sles15-1:~ # ls /dev/sd*
/dev/sda  /dev/sda1  /dev/sda2  /dev/sda3  /dev/sda4  /dev/sdb  /dev/sdb1  /dev/sdc
test-cluster-sles15-1:~ # gdisk /dev/sdc
GPT fdisk (gdisk) version 1.0.1

Partition table scan:
  MBR: not present
  BSD: not present
  APM: not present
  GPT: not present

Creating new GPT entries.

Command (? for help): n
Partition number (1-128, default 1):
First sector (34-2147483614, default = 2048) or {+-}size{KMGTP}:
Last sector (2048-2147483614, default = 2147483614) or {+-}size{KMGTP}:
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300):
Changed type of partition to 'Linux filesystem'

Command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING
PARTITIONS!!

Do you want to proceed? (Y/N): y
OK; writing new GUID partition table (GPT) to /dev/sdc.
The operation has completed successfully.
test-cluster-sles15-1:~ #
```

a. test-cluster-sles15-2 でパーティションテーブルを更新します。

```
test-cluster-sles15-2:~ # ls -l /dev/sd*
brw-rw---- 1 root disk 8,  0 Apr 22 20:01 /dev/sda
brw-rw---- 1 root disk 8,  1 Apr 22 20:01 /dev/sda1
brw-rw---- 1 root disk 8,  2 Apr 22 20:01 /dev/sda2
brw-rw---- 1 root disk 8,  3 Apr 22 20:01 /dev/sda3
brw-rw---- 1 root disk 8,  4 Apr 22 20:01 /dev/sda4
brw-rw---- 1 root disk 8, 16 Apr 22 20:01 /dev/sdb
brw-rw---- 1 root disk 8, 17 Apr 22 20:01 /dev/sdb1
brw-rw---- 1 root disk 8, 32 Apr 22 20:01 /dev/sdc
test-cluster-sles15-2:~ # gdisk /dev/sdc
GPT fdisk (gdisk) version 1.0.1

Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present

Found valid GPT with protective MBR; using GPT.

Command (? for help): p
Disk /dev/sdc: 2147483648 sectors, 1024.0 GiB
Logical sector size: 512 bytes
Disk identifier (GUID): 433D6ED3-4D2A-41EE-983D-B7802FD78927
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 2147483614
Partitions will be aligned on 2048-sector boundaries
Total free space is 2014 sectors (1007.0 KiB)

Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048      2147483614   1024.0 GiB  8300  Linux filesystem

Command (? for help): q
test-cluster-sles15-2:~ # ls -l /dev/sd*
brw-rw---- 1 root disk 8,  0 Apr 22 20:01 /dev/sda
brw-rw---- 1 root disk 8,  1 Apr 22 20:01 /dev/sda1
brw-rw---- 1 root disk 8,  2 Apr 22 20:01 /dev/sda2
brw-rw---- 1 root disk 8,  3 Apr 22 20:01 /dev/sda3
brw-rw---- 1 root disk 8,  4 Apr 22 20:01 /dev/sda4
brw-rw---- 1 root disk 8, 16 Apr 22 20:01 /dev/sdb
brw-rw---- 1 root disk 8, 17 Apr 22 20:01 /dev/sdb1
brw-rw---- 1 root disk 8, 32 Apr 22 20:38 /dev/sdc
brw-rw---- 1 root disk 8, 33 Apr 22 20:38 /dev/sdc1
test-cluster-sles15-2:~ #
```

10. パーティションにファイルシステムを作成します。

```
test-cluster-sles15-1:~ # mkfs.xfs /dev/sdc1
meta-data=/dev/sdc1              isize=512    agcount=4, agsize=67108799 blks
         =                       sectsz=4096  attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=0, rmapbt=0
         =                       reflink=0
data     =                       bsize=4096   blocks=268435195, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=131071, version=2
         =                       sectsz=4096  sunit=1 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
test-cluster-sles15-1:~ # mount /dev/sdc1 /test0
```

11. ファイルシステムリソースを作成(Create)します。

```
test-cluster-sles15-1:~ # lkcli resource create fs --tag /test0 --mountpoint /test0
devicehier: Using /opt/LifeKeeper/lkadm/subsys/scsi/scsi3p/bin/devicehier to construct the hierarchy
devicehier: Using /opt/LifeKeeper/lkadm/subsys/scsi/scsi3/bin/devicehier to construct the hierarchy
devicehier:  resource "scsi312295" already exists on server "test-cluster-sles15-1" with identical information, app="scsi" type="scsi3"
Creating resource "scsi3p12846" on system "test-cluster-sles15-1".
Creating dependency "scsi3p12846"-"scsi312295" on machine "test-cluster-sles15-1".
BEGIN restore of "scsi3p12846" on server "test-cluster-sles15-1"
END successful restore of "scsi3p12846" on server "test-cluster-sles15-1"
```

12. スタンバイノードにファイルシステムを拡張(Extend)します。

```
test-cluster-sles15-1:~ # lkcli resource extend fs --tag /test0 --dest test-cluster-sles15-2
Extending resource instances for /test0
 Creating dependencies
 Setting switchback type for hierarchy
 Creating equivalencies
 LifeKeeper Admin Lock (/test0) Released
 Hierarchy successfully extended

test-cluster-sles15-1:~ #
```

13. 以上で手順は完了です。

```
test-cluster-sles15-1:~ # lcdstatus -q
LOCAL                 TAG          ID                                      STATE     PRIO  PRIMARY
test-cluster-sles15-1 /test0       /test0                                  ISP          1  test-cluster-sles15-1
test-cluster-sles15-1  scsi3p12846 360022480f4058f4594d104121d7a344f-1      ISP          1  test-cluster-sles15-1
test-cluster-sles15-1   scsi312295 360022480f4058f4594d104121d7a344f        ISP          1  test-cluster-sles15-1

MACHINE               NETWORK ADDRESSES/DEVICE     STATE     PRIO
test-cluster-sles15-2 TCP     10.10.0.4/10.10.0.5  ALIVE        1
test-cluster-sles15-1:~ #
```

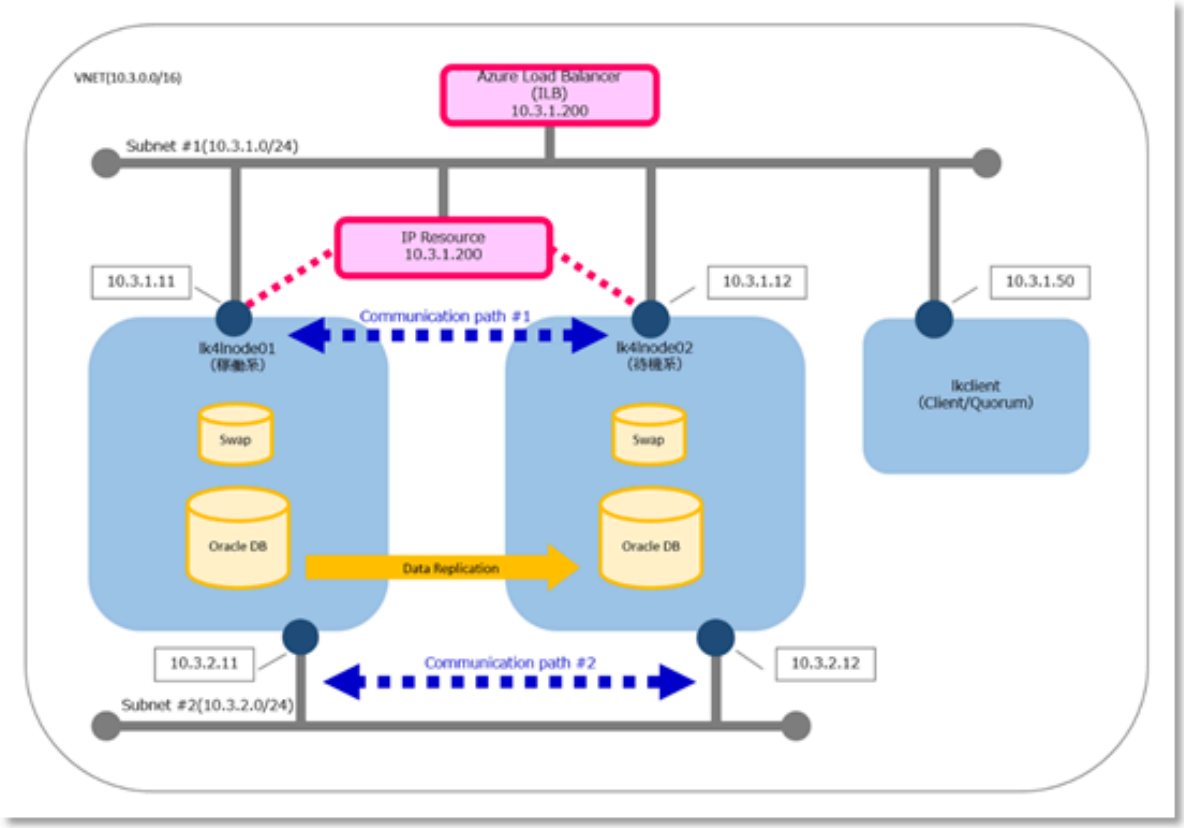
```
test-cluster-sles15-2:~ # lcdstatus -q
LOCAL                 TAG          ID                                      STATE     PRIO  PRIMARY
test-cluster-sles15-2 /test0       /test0                                  OSU         10  test-cluster-sles15-1
test-cluster-sles15-2  scsi3p12846 360022480f4058f4594d104121d7a344f-1      OSU         10  test-cluster-sles15-1
test-cluster-sles15-2   scsi312295 360022480f4058f4594d104121d7a344f        OSU         10  test-cluster-sles15-1

MACHINE               NETWORK ADDRESSES/DEVICE     STATE     PRIO
test-cluster-sles15-1 TCP     10.10.0.5/10.10.0.4  ALIVE        1
test-cluster-sles15-2:~ #
```

# 12.2.3. 構成情報

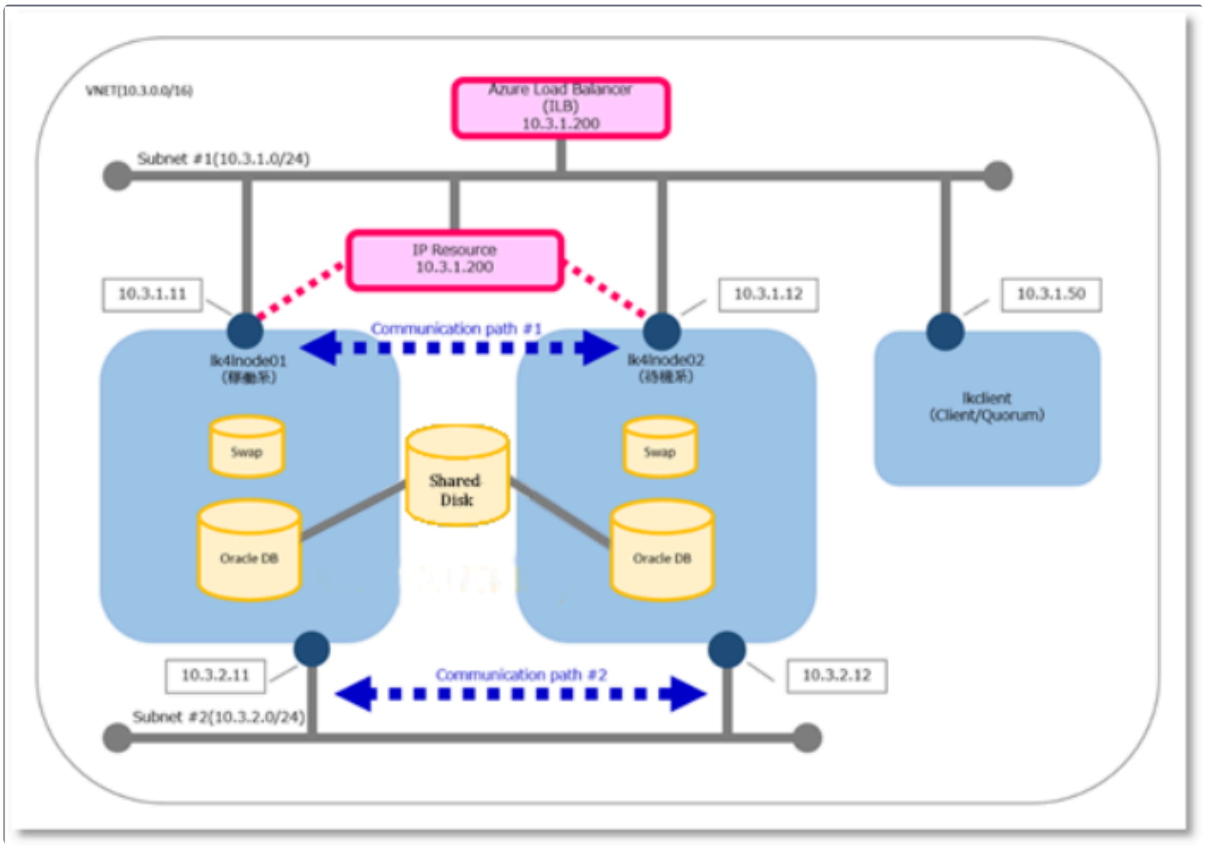
本構成では LifeKeeper を使って、以下の構成のアクティブ/スタンバイクラスターを構築します。 マルチ NIC 構成の Azure VM に LifeKeeper を導入します。

LifeKeeper は、リージョン内の共有ストレージとして Microsoft Azure 共有ディスクをサポートしており、ローカル冗長ストレージ (LRS) とゾーン冗長ストレージ (ZRS) の両方がサポートされています。 LifeKeeper は、リージョン間の共有ストレージとして Microsoft Azure 共有ディスクをサポートしていません。GEO 冗長ストレージと GEO ゾーン冗長ストレージの両方がサポートされていません。 これらの構成では、以下に示すように、LifeKeeper を使用してアクティブ/スタンバイ クラスタを構築します。 LifeKeeper は、マルチ NIC 構成の Azure VM にデプロイされます。

## データーレプリケーション



## Azure 共有ディスク




## サーバー構成

| | |
|---|---|
| 仮想マシンサイズ | クラスターノード: Standard_A1 (1 コア, 1.75 GB メモリ)<br>Client/Witness ServerはStandard B1s (1 vcpu 数、1 GiB メモリ)<br>(0.25 core, 0.75GB memory)<br>*注1 |
| データディスク | 30GiB (Oracle DB 用)<br>10GiB (swap 領域用)<br>(/dev/disk/cloud/azure_resourceはデータディスクとして使用できません ※注2)<br>*注1 |
| **[IP リソースが保護する VIP]** | 10.3.1.200 |
| **[ILB に設定する IP]** | 10.3.1.200 |

| | |
|---|---|
| **ILB** が転送するポート | 1521 |
| **ILB** の負荷分散先ホストおよびポート | lk4lnode01/lk4lnode02 の各 1521 ポート |
| **[Azure** のプライベート **IP** アドレス**]** | クラスターノード (稼働系) 10.3.1.11 / 10.3.2.11<br>クラスターノード (待機系) 10.3.1.12 / 10.3.2.12<br>クライアント兼 Witness サーバー 10.3.1.50 |

※注１：Oracle のインストール要件（メモリや Oracle データベース･インストールディスクサイズ、Swap エリアサイズ）に合うよう、インスタンスサイズおよび仮想マシンのディスクを用意してください。

※注２：ほとんどの Azure VM は一時ディスク (/dev/disk/cloud/azure_resource) を含んでいますが、これは管理された Disk ではありません。一時ディスクは、アプリケーションとプロセスのための短期ストレージを提供し、ページまたはスワップファイルのようなデータのみを格納することを意図しています。一時ストレージは自動的に /mnt にマウントされますが、以下の例では /mnt/resource にマウントされています。azure_resource は /dev/sdb デバイスノードであることが多いですが、設定によっては別のデバイスノードになることもあります。この一時ディスクは、DataKeeper で使用されるストレージのような LifeKeeper で保護されたデバイスとして使用するのには適していません。詳細については、/mnt/DATALOSS_WARNING_README.txt を参照してください。

※注 3: Microsoft Azure 共有ディスクには、LifeKeeper 9.6.2 以降で利用可能な steeleye-lkSCSI3 リカバリキットが必要です。

## ソフトウェア構成

| | |
|---|---|
| **OS** | RedHat Enterprise Linux 7.8 64bit |
| **LifeKeeper** | SIOS Protection Suite for Linux v9.5.2 |
| **Oracle** | Oracle Database 19c (19.3) Enterprise Edition |

# 監視対象

| | |
|---|---|
| 監視対象 | IP リソースが保護する VIP / ファイルシステム / DataReplication / Oracle DB / Oracle Listener / Oracle PDB / LB Health Check |

## ネットワーク構成

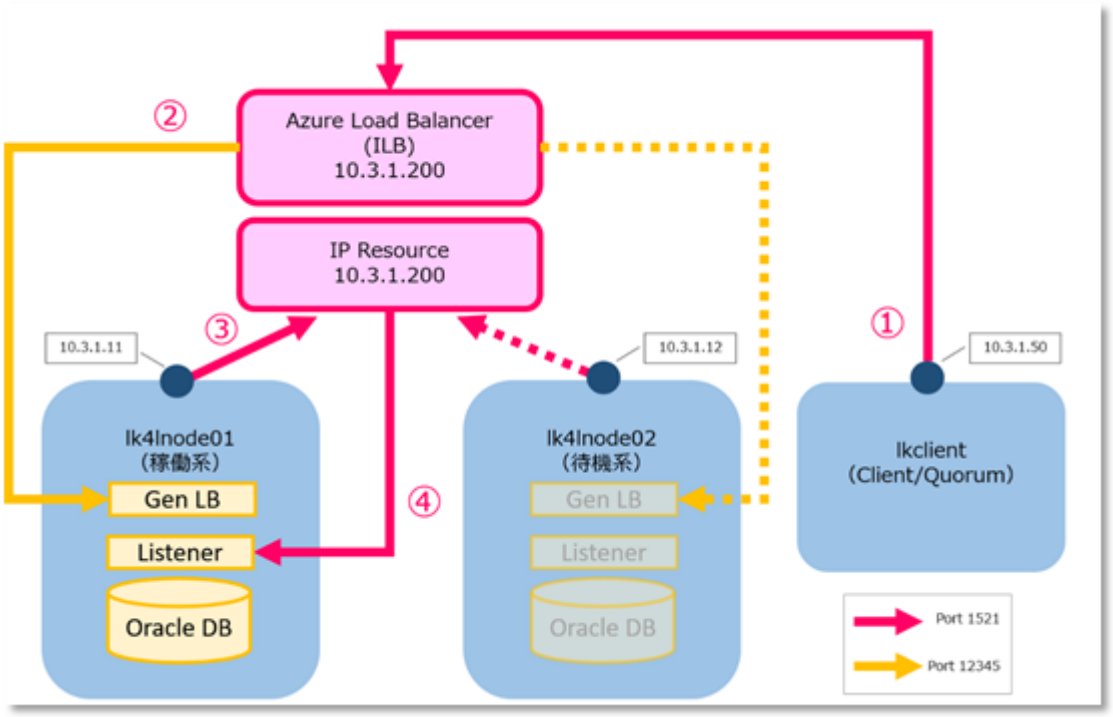
Azure では、仮想マシン（VM）間での通信を行えるようにする為、“仮想ネットワーク”（VNET）を作成します。VNET は、サブネットを指定する事で、このサブネット内で VM間での通信が行えるようにしています。

Azure で VM にネットワークを構成する場合、あらかじめ VNET を作成し、VM にこの VNET を指定するのが一般的です。VNET を使用すると、VM に仮想プライベートネットワーク (VPN) を提供できます。オプションとして、VPN をオンプレミス環境に接続して、ハイブリッドソリューションやクロスプレミスソリューションの実現もできます。VNET は管理ポータルサイトや Azure PowerShell を通して、DNS および IP アドレス範囲の構成を含むネットワークトポロジーを制御できます。

# 12.2.4. Azure 上の LifeKeeper 特有の設定について

## 1. インターナルロードバランサー（ILB）の導入

Azure では [IP リソースが保護する VIP] を Azure の VNET で認識する事が出来ません。この影響で、通常 LifeKeeper for Linux が想定している [IP リソースが保護する VIP] によるネットワーク通信を行うことが出来ません。その為 LifeKeeper では、以下の様に ILB の導入を行い、[ILB が設定する VIP] をネットワーク通信経路として設定します。




① Oracle Client が Oracle Listener へ接続する為、10.3.1.200(port1521)への接続を ILB 経由で開始します。
② ILB に負荷分散先（backend pool) に 10.3.1.11（稼働系ノード：lk4lnode01）と 10.3.1.12（待機系ノード：lk4lnode02） が 登録されています。ILBは各ノードに対して正常性プローブによる監視を行い、ポートがオープンになっているノードを確認して、受け取ったOracle Listener宛のパケットを転送します。正常性プローブへの応答には LB Health Check リソースを使用します。

③ LifeKeeperの仕様上Oracle Listenerリソースは[IPリソースが保護するVIP]を指定する必要があります。その為、受信用にIPリソースを作成します。稼働系のIPリソースがIn-Serviceになっているため、10.3.1.200宛のリクエストを受け取ります。
④ 結果として Oracle Client からの接続要求は稼働系の Oracle Listener（10.3.1.200：port1521）が受け取ります。

## アドレス変換

LifeKeeper では、IP リソースの監視処理に外部ホストへの PING を使用しますが、Azure の仕様上、仮想

IP を送信元とする PING 処理は行えません。その為、以下のようにパケットの送信元情報を変更する設定を行うことで、PING 処理を有効にし、外部ホスト間とのネットワークの障害を検出可能にします。

① IP リソースから IP リソースで保護する VIP をソースとして指定した PING を開始します。
② PING の送信元アドレスを、10.3.1.200 (IP リソースで保護する VIP) から10.3.1.11 (Azure のプライベート IP アドレス) に置き換えます。
③ ソース置き換え後、PING の送信先にパケットを送信します。[Azure のプライベート IP アドレス] に置き換えられる事で、ネットワークが正常であれば応答が返ります。

## Quorum / Witness の導入

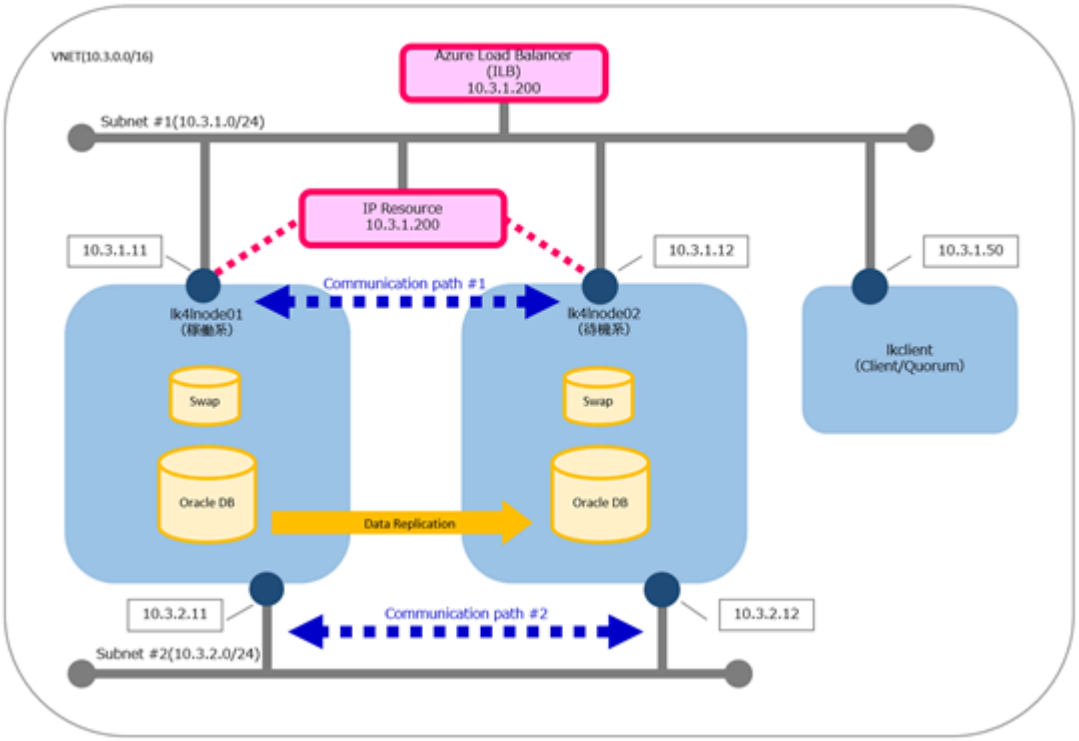
Azure 環境上では、クラスターの共有ディスクの代替として DataKeeper によるレプリケーションディスクを使用します。DataKeeper を使用したクラスターでは、コミュニケーションパスが全断するようなネットワークの障害が発生した際、その仕組み上スプリットブレイン状態になる可能性があります。これを回避するため、Quorum / Witness 機能による I/O Fencing を設定してください。Quorum / Witness機能の詳細につきましては、テクニカルドキュメンテーションQuorum/Witness のページおよび、LifeKeeper for Linux Quorum Check/Witness Server 機能利用ガイド をご参照ください。

# 12.2.5. 仮想マシンの構築、OS の起動

## 仮想マシンの構築、OS の作成

Azure 上に仮想マシンを作成します。
前述の通り、Azure で LifeKeeper を使用する場合、Azure 環境に特有な設定（以下 3 点）を行う必要があります。

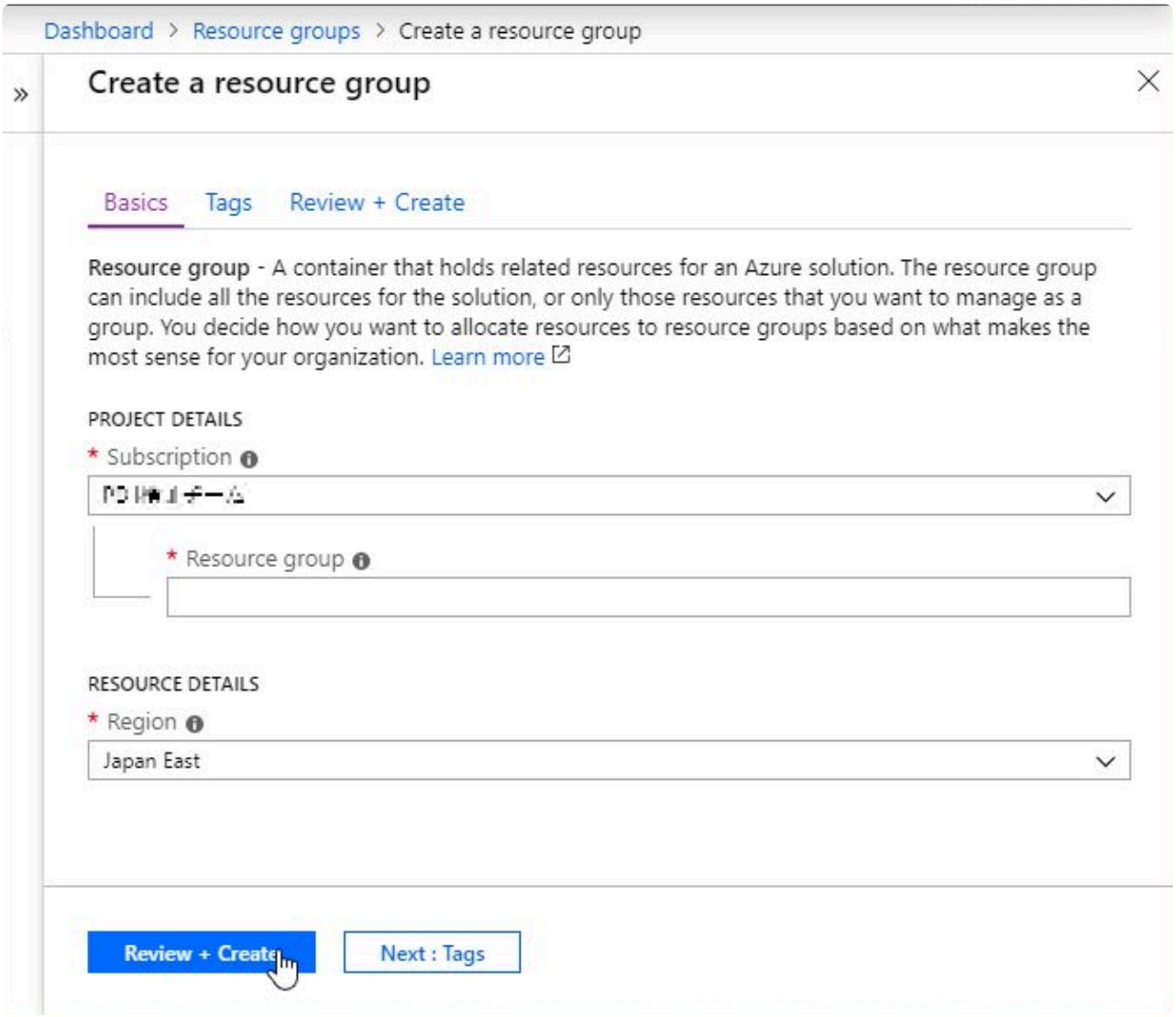
- 内部負荷分散（Internal Load Balancing または ILB）を使用し、LifeKeeper で保護する Oracle リソースへの接続を有効にする。
- 仮想 IP アドレスを使用した通信のため、アドレス変換設定を行う。
- スプリットブレインを回避するため Quorum の設定を行う。




※ 以降の Azure コンソールのスクリーンショットや入力例は、2020 年 4 月時点のものです。本書のイメージと実際の画面に違いがある場合は、適宜読み替えてください。

# 12.2.5.1. リソースグループの作成

1. **Azure Portal** > **Resource groups** > **Create** を選択し、本環境で使用するリソースグループを作成します。必要な値を入力し、 **Review + Create** をクリックします。

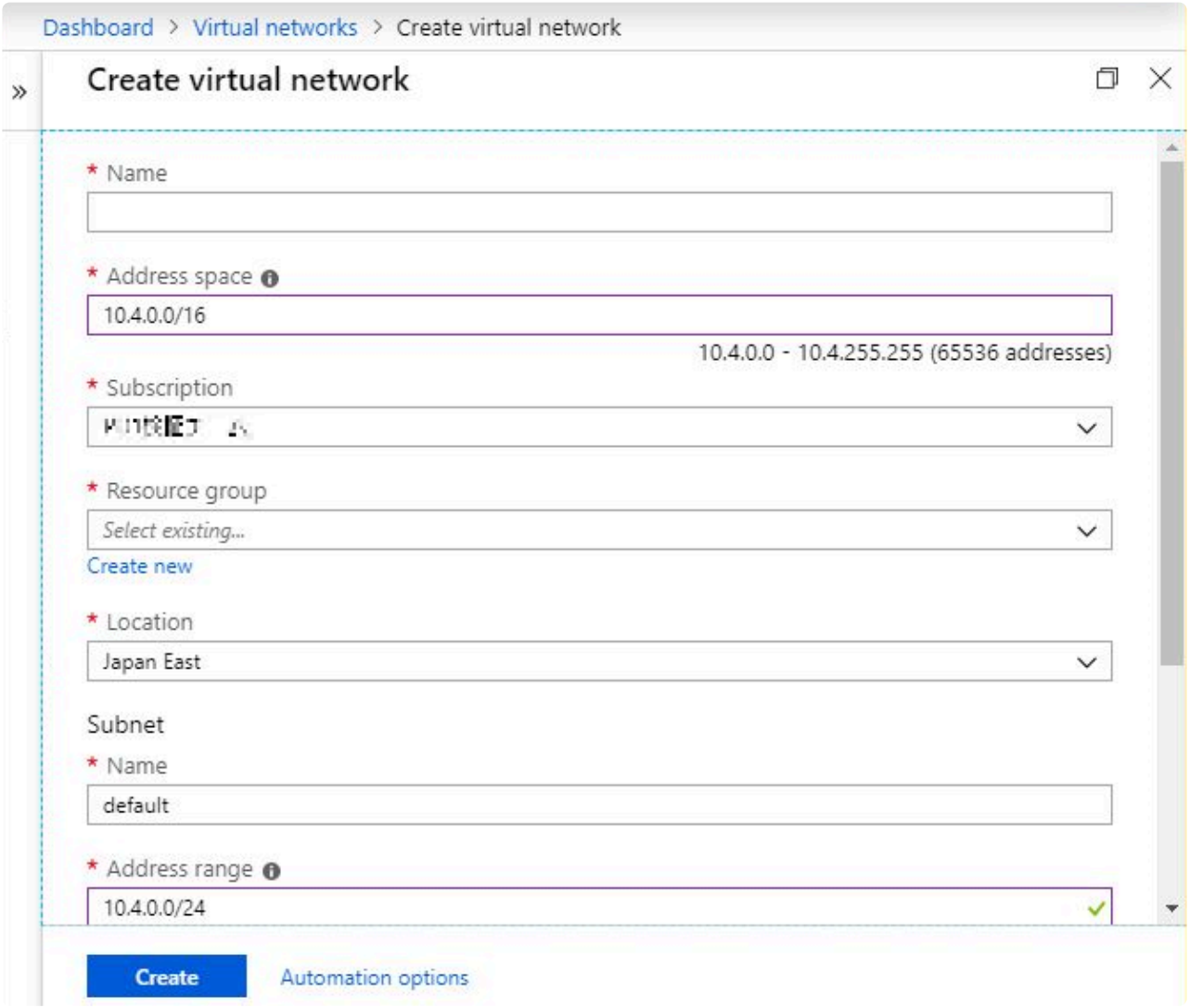



2. 内容を確認し、 **Create** をクリックします。

# 12.2.5.2. 仮想ネットワークの作成

本環境で使用する仮想ネットワークを作成します。今回は、アドレススペース１つ、サブネット２つを作成します。このうちひとつのサブネットはサービス用、もうひとつのサブネットは DataKeeper レプリケーション用になります。

1. **Azure Portal > Virtual networks > Create** を選択し、本環境で使用する仮想ネットワークと、同時にサブネットを１つ作成します。



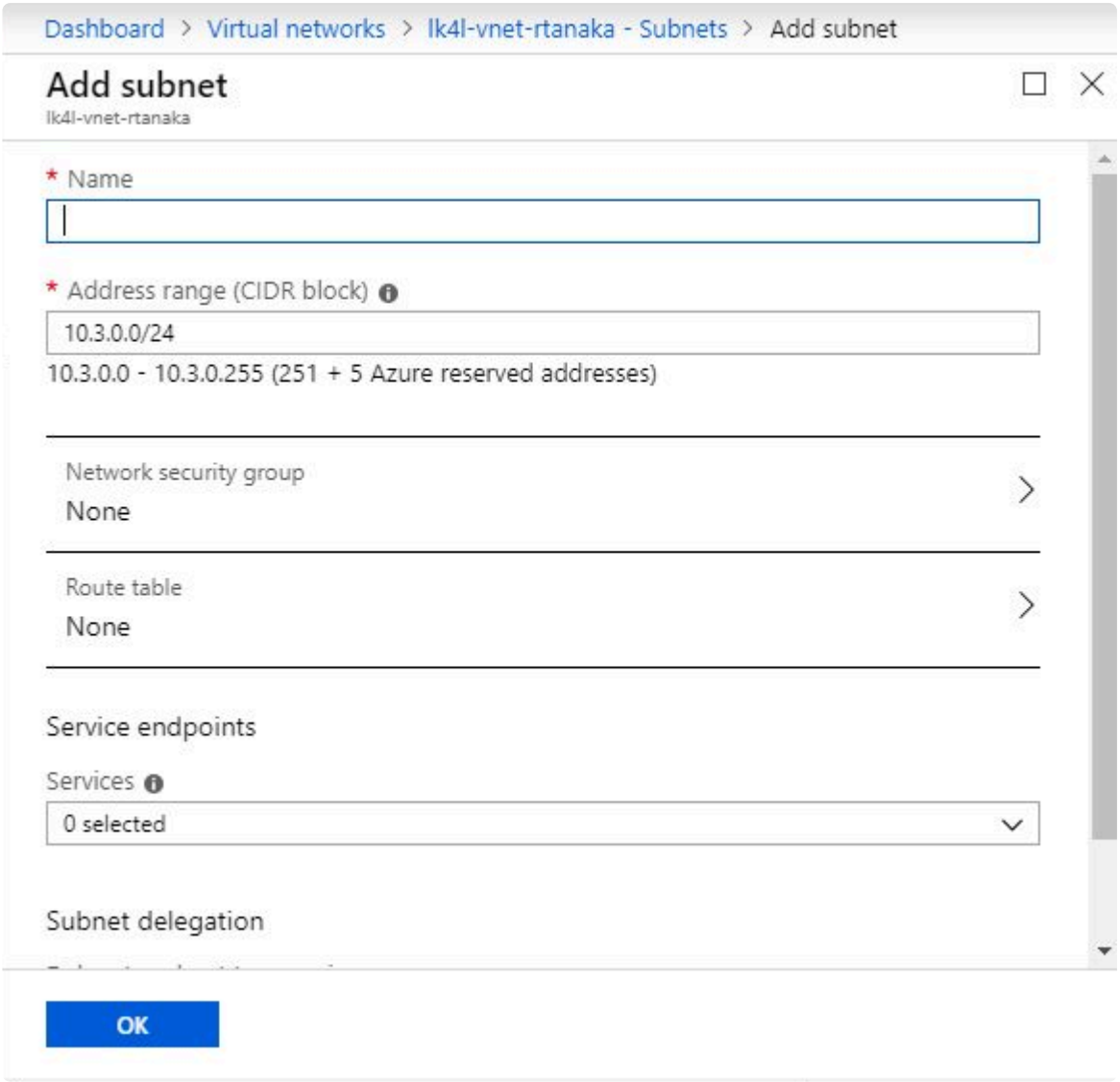

2. 必要な値を入力し、 **Create** をクリックします。

   今回は以下の値を使用します。

| 項目 | 入力もしくは選択する値 | 備考 |
|---|---|---|
| Name | lk4l-vnet | |
| Address space | 10.3.0.0/16 | |
| **Subnet** | | |

| | | |
|---|---|---|
| Name | lk4l-nw01 | |
| Address range | 10.3.1.0/24 | |
| Firewall | Disabled (default) | |

3. **Azure Portal** > **Virtual networks** > [(仮想ネットワーク名)] > **Subnets** > **Subnet** を選択し、仮想ネットワークに対して２つめのサブネットを追加します。




4. 必要な値を入力し、 **OK** をクリックします。

   今回は以下の値を使用します。

| 項目 | 入力もしくは選択する値 | 備考 |
|---|---|---|
| Name | lk4l-nw02 | |
| Address range | 10.3.2.0/24 | |

# 12.2.5.3. 仮想マシンの作成

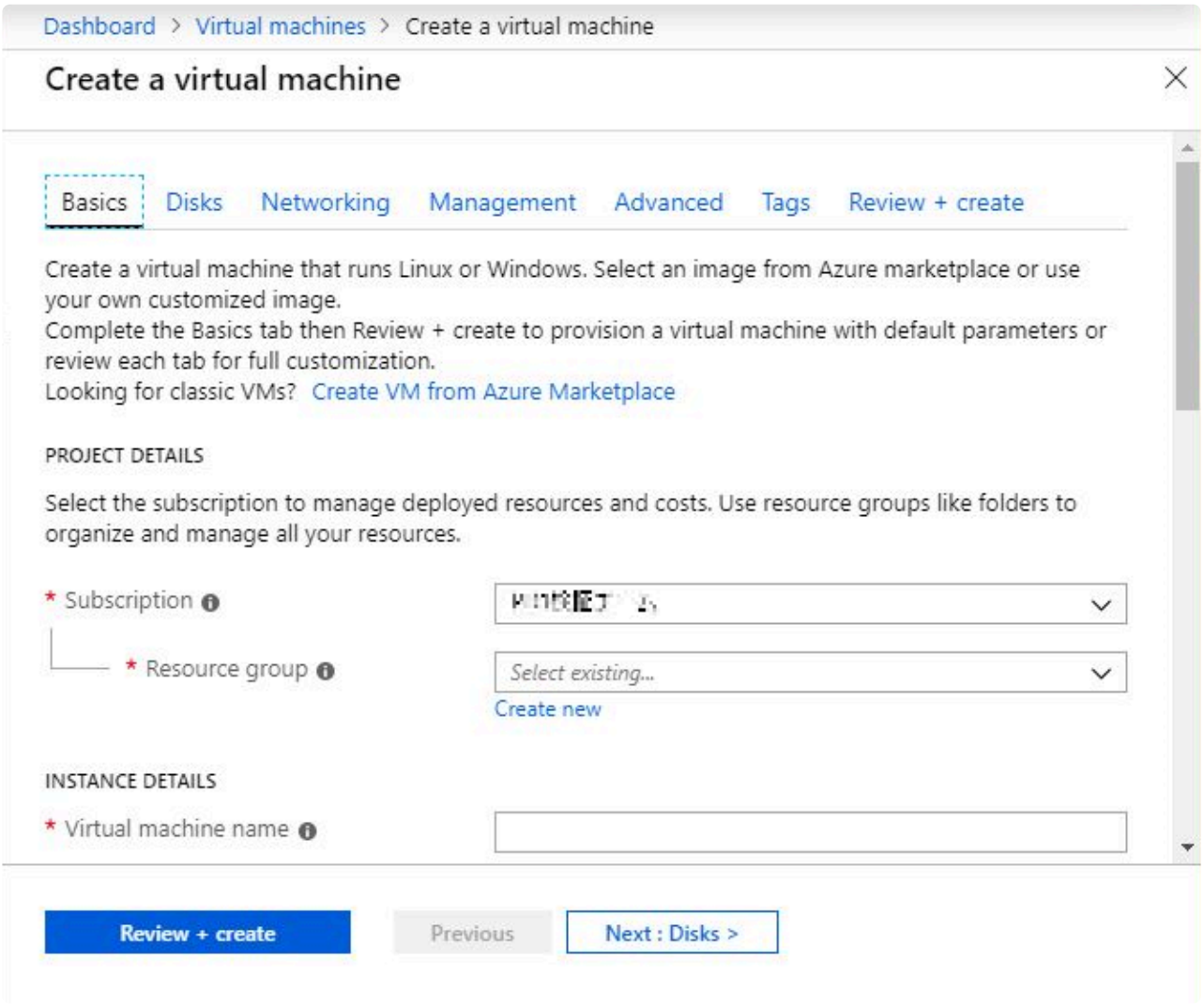
本環境で使用する仮想マシンを作成します。

作成する仮想マシンは以下の３台です。

- クラスターノード（稼働系）
- クラスターノード（待機系）
- クライアント兼 Witness サーバー

# 12.2.5.3.1. クラスターノード（稼働系）の作成

1. **Azure Portal > Virtual machines > Create** を選択します。
   はじめに、基本設定を入力します。




今回は以下の値を使用します。

| **Basics** | | |
|---|---|---|
| 項目 | 入力もしくは選択する値 | 備考 |
| **INSTANCE DETAILS** | | |
| Virtual machine name | lk4lnode01 | |
| Availability options | Availability Set またはAvailability Zone | *注1 |
| | (create new)lk4lcluster | |
| Image | Red Hat Enterprise Linux 7.8 | *注2 |
| Size | Standard_A1_v2 | *注3 *注4 |

| ADMINISTRATOR ACCOUNT | | |
|---|---|---|
| Authentication type | Password | |
| Username | lkadmin | |
| Password | XXXXXXXX | |
| **INBOUND PORT RULES** | | |
| Public inbound ports | Allow selected ports | |
| Select inbound ports | SSH | |

- **注1:** ここではAvailability Setで説明します。Availiavility Zoneを選択した場合の設定については「Availability Zone (高可用性ゾーン) 」のページを参照してください。Availability Setを選択した場合、クラスタを構成するそれぞれのノードとQWKノードが別々のフォールトドメイン・アップデートドメインに属するようにする必要があります。

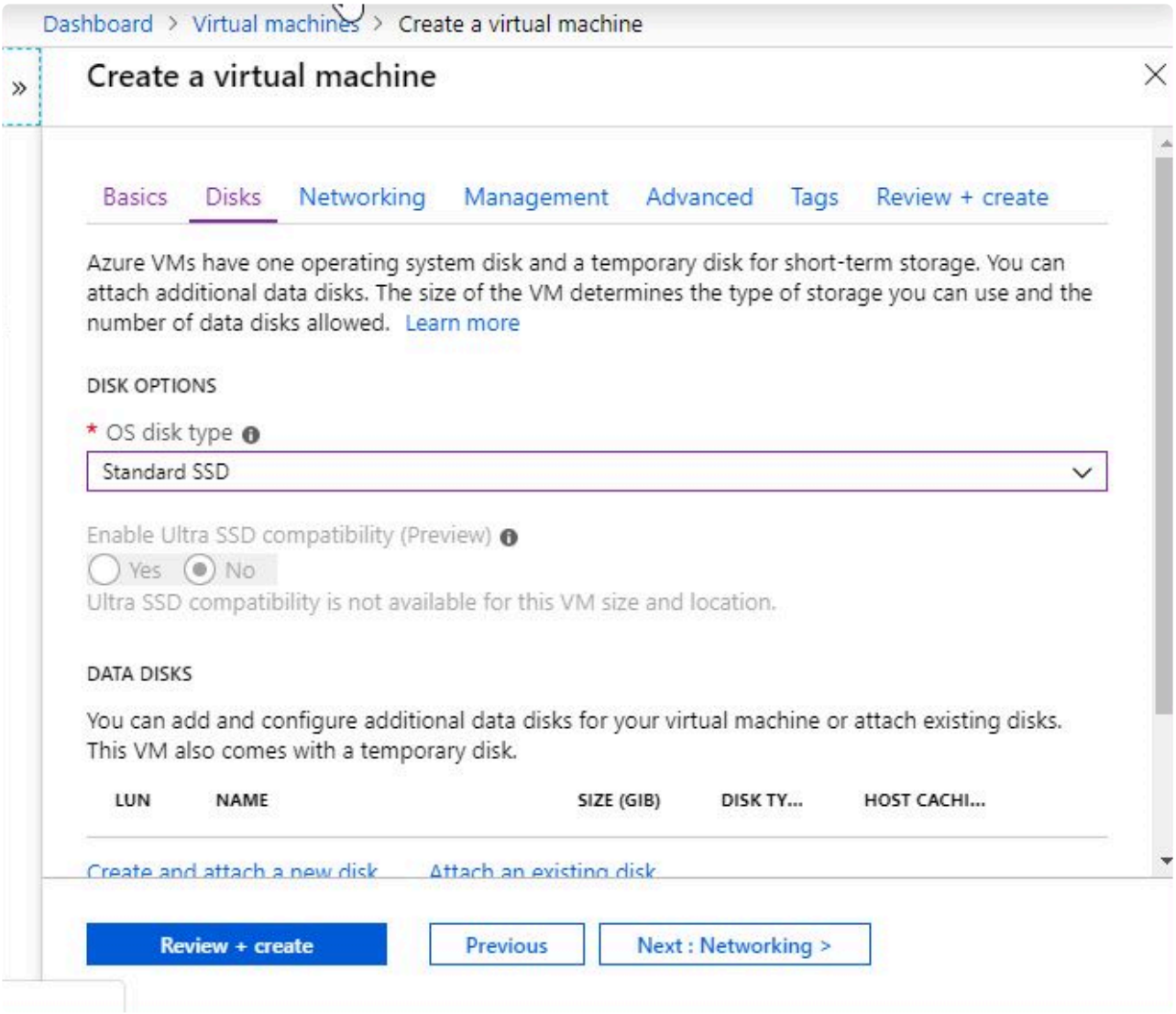
- **注2:** 導入する LifeKeeper がサポートする OS バージョンを選択してください。 LifeKeeper がサポートする OS バージョンは、各バージョンのサポートマトリクス を参照ください。

- **注3:** 必ず Availability Set が利用できるインスタンスサイズを選択してください。 一部のインスタンスサイズでは Availability Set が利用できません。

- **注4:** 必ず負荷分散規則が利用できるインスタンスサイズを選択してください。一部 のインスタンスサイズでは、負荷分散規則が利用できません。

2. 必要な値を入力したら、 **Next:Disks** をクリックします。

3. 次にディスク設定を入力します。
   今回の環境では、OS ディスクに加えて Oracle DB 用のディスク（30GiB）と Swap 領域用のディスク (10GiB) が必要なため、 **Create and attach a new disk** をクリックして追加のディスクをアサインします。




今回は以下の値を使用します。

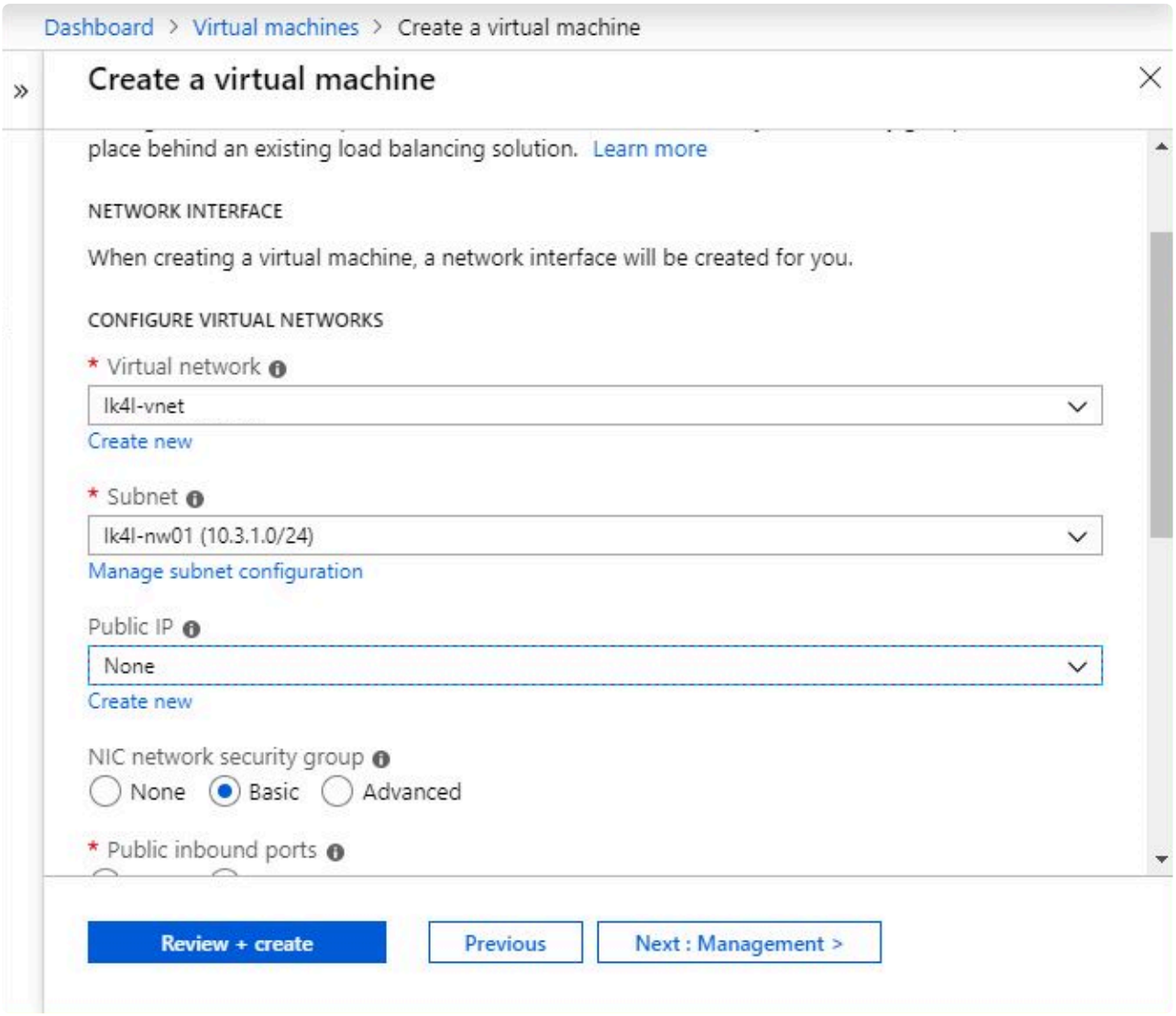
| **Disks** | | |
|---|---|---|
| 項目 | 入力もしくは選択する値 | 備考 |
| **DISK OPTIONS** | | |
| OS Disk type | Standard SSD | |
| **DATA DISKS (1st)** | | |
| LUN | 0 | |
| NAME | (create and attach a new disk)<br><br>lklnode01_DataDisk_0 | |
| Size(GiB) | 30 | |
| DISK TYPE | Standard SSD | |
| HOST CACHING | None | *注1 |

| DATA DISKS (2nd) | | |
|---|---|---|
| LUN | 1 | |
| NAME | (create and attach a new disk)<br><br>lklnode01_DataDisk_1 | 任意の入力値 |
| Size(GB) | 10 | |
| DISK TYPE | Standard SSD | |
| HOST CACHING | None | *注1 |

- **注 1 :** 本環境ではデータベース用途のため、キャッシュを無効化しましたが、要件に応じて適切に設定してください。

4. 必要な値を入力したら、 **Next:Networking** をクリックします。

5. 次にネットワーク設定を入力します。




このとき、前の手順で作成した１つめのサブネットを、仮想マシンに対して設定します。この設定

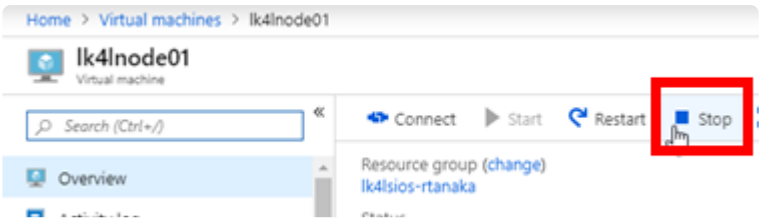
により、ネットワークインターフェースがひとつアタッチされます。（２つ目のネットワークインターフェースは、仮想マシン作成後に設定します。）

今回は以下の値を使用します。

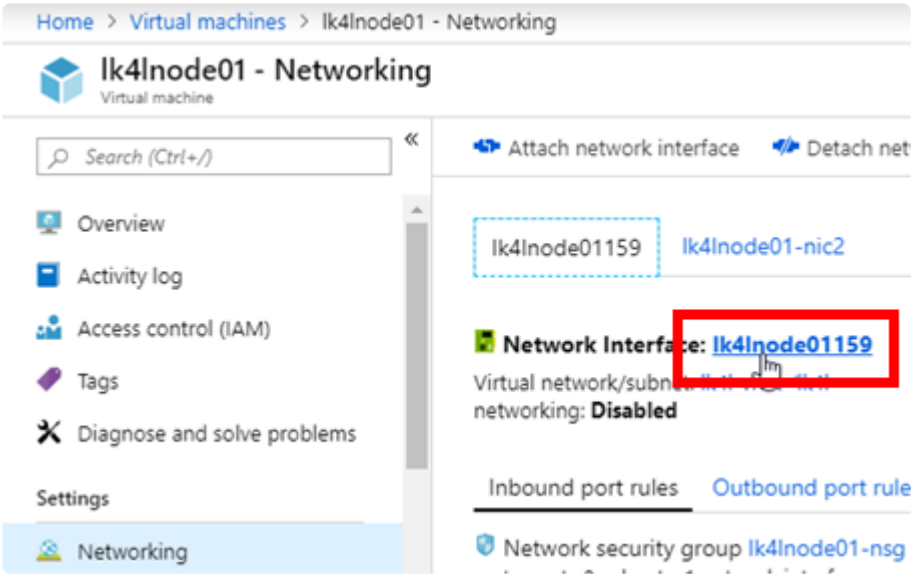
| **Networking** | | |
|---|---|---|
| **項目** | **入力もしくは選択する値** | **備考** |
| Virtual network | lk4l-vnet | |
| Subnet | lk4l-nw01(10.3.1.0/24) | *注1 |
| Public IP | (create new)<br><br>Name:lk4lnode01pubip<br><br>SKU:Basic<br><br>Assignment:Static | *注2 |
| NIC network security group | Basic | |
| Public inbound ports | Allow selected ports | |
| Selected inbound ports | SSH | |

- **注1:** 前の手順で作成した１つめのサブネットを、仮想マシンに対して設定します。
- **注2:** 本手順ではインターネット経由でアクセスするため、Public IP を設定しています。

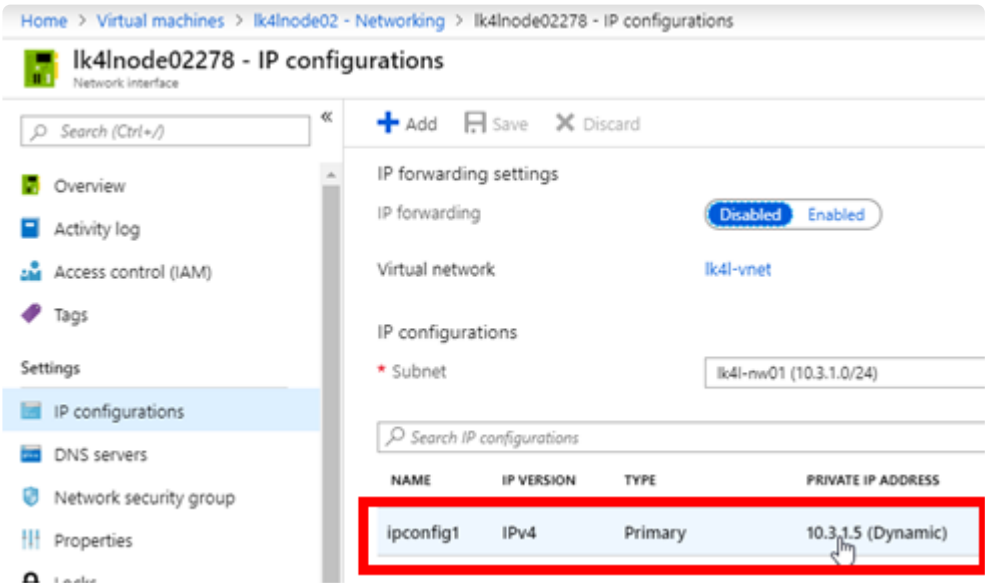
6. 必要に応じて管理設定、詳細設定、タグ設定を行い、最後に **Review + create** をクリックします。
7. 内容を確認し、 **Create** をクリックします。上記の構成でデプロイすると、Redhat Enterprise Linux のインストールが完了した状態で起動します。
8. このあとネットワーク設定変更を行うため、一旦仮想マシンを停止します。



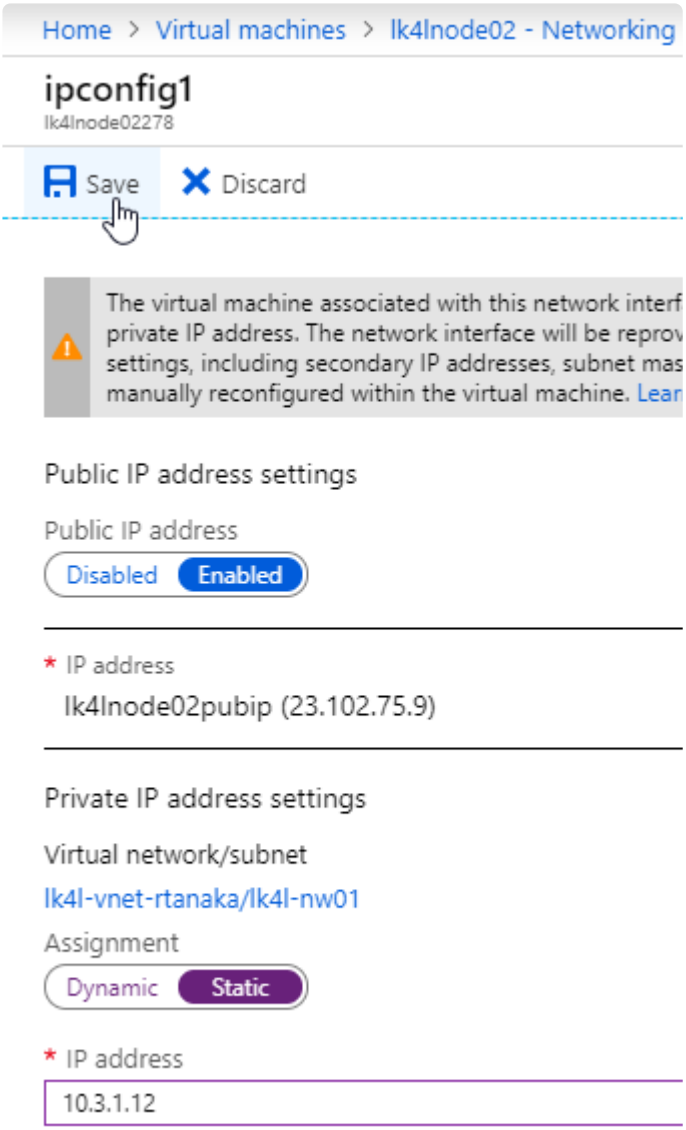

9. １つめのネットワークインターフェースを静的 IP アドレスに変更します。 **Azure Portal > Virtual machines > (仮想マシン名) > Networking >** （１つめのインターフェース名） をクリックします。




10. **IP configurations > ipconfig1** をクリックします。この時点では、IP アドレスの設定が動的 (Dynamic) になっています。




11. IP アドレスを静的に設定変更します。

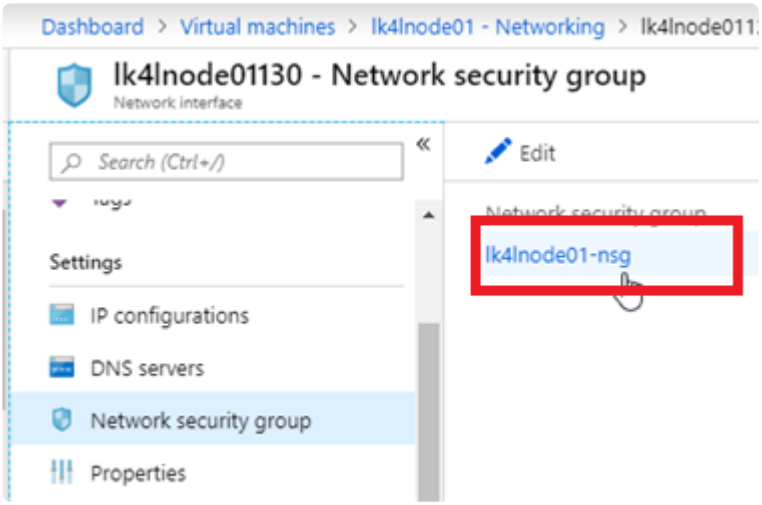



今回は以下の値を使用します。

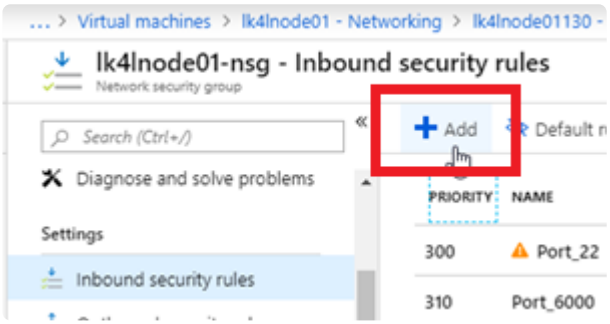
| 項目 | 入力もしくは選択する値 | 備考 |
|---|---|---|
| Assignment | Static | |
| IP address | 10.3.1.11 | |

12. **Save** をクリックして設定を保存します。

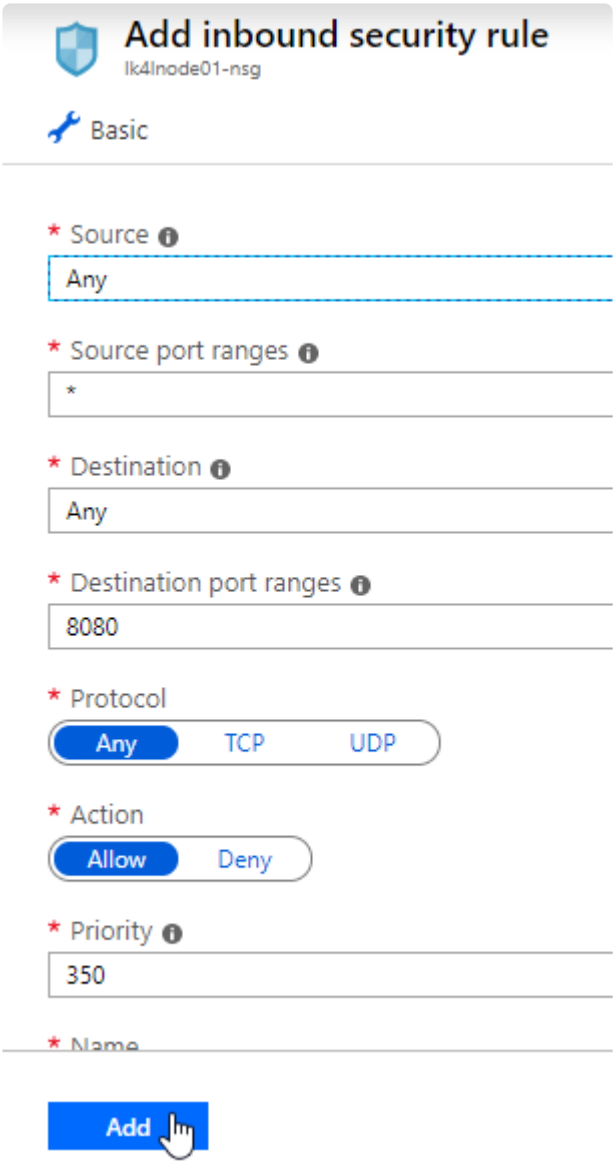
13. Oracle リスナーサービスで、TCP Port 1521 を使用して通信できるようにするため、ネットワークセキュリティーグループを編集します。 **Azure Portal > Virtual machines > (仮想マシン名) > Networking >** （１つめのインターフェース名） **> Network security group > (セキュリティーグループ名)** をクリックします。




14. **Inbound security rules** > **Add** をクリックします。




15. インバウンドルールを追加します。



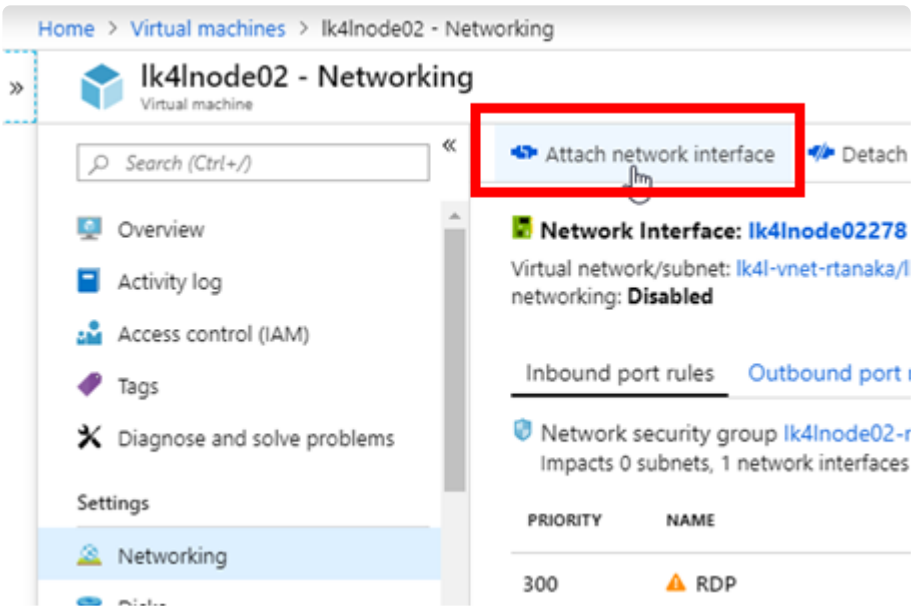

以下の値を使用します。

| 項目 | 入力もしくは選択する値 | 備考 |
|---|---|---|
| Source | Any | |
| Source port ranges | * | |
| Destination | Any | |
| Destination port ranges | 1521 | *注1 |
| Protocol | TCP | |
| Action | Allow | |
| Name | Port_1521 | |

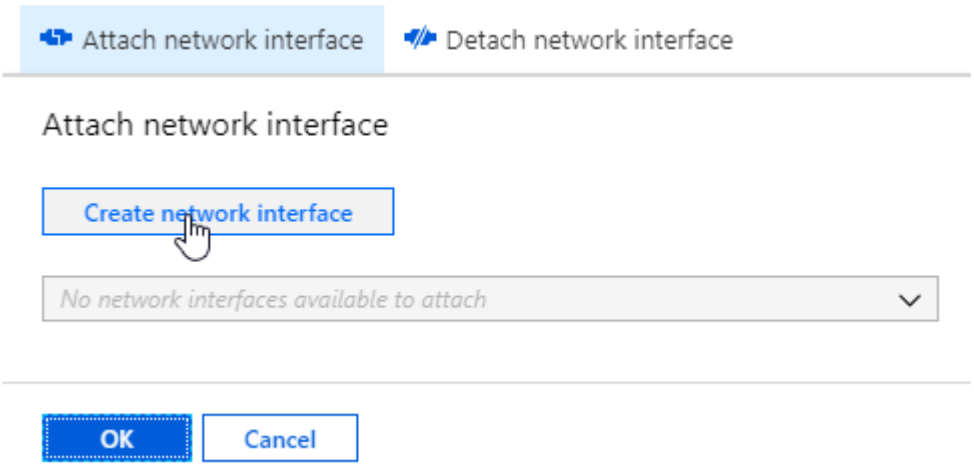
- **注1:** Oracleリスナーがport1521を使用するため、該当のポートへの通信を許可します。

16. **Add** をクリックし、設定を保存します。

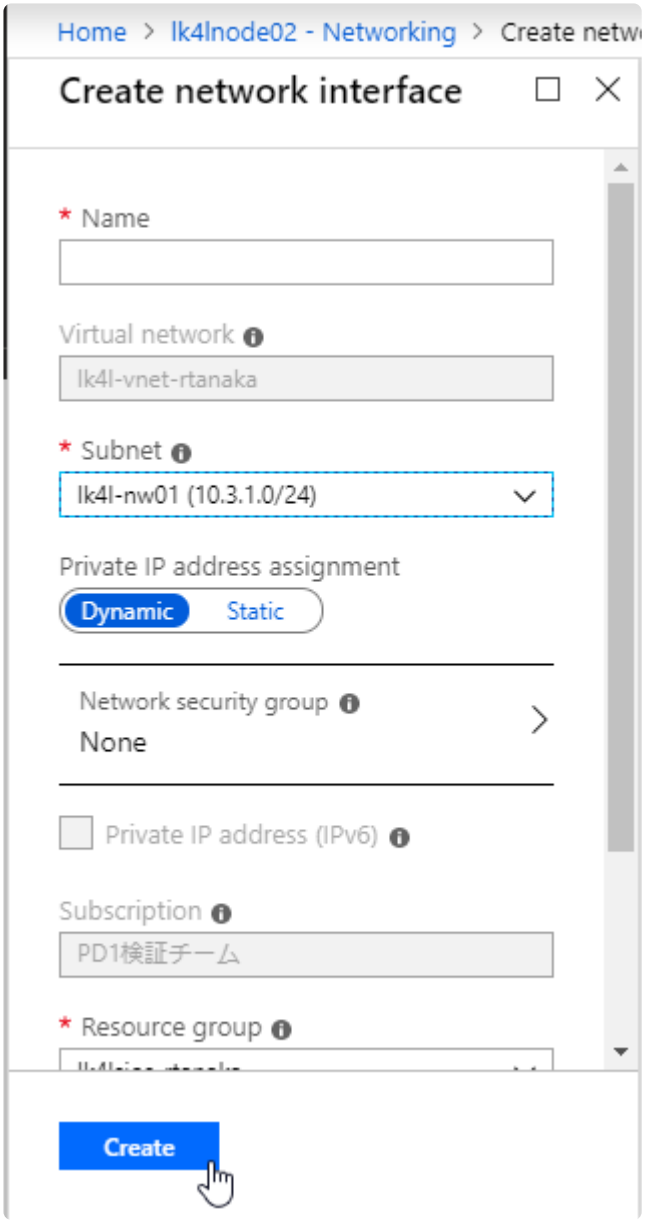
17. 次に、２つ目のネットワークインターフェースを追加します。 **Azure Portal > Virtual machines > (仮想マシン名) > Networking > Attach network interface** をクリックします。




18. **Create network interface** をクリックします。




19. 必要な値を入力したら、 **Create** をクリックします。



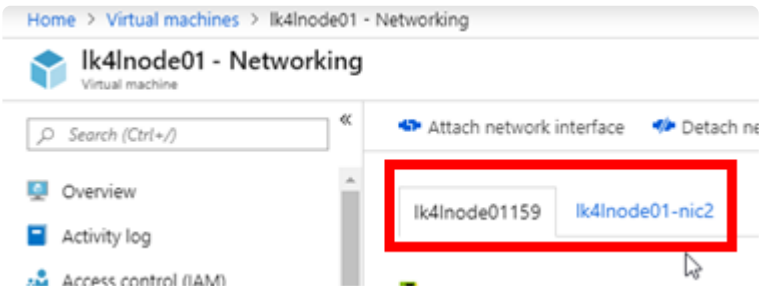

以下の値を使用します。

| 項目 | 入力もしくは選択する値 | 備考 |
|---|---|---|
| Name | lk4lnode01-nic2 | |
| Subnet | 10.3.2.0/24 | *注1 |
| Private IP address assignment | Static | |
| Private IP address | 10.3.2.11 | |
| Network security group | None | |

- **注1:** 前の手順で作成した２つめのサブネットを、仮想マシンに対して設定します。

20. **OK** をクリックします。

21. インターフェースが２つアタッチされていることを確認します。




22. 仮想マシンを起動します。

# 12.2.5.3.2. クラスターノード（待機系）の作成

1. 同様の手順で、クラスターノード（待機系）に使用する仮想マシンを作成します。基本設定は、以下の値を使用します。

| **Basics** | | |
|---|---|---|
| **項目** | **入力もしくは選択する値** | **備考** |
| **INSTANCE DETAILS** | | |
| Virtual machine name | lk4lnode02 | |

その他の設定はすべて稼働系と同じにしてください。

ディスク設定は、以下の値を使用します。

| **Disks** | | |
|---|---|---|
| **項目** | **入力もしくは選択する値** | **備考** |
| **DATA DISKS (1st)** | | |
| LUN | 0 | |
| NAME | (create and attach a new disk)<br><br>lklnode02_DataDisk_0 | |
| **DATA DISKS (2nd)** | | |
| LUN | 1 | |
| NAME | (create and attach a new disk)<br><br>lklnode02_DataDisk_1 | 任意の入力値 |

その他の設定はすべて稼働系と同じにしてください。

ネットワーク設定は、以下の値を使用します。

| **Networking** | | |
|---|---|---|
| **項目** | **入力もしくは選択する値** | **備考** |
| Public IP | (create new)<br><br>Name:lk4lnode02pubip<br><br>SKU:Basic | *注1 |

| | | |
|---|---|---|
| | Assignment:Static | |

**注1:** 本手順ではインターネット経由でアクセスするため、Public IP を設定しています。

その他の設定はすべて稼働系と同じにしてください。

2. 必要に応じて管理設定、詳細設定、タグ設定を行い、最後に **Review + create** をクリックします。
3. 内容を確認し、 **Create** をクリックします。
4. このあとネットワーク設定変更を行うため、一旦仮想マシンを停止します。
5. １つめのネットワークインターフェースを静的 IP アドレスに変更します。 **Azure Portal > Virtual machines > (仮想マシン名) > Networking >** （１つめのインターフェース名） をクリックします。
6. **IP configurations > ipconfig1** をクリックします。この時点では、IP アドレスの設定が動的 (Dynamic) になっています。
7. IP アドレスを静的に設定変更します。

   以下の値を使用します。

| 項目 | 入力もしくは選択する値 | 備考 |
|---|---|---|
| Assignment | Static | |
| IP address | 10.3.1.12 | |

8. **Save** をクリックして設定を保存します。
9. 次に、２つ目のネットワークインターフェースを追加します。 **Azure Portal > Virtual machines > (仮想マシン名) > Networking > Attach network interface** をクリックします。
10. **Create network interface** をクリックします。
11. 必要な値を入力したら、 **Create** をクリックします。

    以下の値を使用します。

| 項目 | 入力もしくは選択する値 | 備考 |
|---|---|---|
| Name | lk4lnode02-nic2 | |
| Subnet | 10.3.2.0/24 | *注1 |
| Private IP address assignment | Static | |
| Private IP address | 10.3.2.12 | |
| Network security group | None | |

- **注1:** 前の手順で作成した２つめのサブネットを、仮想マシンに対して設定します。

12. **OK** をクリックします。
13. インターフェースが２つアタッチされていることを確認します。
14. 仮想マシンを起動します。

# 12.2.5.3.3. クライアント兼 Witness サーバーの作成

1. 上記と同様の手順で、クライアント兼 Witness サーバーに使用する仮想マシンを作成します。

   基本設定として、主に以下の値を使用します。

| **Basics** | | |
|---|---|---|
| **項目** | **入力もしくは選択する値** | **備考** |
| **INSTANCE DETAILS** | | |
| Virtual machine name | lk4lclient | |
| Availability options | No Infrastructure redundancy required | *注1 |
| Image | Red Hat Enterprise Linux 7.8 | |
| Size | Standard_B1s | |
| **INBOUND PORT RULES** | | |
| Public inbound ports | Allow selected ports | |
| Select inbound ports | SSH | |

- **注1:** クラスターノード以外は Availability Set の設定は必須ではありませんが、今回のようにクライアントが Quorum に利用される場合、可用性を考慮して同一Availiavility Setに含めた上で、異なるフォールトドメイン・アップデートドメインとなるように設定することも検討してください。

   必要な値を入力したら、 **Next:Disks** をクリックします。

2. 次にディスク設定を入力します。

   今回はディスクの追加はしません。 **Next:Networking** をクリックします。

3. 次にネットワーク設定を入力します。

   以下の値を使用します。

| **Networking** | | |
|---|---|---|
| **項目** | **入力もしくは選択する値** | **備考** |
| Virtual network | lk4l-vnet | |
| Subnet | lk4l-nw01(10.3.1.0/24) | *注1 |

| | | |
|---|---|---|
| Public IP | (create new)<br>Name:lk4lclientpubip<br>SKU:Basic<br>Assignment:Static | *注2 |
| NIC network security group | Basic | |
| Public inbound ports | Allow selected ports | |
| Selected inbound ports | SSH | |

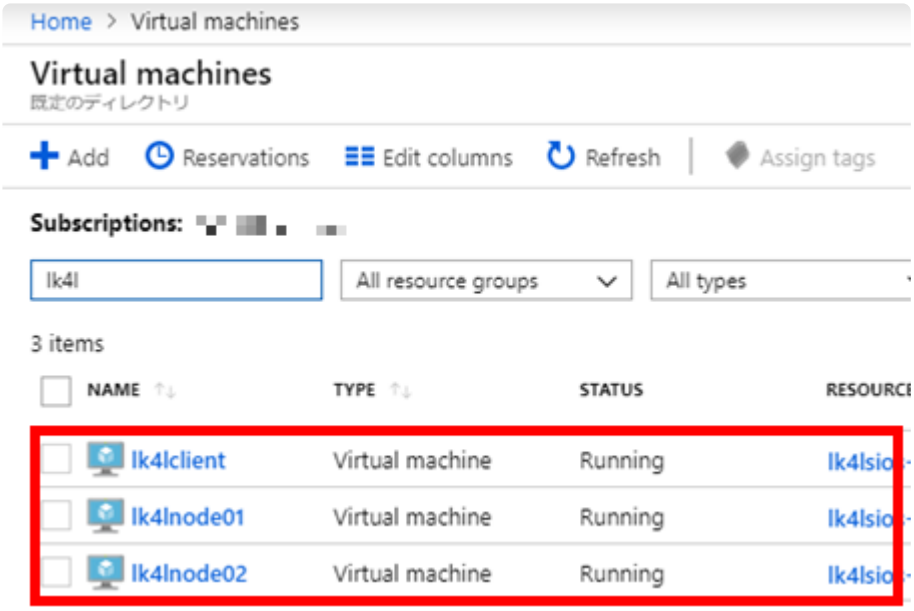
- **注1:** 前の手順で作成した１つめのサブネットを、仮想マシンに対して設定します。
- **注2:** 本手順ではインターネット経由でアクセスするため、Public IP を設定しています。

4. 必要に応じて管理設定、詳細設定、タグ設定を行い、最後に **Review + create** をクリックします。
5. 内容を確認し、 **Create** をクリックします。
6. 仮想マシンが起動したら、１つめのネットワークインターフェースを静的 IP アドレスに変更します。 **Azure Portal > Virtual machines > (仮想マシン名) > Networking >** （１つめのインターフェース名） をクリックします。
7. **IP configurations > ipconfig1** をクリックします。この時点では、IP アドレスの設定が動的 (Dynamic) になっています。
8. IP アドレスを静的に設定変更します

   以下の値を使用します。

| 項目 | 入力例 | 備考 |
|---|---|---|
| Assignment | Static | |
| IP address | 10.3.1.50 | |

9. **Save** をクリックして設定を保存します。
10. 以上で、三台の仮想マシンが作成されました。


Home > Virtual machines
Virtual machines
既定のディレクトリ
Add
Reservations
Edit columns
Refresh
Assign tags
Subscriptions:
lk4l
All resource groups
All types
3 items
NAME
TYPE
STATUS
RESOURCE
lk4lclient
Virtual machine
Running
lk4lnode01
Virtual machine
Running
lk4lnode02
Virtual machine
Running

# 12.2.5.4. ロードバランサーの作成

本環境で使用するロードバランサーを作成します。

Azure 上では、そのままでは LifeKeeper で作成した仮想 IP による通信が行えません。仮想 IP を実際に機能させるためには、ロードバランサーが必要です。

Azure のロードバランサーには外部ロードバランサーと内部ロードバランサーが存在しますが、今回の用途では、内部ロードバランサーを使用します。また、この内部ロードバランサーの IP アドレスの値は、後に設定する仮想 IP アドレスの値としても使用され、保護対象サービスのアクセスポイントとなります。

1. **Azure Portal > Load balancers > Create** を選択し、本環境で使用するロードバランサーを作成します。必要な値を入力し、 **Review + Create** をクリックします。

2. 以下の値を使用します。

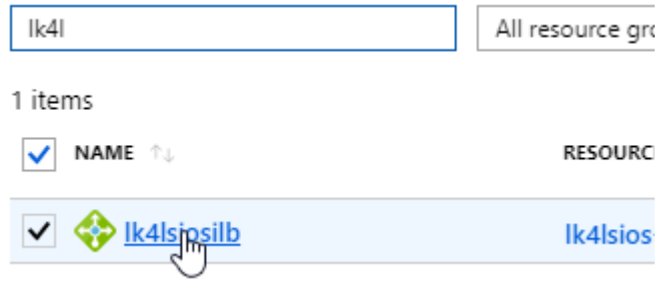
| 項目 | 入力もしくは選択する値 | 備考 |
|---|---|---|
| **Basics** | | |
| Name | lk4lsiosilb | |
| Type | Internal | *注1 |
| SKU | Basic | |
| **Frontend IP configuration** | | |
| Virtual network | lk4l-vnet | |
| Subnet | lk4l-nw01(10.3.1.0/24) | |
| IP address assignment | Static | |
| Private IP address | 10.3.1.200 | *注2 |

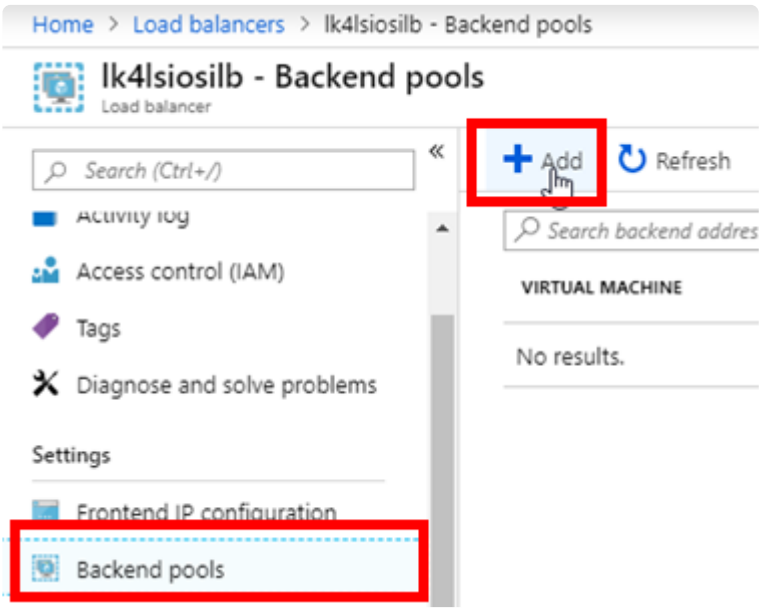
- **注1:** 内部ロードバランサーを作成するため、必ず「Internal」を選択します。

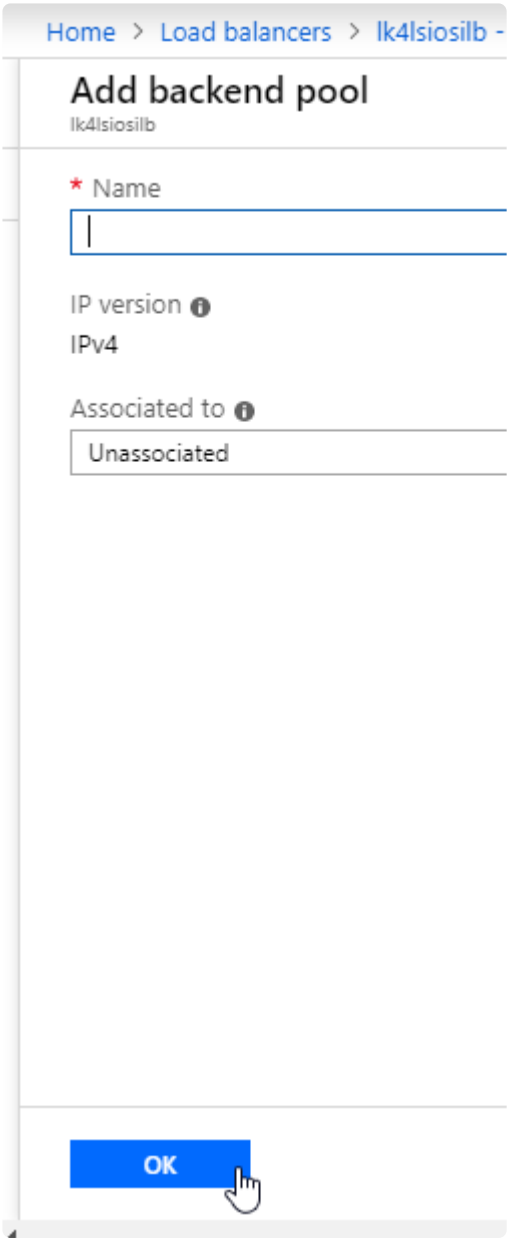
- **注2:** この値が、後に設定する仮想 IP アドレスの値としても使用され、保護対象サービスのアクセスポイントとなります。

3. 内容を確認し、 **Create** をクリックします。

4. 次に、作成済のクラスター用の仮想マシン 2 台を内部ロードバランサーのバックエンドプールに追加します。 **Azure Portal > Load balancers >** （ロードバランサー名） **> Backend pools > Add** を選択します。


lk4l
All resource gr
1 items
NAME
RESOURC
lk4lsiosilb
lk4lsios


Home > Load balancers > lk4lsiosilb - Backend pools
lk4lsiosilb - Backend pools
Load balancer
Search (Ctrl+/)
Add
Refresh
Search backend addres
VIRTUAL MACHINE
No results.
Access control (IAM)
Tags
Diagnose and solve problems
Settings
Frontend IP configuration
Backend pools

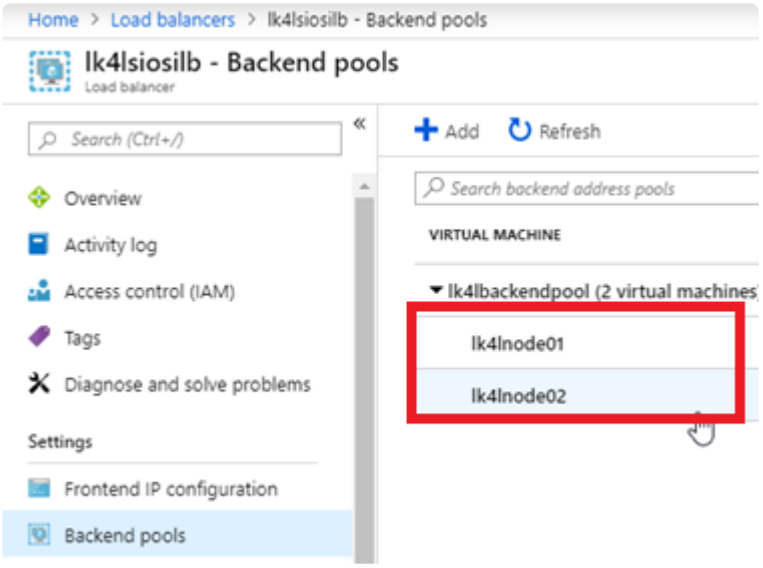



以下の値を使用します。

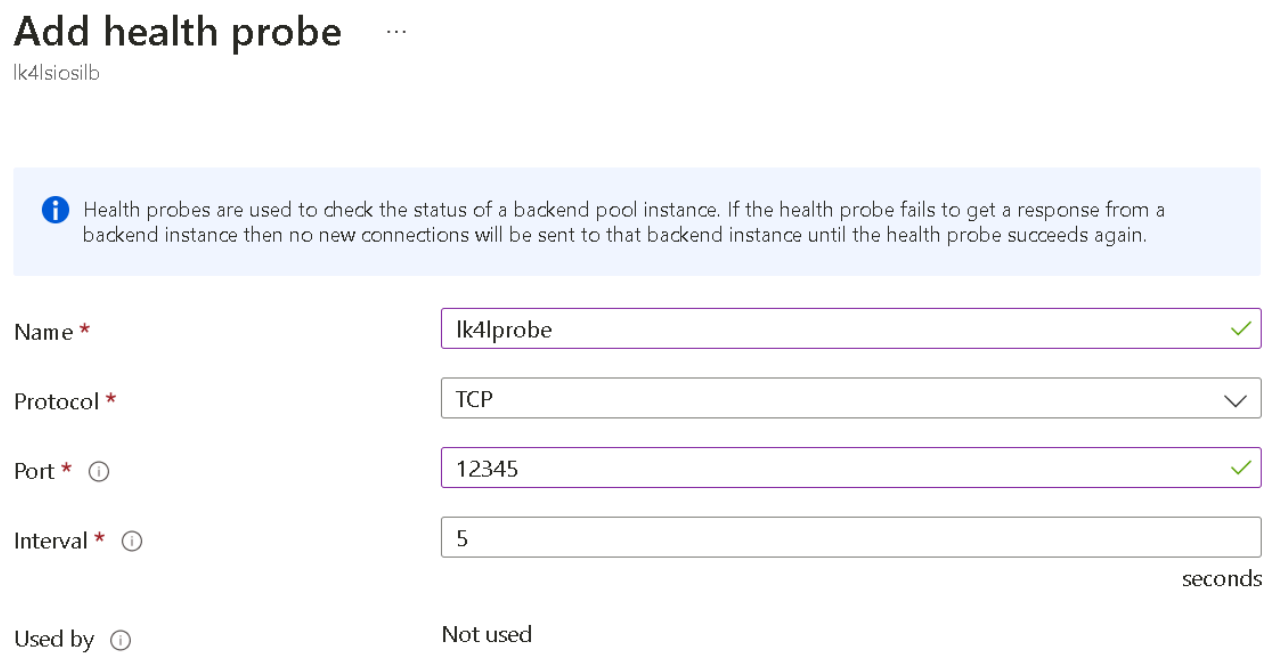
| 項目 | 入力例 | 備考 |
|---|---|---|
| Name | lk4lbackendpool | |
| Associated to | Virtual machines | |
| IP Version | IPv4 | |
| **Virtual machines** | | |
| lk4lnode01 | ipconfig1(10.3.1.11) | |
| lk4lnode02 | ipconfig1(10.3.1.12) | |

必要な値を入力したら、 **Add** をクリックします。

5. バックエンドプールが作成されたことを確認します。




6. 続けて、プローブを設定します。*Azure Portal* > **Load balancers** > （ロードバランサー名） > **Health Probes** > **Add** をクリックします。



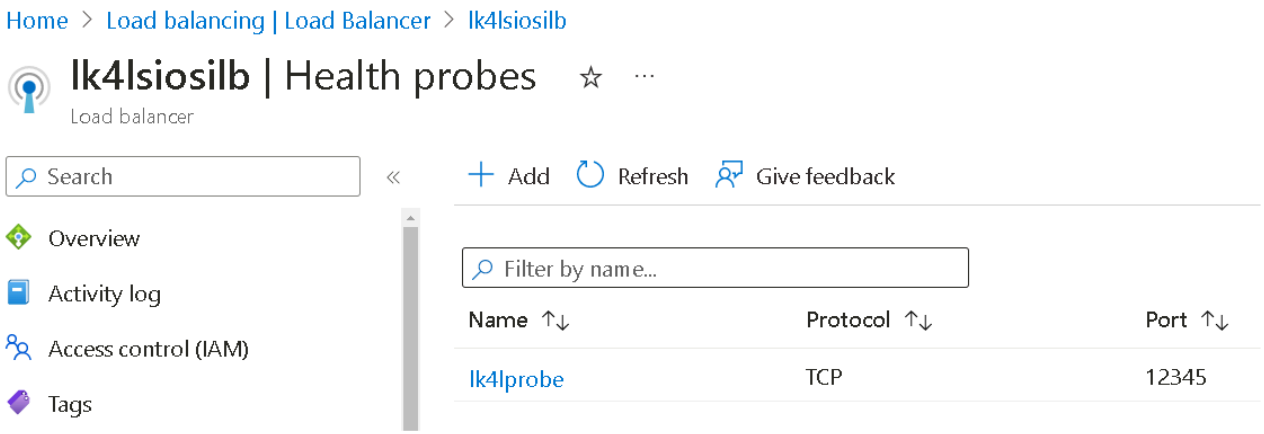

以下の値を使用します。

| 項目 | 入力例 | 備考 |
|---|---|---|
| Name | lk4lprobe | |
| Protocol | TCP | *注1 |
| Port | 12345 | *注1 |

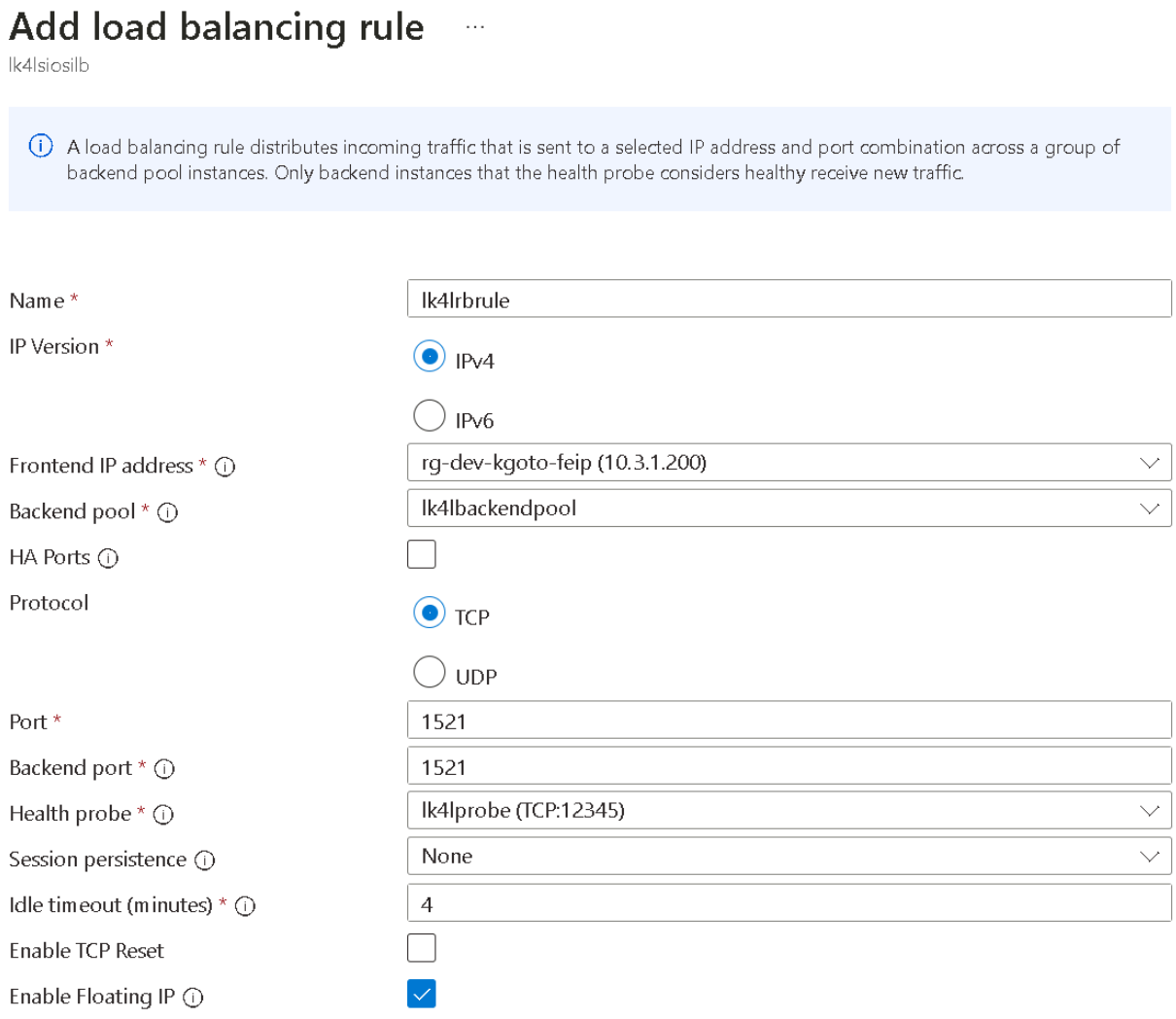
- **注1:** プローブ用には、LB Health Check が使用するプロトコルおよびポートを指定します。

7. 必要な値を入力したら、 **Add** をクリックします。

8. プローブが作成されたことを確認します。




9. 続けて、負荷分散規則を設定します。 **Azure Portal > Load balancers >** （ロードバランサー名） **> Load balancing rules > Add** をクリックします。

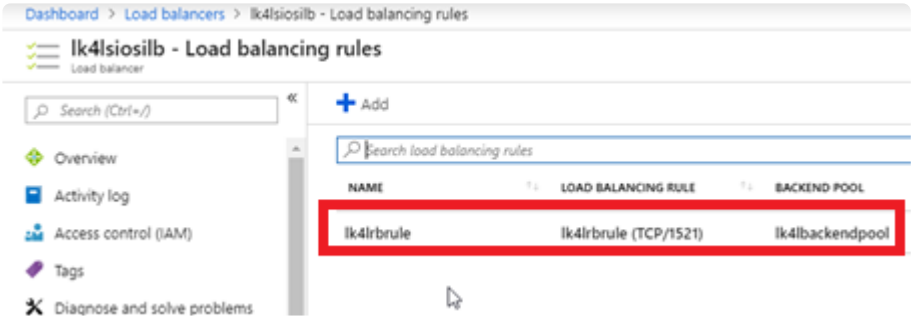



以下の値を使用します。

| 項目 | 入力例 | 備考 |
|---|---|---|
| Name | lk4lrbrule | |
| IP Version | IPv4 | |
| Frontend IP Address | 10.3.1.200 | |
| Protocol | TCP | |
| Port | 1521 | *注1 |
| Backend port | 1521 | *注1 |
| Backend pool | lk4lbackendpool | |
| Health probe | lk4lprobe | |
| Floating IP | Enabled | *注2 |

- **注1:** 負荷分散用のポートには、Oracle リスナーが使用するポートを指定します。
- **注2:** クライアントからクラスターへ向けた通信は、IP リソース(VIP）宛のパケット通信とする必要があるため、Floating IP address は必ず有効にしてください。

10. 必要な値を入力したら、 **OK** をクリックします。
11. 負荷分散規則が作成されていることを確認します。




以上でロードバランサーの設定は完了です。

# 12.2.5.5. OSの設定

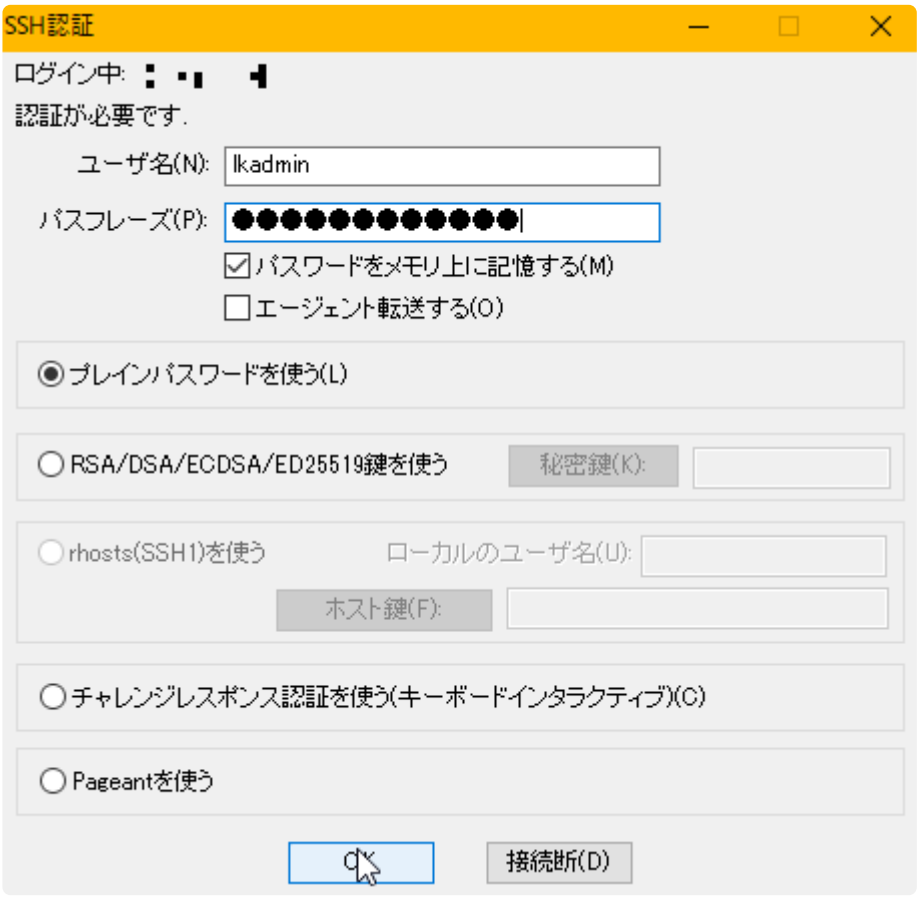
作成した 3 台の仮想マシンに対して OS 設定を行い、LifeKeeper がインストールできるようにします。

## 仮想マシンへのログイン

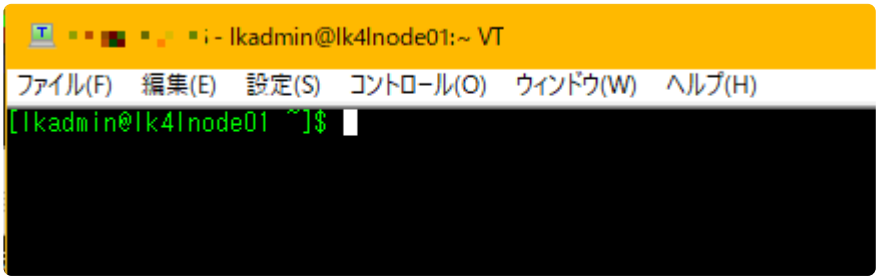
両クラスターノードおよびクライアント で、以下の設定を行います。

1. **Azure Portal** > **Virtual machines** > **Overview** > **Connect** を選択し、接続に必要な情報を確認します。

2. ローカル PC でターミナルソフトウェア (TeraTerm 等) を使用し、仮想マシン作成時に作成したアカウントで、仮想マシンの Public IP に SSH アクセスします。




3. 仮想マシンにログインできました。



4. 次に、以降 root が使用できるよう設定します。root 権限に変更します。

```
$ sudo su -
```

5. root パスワードを設定します。

```
# passwd
```

6. root ログインを許可します。

```
# cp /etc/ssh/sshd_config /etc/ssh/sshd_config.org
# vi /etc/ssh/sshd_config
<略>
PermitRootLogin yes   //コメントアウトを外す
<略>
#systemctl restart sshd
```

ターミナルソフトウェアで別ウィンドウを起動し、root でログインできることを確認します。
（ 今まで使用していたターミナルウィンドウは、念のため閉じずに実施してください。 ）

ログインに成功したら、これ以降の作業は、すべて root で実施します。

## NIC アドレスの固定化

Azure インフラストラクチャー更新に伴って NIC アドレスが更新されることを防ぐため、NIC アドレスを固定します。 両クラスターノード で、以下の設定を行います。

1. 仮想マシンの NIC 情報を表示し、各インターフェース名と MAC アドレスを控えます。

```
#ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.3.1.11  netmask 255.255.255.0  broadcast 10.3.1.255
        inet6 fe80::20d:3aff:fe50:42bf  prefixlen 64  scopeid 0x20<link>
        ether 00:0d:3a:50:42:bf  txqueuelen 1000  (Ethernet)
<omitted>
eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.3.2.11  netmask 255.255.255.0  broadcast 10.3.2.255
        inet6 fe80::e3c9:ec16:62a:bbd0  prefixlen 64  scopeid 0x20<link>
        ether 00:0d:3a:51:fc:01  txqueuelen 1000  (Ethernet)
<omitted>
```

2. デフォルトでは eth0 のネットワーク設定ファイルしかないため、eth1 のネットワーク設定ファイルを作成します。

```
#cp /etc/sysconfig/network-scripts/ifcfg-eth0 /etc/sysconfig/network-scripts/ifcfg-eth1
```

3. eth0、eth1 それぞれの設定ファイルを編集します。HWADDR には、前の手順で控えた値を使用します。

```
#vi /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
ONBOOT=yes
<略>
HWADDR=00:0d:3a:50:42:bf   //追記

#vi /etc/sysconfig/network-scripts/ifcfg-eth1
DEVICE=eth1    //変更
ONBOOT=yes
<略>
HWADDR=00:0d:3a:51:fc:01   //追記
```

# SELinux

SELinux を無効化します。 両クラスターノード で、以下の設定を行います。

1. ブートローダーを設定して、カーネルコマンドラインに `selinux=0` を追加します。

```
# grubby --update-kernel ALL --args selinux=0
```

2. システムを再起動します。

```
# reboot
```

3. 再起動後、 `getenforce` コマンドが “Disabled” を返すことを確認します。

```
# getenforce
Disabled
```

## ファイアーウォール

LifeKeeper は特定のポートを使用するため、 ファイアーウォールを使用した状態での LifeKeeper の実行 を参考に、OS のファイアーウォールの設定を変更します。
今回はファイアーウォールを無効化します。 両クラスターノードおよびクライアント で、以下の設定を行います。

- 以下のコマンドを実行します。

```
# systemctl stop firewalld.service
# systemctl disable firewalld.service
```

# アドレス変換

ここでは、以下の設定を実施します。

- IP リソースの監視処理が正常に行えるよう、ICMP パケットのソースアドレスを [Azureのプライベート IP アドレス] に変換します。

両クラスターノード で、以下の設定を行います。

1. iptables パッケージをインストールします。

```
# yum -y install iptables-services
```

2. iptables を有効にします。

```
#systemctl start iptables.service
#systemctl enable iptables.service
```

3. iptables が有効であることを確認します。

```
#systemctl list-unit-files | grep iptables
iptables.service                                enabled
```

4. iptables の設定を行います。

   稼働系ノード（lk4lnode01）では以下のコマンドを実行します。

```
# iptables -t nat -A PREROUTING -p tcp --dport 1521 -j DNAT --to-destination
10.3.1.200:1521
# iptables -t nat -A POSTROUTING -p icmp -s 10.3.1.200 -j SNAT --to-source 10.3.1.11
# service iptables save
```

   待機系ノード（lk4lnode02）では以下のコマンドを実行します。

```
# iptables -t nat -A PREROUTING -p tcp --dport 1521 -j DNAT --to-destination
10.3.1.200:1521
# iptables -t nat -A POSTROUTING -p icmp -s 10.3.1.200 -j SNAT --to-source 10.3.1.12
# service iptables save
```

5. iptables の設定が追加されていることを確認します。以下は稼働系の例です。

```
#cat /etc/sysconfig/iptables
<略>
:POSTROUTING ACCEPT [16:960]
-A POSTROUTING -s 10.3.1.200/32 -p icmp -j SNAT --to-source 10.3.1.11
<略>
```

6. 設定を有効にするため、仮想マシンを再起動します。

## GUI とロードバランサーの接続設定

Azure 提供の仮想マシンは、デフォルトでは GUI 環境がインストールされていません。LifeKeeper GUI を操作できるよう、クラスターノード上に X11 環境、クライアント上に GUI 環境を設定します。

両クラスターノード で、以下の設定を行います。

1. GUI パッケージを導入します。以下のコマンドを実行します。

```
# yum -y groupinstall "X11"
```

2. パッケージの導入が完了したら、以下のコマンドを実行します。

```
# systemctl set-default graphical.target
```

3. 次に、サブネット内通信 10.3.1.0/24, 10.3.2.0/24 とロードバランサーのプローブのソース IP アドレス 168.63.129.16 からの通信の許可を設定します。 /etc/sysconfig/iptables ファイルを編集し、エントリーを [REJECT] の前に追加します。

```
<略>
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [4710:957955]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -s 10.3.1.0/24 -p tcp -j ACCEPT # 追記
-A INPUT -s 10.3.2.0/24 -p tcp -j ACCEPT # 追記
-A INPUT -s 168.63.129.16/32 -p tcp -j ACCEPT # 追記
-A INPUT -j REJECT --reject-with icmp-host-prohibited
<略>
```

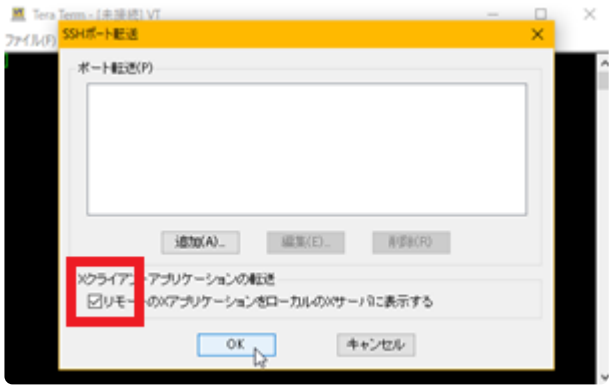
4. 設定を有効にするため、仮想マシンを再起動します。

```
# reboot
```

5. 続けて、ローカル PC 側の設定を行います。ローカル PC に X Server ソフトウェア (Xming 等) をインストールし、起動します。

6. 次に、ローカル PC のターミナルソフトウェア (TeraTerm 等) に、X11 forwarding の設定を行います。

   設定方法はご利用のターミナルソフトウェアによって異なります。以下は、Teraterm での設定例です。




## 名前解決

両クラスターノードおよびクライアント で、以下の設定を行います。

1. /etc/hosts に、クラスターノード（稼働系）、クラスターノード（待機系）クライアント兼 Witness サーバー３台の情報を登録します。

```
#vi /etc/hosts
10.3.1.11 lk4lnode01    //追記
10.3.1.12 lk4lnode02    //追記
10.3.1.50 lkclient    //追記
```

2. それぞれが、相互にホスト名で通信できることを確認します。

```
[root@lk4lnode01 ~]# ping lk4lnode02 -c 4
[root@lk4lnode01 ~]# ping lk4lclient -c 4
```

```
[root@lk4lnode02 ~]# ping lk4lnode01 -c 4
[root@lk4lnode02 ~]# ping lk4lclient -c 4
```

```
[root@lk4lclient ~]# ping lk4lnode01 -c 4
[root@lk4lclient ~]# ping lk4lnode02 -c 4
```

# カーネルバージョン確認する

確認の為、RHEL のカーネルバージョンを調べます。

```
# uname -r
3.10.0-862.11.6.el7.x86_64
# cat /etc/redhat-release
Red Hat Enterprise Linux Server release 7.5 (Maipo)
```

LifeKeeper をサポートする各ディストリビューションについては、 LifeKeeper for Linux サポートマトリックス を参照してください。

以上で LifeKeeper 導入のための事前準備は完了です。

# 12.2.6. LifeKeeper による HA クラスターの構築

仮想マシンの OS に LifeKeeper をインストールし、 HA クラスターを構築します。

## LifeKeeper のインストールと起動

1. 事前に、scp コマンド等を使用して仮想マシン上に LifeKeeper インストールイメージおよびライセンスファイルをアップロードしておきます。

2. 両クラスターノードに LifeKeeper をインストールします。 オンラインマニュアル 等をご参照ください。

- 「Use Quorum / Witness Functions」を選択します。
- 「LifeKeeper Authentication」では、rootを入力します。
- 「Recovery Kit Selection Menu」では、「LifeKeeper Oracle RDBMS Recovery Kit」と「DataKeeper for Linux」を選択します。
- 「LifeKeeper Startup After Install」選択します。

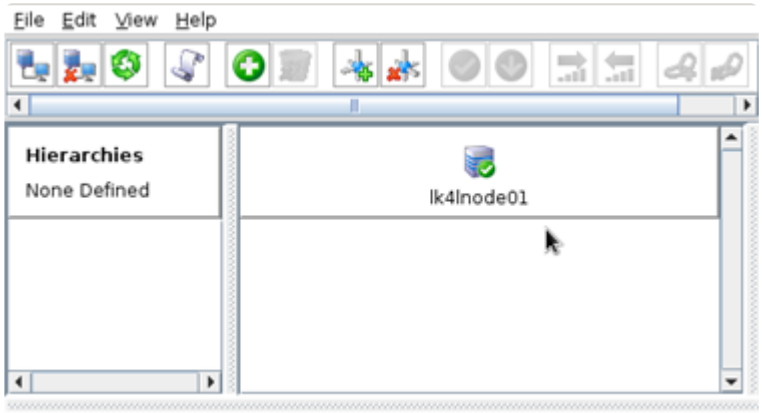
3. インストール完了後、シェルの環境変数に LifeKeeper コマンドパスを追加します。

```
# vi /root/.bash_profile
PATH=$PATH:/opt/LifeKeeper/bin MANPATH=$MANPATH:/opt/LifeKeeper/man
export PATH MANPATH
# source /root/.bash_profile
```

4. LifeKeeper GUI を起動します。

```
# lkGUIapp &
```

LifeKeeper が起動しました。


File Edit View Help
Hierarchies
None Defined
lk4lnode01
<-- lk4lnode01: Adding app/res: scsi netraid

# 12.2.6.1. コミュニケーションパスの作成

LifeKeeperは設定した全てのコミュニケーションパスを通じ、ハートビートの応答確認を行うことによってノードの死活監視を実施しています。ハートビート通信において所定の回数応答が得られなかった場合、そのコミュニケーションパスの接続が切れていると判断しステータスをDEADとします。全てのコミュニケーションパスのステータスがDEADとなった場合、ノード障害と判定してフェイルオーバーを開始します。

LifeKeeperではサーバー間に2経路以上のコミュニケーションパスを推奨しているため、今回は以下のように構成します。

| 種別 | プライオリティ | 稼働系 | 待機系 |
|---|---|---|---|
| TCP | 1 | 10.3.1.11 | 10.3.1.12 |
| TCP | 2 | 10.3.2.11 | 10.3.2.12 |

コミュニケーションパスの作成方法は、コミュニケーションパスの作成 をご参照ください。

# 12.2.6.2. Quorum / Witness の設定

スプリットブレインの発生を抑制するため、クライアント上に Quorum / Witness を設定します。Quorum / Witnessの設定につきましては、以下を参照のうえ、お客様の要件にあったモードを選んで設定を行ってください。

Quorum / Witness

Quorum パラメーター一覧

Quorum / Witness Server Kit のご利用について

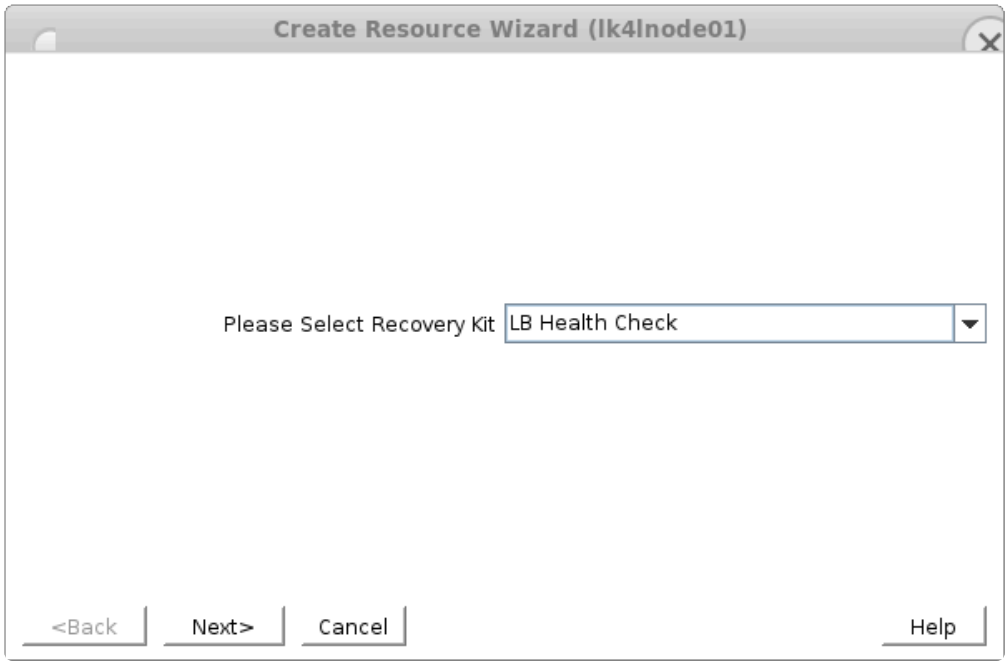
# 12.2.6.3. LB Health Checkリソース階層の作成

LB Health Check Kit は、クラウド環境でLoad Balancerの負荷分散対象インスタンスへのヘルスチェックプローブを待ち受けて応答する仕組みを提供します。

## LB Health Check リソース階層の作成

次に LB Health Check のリソースを作成します。

1. リソース階層の作成 を参考に [Create Resource Wizard] を起動します。
2. Recovery Kit として [LB Health Check] を選択します。



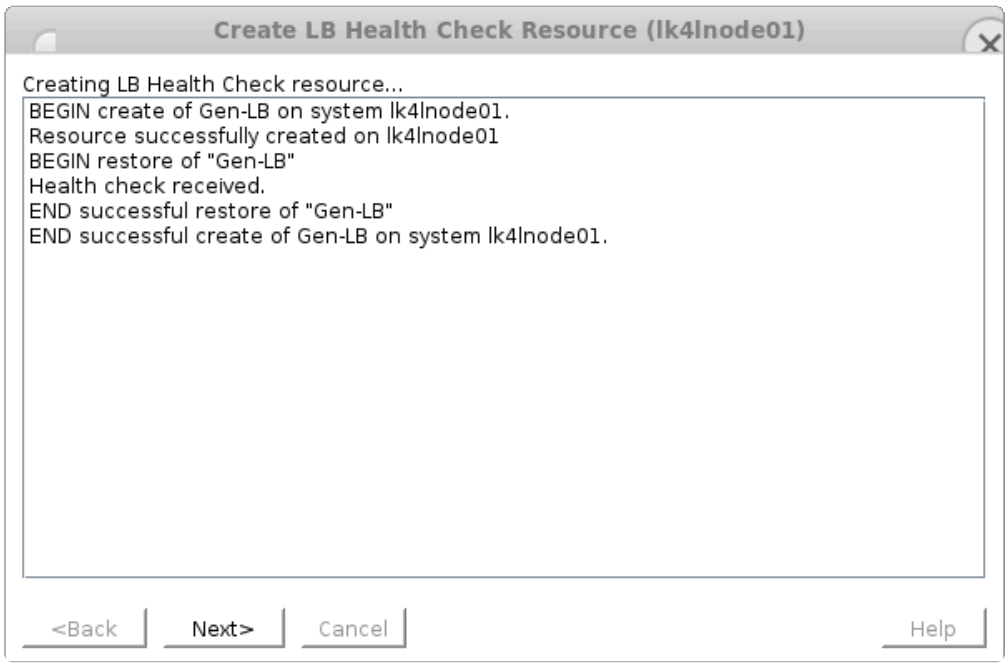

3. ウィザードで、以下の値を入力します。

| 項目 | 値 |
|---|---|
| Switchback Type | Intelligent |
| Server | lk4lnode01 |
| Reply daemon Port | 12345 |
| Reply daemon message | |
| LB Health Check Resource Tag | Gen-LB (注**1**) |

(注**1**) LB Health Check Resource Tag はデフォルトで lbhc-<port> になります。ここでは他のサンプルに合わせて Gen-LB タグを使用しています。

4. リソース作成が開始され、「End successful create of...」と表示されれば成功です。[Next]をクリッ

クして進めてください。




5. Pre – Extendウィザードで、以下の値を入力します。

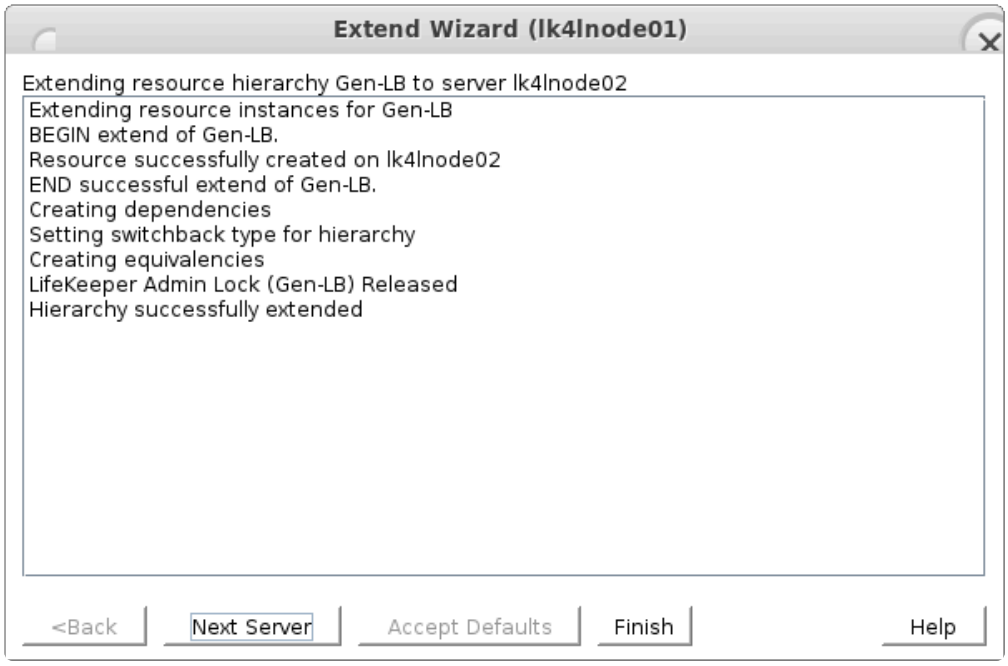
| 項目 | 値 |
|---|---|
| Target Server | lk4lnode02 |
| Switchback Type | intelligent |
| Template Priority | 1 |
| Target Priority | 10 |

6. Pre-Extendが開始され、「Pre Extend checks were successful」と表示されれば成功です。[Next]をクリックして進めてください。

7. Extendウィザードで、以下の値を入力します。

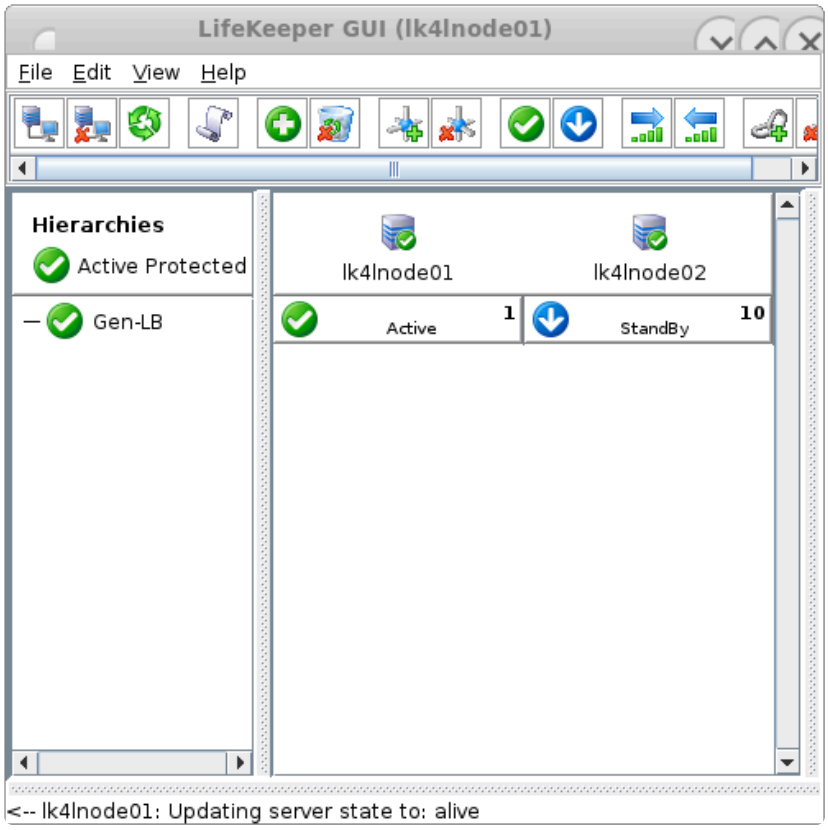
| 項目 | 値 |
|---|---|
| LB Health Check Resource Tag | Gen-LB |

8. Extendが開始され、「Hierarchy successfully extend」と表示されれば成功です。[Finish]をクリックします。




9. [Done]で完了させます。

10. 完成したリソースです。


LifeKeeper GUI (lk4lnode01)
File Edit View Help
Hierarchies
Active Protected
Gen-LB
lk4lnode01
lk4lnode02
Active 1
StandBy 10
<-- lk4lnode01: Updating server state to: alive

# 12.2.6.4. IP リソースの作成

LifeKeeper GUI で IP リソースを作成します。

1. IPパラメーターの設定
   Azureの仮想マシン環境では、IPリソース作成直後Broadcast Pingへの応答が得られません。その為、Broadcast Pingによる死活監視を両クラスターノードで無効にします。また、IPリソースに設定する仮想IPアドレスはロードバランサーに設定しているIPアドレスと同じです。その為、同一IPアドレスがネットワーク上に存在するかチェックする機能を無効にします。
   以下のファイルを編集します。

```
# vi /etc/default/LifeKeeper
<略>
NOBCASTPING=1     //初期値 0 から変更
NOIPUNIQUE=1      //初期値 0 から変更
```

2. LifeKeeper GUI 管理画面より **Create Resource Hierarchy** を選択します。

3. Create Resource ウィザードに沿って設定します。今回は以下の値を使用します。

| 項目 | 入力もしくは選択する値 | 備考 |
|---|---|---|
| Select Recovery Kit | IP | |
| Switchback Type | Intelligent | |
| IP Resource | 10.3.1.200 | ※注１ |
| Netmask | 255.255.255.255 | |
| Network Interface | eth0 | |
| IP Resource Tag | ip-10.3.1.200 | |

- 注１： 設定済の Azure ロードバランサーのフロントエンド IP と同じ値を使用します。

Extend 時には、以下の値を使用します。

| 項目 | 入力もしくは選択する値 | 備考 |
|---|---|---|
| Target Server | lk4lnode02 | |
| Switchback Type | Intelligent | |

| | | |
|---|---|---|
| Template Priority | 1 | |
| Target Priority | 10 | |
| IP Resource | 10.3.1.200 | ※注１ |
| Netmask | 255.255.255.255 | |
| Network Interface | eth0 | |
| IP Resource Tag | ip-10.3.1.200 | |

- 注１： 設定済の Azure ロードバランサーのフロントエンド IP と同じ値を使用します。

4. IP リソースが作成されました。

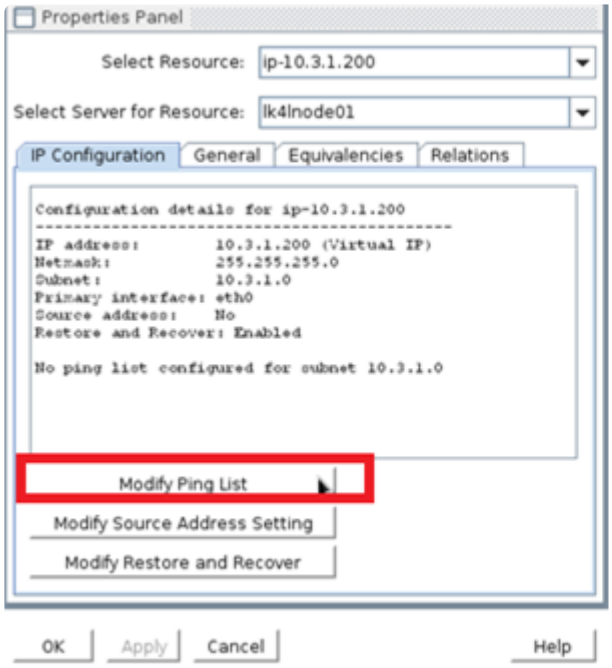
IP リソース階層の作成

5. 今回の環境では、Broadcast Ping による監視を無効にしているため、Unicast Ping を利用して死活監視を行います。以下に設定例を記載します。

6. ping応答が得られることを確認します。

```
# ping -I 10.3.1.200 10.3.1.50
```

7. IP リソースのプロパティ画面を表示し、 **Modify Ping List** をクリックします。




8. Ping List に設定する IP アドレスを入力し、 **Add address** をクリックします。

9. **Save List** をクリックし、設定値を保存します。

※Ping Listの更新は両ノードで実施してください。

# 12.2.6.5. Data Replication リソース階層の作成

ここでは Data Replication リソースの作成方法を解説します。

## ディスクの設定

まず、Data Replication リソースで使用するディスクの設定をします。

1. クラスターの両ノードで、oracle データベースのマウントポイント用ディレクトリを作成します。

```
# mkdir /mnt/ORA
```

2. 両ノードで、ディスク(インスタンス作成時に指定したディスク)がアタッチされていることを確認します。

```
#parted -l
 (略)
/dev/sdc: unrecognised disk label
Model: Msft Virtual Disk (scsi)
Disk /dev/sdc: 33.3GB
 (略)
Model: Msft Virtual Disk (scsi)
Disk /dev/sdd: 11.8GB
 (略)
```

3. 両ノードで、Oracle 用ディスクのパーティションを作成します。

```
#parted /dev/sdc
GNU Parted 3.1
Using /dev/sdc
..... (略)
```

4. 作成したパーティションを確認します。

```
# parted -l
(omitted)
Model: Msft Virtual Disk (scsi)
Disk /dev/sdc: 32.2GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags
 1      1049kB  32.2GB                       ora

Model: Msft Virtual Disk (scsi)
Disk /dev/sdd: 10.7GB
Sector size (logical/physical): 512B/512B
Partition Table: loop
Disk Flags:

Number  Start   End     Size    File system     Flags
 1      0.00B   10.7GB  linux-swap(v1)
```

5. 次に、DataKeeper がデバイスを認識できるように設定します。

   Azure 上で DataKeeper for Linux を使用してデータレプリケーションを行う場合、標準の設定ではストレージの識別に使用する ID を取得できないため、GUID Partition Table（GPT）を作成し一意の ID をパーティションに割り当てるか、LVM を使用する必要があります。詳細については、オンラインマニュアルの トラブルシューティング を参照してください。

   ディスク設定は以上です。

## Data Replication リソース階層の作成

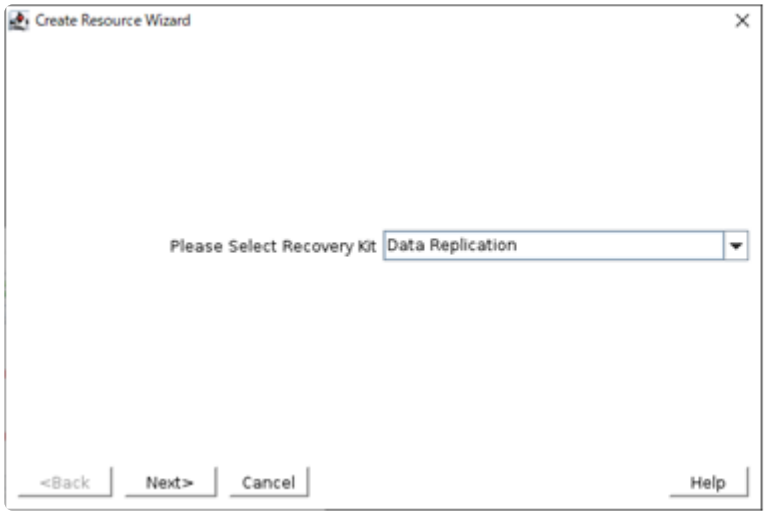
次に、LifeKeeper GUI で Data Replication リソースを作成します。

1. プライマリーノードで以下のコマンドで LifeKeeper GUI を起動します。

```
# lkGUIapp &
```

2. **Create Resource Hierarchy** のアイコンをクリックしてリソース作成を開始します。

3. Create Resource ウィザードが表示されます。プルダウンメニューで Recovery Kit に対し **Data Replication** を選択して、 **Next** をクリックします。

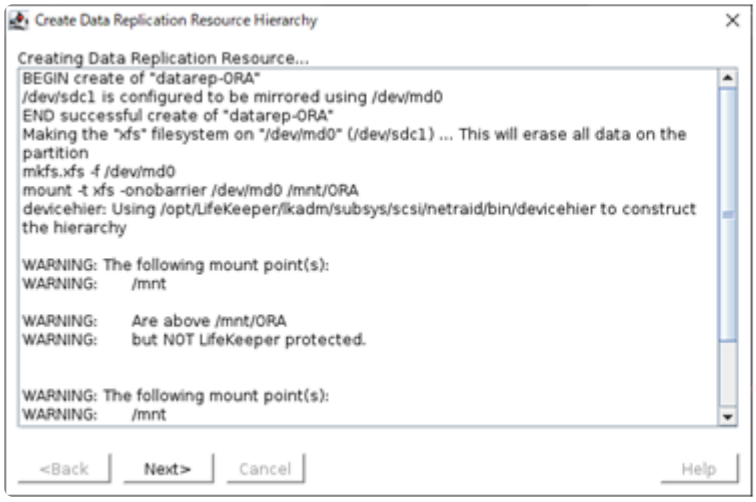



4. Create Resource ウィザードで、以下の値を入力します。

| 項目 | 入力もしくは選択する値 | 備考 |
|---|---|---|
| Switchback Type | Intelligent | |
| Server | lk4lnode01 | |
| Hierarchy Type | Replicate New Filesystem | |
| Source DIsk | /dev/sdc1 | ※注１ |
| New Mount Point | /mnt/ORA | ※注２ |
| New Filesystem Type | xfs | |
| Data Replication Resource Tag | datarep-ORA | |
| File System Resource Tag | /mnt/ORA | |
| Bitmap File | /opt/LifeKeeper/bitmap_mnt_ORA | ※注３ |
| Enable Asynchronous Replication? | no | |

- ※注１ ：この後「 ATTENTION! <デバイス名> is not shareable with any other server 」の表示が出力されますが、レプリケーションを作成する場合は必ず出力されるため、無視して問題ありません。
- ※注２ ：作成済のマウントポイントを指定します。
- ※注３ ：今回はデフォルトのパスを設定していますが、パフォーマンス向上のため、より高速な専用領域を指定することも検討してください。

5. Data Replication リソースの作成が開始されます。
6. 「End of successful Create of...」と表示されれば成功です。 **Next** をクリックして Pre-Extend ウィザードへ遷移します。




7. 以下の値を入力します。

| 項目 | 入力もしくは選択する値 |
|---|---|
| Target Server | lk4lnode02 |
| Switchback Type | Intelligent |
| Template Priority | 1 |
| Target Priority | 10 |

8. Pre-Extend が開始されます。「Pre Extend checks were successful」と表示されれば成功です。**Next** をクリックして次に進みます。

9. Extend Resource ウィザードで、以下の値を入力します。

| 項目 | 入力もしくは選択する値 | |
|---|---|---|
| Mount Point | /mnt/ORA | |
| Root Tag | /mnt/ORA | |

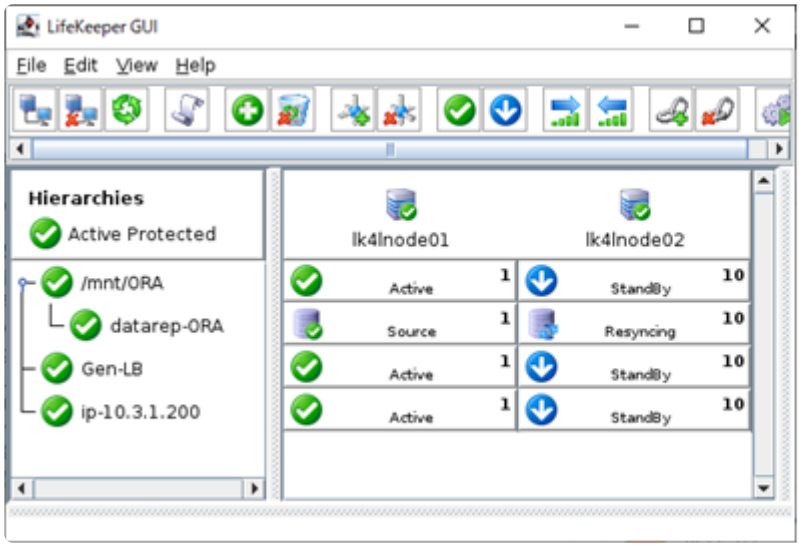
| | | |
|---|---|---|
| Target Disk | /dev/sdc1 | |
| Data Replication Resource Tag | datarep-ORA | |
| Bitmap File | /opt/LifeKeeper/bitmap_mnt_ORA | |
| Replication Path | 10.3.2.11/10.3.2.12 | ※注１ |

- ※注１ ：２つめのサブネット上のコミュニケーションパスを指定します。

10. Extend が開始されます。 **Hierarchy successful extended** と表示されれば成功です。 **Finish** をクリックします。

11. **Done** を押下し、ウィザードを終了します。

Data Replication リソースが作成されました。




※Swap 領域に対しては Data Replication リソースを作成しません。

参考： DataKeeper リソース階層の作成

# 12.2.6.6. Oracle リソース階層の作成

Oracle Database 19c をインストールのための環境設定を行います。Oracle インストールの前提条件については、Oracle 社のドキュメントをご参照ください。

両クラスターノード で、以下の設定を行います

1. swap 領域を作成します。

   Oracle をインストールするには 3 GB 以上の Swap 領域が必要です。インスタンスにアサイン済の 10GB のディスク（/dev/sdd）を使用して Swap エリアを構成します。

```
# mkswap /dev/sdd
# swapon /dev/sdd
# free
              total        used        free      shared  buff/cache   available
Mem:        1706084      189248     1348300        9184      168536     1300564
Swap:      12582904           0    12582904
# blkid | grep /dev/sdd
/dev/sdd: UUID="268e56bd-c64f-4798-9197-e411a497ba11" TYPE="swap" //UUID を控える
# vi /etc/fstab
UUID=268e56bd-c64f-4798-9197-e411a497ba11 swap swap defaults 0 0     ←追加
```

2. RHEL のオプションレポジトリを登録します。必須パッケージ「compat-libstdc++-33」を含むリポジトリを追加します。
   ※以下は検証環境での例になります。適宜読み替えてください。

```
# yum-config-manager --enable rhui-rhel-7-server-rhui-optional-rpms
```

3. 必須パッケージをインストールします。

```
# yum -y install binutils compat-libcap1 compat-libstdc++-33 compat-libstdc++-33.i686 gcc gcc-c++ glibc glibc.i686 glibc-devel glibc-devel.i686 ksh libgcc libgcc.i686 libstdc++ libstdc++.i686 libstdc++-devel libstdc++-devel.i686 libaio libaio.i686 libaio-devel libaio-devel.i686 libXext libXext.i686 libX11 libX11.i686 libxcb libxcb.i686 libXi libXi.i686 make sysstat
```

4. カーネルパラメータを編集します。

   注意： 記号、空白はワード、エクセルの仕様で勝手に異なる文字に変更されてコピーペーストでコ

マンド実行すると、エラーになる場合があります。また、末尾に空白を絶対につけないでください。変数が数値ではなく文字列に認識されてしまいまい、コマンドが失敗する場合があります。

```
# MEMTOTAL=$(free -b | sed -n '2p' | awk '{print $2}')
# SHMMAX=$(expr $MEMTOTAL / 2)
# SHMMNI=4096
# PAGESIZE=$(getconf PAGE_SIZE)
# cat >> /etc/sysctl.conf << EOF
>fs.aio-max-nr = 1048576
>fs.file-max = 6815744
>kernel.shmmax = $SHMMAX
>kernel.shmall = `expr ¥( $SHMMAX / $PAGESIZE ¥) ¥* ¥( $SHMMNI / 16 ¥)`
>kernel.shmmni = $SHMMNI
>kernel.sem = 250 32000 100 128
>net.ipv4.ip_local_port_range = 9000 65500
>net.core.rmem_default = 262144
>net.core.rmem_max = 4194304
>net.core.wmem_default = 262144
>net.core.wmem_max = 1048576
>EOF
```

5. カーネルに反映させます。

```
#   sysctl -p
```

6. Oracle 専用のユーザー/グループを作成し、システムの環境設定をします。

```
# i=54321; for group in oinstall dba backupdba oper dgdba kmdba; do groupadd -g $i
$group; i=`expr $i + 1` ;done
# useradd -u 1200 -g oinstall -G dba,oper,backupdba,dgdba,kmdba -d /home/oracle
```

7. oracle ユーザーのパスワードを設定します.

```
# passwd oracle
Changing password for user oracle.
New password:
Retype new password:
Passwd: all authentication tokens updated successfully.
```

8. */mnt/ORA* が DK により保護されておりプライマリーノードでマウントされていることを確認します。 (この作業はプライマリーノードのみ行います)

```
# df
Filesystem     1K-blocks    Used Available Use% Mounted on
<<中略>>
/dev/md0       31864752  7396580  22842880  25% /mnt/ORA
<<後略>>
#lcdstatus -e
lk4lnode02   LSNR.LISTENER   dblistener-11079   ISP     1  lk4lnode01
lk4lnode02   /mnt/ORA       /mnt/ORA           ISP     1  lk4lnode01
lk4lnode02    datarep-ORA   /dev/sdc           ISP     1  lk4lnode01
lk4lnode02   ip-10.3.1.200 IP-10.3.1.200      ISP     1  lk4lnode01
<<後略>>
```

9. ディレクトリ作成とアクセス権、オーナーを変更します。（この作業はプライマリーノードのみ行います）

```
# mkdir -p /mnt/ORA/app/oracle/product/19.3.0/dbhome_1
# chown -R oracle:oinstall /mnt/ORA
# chmod -R 775 /mnt/ORA
```

10. */etc/pam.d/login* 14 行目あたりに追記します。

```
# vi /etc/pam.d/login
session    required     pam_selinux.so open
session    required     pam_namespace.so
session    required     pam_limits.so  //追記
session    optional     pam_keyinit.so force revoke
```

11. */etc/security/limits.conf* の最終行に追記します。

```
# vi /etc/security/limits.conf
oracle soft nproc  2047  //追記
oracle hard nproc  16384  //追記
oracle soft nofile 1024  //追記
oracle hard nofile 65536  //追記
oracle soft stack  10240  //追記
oracle hard stack  32768  //追記
```

12. Oracle 用の環境設定をしておきます。作業は前の手順で作成した「oracle」ユーザーで行います。

```
# su oracle
$ vi ~/.bash_profile
//以下を最終行に追記
umask 022
export ORACLE_BASE=/mnt/ORA/app/oracle
export ORACLE_HOME=/mnt/ORA/app/oracle/product/19.3.0/dbhome_1
export PATH=$PATH:$ORACLE_HOME/bin
export ORACLE_SID=<SIDの名前>
export NLS_LANG=Japanese_Japan.AL32UTF8

$source ~/.bash_profile
```

ここで設定した ORACLE_SID は、後の手順で使用するため控えておきます。今回は “lkoracle” とします。

13. root の xauth の設定を oracle ユーザーにコピーします。（この作業はプライマリーノードのみ行います）。

```
# xauth list
# su oracle
$ xauth add <DisplayName> <ProtocolName> <Hexkey>
```

14. oracle ユーザーの xauth の出力結果が root と同じであることを確認します。

```
$ xauth list
```

15. インストール作業用ディレクトリ作成（この作業はプライマリーノードのみ行います）

```
$ mkdir /home/oracle/tmp
```

以上でインストールするための準備は完了です。

# 12.2.6.6.1. Oracle DB のインストール

ここからの作業はクラスターノード(プライマリー)で行います。Oracle のインストールについては、Oracle 社のドキュメントをご参照ください。

1. Oracle 19cのインストールパッケージをダウンロードし、$ORACLE_HOMEに解凍します。
2. su oracle で oracle ユーザーになります。
3. $ORACLE_HOME以下にあるインストーラーを起動します。

```
$ $ORACLE_HOME/runInstaller &
```

4. 以下の表を参考にOracleをインストールしてください。

| Step. | 項目 | 入力もしくは選択する値 |
|---|---|---|
| Configration Option | - | Set Up Software Only |
| Database Installation Option | - | Single instance database installation |
| Database Edition | - | Enterprise Edition |
| Installation Location | Oracle base | /mnt/ORA/app/oracle |
| | Software location | /mnt/ORA/app/oracle/product/19.3.0/dbhome_1 |
| Create Inventory | Inventory Directory | /mnt/ORA/app/oraInventory |
| | Orainventory Group Name | oinstall |
| Operating System Groups | OSDBA | dba |
| | OSOPER | oper |
| | OSBACKUPDBA | backupdba |
| | OSDGDBA | dgdba |
| | OSKMDBA | kmdba |
| | OSRACDBA | dba |
| Root script execution | - | 適宜、実行またはアカウント指定を行ってください。 |

5. Summaryを確認します。

問題なければそのまま[Install]を実行します。

6. インストールが開始されます。
   インストール完了後、[close]を押してインストーラーを終了します。

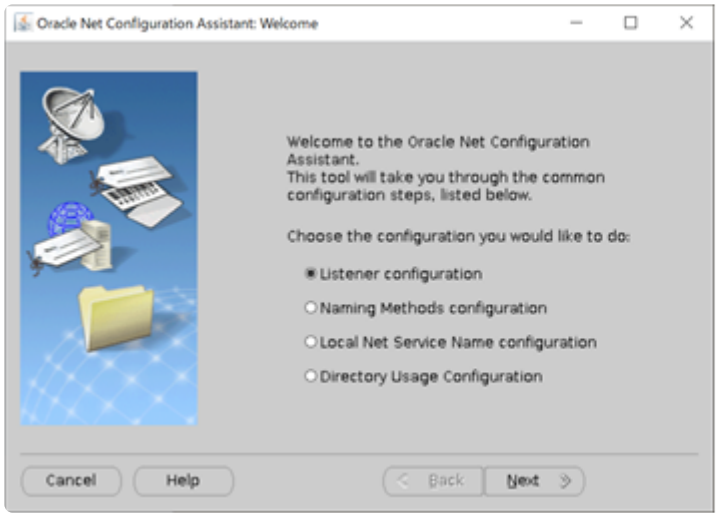
# 12.2.6.6.2. リスナー設定

リスナーの設定をします。クラスターノード(プライマリー)で、以下の設定を行います。

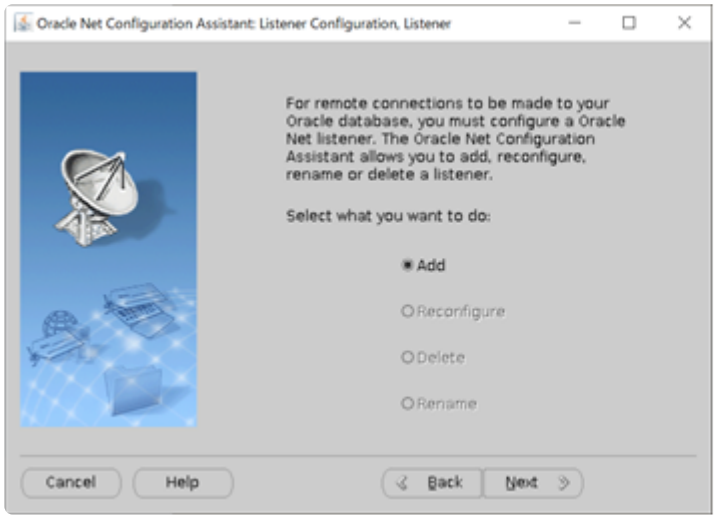
1. oracle ユーザーで以下のコマンドを実行します。

```
$ $ORACLE_HOME/bin/netca &
```

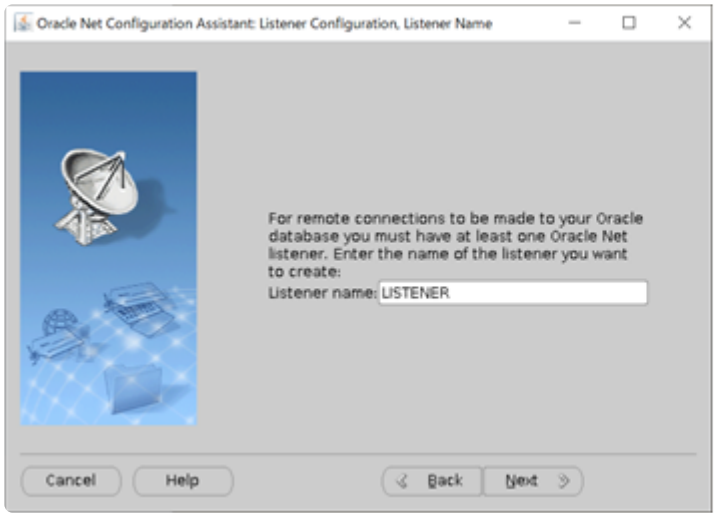
2. **Listener Configuration** を選択して **Next** をクリックします。




3. **Add** を選択して **Next** をクリックします。

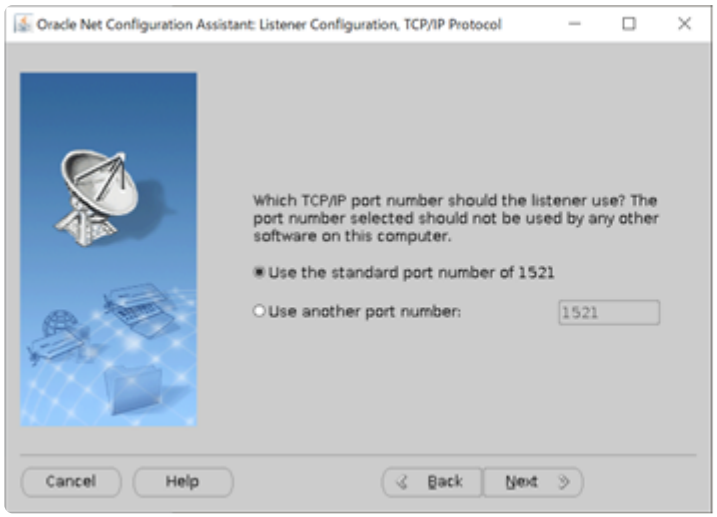



4. Listener の名前を指定して **Next** をクリックします。

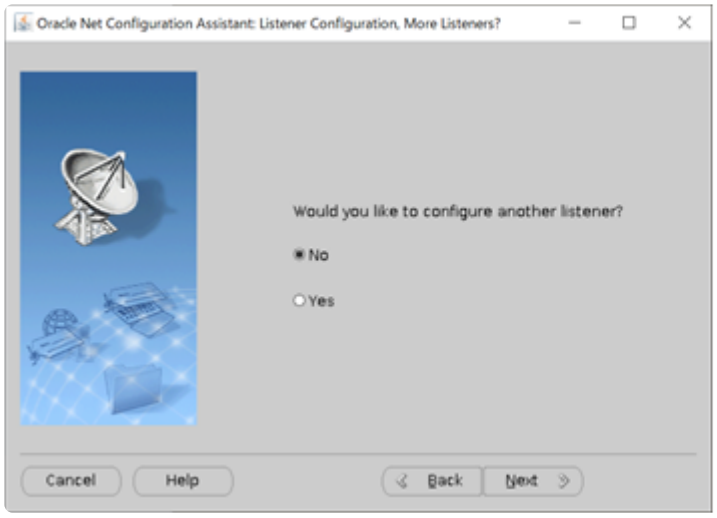



5. このまま **Next** をクリックします。

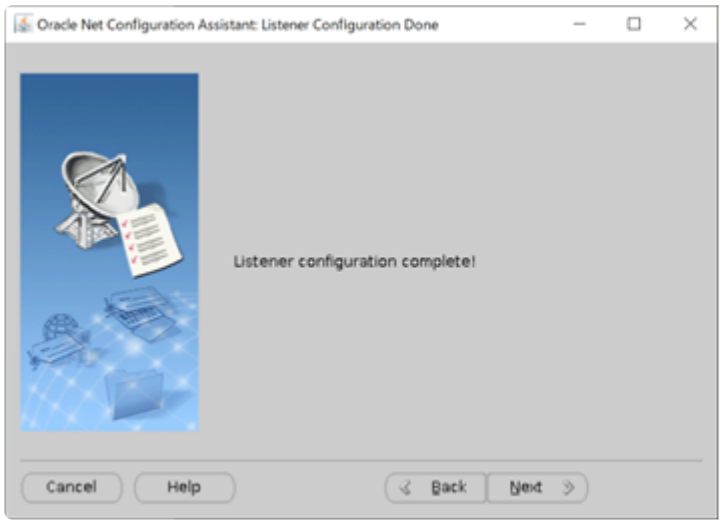
6. ポートはデフォルトの **1521** を選択して **Next** をクリックします。




7. Listener 設定を終了します。 **Next** をクリックします。




8. Listener 設定が完了しました。ダイアログ右上の **[X]** をクリックして完了します。




以上で Listener 設定は完了です。

# 12.2.6.6.3. DB の作成

クラスターノード(プライマリー)で、以下の設定を行います。

1. oracle ユーザーで以下のコマンドを実行して下さい

```
$ $ORACLE_HOME/bin/dbca &
```

2. 以下の表を参考にDatabaseを作成してください。

| Step. | 項目 | 入力もしくは選択する値 |
|---|---|---|
| Database Operation | - | Crate a database |
| Creation Mode | - | Advanced configuration |
| Deployment Type | Database type | Oracle Single Instance database |
| | Template | General Purpose or Transaction Processing |
| Database Identification | Global database name / SID | lkoracle |
| | PDB name | lkpdb |
| Storage Option | - | 適宜設定してください |
| Fast Recovery Option - | | |
| Network Configration | Listener selection | LISTENER |
| Data Vault Option | - | 適宜設定してください |
| Configuration Options | - | |
| Management Option | - | |
| User Credentials | - | |
| Creation Option | - | |

3. Summary

   設定に問題なければ、 **Finish** をクリックしてデータベース作成をスタートします。
   データベース作成完了後、ウィンドウを閉じます。

4. 続いて、/etc/oratabを修正します。なお、本設定は後の手順で元に戻します。

```
# vi /etc/oratab
oracle:/mnt/ORA/app/oracle/product/19.3.0/dbhome_1:Y　Nを Y に変更
```

5. */etc/oratab* をプライマリーノードからセカンダリーノードにコピーします

これで Oracle DB のインストールは完了です。
※セカンダリーノードへの Oracle DB インストールは不要です。

# 12.2.6.6.4. Oracle の設定

リスナーリソース作成前に、設定ファイルの修正が必要です。これは、リスナーに対して仮想 IP アドレスでアクセスするためです。

クラスターノード(プライマリー)で、以下の設定を行います。

1. *$ORACLE_HOME/network/admin/listener.ora* を修正します。

```
$vi $ORACLE_HOME/network/admin/listener.ora
<<中 omitted>>
        SID_LIST_LISTENER =
          (SID_LIST =
            (SID_DESC =
              (SID_NAME = LKORACLE)
            )
          )

        LISTENER =
          (DESCRIPTION_LIST =
            (DESCRIPTION =
              (ADDRESS = (PROTOCOL = TCP)(HOST = 10.3.1.200)(PORT = 1521))
            )
    )
```

2. *$ORACLE_HOME/network/admin/tnsnames.ora* を修正します。

```
$vi $ORACLE_HOME/network/admin/tnsnames.ora
LISTENER_LKORACLE =
  (ADDRESS = (PROTOCOL = TCP)(HOST = 10.3.1.200)(PORT = 1521))


LKORACLE =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 10.3.1.200)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = lkoracle)
    )
  )
```

3. リスナー経由のログインで使用する、パスワードファイルを作成します。パスワードファイルの記述形式については、Oracle 社の公式ドキュメントを参照してください。

```
$orapwd file=orapwlkoracle force=y password=XXXXXXXX
```

4. リスナー経由で、データベースに接続確認を行います。

```
$ sqlplus sys/XXXXXXXX@10.3.1.200:1521/lkoracle as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on 木 2月 18 04:18:46 2021
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.

最終正常ログイン時間: 木 2月  18 2021 04:17:04 +00:00


Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
に接続されました。
SQL> select instance_name from v$instance;

INSTANCE_NAME
----------------
lkoracle
```

# 12.2.6.6.5. Oracle database listener リソース階層の作成

最初に Oracle Database Listener リソースを作成します。

1. **Create Resource Hierarchy** のアイコンをクリックしてリソース作成を開始します。

2. Create Resource ウィザードが表示されます。プルダウンメニューで Recovery Kit に対し **Oracle Database Listener** を選択して、 **Next** をクリックします。

3. Create Resource ウィザードで、以下の値を入力します。

| 番号. | 項目 | 入力もしくは選択する値 |
|---|---|---|
| 1 | Switchback Type | Intelligent |
| 2 | Server | lk4lnode01 |
| 3 | Listener Configuration File Path | /mnt/ORA/app/oracle/product/19.3/dbhome_1/network/admin/listener.ora |
| 4 | Listener Name(s) | LISTENER |
| 5 | Listener Executable(s) | /mnt/ORA/app/oracle/product/19.3/dbhome_1/bin |
| 6 | Listener Protection Level | Full Control(Start,Stop,Monitor,& Recover) |
| 7 | Listener Recovery Level | Standard, (On) |
| 8 | IP Address Name | ip-10.3.1.200 |
| 9 | Listener Tag | LSNR.LISTENER |

4. Oracle Database Listener リソースの作成が開始されます。

5. 「End of successful Create of...」と表示されれば成功です。 **Next** をクリックして Pre-Extend ウィザードへ遷移します。

6. 以下の値を入力します。

| 番号. | 項目 | 入力もしくは選択する値 |
|---|---|---|
| 1 | Target Server | lk4lnode02 |
| 2 | Switchback Type | Intelligent |
| 3 | Template Priority | 1 |
| 4 | Target Priority | 10 |

7. Pre-Extendが開始されます。「Pre Extend checks were successful」と表示されれば成功です。[Next]をクリックして次に進みます。

8. Pre-Extendが開始されます。

9. 「Pre Extend checks were successful」と表示されれば成功です。[Next]をクリックして進めてくだ

さい。

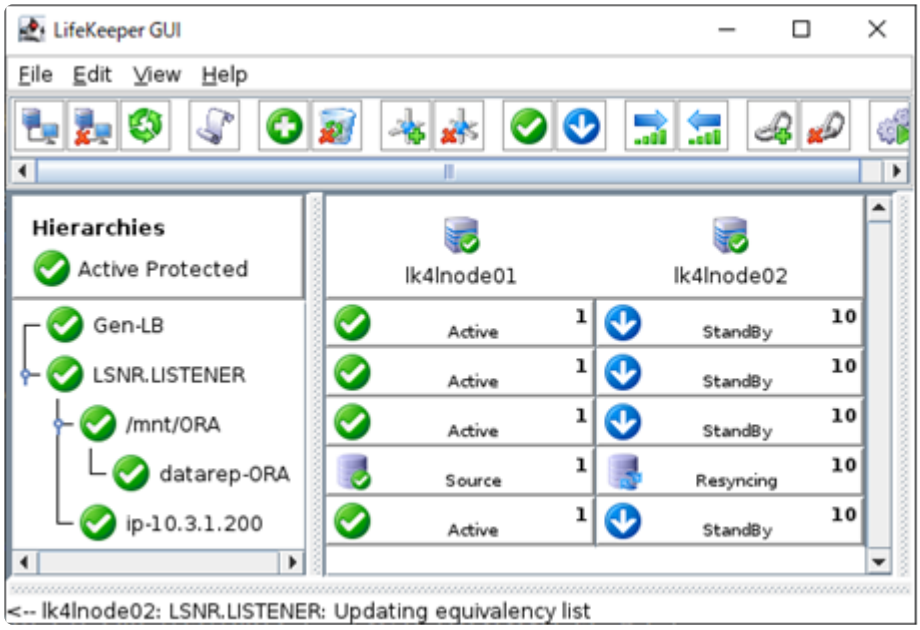
10. Extend Oracle Listener Resource ウィザードで、以下の値を入力します。

| 番号. | 項目 | 入力もしくは選択する値 |
|---|---|---|
| 1 | Enter the Listener Configuration File Path | /mnt/ORA/app/oracle/product/19.3/dbhome_1/network/admin/listener.ora |
| 2 | Enter the Path to the Listener Executable(s) | /mnt/ORA/app/oracle/product/19.3/dbhome_1/bin |
| 3 | Listener Tag | LSNR.LISTENER |

11. Extend が開始されます。「Hierarchy successful extended」と表示されれば成功です。 **Finish** をクリックします。

12. **Done** を押下し、ウィザードを終了します。

13. Oracle Database Listener リソースが作成されました。



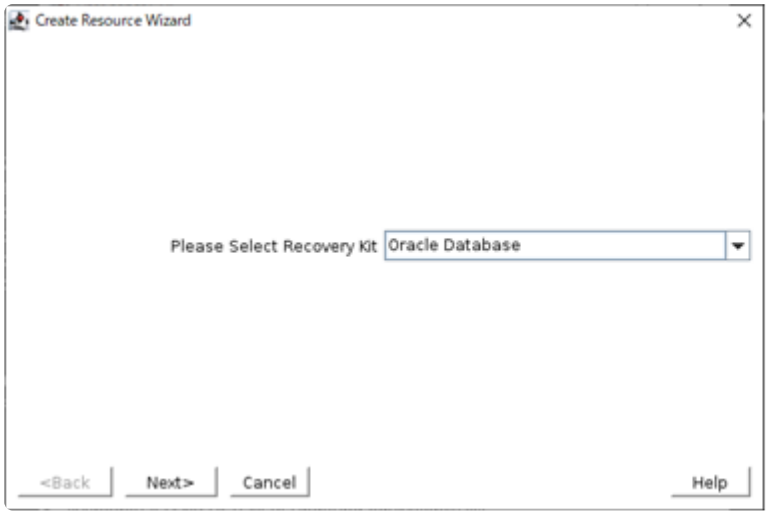

参考： Oracle Recovery Kit 管理ガイド リソース階層の作成

# 12.2.6.6.6. Oracleリソース階層の作成

次に Oracle database のリソースを作成します。

1. **Create Resource Hierarchy** のアイコンをクリックしてリソース作成を開始します。

2. Create Resource ウィザードが表示されます。プルダウンメニューで Recovery Kit に対し **Oracle Database** を選択して、 **Next** をクリックします。

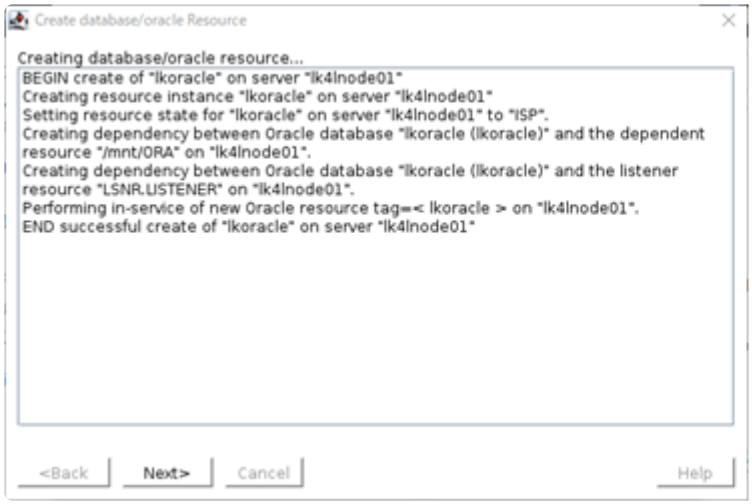



3. Create Resource ウィザードで、以下の値を入力します。

| 番号 | 項目 | 入力もしくは選択する値 |
|---|---|---|
| 1 | Switchback Type | Intelligent |
| 2 | Server | lk4lnode01 |
| 3 | Oracle_SID for Database | lkoracle |
| 4 | User Name | << アクセスに使用するユーザー名 >> |
| 5 | Password | << 上記で指定したユーザーのパスワード >> |
| 6 | Select the Oracle Listener | LSNR.LISTENER |
| 7 | Database Tag | lkoracle |

4. Oracle Database リソースの作成が開始されます。

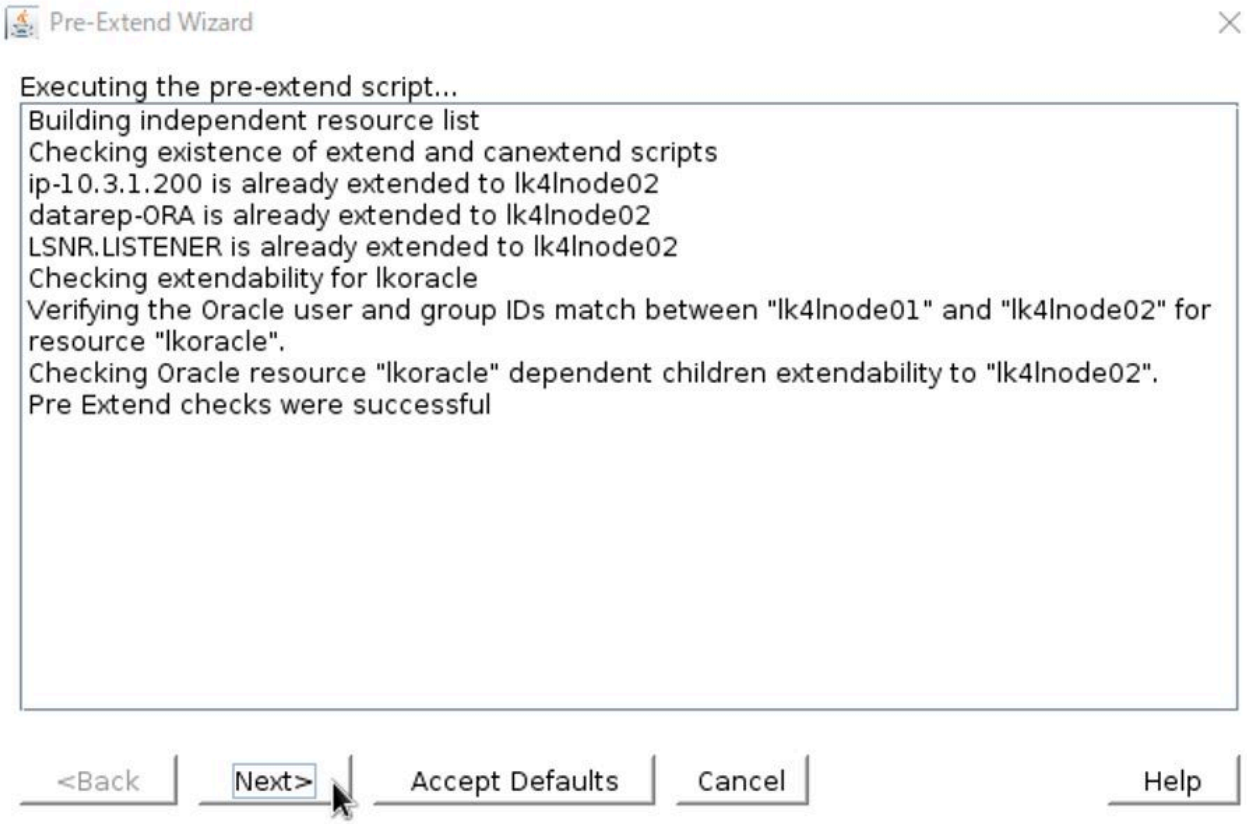
5. 「End of successful Create of...」と表示されれば成功です。 **Next** をクリックして Pre-Extend ウィザードへ遷移します。



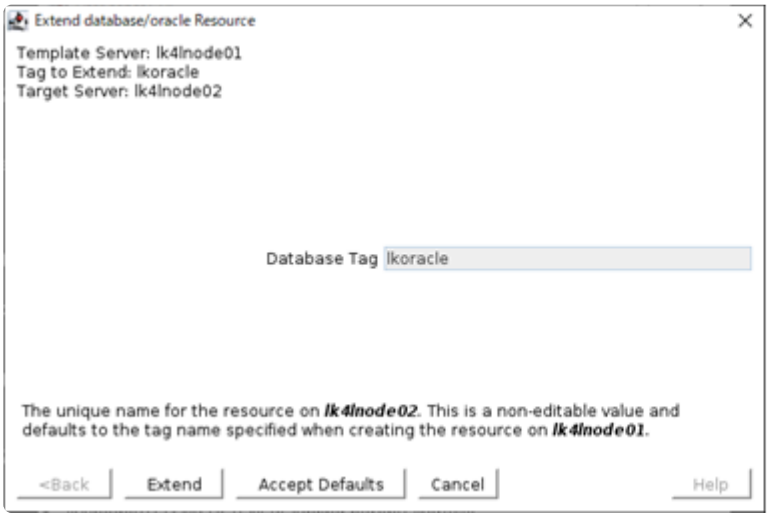

6. 以下の値を入力します。

| 番号 | 項目 | 入力もしくは選択する値 |
|---|---|---|
| 1 | Target Server | lk4lnode02 |
| 2 | Switchback Type | Intelligent |
| 3 | Template Priority | 1 |
| 4 | Target Priority | 10 |

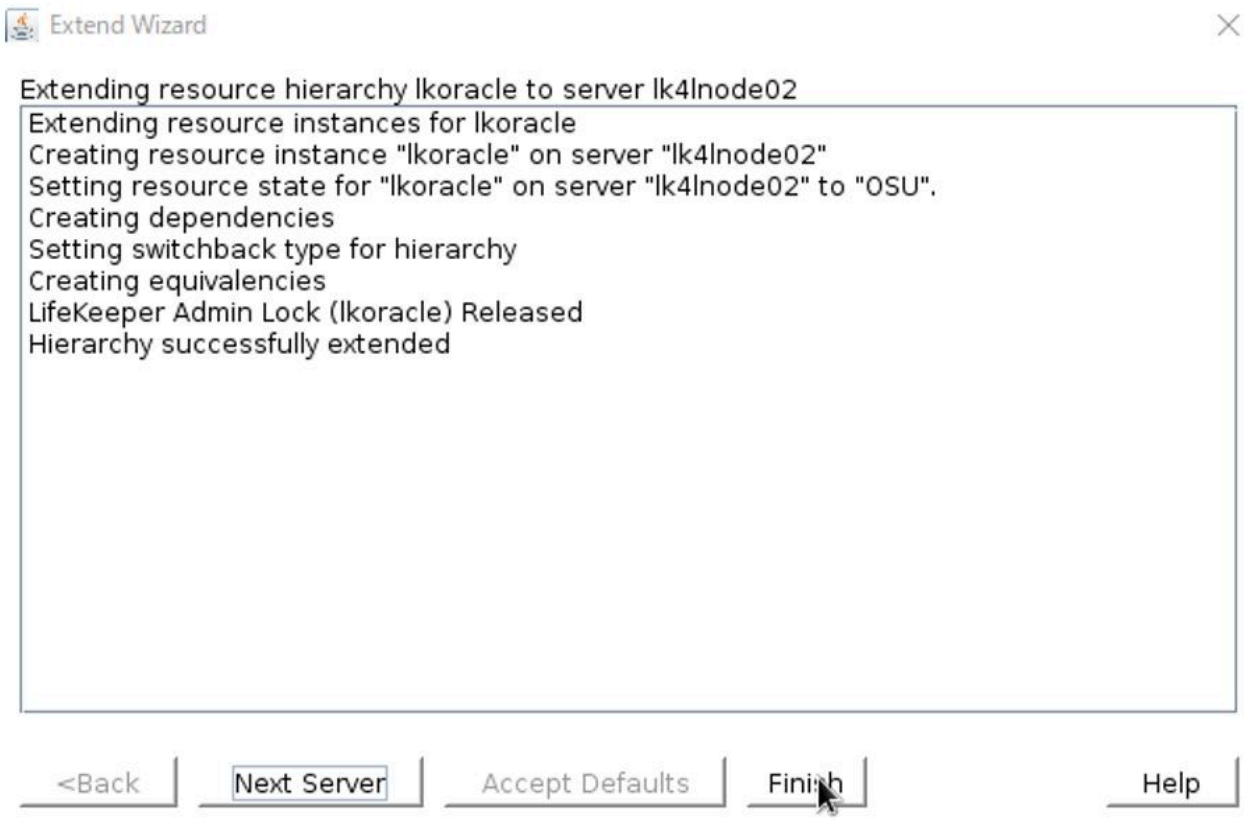
7. Pre-Extend が開始されます。「Pre Extend checks were successful」と表示されれば成功です。**Next** をクリックして次に進みます。



8. Extend database/oracle Resource ウィザードが表示されたら、 **Extend** をクリックします。

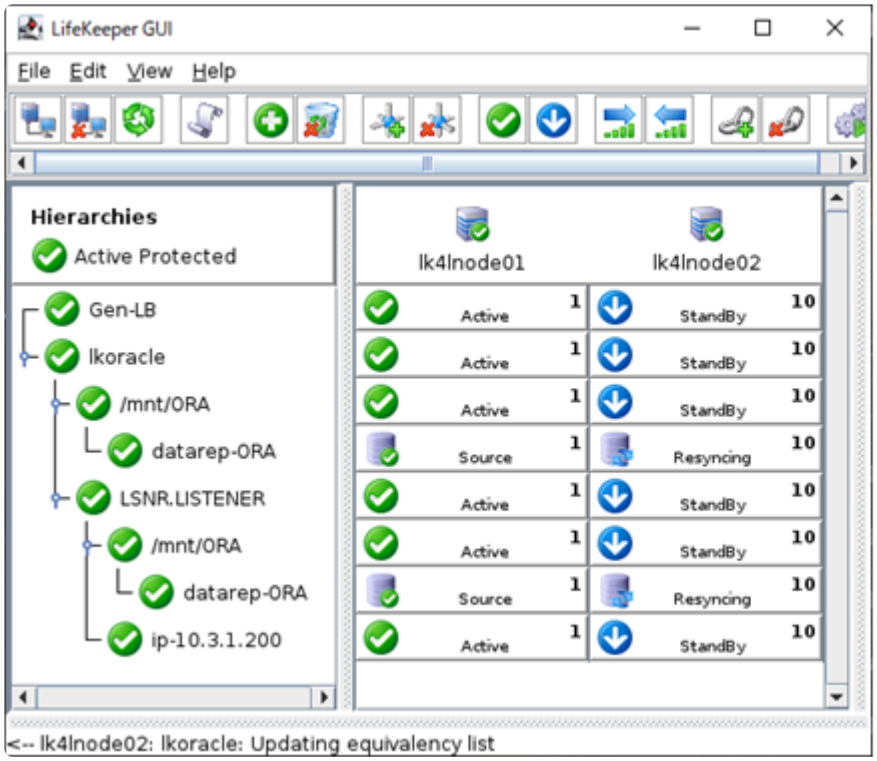



9. Extend が開始されます。「Hierarchy successful extended」と表示されれば成功です。 **Finish** をクリックします。




10. **Done** を押下し、ウィザードを終了します。

11. Oracle Database リソースが作成されました。




12. */etc/oratab* の修正を元に戻します。

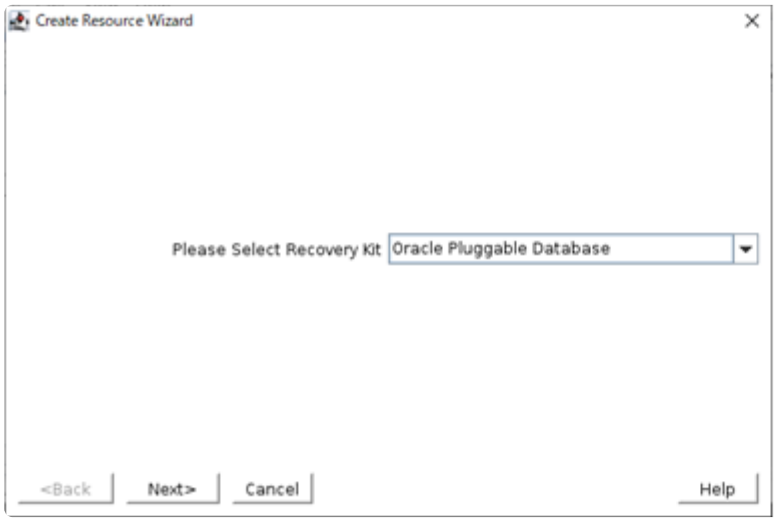
```
# vi /etc/oratab
oracle:/mnt/ORA/app/oracle/product/19.3.0/dbhome_1:N ←Y を N に変更
```

# 12.2.6.6.7. Oracle Pluggable Databaseリソース階層の作成

次にOracle Pluggable databaseのリソースを作成します。

1. [Create Resource Hierarchy]のアイコンをクリックしてリソース作成を開始します。

2. Create Resourceウィザードが表示されます。プルダウンメニューでRecovery Kitに対し[Oracle Pluggable Database]を選択して、[Next]をクリックします。

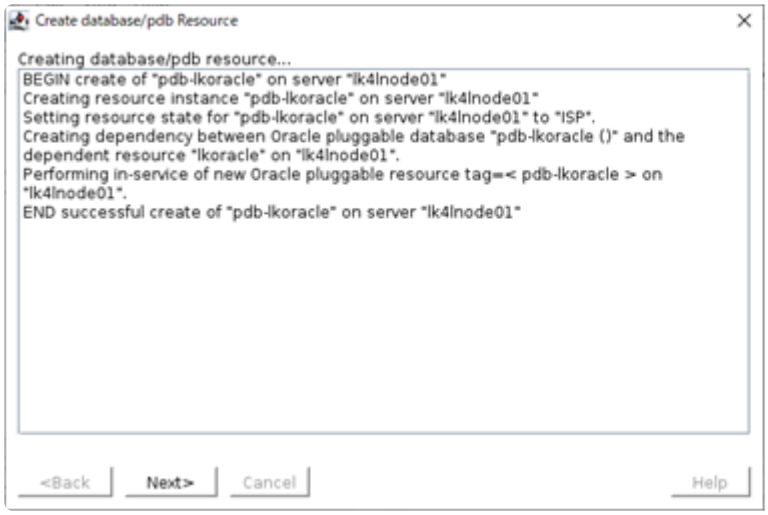



3. Create Resourceウィザードで、以下の値を入力します。

| 番号 | 項目 | 入力もしくは選択する値 |
|---|---|---|
| 1 | Switchback Type | Intelligent |
| 2 | Server | lk4lnode01 |
| 3 | Oracle_SID for Database | lkoracle |
| 4 | Select the Oracle PDBs | LKPDB |
| 5 | Database Tag | pdb-lkoracle |

4. Oracle Pluggable Databaseリソースの作成が開始されます。

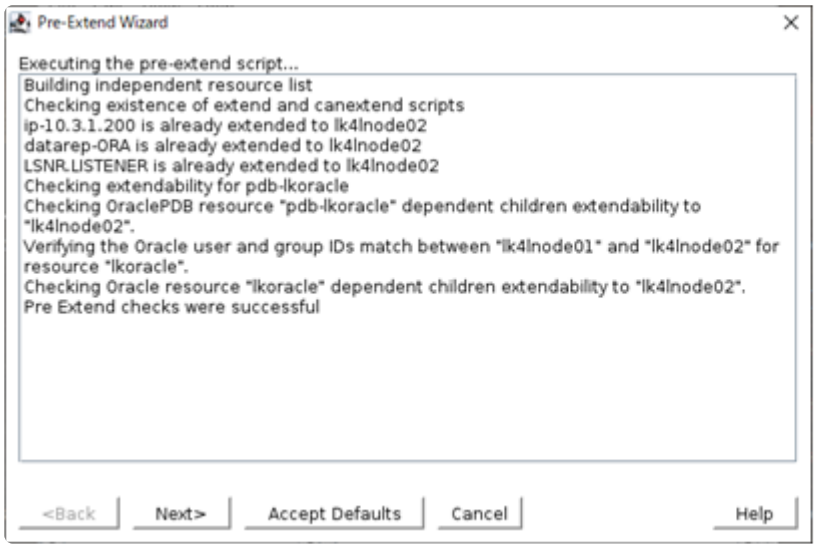
5. 「End of successful Create of...」と表示されれば成功です。[Next]をクリックしてPre-Extend ウィザードへ遷移します。



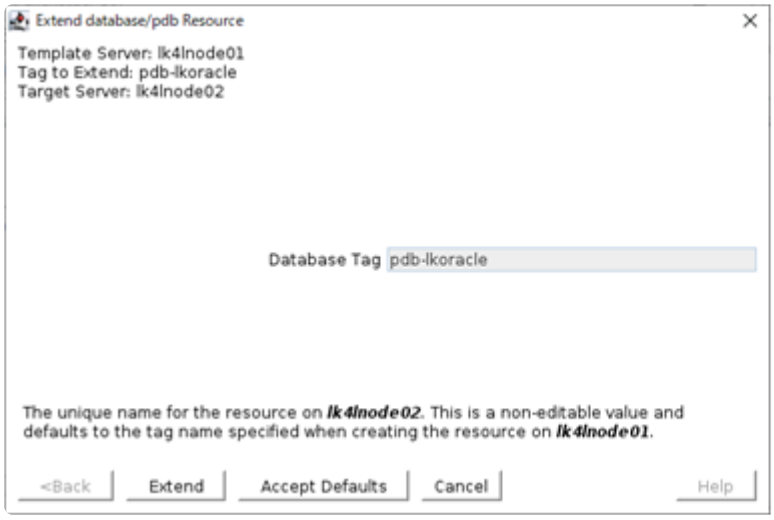

6. 以下の値を入力します。

| 番号 | 項目 | 入力もしくは選択する値 |
|---|---|---|
| 1 | Target Server | lk4lnode02 |
| 2 | Switchback Type | Intelligent |
| 3 | Template Priority | 1 |
| 4 | Target Priority | 10 |

7. Pre-Extendが開始されます。「Pre Extend checks were successful」と表示されれば成功です。[Next]をクリックして次に進みます。



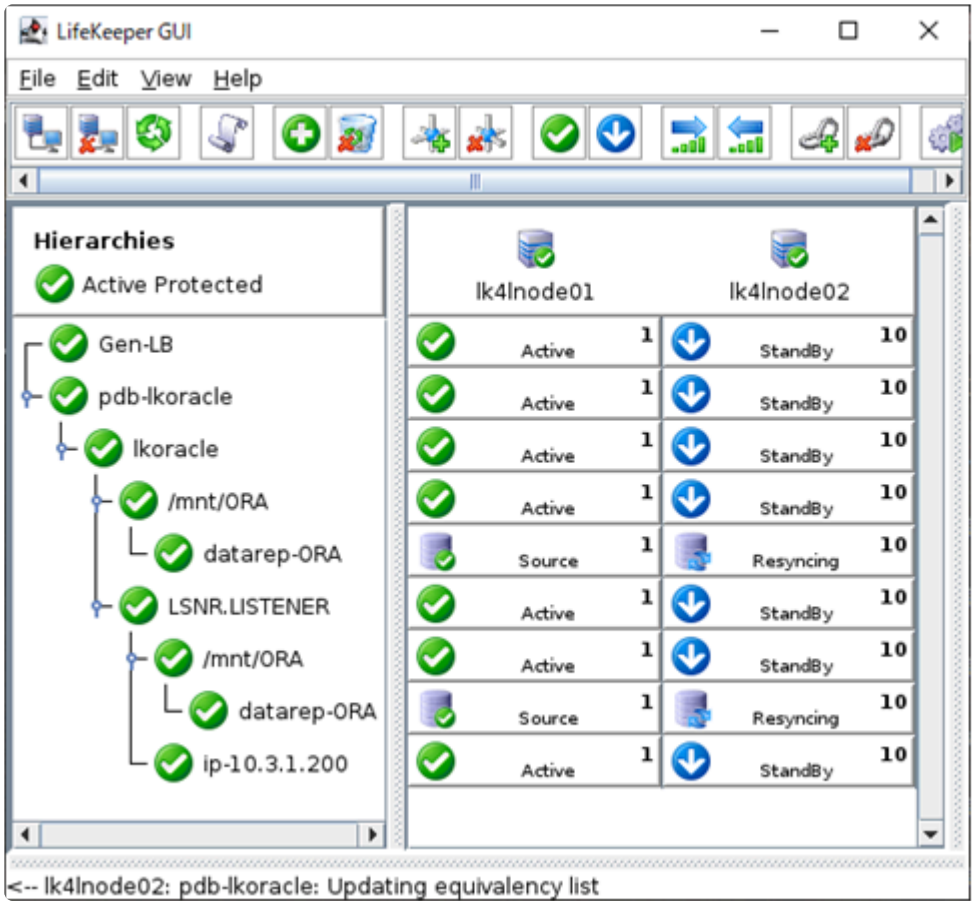

8. Extend database/pdb Resourceウィザードが表示されたら、[Extend]をクリックします。




9. Extendが開始されます。「Hierarchy successful extended」と表示されれば成功です。[Finish]をクリックします。

10. [Done]を押下し、ウィザードを終了します。

11. Oracle Databaseリソースが作成されました。
    完成したリソースツリーです。




参考：[Oracle Multitenant 構成による Pluggable Database の設定]

# 12.2.6.6.8. 依存関係の作成

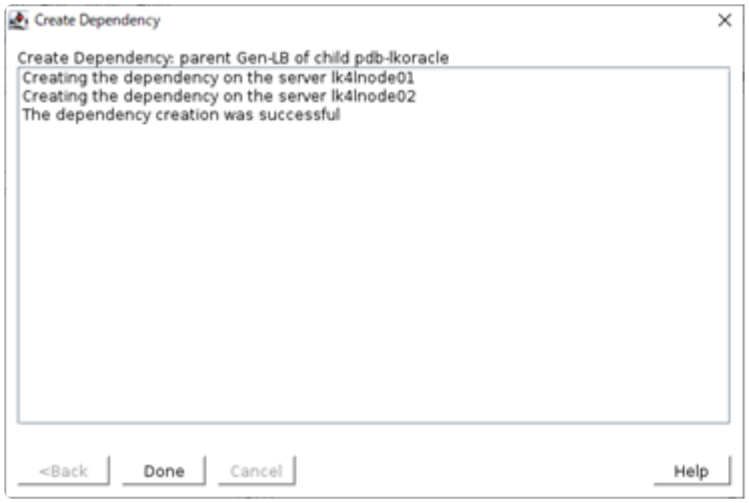
作成済みの LB Health Check リソースとOracle Pluggable Database リソースの依存関係を作成します。

1. . LifeKeeper GUIで[Create Dependency]のアイコンをクリックして依存関係の作成を開始します。

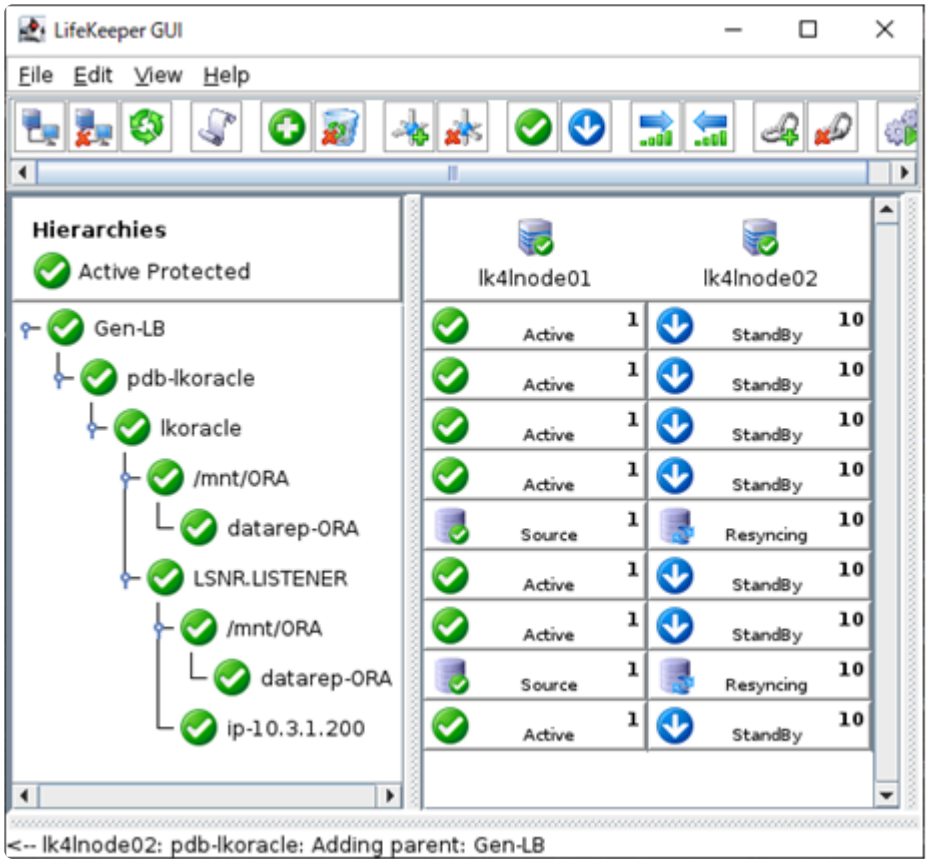
2. Create Dependencyウィザードで、以下の値を入力します。

| 項目 | 値 |
|---|---|
| Server | lk4lnode01 |
| Parent Resource Tag | Gen-LB |
| Child Resource Tag | pdb-lkoracle |

3. 依存関係の作成が開始され、「The dependency creation was successful」と表示されれば成功です。[Done]をクリックします




4. 完成したリソースツリーです。

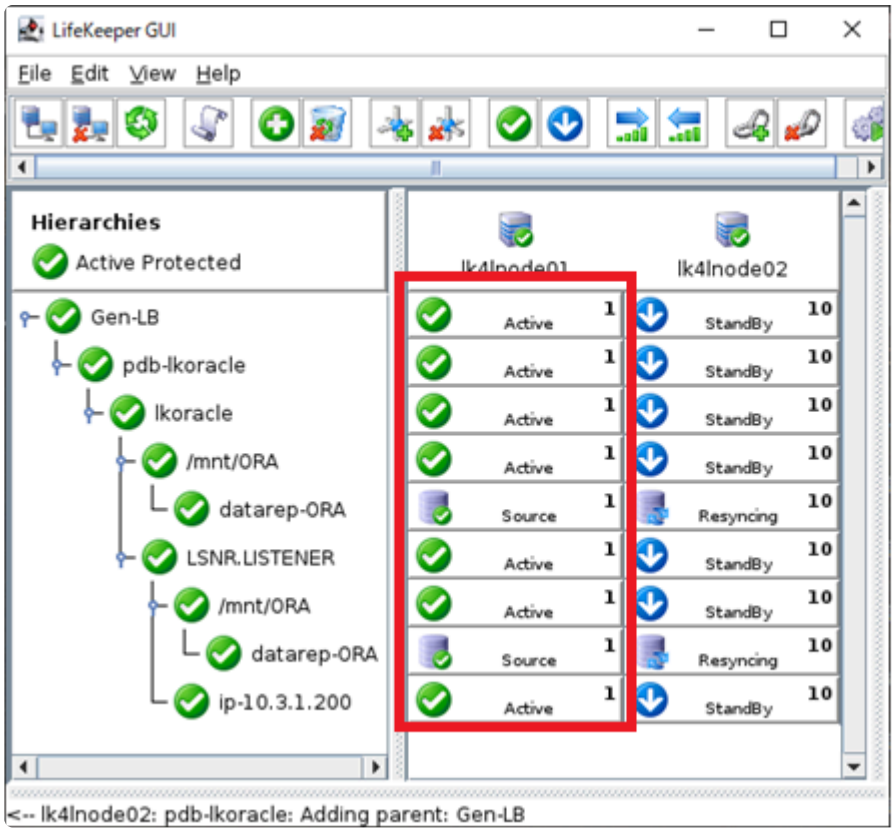

LifeKeeper GUI
File Edit View Help
Hierarchies
Active Protected
Gen-LB
pdb-lkoracle
lkoracle
/mnt/ORA
datarep-ORA
LSNR.LISTENER
/mnt/ORA
datarep-ORA
ip-10.3.1.200
lk4lnode01
lk4lnode02
Active 1 StandBy 10
Active 1 StandBy 10
Active 1 StandBy 10
Active 1 StandBy 10
Source 1 Resyncing 10
Active 1 StandBy 10
Active 1 StandBy 10
Source 1 Resyncing 10
Active 1 StandBy 10
<-- lk4lnode02: pdb-lkoracle: Adding parent: Gen-LB

# 12.2.6.7. 接続確認

最後にリモートのクライアントより、データベースへの接続確認を行います。

1. クライアント上に Oracle クライアントソフトウェアを導入します。

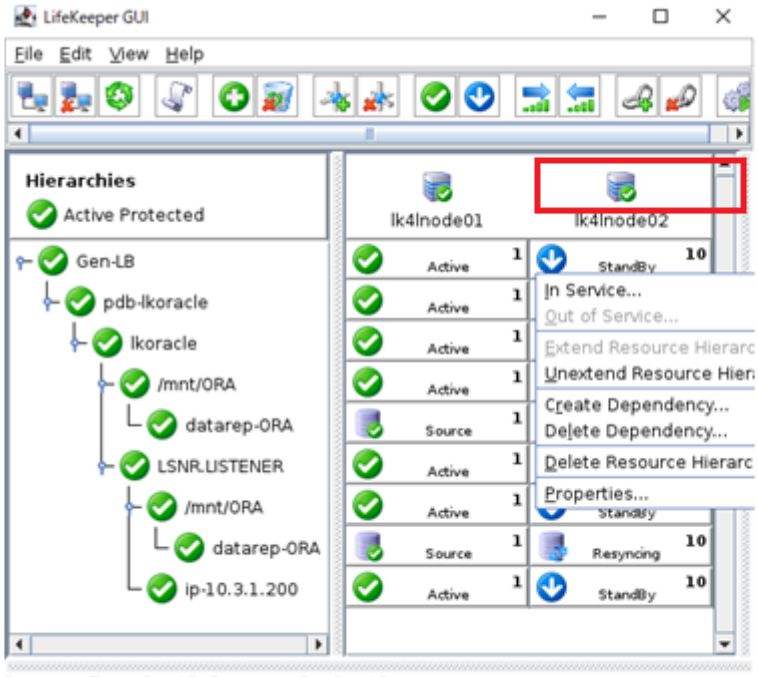
2. まず、プライマリーノード上でリソースがアクティブであることを確認します。




3. リモートのクライアントから、リスナー経由でデータベースに接続します。

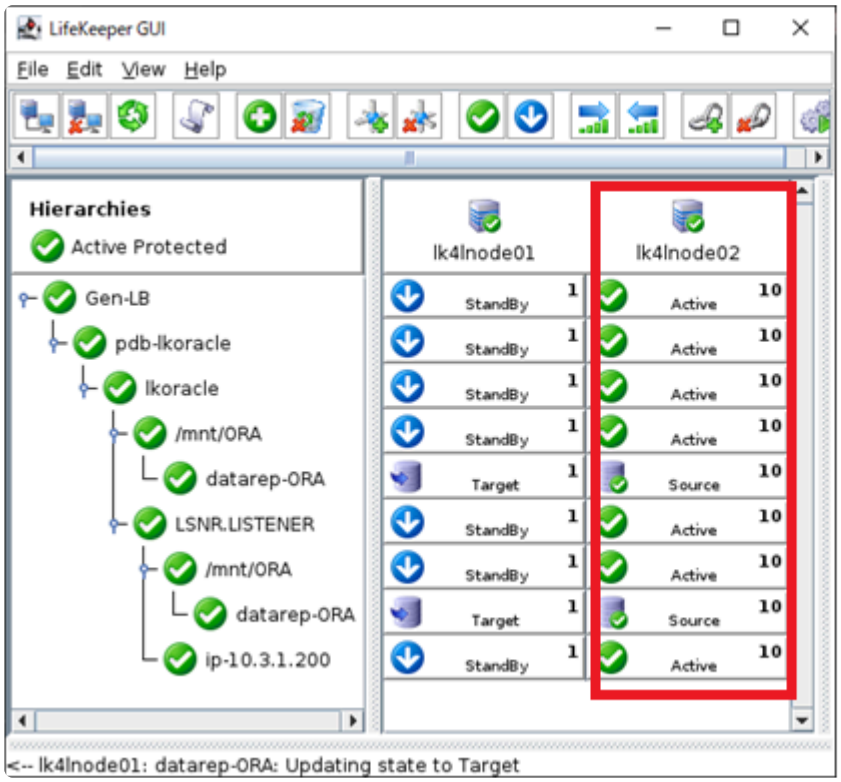
```
# hostname
lk4lclient
# sqlplus sys/XXXXXXXX@10.3.1.200:1521/lkoracle as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Feb 19 00:04:00 2021
Version 19.8.0.0.0

Copyright (c) 1982, 2020, Oracle.  All rights reserved.

Last Successful login time: Thu Feb 18 2021 07:50:03 +00:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> select instance_name from v$instance;

INSTANCE_NAME
----------------
lkoracle
```

4. LifeKeeper GUIで LB Health Check リソースのスイッチオーバを実施します。lk4lnode02（スタンバイノード）側の LB Health Check リソースを右クリック > [In Service]を選択します。



Resource Tag= Gen-LB, Resource ID= Gen-LB

5. スイッチオーバーが完了しました。



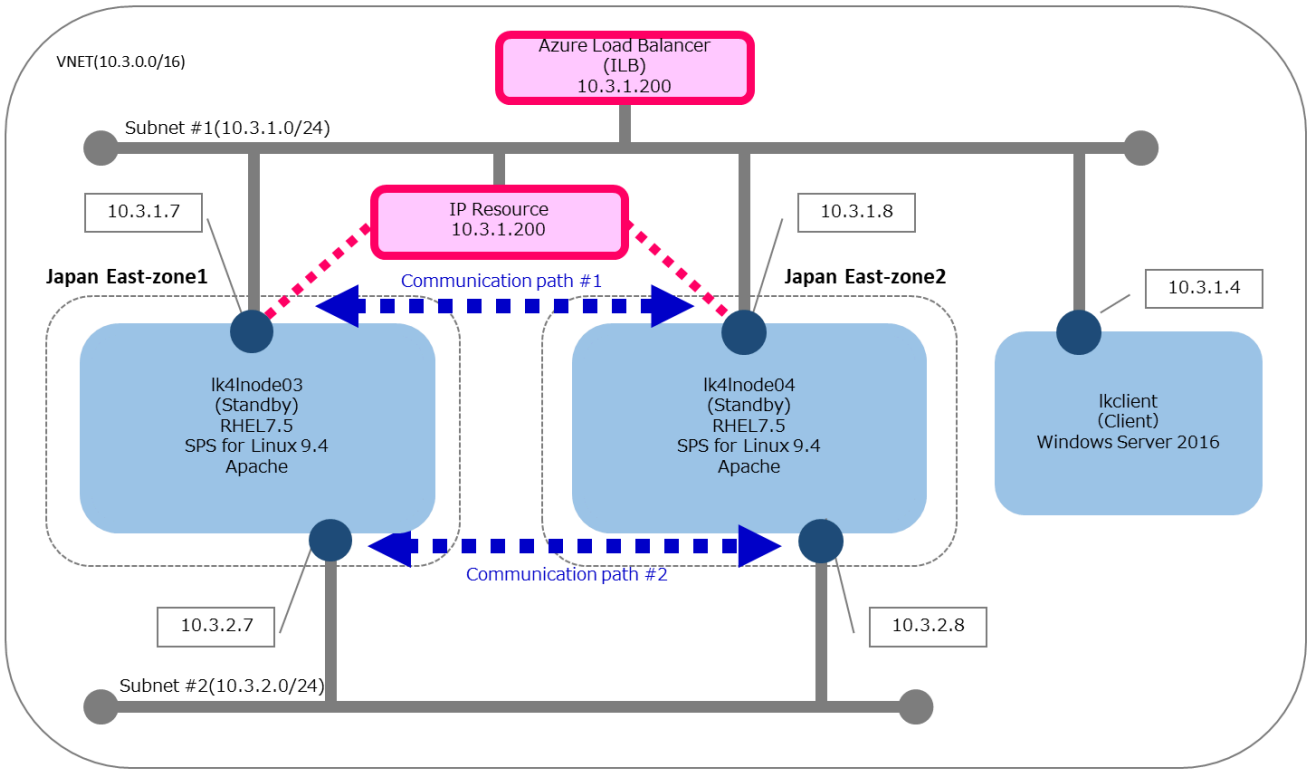

6. リモートのクライアントから以下の通りリスナー経由で接続します。(クラスターホスト名でアクセスします)

```
# hostname
lk4lclient
# sqlplus sys/XXXXXXXX@10.3.1.200:1521/lkoracle as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Feb 19 00:16:34 2021
Version 19.8.0.0.0

Copyright (c) 1982, 2020, Oracle.  All rights reserved.

Last Successful login time: Fri Feb 19 2021 00:04:01 +00:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> select instance_name from v$instance;

INSTANCE_NAME
----------------
lkoracle
```

7. 切り替え前後において、同じIPアドレスでOracleに接続できることが確認できました。

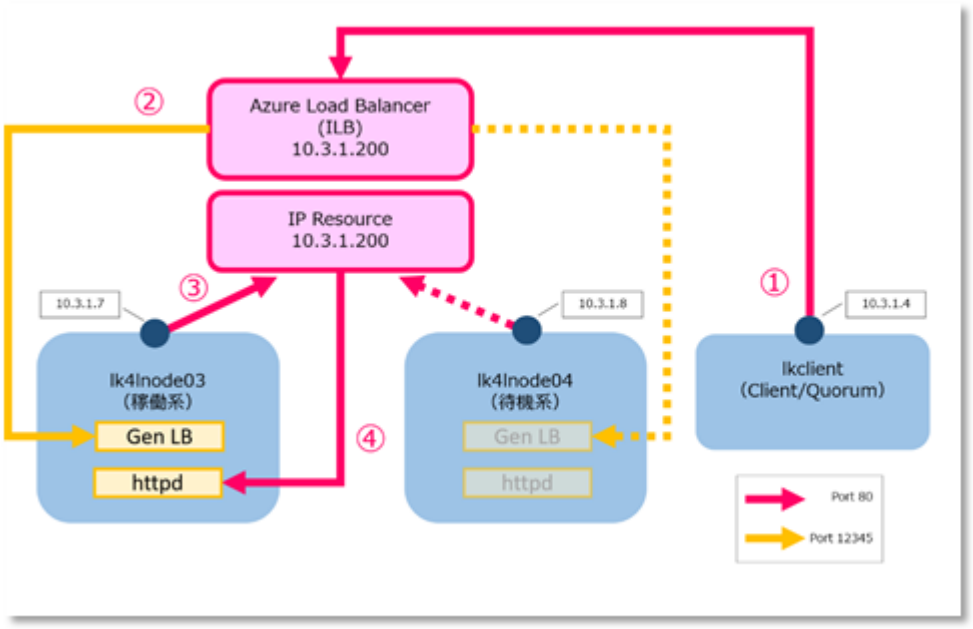
# 12.2.7. Availability Zone (高可用性ゾーン)

## 動作確認内容




保護対象アプリケーションは Apache で、クライアント PC から Web ブラウザ等による接続確認を行いました。稼働系サーバー（lk4lnode03）と待機系サーバー（lk4lnode04）はそれぞれ異なるzone に属しています。クライアント（lkclient）からの http リクエストは 12345番でILB経由でアクティブなサーバーにリダイレクトされます。

# ILBによるクライアントからの接続の動作概要




ILBによるクライアントからの接続の動作概要

クライアント PC からの接続動作の概要は以下の通りです。

① Clientがhttpdへ接続する為、10.3.1.200(port80)への接続をILB経由で開始します。

② ILB に負荷分散先（backend pool) に 10.3.1.7 (稼働系：lk4lnode03)と 10.3.1.8 (待機系：lk4lnode04) が 登録されています。ILBは各ノードに対して正常性プローブによる監視を行い、ポートがオープンになっているノードを確認して、受け取ったhttpd宛のパケットを転送します。正常性プローブへの応答には LB Health Check リソースを使用します。

③ LifeKeeperの仕様上Apacheリソースは[IPリソースが保護するVIP]を指定する必要があります。その為、受信用にIPリソース（10.3.1.200）を作成します。LifeKeeper のIPリソース（10.3.1.200）がIn-Serviceになっているのは稼働系のみとなる為、実際には 稼働系（10.3.1.7）のみリクエストを受け取ります。

④ 結果としてClientからの接続要求は稼働系のhttpd（10.3.1.200：port80）が受け取ります。

# 12.2.7.1. Azure の構成

Availability Zone を使用するには VM の作成時に以下のように設定します。

1. [Region] で「Availability zone」をサポートするリージョンを選択します。※１
2. [Availability options] 項目で「Availability zone」を選択します。
3. [Availability zone] 項目で、zone を [１ ，２ ，３] のいずれかを選択します。稼働系と待機系はそれぞれ異なるゾーン（稼働系(Zone 1)、待機系 (Zone 2)となるようにします。

※１例えば [(Asia Pacific) Japan East]は「Availability zone」をサポートするリージョンです。 [Japan West] のような Availability Set しか利用できないリージョンを選択すると [Availability options] 項目で「Availability zone」が選択できません。

基本設定はAvailability Setを使用した構成と同じです。

Availability zone を使用した構成の環境は以下の通りです。

1. ネットワーク、ロードバランサー

**[Virtual networks]**
Address space: 10.3.0.0/16
Subnets:

Subnet1 : 10.3.1.0/24

Subnet2 : 10.3.2.0/24

**[Load balancers]**
Location : Japan East
SKU : Standard
Frontend IP configuration:

Type : Private

Subnet : 10.3.1.0/24

IP Address : 10.3.1.200(Static)

Availability Zone : Zone-redundant

Backend pools:

lk4lnode03

lk4lnode04

Health probes:

Protocol : TCP

Port : 12345

Load balancing rules:

Port : 80

Backend Port : 80

Session persistence : None

Floating IP(Direct Server Return) : Enabled

Inbound NAT rules :

0 rules

２．仮想サーバー

**[Active Node]**
Name : lk4lnode03
Operating system : Linux(RHEL 7.8)
Size : Standard A1 v2 (1 vcpus, 2 GiB memory)
Networking:

Private IP : 10.3.1.7

Installed Software :

httpd : httpd-2.4.6-93.el7.x86_64

LifeKeeeper : SPS for Linux v9.5.2

IP tables :

```
...
-A POSTROUTING -s 10.3.1.200/32 -p icmp -j SNAT --to-source 10.3.1.8
...
:OUTPUT ACCEPT [4188:806503]
-A INPUT -s 10.3.0.0/16 -p tcp -j ACCEPT
-A INPUT -s 168.63.129.16/32 -p tcp -j ACCEPT
...
```

httpd.conf :

```
...
Listen 10.3.1.200:80···
...
ServerName 10.3.1.200:80
...
```

**[Standby Node]**
Name : lk4lnode04
Networking: Private IP : 10.3.1.8
IP tables :

```
...
-A POSTROUTING -s 10.3.1.200/32 -p icmp -j SNAT --to-source 10.3.1.7
...
:OUTPUT ACCEPT [4188:806503]
-A INPUT -s 10.3.0.0/16 -p tcp -j ACCEPT
-A INPUT -s 168.63.129.16/32 -p tcp -j ACCEPT
...
```

そのほか、Active Nodeと同じ。

**[Client Node]**
Name : lkclient
Operating System : Windows (Windows Server 2019 Datacenter)
Size : Standard B1s (1 vcpus, 1 GiB memory)

Networking:

Private IP : 10.3.1.4

Installed Software :

Google Chrome： Version 80.0.3987.149 (Official Build) (64-bit)

Xming ：6.9.0.31

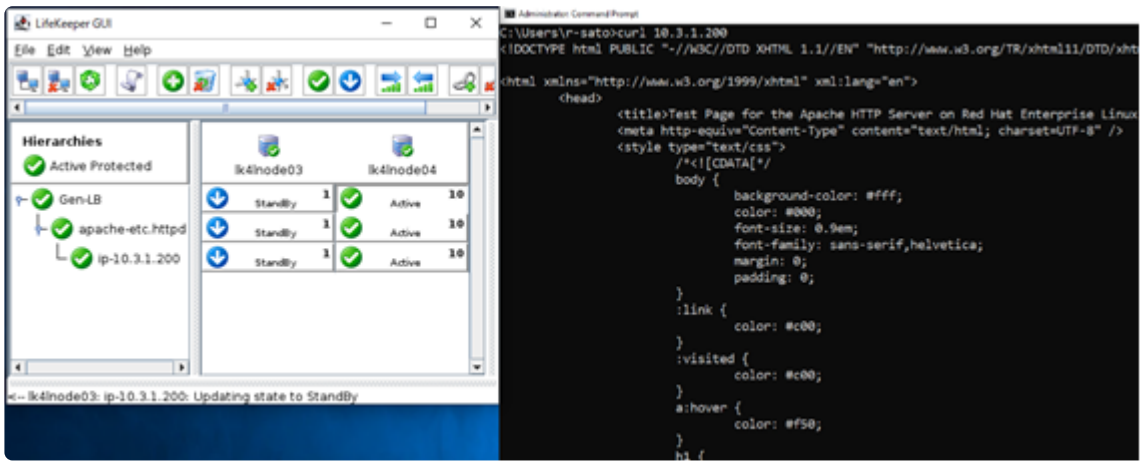
Tera Team： 4.102

WinSCP： 5.17.2

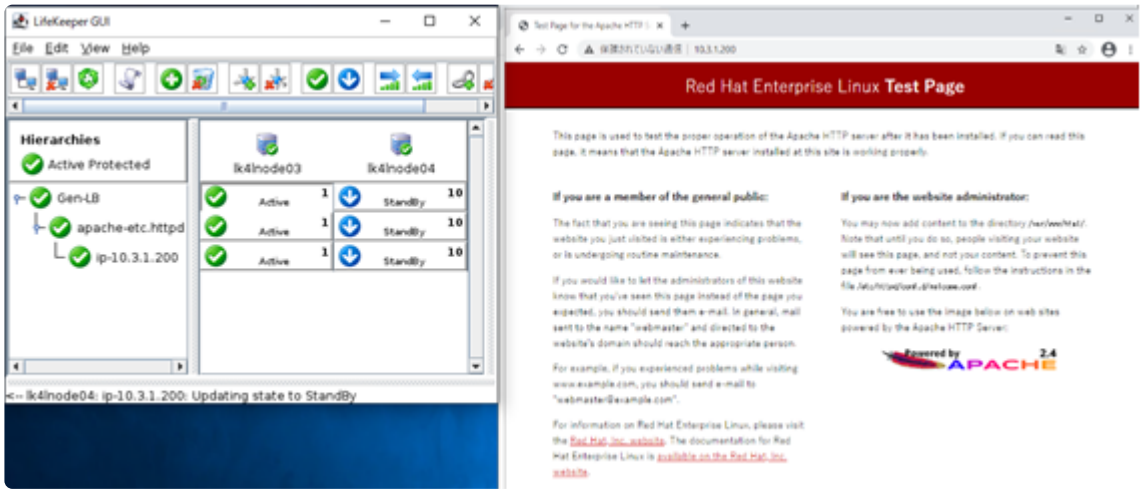
# 12.2.7.2. クライアントリダイレクトの動作確認方法

lkclient から、以下２つの方法でアクセスを確認し、いずれの方法でも仮想 IP への接続が行えることを確認しました。

Windows Server 2019（クライアント）のターミナルからcurl（コマンドラインベースのhttpアクセスツール）で http://10.3.1.200/ にアクセスし稼働系のindex.htmlが表示されることを確認しました。
スイッチオーバーを行いに伴い待機系のindex.htmlが表示されることを確認しました。




2. クライアントの Web ブラウザ上で http://10.3.1.200/ を入力し Web ページにアクセスし index.html が表示されることを確認しました。（*確認の都度キャッシュをクリアする）スイッチオーバーに伴いスイッチオーバー先の index.html が表示されることを確認しました。



# 12.2.8. 参考資料

サイオステクノロジー：

SIOS テクニカルドキュメンテーション

SIOS Protection Suite for Linux テクニカルドキュメンテーション

Oracle Recovery Kit 管理ガイド

LifeKeeper / DataKeeper ユーザーポータル

社外リンク：

Microsoft Azure ホームページ
http://azure.microsoft.com/ja-jp/

Microsoft Azure Linux 仮想マシンのドキュメント
https://docs.microsoft.com/ja-jp/azure/virtual-machines/linux/

Microsoft Azure Japan Team Blog (ブログ)
http://blogs.msdn.com/b/windowsazurej/

High Availability for a file share using WSFC, ILB and 3rd-party Software SIOS Datakeeper（ILB の設定で参考資料として利用）
https://azure.microsoft.com/ja-jp/blog/high-availability-for-a-file-share-using-wsfc-ilb-and-3rd-party-software-sios-datakeeper/

謝辞

本ドキュメント作成に際して、日本マイクロソフト様より Microsoft Azure の利用をご承認いただき、また環境構築や設定、テストケースの作成に際して多大なるご支援をいただきました。御礼を申し上げます。

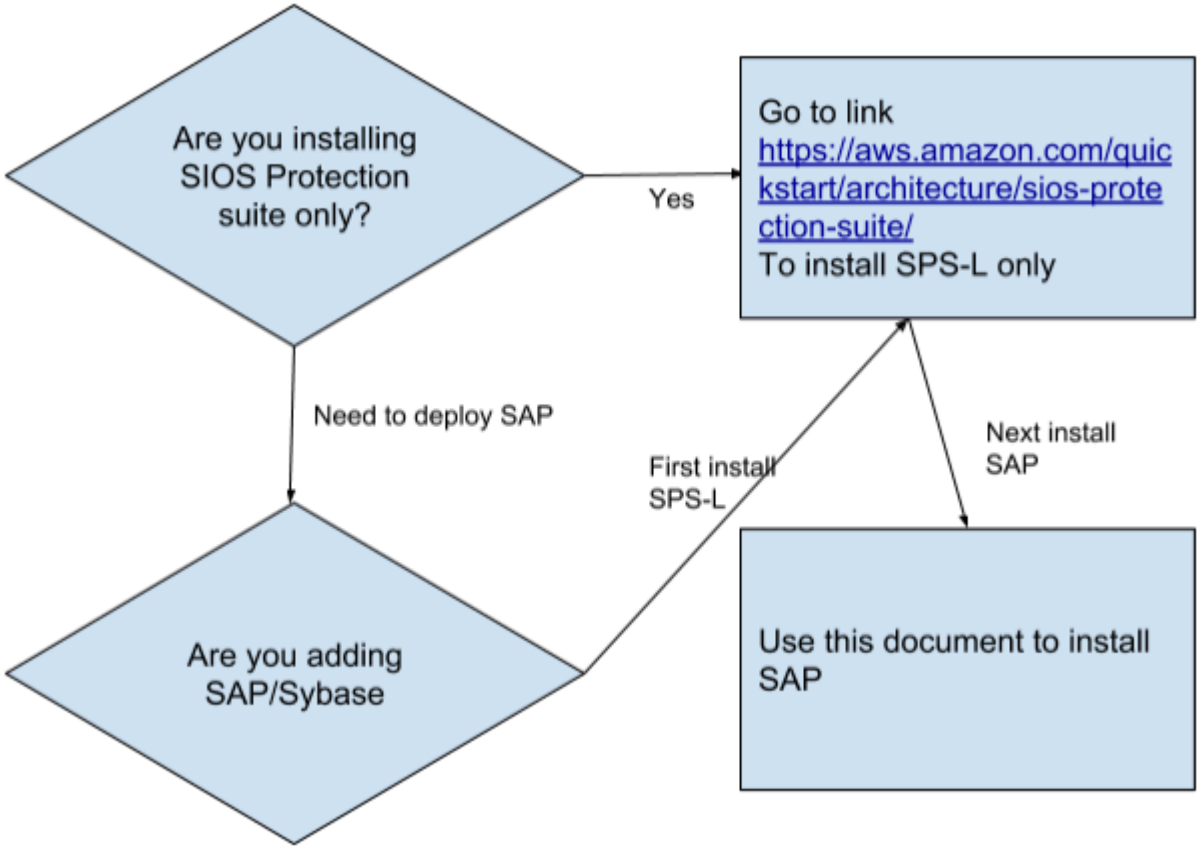
Microsoft Azure ホームページ
https://azure.microsoft.com/ja-jp/

# 12.3. LifeKeeper for Linux in the AWS Cloud (SAP)

## 概要

本書は、LifeKeeper for Linux のインストールのユーザーガイドです。

LifeKeeper for Linux for SAP 環境のインストール方法を理解するには、以下のマトリックスに従ってください。




注記：図中のリンクを こちら にも記載します。

# 12.3.1. LifeKeeper 上で SAP を設定する際の追加手順

LifeKeeper上で SAP を設定するには、以下の手順に従ってください。

## ステップ 1. デプロイメントのテスト

LifeKeeper for Linux ノードに接続するには、Windows JumpBox に接続する必要があります。Windows マシンに接続するには、リモートデスクトップ端末に接続してください。

AWS コンソールで、作成された Windows JumpBox ノードを選択し、 **[Action]** の次に **[Connect]** をクリックします。これでリモートデスクトッププログラムをダウンロードして接続できます。また、マシンにログインするために必要なパスワードを複合化する必要があります。

Windows マシンに接続したら、PuTTY と VNC Viewer をダウンロードすることを推奨します。以下のサイトからダウンロードしてください。

- **Putty** – www.putty.org
- **VNC Viewer** – https://www.realvnc.com/en/connect/download/viewer/

これで、PuTTY を使用して各ノードのプライベート IP アドレスに接続することができます。また、同じプライベート IP アドレスを使用しているノードに接続するために VNC Viewer を使用することもできます。ノードへは Windows JumpBox の外ではアクセスできませんが、ノードは NAT ゲートウェイを使用してインターネットにアクセスできます。（ **注記：** NAT ゲートウェイに問題がある場合は、セキュリティグループのルール/メインルートを確認してください。）

いずれかのノードに接続したら、先ほどのテンプレートで作成したパスワードを使用して root で su を実行し、プログラム vncserver を実行します。これにより、VNC Viewer を使用してグラフィカルインターフェース内のそのノードに接続できます。

**VNC サーバーをインストールするためのコードスニペット**

次のオプションを指定して、vncserver コマンドを root として実行します。

パスワードを入力し、確認のために繰り返します
読み取り専用パスワードを **No** に設定します

オプションで /root/.vnc/config を編集し、以下の記述を追加します。
securitytypes=none
vncserver -kill:1

VNC へのアクセスは ipv4:5901 です。5901 は指定されたポート番号です。

デスクトップを右クリックして **Open Terminal** をクリックし、LifeKeeper GUI に接続するコマンド **/opt/LifeKeeper/bin/lkGUIapp** を入力します。事前に設定したパスワードで root としてログインしてくださ

い。2 つのノードが接続されていることがわかります。

この時点で、基本的な LifeKeeper 2 ノードがセットアップされました。LifeKeeper を使用して、SAP のインストールと SAP サービスの保護を続行します。

# ステップ 2. 仮想 IP の設定

ノード上で SAP が設定されたので、引き続き LifeKeeper を設定して SAP サービスとファイルシステムを保護します。

## Amazon AWS Elastic Compute Cloud (EC2) のセットアップ

AWS Command Line Interface（CLI）を各ノードにインストールする必要があります。詳細は、 AWS Command Line Interface のインストール を参照してください。すべての EC2 インスタンスは、HTTP および HTTPS を使用して Amazon EC2 サービスエンドポイントにアクセスできる必要があります。Amazon EC2 インスタンスのメタデータを取得するには、HTTP プロトコルを使用して IP アドレス 169.254.169.254 にアクセスしてください。

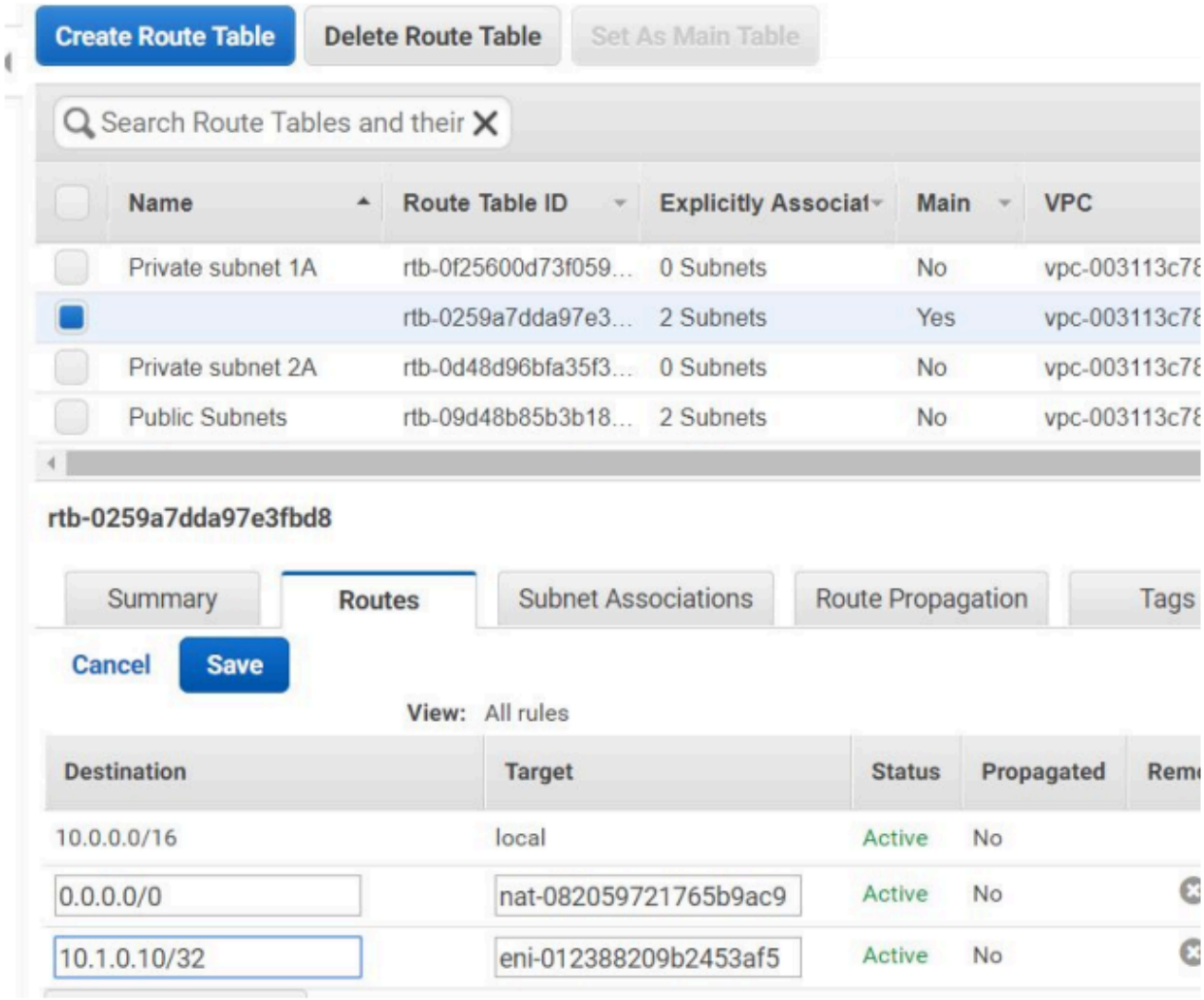
LifeKeeper が AWS を操作するためには、以下のアクセス権を持つ IAM ユーザーまたは IAM ロールが必要です。EC2 インスタンスの root ユーザーからアクセスできるように、 EC2 IAM ロール を設定するか、AWS CLIを適切に設定してください 。

- ec2:DisassociateAddress
- ec2:DescribeAddresses
- ec2:AssociateAddress
- ec2:DescribeRouteTables
- ec2:ReplaceRoute

## 仮想 IP リソースの作成

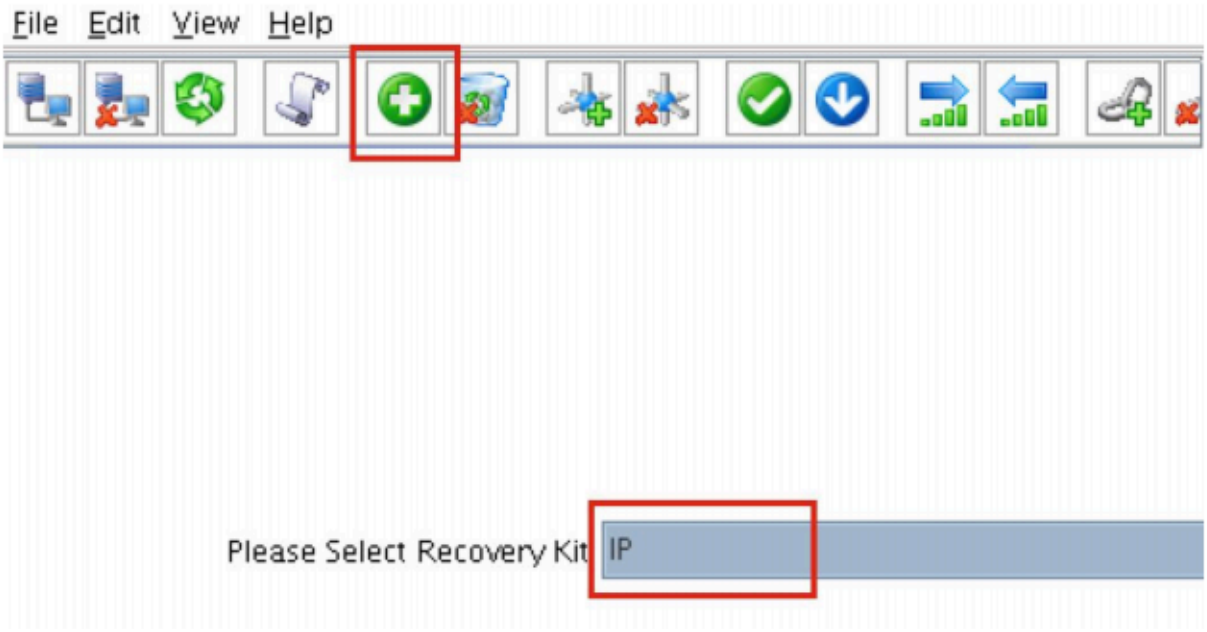
IP アドレスを決定します。IP アドレスは、ノードの現在の IP の CIDR ブロック範囲外の IP アドレスである必要があります。IP アドレスは、ノードの VPC ルートテーブルに配置してください。

以下の図では eni- ネットワークアダプタを使用して IP アドレスを 10.1.0.10/32 に設定し、ノードの 1 つに関連付けました。




/etc/default/LifeKeeper を編集して NOBCASTPING=1 を設定し、続行する前にブロードキャスト ping を無効にします。

**緑色のプラスアイコン** をクリックして、新規リソースを作成します。




ウィザードに従い、以下のように選択して IP リソースを作成します。

Select Recovery Kit: IP

Switchback Type: Intelligent

IP Resource: 10.1.0.10

Netmask: 255.255.255.0

Network Interface: eth0

IP Resource Tag: ip-10.1.0.10

以下のように選択して IP リソースを拡張します。

Switchback Type: Intelligent

Template Priority: 1

Target Priority: 10

IP Resource: 10.1.0.10

Netmask: 255.255.255.0

Network Interface: eth0

IP Resource Tag: ip-10.1.0.10

クラスターは次のようになり、ミラーリソースと IP リソースの両方が作成されます。

## ステップ 3. SAP のセットアップ

SAP ソフトウェアをダウンロードし、ノード上でセットアップします。SAP マーケットプレイスにアクセスして、各ノードに SAP ソフトウェアをダウンロードできます。

SAP のセットアップにはいくつかの選択肢があります。どれを実装するかは、コスト、経験、RAS（信頼性、可用性、サービサビリティ）などのさまざまな要因に依存します。

- NFS なしの ASCS の構成
- NFS を使用している ASCS + ERS 構成

各構成にはメリットとデメリットがあります。現場で SAP エキスパートと作業するか、SIOS プロフェッ

ショナルサービスに加入して、お使いの環境に最適なものを決定することをお勧めします。

**注記：** 今後作成するドキュメントでは、EFS（AWS Elastic File System）と Cloudwatch を使用したオールインワンの HANA のインストールについて詳しく説明します。自動インストールのクイックスタートスクリプトやインストールを管理するための SAP Landscape Manager（LaMa）の使用も計画しています。

# 12.3.2. NFS なしの ASCS 構成

NFS なしで ASCS をセットアップする手順は次の通りです。

- 一般的なセットアップのステップ概要
- SAP のインストール
- SAP リソース階層の作成

# 12.3.2.1. NFS を使用しない ASCS の一般的なセットアップ手順

1. 仮想 IP を作成します。（先のステップで実施済み）
2. EC2 リソースを作成し、仮想 IP の依存関係として作成します。（先のステップで実施済み）
3. 「仮想 IP」に基づいた「仮想ホスト名」のノード 1 に SAP をインストールします。
4. ノード 1 で stopsap を実行します。
5. LifeKeeper GUI を使用して仮想 IP をノード 2 に対して「In Service」にし、「仮想 IP」に基づいてノード 2 の「仮想ホスト名」に SAP をインストールします。
6. ノード 2 で stopsap を実行し、両方のノード上でプロファイルファイルを変更します。（以下参照）
7. LifeKeeper GUI を使用して、仮想 IP をノード 1 で「In Service」にします。
8. SAP コンサルタントのアドバイスに従って、SAP に必要なマウントポイントの複製リソースを作成します。（先のステップで実施済み）
9. ノード 1 で Startsap を実行し、SAP が正常に動作していることを確認します。
10. /etc/default/LifeKeeper で、両方のノードの最後に次のエントリを追加します。

    SAP_EXPERTMODE=1

    SAP_DB_CHECK_IGNORE=1

11. SPS セットアッププログラムを再実行して、SAP Recovery Kit を追加します 。

    ./setup

    Recovery Kit Selection Menu から SAP Recovery Kit をインストールしてください。矢印キーを使用して使用可能な Recovery Kit のメニューから SAP Recovery Kit を選択し、<スペース>で決定し、<Enter> を押してインストールを続行します。

12. SAP Recovery Kit ガイド に従って SAP リソースを作成します。本書では簡素化した手順を説明します。

# 12.3.2.2. SAP のインストール

1. ASCS は、先のインストール手順でホストファイルに追加された「仮想 IP」に基づく「仮想ホスト名」に基づいてインストールする必要があります。まだ完了していない場合は、SAP のインストール前に必ず実行してください。

- SAP（具体的には ASCS または SCS）をインストールする場合、SAPINST_USE_HOSTNAME=vip オプションを指定する必要があります。これは ERS では必要ありません。（**注意：** SAPINST_USE_HOSTNAME 仮想 IP アドレスは後で LifeKeeper で SAP リソースを作成する際に使用するので、書き留めておいてください。）

  **注意：** `sapinst SAPINST_USE_HOSTNAME=vip` を指定してください。 **vip** は、ノード間でフロートする仮想IPです。

> **!** SAP は **ASCS および ERS インスタンスプロファイル設定の変更** で仮想ホスト名オプションを使用する必要があります。これは、SAPINST_USE_HOSTNAME=vip が LifeKeeper で動作するために必要なことです。SAP 構成を準備するための追加情報については、 SAP 構成を計画する を参照してください。

以下を実行します。

./sapinst SAPINST_USE_HOSTNAME= *{virtual hostname}*

- 7 段階にわけて、 **Core** サービス を作成して開始する必要があります。 *jdbcconnect.jar* で権限エラーが発生した場合は、 */sapmnt/STC/exe/uc/linuxx86_64* に移動し、このディレクトリでも *jdbcconnect.jar* を書き込み可能にしてください（chmod 777 —-）。

  ASCS プロファイルは、/usr/sap、sapmnt を含むローカルマウントポイント、またはご利用の環境のSAP ファイルに必要なその他のローカルマウントポイントを指す必要があります。

  エンキューサーバーおよびエンキューレプリケータープロセスが sapstart ユーティリティによって自動的に再起動されるのを防ぐために、ASCS および ERS インスタンスプロファイルを変更する必要があります。インスタンスプロファイルを更新した後、これらのインスタンスのSAP Start Serviceを再起動して、変更を有効にする必要があります。「 ASCS および ERS インスタンスプロファイル設定の変更 」に記載されている手順を実行し、その後このページに戻ってセットアッププロセスを続行してください。

2. ノード 1 で sapstop を実行して SAP を起動します。

3. LifeKeeper GUI を使用して、ASCS 用に作成された IP アドレスリソースを右クリックし、「In-Service」を選択し、ノード 2 を選択して IP をノード 2 に切り替えます。

4. ステップ 1 を繰り返してノード 2 に SAP をインストールし、正しく実行できることを確認します。

5. ノード 2 で sapstop を実行します。

6. LifeKeeper GUI を使用して ASCS 用に作成された IP アドレスリソースを右クリックし、「In-Service」を選択し、node1 を選択して IP をノード 1 に戻します。

7. ノード 1 で sapstart を実行し、正しく実行できることを確認します。

# 12.3.2.3. SAP リソース階層の作成

1. LifeKeeper GUI メニューから **[Edit]** を選択し、次に **[Server]** を選択します。ドロップダウンメニューから、 **[Create Resource Hierarchy]** 選択します。

   クラスター内で認識されたすべてのインストール済みのリカバリーキットがドロップダウンリストで表示されます。ドロップダウンリストから **SAP** を選択します。

   **[Next]** をクリックします。

   いずれかのダイアログボックスで **[Back]** ボタンが有効になっている場合は、前のダイアログボックスに戻ることができます。これは、以前に入力した情報を修正する必要のあるエラーが発生した場合に便利です。

   階層の作成中に **[Cancel]** をクリックすると、LifeKeeper は作成プロセス全体をキャンセルします。

2. **スイッチバックタイプ** を選択します。これは、バックアップサーバーへのフェールオーバー後にサービスに復帰したときに SAP インスタンスをこのサーバーにどのようにスイッチバックするかを指示します。 **Intelligent** または **Automatic** のいずれかを選択できます。 **Intelligent** スイッチバックでは、インスタンスをプライマリ/元のサーバーに戻すために管理者の介入が必要です。 **Automatic** スイッチバック とは、プライマリサーバーがオンラインに戻り、LifeKeeper のコミュニケーションパスを再確立するとすぐにスイッチバックが実行されることを意味します。

   スイッチバックタイプは必要に応じて、 **[Resource Properties]** ダイアログボックスの **[General]** タブで後から変更できます。

   **[Next]** をクリックします。

3. SAP PAS、ASCS または SCS を配置するサーバーを選択します（通常、これをプライマリまたはテンプレートサーバーと呼びます）。クラスター内のすべてのサーバーがドロップダウンリストボックスに含まれています。

4. **SAP SID** を選択します。これは、保護対象の SAP PAS、ASCS、または SCS システムのシステム識別子です。

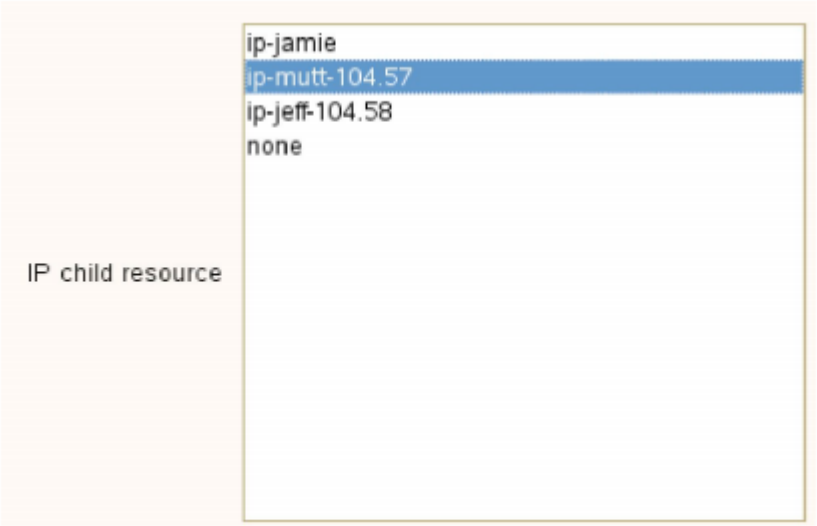
**[Next]** をクリックします。

5. 保護対象の SID の SAP インスタンス名（例：ASCS<*No.*>）（コアインスタンスが先頭）を選択します。

**[Next]** をクリックします。

**注記：** 保護およびリカバリーレベルのカスタマイズに関連する追加の画面が表示される場合があります。

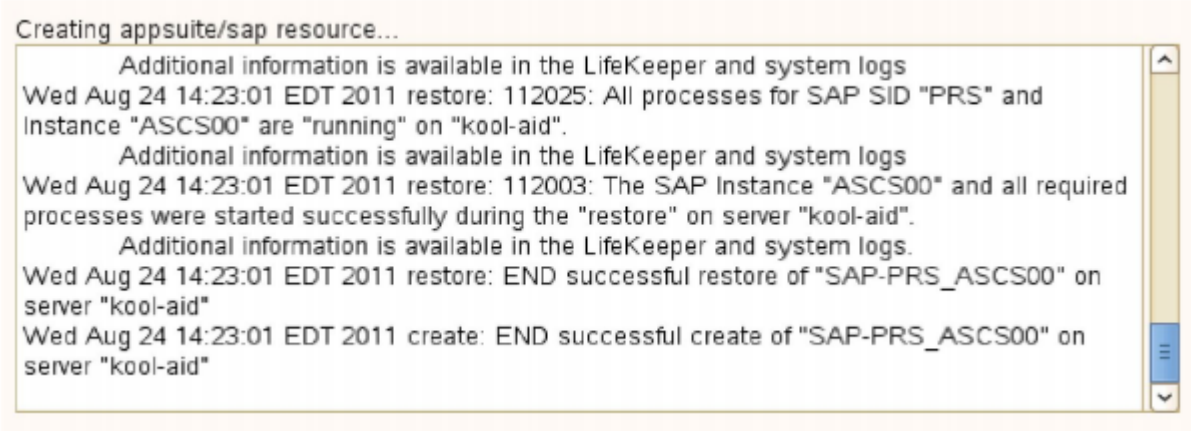
6. **IP の子リソース** を選択します。これは通常、SAP インストール時に指定された仮想ホスト IP アドレス（SAPINST_USE_HOSTNAME）またはフェールオーバーに必要な IP アドレスのいずれかです。




7. **SAP Tag** を選択または入力します。これは LifeKeeper が SAP の階層に付与するタグ名です。デフォルトを選択するか、独自のタグ名を入力することができます。デフォルトのタグは SAP-<SID>_<ID> です。

**[Create]** をクリックすると、 **SAP リソースの作成ウィザード** が SAP リソースを作成します。

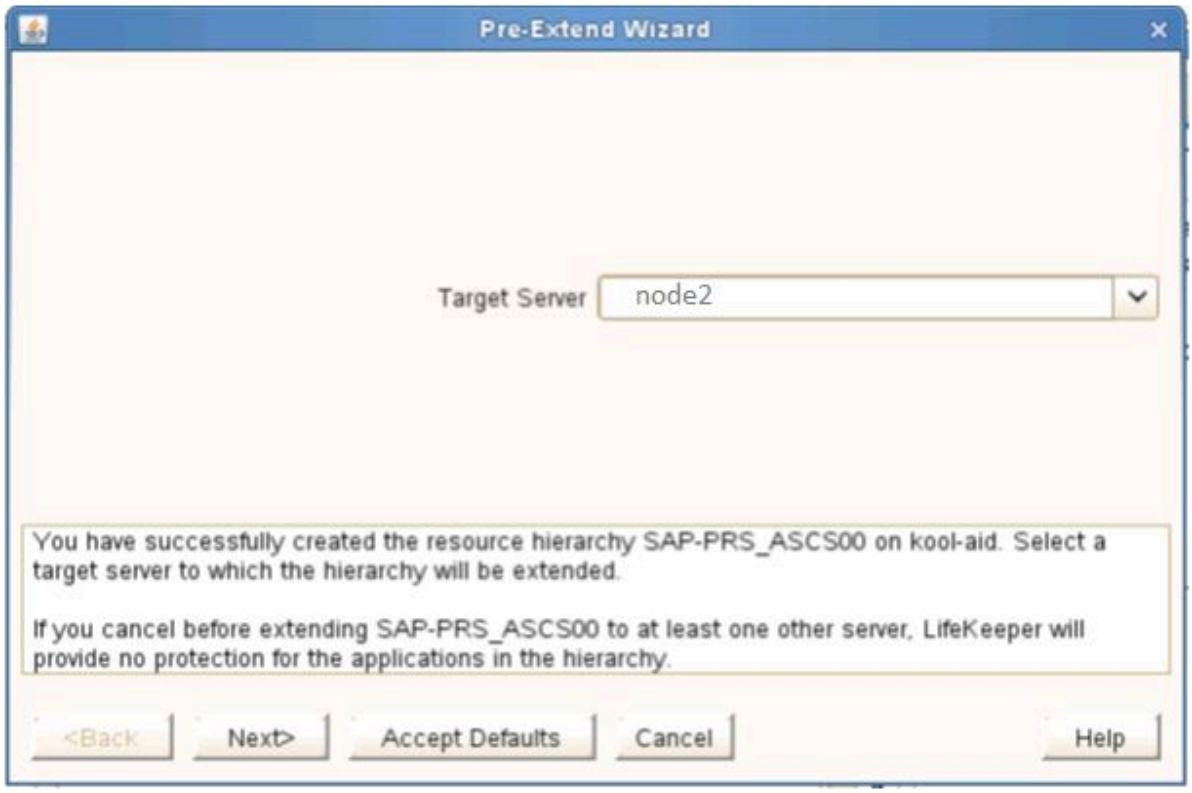
8. この時点で情報ボックスが表示され、LifeKeeper は SAP リソース階層を作成するための有効なデータが提供されているかどうかを検証します。LifeKeeper が問題を検出すると、情報ボックスに ERROR と表示されます。検証が成功すると、リソースが作成されます。情報ボックスに表示される SAP 起動スクリプトから出力されるエラーまたはメッセージが表示される場合もあります。




**[Next]** をクリックします。

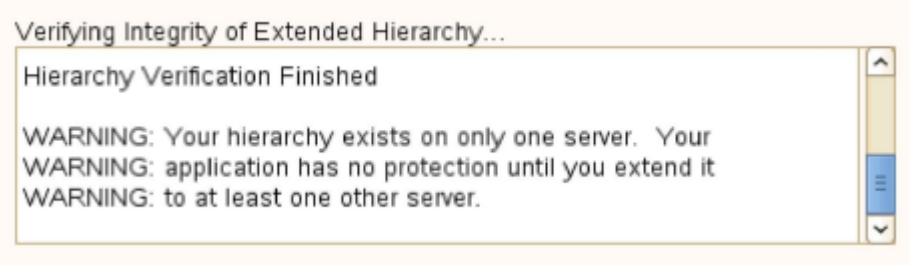
9. SAP リソース階層を正常に作成したことを示す別の情報ボックスが表示されます。そして、LifeKeeper 保護下に置くためにその階層をクラスター内の別のサーバーに **拡張** する必要があります。

**[Next]** をクリックすると、LifeKeeper はこのセクションで後述する ***Pre-Extend Wizard*** を起動します。

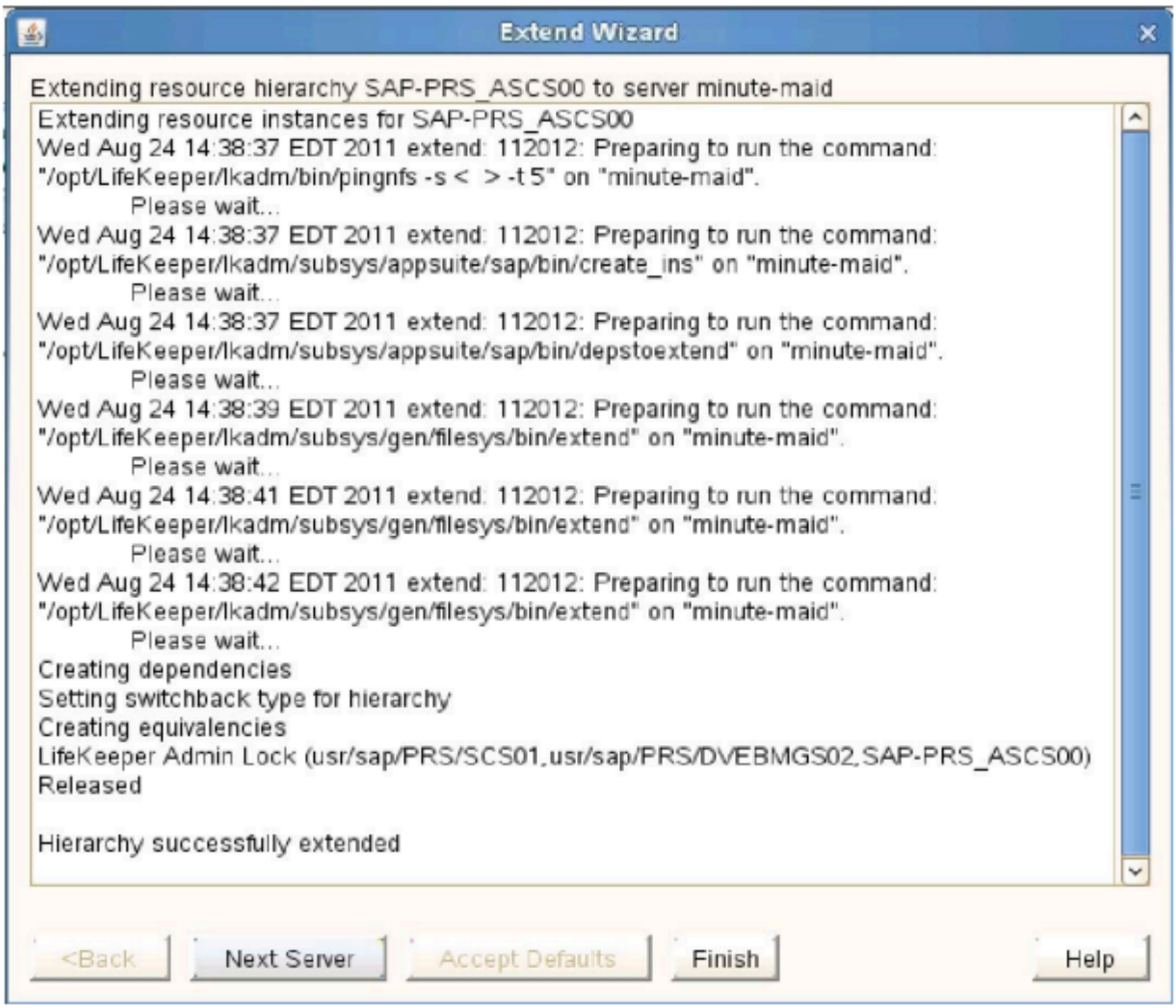



ここで **[Cancel]** をクリックすると、後で SAP リソース階層を LifeKeeper の保護下に置くにはこ

のステップに戻って SAP リソース階層を別のサーバーに拡張する必要があることを警告するダイアログボックスが表示されます。




10. 階層が正常に拡張された ことを示す **[Extend Wizard]** ダイアログが表示されます。 **[Finish]** をクリックします。




**[Hierarchy Integrity Verification]** ダイアログが表示されます。階層の検証が終了したら、 **[Done]** をクリックして **[Create Resource Hierarchy]** メニューを終了します。

### Hierarchy with the Core as the Top Level

SIOS Protection Suite を使用して PAS および AAS サーバーを保護することはできますが、ほとんどのお客様は、それらを追加の HA なしで独立したスタンバイサーバーとして使用しているだけです。このガイドでは保護手順については説明しないので、詳細と手順については SAP Recovery Kit を参照してください。

# 12.3.3. NFS を使用している ASCS + ERS 構成

> **!** NFS 共有をエクスポートするサーバー (またはクラスター) が、ASCS + ERS を実行しているサーバーとは別のサーバーグループである場合、NFS 共有を使用した ASCS + ERS がサポートされます。

NFS を使用して ASCS + ERS をセットアップする手順は次の通りです。

- 一般的なセットアップのステップ概要
- SAP のインストール
- NFS のセットアップ
- NFS リソース階層の作成
- SAP リソース階層の作成
- ERS リソースの作成

# 12.3.3.1. 一般的なセットアップ手順

1. 仮想 IP を作成し（先のステップでノード 1 で実施済み）、拡張します（先のステップで実施済み）。
2. EC2 リソースを作成し、仮想 IP の依存関係を作成します（先のステップで実施済み）。
3. 「仮想 IP」に基づく「仮想ホスト名」のノード 1 に SAP をインストールします。
4. ノード 1 で stopsap を実行します。
5. LifeKeeper GUI を使用して仮想 IP をノード 2 で「In-Service」にし、「仮想 IP」に基づく「仮想ホスト名」のノード 2 に SAP をインストールします。
6. ノード 2 で Stopsap を実行し、両方のノードでプロファイルファイルを変更します（下記参照）。
7. LifeKeeper GUI を使用して仮想 IP をノード 1 で「In-Service」に戻します。
8. SAP コンサルタントのアドバイスに従って、SAP に必要なマウントポイントのレプリケーションリソースを作成します（先のステップで実施済み）。
9. ノード 1 で startsap を実行し、SAP が正常に稼働していることを確認します。
10. 両方のノードの /etc/default/LifeKeeper で、次のエントリを最後に追加します。

    SAP_EXPERTMODE=1

    SAP_DB_CHECK_IGNORE=1

11. SPS セットアッププログラムを再実行して、SAP Recovery Kit を追加します。

    次のコマンドを使用して sps.img ファイルをマウントします（前述の手順に従ってダウンロードしてください）：

    mount <PATH/IMAGE_NAME> <MOUNT_POINT> -t iso9660 -o loop

    PATH：イメージへのパス

    IMAGE_NAME：イメージ名

    MOUNT_POINT：マウントする場所へのパス

    マウントされた sps.img ディレクトリに移動し、次のように入力します：

    ./setup

    Recovery Kit Selection Menu から SAP Recovery Kit をインストールしてください。SAP の

Recovery Kit を選択するには、矢印キーを使用して選択して <スペースバー>を押して決定し、<Enter>を押してインストールを続行します。

12. NFS サーバーをセットアップします。

13. ファイルシステムを SAP サーバーにコピーし、冗長化とフェールオーバーのために、ファイルシステムにレプリケーションリソースを作成します。

14. NFS Recovery Kit のガイド に従って、NFS リソースを作成します。

    こちら では簡略化した手順を説明します。

15. SAP Recovery Kit のガイド に従って、SAP リソースを作成します。

    こちら では簡略化した手順を説明します。

# 12.3.3.2. SAP のインストール（NFS使用）

1. ASCS と ERS は、「仮想 IP」の「仮想ホスト名」に基づいてインストールする必要があります。これは、以前のインストール手順でホストファイルに追加されています。まだ完了していない場合は、SAP のインストール前に必ず実行してください。

- SAPinst を起動する前に SAPinst プロパティ SAPINST_USE_HOSTNAME を設定し、必要な仮想ホスト名を指定します。（ **注意：** SAPINST_USE_HOSTNAME の仮想 IP アドレスをメモしておいてください。LifeKeeper で SAP リソースを作成する際に後で使用します。）

  **注意：** `sapinst SAPINST_USE_HOSTNAME=vip` を指定してください。 **vip** は、ノード間でフロートする仮想IPです。

> ! SAP は **ASCS および ERS インスタンスプロファイル設定の変更** で仮想ホスト名オプションを使用する必要があります。これは、SAPINST_USE_HOSTNAME=vip が LifeKeeper で動作するために必要なことです。SAP 構成を準備するための追加情報については、 SAP 構成を計画する を参照してください。

以下を実行します。

./sapinst SAPINST_USE_HOSTNAME= *{virtual hostname}*

- 7 段階にわけて、 **Core サービス** を作成して開始する必要があります。 *jdbcconnect.jar* で権限エラーが発生した場合は、 */sapmnt/STC/exe/uc/linuxx86_64* に移動し、このディレクトリでも *jdbcconnect.jar* を書き込み可能にしてください（chmod 777 —-）。

エンキューレプリケーションは、SAP のドキュメントおよびとベストプラクティスに基づいて構成し、動作を確認する必要があります。

2. ASCS および ERS プロファイルは、/usr/sap、sapmnt を含むローカルマウントポイント、またはご利用の環境の SAP ファイルに必要なその他のローカルマウントポイントを指す必要があります。実際のファイルは、インストールおよび構成後に NFS マウントポイントに移動されます。

エンキューサーバーおよびエンキューレプリケータープロセスが sapstart ユーティリティによって自動的に再起動されるのを防ぐために、ASCS および ERS インスタンスプロファイルを変更する必要があります。インスタンスプロファイルを更新した後、これらのインスタンスのSAP Start Serviceを再起動して、変更を有効にする必要があります。「 ASCS および ERS インスタンスプロファイル設定の変更 」に記載されている手順を実行し、その後このページに戻ってセットアッププロセスを続行してください。

3. ノード 1 で SAP の sapstop を実行します。

4. LifeKeeper GUI を使用して、ASCS 用に作成された IP アドレスリソースを右クリックし、「In-Service」を選択し、ノード 2 を選択して IP をノード 2 に切り替えます。

5. ステップ 1 を繰り返してノード 2 に SAP をインストールし、正しく実行できることを確認しま

す。

6. ノード 2 で SAP の sapstop を実行します。

7. LifeKeeper GUI を使用して、ASCS 用に作成された IP アドレスリソースを右クリックし、「In-Service」を選択し、ノード 1 を選択して IP をノード 1 に切り替えます。

8. ノード 1 で SAP の sapstop を実行し、正しく実行できることを確認します。

# 12.3.3.3. NFS のセットアップ

> **重要:** 業界のベストプラクティスに従い、SIOS では、SAP システムのセットアップ (「ループバック構成」とも呼ばれます) 時に、NFS サーバー/クライアントを SAP サービスと同じノードにインストールすることを推奨しなくなりました。 代わりに、次のような代替構成を使用することをお勧めします。
> • EFS in AWS
> • Azure Premium File Shares with NFS
> • Google NFS Share
> • NFS as separate cluster
> 今後、「ループバック構成」を使用した新規インストールはサポートされません。 問題が SIOS ソフトウェア内にある場合、既存のお客様は可能な範囲でサポートいたしますが、代替構成への移行が推奨されます。 次のリリースでは、既存の構成はサポートされません。

SIOS 製品をインストールする前に、両方のクラスターノードに NFS サーバーをインストールしておく必要があります。

お使いの SAP 設計での SAP 要件に基づいて NFS エクスポートを作成します。以下は参考例であり、お使いの SAP 環境を表すものではありません。

LifeKeeper は inode を使用して NFS 共有情報を管理します。したがって、すべての NFS 共有には固有の inode が必要です。すべてのファイルシステムの root ディレクトリは同じ inode を持つので、LifeKeeper で保護するためには、NFS 共有は root から少なくとも 1 つ下のディレクトリレベルでなければなりません。

たとえば、上記の情報を参照すると、 */usr/sap/trans* ディレクトリが SAP サーバー上で NFS 共有である場合、共有ストレージデバイス上に */trans* ディレクトリが作成され、共有ストレージデバイスを */usr/sap* としてマウントすることが求められます。 しかし、 */usr/sap* 配下にあるすべてのファイルをこの配列で必要な共有ストレージに置くことは必ずしも望ましいとは限りません。

この問題を回避するには、NFS 共有のディレクトリを含むすべての共有ファイルシステムをマウントするための */exports* ディレクトリツリーを作成するか、SAP ディレクトリと */exports* ディレクトリの間にソフトリンクを作成することを推奨します。または、NFS 共有ディレクトリをローカルで NFS マウントすることを推奨します。 ( **注記：** ここで */exports* と記載しているディレクトリの名前は、ユーザーの設定よって異なる場合があります。ただし、わかりやすくするために、本書では全体を通じてこのディレクトリを */exports* と記載します。 )

たとえば、SAP プライマリサーバーでのこの例のディレクトリとリンク/マウントは、次のようになります。

| **For the */usr/sap/trans* share** | |
|---|---|
| **Directory** | **Notes** |
| */trans* | created on share file system and shared through NFS |
| */exports/usr/sap* | mounted to / (on shared file system) |

| | |
|---|---|
| */user/sap/trans* | soft linked to */exports/usr/sap/trans* |

<sapmnt>/<SAPSID> 共有のディレクトリとリンクは次のようになります。

| For the <sapmnt>/<SAPSID> share | |
|---|---|
| **Directory** | **Notes** |
| /<SAPSID> | created on shared file systems and shared through NFS |
| */exports/sapmnt* | mounted to / (on shared file system) |
| <sapmnt>/<SAPSID> | NFS mounted to *<virtual SAP server>:/exports/sapmnt/<SAPSID>* |

## ローカル NFS マウント

LifeKeeper 環境における SAP の推奨ディレクトリ構造では、1 つ以上の SAP システムディレクトリに対してローカルにマウントされた NFS 共有が必要です。

ローカルにマウントされた NFS 共有のいずれかの NFS エクスポートポイントが使用できなくなった場合、エクスポートポイントが再び利用可能になるのを待っている間にシステムがハングすることがあり、システムの再起動を含め、多くのシステム操作が正しく機能しなくなります。SAP クラスター用の NFS サーバーは LifeKeeper で保護されている必要があり、ローカルマウントポイントが存在する間は手動でサービスを停止しないでください。

## <INST>ディレクトリの場所

/usr/sap/<SAPSID> パスは NFS 共有ではないため、ファイルシステムのルートディレクトリにマウントできます。/usr/sap/<SAPSID> パスには、SYS サブディレクトリと、サーバー上で実行できる各 SAP インスタンスの <INST> サブディレクトリが含まれています。構成によっては <INST> ディレクトリは 1 つしかないので、共有ファイルシステムの /usr/sap/<SAPSID> の下に置いても問題ありません。

ただし他の構成では、バックアップサーバーにローカル AS インスタンスが含まれている可能性があります。このローカル AS インスタンスの <INST> ディレクトリは常時利用できるわけではなく、共有ファイルシステム上に置くべきではありません。この問題を解決するには、ある構成については、ローカルサーバーに配置する必要のある AS の /usr/sap/<SAPSID>、/usr/sap/<SAPSID>/SYS および /usr/sap/<SAPSID>/<AS-INST> の代わりに、PAS、ASCS または SCS の /usr/sap/<SAPSID>/<INST>、/usr/sap/<SAPSID>/<ASCS-INST> または /usr/sap/<SAPSID>/<SCS-INST> ディレクトリを共有ファイルシステムにマウントすることを推奨します。

たとえば、次のディレクトリとマウントポイントを ABAP + Java 構成用に作成する必要があります。

| Directory | Notes |
|---|---|
| *usr/sap/<SAPSID>/DVEBMS<**No.**>* | mounted to / (Replicated non-NFS file system ) |

| | |
|---|---|
| *usr/sap/<SAPSID>/SCS<**No.**>* | mounted to / (Replicated non-NFS file system ) |
| *usr/sap/<SAPSID>/ERS<**No.**>*<br><br>*(for SCS instance)* | should be locally mounted on all cluster nodes or mounted from a NAS share (*should not be mounted on shared storage*) |
| *usr/sap/<SAPSID>*<br>*/ASCS<**Instance No.** >* | mounted to / (Replicated from non-NFS file system) |
| *usr/sap/<SAPSID>/ERS<**No.**>*<br><br>*(for ASCS instance)* | should be locally mounted on all cluster nodes or mounted from a NAS share (*should not be mounted on shared storage*) |
| *usr/*<br>*sap/<SAPSID>AS<**Instance No.** >* | created for AS backup server |

## NFS のマウントとファイルシステムの移動

メイン SAP ファイルシステムのマウントポイントが作成されたら、必要に応じてマウントします。この時点で、これらの手順に進む前にすべての SAP サービスを停止します。

**mount /dev/sap/sapmnt /exports/sapmnt**

**mount /dev/sap/saptrans /exports/saptrans**

## データの NFS への移動

1. */etc/exports* を編集し、SAP のメインディレクトリのマウントポイントを挿入します。

```
/exports/sapmnt * (rw,sync,no_root_squash)

/exports/saptrans * (rw,sync,no_root_squash)
```

## NFS エクスポートの例

各 <nfsvip>を、その NFS 共有で使用される適切な仮想 IP に置き換えます。SAP システムの設計に応じて、異なる仮想 IP を使用して異なるファイルシステムを共有できます。

# more /etc/exports

/exports/sapmnt 10.2.0.69(rw,sync,all_squash,anonuid=0,anongid=1001)

/exports/sapmnt 10.2.0.11(rw,sync,all_squash,anonuid=0,anongid=1001)

/exports/usr/sap/<instance name>/ASCS01 10.2.0.69(rw,sync,all_squash,anonuid=0,anongid=1001)

/exports/sap/<instance name>/ASCS01
10.2.0.11(rw,sync,all_squash,anonuid=0,anongid=1001)

# more /etc/fstab

#

# /etc/fstab

# Created by anaconda on Mon Nov 9 20:20:10 2015

#

# Accessible filesystems, by reference, are maintained under '/dev/disk'

# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info

#

UUID=367df610-4210-4a5a-8c8d-51ddf499fc17 / xfs defaults 0 0

/dev/xvdb swap swap defaults 0 0

/dev/xvdc /tmp xfs nodev,nosuid,noexec,relatime 0 0

/dev/xvdp1 /var xfs defaults 0 0

/dev/xvdp2 /var/log xfs defaults 0 0

/dev/xvdp3 /var/log/audit xfs defaults 0 0

/dev/xvdp4 /home xfs defaults,nodev 0 0

/tmp /var/tmp none bind,nodev,nosuid 0 0

/dev/xvdj /usr/sap xfs defaults 0 0

/dev/xvdg /exports/usr/sap/P4G/ASCS01 xfs defaults 0 0

/dev/xvdh /usr/sap/P4G/D00 xfs defaults 0 0

/dev/xvdi /sapcd xfs defaults 0 0

/dev/xvdk /exports/sapmnt xfs defaults 0 0

**<nfsvip>**:/exports/usr/sap/P4G/ASCS01 /usr/sap/**<instance name>**/ASCS01 nfs nfsvers=3,proto=udp,rw,sync,bg 0 0

**<nfsvip>**:/exports/sapmnt /sapmnt nfs nfsvers=3,proto=udp,rw,sync,bg 0 0

**<nfsvip>**:/exports/usr/sap/P4G/ASCS01 /usr/sap/PG4/ASCS01 nfs nfsvers=3,proto=udp,rw,sync,bg 0 0 (**Note:** This ERS entry will only be present if using an ERSv2 configuration with a shared ERS filesystem.)

2. `systemctl start nfs-server.service` コマンドを使用して NFS サーバーを起動します。NFS サーバーがすでにアクティブな場合は、"`exportfs -va`" を実行してこれらのマウントポイントをエクスポートします。

3. ノード 1 および 2 の両方で、次のマウントコマンドを実行します（**udp の使用方法に注意してください。これはフェールオーバーとリカバリーで重要です**）。これにより、NFS 共有をマウントできるようになります。

   mount {virtual ip}:/exports/sapmnt/<PG4> /sapmnt/<PG4> -o rw,sync,bg,udp

   mount {virtual ip}:/exports/saptrans /usr/sap/trans -o rw,sync,bg,udp

4. ノード 1 から、必要なファイルシステムを、/usr/sap および/sapmnt またはその他の必要なファイルから、NFS サーバーからノード 1 にマウントされた NFS マウントポイントにコピーします。

5. SAP にログインし、SAP を起動します（su の後に stcadm を実行します）。

   ```
   startsap sap{No.}
   ```

6. すべてのプロセスが開始されていることを確認してください。

   ```
   ps -ef | grep en.sap (2 processes)

   ps -ef | grep ms.sap (2 processes)

   ps -ef | grep dw.sap (17 processes)
   ```

   「SAP Logon」または「SAP GUI for Windows」は、SAP が提供する Windows クライアントです。プログラムは SAP のダウンロードサイトからダウンロードできます。仮想 IP アドレスは **[Properties]** ページで「アプリケーションサーバー」として使用できます。これにより、仮想 IP が存在するプライマリマシンへの接続がアクティブになります。

7. まだ実行していない場合は、NFS 共有マウントポイントにデータレプリケーションクラスターリソースを作成して、ノード 1 からノード 2 にデータを複製します。

## SAP で NFSv4/TCP を使用するときにスイッチオーバー時間を短縮

SAP/NFS 環境では次の変更を行うことでスイッチオーバー時間を短縮することができます。

すべての LK NFS リソースを out of service にして以下を実行します。

```
# systemctl stop nfs
# echo 10 > /proc/fs/nfsd/nfsv4gracetime
# echo 10 > /proc/fs/nfsd/nfsv4leasetime
# systemctl start nfs
```

**Note:** LEASE 時間は GRACE 時間と等しくなければならないので、両方を設定します。この変更により、NFSリソースの remove にかかる時間がデフォルトの 90 秒から10 秒になります。

LEASE 時間と GRACE 時間を永続的に設定するには、次のようにします。
RHEL 8 では以下を実行します。

```
# nfsconf —set nfsd grace-time 10
# nfsconf —set nfsd lease-time 10
# systemctl restart nfs-server
```

# 12.3.3.4. NFS リソース階層の作成

複数の NFS 共有の複数のリソース階層を作成するには、以下の手順を繰り返す必要があります。プライマリサーバーからリソースインスタンスを作成するには、次の手順を実行してください。

1. LifeKeeper GUI メニューから **[Edit]** → **[Server]** を選択し、ドロップダウンメニューから **[Create Resource Hierarchy]** を選択します。

   すでに選択されている内容を変更する場合や、NFS リソース階層の作成中にエラーメッセージが表示された場合は、 **[Back]** ボタンを使用して選択内容を変更するか、または修正します（ **[Back]** ボタンが有効な場合）。

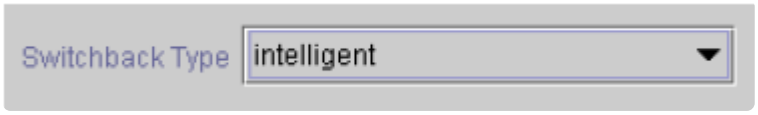
   ダイアログボックスに、クラスター内にインストールされ認識されているすべての Recovery Kit の一覧がドロップダウンメニューで表示されます。ドロップダウンメニューから **NFS** を選択します。

   **[Next]** をクリックして次のダイアログボックスに進みます。

   注記： 階層の作成中に **[Cancel]** ボタンをクリックすると、LifeKeeper は作成プロセス全体をキャンセルします。

2. スイッチバックタイプ を選択します。これは、バックアップサーバーへのフェールオーバー後にサービスに戻るときに、NFS インスタンスがプライマリサーバーに戻される方法を指示します。 **Intelligent** または **Automatic** のいずれかを選択してください。 **Intelligent** スイッチバック では、インスタンスをプライマリ/元のサーバーに戻す際に管理者の介入が必要です。 **Automatic** スイッチバック では、プライマリサーバーがオンラインに戻って LifeKeeper 保護下に入るとすぐにスイッチバックが行われます。

   スイッチバックタイプは、必要に応じて、 **[Resource Properties]** ダイアログボックスの **[General]** タブから後で変更できます。

3. NFS リソースを作成する サーバー を選択します（通常、これはプライマリまたはテンプレートサーバーと呼ばれます）。クラスター内のすべてのサーバーがドロップダウンメニューに含まれています。

   **[Next]** をクリックして次のダイアログボックスに進みます。

4. **Export Point** ダイアログには、次の基準を満たしている NFS ファイルシステムのエクスポートポイントのドロップダウンリストが表示されます。

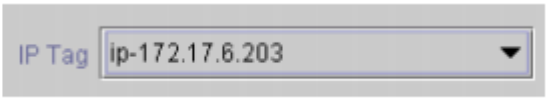
- エクスポートポイントは NFS によってエクスポートされている。
- エクスポートポイントは共有ドライブ上にある。
- 基礎となるファイルシステムが LifeKeeper で保護されている場合、そのファイルシステムは In-Service 状態で、 **サーバー** ダイアログで選択したサーバーで優先度の最も高いファイルシステムである必要があります。

- NFSv4 の基準：
  - NFS v4 のルートエクスポートがすでに保護されている場合は、選択肢はありません（v4 が 1 つのみで存在する必要があります。v4 と v2/v3 が混在すると保護できません）。
  - NFS v2/v3 がすでに保護されている場合は、選択肢に NFS v4 はリストされません。
  - 何も保護されていない場合、リストには v2/v3 と v4 の両方が含まれます。

ドロップダウンリストから、保護する NFS エクスポートポイントを選択します。

**[Next]** をクリックして次のダイアログボックスに進みます。

5. **[IP Tag]** ダイアログには、LifeKeeper 保護下にあり、NFS リソースが作成されているサーバー上で In-Service 状態の仮想 IP アドレスに対応するタグのドロップダウンリストが表示されます。クライアントが保護された NFS ファイルシステムにアクセスするために使用する仮想 IP アドレスの タグ を選択します。




**注記：** この時点で、LifeKeeper は利用可能な保護された IP リソースがあることを確認します。また、NFS リソース階層を作成するために有効なデータが提供されているかどうか検証します。LifeKeeper がこれらの検証のいずれかで問題を検出すると、ERROR ボックスが画面に表示されます。ディレクトリパスは有効であるのに NFS の設定自体にエラーがある場合、一時停止してこれらのエラーを修正し、階層作成を続行することができます。必要な LifeKeeper の IP リソースを作成するために一時停止することもできます。

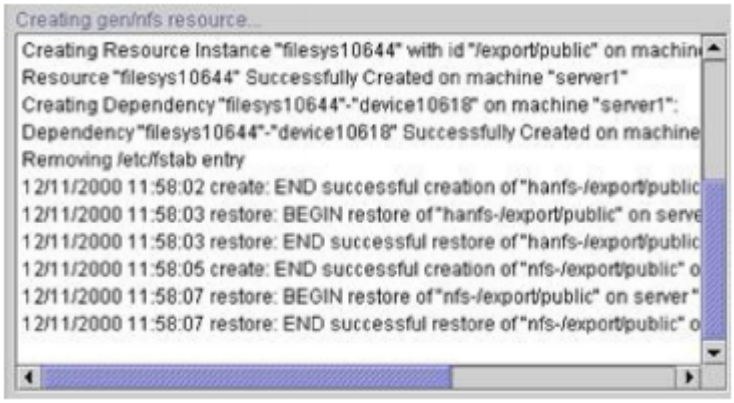
**注記：** 仮想 IP アドレスの依存関係を持つ他の LifeKeeper Recovery Kit を使用している場合は、NFS リソース用に別の仮想 IP アドレスを作成する必要があります。そうしないと、仮想 IP リソースがバックアップサーバーにフェールオーバーした場合、その IP リソースに依存するすべてのリソースが同時にフェールオーバーされてしまいます。

**[Next]** をクリックして次のダイアログボックスに進みます。

6. **NFS タグ** を選択または入力します。これは NFS 階層に与えられたタグ名です。デフォルトを選択

するか、独自のタグ名を入力することができます。

**[Create]** ボタンをクリックすると、 **リソース作成ウィザード**が NFS リソースを作成します。




**[Next]** をクリックすると、LifeKeeper は階層の拡張で説明している ***Pre-Extend Wizard*** を起動します。

**注記：** この時点で NFS リソース階層が正常に作成されている必要があります。ただし、新しい NFS インスタンスが正しく起動できなかったことを示すエラーメッセージが表示されることがあります。新しい NFS 階層は、別のシステムに拡張する前に起動する（In-Service 状態にする）必要があります。

起動に失敗すると階層が削除されることがありますが、そうでない場合は、この時点で一時停止し、表示されるエラーメッセージに基づいて問題を修正することができます。エラーを修正できない場合はキャンセル以外に選択肢はなく、リソース作成がキャンセルされます。

階層の拡張に進む前に、新しい階層を In-Service にしてください。

*** 各 NFS 共有の追加のリソース階層を作成するには、上記の手順を繰り返してしてください。

**注記：** RHEL 7.1 以降および SLES12 SP1 以降で NFS リソースを作成した後、nfs-server.service の自動起動を無効にしてください。NFS リソースの起動時に rpcbind.service を実行する必要があるため、 rpcbind.service を自動的に起動するように設定します。

# 12.3.3.5. SAP リソース階層の作成

1. 1. LifeKeeper GUI メニューから、 **[Edit]** → **[Server]** を選択します。 ドロップダウンメニューから、 **[Create Resource Hierarchy]** を選択します。

   クラスター内にインストールされ認識されているすべての Recovery Kit 一覧のドロップダウンリストのあるダイアログボックスが表示されます。 ドロップダウンリストから **SAP** を選択します。

   **[Next]** をクリックします。

   いずれかのダイアログボックスで **[Back]** ボタンが有効になっている場合は、前のダイアログボックスに戻ることができます。 これは、以前に入力した情報を修正する必要のあるエラーが発生した場合に特に役立ちます。

   階層の作成中に **[Cancel]** をクリックすると、LifeKeeper は作成プロセス全体をキャンセルします。

2. スイッチバックタイプ を選択します。 これは、バックアップサーバーへのフェールオーバー後にサービスに復帰したときに SAP インスタンスをこのサーバーに戻す方法を指示します。 **Intelligent** または **Automatic** のいずれかを選択できます。 **Intelligent** スイッチバック では、インスタンスをプライマリサーバー/元のサーバーに戻すために管理者の介入が必要です。 **Automatic** スイッチバック では、プライマリサーバーがオンラインに戻り、LifeKeeper のコミュニケーションパスが再確立されるとすぐにスイッチバックが行われます。

   スイッチバックタイプは、後で **[Resource Properties]** ダイアログボックスの **[General]** タブから変更できます。

   **[Next]** をクリックします。

3. SAP PAS、ASCS または SCS を配置するサーバーを選択します（通常、これをプライマリまたはテンプレートサーバーと呼びます）。 クラスター内のすべてのサーバーがドロップダウンリストボックスに含まれています。

4. **SAP SID** を選択します。 これは、保護対象の SAP PAS、ASCS、または SCS システムのシステム識別子です。

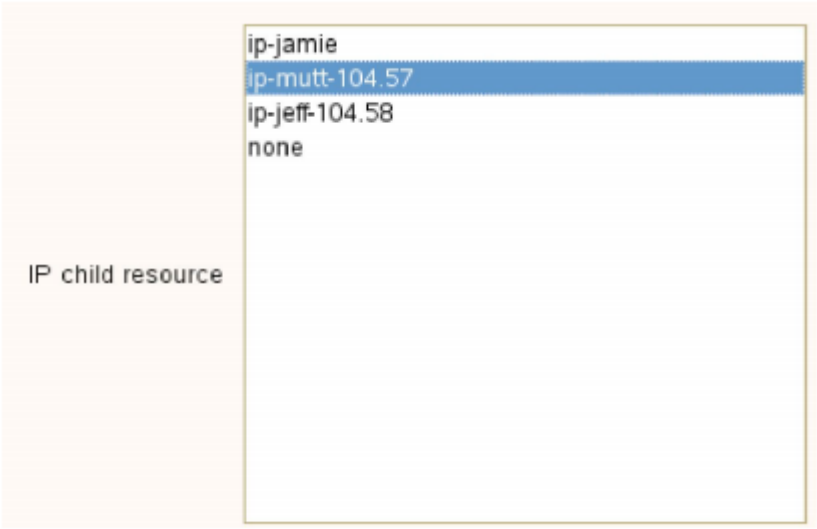
**[Next]** をクリックします。

5. 保護対象の SID の SAP インスタンス名（例：ASCS<*No.*>）（コアインスタンスが先頭）を選択します。

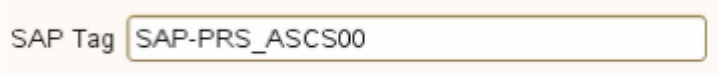
**[Next]** をクリックします。

**注記：** 保護およびリカバリーレベルのカスタマイズに関連して、追加の画面が表示されることがあります。

6. **IP の子リソース** を選択します。これは通常、SAP インストール時に指定された仮想ホスト IP アドレス（SAPINST_USE_HOSTNAME）またはフェールオーバーに必要な IP アドレスのいずれかです。



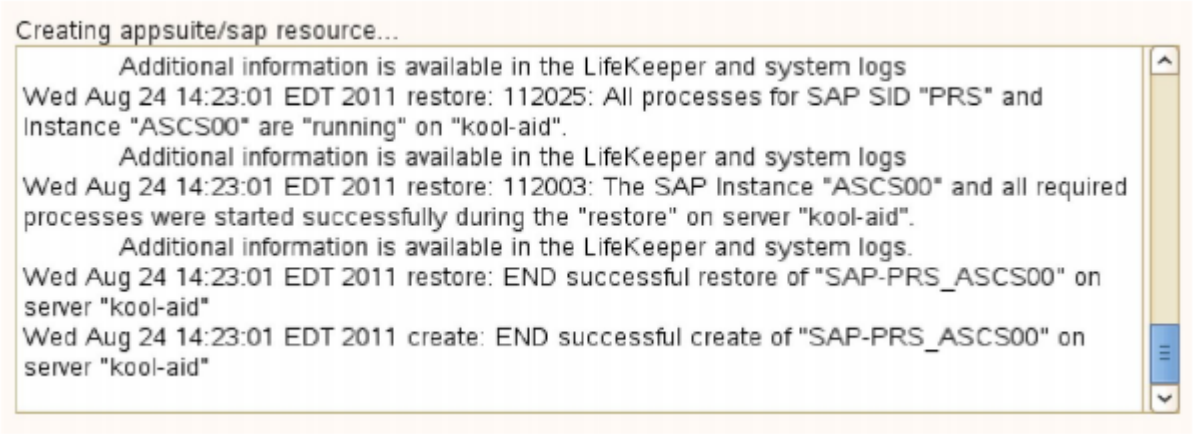

7. **SAP Tag** を選択または入力します。これは LifeKeeper が SAP 階層に付与するタグ名です。デフォルトを選択するか、独自のタグ名を入力することができます。デフォルトのタグは *SAP-<SID>_<ID>* です。




**[Create]** をクリックすると、 **SAP リソース作成ウィザード** が SAP リソースを作成します。

8. ここで情報ボックスが表示され、LifeKeeper は SAP リソース階層を作成するために有効なデータが提供されているかどうか検証します。LifeKeeper が問題を検出すると、ERROR が情報ボックスに

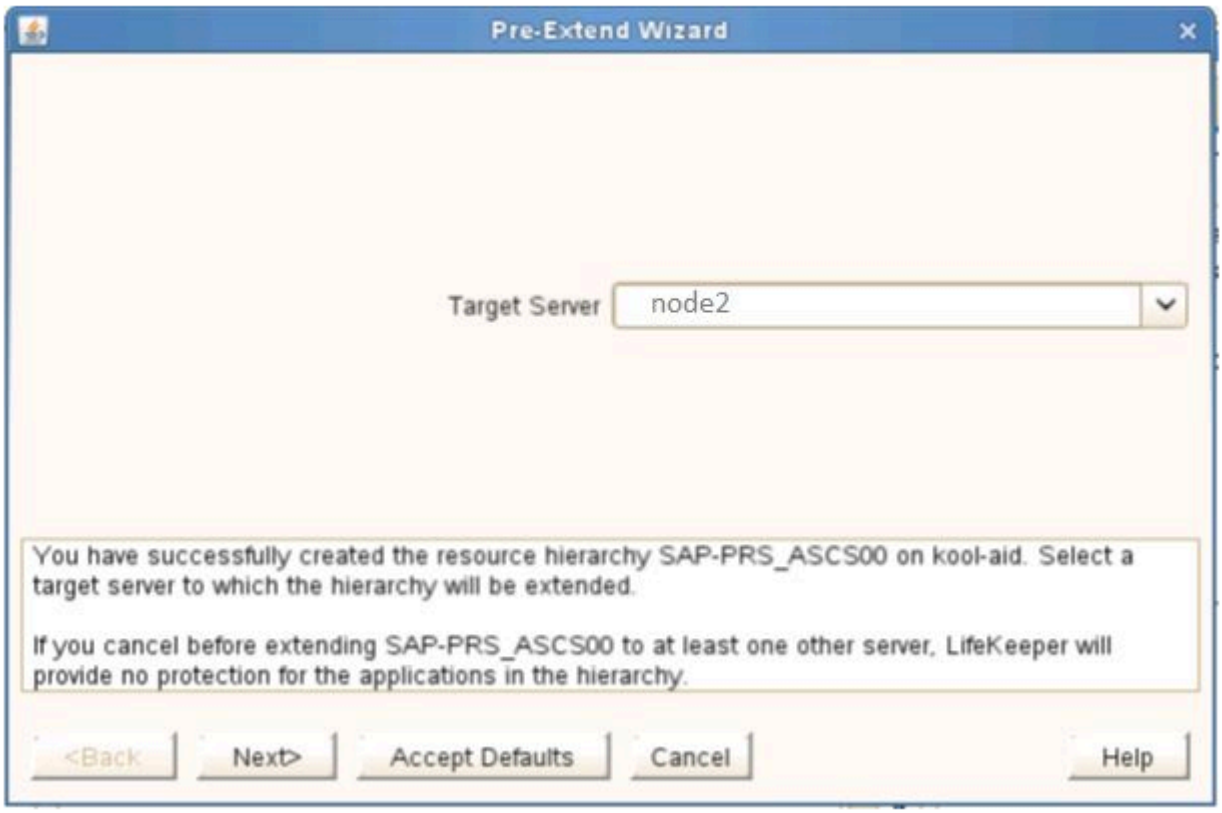
表示されます。検証が成功すると、リソースが作成されます。情報ボックスに表示される SAP 起動スクリプトからエラーまたはメッセージが出力される場合もあります。



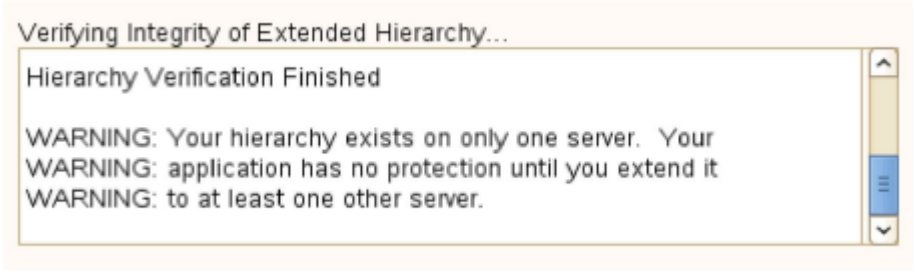

**[Next]** をクリックします。

9. SAP リソース階層が正常に作成されたことを示す別の情報ボックスが表示されます。その階層を LifeKeeper 保護下に置くためには、クラスター内の別のサーバーに **拡張** する必要があります。

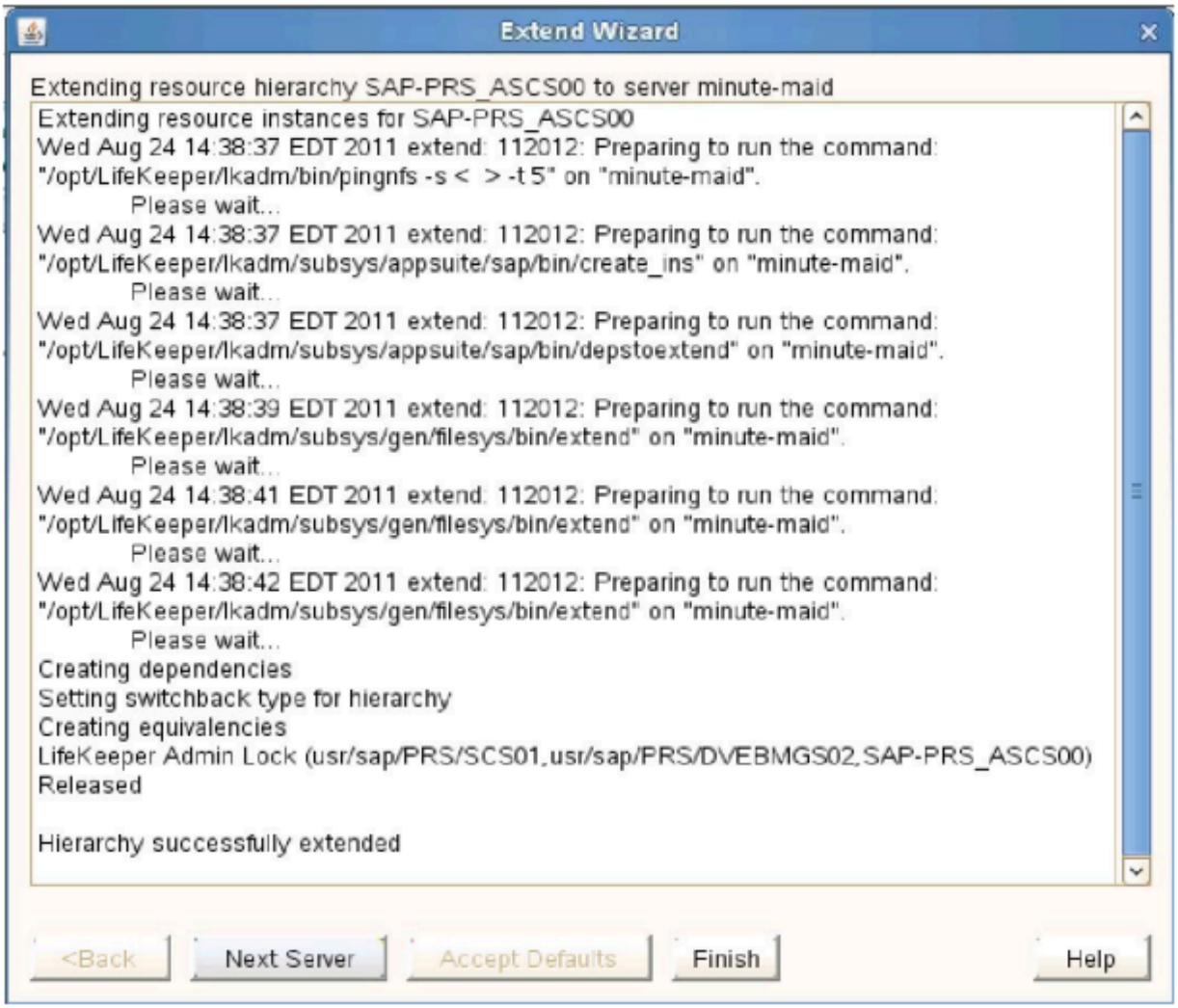
   **[Next]** をクリックすると、LifeKeeper は ***[Pre-Extend Wizard]*** を起動します。このウィザードについては、本セクションで後述します。




ここで **[Cancel]** をクリックすると別のダイアログボックスが表示され、いずれかの時点でここに戻り、SAP リソース階層を別のサーバーに拡張して LifeKeeper の保護下に置く必要があることが警告されます。




10. 階層が正常に拡張された ことを示す ***Extend Wizard*** ダイアログが表示されます。 **[Finish]** をクリックしてください。




**[Hierarchy Integrity Verification]** ダイアログが表示されます。 階層の検証が終了したら、 **[Done]** をクリックして、 **[Create Resource Hierarchy]** メニューを終了します。

## Hierarchy with the Core as the Top Level

# 12.3.3.6. ERS リソースの作成

ERS リソースは、コアインスタンス（セントラルサービスインスタンス）またはエンキューサーバープロセスの単一障害点に対する追加の保護を提供します。コアインスタンス（セントラルサービスインスタンス）で障害が発生して再起動されると、ERS リソースはロックテーブルとトランザクションの現在のステータスを取得します。その結果、エンキューサーバーに障害が発生した場合でもトランザクションや更新は失われず、SAP システムのサービスは継続されます。

この ERS リソースを作成するには、以下の手順を実行してください。

1. この同じ SAP SID について上記の手順を繰り返し、 **[ERS instance]** を選択して、ERS リソースを作成します。

2. その後、 **[Dependent Instances]** を選択するよう要求されます。上記で作成した **コアリソース** を選択し、 **[Next]** をクリックします。

3. 画面の指示に従って **リソース階層を拡張** してください。

4. 「 **Hierarchy Successfully Extended** 」と表示されたら、 **[Finish]** をクリックします。

5. **[Done]** を選択してください。

   **注記：** エンキューレプリケーションサーバ (ERS) はクラスターの各ノードで In-Service（ISP）になります。ただし、ERS のアーキテクチャと機能では、インスタンスの実際のプロセスはバックアップノードで実行する必要があります。これにより、スタンバイサーバーは、プライマリサーバーおよびプライマリエンキューサーバーインスタンスのロックテーブル情報の完全なコピーを保持できるようになります。エンキューサーバーを実行しているプライマリサーバーで障害が発生すると、ERS プロセスが現在実行されているバックアップサーバー上の SIOS Protection Suite によって再起動されます。ERS に格納されているロックテーブル（レプリケーションテーブル）は復旧中のエンキューサーバープロセスに転送され、そこから新しいロックテーブルが作成されます。このプロセスが完了すると、アクティブなレプリケーションサーバーは非アクティブになります（エンキューサーバーへの接続が切断され、レプリケーションテーブルが削除されます）。SIOS Protection Suite は、新しいバックアップノード（以前はプライマリノード）上の、今まで非アクティブだった ERS プロセスを再起動します。ERS プロセスがアクティブになると、エンキューサーバーに接続してレプリケーションテーブルを作成します。ERS プロセスがアクティブになると、エンキューサーバーに接続してレプリケーションテーブルを作成します。

   ERS プロセスおよび SAP アーキテクチャの機能の詳細については、 http://help.sap.com にアクセスし、 **Enqueue Replication Service** を検索してください。

**Hierarchy with ERS as Top Level**

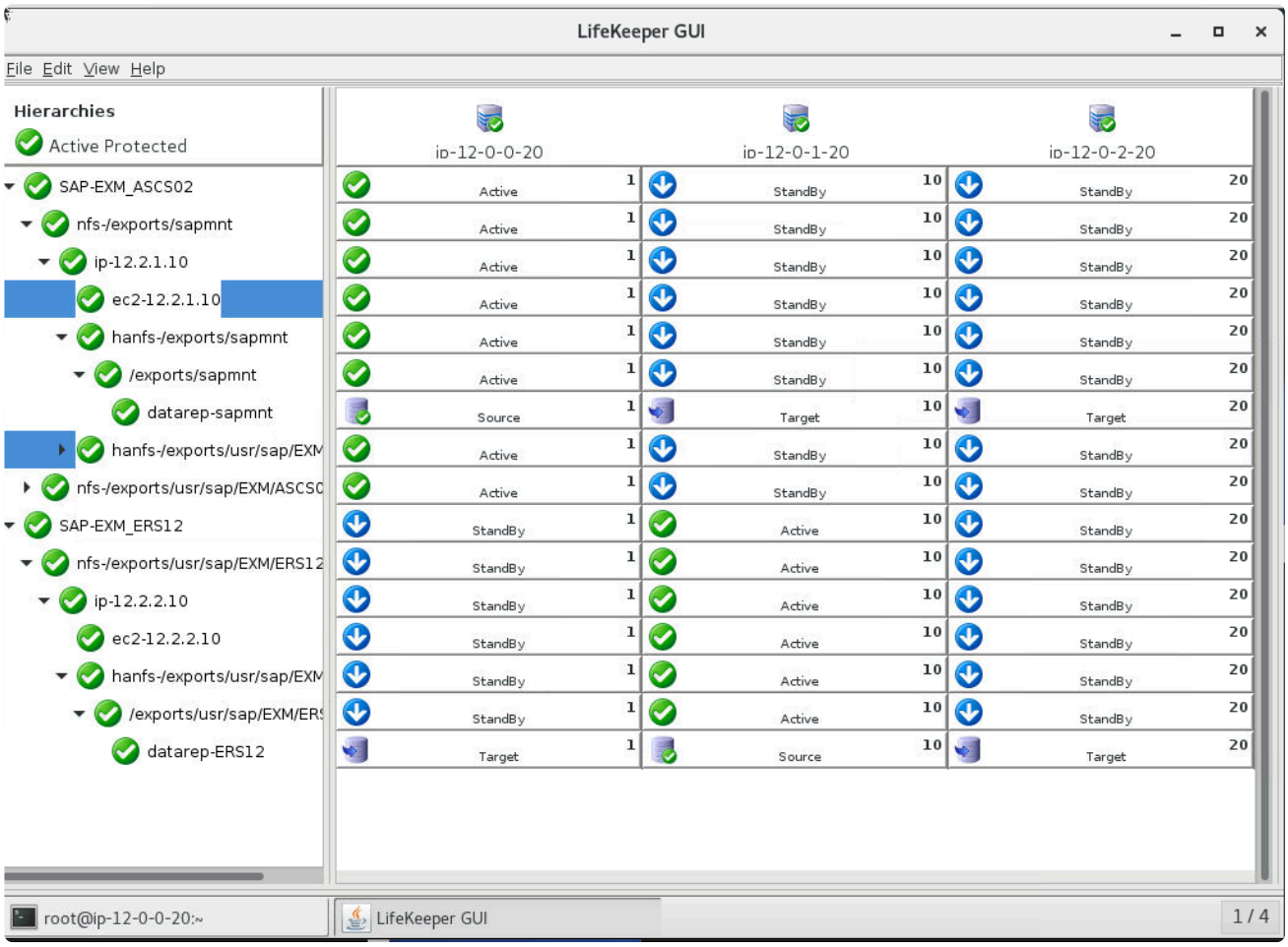
SIOS Protection Suite を使用して PAS および AAS サーバーを保護することはできますが、ほとんどのお客様は、それらを追加の HA なしで独立したスタンバイサーバーとして使用しているだけです。このガイドでは保護手順については説明しないので、詳細と手順については SAP Recovery Kit を参照してください。

# 12.3.3.7. ENSAv2/ERSv2使用時にASCS/ERSに回避動作を強制する

ERSv2は、ASCSリソースがアクティブでないクラスター内のノードでアクティブ（In-service）になります。ERS quickCheckは、ERSとASCSが同じノードでアクティブであり、別のノードが利用可能な場合、ERS階層を自動的に転送します。スイッチオーバーまたはフェイルオーバー後に同じノードでASCSとERSの両方がアクティブ（In-service）になる状況を回避するために、gen/appターミナルリーフリソースを作成して、対応するリソース階層がアクティブ（In-service）ではないノードにIn-service状態を自動的にルーティングできます。このターミナルリーフノードの作成を容易にするために、新しいユーティリティ /opt/LifeKeeper/bin/create_terminal_leaf (1M) が提供されています。

回避ターミナルリーフを作成するために、ユーティリティは2つのパラメーター、ASCSおよびERS階層ルートリソースタグを使用します。2つの階層は、クラスター内のすべてのノードに完全に拡張し、クラスター内のノードでIn-serviceにする必要があります。ユーティリティがクラスター内のノードで実行されている限り、ユーティリティが実行されているノードで階層をIn-serviceにする必要はありません。tagが回避する適切なルートノードである場合、ターミナルリーフノードには「avoid_<tag>」という名前が付けられます。ターミナルリーフノードは、階層の各ブランチの子依存関係としてアタッチされます。

たとえば、SAP-EXM_ASCS02およびSAP-EXM_ERS12をルートノードとして使用する構成では、次のようになります。

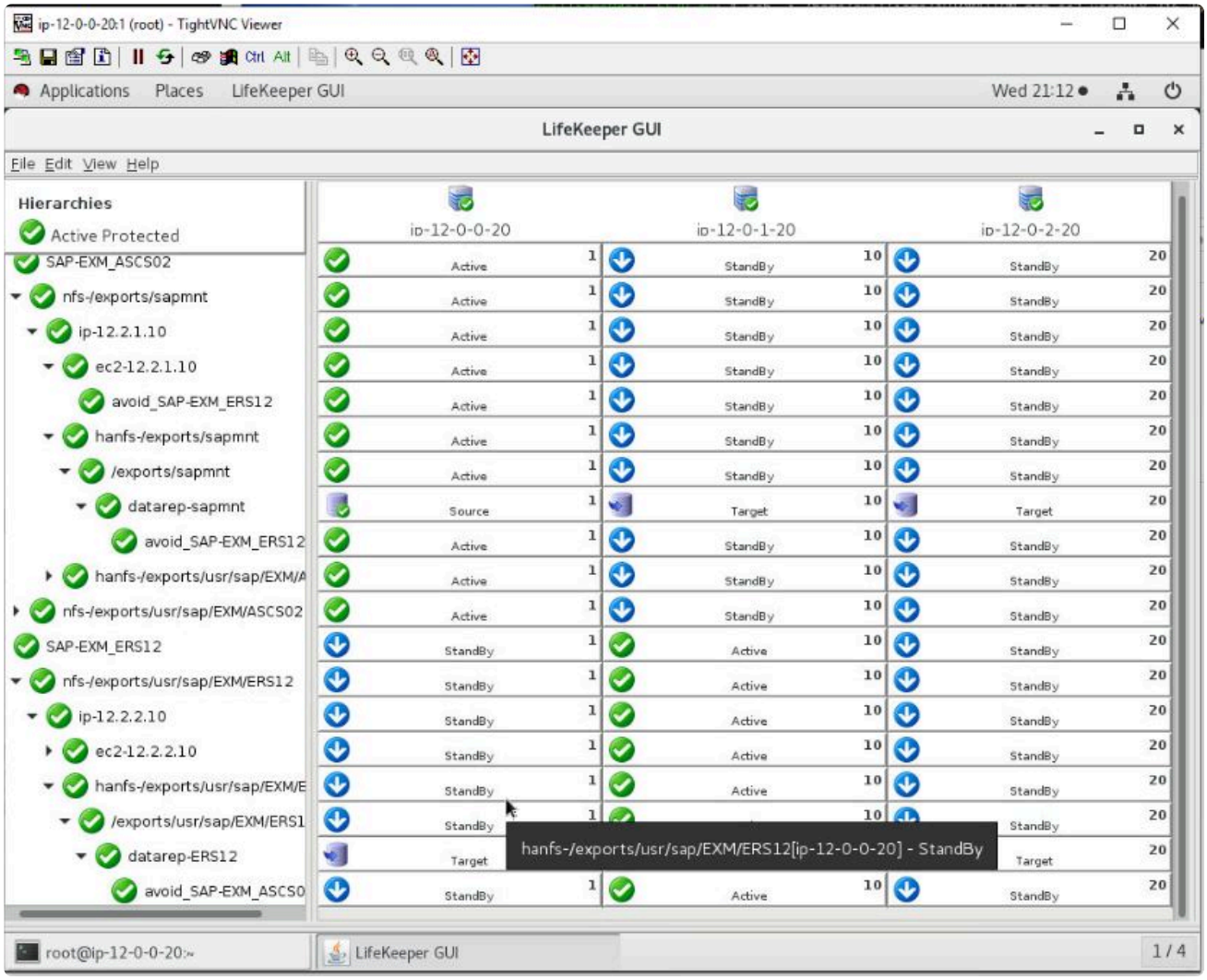



/opt/LifeKeeper/bin/create_terminal_leaf SAP-EXM_ASCS02 SAP-EXM_ERS12を実行することにより、

適切なターミナルリーフノードが作成されます。

```
[root@ip-12-0-2-20 ~]# /opt/LifeKeeper/bin/create_terminal_leaf SAP-EXM_ASCS02 SAP-EXM_ERS12
Create avoidance terminal leaf resource for root hierarchy 'SAP-EXM_ASCS02' and 'SAP-EXM_ERS12'.
creapphier: WARNING No quickCheck script specified
creapphier: WARNING No local recovery script specified
BEGIN create of "avoid_SAP-EXM_ERS12"
creating resource "avoid_SAP-EXM_ERS12"
resource "avoid_SAP-EXM_ERS12" successfully created
restoring resource "avoid_SAP-EXM_ERS12"
BEGIN restore of "avoid_SAP-EXM_ERS12"
Attempting to avoid resource avoid_SAP-EXM_ASCS02. Since the resource was not found on ip-12-0-0-20, res
tore of avoid_SAP-EXM_ERS12 is successful on ip-12-0-0-20.
END successful restore of "avoid_SAP-EXM_ERS12"
resource "avoid_SAP-EXM_ERS12" restored
END successful create of "avoid_SAP-EXM_ERS12"
Extending resource instances for avoid_SAP-EXM_ERS12
Creating dependencies
Setting switchback type for hierarchy
Creating equivalencies
LifeKeeper Admin Lock (avoid_SAP-EXM_ERS12) Released
Hierarchy successfully extended
BEGIN extend of "avoid_SAP-EXM_ERS12"
END successful extend of "avoid_SAP-EXM_ERS12"
Extending resource instances for avoid_SAP-EXM_ERS12
Creating dependencies
Creating equivalencies
LifeKeeper Admin Lock (avoid_SAP-EXM_ERS12) Released
Hierarchy successfully extended
BEGIN extend of "avoid_SAP-EXM_ERS12"
END successful extend of "avoid_SAP-EXM_ERS12"
Create dependency on 'ip-12-0-0-20' for 'avoid_SAP-EXM_ERS12' to 'ec2-12.2.1.10'.
Create dependency on 'ip-12-0-1-20' for 'avoid_SAP-EXM_ERS12' to 'ec2-12.2.1.10'.
Create dependency on 'ip-12-0-2-20' for 'avoid_SAP-EXM_ERS12' to 'ec2-12.2.1.10'.
Create dependency on 'ip-12-0-0-20' for 'avoid_SAP-EXM_ERS12' to 'datarep-ASCS02'.
Create dependency on 'ip-12-0-1-20' for 'avoid_SAP-EXM_ERS12' to 'datarep-ASCS02'.
Create dependency on 'ip-12-0-2-20' for 'avoid_SAP-EXM_ERS12' to 'datarep-ASCS02'.
Create dependency on 'ip-12-0-0-20' for 'avoid_SAP-EXM_ERS12' to 'datarep-sapmnt'.
Create dependency on 'ip-12-0-1-20' for 'avoid_SAP-EXM_ERS12' to 'datarep-sapmnt'.
Create dependency on 'ip-12-0-2-20' for 'avoid_SAP-EXM_ERS12' to 'datarep-sapmnt'.
creapphier: WARNING No quickCheck script specified
creapphier: WARNING No local recovery script specified
BEGIN create of "avoid_SAP-EXM_ASCS02"
creating resource "avoid_SAP-EXM_ASCS02"
resource "avoid_SAP-EXM_ASCS02" successfully created
restoring resource "avoid_SAP-EXM_ASCS02"
BEGIN restore of "avoid_SAP-EXM_ASCS02"
Attempting to avoid resource avoid_SAP-EXM_ERS12. Since the resource is not ISP on ip-12-0-1-20, restore
 of avoid_SAP-EXM_ASCS02 is successful on ip-12-0-1-20.END successful restore of "avoid_SAP-EXM_ASCS02"
resource "avoid_SAP-EXM_ASCS02" restored
END successful create of "avoid_SAP-EXM_ASCS02"
Extending resource instances for avoid_SAP-EXM_ASCS02
Creating dependencies
Creating equivalencies
LifeKeeper Admin Lock (avoid_SAP-EXM_ASCS02) Released
Hierarchy successfully extended
BEGIN extend of "avoid_SAP-EXM_ASCS02"
END successful extend of "avoid_SAP-EXM_ASCS02"
Extending resource instances for avoid_SAP-EXM_ASCS02
Creating dependencies
Creating equivalencies
LifeKeeper Admin Lock (avoid_SAP-EXM_ASCS02) Released
Hierarchy successfully extended
BEGIN extend of "avoid_SAP-EXM_ASCS02"
END successful extend of "avoid_SAP-EXM_ASCS02"
Create dependency on 'ip-12-0-0-20' for 'avoid_SAP-EXM_ASCS02' to 'ec2-12.2.2.10'.
Create dependency on 'ip-12-0-1-20' for 'avoid_SAP-EXM_ASCS02' to 'ec2-12.2.2.10'.
Create dependency on 'ip-12-0-2-20' for 'avoid_SAP-EXM_ASCS02' to 'ec2-12.2.2.10'.
Create dependency on 'ip-12-0-0-20' for 'avoid_SAP-EXM_ASCS02' to 'datarep-ERS12'.
Create dependency on 'ip-12-0-1-20' for 'avoid_SAP-EXM_ASCS02' to 'datarep-ERS12'.
Create dependency on 'ip-12-0-2-20' for 'avoid_SAP-EXM_ASCS02' to 'datarep-ERS12'.
[root@ip-12-0-2-20 ~]#
```

ターミナルリーフノードがアタッチされました。




avoid_SAP-EXM_ASCS02リソース（SAP-EXM_ERS12階層内）がノードでIn-serviceであり、クラスター内に利用可能な別のノードがある場合、avoid_SAP-EXM_ERS12リソースはSAP-EXM_ASCS02階層がそのノード上でIn-serviceになることを許可しません。

次の場合、 ノードは利用できません。

1. ノードが応答していない場合。
2. LifeKeeperがそのノードで実行されていない場合。
3. ノードでローカルリカバリーに失敗した場合。失敗したかどうかは、/opt/LifeKeeper/bin/flg_list の出力から '!volatile!recover_fail_<tag>'フラグを確認することにより判断できます。

フラグを作成することにより、特定のシステムで回避リーフを無効にできます。

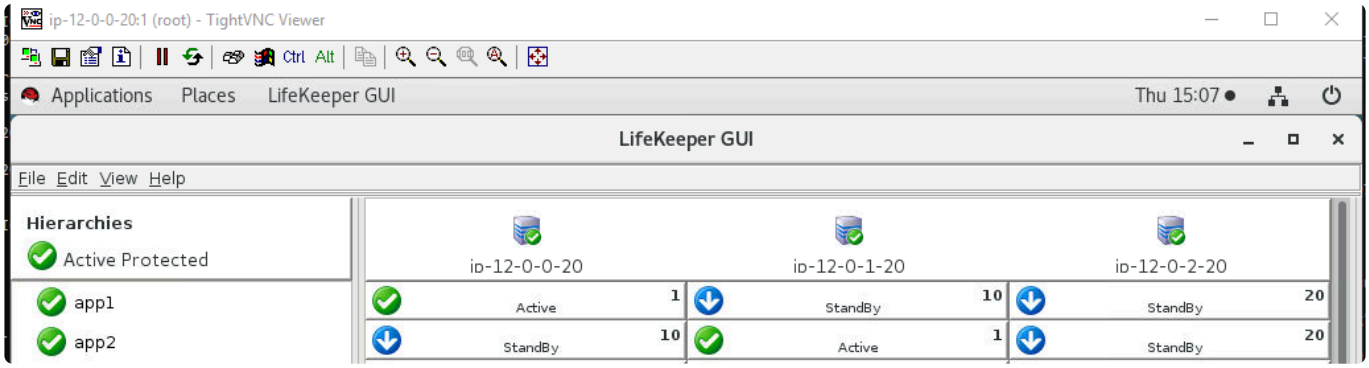
1. "ignore_avoidance_leaf" – リソースの回避リーフチェックを無効にします。つまり、回避リーフは常にIn-serviceになります。
2. "ignore_<tag>" – 特定の<tag>を無効にして常にIn-serviceになるようにしますが、他の回避リーフはIn-serviceを回避します。

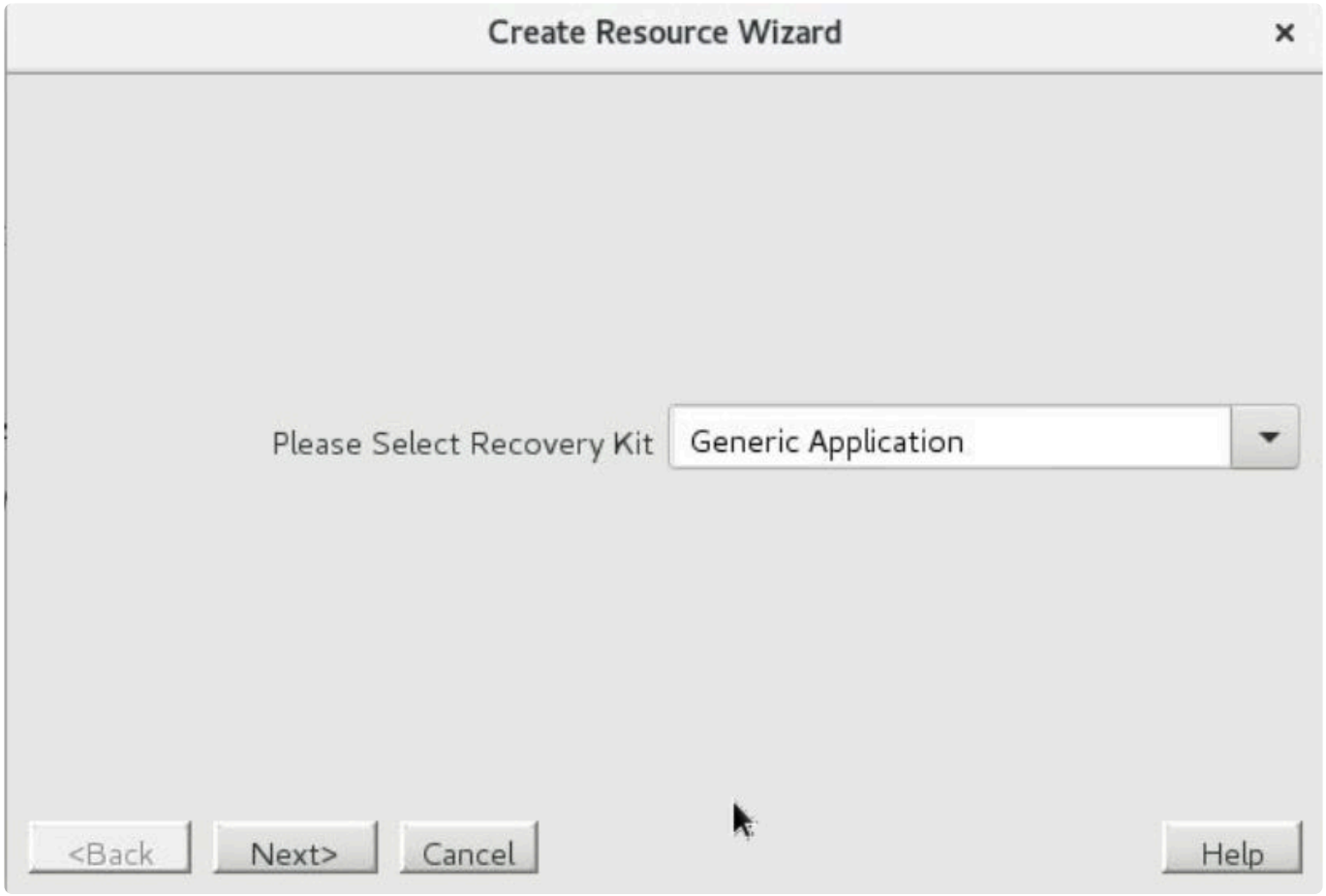
注記 ：回避リーフを無効にするための上記のフラグは、SAP quickCheckがASCSで実行されていることを検出した場合、SAP quickCheckがERSを移行することには影響しませ

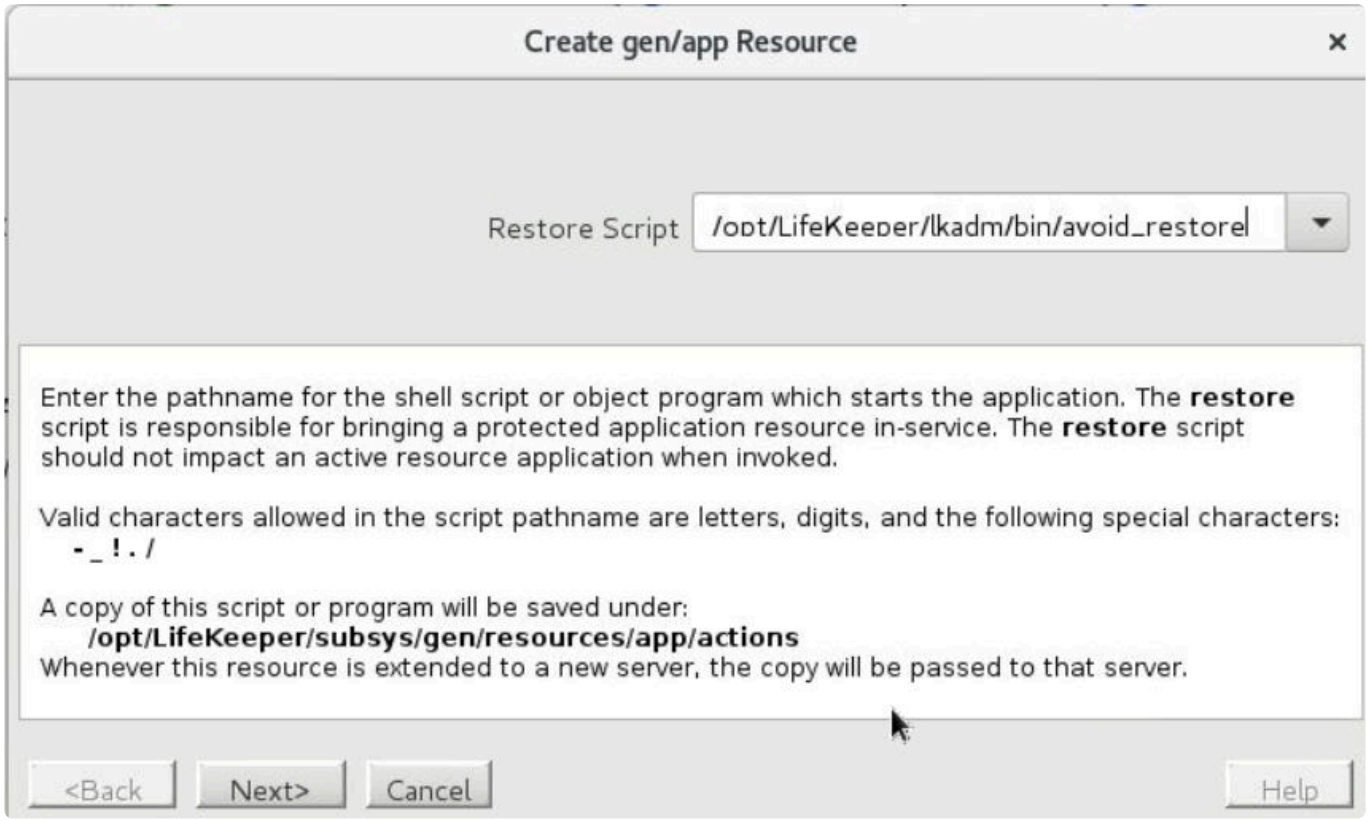
ん。 フラグ "sap_no_ers_relocation_<tag>" は、<tag>がERSリソースタグの場合、quickCheckがERSを再配置できないようにします。

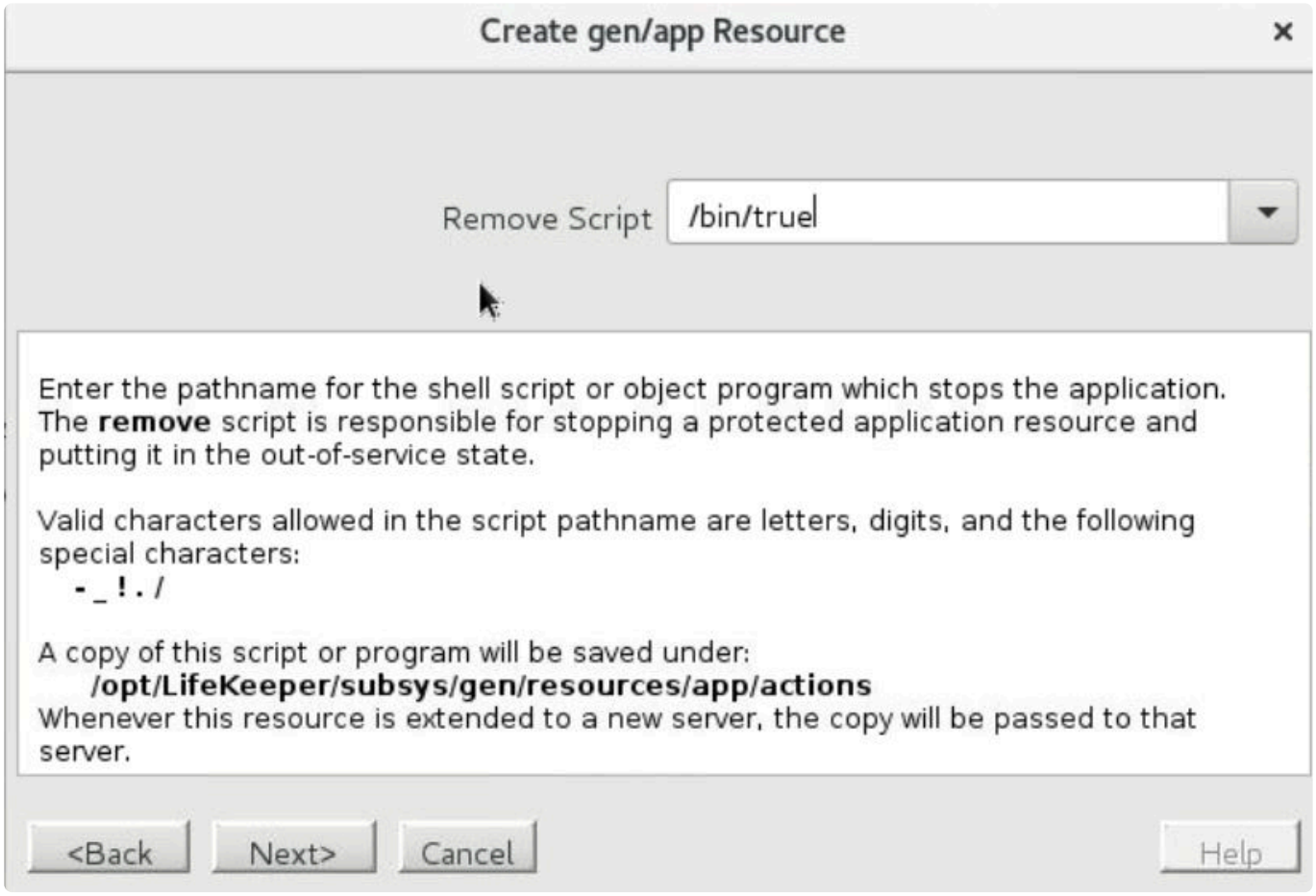
# GUIを使用して回避ターミナルリーフノードを作成する

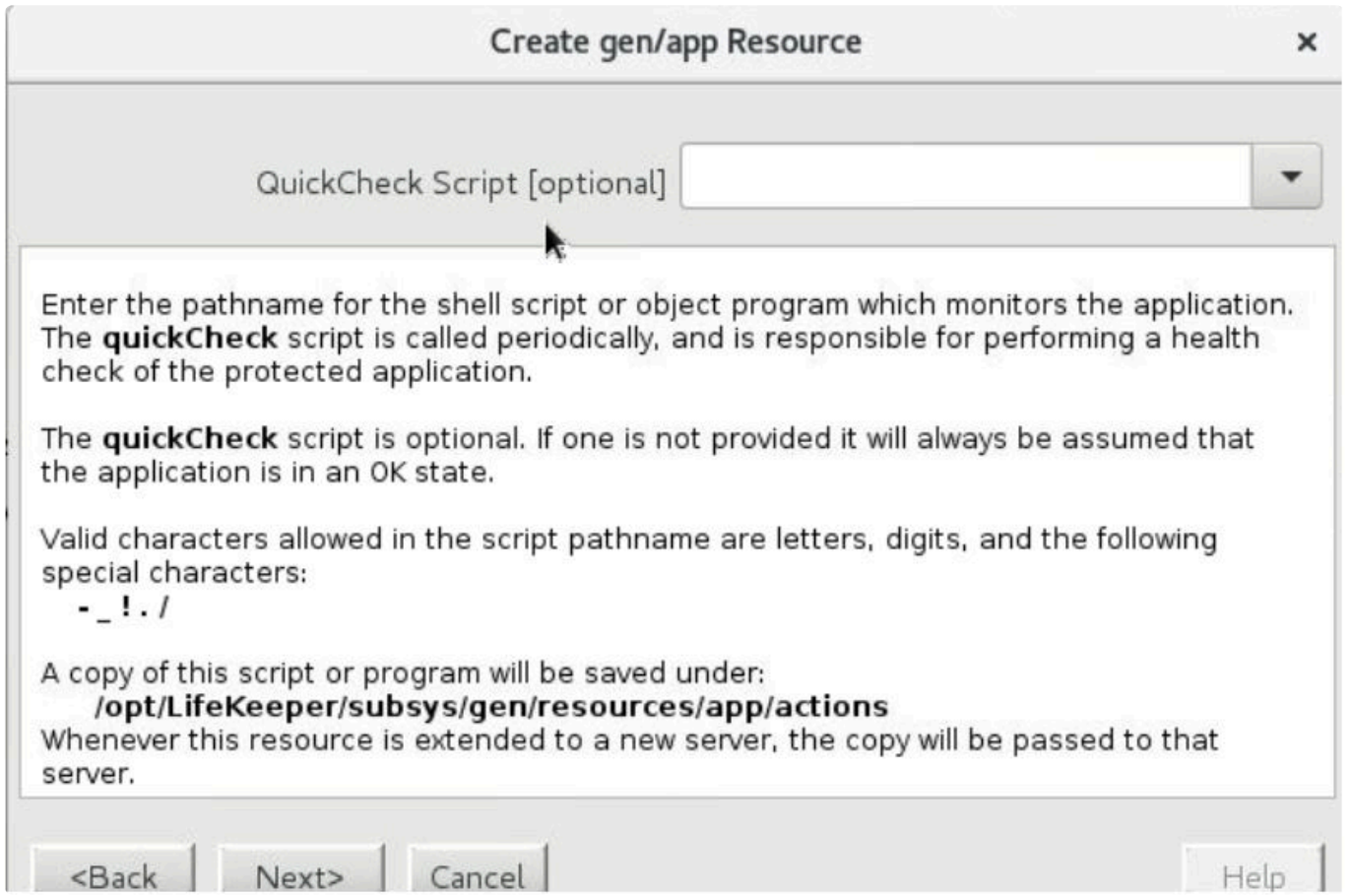
回避ターミナルリーフノードは、GUIを使用して作成できるgen/appリソースです。回避ターミナルリーフの復元スクリプトは「/opt/LifeKeeper/lkadm/bin/avoid_restore」で、削除スクリプトは「/bin/true」です。quickCheckスクリプトはありません。情報フィールドは、回避するタグの名前にする必要があります。たとえば、app1とapp2の2つのリソースがあり、可能であれば異なるノードに配置したい場合、「avoid_app1」と「avoid_app2」の2つのgen/appリソースを作成できます。avoid_app1の「情報」フィールドには「avoid_app2」が、avoid_app2の「情報」フィールドには「avoid_app1」が入ります。「avoid_app2」は「app1」の従属子リソースに、「avoid_app1」は「app2」の子リソースになります。
注記 ：タグ名は「avoid_<tag>」である必要はありませんが、これによりリソースが何をしているのかが明確になります。

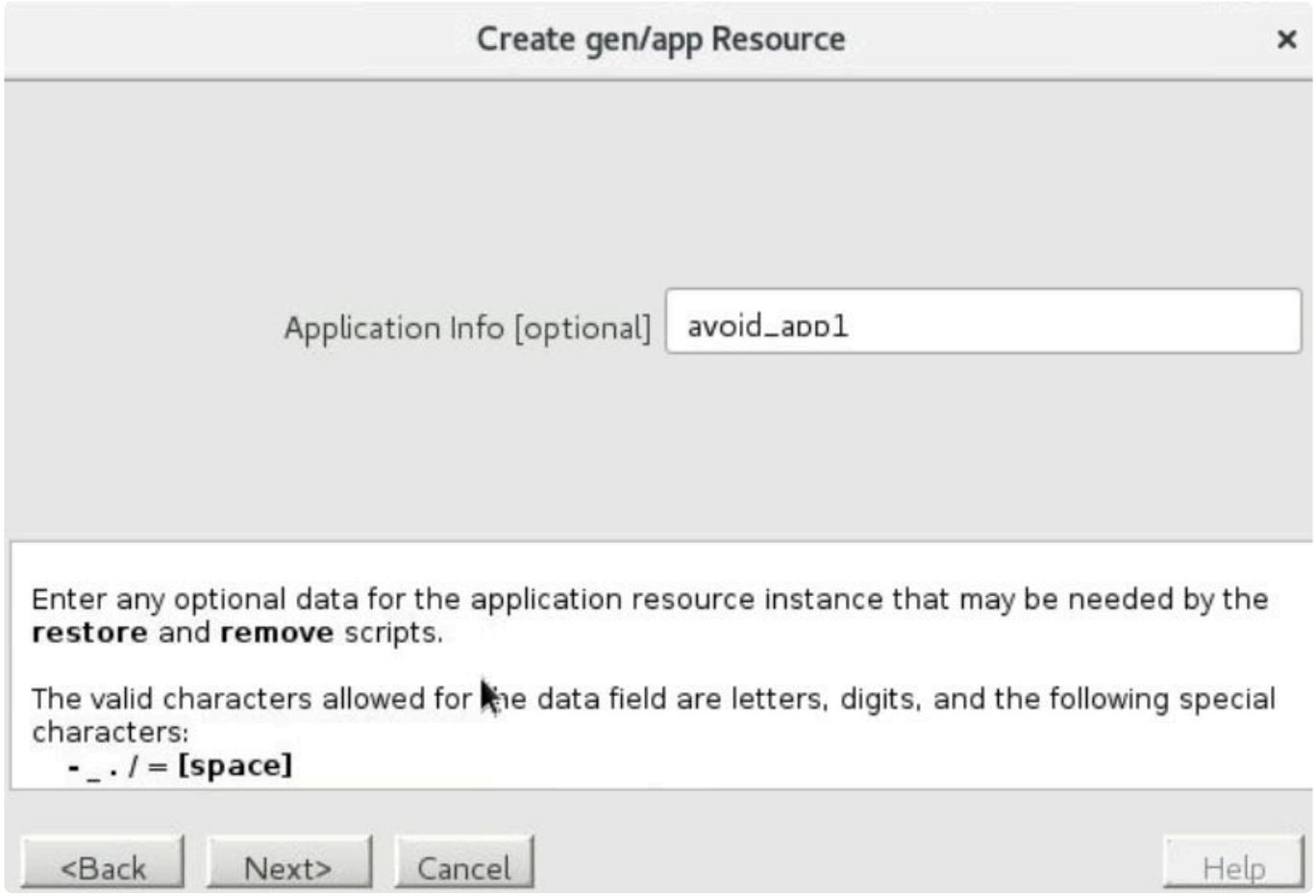



Create Resource Wizard
Please Select Recovery Kit
Generic Application
<Back
Next>
Cancel
Help


Create gen/app Resource
Restore Script
/opt/LifeKeeper/lkadm/bin/avoid_restore
Enter the pathname for the shell script or object program which starts the application. The restore script is responsible for bringing a protected application resource in-service. The restore script should not impact an active resource application when invoked.
Valid characters allowed in the script pathname are letters, digits, and the following special characters:
- _ ! . /
A copy of this script or program will be saved under:
/opt/LifeKeeper/subsys/gen/resources/app/actions
Whenever this resource is extended to a new server, the copy will be passed to that server.
<Back
Next>
Cancel
Help

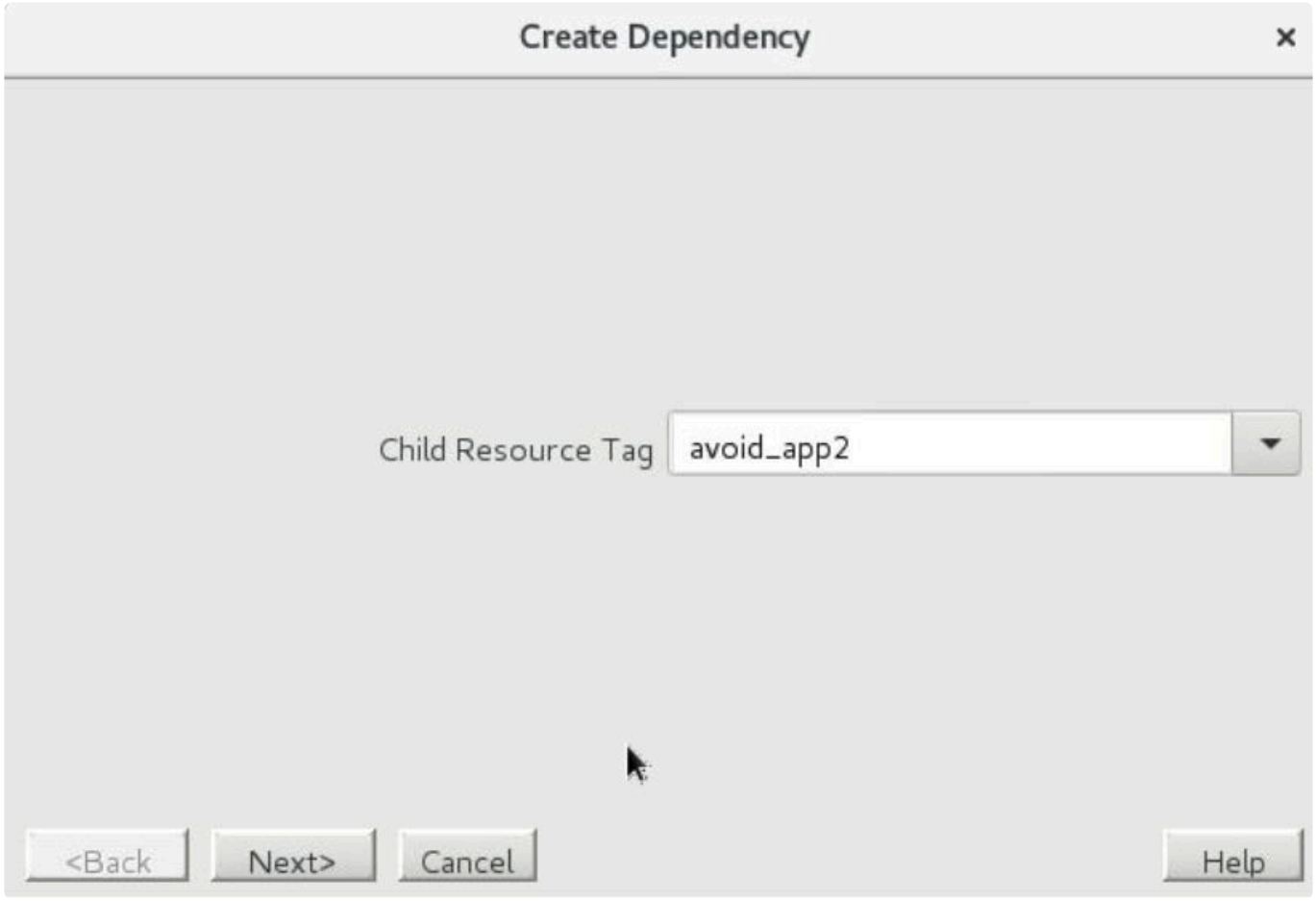

Create gen/app Resource
Remove Script
/bin/truel
Enter the pathname for the shell script or object program which stops the application. The **remove** script is responsible for stopping a protected application resource and putting it in the out-of-service state.
Valid characters allowed in the script pathname are letters, digits, and the following special characters:
- _ ! . /
A copy of this script or program will be saved under:
/opt/LifeKeeper/subsys/gen/resources/app/actions
Whenever this resource is extended to a new server, the copy will be passed to that server.
<Back
Next>
Cancel
Help


Create gen/app Resource
QuickCheck Script [optional]
Enter the pathname for the shell script or object program which monitors the application. The **quickCheck** script is called periodically, and is responsible for performing a health check of the protected application.
The **quickCheck** script is optional. If one is not provided it will always be assumed that the application is in an OK state.
Valid characters allowed in the script pathname are letters, digits, and the following special characters:
- _ ! . /
A copy of this script or program will be saved under:
/opt/LifeKeeper/subsys/gen/resources/app/actions
Whenever this resource is extended to a new server, the copy will be passed to that server.
<Back
Next>
Cancel
Help

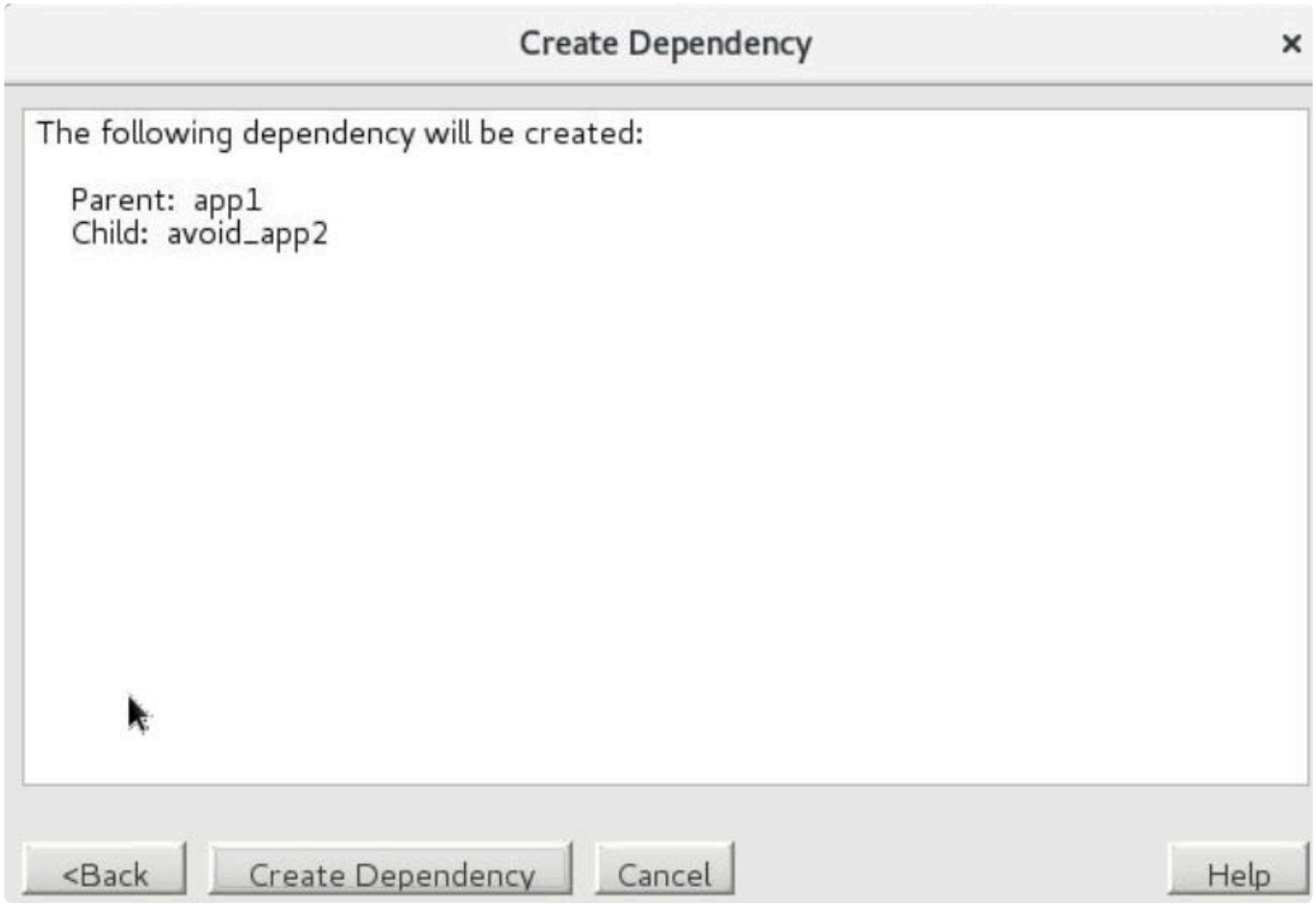


「avoid_app2」を作成したら、app1と同じ優先順位を持つすべてのノードに拡張します。

次に、「app1」リソースを選択し、「avoid_app2」との子依存関係を作成します。



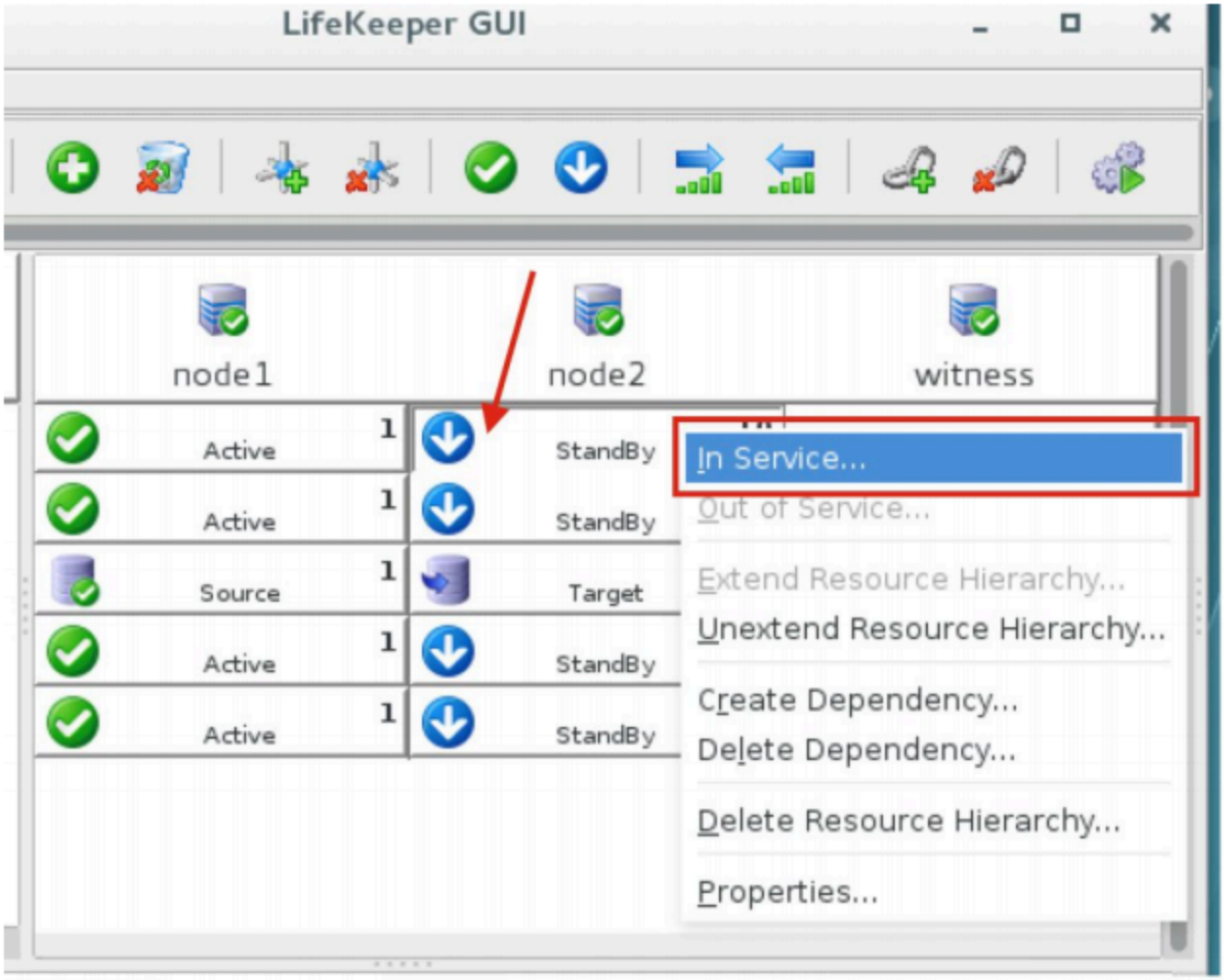

avoid_app1リソースを同様に作成すると、階層は次のようになります。


Create Dependency
The following dependency will be created:
Parent: app1
Child: avoid_app2
<Back
Create Dependency
Cancel
Help

# 12.3.4. スイッチオーバーおよびフェールオーバーのテスト

次の手順は、SAP の SIOS クラスターのスイッチオーバーとフェールオーバーをテストするための手順です。SAP 提供の Windows クライアントである「SAP Logon」または「SAP GUI for Windows」を開きます。プログラムは SAP ダウンロードサイトからダウンロードできます。仮想 IP アドレスは、**[Properties]** ページで「アプリケーションサーバー」として使用できます。これにより、仮想 IP が存在するプライマリマシンへの接続がアクティブになります。

1. LifeKeeper GUI を使用し、ノード 1 → ノード 2 にフェールオーバーします。ノード 2 の下にあるクラスターの一番上のリソースを右クリックし、[In Service ...] を選択します。これは、ノード 2 が障害発生時にノード 1 から引き継ぐことができることを示しています。




スイッチオーバーが完了したら、SAP GUI をチェックするか、必要に応じて再接続し、SAP がまだ正常に動作しているかどうかを確認します。

SAP プロセスが OS で実行されているかどうかを確認することもできます。

2. LifeKeeper GUI を使用し、ノード 2 → ノード 1 にフェールオーバーします。ノード 2 の下にあるクラスターの一番上のリソースを右クリックし、[In Service ...] を選択します。これは、ノード 1 が障

害発生時にノード 2 から引き継ぐことができることを示しています。

スイッチオーバーが完了したら、SAP GUI をチェックするか、必要に応じて再接続し、SAP がまだ正常に動作しているかどうかを確認します。

SAP プロセスが OS で実行されているかどうかを確認することもできます。

3. ノード1（アクティブノード）のコマンドラインインターフェースで、システム管理者から提供された halt を発行し、OS のクリーンシャットダウンを即座に実行します。

   フェールオーバーが完了したら、ノード 2 の LifeKeeper GUI を使用して、サービスが正常にフェールオーバーされたことを目視で確認します。

   SAP GUI をチェックするか、必要に応じて再接続し、SAP がまだ正常に動作しているかどうかを確認します。

   SAP プロセスが OS で実行されているかどうかを確認することもできます。

   ノード 1 を再びオンにし、ノード 2 の LifeKeeper GUI を使用して、ノード 1 のサービスがスタンバイになり、レプリケーションが開始されたことを目視で確認します。

   **注記：** スイッチオーバーまたはフェールオーバーのテストをさらに行う前に、データレプリケーションリソースがすでに同期を完了し、同期中であることを確認してください。

4. 必要に応じて手順 2 または手順 1 を繰り返し、ノード 1 にスイッチオーバーするか、ノード 2 でさらにクラッシュテストを実行します。

# 12.3.4.1. その他のリソース

## AWS サービス

- Amazon EC2
  https://aws.amazon.com/documentation/ec2/

- AWS CloudFormation
  https://aws.amazon.com/documentation/cloudformation/

- Amazon VPC
  https://aws.amazon.com/documentation/vpc/

## SIOS Protection Suite for Linux

- ステップバイステップ：共有ストレージを使用せずに Amazon EC2 で Linux フェールオーバークラスターを構成する方法
  http://www.linuxclustering.net/2016/03/21/step-by-step-how-to-configure-a-linux-failover-cluster-in-amazon-ec2-without-shared-storage-amazon-aws-sanless-cluster/

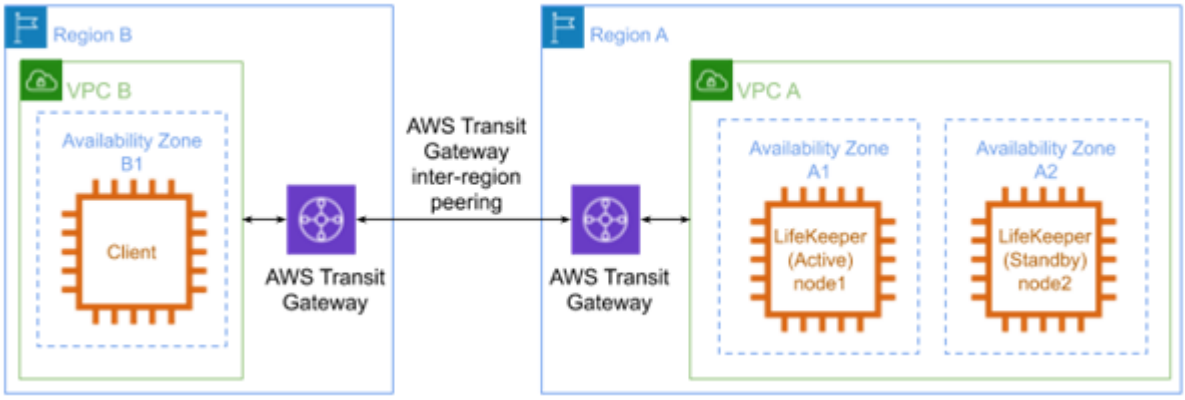
## クイックスタート

- AWS クイックスタート ホームページ
  https://aws.amazon.com/quickstart/

# 12.4. AWS Transit Gatewayを使用したLifeKeeperクラスターとクライアント間接続クイックスタートガイド

## 目的

AWS Transit Gateway、および AWS Transit Gateway inter-region peering のリリースにより、VPC（下図の VPC A）に配置された HA クラスターノードに、別リージョンの VPC（下図の VPC B）のクライアントから接続する構成で、Recovery Kit for EC2 のルートテーブルシナリオが利用可能になりました。




本資料は、LifeKeeper for Linux で、HAクラスターノードに、別リージョンの VPC のクライアントから接続する構成を構築するための要件や基本操作を解説するものです。

なお、本資料は LifeKeeper や Amazon Web Service (以下 AWS )の基本的な設定や操作、技術的な詳細情報を解説するものではありません。本構成の前提となる LifeKeeper や AWS に関する用語･操作･技術情報等につきましては、関連のマニュアルやユーザーサイト等であらかじめご確認ください。

> ✻ 注記: AWS Transit Gateway inter-region peeringは、本ドキュメント執筆時点（2020年2月）で米国東部(バージニア北部)、米国東部(オハイオ)、米国西部(オレゴン)、欧州(アイルランド)、欧州(フランクフルト)の各リージョンで利用可能です。その他のリージョンにサーバーまたはクライアントを配置する場合には本ドキュメントの構成は利用できません。AWS Transit Gateway inter-region peering が利用できないリージョンにサーバーまたはクライアントを配置する場合は、Route53 Recovery Kit を使用して、AWS の DNS サービスの「Route53」に登録されている DNS の A レコード（ホスト名に対応するIPアドレス）を更新する構成を検討してください。

> ✻ 注記: 本ドキュメントは、単一の VPC 内にクラスターノードを配置する構成を対象としています。クラスターノードを複数のリージョンや複数の VPC にまたがって配置する構成では、ルートテーブルシナリオは利用できません。

> ✻ 注記:「Amazon Web Services」、「Powered by Amazon Web Services」のロゴ、「AWS」、

「Amazon EC2」、「EC2」、「Amazon Elastic Compute Cloud」、「Amazon Virtual Private Cloud」、「AWS Direct Connect」、「AWS Identity and Access Management」、「AWS Transit Gateway」、「AWS Transit Gateway inter-region peering」および「Amazon VPC」は、米国その他の国における Amazon.com, Inc. またはその関連会社の商標です。

# 12.4.1. 利用のための要件

本構成を利用するためには、環境を準備する段階で満たすべきいくつかの要件があります。以下にAWS環境とその上に作成するインスタンスに関する要件をまとめます。

## AWS環境の要件

サービスを提供するための基盤となる環境をAWS上に作成します。本構成を利用するための要件は以下の通りです。

### Amazon Virtual Private Cloud (VPC)

- クラスターノードを配置する VPC を AWS 内に設定する必要があります。
- クライアントを配置する VPC は、クラスターノードを配置する VPC とは別リージョンに設定する必要があります。
- クラスターノードを配置する VPC に、プライマリ用インスタンスを配置するサブネットとスタンバイ用インスタンスを配置するサブネットを、それぞれ異なる Availability Zone(AZ) に作成する必要があります。
- クラスターノードを配置したVPC内のサブネットのセキュリティグループは、クライアントを配置したVPC内のサブネットからのインバウンドアクセスを許可するように設定する必要があります。

### Amazon Elastic Compute Cloud (EC2)

- クラスターノード用のインスタンスが2つ以上必要です。
- プライマリインスタンスとスタンバイインスタンスは、それぞれ異なるAZに構成する必要があります。
- クラスターノード用インスタンスは、Elastic Network Interface (ENI) に接続されます。
- クラスターノード用インスタンスは、LifeKeeper のインストール要件を満たす必要があります。
- AWS Command Line Interface (AWS CLI) を全てのクラスターノード用インスタンスにインストールする必要があります。インストール方法は、「AWS Command LineInterface のインストール 」を参照してください。AWS CLIの実行可能ファイルへのパスがLifeKeeperのデフォルトファイルである/etc/default/LifeKeeperのPATHパラメータに設定されていない場合、追加する必要があります。
- https を使用して Amazon EC2 Web サービスエンドポイントの URL ( https://docs.aws.amazon.com/ja_jp/general/latest/gr/rande.html#ec2_region ) に、および http を使用して Amazon EC2 メタデータの URL (http://169.254.169.254/) にアクセス可能でなければなりません。

### AWS Identity and Access Management (IAM)

LifeKeeper が AWS を操作するために、以下のアクセス権を持った IAM ユーザーもしくは IAM ロールが必要です。EC2 インスタンスの root ユーザーからアクセスできるように EC2 の IAM ロール を設定する

か、AWS CLI の設定を適切に行ってください。

- ec2:DisassociateAddress
- ec2:DescribeAddresses
- ec2:AssociateAddress
- ec2:DescribeRouteTables
- ec2:ReplaceRoute

### AWS Transit Gateway

- クラスターノードを配置した VPC と、クライアントを配置した VPC とは、VPC 同士を AWS Inter-Region VPC Peering で直接接続するのではなく、それぞれのリージョンで AWS Transit Gateway を作成し、AWS Transit Gateway 同士を AWS Transit Gateway inter-region peering で接続する必要があります。

- 各AWS Transit Gatewayを作成する際に、Default route table associationとDefault route table propagationを有効にします。

- 各リージョンにTransit Gateway Attachmentを作成して、各AWS Transit Gatewayを対応するVPCに接続します。

- Transit Gateway Attachmentを作成して、AWS Transit Gateway間のAWS Transit Gateway inter-region peering を有効にする必要があります。このステップでは、AWSによってTransit Gateway Attachmentが実際に作成される前に、対象リージョンで手動で確認する必要があることに注意してください。

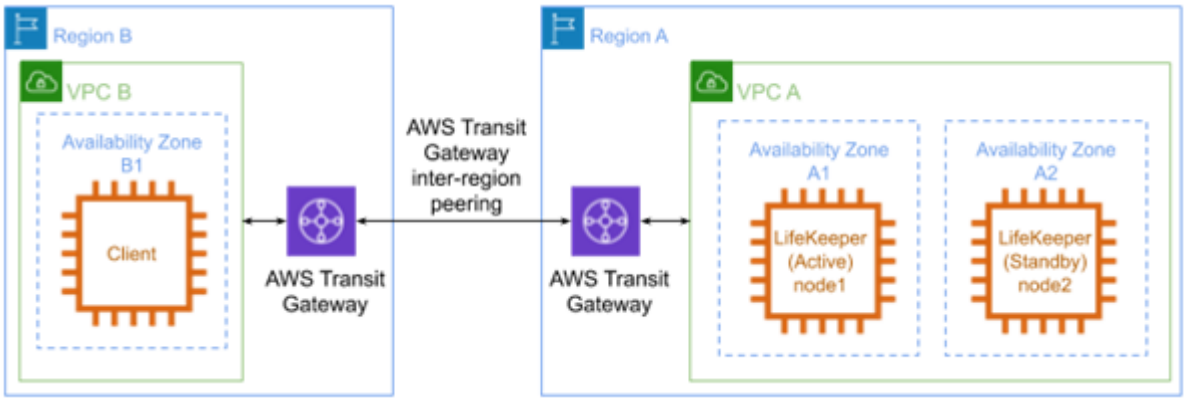
# 12.4.1.1. LifeKeeper ソフトウェアの要件

高可用性クラスターの各サーバーに、同じバージョンのLifeKeeperソフトウェアとパッチをインストールします。本構成で必要な Application Recovery Kit (ARK) は以下の通りです。具体的な LifeKeeper の要件については、 SPS for Linux テクニカルドキュメンテーション および SPS for Linux リリースノート を参照してください。

- LifeKeeper IP Recovery Kit
- LifeKeeper EC2 Recovery Kit

# 12.4.2. AWS VPC ピア接続構築手順

以下の構成図の様な環境を構築するための一般的な手順を解説します。




## 準備

「利用のための要件」を満たす環境を構築してください。それぞれのインスタンスに LifeKeeper をインストールして、 node1/node2 間にコミュニケーションパスを作成してください。

## IPリソース作成

ノード1で仮想IPリソースを作成し、ノード2に拡張します。IPリソースアドレスは、VPC Aによって管理されるCIDRブロック外にある必要があります。

## EC2 リソース作成

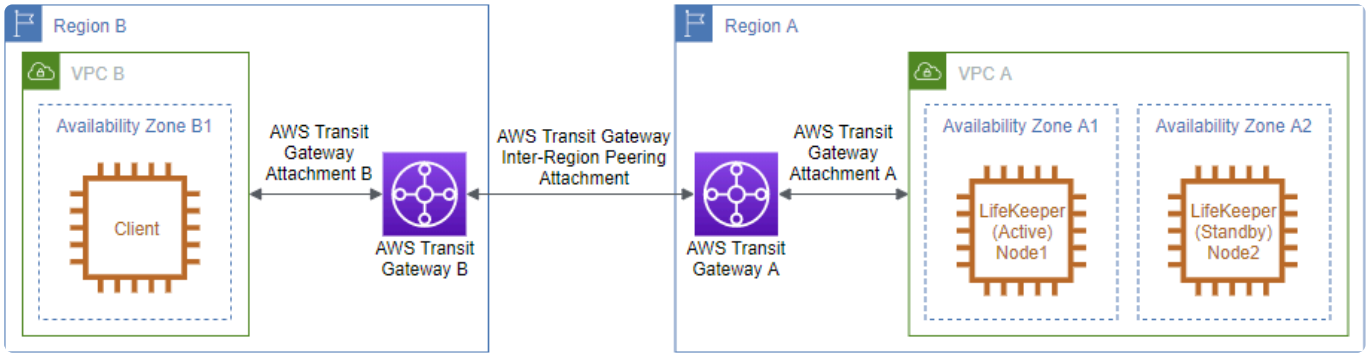
ノード1でEC2リソースを作成し、ノード2に拡張します。リソース作成時に要求される IP リソースは、上述の「IP リソース作成」で作成したリソースを指定してください。リソース作成時に要求される EC2 リソースタイプは Route Table (Backend Cluster) を指定してください。

## 保護するサービスのリソース作成

保護するサービスのためのリソースを作成してください。リソース作成に IP リソースが要求される場合は、上述の「IPリソース作成」で作成したリソースを指定してください。必要に応じて、サービス/アプリケーションが親リソース、EC2リソースが子リソースになるようにリソースの依存関係を設定して下さい。

# 12.4.3. ルートテーブルの設定

AWS環境は、以下の図に示すように構成する必要があります。




VPC BまたはクライアントインスタンスをSubネットのルートテーブルに次のルートを追加します。

| 送信先 | ターゲット |
|---|---|
| VPC A CIDRブロック | AWS Transit Gateway B |
| 仮想IPアドレス | AWS Transit Gateway B |

AWS Transit Gateway Bのルートテーブルに次のルートを追加します。

| CIDR | Attachment |
|---|---|
| VPC A CIDRブロック | AWS Transit Gateway Inter-Region Peering Attachment |
| 仮想 IP アドレス | AWS Transit Gateway Inter-Region Peering Attachment |

AWS Transit Gateway Aのルートテーブルに次のルートを追加します。

| CIDR | Attachment |
|---|---|
| VPC B CIDRブロック | AWS Transit Gateway Inter-Region Peering Attachment |
| 仮想 IP アドレス | AWS Transit Gateway Attachment A |

VPC AまたはLifeKeeperインスタンスを含むサブネットのルートテーブルに次のルートを追加します。

| 送信先 | ターゲット |
|---|---|
| VPC B CIDRブロック | AWS Transit Gateway A |

設定が完了したら、クライアントが高可用性クラスター内の各サーバーのプライベートアドレスと仮想IPアドレスにアクセスできることを確認してください。

# 12.4.4. 本構成における設定および運用上の留意点

LifeKeeper I/O フェンシングの利用を検討してください

# 12.4.4.1. LifeKeeper I/O フェンシングの利用を検討してください

AWS 環境では共有ディスク環境が構成できないため、スプリットブレイン問題を防ぐために SCSI リザーブ方式を採ることができません。
これを踏まえて、本構成ではより安全に運用できるように LifeKeeper の I/O フェンシング機能の Quorum/Witness Server もしくは STONITH の利用をご検討ください。
特に、Quorum モードの TCP_REMOTE 設定を利用すれば、別途 Quorum サーバーを立てずに I/O フェンシング機能を実装できるため、クラウド環境においては利用しやすいと考えられます。

利用方法については以下のURLをご確認ください。

- Quorum/Witness
- STONITH

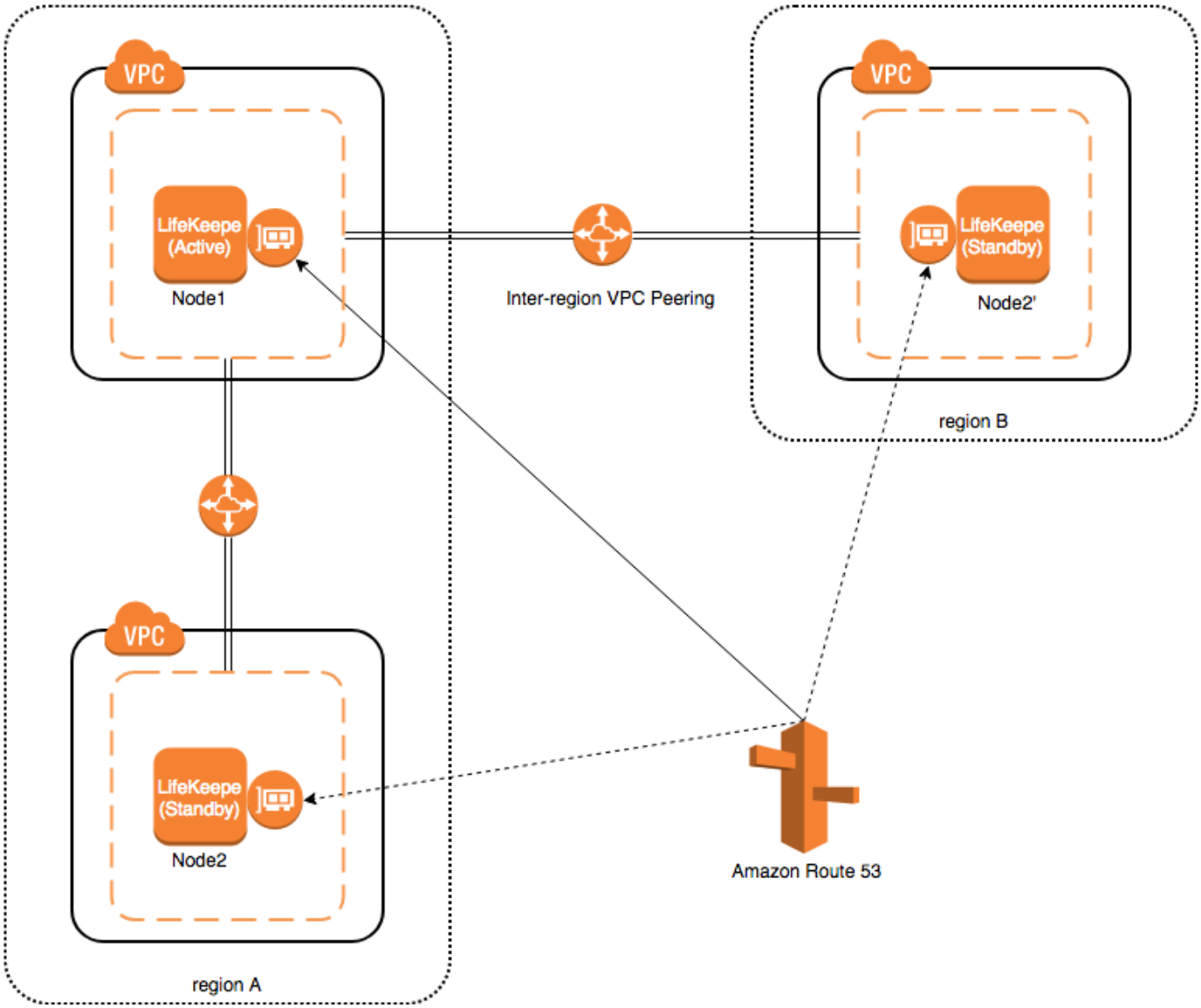
# 12.4.5. AWS Direct Connect 既知の問題とトラブルシューティング

現時点での情報はありません。

# 12.5. AWS VPC ピアリング接続を使用した複数 VPC クラスター構成クイックスタートガイド

## 目的

LifeKeeper for Linux v9.2.2 より、VPC ピアリング接続を使用して、クラスターノードを別々の VPC に配置した構成（以下、複数VPCクラスター構成）がサポートされました。リージョン間 VPC ピアリング接続を使用することで、リージョンを跨いでフェイルオーバーする構成（以下、複数リージョンクラスター構成）も取ることができます。




本資料は、LifeKeeper for Linux v9.5.0 で複数 VPC クラスター構成を構築するための要件や基本操作を解説するものです。

Recovery Kit for EC2™ は全てのクラスターノードが単一の VPC 内に配置されていることを前提としているため、複数 VPC クラスター構成では使えません。上記の図にある通り Recovery Kit for Route 53™ を

利用してクライアントからの接続先を切り替えてください。

なお、本資料は LifeKeeper や AWS の基本的な設定や操作、技術的な詳細情報を解説するものではありません。本構成の前提となる LifeKeeper や AWS に関する用語・操作・技術情報等につきましては、関連のマニュアルやユーザーサイト等であらかじめご確認ください。

注意：「Amazon Web Services」、「Powered by Amazon Web Services」のロゴ、「AWS」、「Amazon EC2」、「EC2」、「Amazon Elastic Compute Cloud」、「Amazon Virtual Private Cloud」、「Amazon Route 53」および「Amazon VPC」は、米国その他の国における Amazon.com, Inc. またはその関連会社の商標です。

# 12.5.1. 利用のための必要要件

本構成を利用するためには、環境を準備する段階で満たすべきいくつかの要件があります。以下に AWS 環境とその上に作成するインスタンスに関する要件をまとめます。

## AWS 環境の要件

サービスを提供するための基盤となる環境を AWS 上に作成します。本構成を利用するための要件は以下の通りです。

### Amazon Virtual Private Cloud (VPC)

- ２つ以上の VPC を作成する必要があります。
  VPC は同じリージョンでも、異なるリージョンに作成しても問題ありません。
- VPC ピアリング接続を使用して、VPC 間を接続する必要があります。
  ３つ以上の VPC を作成した場合は、それぞれの VPC 同士を VPC ピアリングで接続するか、または各 VPC を AWS Transit Gateway を経由してスター型に接続してください。

### Amazon Elastic Compute Cloud (EC2)

- 各 VPC に１つ以上の EC2 インスタンスが必要です。
- EC2 インスタンスは、Elastic Network Interface (ENI)に接続されます。
- EC2 インスタンスは、LifeKeeper のインストール要件を満たす必要があります。
- AWS Command Line Interface (AWS CLI) を全ての EC2 インスタンスにインストールする必要があります。インストール方法は、「 AWS CLI の最新バージョンをインストールまたは更新します。 」を参照してください。
- Amazon Route 53 サービスのエンドポイント route53.amazonaws.com にプロトコル HTTPS を使用してアクセスできる必要があります。EC2 および OS の設定を適切に行ってください。

### AWS Identity and Access Management (IAM)

LifeKeeper が AWS を操作するために、以下のアクセス権を持った IAM ユーザーもしくは IAM ロールが必要です。EC2 インスタンスの root ユーザーからアクセスできるように Amazon EC2 の IAM ロール を設定するか、 AWS CLI の設定 を適切に行ってください。

- route53:GetChange
- route53:ListHostedZones
- route53:ChangeResourceRecordSets
- route53:ListResourceRecordSets

### Amazon Route 53

- Amazon Route 53 にドメイン名を登録しサービスを利用できるようにする必要があります。これは Route53 リソース作成時に必要となります。

# 12.5.1.1. LifeKeeper ソフトウェアの要件

各サーバーに同じバージョンの LifeKeeper ソフトウェアとパッチをインストールする必要があります。本構成で必要な Application Recovery Kit は以下の通りです。具体的な LifeKeeper の要件については、LifeKeeper for Linux テクニカルドキュメンテーション および LifeKeeper for Linux リリースノート を参照してください。

- LifeKeeper IP Recovery Kit
- LifeKeeper Recovery Kit for Route 53

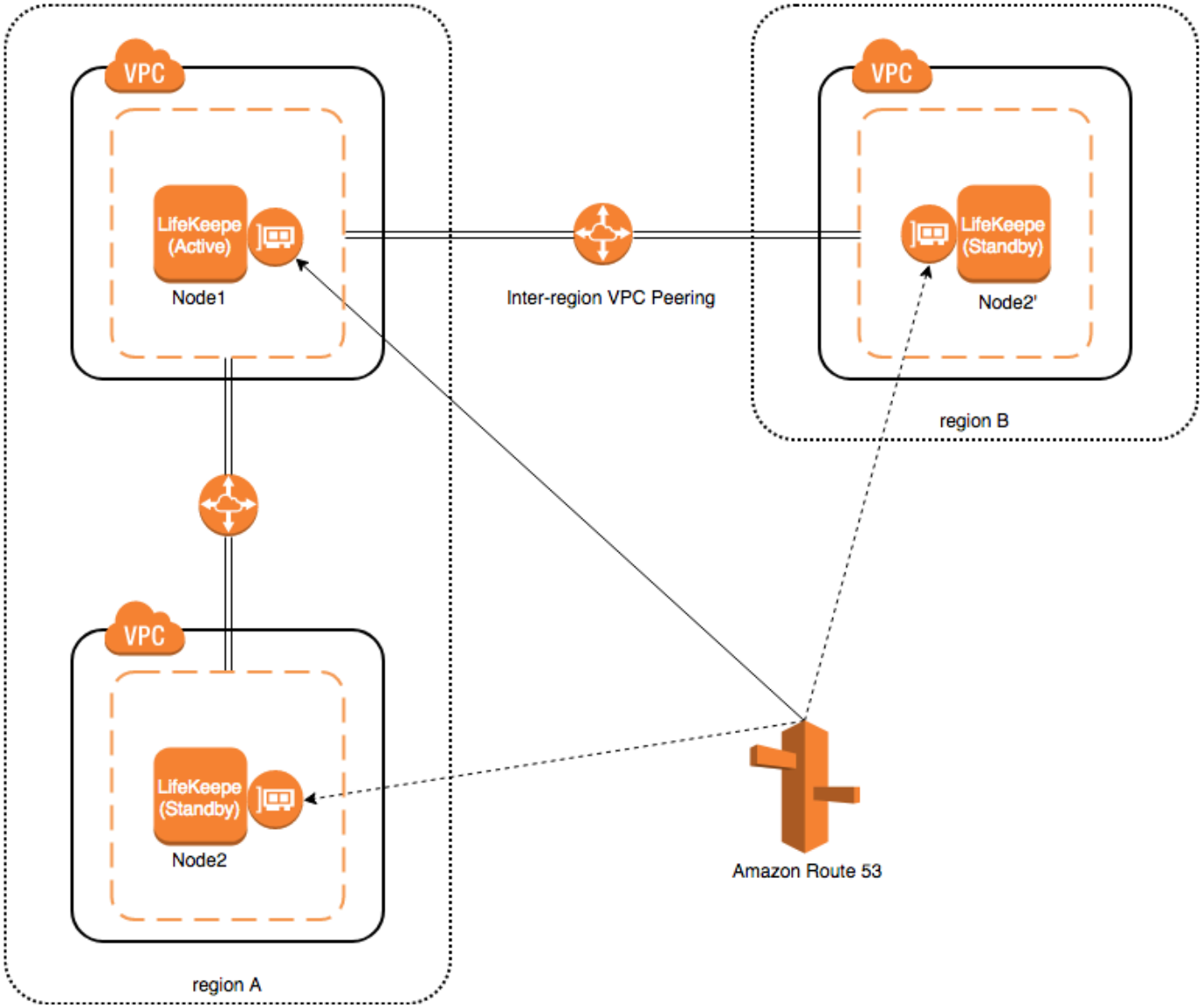
# 12.5.1.2. その他

本構成で別 VPC からサービスを利用するための要件は以下の通りです。

- サービスを利用するクライアントは、Route53 リソースで保護されるホスト名を名前解決できること。
- サービスを利用するクライアントは、Route53 リソースで保護されるホスト名でアクセスすること。

# 12.5.2. 構築手順

以下の構成図の様な環境を構築するための一般的な手順を解説します。




## 準備

前述の 利用のための必要要件 を満たす環境を構築します。それぞれのインスタンスに LifeKeeper をインストールして、Node1 / Node2 (あるいは Node2') 間にコミュニケーションパスを作成してください。
別 VPC 環境から Node1 / Node2 (あるいは Node2') に接続された ENI の実 IP アドレスにアクセスできることを確認してください。

## IP リソース作成

IP リソースを作成します。ここでは、仮想 IP リソースではなく、実 IP リソース（ネットワークインターフェース用に構成されたプライマリー IP アドレスのためのリソース）を作成します。リソース作成時に、IP アドレス “0.0.0.0” を指定してください。また、拡張時も IP アドレス “0.0.0.0” を指定してください。
詳しくは IP Recovery Kit 管理ガイド を参照してください。

# Route53 リソース作成

Route53 リソースを作成します。リソース作成時に要求される IP リソースは、前述の「IP リソース作成」で作成したリソースを指定してください。

## 保護するサービスのリソース作成

保護するサービスのためのリソースを作成してください。リソース作成に IP リソースが要求される場合は、前述の「IP リソース作成」で作成したリソースを指定してください。
親リソースが保護するサービスのリソース、子リソースが Route53 リソースとなるようにリソースの依存関係を設定してください。

# 12.5.3. 関連する LifeKeeper リソースについて

## Route53 リソース

### 動作概要

スイッチオーバーが発生すると、サービスへの接続を継続して確保するために Amazon Route 53 DNS 情報の更新が必要となります。この機能は、Route53 リソースで提供されています。

## IP リソース

### 動作概要

IP リソースとは、LifeKeeper Core 製品に含まれる IP リカバリキットを使用して生成したリソースです。本構成をサポートするために、実 IP アドレスで IP リソース（実 IP リソース）が生成できるようになりました。これにより、実 IP アドレスをLifeKeeper のリソースとして扱うことができます。

さらに詳細な情報につきましては、IP Recovery Kit テクニカルドキュメンテーション をご覧ください。

# 12.5.4. 本構成における設定および運用上の留意点

VPC ピアリングおよび AWS Route53 を使用したAWS クラスターへの接続の際の追加の留意点については以下の項目を参照してください。

- Quorum/Witness もしくは STONITH の利用を検討してください
- 利用のための必要要件

# 12.5.4.1. Quorum/Witness もしくは STONITH の利用を検討してください

AWS 環境では共有ディスク環境が構成できないため、スプリットブレイン問題を防ぐために SCSI リザーブ方式を採ることができません。また、IP リソースも各ノードで異なったアドレスを持った実 IP リソースを利用しますので、スプレットブレインが発生し得ます。

これを踏まえて、本構成ではより安全に運用できるように、Quorum/Witness もしくは STONITH の利用をご検討ください。

特に、Quorum モードの TCP_REMOTE 設定を利用すれば、別途 Quorum サーバーを立てずにスプリットブレインの発生を抑止できるため、クラウド環境においては利用しやすいと考えられます。

# 12.6. PostgreSQL Cluster with Shared Storage (ISCSI)

## Objective

This document is intended to aid you in installing, configuring and using the LifeKeeper for Linux evaluation product, to make PostgreSQL highly available. If PostgreSQL is not already installed, please allocate some time to install it on your servers. Once this task has been completed, you may install and configure LifeKeeper for Linux.

There are five phases in this process:

- Phase 1 – Prepare to Install
- Phase 2 – Configure Storage
- Phase 3 – Install and Configure PostgreSQL
- Phase 4 – Install LifeKeeper for Linux
- Phase 5 – Configure your LifeKeeper Cluster
- Phase 6 – Test Your Environment

# 12.6.1. Terms to Know – PostgreSQL

The following terms are used throughout this document and, while some may be familiar to you, it may be helpful to review how SIOS defines and uses these terms.

## Network Communication Terms

**Crossover cable** – A cable used to directly connect computing devices together, instead of being connected to a network switch, hub or router. This cable creates an isolated, private network to allow cluster-related and data replication traffic to flow between systems.

## Types of LifeKeeper Servers

**Server** – A computer system dedicated to running software application programs.

**Active Server** – This is the server where the resource hierarchy is currently running (IN SERVICE).

**Standby Server** – This is the server where the resource hierarchy is defined, but is not currently running. This server is available to bring the resource hierarchy into service should something happen to the resource hierarchy on the Active Server.

**Primary Server** – This is the server in a LifeKeeper configuration with the highest priority for a given resource hierarchy. It is the server that provides services for the resource hierarchy under normal circumstances.

**Secondary Server** – This is the server in a LifeKeeper configuration with the 2nd highest priority for a given resource hierarchy.

**Source Server** – In a LifeKeeper cluster, using data replication, this is the Active Server. It is where the resource hierarchy is currently running and the replicated partition (Source Partition) is accessible for writes.

**Target Server** – In a LifeKeeper cluster, using data replication, this is the Standby Server. The replicated partition (Target Partition) is updated with writes from the Source Partition by the SIOS Data Replication system. This partition should not be accessed/modified manually.

## SIOS Data Replication Terms

**Replication** – Transferring data from one partition to another via a sector-by-sector copy. During replication, the target partition should not be accessed or modified assuring your data integrity.

**Synchronous** – A replication scheme in which the data is confirmed written and valid on the target before the write operation occurs on the source disk through a series of information exchanges. Synchronous mirrors should only be implemented on high speed (100Mbps+) networks due to the network overhead involved.

**Asynchronous** – A replication scheme in which the data is released for writing on the source immediately and is sent to the target(s) simultaneously for writing as fast as the data can get there and can be written on them.

**Rate of Change** – A measure of the amount of data which is changing over a set period of time.

**Compression** – An algorithm which is optionally implemented to reduce the amount of traffic between source and target nodes. Nine levels of compression are offered. Compression is turned off by default.

**Throttling** – An optionally implemented mechanism to limit the bandwidth used for replication.

## LifeKeeper Product Terms

**Communications Path** – A mechanism supporting communication between nodes in a LifeKeeper cluster. SIOS highly recommends implementing multiple communication paths between all servers in the cluster to eliminate a single point of failure.

**Heartbeat** – A periodic message exchanged between nodes in a LifeKeeper cluster that provides server health monitoring. A heartbeat message is one type of inter-node cluster communication sent over a communications path.

**Split Brain** – A situation in which all communications paths between cluster members fail, but all servers remain up and running. In this situation, both systems believe the other has failed and both believe they should keep or bring resources into service.

**Failover** – The unplanned migration of a resource hierarchy to the Standby Server because of a system or resource failure on the Active Server.

**Switchover** – The planned migration of a resource hierarchy from the Active Server to the Standby Server.

**Switchback** – The setting that governs the recovery behavior of the server where the resource was in service when it failed. If the setting is “intelligent”, the server acts as a possible backup for the given resource. If the setting is “Automatic”, the server actively attempts to re-acquire the resource without further notice.

**Resource** – A system asset that can be protected by LifeKeeper. Resources can be used to represent disk partitions, virtual IP addresses, applications, etc.

**Extend a Resource** – Create or define an already configured LifeKeeper resource onto another server in the cluster and build an equivalency relationship that prevents the resource from coming in service on both systems simultaneously.

**Resource Hierarchy** – A grouping of resources, in a predetermined order, from high to low. This may also be referred to as simply a Hierarchy.

**Shared Storage** – One or more logical disk partitions that are physically attached to all nodes in a cluster. LifeKeeper ensures that the volume is only accessible by one server at a time. This is formally

called I/O fencing.

**Data Replication (Disk Mirroring)** – The replication of logical disk partitions to separate physical hard disks in real time to ensure continuous availability, currency and accuracy of data.

**Source** – The partition on the source server used for replication. The “gold” copy of the data.

**Target** – The partition on the target server used for replication.

**Switchable IP Address** – A unique IP address that may be moved between systems in the cluster. Client systems connect to this address and the system where the virtual IP resource is active will respond to requests.

# 12.6.2. The Evaluation Process – PostgreSQL

SIOS strongly recommends performing your evaluation of LifeKeeper for Linux in a test lab environment. SIOS is not responsible and cannot provide support for evaluation software installed in a production environment.

All questions during the evaluation period should be directed to evalsupport@us.sios.com or your local Pre-Sales Engineering contact. Pre-sales support will contact you by the next business day to answer questions. Once you are a licensed customer on software maintenance, you will have access to 24 × 7 post-sales technical support.

**Important** Your evaluation license is valid for a limited period of time from the day you receive the SIOS product evaluation package and licenses via email from the SIOS sales team.

# 12.6.3. Prepare to Install – PostgreSQL

## Hardware Requirements

### Primary and Secondary Servers

- Systems must meet the minimum requirements for the Linux distribution to be used during the evaluation.
- 512MB RAM minimum; 1GB RAM recommended.
- 2GB of available hard disk space recommended.
- Multiple Network Interface Cards (NIC's) are recommended.
- Configure one or more additional disk partitions to be used for data replication. On the primary server, these will become the source partitions. On the secondary server(s), these will become the target partitions. In this evaluation example we will be replicating MySQL data (which will be a partition mounted at /var/lib/mysql)
- For replicated partitions, a target partition's size must equal to or larger than the size of its source partition.
- The system ( / ) and boot (/boot) partitions are not eligible for replication.

**Note:** You may use more than one partition for replicated data, allowing for separation of multiple database files and/or log directories for performance reasons.

### Client

This system is not required but is recommended for testing the cluster environment.

- A standard linux terminal running the MySQL client can be used to test the configuration.

## Software Requirements

### Primary Server and Secondary Server

- Linux Distribution x86_64, AMD 64:
  - RedHat Enterprise Linux 5 (5.4+ recommended) or 6.x
  - CentOS Linux 5 (5.4+ recommended) or 6.x
  - Oracle Enterprise Linux 5 (5.4+ recommended), 6.3, 6.4
    - RedHat Compatibility Kernel Only
  - SuSE Linux Enterprise Server 10 or 11 (11 recommended)
  - See Linux Release Notes for a full list of supported Operating Systems
- Current patches / security updates are recommended.
- Satisfied dependencies; especially if the Linux installation package selection was base/minimal you will need to refer to the dependencies documentation at Linux Dependencies

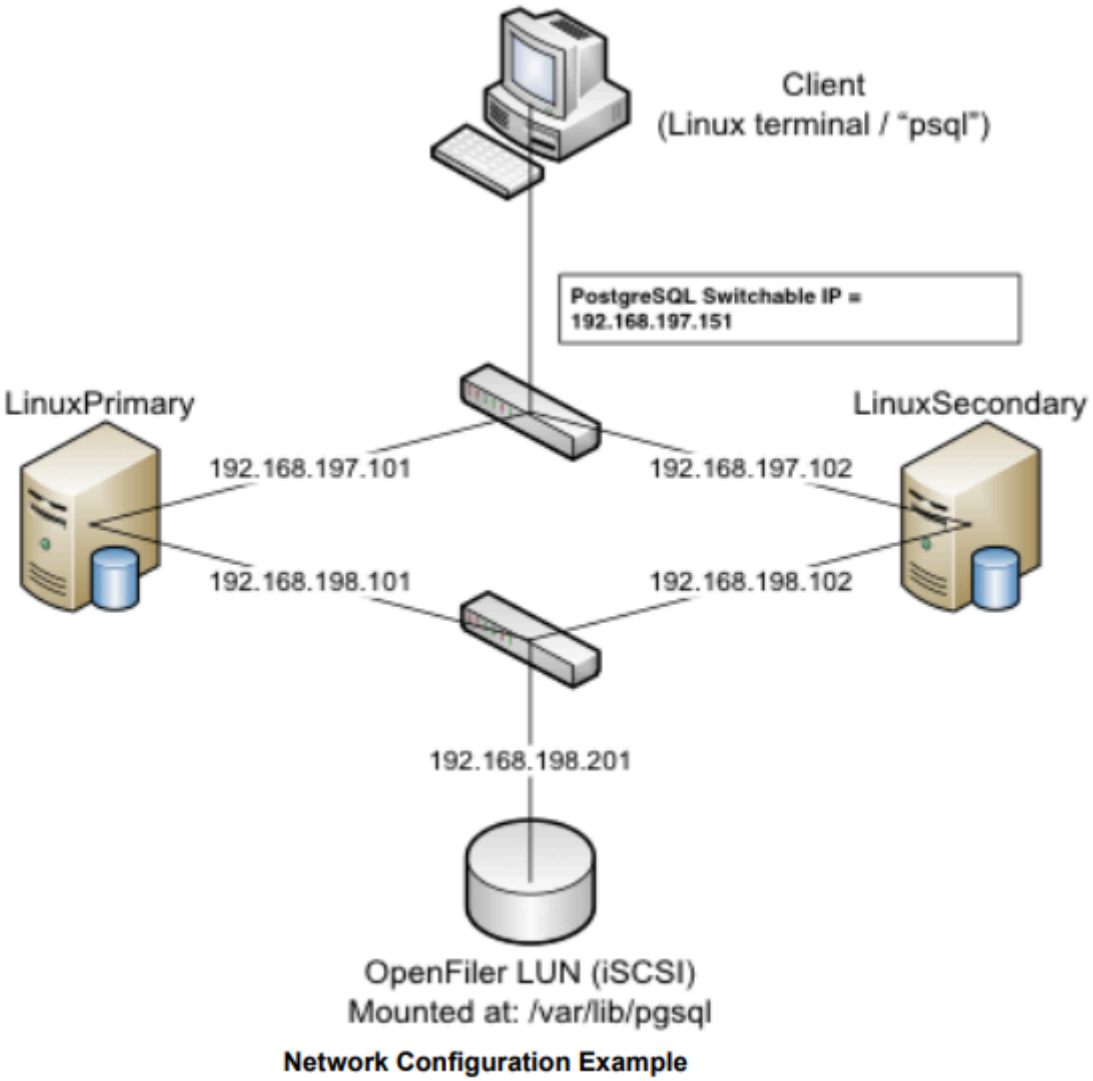
- Its recommended that IPtables is disabled
    - # /etc/init.d/iptables off
    - # chkconfig iptables off
    - See here for information regarding the ports LifeKeeper for Linux uses.
- Disable SELinux :
    - Edit /etc/selinux/config
    - Set SELINUX=disabled (note: permissive mode is also acceptable)
- Check the configuration of your /etc/hosts file
    - localhost.localdomain and localhost are the only entries that can be on 127.0.0.1
    - Create a separate entry for your hostname with a static address
- GUI Authentication with PAM
    - LifeKeeper for Linux now leverages the Pluggable Authentication Module (PAM) provided in the Linux Standard Base (LSB).
    - Users are identified and authenticated against the system's PAM configuration. Privilege levels are determined from group membership as provided through PAM.
    - In order to access the GUI, a user must be a member in one of the three LifeKeeper groups: **lkadmin**, **lkoper** or **lkguest**.
    - See Configuring GUI Users for more information.

# Network Requirements

For your evaluation, we recommend configuring your machines similarly to the following example.

LinuxPrimary and LinuxSecondary are multi- in production environments to avoid a single point of failure.

In this example evaluation scenario we will be leveraging Shared (iSCSI) Storage with our PostgreSQL database. OpenFiler is a storage appliance server that will serve an iSCSI target to LinuxPrimary and LinuxSecondary.




**Network Configuration Example**

## Primary Server and Secondary Servers

- Configure the Host file with entries for all LifeKeeper protected servers. This is typically /etc/hosts.

  Example:

  192.168.197.101 LinuxPrimary

  192.168.197.102 LinuxSecondary

- See your Network Administrator to obtain an unused IP Address to be used as the switchable IP Address. This switchable IP Address will be created later in the configuration process.
- Public Network connection(s) configured with:
  - Static IP address
  - Correct subnet mask
  - Correct gateway address

    - Correct DNS server address(es)

- Private Network connection(s) configured with:
    - Static IP address (on a different subnet from the public network)
    - Correct network mask
    - No gateway IP address
    - No DNS server addresses

## Client

Must be able to communicate on the same subnet/network as the servers Public interface addresses. In our example, this is the 192.168.197.0/24 network.

# 12.6.4. Configure Storage – PostgreSQL

## Before you Begin

Ensure the following:

- If planning to use replicated storage, have an extra volume/partition on both servers that can be used for data replication. A target volume's size must equal to or larger than the size of its source volume.
- If planning to use shared storage, as in this example, ensure the Shared storage is configured and accessible to your cluster nodes. This can either be Fiber Channel SAN, iSCSI, NAS, etc. In this example we will review configuration of an iSCSI target for use as our PostgreSQL database storage repository.

## Configure iSCSI initiator, discover and login to iSCSI target

This Evaluation guide will not cover how to setup an iSCSI Target Server. It is assumed that the shared storage already exists in your environment. If you don't have shared storage and wish to configure it, a simple solution is to use OpenFiler (http://www.openfiler.com/), an Open Source storage management appliance, which can be run on physical hardware or as a virtual machine.

On both Primary and Secondary servers, perform the following functions:

1. If not already installed, ensure that the **iscsi-initiator-utils** rpm package is installed:

   ```
   # yum install iscsi-initiator-utils
   ```

2. Start the iscsid service and enable it to automatically start when the system boots

   ```
   # service iscsid start
   ```

   ```
   # chkconfig iscsid on
   ```

3. Configure the iscsi service to automatically start, which logs into iSCSI targets needed at system start up.

   ```
   # chkconfig iscsi on
   ```

4. Use the iscsiadm command to discover all available targets on the network storage server (OpenFiler)

   ```
   # iscsiadm -m discovery -t sendtargets -p <name or IP of iSCSI server>
   ```

**Example**
[root@LinuxPrimary init.d]# iscsiadm -m discovery -t sendtargets -p 192.168.198.201
**iqn.2006-01.com.openfiler:tsn.postgres**

5. Manually Login to the iSCSI Target

   # iscsiadm -m node –T iqn.2006-01.com.openfiler:tsn.postgres –p 192.168.198.201 – login

6. Configure Automatic Login

   # iscsiadm -m node –T iqn.2006-01.com.openfiler:tsn.postgres –p 192.168.198.201 —op update -n node.startup -v automatic

7. Use the “gdisk” command to format your iSCSI LUN, if needed

   # gdisk /dev/sdc

8. Create a filesystem on your new iSCSI LUN Partition, sdc1

   # mkfs.ext3 /dev/sdc1

9. Mount your iSCSI LUN at /var/lib/pgsql (assuming a default postgres configuration). If data already exists in this directory, make sure to move it into the shared iSCSI LUN

   # mount mount /dev/sdc1 /var/lib/pgsql

10. At this point you now have an iSCSI shared LUN, /dev/sdc1, mounted at /var/lib/pgsql. Our disk layout now look as follows (example):

**Example**
[root@LinuxPrimary pgsql]# df
Filesystem 1K-blocks Used Available Use% Mounted on
/dev/sda2 25967432 3683016 1976400 66% /
/dev/sda1 101086 24659 71208 26% /boot
tmpfs 517552 0 517552 0% /dev/shm
**/dev/sdc1 966644 38944 878596 5% /var/lib/pgsql**

# 12.6.5. Install, Configure and Start PostgreSQL

## Primary Server

On your Primary server, perform the following actions:

1. Install both the “postgresql-server” and “postgresql” rpm packages if they do not exist on your system. Apply any required dependencies as well

   # yum install postgresql postgresql-server

2. Verify that your Shared iSCSI LUN is still mounted at /var/lib/pgsql via the “df” command

3. If this is a fresh PostgreSQL install, initialize a sample PostgreSQL database:

   # su – postgres

   # initdb —pgdata=/var/lib/pgsql/data

4. Ensure that all files in your PostgreSQL data directory (/var/lib/pgsql) have correct permissions and ownership

   # chown –R postgres:postgres /var/lib/pgsql

   # chmod 755 /var/lib/pgsql

5. Finally, manually start the PostgreSQL daemon from the command line. Note: **Do Not** start it via the “service” command, or the /etc/init.d/ scripts

   # su – postgres

   # pg_ctl start -D /var/lib/pgsql/data -l /var/lib/pgsql/pgstartup.log -o “-p 5432” -w

6. Verify PostgreSQL is running by connecting with the psql client (ensure you are still running as the “postgres” linux user):

   -bash-3.2$ psql

   Welcome to psql 8.1.22, the PostgreSQL interactive terminal.

   Type: \copyright for distribution terms

   \h for help with SQL commands

   \? for help with psql commands

```
\g or terminate with semicolon to execute query

\q to quit

postgres=# \q

-bash-3.2$
```

## Secondary Server

On your Secondary Server:

1. install both the “postgresql” and “postgresql-server” rpm packages if they do not exist on your system. Apply any required dependencies as well

   ```
   # yum install postgresql postgresql-server
   ```

2. Ensure that the PostgreSQL data directory (/var/lib/pgsql) has correct permissions and ownership

   ```
   # chown –R postgres:postgres /var/lib/pgsql

   # chmod 755 /var/lib/pgsql
   ```

3. There is no need to perform any of the additional steps taken on the Primary Server

# 12.6.6. Install LifeKeeper for Linux – PostgreSQL

For ease of installation, SIOS has provided the LifeKeeper for Linux with an installation script. Towards the end of the script, the desired Application Recovery Kits (ARKs) should be selected for installation. The software will be installed to the following locations:

| SPS for Linux Component | Install Location |
|---|---|
| LifeKeeper Software | /opt/LifeKeeper |
| LifeKeeper Config File | /etc/default/LifeKeeper |

Perform the following actions on **both** Primary and Secondary server.

## Download Software

1. Open the LifeKeeper evaluation email you received from SIOS.

2. Download the LifeKeeper for Linux Software from the link provided in your email. It is generally easiest to use “wget” to recursively download all files. Example:

   a. # cd /root

   b. # wget -r <URL>

   c. After successful download you will have downloaded contents similar to the follow directory listing:

      [root@LinuxPrimary ~]# ls -l <directory> total 63680

      -rw-r—r— 1 root root 23163 May 30 14:03 EULA.pdf

      -rw-r—r— 1 root root 536 May 30 14:03 readme.txt

      -rw-r—r— 1 root root 65179648 May 30 14:03 sps.img

3. Download your Evaluation license key from the link specified in your evaluation email. Save the license file to an easy to remember location on both servers.

## Run the LifeKeeper Installer Script

1. Loopback mount the sps.img file previously downloaded, which is an ISO9660 image file. Run the “setup” script inside:

   # mount -o loop sps.img /mnt

```
# cd /mnt

# ./setup
```

2. During this procedure, you will hit Enter in most cases to accept default values and continue to the next screen. Note the following exceptions:

   a. On the screen titled “High AvailabilityNFS” you may select “n” as in this particular eval guide we will not be creating a highly available NFS server cluster configuration.

   b. If you have plans to create a highly available NFS service, adjust your response accordingly.

3. Towards the end of the setup script, you can choose to install a trial license key now, or later. We will install the license key in the next step, so you can safely select “n” at this point

4. In the final screen of the “setup” select the DataKeeper from the list displayed on the screen.

5. The following RPMs should be be installed:

   a. steeleye-lkDR-<version>.noarch.rpm

6. Un-mount the Distribution Enabling disk image:

```
# cd /root

# umount /mnt
```

# Install the Evaluation License Keys

The last phase of the setup process installs the licensing keys. You must install the evaluation license key file (“.lic”) that you downloaded with your evaluation software before starting the LifeKeeper for Linux.

1. To install your trial license key, run the “lkkeyins” command on both Primary and Secondary Server. This command is located at /opt/LifeKeeper/bin/lkkeyins . Example:

```
# /opt/LifeKeeper/bin/lkkeyins <path_to_license/<filename>.lic
```

2. Validate your license keys were installed via the /opt/LifeKeeper/bin/lklicmgr command

```
# /opt/LifeKeeper/bin/lklicmgr

License File: 20101230.lic
```

| Product | Type | Expiry |
|---|---|---|
| LifeKeeper for Linux | Eval | 27 Mar 2013 (87 days) |

| | | |
|---|---|---|
| SIOS Data Replication ARK | Eval | 27 Mar 2013 (87 days) |
| PostgreSQL Recovery Kit | Eval | 27 Mar 2013 (87 days) |

# Start the LifeKeeper for Linux

1. Start:

```
# /opt/LifeKeeper/bin/lkstart
```

# 12.6.7. Configure the Cluster – PostgreSQL

## Primary Server

Complete the following steps on the primary server to configure the cluster:

- Create TCP Communication (Comm) Path(s)
- Verify the Communication (Comm) Path(s)

Before you begin, SIOS recommends at least two TCP communications paths between each server within the cluster to each remote server for heartbeat redundancy.

> **Important** Supported configurations require that you define redundant comm. paths, so that the failure of a single communication line will not cause a split brain where resource hierarchies may come in-service on multiple servers simultaneously.

### Access the LikeKeeper GUI

The LifeKeeper Graphical User Interface (GUI) is a Java based application that can be run as a native Linux application.

The LifeKeeper GUI is based on Java RMI with callbacks. Hostnames must be resolvable or you may receive a Java 115 or 116 error.

1. Verify that both short and fully qualified hostnames of all cluster nodes resolve to the proper locations

   # ping LinuxPrimary

   # ping LinuxPrimary.domain.com

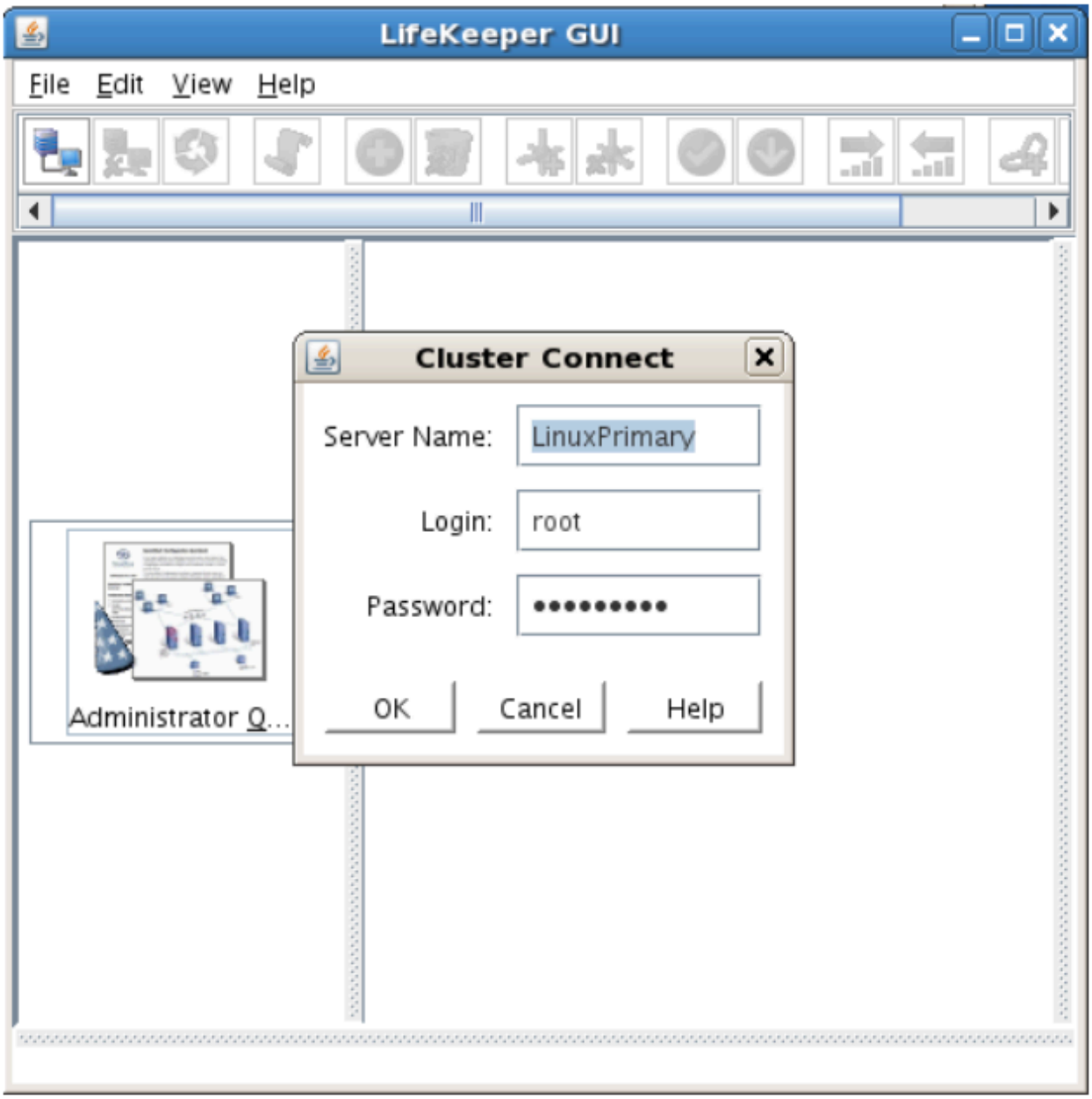
   # ping LinuxSecondary

   # ping LinuxSecondary.domain.com

2. To start the LifeKeeper Linux GUI Application:

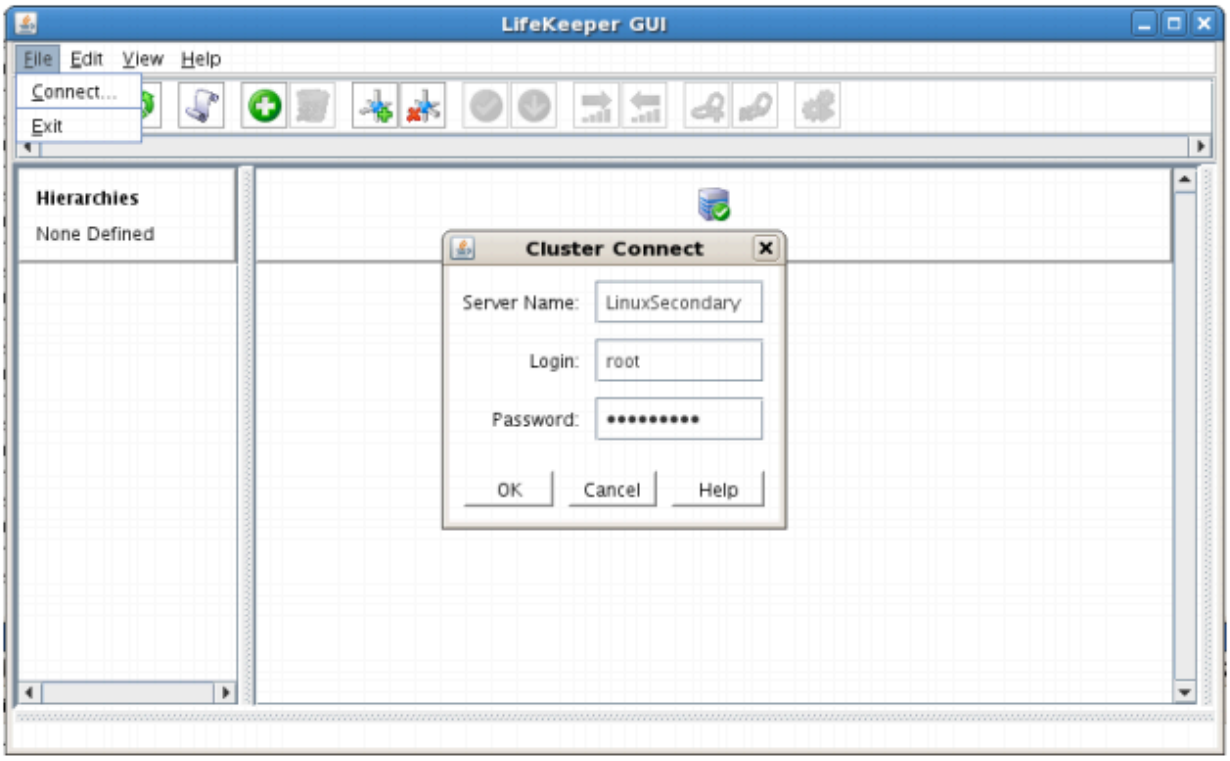
   a. /opt/LifeKeeper/bin/lkGUIapp &

3. Enter the name of the server you wish to connect to (this field will be populated with the name of the server you are on, if you are running the GUI from a server with LifeKeeper installed) along with your root credentials and click OK.



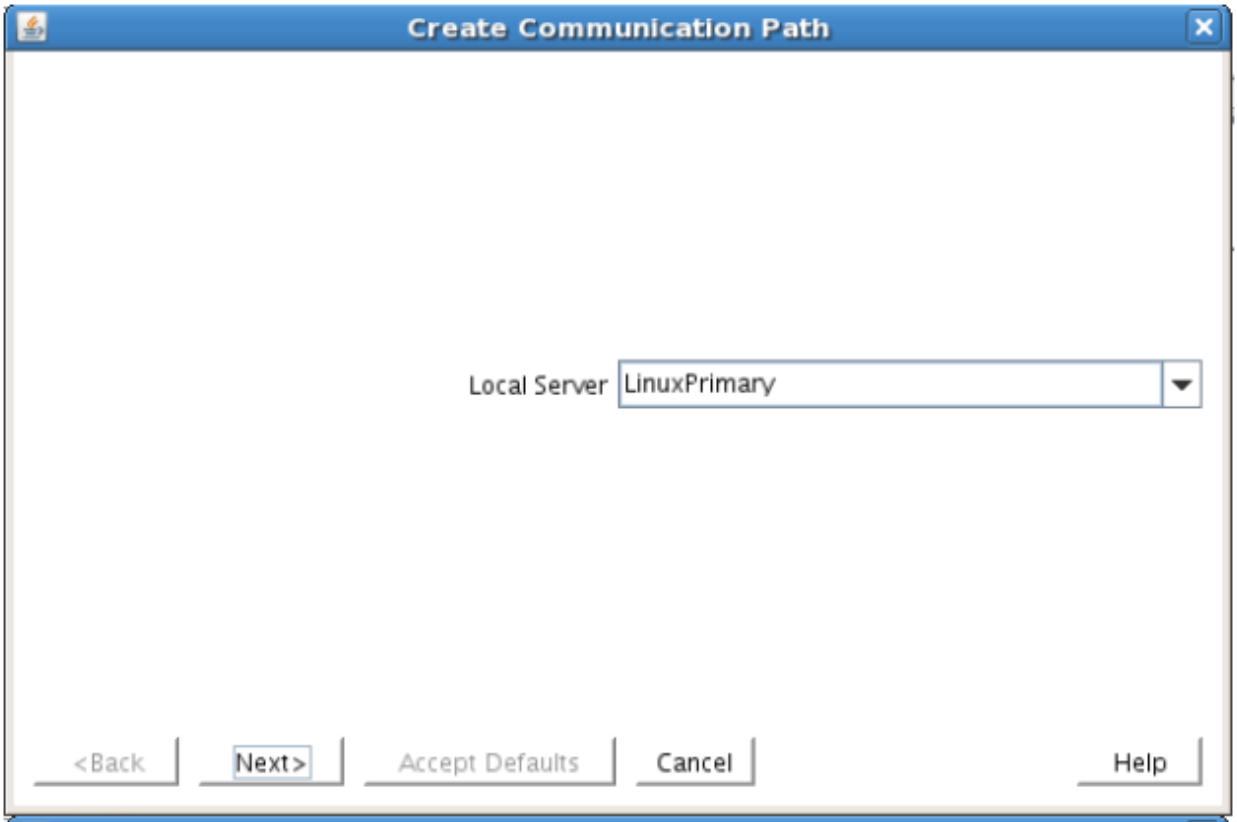

## Create Communication (Comm) Paths

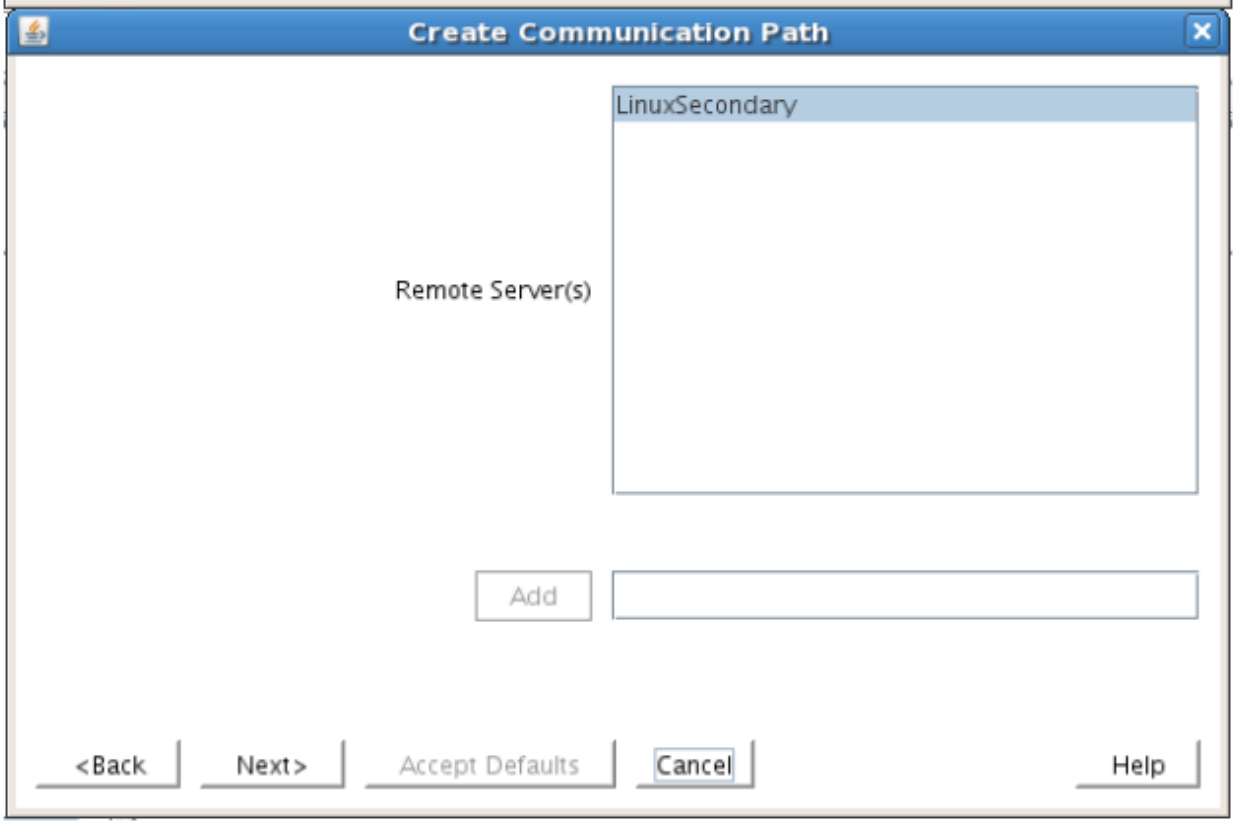
4. Within the LifeKeeper GUI, from the File menu, select Connect. Enter the name of your Secondary server, login and password when the Cluster Connect window displays.



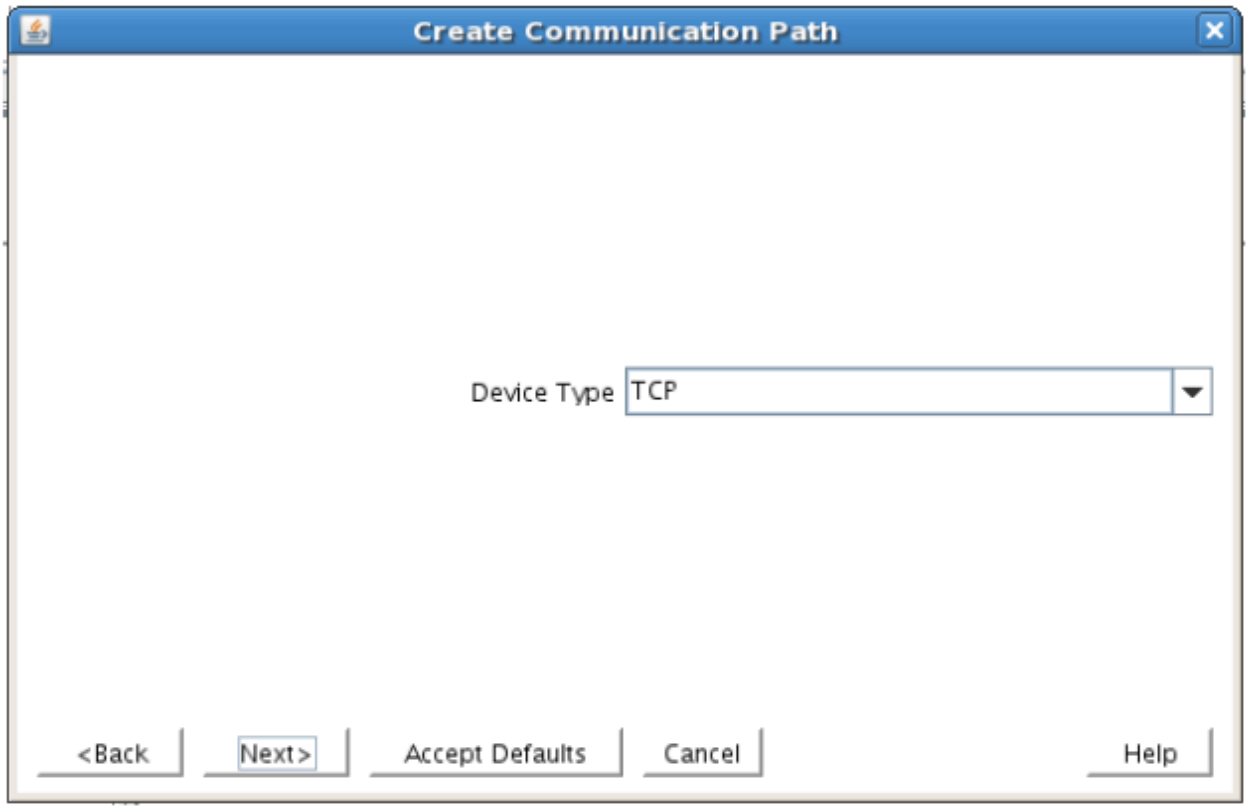

5. Within the LifeKeeper GUI, click the Create Comm Path button on the toolbar. You can also right click one of the servers and click Create Comm Path from the pop-up menu as well.

6. Select your Local and Remote Server(s) from the list box. If a server does not appear in the list box, you may enter it by typing its name and clicking the Add Server button. When using the Add Server procedure, you must make sure that the computer names for both network interfaces on the servers respond correctly when you ping them (from all of the partner server(s)) using the **ping –a IP ADDRESS** syntax. If they do not, this must be corrected prior to continuing. Click Next.

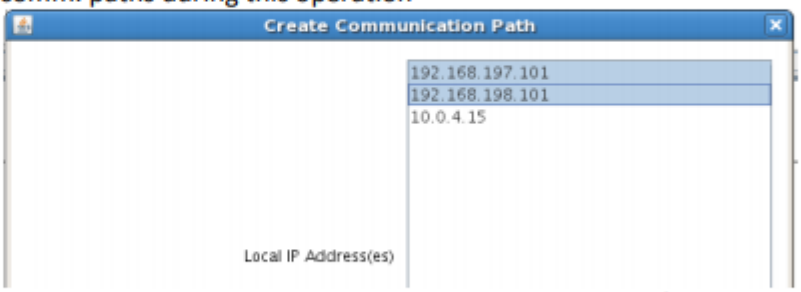


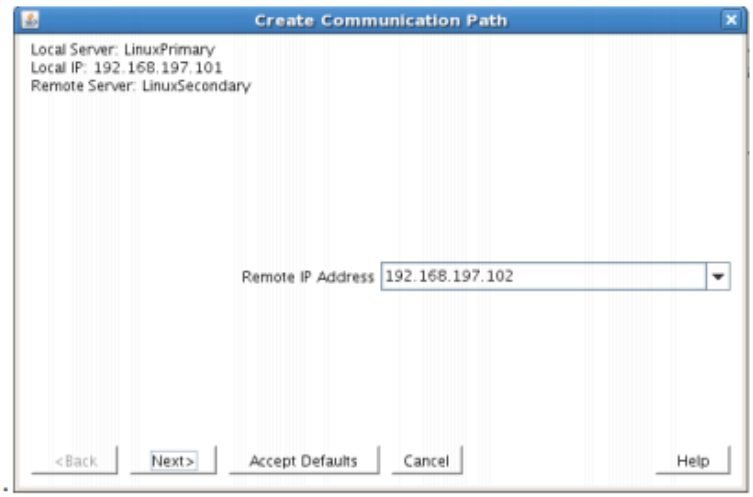




7. Select TCP for Device Type and Click Next.



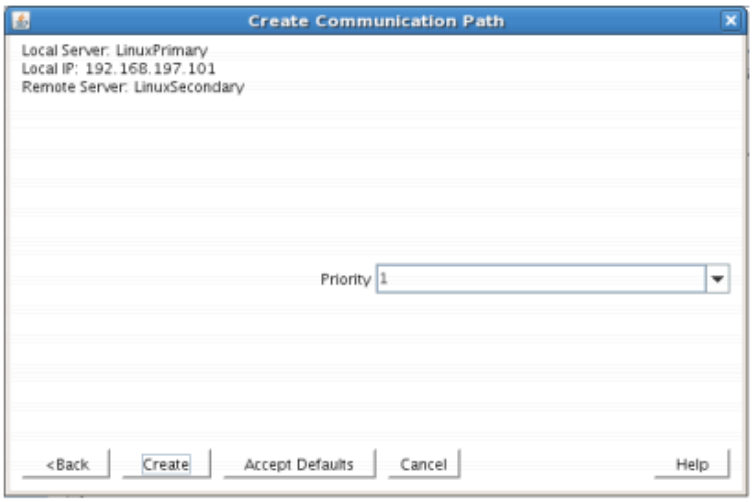

8. Provide all the required information and click Next for the following series of dialog boxes. For each field in the dialog box you can click Help for further information or refer to the table below for an explanation or recommendation

| Field | Tips |
| --- | --- |
| ***For TCP/IP Comm Path...*** | |
| Local IP Address | Choose the IP address to be used by the local server for this comm path. Select both interfaces so that the wizard creates multiple comm. paths during this operation   |
| Remote IP Address | Choose the IP address to be used by the remote server for this comm path |

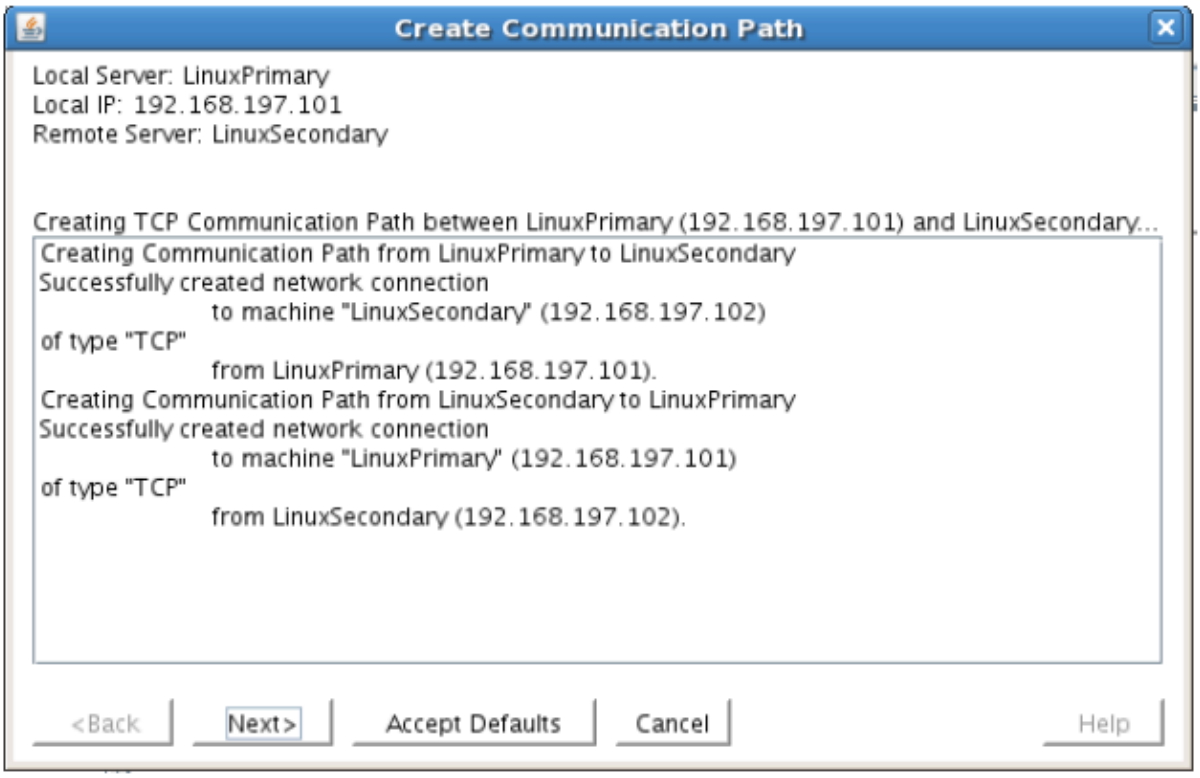



Enter the priority for the comm path on the local server. The priority will be used to determine the order that the comm paths between two servers will be used. Priority 1 is the highest; 99 is the lowest.

Priority




9. After entering data in all the required fields, select Create. A message will display indicating the network communication path is successfully created. Click Next.



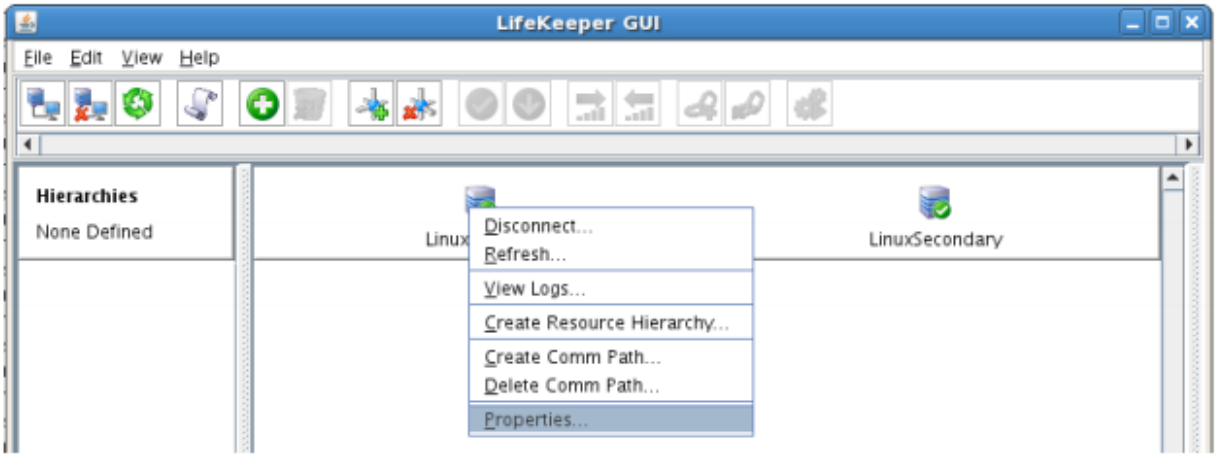

If you selected multiple Local IP Addresses or multiple Remote Servers and the Device Type was set to TCP, then the procedure will return you to the setup wizard the next Comm Path.

10. Select Done in the last dialog box.

    Repeat this process until you have defined all the communication paths you plan to use. SIOS strongly recommends that you define at least two communication paths for redundancy.

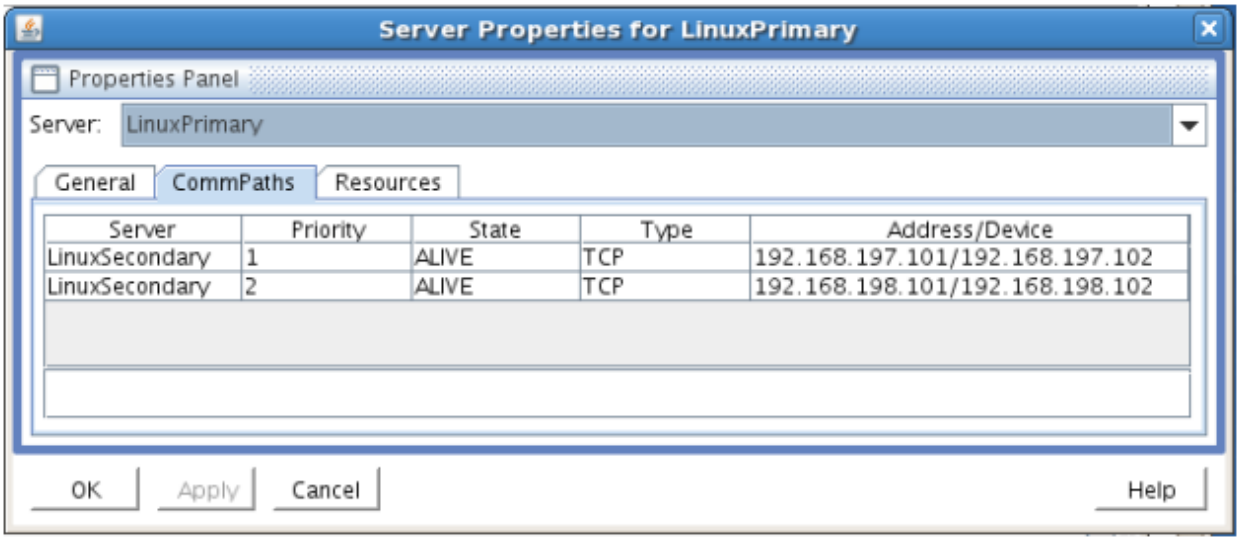
## Verify the Communications Paths

1. Verify that the communications paths are configured properly by viewing the Server Properties dialog box. From the LK GUI, select Edit, Server, Properties and then the Comm Paths tab.




2. Note the State displayed is ALIVE. You can also check the server icon in the right, main pane of

the GUI. If only one comm path has been created, the server icon shows a yellow warning icon on the server icon, indicating that one comm. path is ALIVE, but there is no redundant comm path. The server icon will display a green heartbeat checkmark when there are at least two comm paths configured and ALIVE.

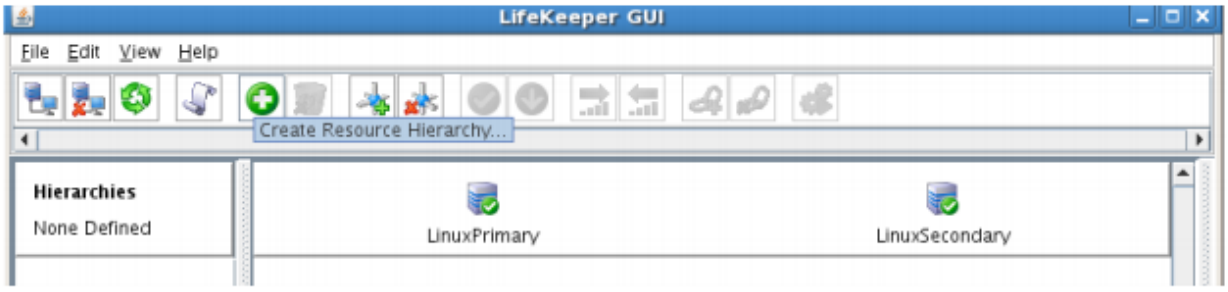



# Create the LifeKeeper Hierarchy

## Create and Extend an IP Resource

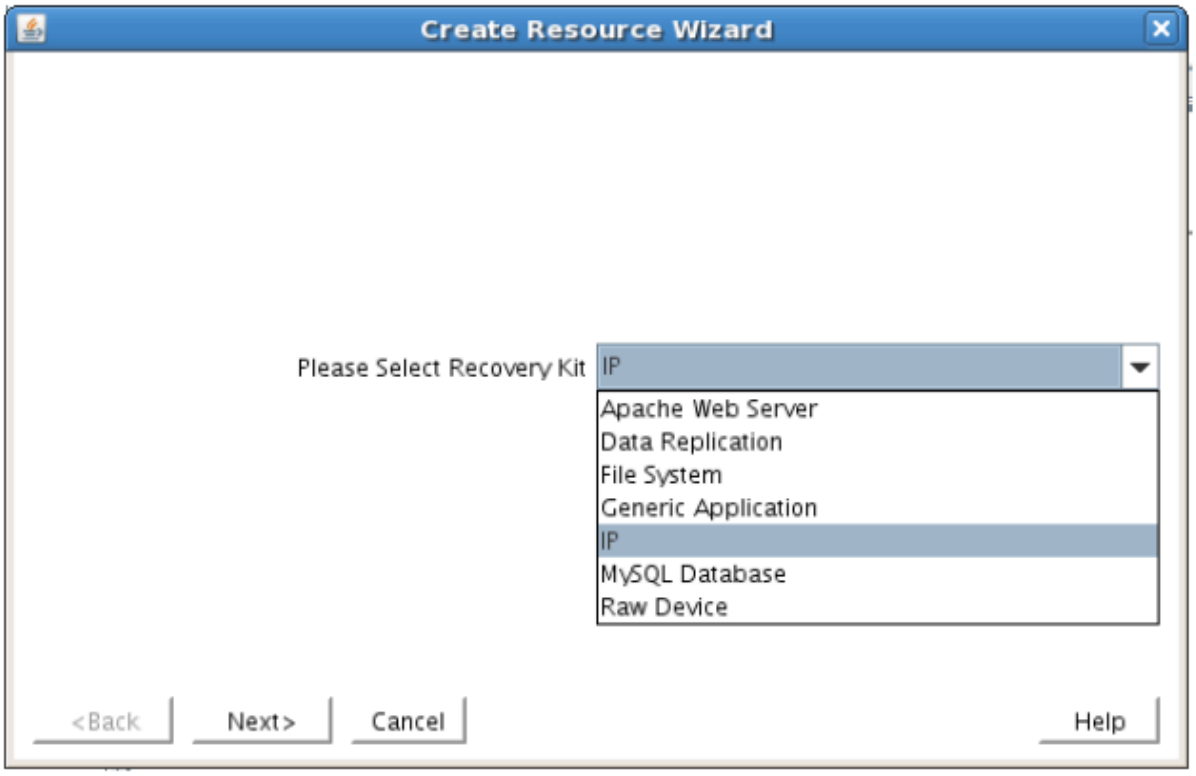
In LifeKeeper, create an IP resource and extend it to the secondary server by completing the following steps. This Virtual IP will have the ability to move between cluster nodes along the application that depends on it.

1. From the LifeKeeper GUI toolbar, click Create Resource Hierarchy.




The Create Resource Wizard dialog box will appear with a drop down list box displaying all recognized Recovery Kits installed within the cluster.

2. Select IP Address and click Next.



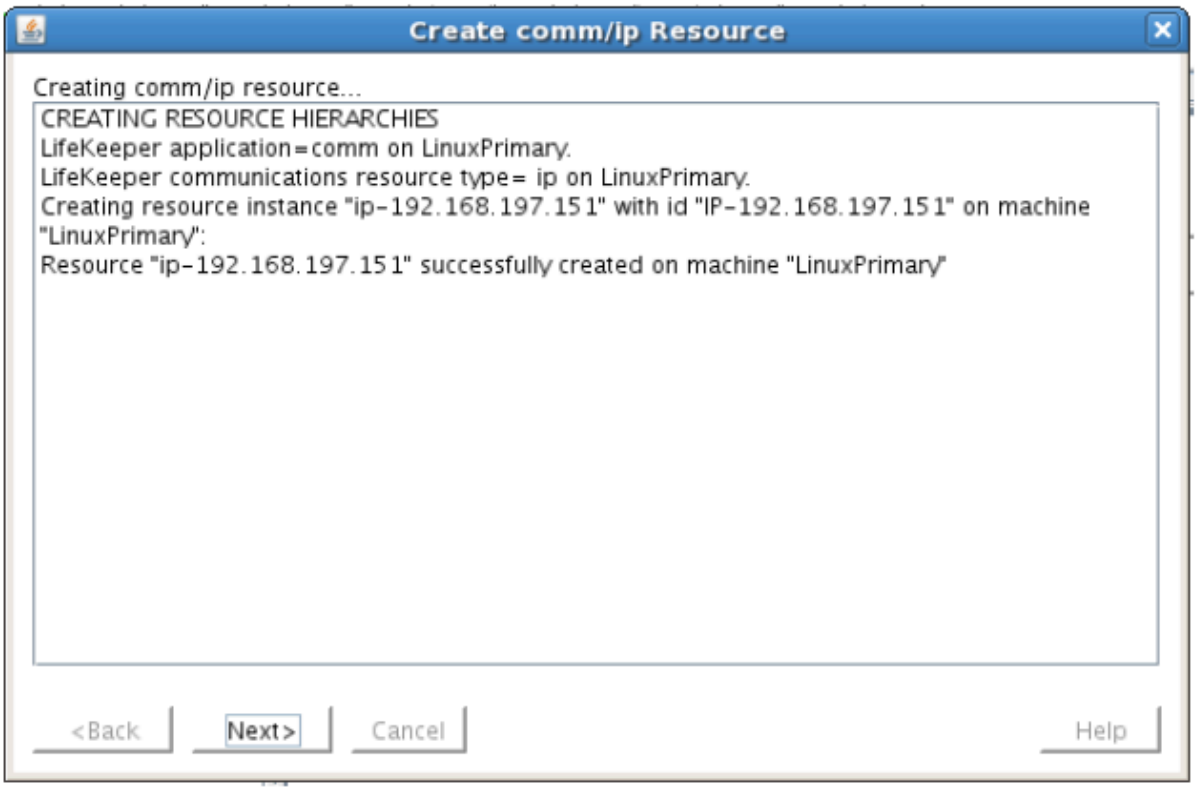

3. Enter the appropriate information for your configuration. The table below contains a list of the fields that display and additional information to assist you as you complete this procedure. Recommended values are also show below. You can also click the Help button for further information. Press Next to continue after entering the required information.

## IP Creation Field Definitions

| Field | Tips |
|---|---|
| Resource Type | Select IP Address as the resource type and click Next. |
| Switchback Type | Select Intelligent and click Next. |
| Server | Select the Server where the IP resource will be created. Select your<br><br>Primary server and click Next. |
| IP Resource | Enter the virtual IP information and click Next<br><br>Example **192.168.167.151**<br><br>**Note** This is an IP address that is not currently in use anywhere on your network. This is the address that all clients will use to connect to the protected resources. |

| | |
|---|---|
| Netmask | The IP subnet mask that your TCP/IP resource will use on the target server. Any standard netmask for the class of the specific TCP/IP resource address is valid.<br>In our sample configuration 255.255.255.0 is used for a subnet mask on both networks.<br><br>**Note:** The subnet mask you choose, combined with the IP address, determines the subnet that will be used by the TCP/IP resource and should be consistent with the network configuration. |
| Network Connection | This is the physical Ethernet card that the IP address is interfacing with. Chose the network connection that will allow your virtual IP address to be routable. Select the correct NIC and click Next. |
| IP Resource Tag | Accept the default value and click Next. This value only affects how the IP is displayed in the GUI. The IP resource will be created on our Primary server. |

4. LifeKeeper will create and validate your resource. After receiving the message that the resource has been created successfully, click Next when the following dialog box appears so that you can complete the process of Extending the IP Resource to our Secondary server, below.



Extending the IP resource will start automatically after you have finished creating an IP address resource if you clicked Next in the dialog box displayed above. You can also start this from an existing IP address resource by right clicking on the active resource and selecting Extend Resource Hierarchy.

Refer to the table below to complete the Extend IP Resource procedure.

| Field | Recommended Entries or Notes |
|---|---|
| Switchback Type | Leave as “intelligent” and click Next |
| Template Priority | Leave as default (1) |
| Target Priority | Leave as default (10) |
| Network Interface | This is the physical Ethernet card that the IP address is interfacing with. Chose the network connection that will allow your virtual IP address to be routable. The correct physical NIC should be selected by default. Please verify and then click Next |
| IP Resource Tag | Leave as default. |
| Target Restore Mode | Select Enable and click Next. |
| Target Local Recovery | Select Yes to enable Local Recovery for the SQL resource on the Target server. |
| Backup Priority | Accept the default value. |

5. After receiving the message Hierarchy extend operations completed, click Finish and then click Done

6. Your IP resource (192.168.197.151) is now fully protected and has the ability to “float” between cluster nodes as needed. Looking at the LifeKeeper GUI you will notice that the IP resource is Active on the Primary cluster node and Standby on the Secondary cluster node

## Create the Shared Filesystem Resource Hierarchy

Create a Filesystem resource to protect the shared iSCSI filesystem and make it high available between

cluster nodes. LifeKeeper leverages SCSI Persistent Group Reservations (PGR) to lock the LUN, ensuring that only the active cluster node for the storage resource can access it.

> **Important** At this point, the shared iSCSI LUN needs to already be mounted on the Primary Server. It should NOT be mounted on the Secondary Server. See section titled “Configure iSCSI initiator, discover and login to iSCSI target” above to review the steps involved.

1. From the LifeKeeper GUI toolbar, click Create Resource Hierarchy.
2. Select File System and click Next.
3. Proceed Through the Resource Creation wizard, providing the following values

| Field | Recommended Entries or Notes |
|---|---|
| Switchback Type | Intelligent |
| Server | LinuxPrimary (Primary Cluster Node) |
| Mount Point | Select /var/lib/pgsql . Note that LifeKeeper scans the system for LUNS that are sharable between cluster nodes. The list of possible shared LUNS is presented automatically in this step of the wizard. |

4. Select Create Instance to define this resource hierarchy on the Primary Server
5. Click Next to Extend the File System Resource to the Secondary Server
6. In the Extend Wizard, select “Accept Defaults”
7. As a result the File System resource is now protected on both cluster nodes. Click Finish to exit the Extend wizard.
8. Your resource hierarchy should look as follows:

## Create the PostgreSQL Resource Hierarchy

Create a PostgreSQL resource to protect the PostgreSQL database and make it high available between cluster nodes.

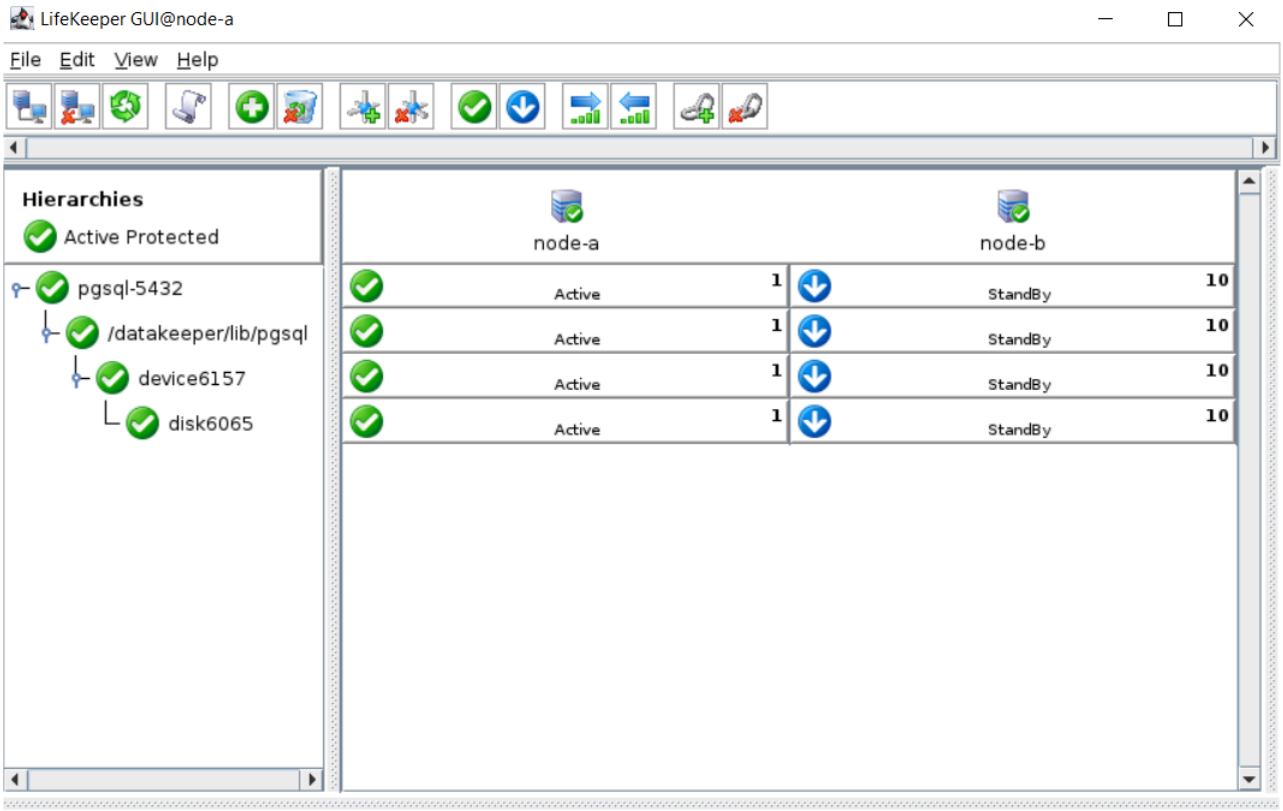
> **Important** At this point, PostgreSQL needs to be already running on the Primary Server. It should NOT be running on the Secondary Server. See section titled “Install, Configure, and Start PostgreSQL” above to review the process to configure and start PostgreSQL as needed.

1. From the LifeKeeper GUI toolbar, click Create Resource Hierarchy.
2. Select PostSQL Database and click Next.
3. Proceed Through the Resource Creation wizard, providing the following values

| Field | Recommended Entries or Notes |
|---|---|
| Switchback Type | Intelligent |
| Server | LinuxPrimary (Primary Cluster Node) |
| PostgreSQL Executable | Leave as default (/usr/bin) since we are using a standard PostgreSQL |
| Location | install/configuration in this example. This field is used to specify the directory path containing the PostgreSQL executables. |
| PostgreSQL Client | Leave as default (/usr/bin/psql) . This field is used to specify the directory |
| Executable Location | path containing the PostgreSQL executable psql. |
| PostgreSQL Administration | Leave as default (/usr/bin/pg_ctl). This field is used to specify the |
| Executable Location | directory path containing the PostgreSQL executable pg_ctl. |
| PostgreSQL Data Directory | /var/lib/pgsql/data. This field is used to specify the location of the PostgreSQL data directory (datadir) that will be placed under LifeKeeper protection. The specified directory must exist and reside on a shared or replicated file system. |
| PostgreSQL Port 5432. | This field is used to specify the TCP/IP port number on which the postmaster daemon is listening for connections from client applications |
| PostgreSQL Socket Path | Leave as default (/tmp/.s.PGSQL.5432 ) . This field is used to specify the full path to the Unix- domain socket on which the postmaster daemon is listening for connections from client applications. |
| Enter Database Administrator User | Enter “postgres” . This field is used to specify a PostgreSQL Database Administrator User name for the specified database instance with connection and administrator privileges for the instance |
| PostgreSQL Logfile | Leave as default (/tmp/pgsql-5432.lk.log) . This field is used to specify the log file path that will be used for the PostgreSQL log file. |
| Database tag | Leave as default |

4. Select Create to define the PostgreSQL resource hierarchy on the Primary Server
5. Click Next to Extend the File System Resource to the Secondary Server
6. In the Extend Wizard, select “Accept Defaults”
7. As a result the PostgreSQL resource is now protected on both cluster nodes. Click Finish to exit the Extend wizard.
8. Note: LifeKeeper will automatically identify that the PostgreSQL resource has a dependency on the FileSystem resource (/var/lib/pgsql). The Filesystem Resource will appear underneath the PostgreSQL resource in the GUI

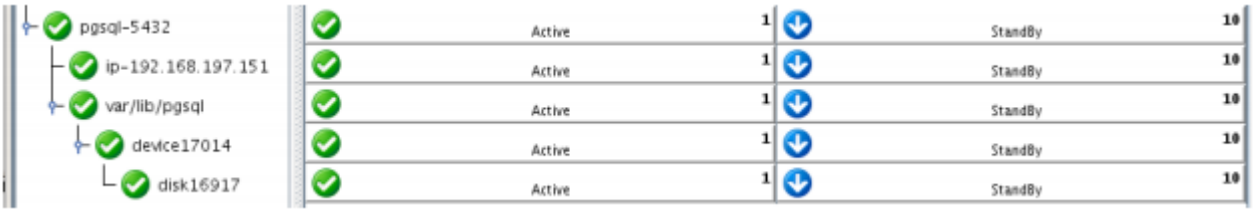
9. Your resource hierarchy should look as follows:




## Create the PostgreSQL IP Address Dependency

In this step will define an additional dependency: that PostgreSQL depends on a Virtual IP (192.168.197.151) so that the IP address follows the PostgresSQL database should it move.

1. From the LifeKeeper GUI toolbar, right-click on the “pgsql-5432” resource
2. Select “Create Dependency” from the right-click context menu
3. In the Child Resource Tag dropdown menu, select “ip-192.168.197.151”
4. Click Next
5. Click Create Dependency
6. Click Done
7. The Virtual IP address resource (192.168.197.151) will now appear underneath the PostgreSQL resource in the LifeKeeper user interface. This ensures that resources move together, and are started/stopped in the proper order.
8. Your resource hierarchy should look as follows




At this point in the Evaluation, we have fully protected PostgreSQL, and its dependent resources: IP addresses, and Shared Storage.

# 12.6.8. Test Your Environment – PostgreSQL

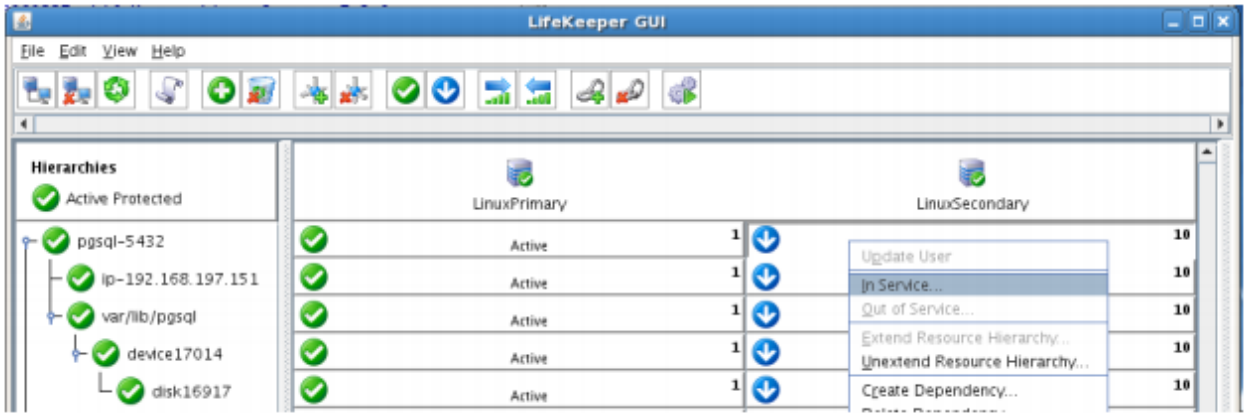
The following test scenarios have been included to guide you as you get started evaluating LifeKeeper for Linux. **Before beginning these tests, make sure the data replication resources are in the mirroring state**.

Note: For these test examples, the Primary Server is referred to as LINUXPRIMARY. The Backup or Secondary Server is referred to as LINUXSECONDARY.

## Manual Switchover of the PostgreSQL Hierarchy to Secondary Server

**Procedure:**

- From the LifeKeeper GUI, right click on the PostgreSQL resource on the Secondary Server (LINUXSECONDARY) and choose IN SERVICE.
- Click “In Service” in the window that pops up




**Expected Result:**

- Beginning with the PostgreSQL resource, all resources will be removed from service on the Active Server (LINUXPRIMARY).
- Beginning with the dependent resources (IP and Shared Volume), all resources will be brought in service on LINUXSECONDARY

**Tests/Verification:**

- Using the LifeKeeper GUI, verify that the PostgreSQL and dependent resources are active on LINUXSECONDARY.
- Run “ifconfig –a” on LINUXSECONDARY to validate that the IP Address 192.168.197.151 is active on LINUXSECONDARY
- Run “df –h” to verify that the /var/lib/pgsql shared iSCSI filesystem is mounted on LINUXSECONDARY
- Verify the PostgresSQL services are running on LINUXSECONDARY by running “ps –ef | grep –i

postgres”
- On LINUXSECONDARY run the following command to verify client connectivity to the PostgreSQL database:
    - # su – postgres
    - # psql
- Verify that the SCSI reservation has properly locked the iSCSI LUN to the currently active cluster node. On LINUXPRIMARY, run “mount /dev/sdc1 /var/lib/pgsql”. This should FAIL because LINUXPRIMARY does not own the SCSI reservation on this LUN.

# Manual Switchover of the PostgreSQL Hierarchy back to Primary Server

**Procedure:**

- From the LifeKeeper GUI, right click on the PostgreSQL resource on the Primary Server (LINUXPRIMARY) and choose IN SERVICE.
- Click “In Service” in the window that pops up

**Expected Result**

- Beginning with the PostgreSQL resource, all resources will be removed from service on the Active Server (LINUXSECONDARY).
- Beginning with the dependent resources (IP and Shared Volume), all resources will be brought in service on LINUXPRIMARY

**Tests/Verification:**

- Using the LifeKeeper GUI, verify that the PostgreSQL and dependent resources are active on LINUXPRIMARY.
- Run “ifconfig –a” on LINUXPRIMARY to validate that the IP Address 192.168.197.151 is active on LINUXPRIMARY
- Run “df –h” to verify that the /var/lib/pgsql shared iSCSI filesystem is mounted on LINUXPRIMARY
- Verify the PostgreSQL services are running on LINUXPRIMARY by running “ps –ef | grep –i postgres”
- On LINUXPRIMARY run the following command to verify client connectivity to the PostgreSQL database:
    - # su – postgres

    - #35; psql

- Verify that the SCSI reservation has properly locked the iSCSI LUN to the currently active cluster node. On LINUXSECONDARY, run “mount /dev/sdc1 /var/lib/pgsql”. This should FAIL because LINUXSECONDARY does not own the SCSI reservation on this LUN.

## Simulate a network failure on the Primary Server by failing the IP resource

> **IMPORTANT NOTE:** Only perform this test if you have more than one communications path configured.

If you perform this test and have only one communications path configured, your system will enter a split-brain scenario as described in the LifeKeeper Administration Guide found here. Refer to this document for more information or contact SIOS presales technical support for assistance in resolving this condition.

**Procedure**

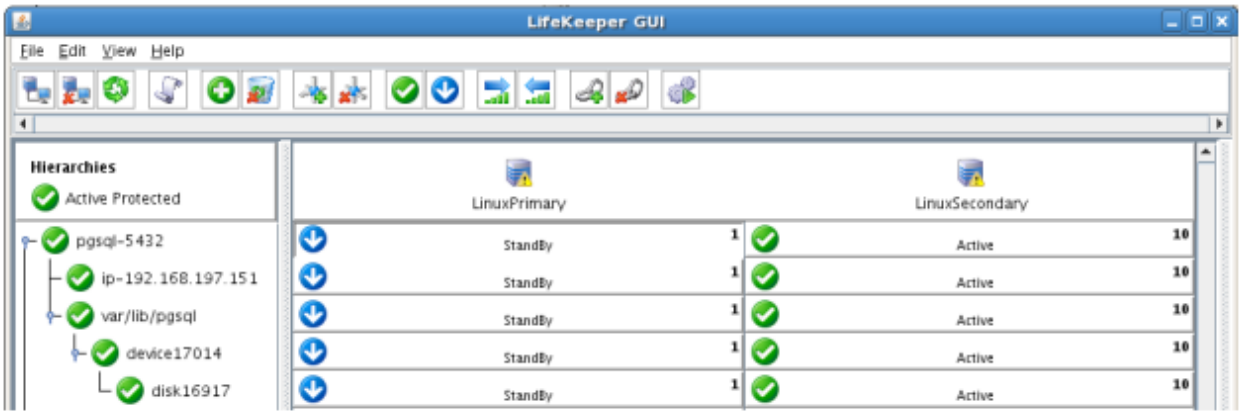
- On LINUXPRIMARY, pull the network cable attached to the NIC that the virtual IP address is configured on

**Expected Result:**

- The IP Resource should fail first.
- The entire hierarchy should failover to LINUXSECONDARY

**Tests/Verification:**

- Check the LifeKeeper Log to verify the IP resource failed – “/opt/LifeKeeper/bin/lk_log log”
- Using the LifeKeeper GUI, verify the PostgreSQL resource hierarchy fails over successfully to LINUXSECONDARY
- After this test has been completed, re-connect the network cable on LINUXPRIMARY



# Hard failover of the resource from the Secondary Server back to the Primary Server

**Procedure:**

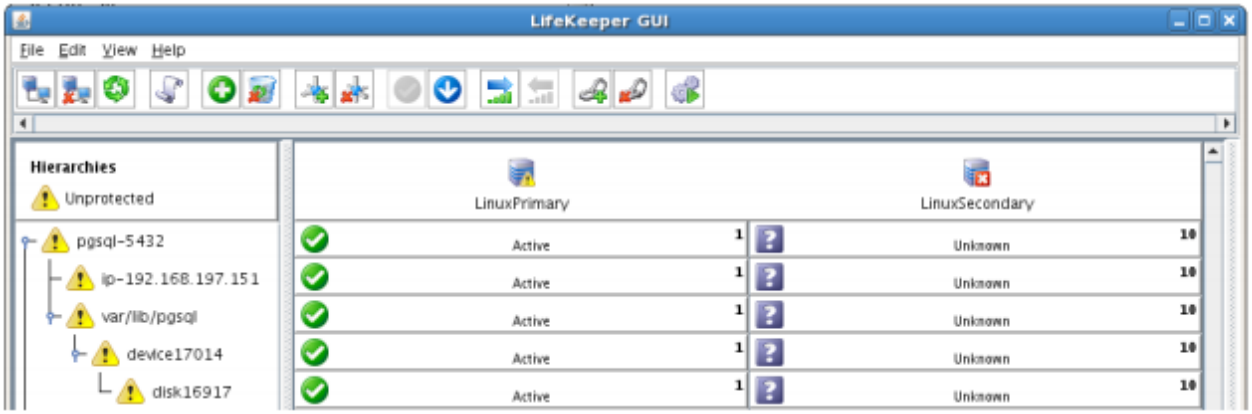
- Pull the power cord on LINUXSECONDARY, as this is the server with all resources currently In Service.

**Expected Result:**

- After failure has been detected, beginning with the dependent resources (IP and Volume), all resources will be brought in service on LINUXPRIMARY.

**Tests/Verification:**

- Using the LifeKeeper GUI, verify the mirror has reversed and is in a Resync Pending state waiting for LINUXSECONDARY to come back on line.
- Verify the PostgreSQL Server services are running on LINUXPRIMARY.




# Bring Failed Server back on line

**Procedure:**

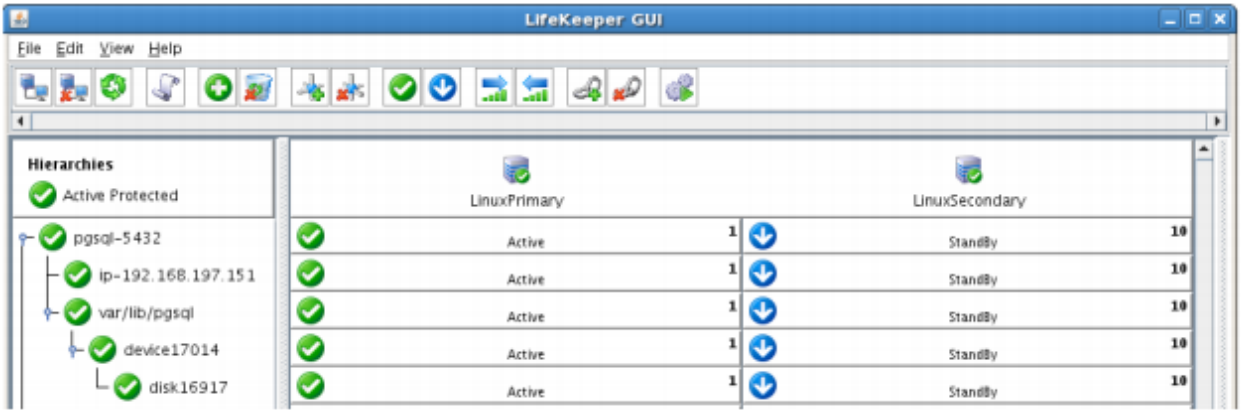
- Plug the power cord back into LINUXSECONDARY and boot it up.

**Expected Result:**

- Using the LifeKeeper GUI, verify that LINUXSECONDARY is coming back up and has become the Standby Server.

**Tests/Verification:**

- Verify the PostgreSQL Hierarchy is in service on LINUXPRIMARY and standby on LINUXSECONDARY.




# Verify Local Recovery of PostgreSQL Server

**Procedure:**

- Kill the PostgreSQL processes via the command line:
- # ps –ef | grep postgres
- # (kill -9 the PIDs returned)
- run “ps –ef | grep postgres once again to verify that the processes no longer exist

**Expected Result: (Assumes Local Recovery for SQL resource is set to YES)**

- The PostgreSQL Server service should stop.
- The PostgreSQL quickcheck process will automatically restart the PostgreSQL Server Service when it runs periodically.
- No failure of PostgreSQL should occur.

**Tests/Verification:**

- Execute “ps –ef | grep postgres” once again to verify that the postgresql processes have been restored locally on LINUXPRIMARY.
- Verify connectivity to the postgreSQL database by running:

  ↑
  - # su – postgres
  - # psql

- If you inspect the LifeKeeper logs, you will see information indicating that LifeKeeper dectected the failure of the PostgreSQL service and recovered it locally. Run /opt/LifeKeeper/bin/lk_log log for more information.

# 12.7. Apache/MySQL Cluster Using Both Shared and Replicated Storage

## Objective

This document is intended to aid you in installing, configuring and using the LifeKeeper for Linux evaluation product to make Apache and MySQL highly available. If Apache and MySQL are not already installed, please allocate some time to install it on your servers. Once this task has been completed, you may install and configure LifeKeeper for Linux.

There are five phases in this process:

- Phase 1 – Prepare to Install
- Phase 2 – Configure Storage
- Phase 3 – Install and Configure Apache/PHP
- Phase 4 – Install and Configure MySQL
- Phase 5 – Install LifeKeeper for Linux
- Phase 6 – Configure your LifeKeeper Cluster
- Phase 7 – Test Your Environment

# 12.7.1. Terms to Know – Apache

The following terms are used throughout this document and while some may be familiar to you, it may be helpful to review how SIOS defines and uses these terms.

## Network Communication Terms

**Crossover cable** – A cable used to directly connect computing devices together instead of being connected to a network switch, hub or router. This cable creates an isolated, private network to allow cluster-related and data replication traffic to flow between systems.

## Types of LifeKeeper Servers

**Server** – A computer system dedicated to running software application programs.

**Active Server** – This is the server where the resource hierarchy is currently running (IN SERVICE).

**Standby Server** – This is the server where the resource hierarchy is defined, but is not currently running. This server is available to bring the resource hierarchy into service should something happen to the resource hierarchy on the Active Server.

**Primary Server** – This is the server in a LifeKeeper configuration with the highest priority for a given resource hierarchy. It is the server that provides services for the resource hierarchy under normal circumstances.

**Secondary Server** – This is the server in a LifeKeeper configuration with the 2nd highest priority for a given resource hierarchy.

**Source Server** – This is the server in a LifeKeeper cluster that is using data replication (Active Server). It is where the resource hierarchy is currently running and the replicated partition (Source Partition) is accessible for writes.

**Target Server** – This is the server in a LifeKeeper cluster using data replication (Standby Server). The replicated partition (Target Partition) is updated with writes from the Source Partition by the SIOS Data Replication system. This partition should not be accessed/modified manually.

## SIOS Data Replication Terms

**Replication** – Transferring data from one partition to another via a sector-by-sector copy. During replication, the target partition should not be accessed or modified assuring your data integrity.

**Synchronous** – A replication scheme in which the data is confirmed written and valid on the target before the write operation occurs on the source disk through a series of information exchanges. Synchronous mirrors should only be implemented on high speed (100Mbps+) networks due to the network overhead involved.

**Asynchronous** – A replication scheme in which the data is released for writing on the source immediately and is sent to the target(s) simultaneously for writing as fast as the data can get there andcan be written on them.

**Rate of Change** – A measure of the amount of data which is changing over a set period of time.

**Compression** – An algorithm which is optionally implemented to reduce the amount of traffic between source and target nodes. Nine levels of compression are offered. Compression is turned off by default.

**Throttling** – An optionally implemented mechanism to limit the bandwidth used for replication.

## LifeKeeper Product Terms

**Communications Path** – A mechanism supporting communication between nodes in a LifeKeeper cluster. SIOS highly recommends implementing multiple communication paths between all servers in the cluster to eliminate a single point of failure.

**Heartbeat** – A periodic message exchanged between nodes in a LifeKeeper cluster that provides server health monitoring. A heartbeat message is one type of inter-node cluster communication sent over a communications path.

**Split Brain** – A situation in which all communications paths between cluster members fail, but all servers remain up and running. In this situation, both systems believe the other has failed and both believe they should keep or bring resources into service.

**Failover** – The unplanned migration of a resource hierarchy to the Standby Server because of a system or resource failure on the Active Server.

**Switchover** – The planned migration of a resource hierarchy from the Active Server to the Standby Server.

**Switchback** – The setting that governs the recovery behavior of the server where the resource was in service when it failed. If the setting is “intelligent”, the server acts as a possible backup for the given resource. If the setting is “Automatic”, the server actively attempts to re-acquire the resource without further notice.

**Resource** – A system asset that can be protected by LifeKeeper. Resources can be used to represent disk partitions, virtual IP addresses, applications, etc.

**Extend a Resource** – Create or define an already configured LifeKeeper resource onto another server in the cluster and build an equivalency relationship that prevents the resource from coming in service on both systems simultaneously.

**Resource Hierarchy** – A grouping of resources, in a predetermined order, from high to low. This may also be referred to as simply a Hierarchy.

**Shared Storage** – One or more logical disk partitions that are physically attached to all nodes in a cluster. LifeKeeper ensures that the volume is only accessible by one server at a time. This is formally

called I/O fencing.

**Data Replication (Disk Mirroring)** – The replication of logical disk partitions to separate physical hard disks in real time to ensure continuous availability, currency and accuracy of data.

**Source** – The partition on the source server used for replication. The “gold” copy of the data.

**Target** – The partition on the target server used for replication.

**Switchable IP Address** – A unique IP address that may be moved between systems in the cluster. Client systems connect to this address and the system where the virtual IP resource is active will respond to requests.

# 12.7.2. The Evaluation Process – Apache

SIOS strongly recommends performing your evaluation of LifeKeeper for Linux in a test lab environment. SIOS is not responsible and cannot provide support for evaluation software installed in a production environment.

All questions during the evaluation period should be directed to evalsupport@us.sios.com or your local Pre-Sales Engineering contact. Pre-sales support will contact you by the next business day to answer questions. Once you are a licensed customer on software maintenance, you will have access to 24 × 7 post-sales technical support.

**Important** Your evaluation license is valid for a limited period of time from the day you receive the SIOS product evaluation package and licenses via email from the SIOS sales team.

# 12.7.3. Prepare to Install – Apache

## Hardware Requirements

### Primary and Secondary Servers

- Systems must meet the minimum requirements for the Linux distribution to be used during the evaluation.
- 512MB RAM minimum; 1GB RAM recommended.
- 2GB of available hard disk space recommended.
- Multiple Network Interface Cards (NIC's) are recommended.
- Configure one or more additional disk partitions to be used for data replication. On the primary server, these will become the source partitions. On the secondary server(s), these will become the target partitions. In this evaluation example we will be replicating MySQL data (which will be a partition mounted at /var/lib/mysql)
- For replicated partitions, a target partition's size must equal to or larger than the size of its source partition.
- The system ( / ) and boot (/boot) partitions are not eligible for replication.

**Note:** You may use more than one partition for replicated data, allowing for separation of multiple database files and/or log directories for performance reasons.

### Client

This system is not required but is recommended for testing the cluster environment.

- A standard linux terminal running the MySQL client can be used to test the configuration.

## Software Requirements

### Primary Server and Secondary Server

- Linux Distribution x86_64, AMD 64:
    - RedHat Enterprise Linux 5 (5.4+ recommended) or 6.x
    - CentOS Linux 5 (5.4+ recommended) or 6.x
    - Oracle Enterprise Linux 5 (5.4+ recommended), 6.3, 6.4
        - RedHat Compatibility Kernel Only
    - SuSE Linux Enterprise Server 10 or 11 (11 recommended)
    - See Linux Release Notes for a full list of supported Operating Systems

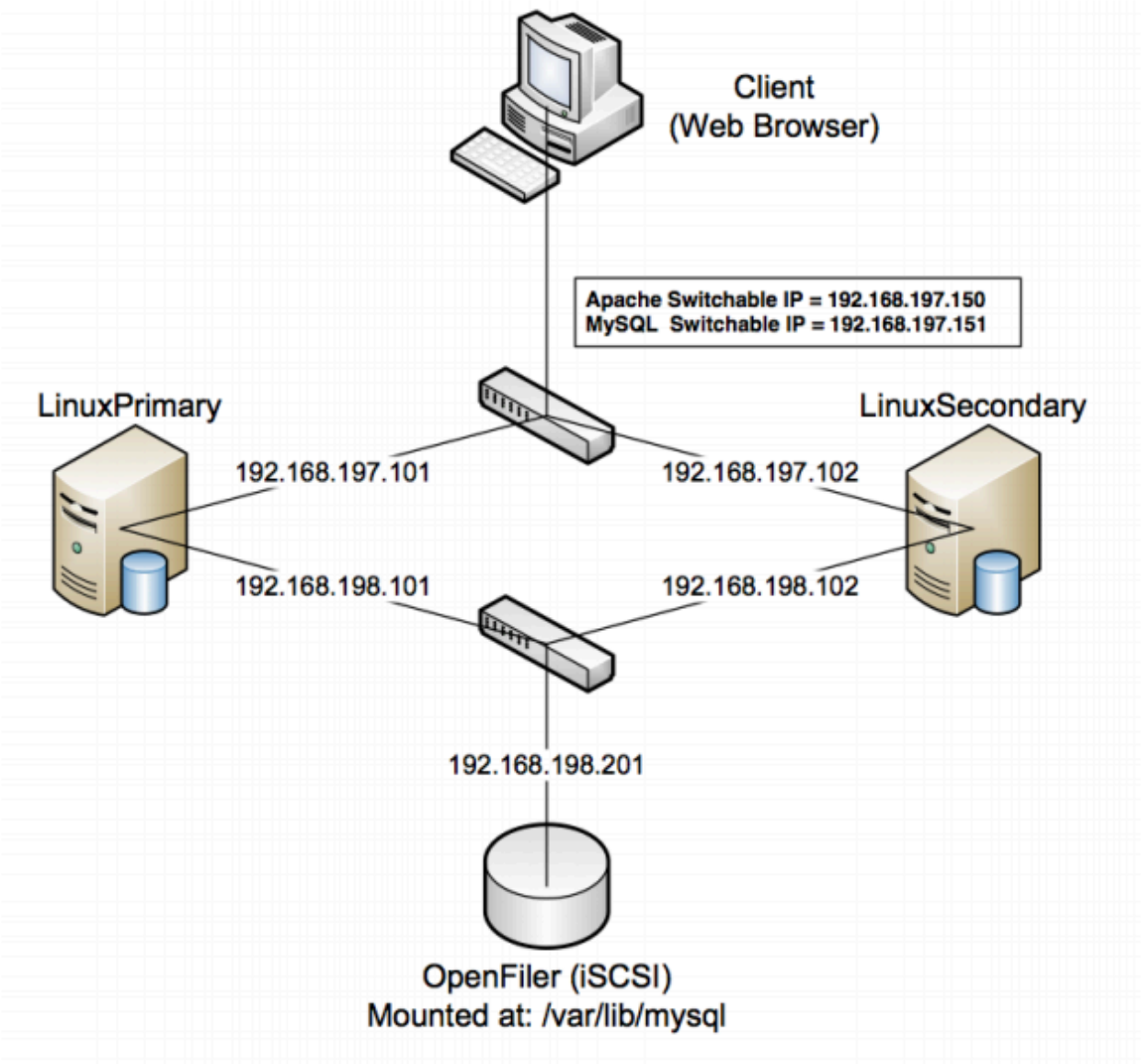
- Current patches / security updates are recommended.
- Satisfied dependencies; especially if the Linux installation package selection was base/minimal you will need to refer to the dependencies documentation at Linux Dependencies
- Its recommended that IPtables is disabled
    - # /etc/init.d/iptables off
    - # chkconfig iptables off
    - See here for information regarding the ports LifeKeeper for Linux uses.
- Disable SELinux :
    - Edit /etc/selinux/config
    - Set SELINUX=disabled (note: permissive mode is also acceptable)
- Check the configuration of your /etc/hosts file
    - localhost.localdomain and localhost are the only entries that can be on 127.0.0.1
    - Create a separate entry for your hostname with a static address
- GUI Authentication with PAM
    - LifeKeeper for Linux now leverages the Pluggable Authentication Module (PAM) provided in the Linux Standard Base (LSB).
        - Users are identified and authenticated against the system's PAM configuration. Privilege levels are determined from group membership as provided through PAM.
        - In order to access the GUI, a user must be a member in one of the three LifeKeeper groups: lkadmin, lkoper or lkguest.
        - See the following URL for more information on this topic:
            - Configuring GUI Users

# Network Requirements

For your evaluation, we recommend configuring your machines similarly to the following example.

LinuxPrimary and LinuxSecondary are multi-homed, between two LAN segments (the second NIC in each server could even be connected via a cross-over cable if a second physical network is not available). The second NIC is optional in this configuration, but highly recommended in production environments to avoid a single point of failure.

In this example evaluation scenario we will be leveraging both local, replicated storage (with the Apache configuration) as well as Shared Storage (iSCSI, for the MySQL configuration). OpenFiler is a storage appliance server that will serve an iSCSI target to LinuxPrimary and LinuxSecondary.




**Network Configuration Example**

## Primary Server and Secondary Servers

- Configure the Host file with entries for all LifeKeeper protected servers. This is typically /etc/hosts.

  Example:

  192.168.197.101 LinuxPrimary

  192.168.197.102 LinuxSecondary

- See your Network Administrator to obtain an unused IP Address to be used as the switchable IP Address. This switchable IP Address will be created later in the configuration process.
- Public Network connection(s) configured with:
    - Static IP address
    - Correct subnet mask
    - Correct gateway address
    - Correct DNS server address(es)
- Private Network connection(s) configured with:
    - Static IP address (on a different subnet from the public network)
    - Correct network mask
    - No gateway IP address
    - No DNS server addresses

## Client

Must be able to communicate on the same subnet/network as the servers Public interface addresses. In our example, this is the 192.168.197.0/24 network.

# 12.7.4. Configure Storage – Apache

## Before you Begin

Ensure the following:

- You have an extra disk/partition on both servers that can be used for data replication. A target volume's size must equal to or larger than the size of its source disk/partition.
- Shared storage is available. This can either be Fiber Channel SAN, iSCSI, NAS, etc. In this example we will review configuration of an iSCSI target for use as our MySQL database storage repository.

## Partition local storage for use with SIOS DataKeeper for Linux

### Primary Server

On your Primary server, perform the following actions:

1. Identify an existing free, unused disk partition to use as our Apache repository. Alternatively, create a new partition. Use the "gdisk" utility to partition your disk appropriately. In this example /dev/sdb is an unused disk where we will create a single partition

    a. gdisk /dev/sdb

    b. Press "n" to create a new partition

    c. Press "p" to create a primary partition

    d. This example uses a new disk, so we will use all default values (Partition 1, entire disk) Hit Enter twice to confirm these parameters

    e. Press "w" to write the partition table and exit gdisks

**Example**

[root@LinuxPrimary ~]# **gdisk /dev/sdb**

Command (m for help): **n**

Command action

e extended

p primary partition (1-4)

**p**

Partition number (1-4): **1**

First cylinder (1-256, default 1): **<enter>**

Using default value 1

Last cylinder or +size or +sizeM or +sizeK (1-256, default 256): **<enter>**

Using default value 256

Command (m for help): **w**

The partition table has been altered! Calling ioctl() to re-read partition table.

Syncing disks.

[root@LinuxPrimary ~]#

[root@LinuxPrimary ~]# df /var/www

Filesystem 1K-blocks Used Available Use% Mounted on

/dev/sdb1 253855 11083 229666 5% /var/www

[root@LinuxPrimary ~]# ls /var/www

cgi-bin error html icons lost+found manual usage

2. Format the newly created disk partition

   # mkfs.ext3 /dev/sdb1

3. Mount the partition temporarily at /mnt

   # mount /dev/sdb1 /mnt

4. Move any existing data from /var/www/ into this new disk partition (assumes a default apache configuration)

   # cd /var/www

   # mv * /mnt

5. Remount /dev/sdb1 at /var/www

   # cd /root

```
# umount /mnt

# mount /dev/sdb1 /var/www
```

6. Note: there is no need to add this partition to /etc/fstab. Lifekeeper will take care of mounting this automatically.

**Result**

```
[root@LinuxPrimary ~]# df /var/www
Filesystem 1K-blocks Used Available Use% Mounted on
/dev/sdb1 253855 11083 229666 5% /var/www
[root@LinuxPrimary ~]# ls /var/www
cgi-bin error html icons lost+found manual usage
```

### Secondary Server

7. On your Secondary server, only perform Step #1 above, where you partition the disk. The size of the Target volume needs to be the same size, or greater, than our Source volume.

## Configure iSCSI initiator, discover and login to iSCSI target

This Evaluation guide will not cover how to setup an iSCSI Target Server. It is assumed that the shared storage already exists in your environment. If you don't have shared storage and wish to configure it, a simple solution is to use OpenFiler (http://www.openfiler.com/), an Open Source storage management appliance, which can be run on physical hardware or as a virtual machine.

On both Primary and Secondary servers, perform the following functions:

1. If not already installed, ensure that the **iscsi-initiator-utils** rpm package is installed:

```
# yum install iscsi-initiator-utils
```

2. Start the iscsid service and enable it to automatically start when the system boots

```
# service iscsid start

# chkconfig iscsid on
```

3. Configure the iscsi service to automatically start, which logs into iSCSI targets needed at system start up.

```
# chkconfig iscsi on
```

4. Use the iscsiadm command to discover all available targets on the network storage server (OpenFiler)

```
# iscsiadm -m discovery -t sendtargets -p <name or IP of iSCSI server>
```

**Example**
[root@LinuxPrimary init.d]# iscsiadm -m discovery -t sendtargets -p 192.168.198.201
**iqn.2006-01.com.openfiler:tsn.mysql**

5. Manually Login to the iSCSI Target

   # iscsiadm -m node –T iqn.2006-01.com.openfiler:tsn.mysql –p 192.168.198.201 –login

6. Configure Automatic Login

   # iscsiadm -m node –T iqn.2006-01.com.openfiler:tsn.mysql –p 192.168.198.201 —op update -n node.startup -v automatic

7. Use the “gdisk” command to format your iSCSI LUN, if needed

   # gdisk /dev/sdc

8. Create a filesystem on your new iSCSI LUN Partition, sdc1

   # mkfs.ext3 /dev/sdc1

9. Mount your iSCSI LUN at /var/lib/mysql (assuming a default mysql configuration). If data already exists in this directory, make sure to move it into the shared iSCSI LUN

   # mount /dev/sdc1 /var/lib/mysql

10. At this point you now have a local partition, /dev/sdb1 mounted at /var/www and an iSCSI shared LUN, /dev/sdc1, mounted at /var/lib/mysql. Our disk layout now look as follows (example):

**Example**
[root@LinuxPrimary mysql]# df
Filesystem 1K-blocks Used Available Use% Mounted on
/dev/sda2 25967432 3683016 1976400 66% /
/dev/sda1 101086 24659 71208 26% /boot
tmpfs 517552 0 517552 0% /dev/shm
**/dev/sdb1 253855 11132 229617 5% /var/www**
**/dev/sdc1 966644 38944 878596 5% /var/lib/mysql**

# 12.7.5. Install and Configure Apache and PHP

## Before you Begin

Ensure the following:

1. If you are not familiar with installing and configuring Apache, please refer to the Apache documentation.
2. Apache should not be running at the time you attempt to protect it with LifeKeeper.

> **Important:** If Apache and PHP are already installed, skip to step #2 in the section below. You will need to verify the Apache configuration and/or relocate the location of the website data.

## Primary Server

1. Install Apache.

   a. This example assumes that you are running a RHEL or CentOS 5.X based Linux distribution. For other Linux distros, please refer to your apache documentation for syntax differences.

   b. If the Apache and PHP packages are not already installed, from the command line, , run: “yum install httpd php”

   c. A number of dependencies will most likely be discovered. Install those as well.

2. Apache should be configured so that it will not automatically start when the server boots. LifeKeeper will control the start/stop of the webserver once its protected

   a. Check the status of the webserver: “/etc/init.d/httpd status

   b. If running, please stop of: /etc/init.d/httpd stop

   c. Disable automatic startup: chkconfig httpd off

3. Apache Configuration. In this example, we will assume default Apache settings and directory locations, specifically:

   a. Validate the following settings in /etc/httpd/conf/httpd.conf:

      i. ServerRoot “/etc/httpd”

      ii. DocumentRoot “/var/www/html”

iii. Listen 192.168.197.150:80 (note 192.168.197.150 is the Apache Switchable IP we will configure later in the LifeKeeper user interface)

b. Edit the Listen parameter in the /etc/httpd/conf.d/ssl.conf configuration file

i. Listen 192.168.197.150:443

4. Create a sample Index page:

a. vi /var/www/html/index.php

b. Insert the following single line of code into this file:

<? phpinfo(); ?>

## Secondary Server

1. Install Apache/PHP exactly as you did on the primary server, making all of the same configuration changes.
2. There is no need to perform Step #4 in which you create a sample index page. All data in /var/www will be replicated from LinuxPrimary to LinuxSecondary
3. Stop all Apache services on the secondary server

# 12.7.6. Install, Configure, and Start MySQL – Apache

## Primary Server

On your Primary server, perform the following actions:

1. Install both the “mysql” and “mysql-server” rpm packages if they do not exist on your system.

   Apply any required dependencies as well

   ```
   # yum install mysql mysql-server
   ```

2. Verify that your Shared iSCSI LUN is still mounted at /var/lib/mysql via the “df” command

3. If this is a fresh MySQL install, initialize a sample MySQL database:

   ```
   # /usr/bin/mysql_install_db —datadir=”/var/lib/mysql” —user=mysql
   ```

4. Ensure that all files in your MySQL data directory (/var/lib/mysql) have correct permissions and ownership

   ```
   # chown –R mysql:mysql /var/lib/mysql
   ```

   ```
   # chmod 755 /var/lib/mysql
   ```

5. Finally, manually start the MySQL daemon from the command line. Note: **Do Not** start it via the “service” command, or the /etc/init.d/ scripts

   ```
   # mysqld_safe —user=root —socket=/var/lib/mysql/mysql.sock —port=3306 —datadir=/var/lib/mysql —log &
   ```

6. Verify MySQL is running by connecting with the mysql client:

   ```
   [root@LinuxPrimary mysql]# mysql

   Welcome to the MySQL monitor. Commands end with ; or \g.

   Your MySQL connection id is 2

   Server version: 5.0.77-log Source distribution

   Type ‘help;’ or ‘\h’ for help. Type ‘\c’ to clear the buffer.

   mysql> exit
   ```

Bye

[root@LinuxPrimary mysql]#

7. Update the root password for your mysql configuration. In this example we set the MySQL root password to “SteelEye”

   # echo “update user set Password=PASSWORD where User=‘root’; flush

   privileges” | mysql mysql

8. Verify your new password:

   # mysql mysql –u root –p

   (Enter “SteelEye” as the password)

   #exit

9. Create a MySQL configuration file. We will place this in the same shared directory (/var/lib/mysql/my.cnf)

   # vi /var/lib/mysql/my.cnf

**Example**

```
 # cat /var/lib/mysql/my.cnf
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
pid-file=/var/lib/mysql/mysqld.pid
user=root
port=3306
 # Default to using old password format for compatibility with mysql 3.x
 # clients (those using the mysqlclient10 compatibility package).
old_passwords=1
 # Disabling symbolic-links is recommended to prevent assorted security risks;
 # to do so, uncomment this line:
 # symbolic-links=0
[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
[client]
user=root
password=SteelEye
```

10. Delete the original MySQL configuration file, located in /etc

    # rm /etc/my.cnf

# Secondary Server

On your Secondary Server:

1. Install both the “mysql” and “mysql-server” rpm packages if they do not exist on your system.

   Apply any required dependencies as well

   ```
   # yum install mysql mysql-server
   ```

2. Ensure that all files in your MySQL data directory (/var/lib/mysql) have correct permissions and ownership

   ```
   # chown –R mysql:mysql /var/lib/mysql
   ```

   ```
   # chmod 755 /var/lib/mysql
   ```

3. There is no need to perform any of the additional steps taken on the Primary Server

# 12.7.7. Install LifeKeeper for Linux – Apache

For ease of installation, SIOS has provided the LifeKeeper for Linux with an installation script. Towards the end of the script, the desired Application Recovery Kits (ARKs) should be selected for installation. The software will be installed to the following locations:

| SPS for Linux Component | Install Location |
|---|---|
| LifeKeeper Software | /opt/LifeKeeper |
| LifeKeeper Config File | /etc/default/LifeKeeper |

Perform the following actions on **both** Primary and Secondary server.

## Download Software

1. Open the LifeKeeper evaluation email you received from SIOS.

2. Download the LifeKeeper Software from the link provided in your email. It is generally easiest to use “wget” to recursively download all files. Example:

   a. # cd /root

   b. # wget -r <URL>

   c. After successful download you will have downloaded contents similar to the follow directory listing:

   [root@LinuxPrimary ~]# ls -l <directory> total 63680

   -rw-r—r— 1 root root 23163 May 30 14:03 EULA.pdf

   -rw-r—r— 1 root root 536 May 30 14:03 readme.txt

   -rw-r—r— 1 root root 65179648 May 30 14:03 sps.img

3. Download your Evaluation license key from the link specified in your evaluation email. Save the license file to an easy to remember location on both servers.

## Run the LifeKeeper Installer Script

1. Loopback mount the sps.img file previously downloaded, which is an ISO9660 image file. Run the “setup” script inside:

   # mount -o loop sps.img /mnt

   # cd /mnt

```
# ./setup
```

2. During this procedure, you will hit Enter in most cases to accept default values and continue to the next screen. Note the following exceptions:

   a. On the screen titled “High AvailabilityNFS” you may select “n” as in this particular eval guide we will not be creating a highly available NFS server cluster configuration.

   b. If you have plans to create a highly available NFS service, adjust your response accordingly.

3. Towards the end of the setup script, you can choose to install a trial license key now, or later. We will install the license key in the next step, so you can safely select “n” at this point

4. In the final screen of the “setup” select the DataKeeper from the list displayed on the screen.

5. The following RPMs should be be installed:

   a. steeleye-lkAPA-<version>.noarch.rpm

   b. steeleye-lkSQL-<version>.noarch.rpm

   c. steeleye-lkDR-<version>.noarch.rpm

6. Un-mount the Distribution Enabling disk image:

```
# cd /root

# umount /mnt
```

## Install the Evaluation License Keys

The last phase of the setup process installs the licensing keys. You must install the evaluation license key file (“.lic”) that you downloaded with your evaluation software before starting the LifeKeeper for Linux.

1. To install your trial license key, run the “lkkeyins” command on both Primary and Secondary Server. This command is located at /opt/LifeKeeper/bin/lkkeyins . Example:

```
# /opt/LifeKeeper/bin/lkkeyins <path_to_license/<filename>.lic
```

2. Validate your license keys were installed via the /opt/LifeKeeper/bin/lklicmgr command

```
# /opt/LifeKeeper/bin/lklicmgr

License File: 20101230.lic
```

| Product | Type | Expiry |
|---|---|---|

| | | |
|---|---|---|
| LifeKeeper for Linux | Eval | 27 Mar 2013 (87 days) |
| Apache Recovery Kit | Eval | 27 Mar 2013 (87 days) |
| SIOS Data Replication ARK | Eval | 27 Mar 2013 (87 days) |
| MySQL Recovery Kit | Eval | 27 Mar 2013 (87 days) |

# Start the LifeKeeper for Linux

1. Start:

```
# /opt/LifeKeeper/bin/lkstart
```

# 12.7.8. Configure the Cluster – Apache

## Primary Server

Complete the following steps on the primary server to configure the cluster:

- Create TCP Communication (Comm) Path(s)
- Verify the Communication (Comm) Path(s)

Before you begin, SIOS recommends at least two TCP communications paths between each server within the cluster to each remote server for heartbeat redundancy.

> **Important:** Supported configurations require that you define redundant comm. paths, so that the failure of a single communication line will not cause a split brain where resource hierarchies may come in-service on multiple servers simultaneously.

### Access the LikeKeeper GUI

The LifeKeeper Graphical User Interface (GUI) is a Java based application that can be run as a native Linux application.

The LifeKeeper GUI is based on Java RMI with callbacks. Hostnames must be resolvable or you may receive a Java 115 or 116 error.

1. Verify that both short and fully qualified hostnames of all cluster nodes resolve to the proper locations

   # ping LinuxPrimary

   # ping LinuxPrimary.domain.com

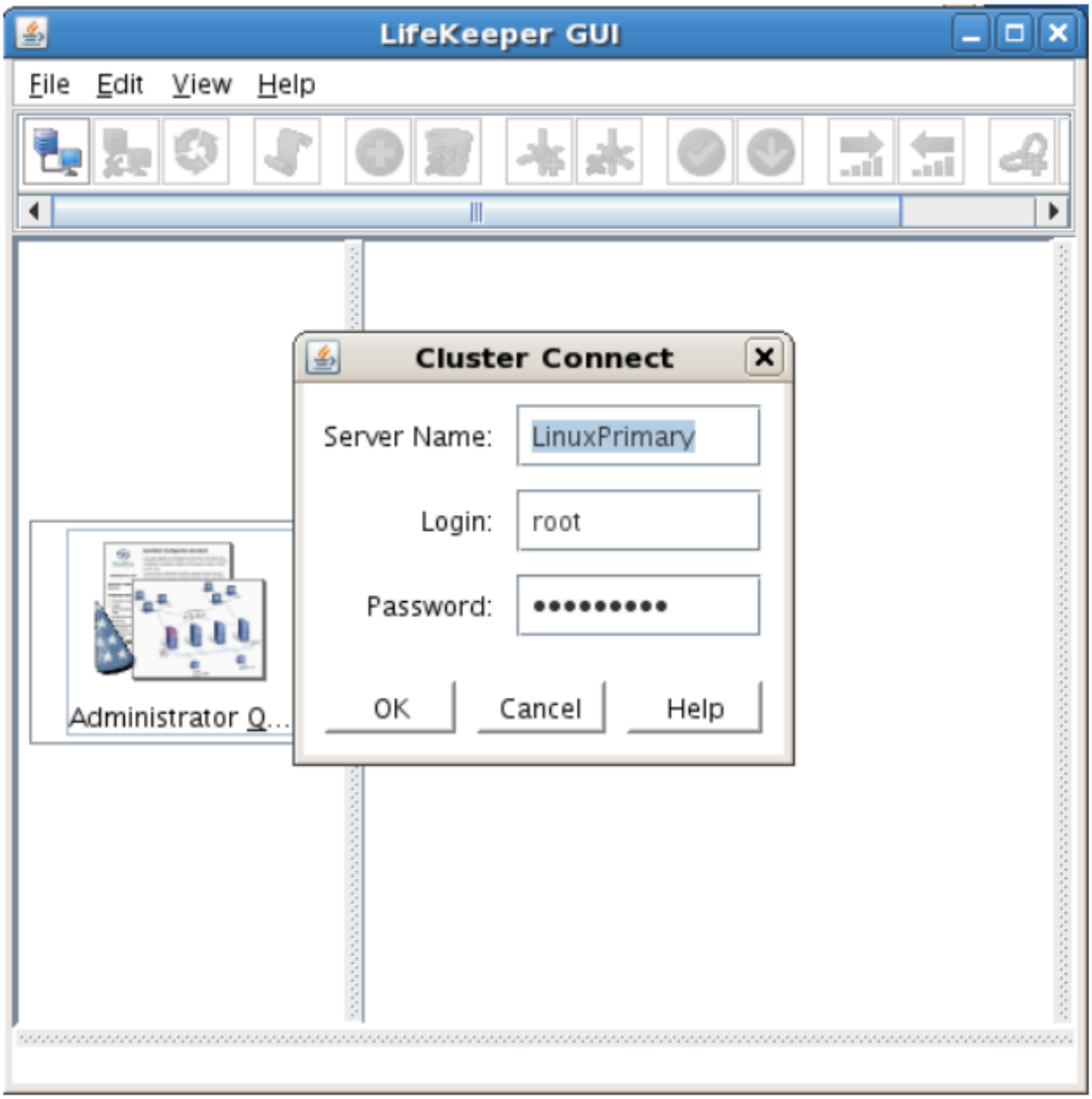
   # ping LinuxSecondary

   # ping LinuxSecondary.domain.com

2. To start the LifeKeeper Linux GUI Application:

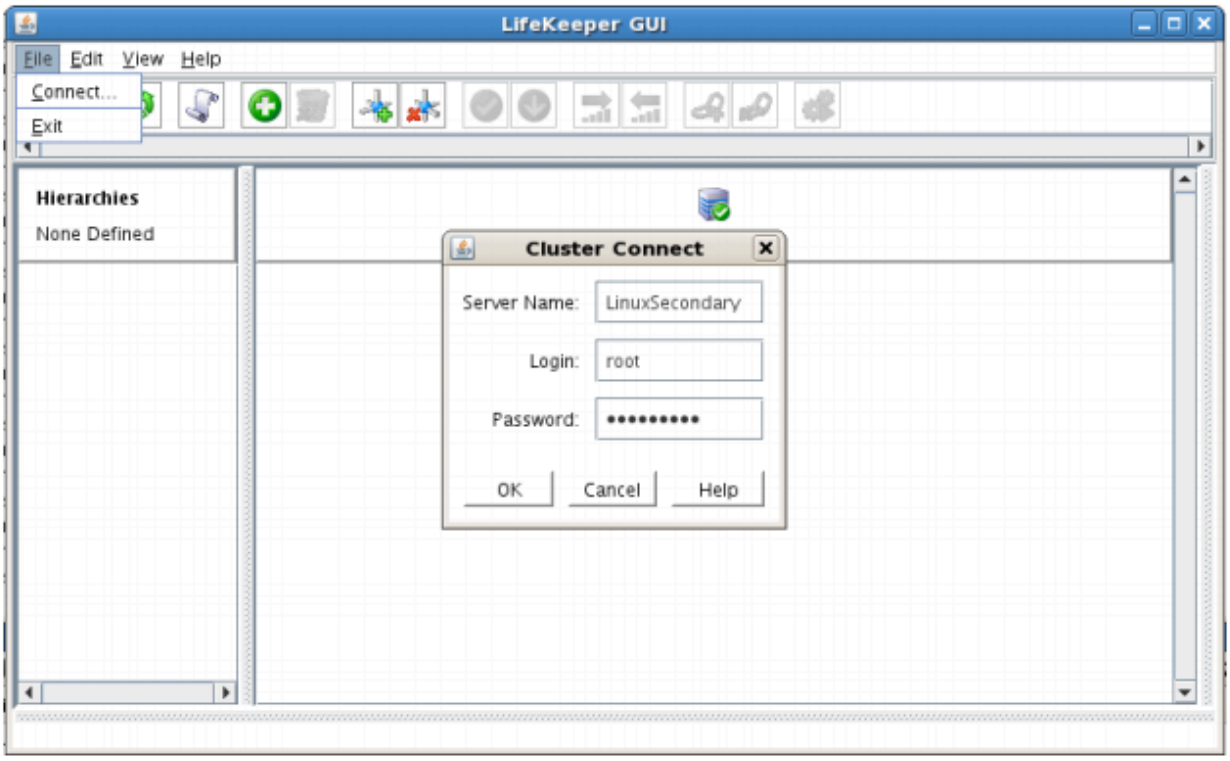
   a. /opt/LifeKeeper/bin/lkGUIapp &

3. Enter the name of the server you wish to connect to (this field will be populated with the name of the server you are on, if you are running the GUI from a server with LifeKeeper installed) along with your root credentials and click OK.



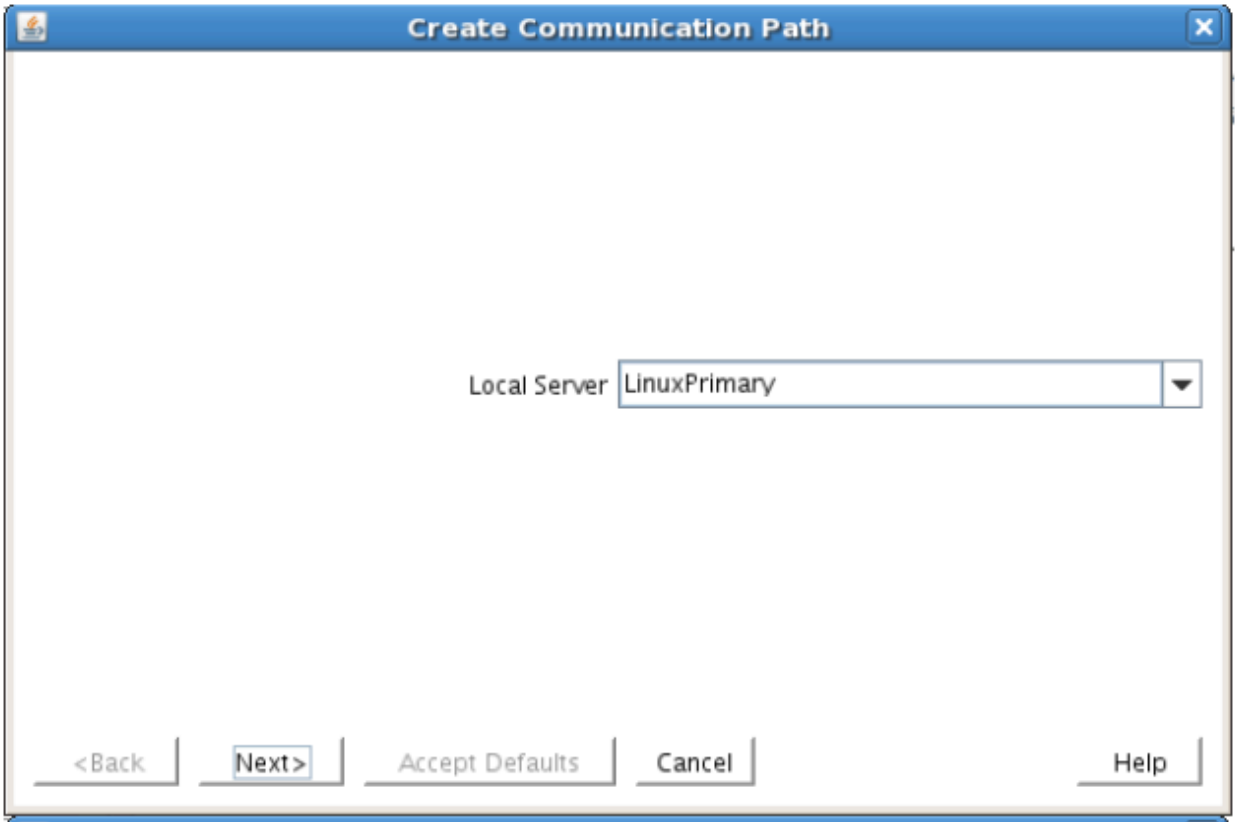

## Create Communication (Comm) Paths

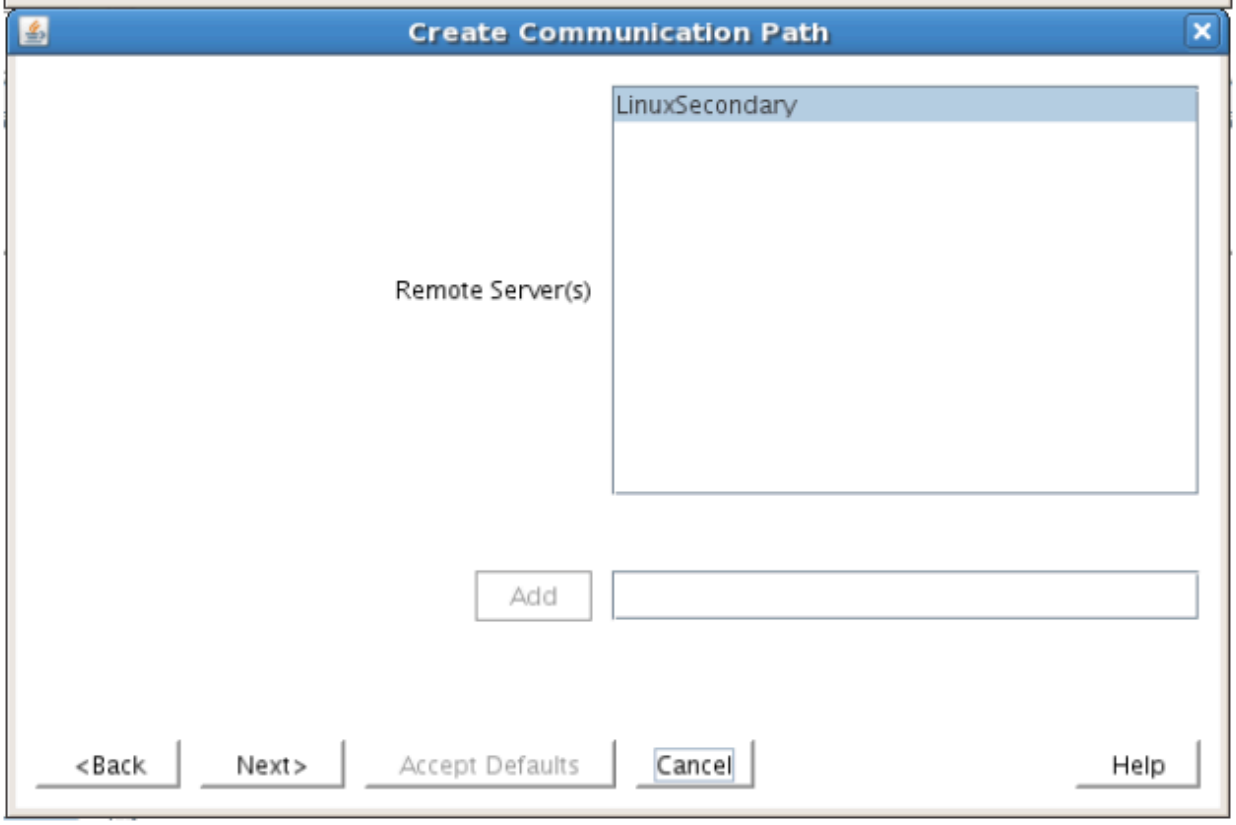
4. Within the LifeKeeper GUI, from the File menu, select Connect. Enter the name of your Secondary server, login and password when the Cluster Connect window displays.



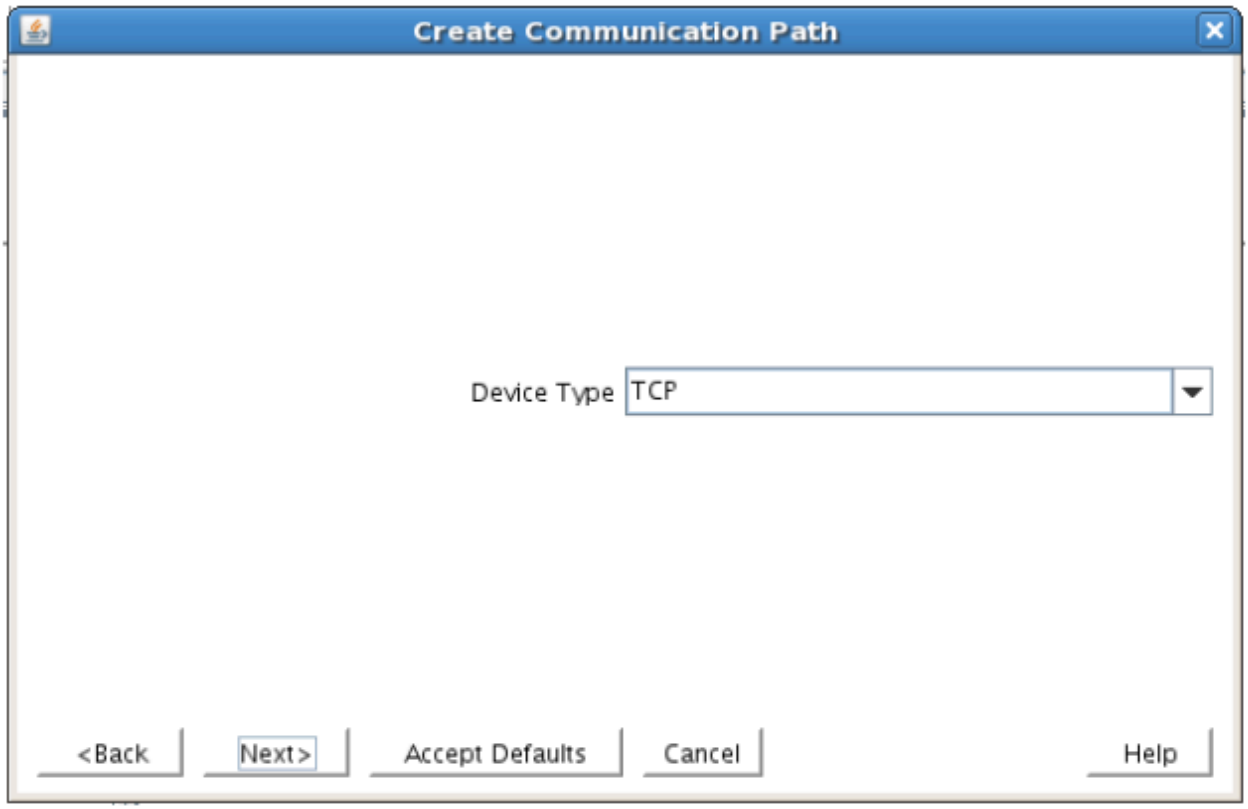

5. Within the LifeKeeper GUI, click the Create Comm Path button on the toolbar. You can also right click one of the servers and click Create Comm Path from the pop-up menu as well.

6. Select your Local and Remote Server(s) from the list box. If a server does not appear in the list box, you may enter it by typing its name and clicking the Add Server button. When using the Add Server procedure, you must make sure that the computer names for both network interfaces on the servers respond correctly when you ping them (from all of the partner server(s)) using the **ping –a IP ADDRESS** syntax. If they do not, this must be corrected prior to continuing. Click Next.

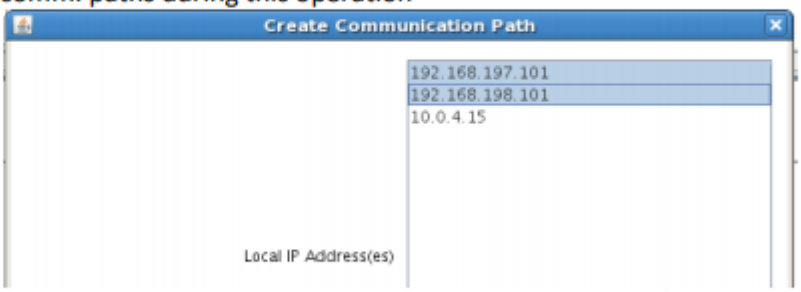


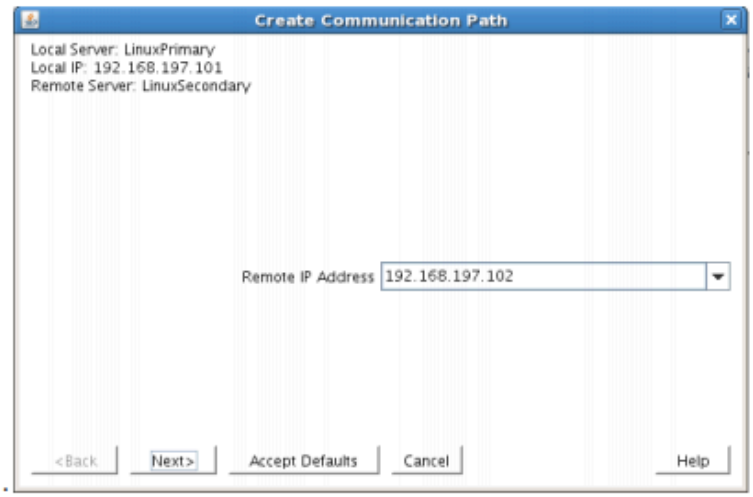




7. Select TCP for Device Type and Click Next.



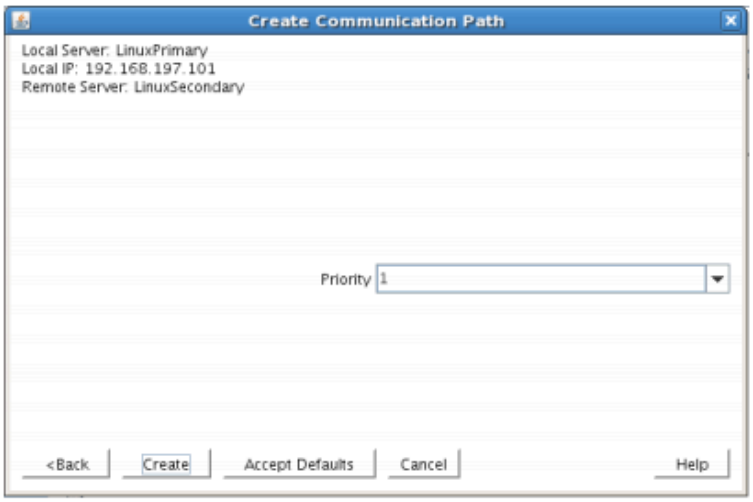

8. Provide all the required information and click Next for the following series of dialog boxes. For each field in the dialog box you can click Help for further information or refer to the table below for an explanation or recommendation

| Field | Tips |
|---|---|
| ***For TCP/IP Comm Path...*** | |
| Local IP Address | Choose the IP address to be used by the local server for this comm path. Select both interfaces so that the wizard creates multiple comm. paths during this operation   |
| Remote IP Address | Choose the IP address to be used by the remote server for this comm path |

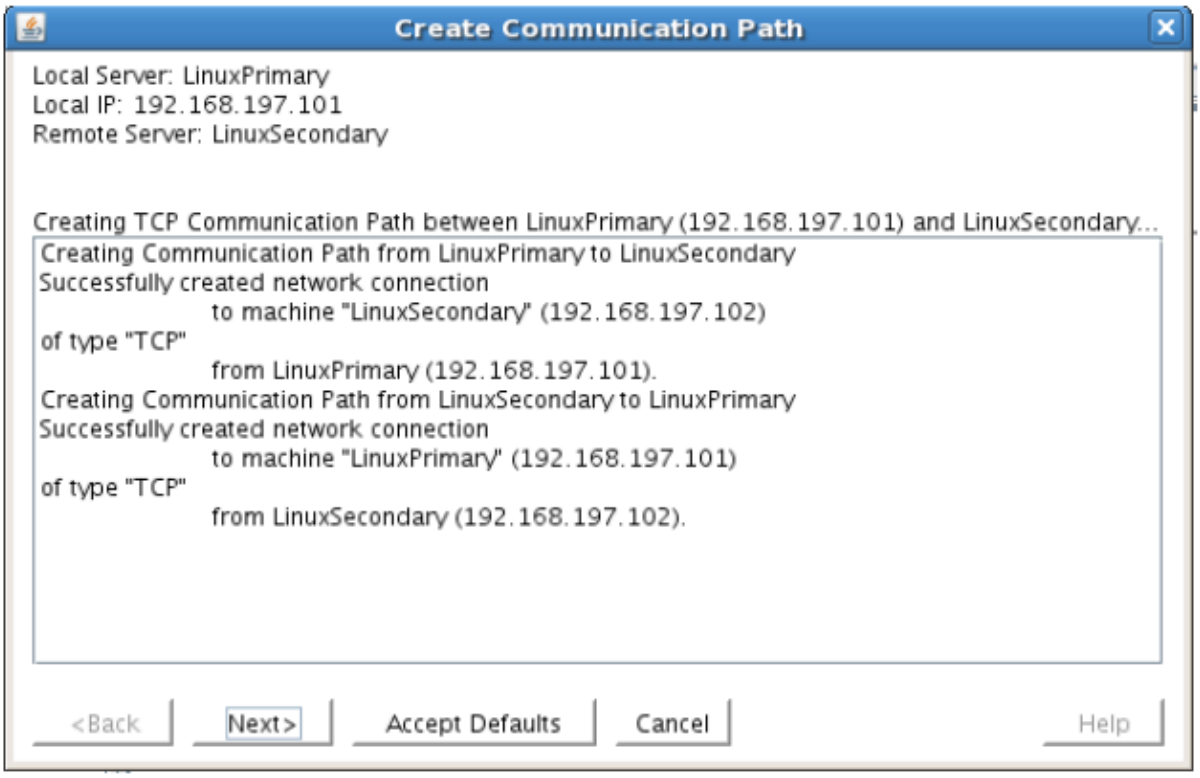



Enter the priority for the comm path on the local server. The priority will be used to determine the order that the comm paths between two servers will be used. Priority 1 is the highest; 99 is the lowest.

Priority




9. After entering data in all the required fields, select Create. A message will display indicating the network communication path is successfully created. Click Next.



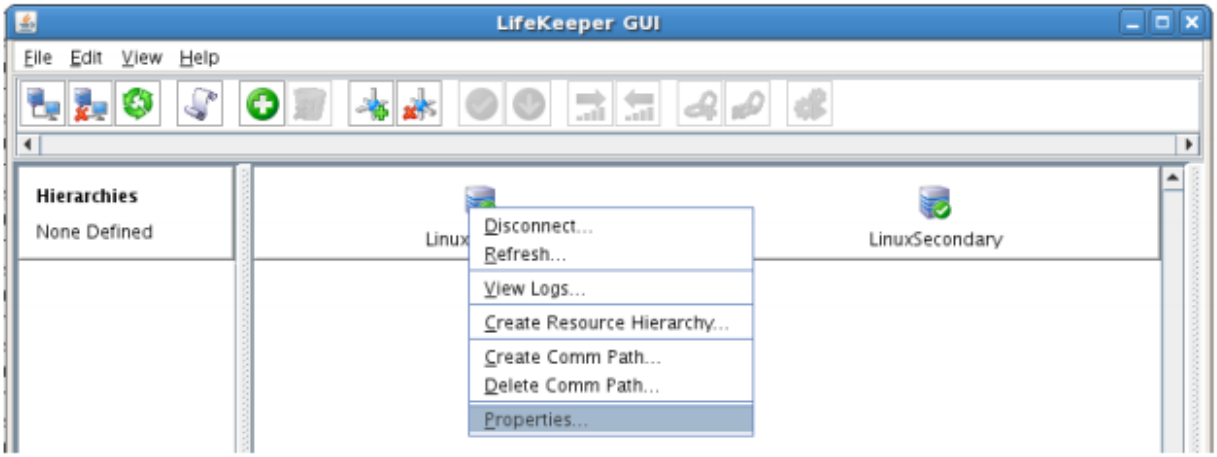

If you selected multiple Local IP Addresses or multiple Remote Servers and the Device Type was set to TCP, then the procedure will return you to the setup wizard the next Comm Path.

10. Select Done in the last dialog box.

    Repeat this process until you have defined all the communication paths you plan to use. SIOS strongly recommends that you define at least two communication paths for redundancy.

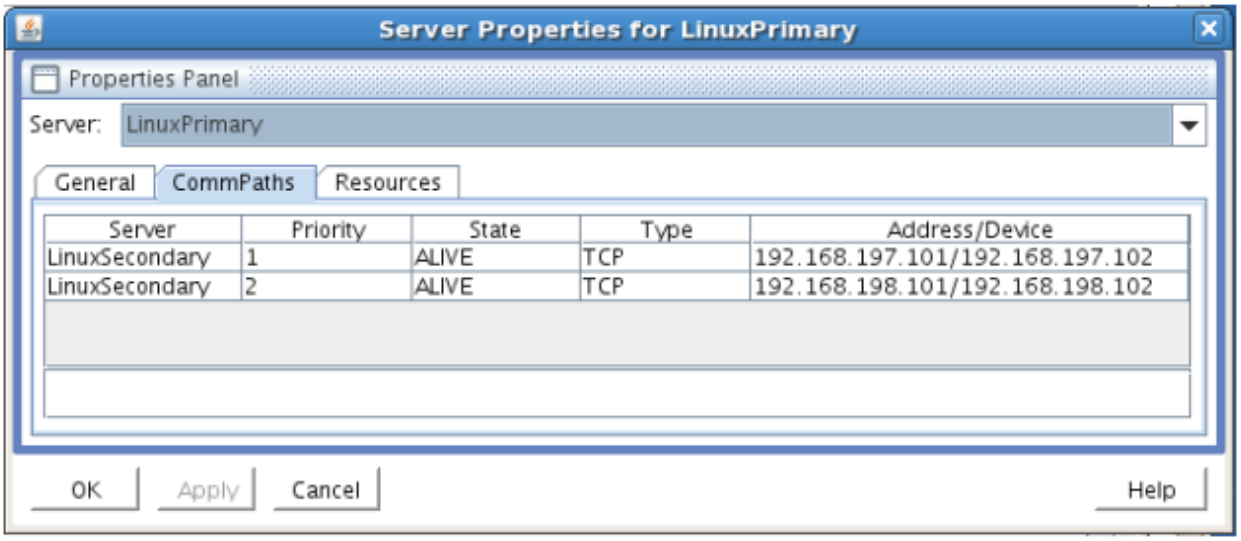
## Verify the Communications Paths

1. Verify that the communications paths are configured properly by viewing the Server Properties dialog box. From the LK GUI, select Edit, Server, Properties and then the Comm Paths tab.




2. Note the State displayed is ALIVE. You can also check the server icon in the right, main pane of

the GUI. If only one comm path has been created, the server icon shows a yellow warning icon on the server icon, indicating that one comm. path is ALIVE, but there is no redundant comm path. The server icon will display a green heartbeat checkmark when there are at least two comm paths configured and ALIVE.

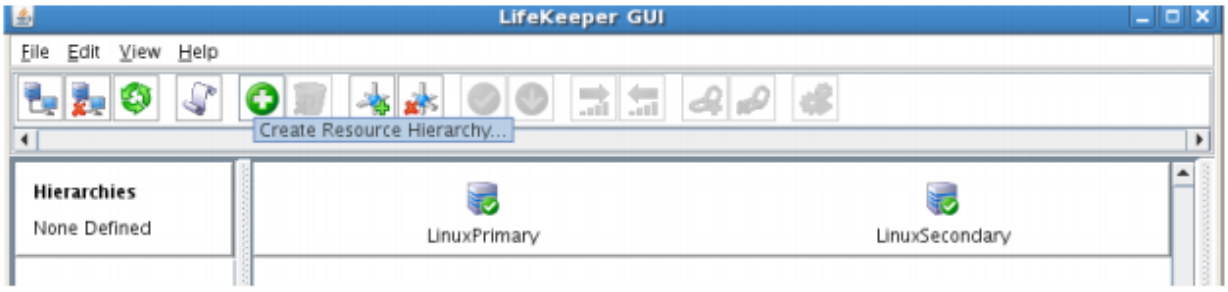



# Create the LifeKeeper Hierarchy

## Create and Extend an IP Resource

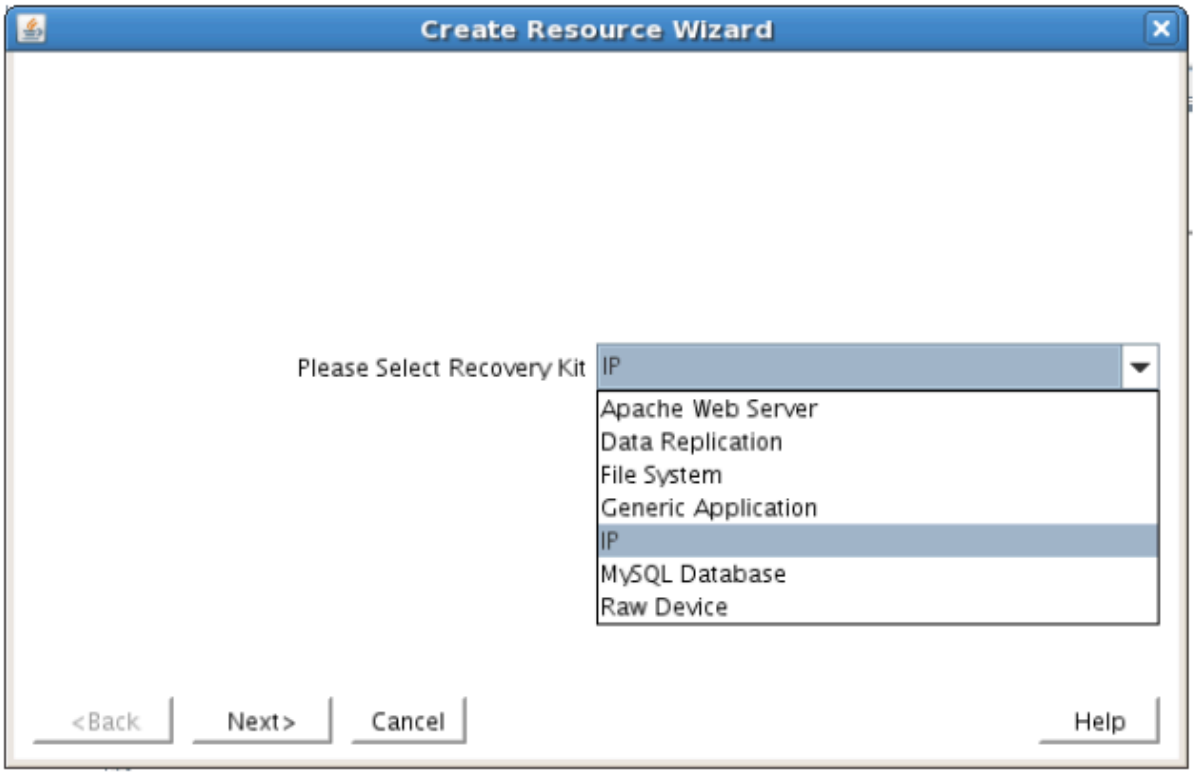
In LifeKeeper, create an IP resource and extend it to the secondary server by completing the following steps. This Virtual IP will have the ability to move between cluster nodes along the application that depends on it.

1. From the LifeKeeper GUI toolbar, click Create Resource Hierarchy.




The Create Resource Wizard dialog box will appear with a drop down list box displaying all recognized Recovery Kits installed within the cluster.

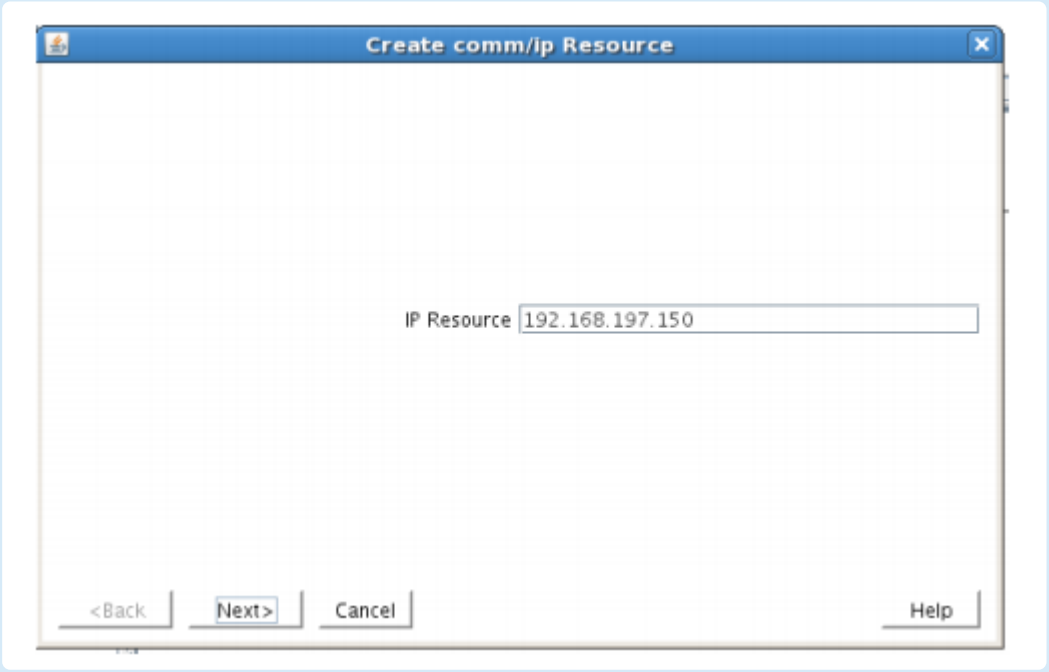
2. Select IP Address and click Next.

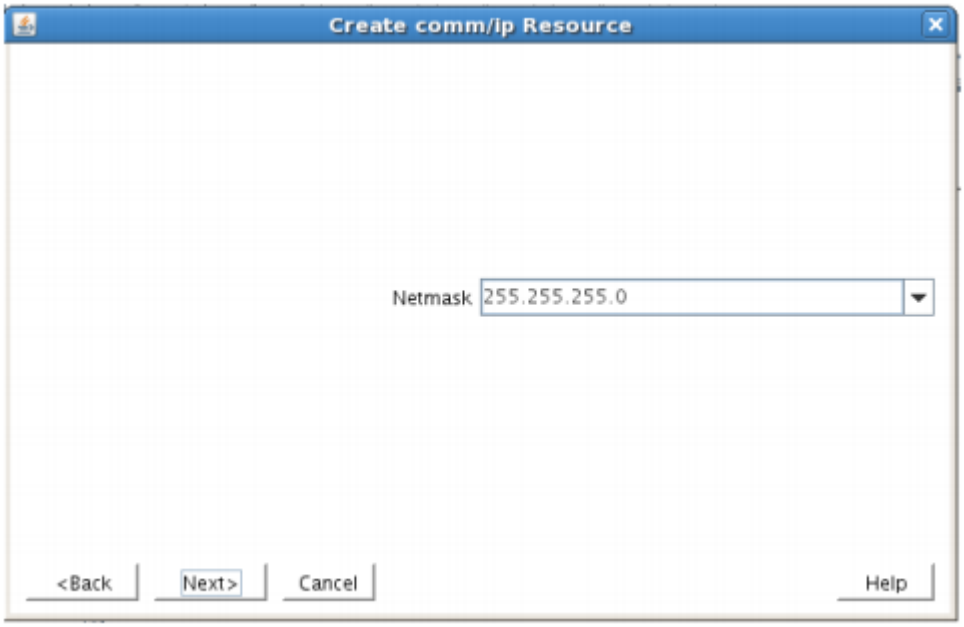



3. Enter the appropriate information for your configuration. The table below contains a list of the fields that display and additional information to assist you as you complete this procedure. Recommended values are also show below. You can also click the Help button for further information. Press Next to continue after entering the required information.

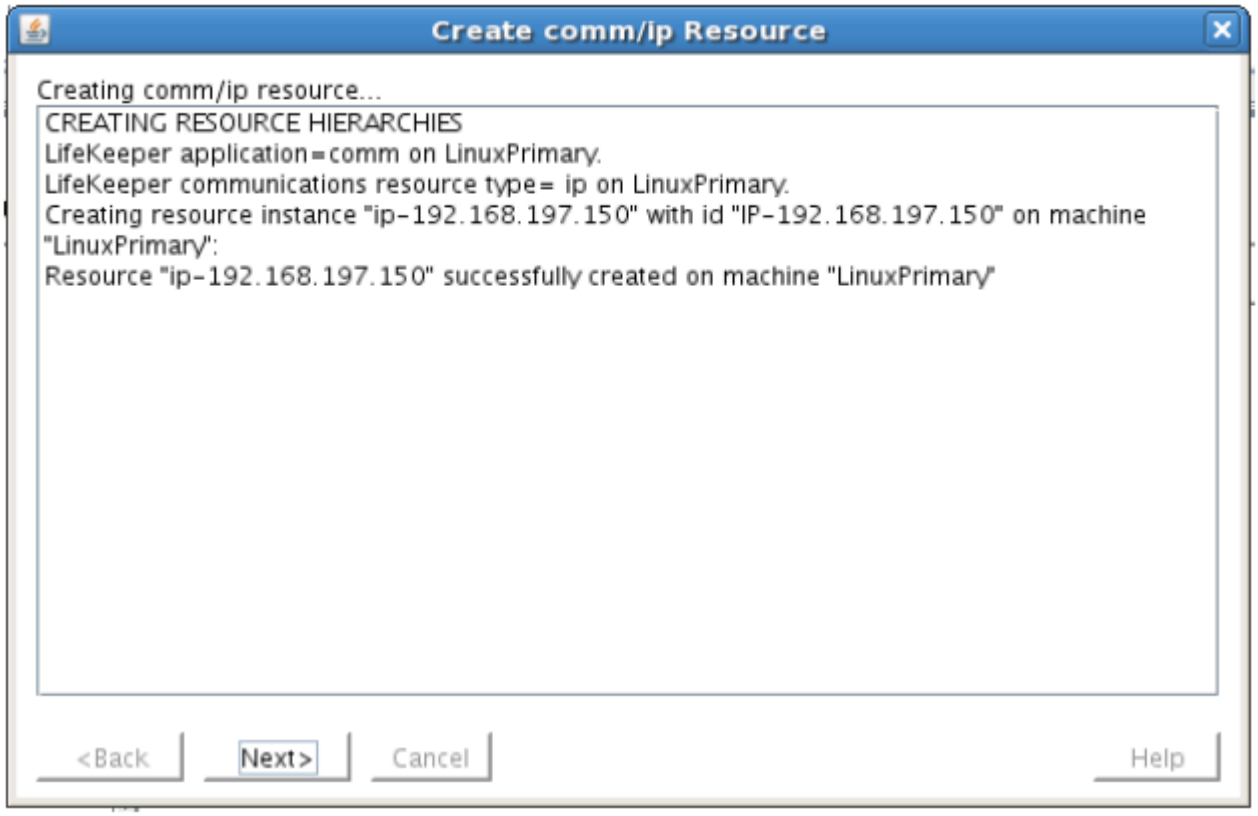
## IP Creation Field Definitions

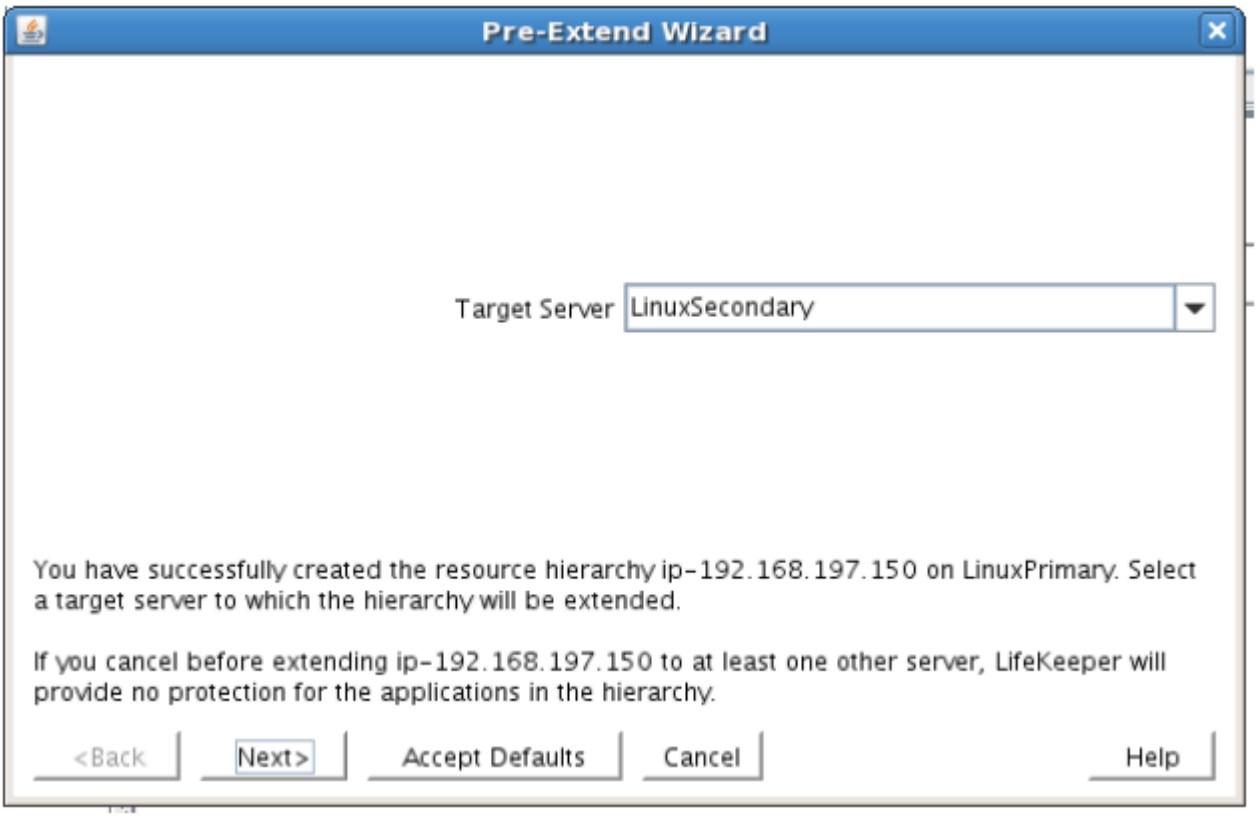
| Field | Tips |
|---|---|
| Resource Type | Select IP Address as the resource type and click Next. |
| Switchback Type | Select Intelligent and click Next. |
| Server | Select the Server where the IP resource will be created. Select your Primary server and click Next. |
| IP Resource | Enter the virtual IP information and click Next<br><br>Example **192.168.167.151**<br><br>**Note** This is an IP address that is not currently in use anywhere on your network. This is the address that all clients will use to connect to the protected resources.<br><br>In this configuration example, we will be protecting two (2) virtual IPs.<br><br>First we will protect 192.168.197.150, which our Apache webserver will |

| | |
|---|---|
| | use.<br>  <br><br>The second time through this wizard we will protect 192.168.197.151, which will be used by MySQL |
| Netmask | The IP subnet mask that your TCP/IP resource will use on the target server. Any standard netmask for the class of the specific TCP/IP resource address is valid<br><br>In our sample configuration 255.255.255.0 is used for a subnet mask on both networks.<br><br>**Note:** The subnet mask you choose, combined with the IP address, determines the subnet that will be used by the TCP/IP resource and should be consistent with the network configuration. |

| | |
|---|---|
| |  |
| Network Connection | This is the physical Ethernet card that the IP address is interfacing with. Chose the network connection that will allow your virtual IP address to be routable. Select the correct NIC and click Next. |
| IP Resource Tag | Accept the default value and click Next. This value only affects how the IP is displayed in the GUI. The IP resource will be created on our Primary server. |

4. LifeKeeper will create and validate your resource. After receiving the message that the resource has been created successfully, click Next when the following dialog box appears so that you can complete the process of Extending the IP Resource to our Secondary server, below.




Extending the IP resource will start automatically after you have finished creating an IP address resource if you clicked Next in the dialog box displayed above. You can also start this from an existing IP address resource by right clicking on the active resource and selecting Extend Resource Hierarchy.

Refer to the table below to complete the Extend IP Resource procedure.

| Field | Recommended Entries or Notes |
|---|---|
| Switchback Type | Leave as “intelligent” and click Next |
| Template Priority | Leave as default (1) |
| Target Priority | Leave as default (10) |
| Network Interface | This is the physical Ethernet card that the IP address is interfacing with. Chose the network connection that will allow your virtual IP address to be routable. The correct physical NIC should be selected by default. Please verify and then click Next |
| IP Resource Tag | Leave as default. |
| Target Restore Mode | Select Enable and click Next. |
| Target Local Recovery | Select Yes to enable Local Recovery for the SQL resource on the Target server. |
| Backup Priority | Accept the default value. |

5. After receiving the message Hierarchy extend operations completed, click Finish and then click Done

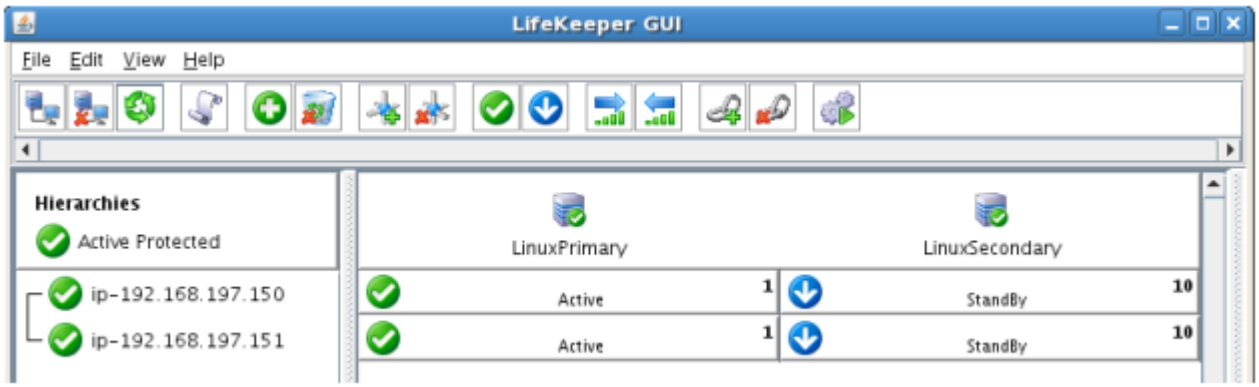
6. Your IP resource (192.168.197.151) is now fully protected and has the ability to “float” between cluster nodes as needed. Looking at the LifeKeeper GUI you will notice that the IP resource is Active on the Primary cluster node and Standby on the Secondary cluster node

## Create a Second IP Resource

Repeat the procedure above to protect a 2nd IP resource.

This second time, protect 192.168.197.151, which is the IP address our MySQL database will later use.

As a result, your LifeKeeper GUI will display as follows, with both IP resources Active and protected on the Primary cluster node:



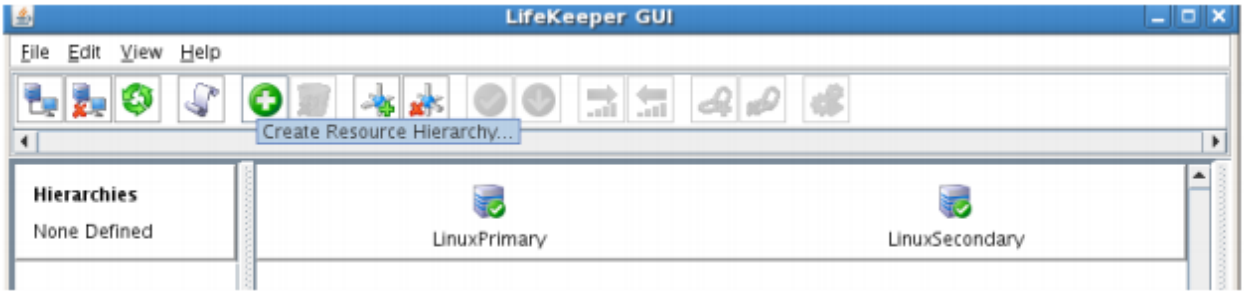

## Create a Mirror and Begin Data Replication

In this section we will setup and configure the Data Replication resource, which be used to synchronize our Apache Webserver's data between cluster nodes. The data we will replicate resides in the /var/www partition on our Primary cluster node

Please note:

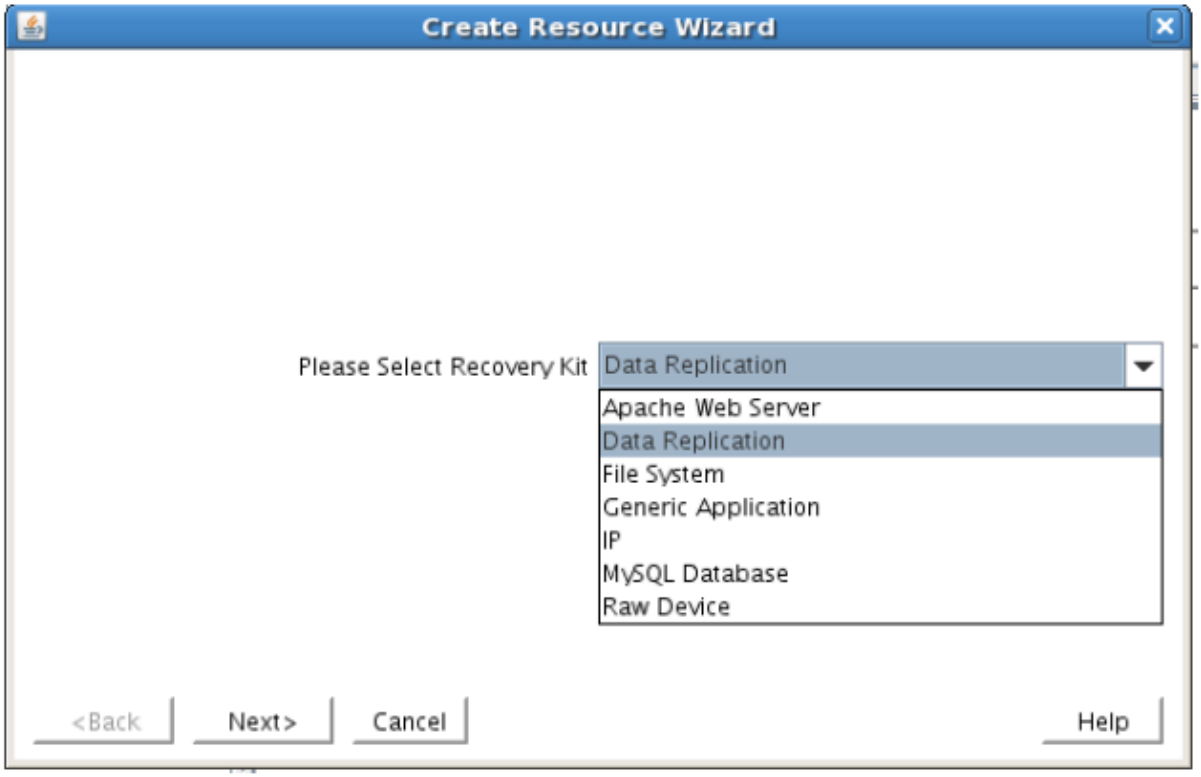
- The source volume to be replicated must be mounted on the Primary server
- The target volume, which will received replicated data, must NOT be mounted on the Secondaryserver.
- The target volume's size must equal to or larger than the size of its source volume.

1. From the LifeKeeper GUI toolbar, click Create Resource Hierarchy.




The Create Resource Wizard dialog box will appear with a drop down list box displaying all recognized Recovery Kits installed within the cluster.

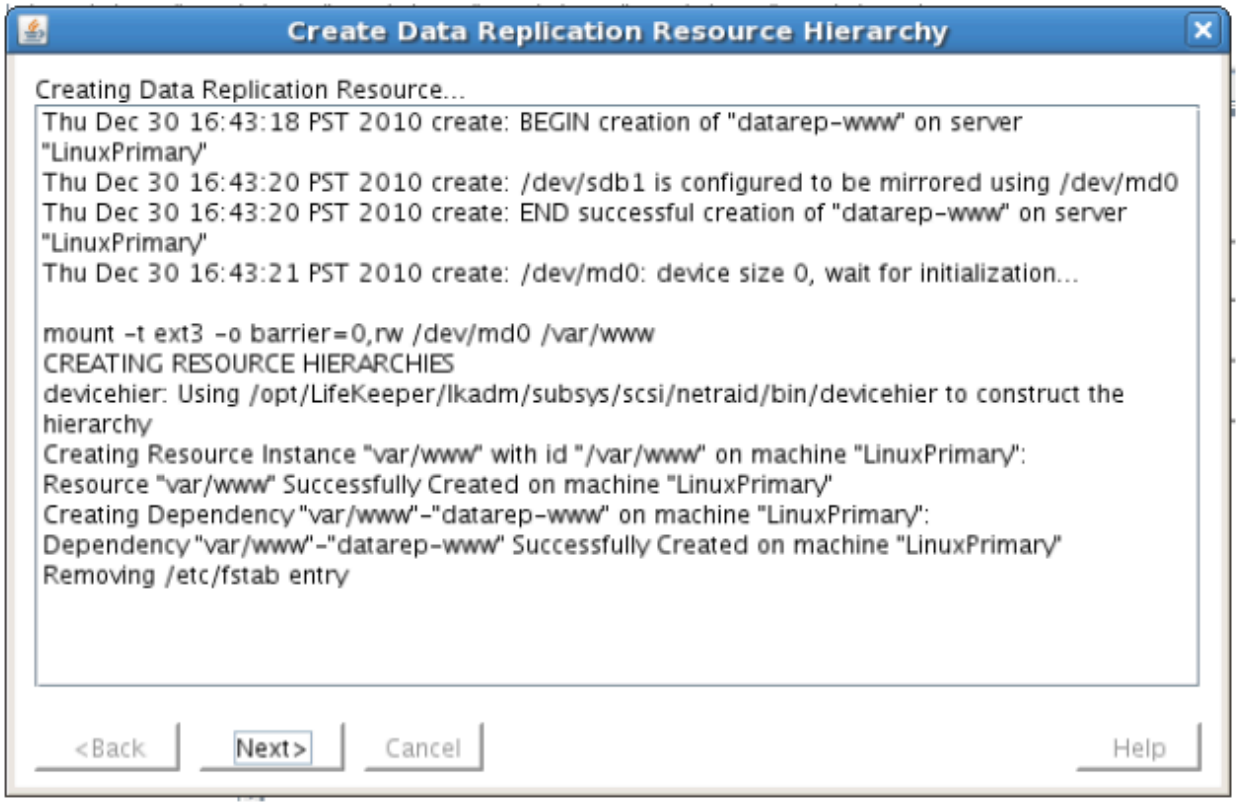
2. Select Data Replication and click Next.



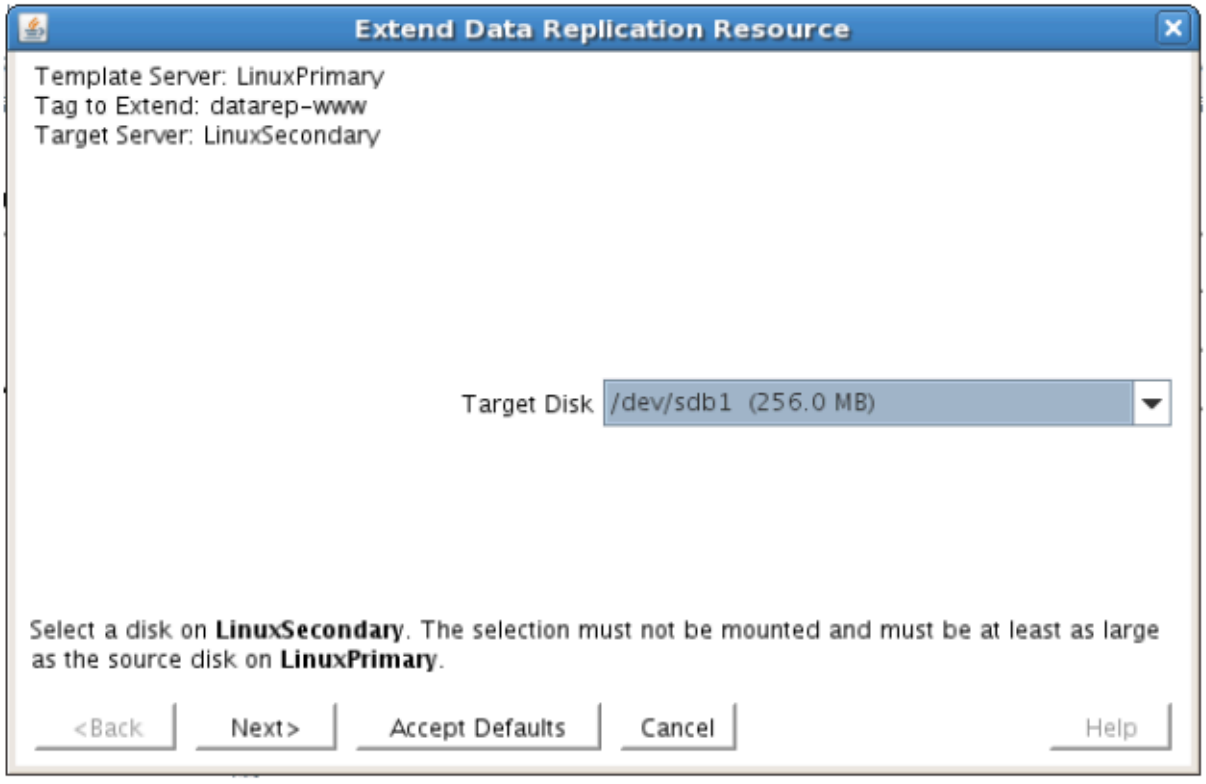

3. Follow the Data Replication wizard, and enter the following values:

| Field | Recommended Entries or Notes |
|---|---|
| Switchback Type | Intelligent |
| Server | LinuxPrimary (Primary Cluster Node, i.e. Mirror Source) |
| Hierarchy Type | Select: “**Replicate Existing Filesystem**” |
| Existing Mount Point | At this step you will select the mounted partition to replicate. In our example, select “/var/lib/mysql” |
| Data Replication Resource Tag | Leave as default |
| File System Resource Tag | Leave as default |
| Bitmap File | Leave as default (Note: if using high speed SSD storage you will want to create a small partition and use it for bitmap placement, i.e. /bitmaps) |
| Enable Asynchronous Replication | Leave as default (Yes) |

4. Click Next to begin creation of the Data Replication resource hierarchy. Status will be displayed in the GUI as follows:

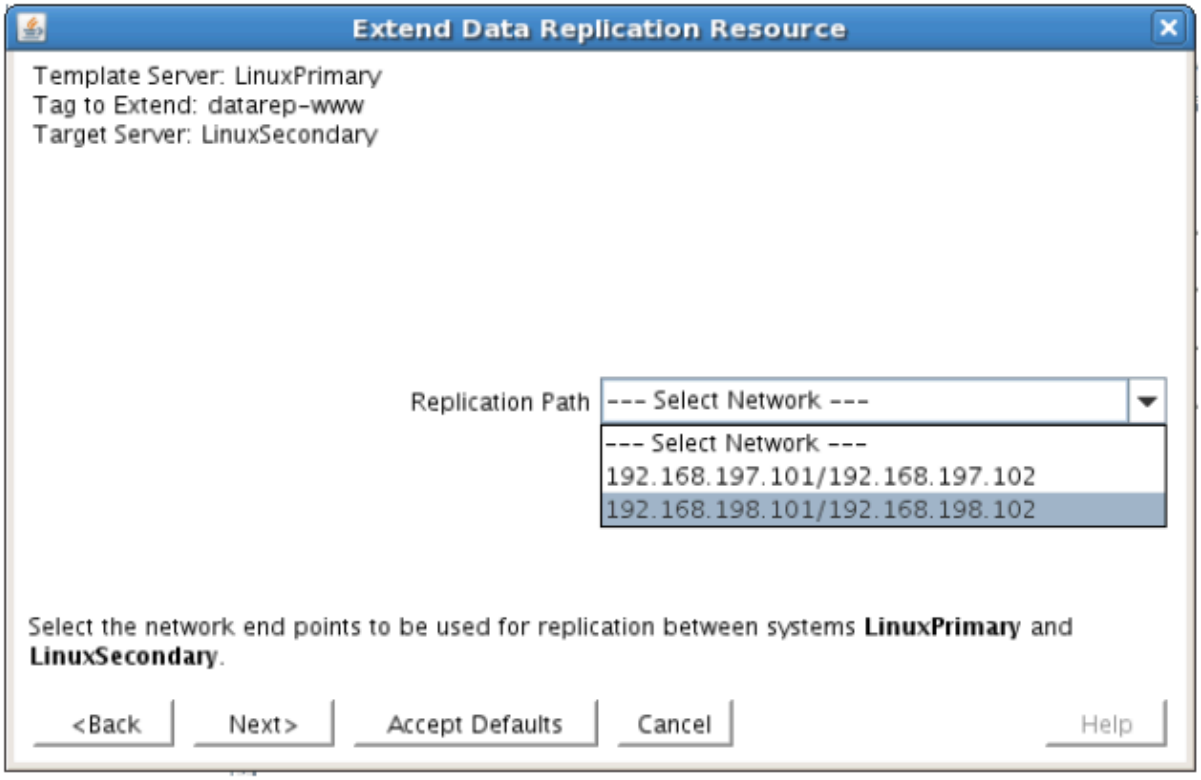



5. Click Next to begin the process to Extend the Data Replication Resource. Select all default settings. When it asks for the target disk, select a free partition on your Target server which is the same size (or greater) than the Source Volume we are replicating. This partition should NOT be mounted on the Target system.




6. Continue through the wizard, and you will be prompted to select the network you would like

replication to take place over. In general, it's a best practice to separate your user/application and your replication traffic. In our example setup we will replicate over our backend network, 192.168.198.X



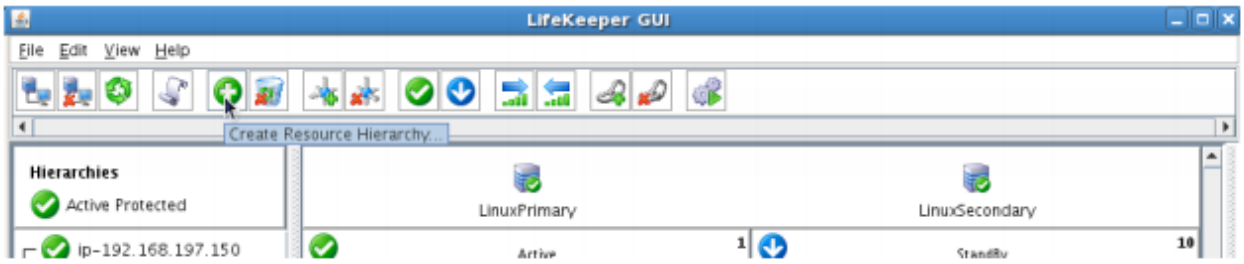

7. Click Next and continue through the wizard. Once completed, your resource hierarchy will look as follows

## Create the Apache Hierarchy

In this section we will create an Apache resource hierarchy on the primary server and extend it to the backup server. This step will create a dependency on the IP resource created in previous the step.

**Important:** The Apache web server should not be running at this time.

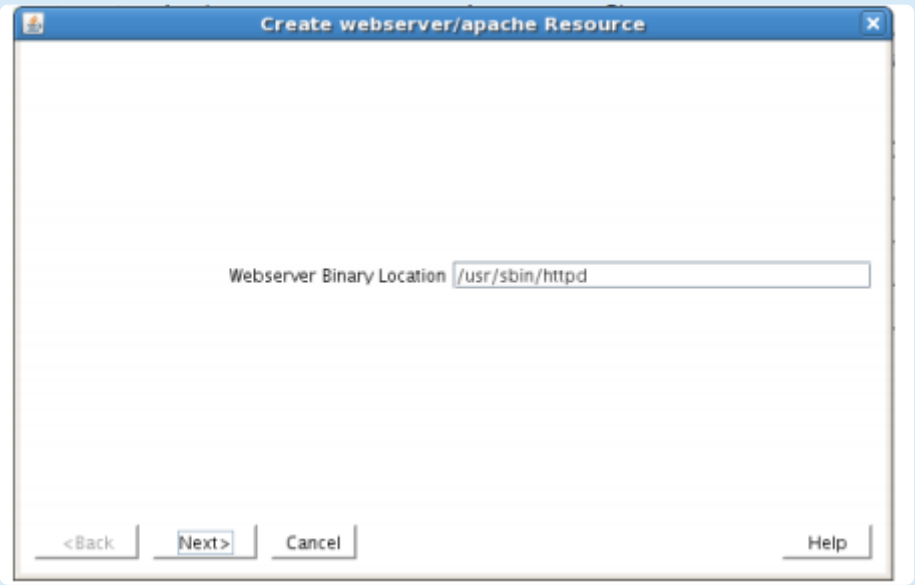
1. From the LifeKeeper GUI toolbar, click Create Resource Hierarchy.

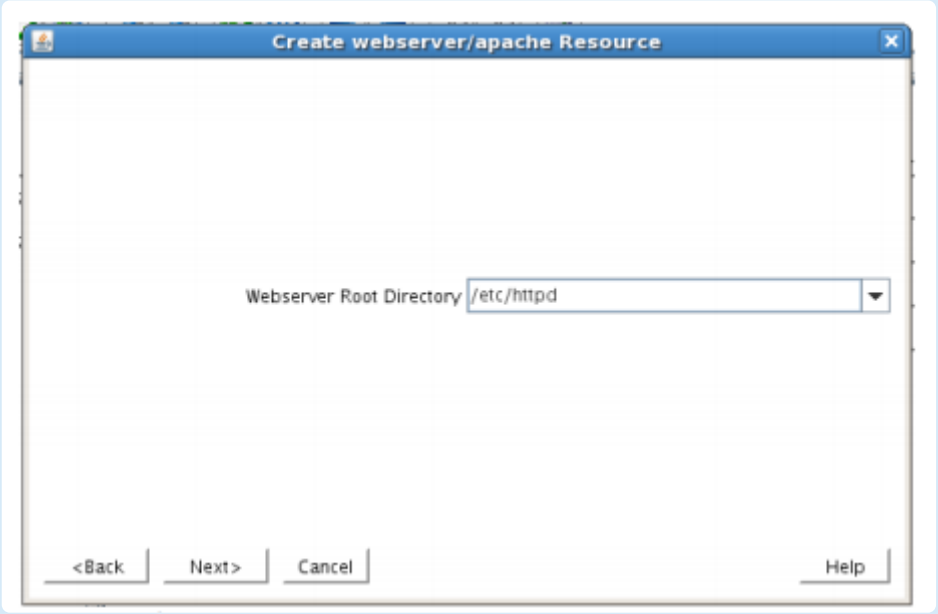



The Create Resource Wizard dialog box will appear with a drop down list box displaying all recognized Recovery Kits installed within the cluster.

2. Select Apache Web Server and click Next.

3. Proceed Through the Resource Creation wizard, providing the following values

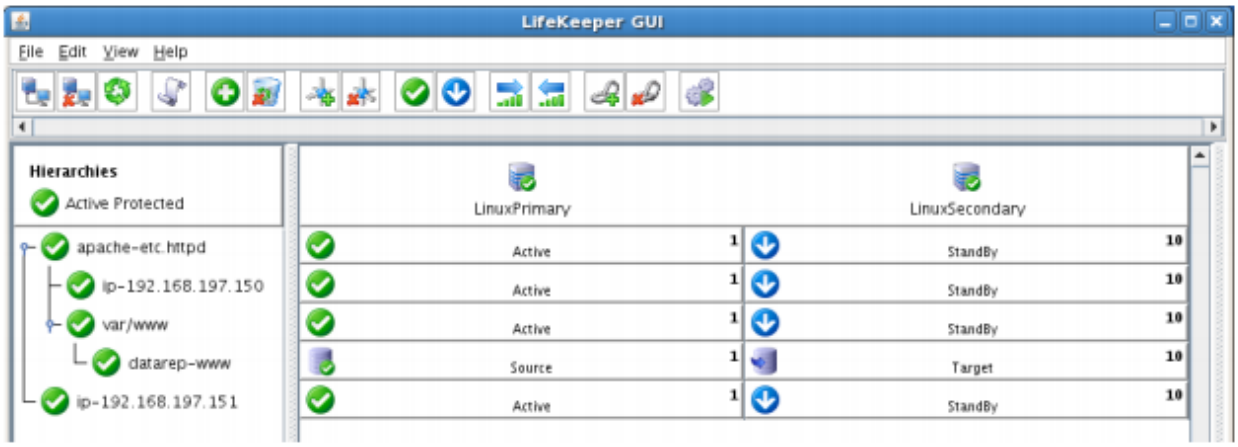
| Field | Recommended Entries or Notes |
|---|---|
| Switchback Type | Intelligent |
| Server | LinuxPrimary (Primary Cluster Node, i.e. Mirror Source) |
| Webserver Binary Location | /usr/sbin/httpd (assumes a standard apache config)<br>  |
| Webserver Root Directory | /etc/httpd (assumes a standard apache config) |

| | |
|---|---|
| |  |
| Root Tag | Leave as default |

4. Click “Create” to begin resource hierarchy creation on the primary server. Once complete, click “Next” to extend this resource to the secondary server.

5. During the Extend Resource wizard, leave all settings as default.

6. Note: during the resource creation process, LifeKeeper extracted the existing configuration of the existing Apache webserver, and identified that it depends on the IP resource (192.168.197.150) and the Data Replication resource (/var/www) that were created in previous steps. These resources now appear underneath the newly created Apache resource, to indicate the dependency relationship.




## Create the Shared Filesystem Resource Hierarchy

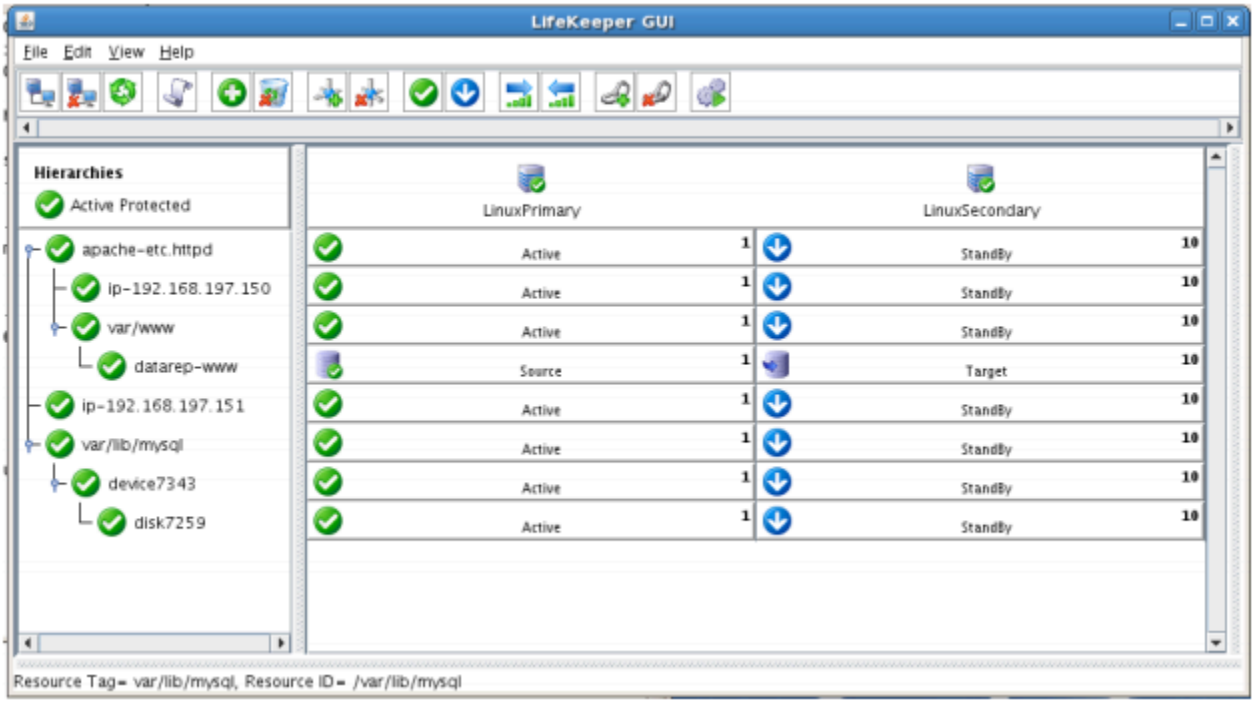
Create a Filesystem resource to protect the shared iSCSI filesystem and make it high available between cluster nodes. LifeKeeper for Linux leverages SCSI Persistent Group Reservations (PGR) to lock the LUN, ensuring that only the active cluster node for the storage resource can access it.

**Important:** At this point, the shared iSCSI LUN needs to already be mounted on the Primary Server. It should NOT be mounted on the Secondary Server. See section titled “Configure iSCSI initiator, discover and login to iSCSI target” above to review the steps involved.

1. From the LifeKeeper GUI toolbar, click Create Resource Hierarchy.
2. Select File System and click Next.
3. Proceed Through the Resource Creation wizard, providing the following values

| Field | Recommended Entries or Notes |
|---|---|
| Switchback Type | Intelligent |
| Server | LinuxPrimary (Primary Cluster Node) |
| Mount Point | Select /var/lib/mysql. Note that LifeKeeper scans the system for LUNS that are sharable between cluster nodes. The list of possible shared LUNS is presented automatically in this step of the wizard. |

4. Select Create Instance to define this resource hierarchy on the Primary Server
5. Click Next to Extend the File System Resource to the Secondary Server
6. In the Extend Wizard, select “Accept Defaults”
7. As a result the File System resource is now protected on both cluster nodes. Click Finish to exit the Extend wizard.
8. Your resource hierarchy should look as follows:



## Create the MySQL Resource Hierarchy

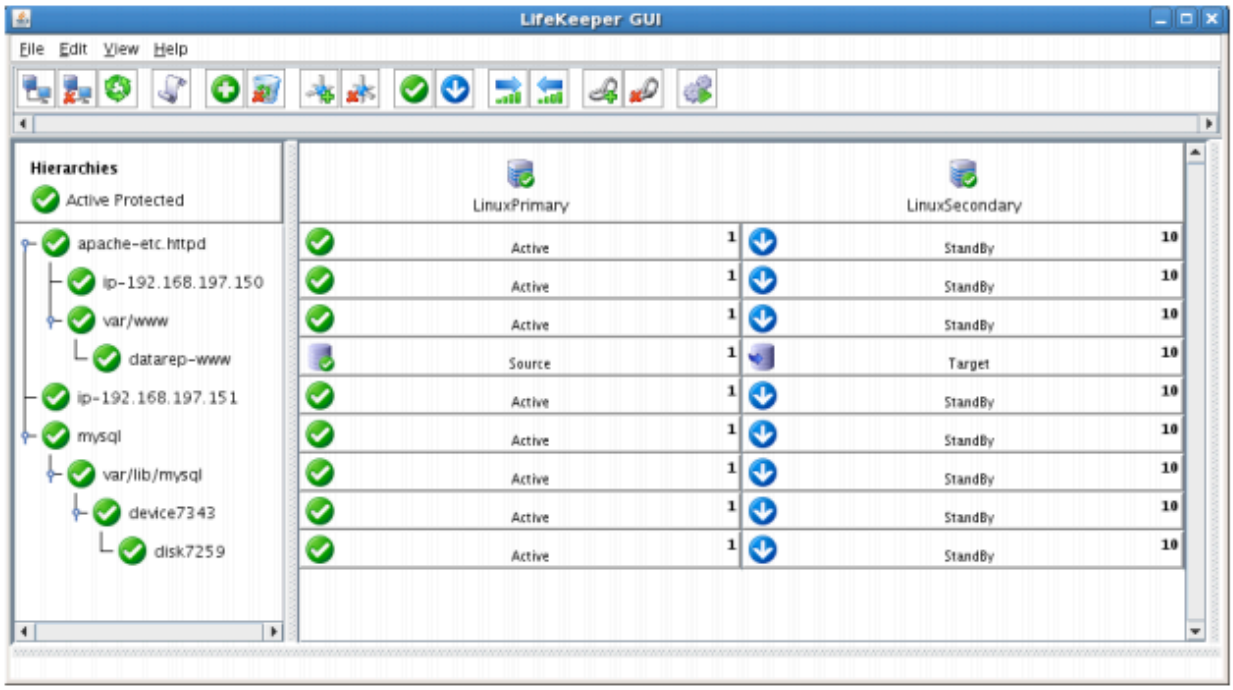
Create a MySQL resource to protect the MySQL database and make it high available between cluster nodes.

> **Important:** At this point, MySQL needs to be already running on the Primary Server. It should NOT be running on the Secondary Server. See section titled “Install, Configure, and Start MySQL” above to review the process to configure and start MySQL as needed.

1. From the LifeKeeper GUI toolbar, click Create Resource Hierarchy.
2. Select **MySQL Database** and click Next.
3. Proceed Through the Resource Creation wizard, providing the following values

| Field | Recommended Entries or Notes |
|---|---|
| Switchback Type | Intelligent |
| Server | LinuxPrimary (Primary Cluster Node) |
| Location of my.cnf | Enter “/var/lib/mysql”. Note that earlier in the MySQL configuration process we created a my.cnf file in this directory. |
| Location of MySQL executables | Leave as default (/usr/bin) since we are using a standard MySQL install/ configuration in this example |
| Database tag | Leave as default |

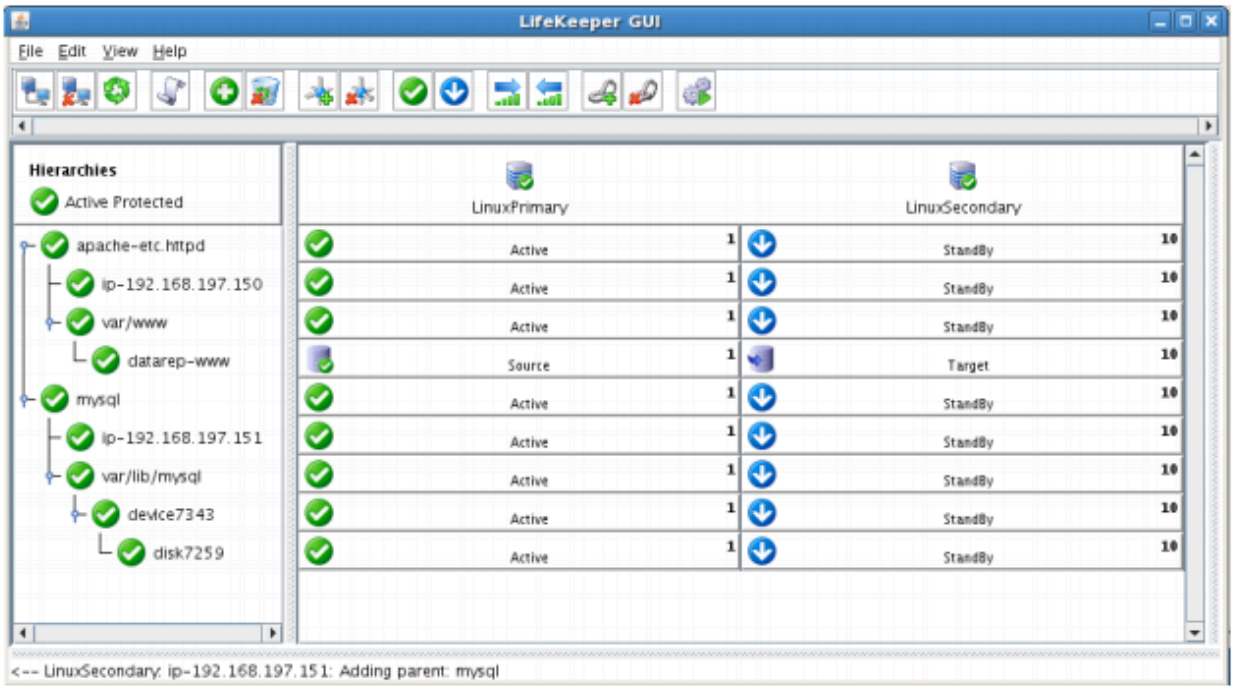
4. Select Create to define the MySQL resource hierarchy on the Primary Server
5. Click Next to Extend the File System Resource to the Secondary Server
6. In the Extend Wizard, select “Accept Defaults”
7. As a result the MySQL resource is now protected on both cluster nodes. Click Finish to exit the Extend wizard.
8. Note: LifeKeeper will automatically identify that the MySQL resource has a dependency on the FileSystem resource (/var/lib/mysql). The Filesystem Resource will appear underneath the MySQL resource in the GUI
9. Your resource hierarchy should look as follows:




## Create the MySQL IP Address Dependency

In this step will define an additional dependency: that MySQL depends on a Virtual IP (192.168.197.151) so that the IP address follows the MySQL database should it move. This IP (.151) is the IP the webserver will use to access the MySQL database.

1. From the LifeKeeper GUI toolbar, right-click on the “mysql” resource
2. Select “Create Dependency” from the right-click context menu
3. In the Child Resource Tag dropdown menu, select “ip-192.168.197.151”
4. Click Next
5. Click Create Dependency
6. Click Done
7. The Virtual IP address resource (192.168.197.151) will now appear underneath the MySQL resource in the LifeKeeper user interface. This ensures that resources move together, and are started/stopped in the proper order.
8. Your resource hierarchy should look as follows




At this point in the Evaluation, we have fully protected Apache, MySQL, and their dependent resources: IP addresses, and Storage, both shared and replicated.

# 12.7.9. Test Your Environment – Apache

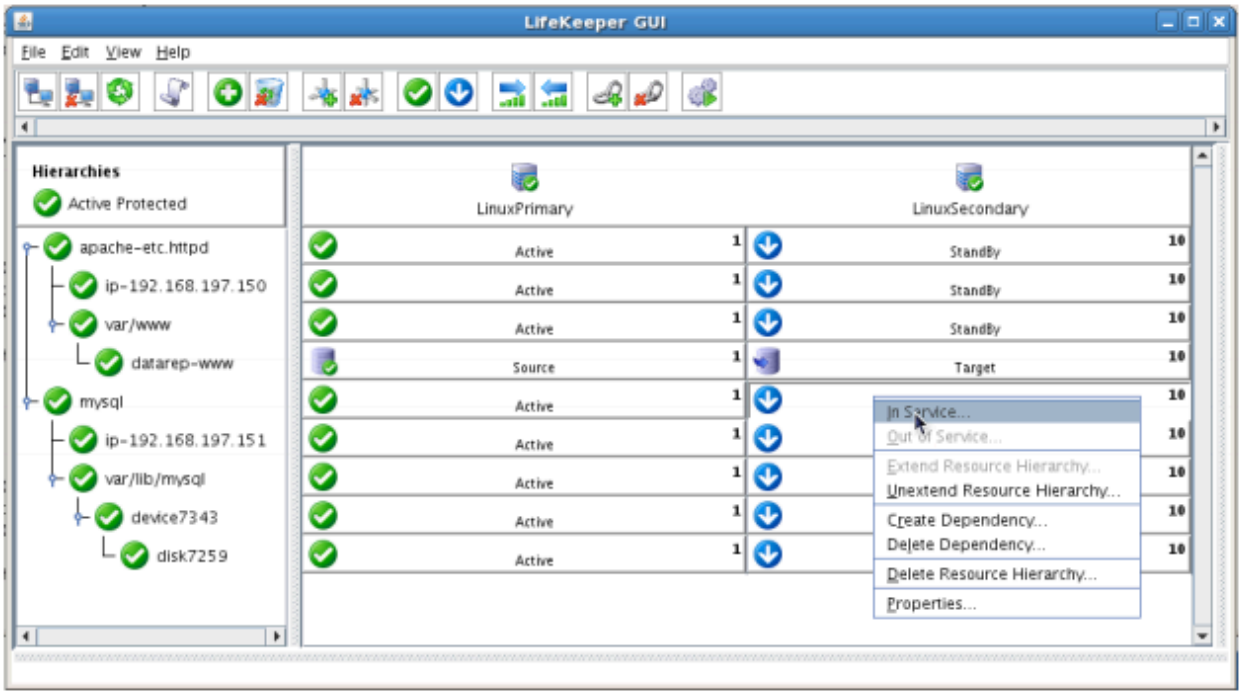
The following test scenarios have been included to guide you as you get started evaluating LifeKeeper for Linux. **Before beginning these tests, make sure the data replication resources are in the mirroring state**.

> **Note:** For these test examples, the Primary Server is referred to as LINUXPRIMARY. The Backup or Secondary Server is referred to as LINUXSECONDARY.

## Manual Switchover of the MySQL Hierarchy to Secondary Server

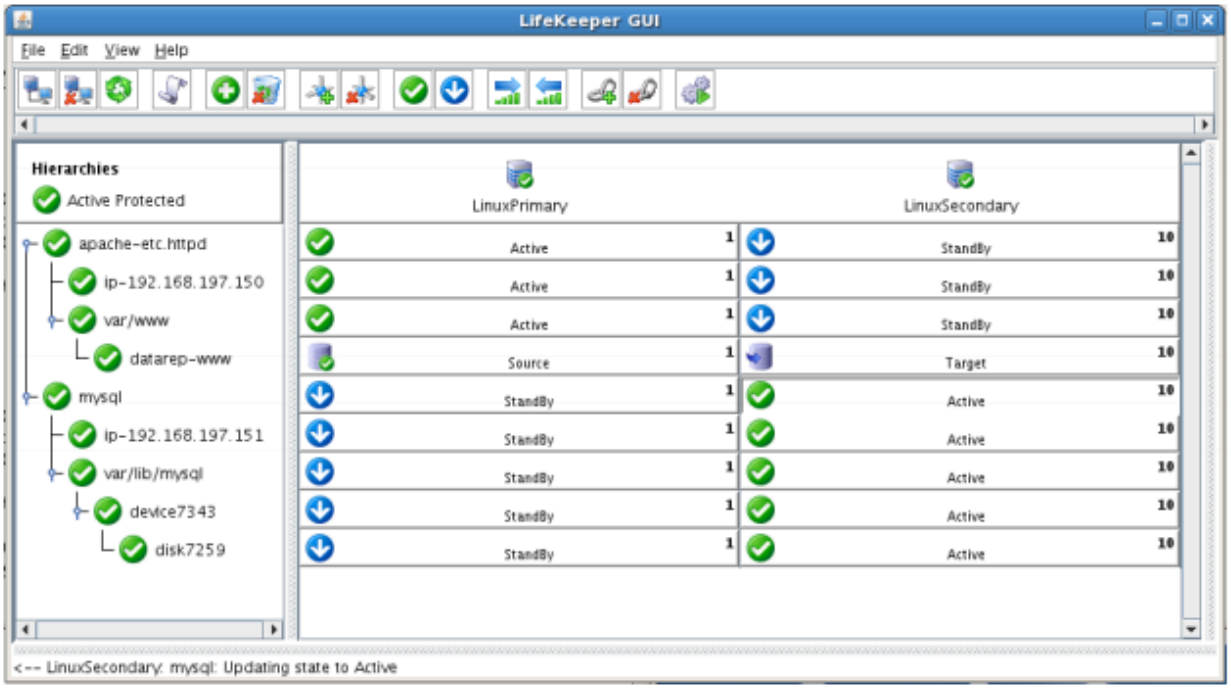
**Procedure:**

- From the LifeKeeper GUI, right click on the MySQL resource on the Secondary Server (LINUXSECONDARY) and choose IN SERVICE.
- Click “In Service” in the window that pops up




**Expected Result:**

- Beginning with the MySQL resource, all resources will be removed from service on the Active Server (LINUXPRIMARY).
- Beginning with the dependent resources (IP and Shared Volume), all resources will be brought in service on LINUXSECONDARY
- At this point, we now have an “Active/Active” cluster because both cluster nodes are actively running resources.



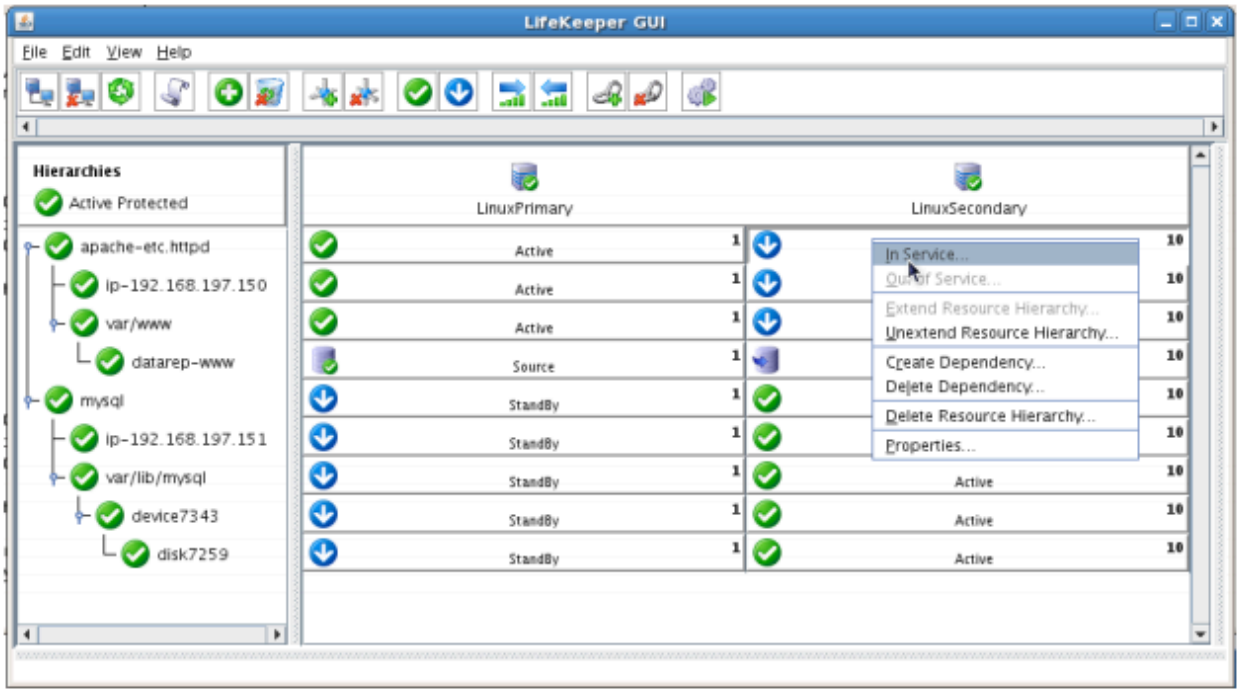

**Tests/Verification:**

- Using the LifeKeeper GUI, verify that the MySQL and dependent resources are active on LINUXSECONDARY.
- Run “ifconfig –a” on LINUXSECONDARY to validate that the IP Address 192.168.197.151 is active on LINUXSECONDARY
- Run “df –h” to verify that the /var/lib/mysql shared iSCSI filesystem is mounted on LINUXSECONDARY
- Verify the MySQL services are running on LINUXSECONDARY by running “ps –ef | grep –i mysql”
- On LINUXSECONDARY run the following command to verify client connectivity to the MySQL database:
  - # mysql –S /var/lib/mysql/mysql.sock –u root –p
  - (enter password “SteelEye”)
- Verify that the SCSI reservation has properly locked the iSCSI LUN to the currently active cluster node. On LINUXPRIMARY, run “mount /dev/sdc1 /var/lib/mysql”. This should FAIL because LINUXPRIMARY does not own the SCSI reservation on this LUN.

# Manual Switchover of the Apache Hierarchy to Secondary Server

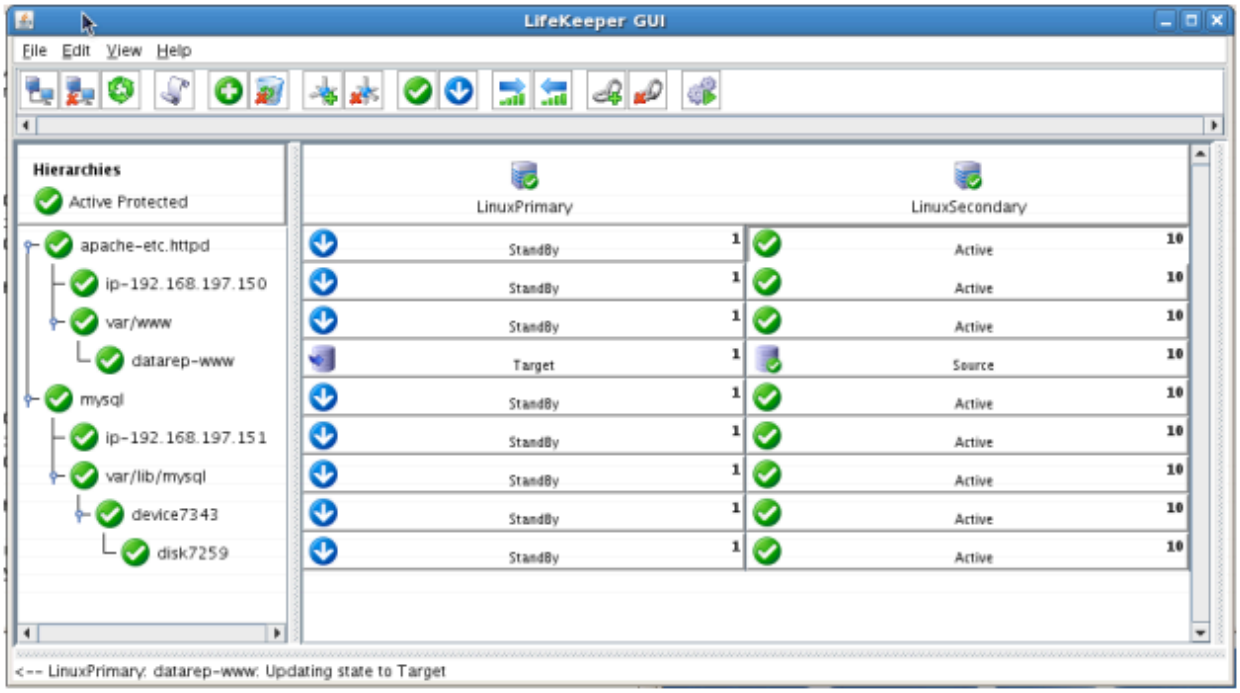
**Procedure:**

- From the LifeKeeper GUI, right click on the Apache resource on the Secondary Server (LINUXSECONDARY) and choose IN SERVICE.
- Click “In Service” in the window that pops up



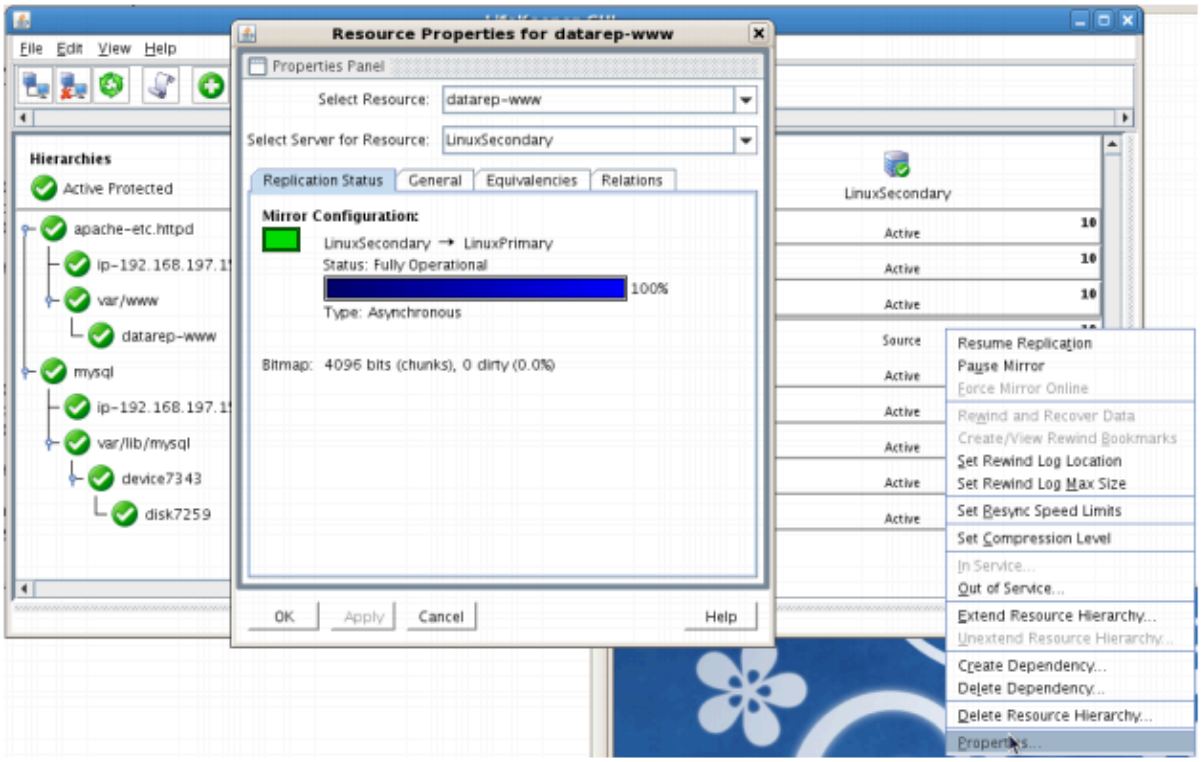

**Expected Result:**

- Beginning with the Apache resource, all resources will be removed from service on the Active Server (LINUXPRIMARY).
- Beginning with the dependent resources (IP and Replicated Volume), all resources will be brought in service on LINUXSECONDARY
- During this process, the direction of the mirror reversed. Data is now transmitting from LINUXSECONDARY -> LINUXPRIMARY
- At this point, we now back to an “Active/Passive” cluster because all services are now actively running on LINUXSECONDARY

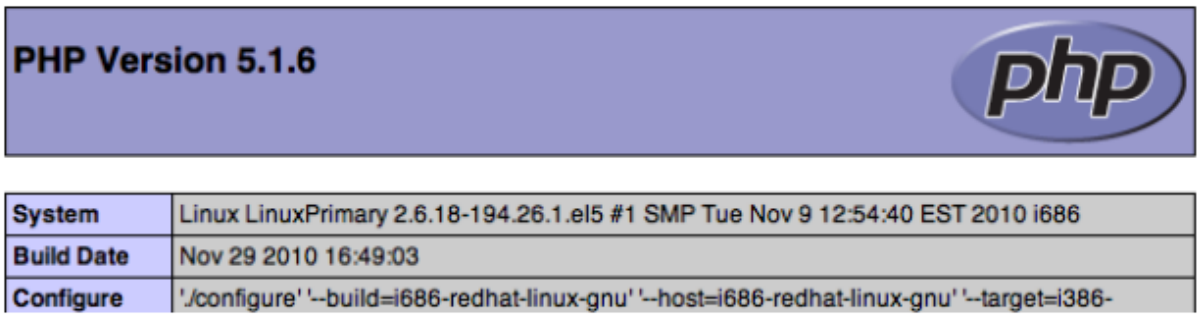


**Tests/Verification:**

- Using the LifeKeeper GUI, verify that the Apache and dependent resources are active on LINUXSECONDARY.
- Using the LifeKeeper GUI, verify the mirror is now reversed and mirroring in the opposite direction. Right click on the “datarep-www” resource and select Properties




- Run “ifconfig –a” on LINUXPRIMARY to validate that the IP Address 192.168.197.150 is active on LINUXPRIMARY
- Run “df –h” to verify that the /var/www replicated filesystem is mounted as an “md” device (example: /dev/md0”) on LINUXPRIMARY
- Verify the Apache services are running on LINUXPRIMARY by running “ps –ef | grep –i httpd”
- Open a Web Browser to http://192.168.197.150 and verify that it can successfully connect. The PHPInfo output should indicate that the system name is “LinuxPrimary



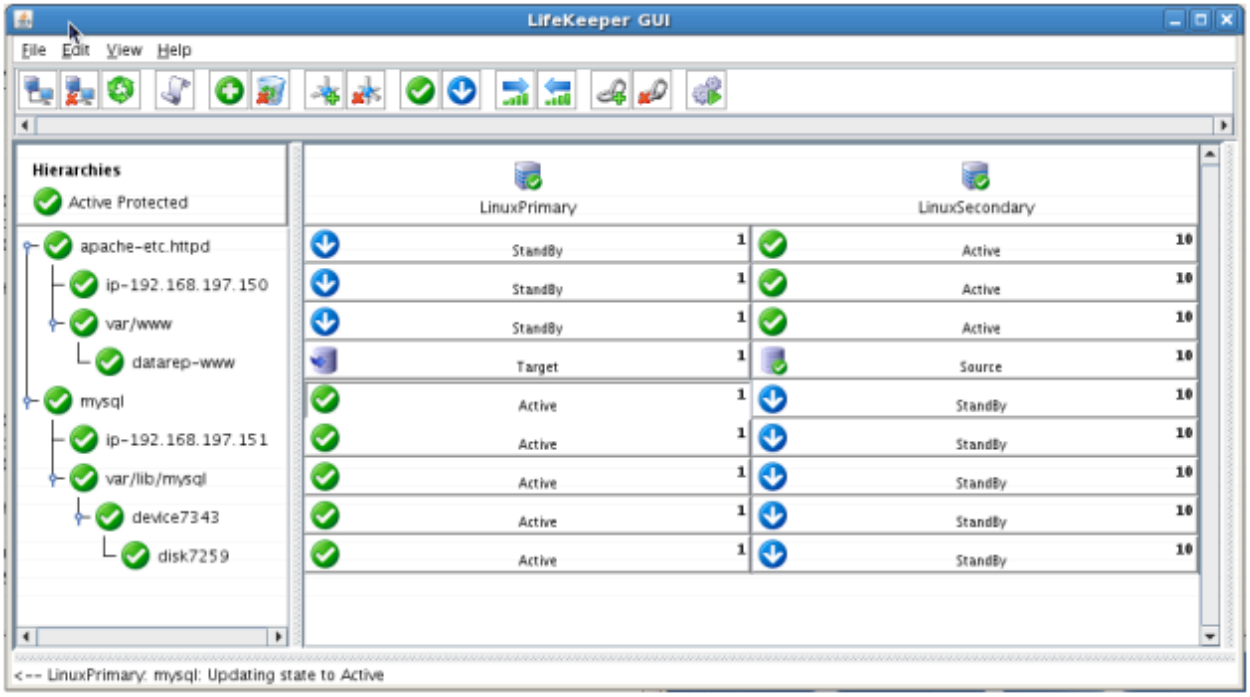
# Manual Switchover of the MySQL Hierarchy back to Primary Server

**Procedure:**

- From the LifeKeeper GUI, right click on the MySQL resource on the Primary Server (LINUXPRIMARY) and choose IN SERVICE.
- Click “In Service” in the window that pops up

**Expected Result:**

- Beginning with the MySQL resource, all resources will be removed from service on the Active Server (LINUXSECONDARY).
- Beginning with the dependent resources (IP and Shared Volume), all resources will be brought in service on LINUXPRIMARY




**Tests/Verification:**

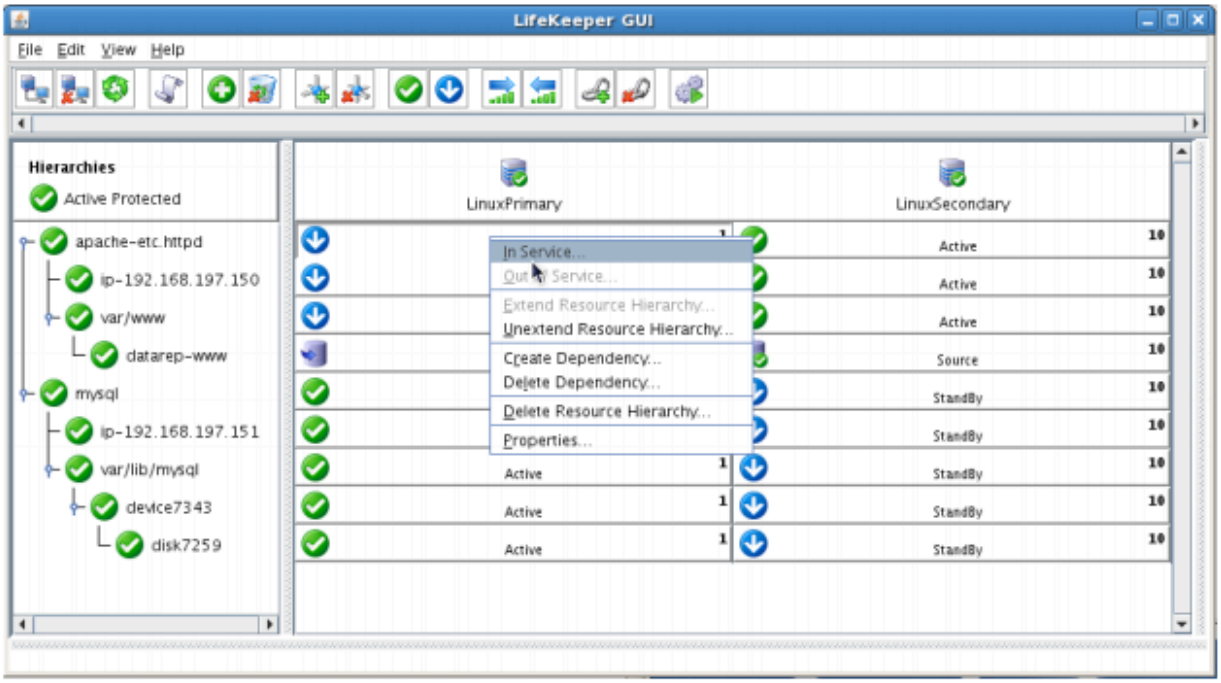
- Using the LifeKeeper GUI, verify that the MySQL and dependent resources are active on LINUXPRIMARY.
- Run “ifconfig –a” on LINUXPRIMARY to validate that the IP Address 192.168.197.151 is active on LINUXPRIMARY
- Run “df –h” to verify that the /var/lib/mysql shared iSCSI filesystem is mounted on LINUXPRIMARY
- Verify the MySQL services are running on LINUXPRIMARY by running “ps –ef | grep –i mysql”
- On LINUXPRIMARY run the following command to verify client connectivity to the MySQL database:
  - # mysql –S /var/lib/mysql/mysql.sock –u root –p

    - (enter password “SteelEye”)
- Verify that the SCSI reservation has properly locked the iSCSI LUN to the currently active cluster node. On LINUXSECONDARY, run “mount /dev/sdc1 /var/lib/mysql”. This should FAIL because LINUXSECONDARY does not own the SCSI reservation on this LUN.

# Manual Switchover of the Apache Hierarchy back to the Primary Server

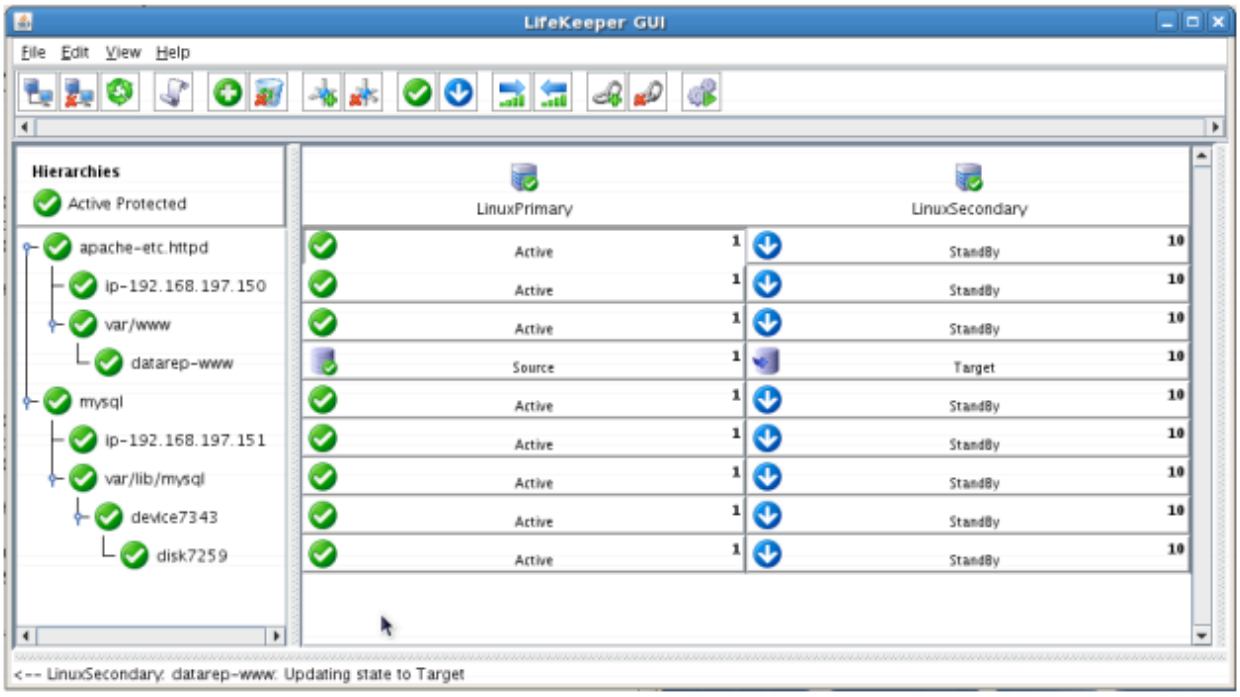
**Procedure:**

- From the LifeKeeper GUI, right click on the Apache resource on the Primary Server (LINUXPRIMARY) and choose IN SERVICE.
- Click “In Service” in the window that pops up



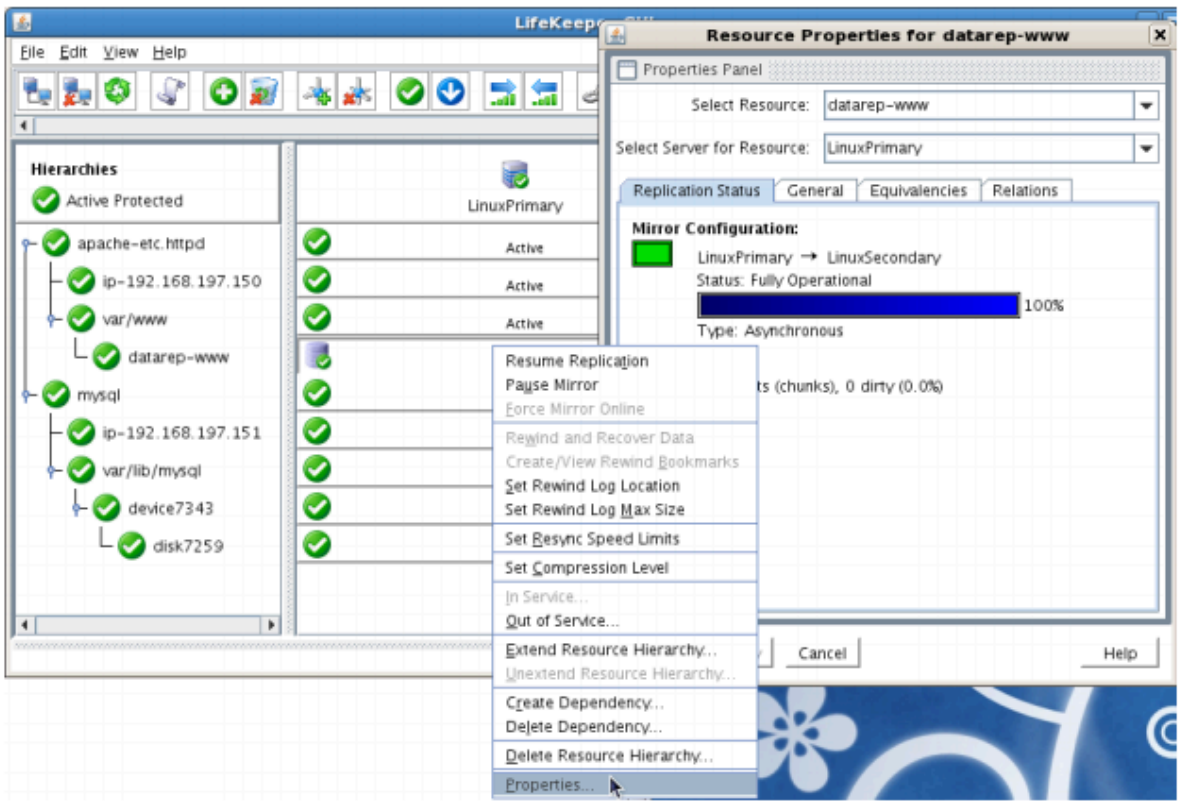

**Expected Result:**

- Beginning with the Apache resource, all resources will be removed from service on the Active Server (LINUXSECONDARY).
- Beginning with the dependent resources (IP and Replicated Volume), all resources will be brought in service on LINUXPRIMARY
- During this process, the direction of the mirror reversed. Data is now transmitting from LINUXPRIMARY -> LINUXSECONDARY




**Tests/Verification:**

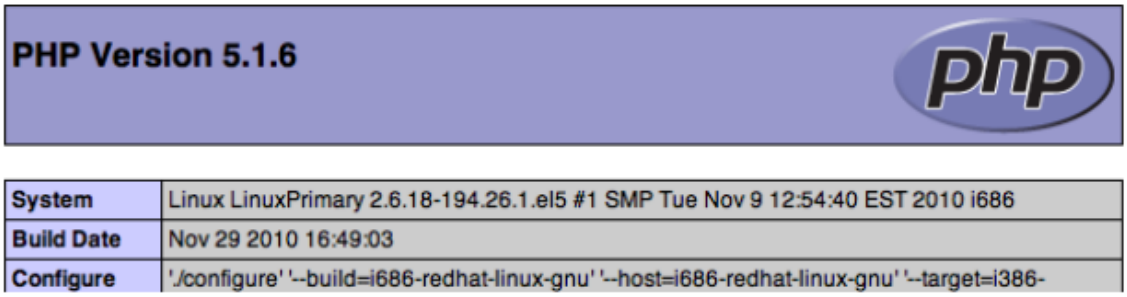
- Using the LifeKeeper GUI, verify that the Apache and dependent resources are active on LINUXSECONDARY.
- Using the LifeKeeper GUI, verify the mirror is now reversed and mirroring in the opposite direction. Right click on the “datarep-www” resource and select Properties




- Run “ifconfig –a” on LINUXPRIMARY to validate that the IP Address 192.168.197.150 is active on

LINUXPRIMARY

- Run “df –h” to verify that the /var/www replicated filesystem is mounted as an “md” device (example: /dev/md0”) on LINUXPRIMARY
- Verify the Apache services are running on LINUXPRIMARY by running “ps –ef | grep –i httpd”
- Open a Web Browser to http://192.168.197.150 and verify that it can successfully connect. The PHPInfo output should indicate that the system name is “LinuxPrimary




## Simulate a network failure on the Primary Server by failing the IP resource

> **IMPORTANT:** Only perform this test if you have more than one communications path configured.

If you perform this test and have only one communications path configured, your system will enter a split-brain scenario as described in the LifeKeeper Administration Guide found here. Refer to this document for more information or contact SIOS presales technical support for assistance in resolving this condition.

**Procedure:**

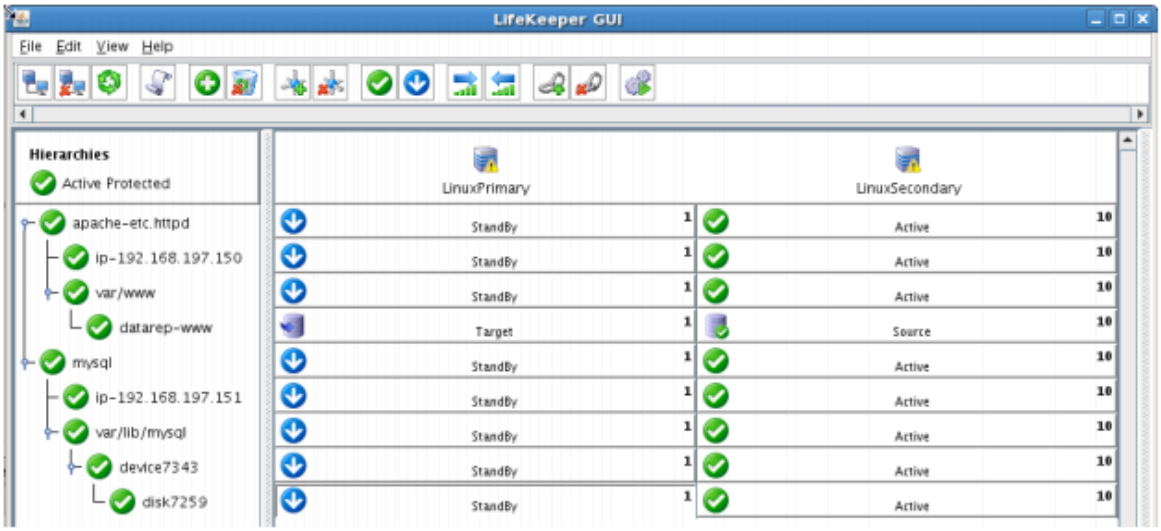
- On LINUXPRIMARY, pull the network cable attached to the NIC that the virtual IP address is configured on

**Expected Result:**

- The IP Resource should fail first.
- The entire hierarchy should failover to LINUXSECONDARY

**Tests/Verification:**

- Check the LifeKeeper Log to verify the IP resource failed – “/opt/LifeKeeper/bin/lk_log log”
- Using the LifeKeeper GUI, verify the MySQL and Apache resource hierarchies fail over successfully to LINUXSECONDARY




## Hard failover of the resource from the Secondary Server back to the Primary Server

**Procedure:**

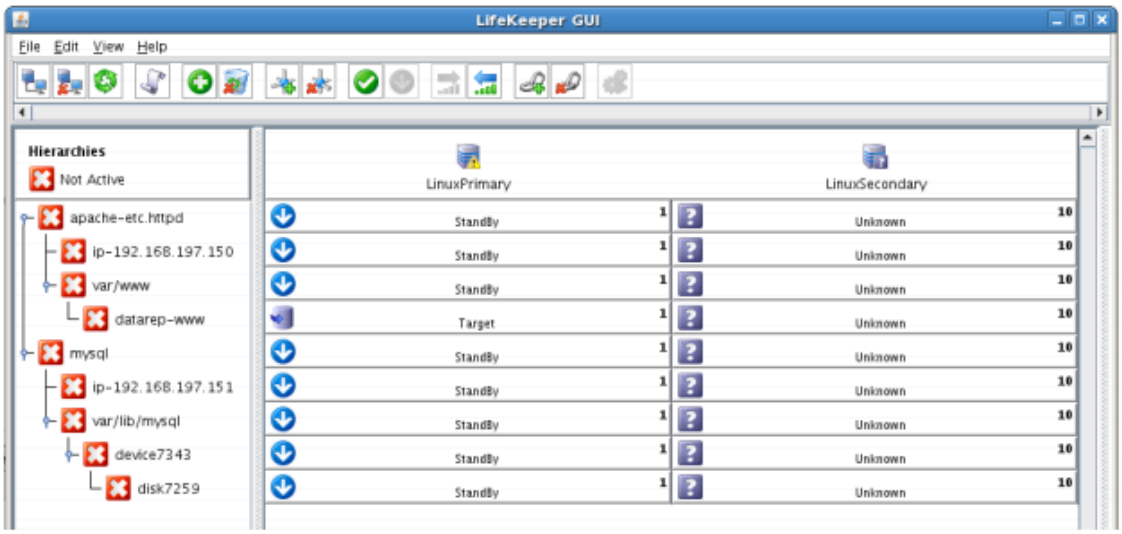
- Pull the power cord on LINUXSECONDARY, as this is the server with all resources currently In Service.

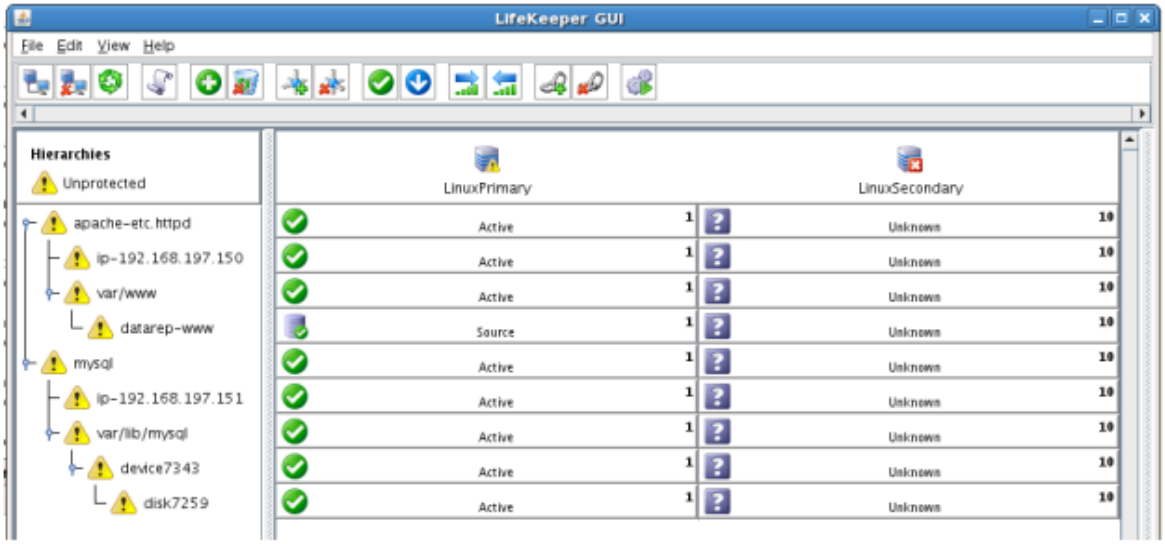
**Expected Result:**

- After failure has been detected, beginning with the dependent resources (IP and Volume), all resources will be brought in service on LINUXPRIMARY.

**Tests/Verification:**

- Using the LifeKeeper GUI, verify the mirror has reversed and is in a Resync Pending state waiting for LINUXSECONDARY to come back on line.
- Verify the PostgreSQL Server services are running on LINUXPRIMARY.
- Verify that the client can still connect to the Webserver and database running on LINUXPRIMARY.
- Verify you can write data to the replicated volume, /var/www on LINUXPRIMARY.




# Bring Failed Server back on line

**Procedure:**

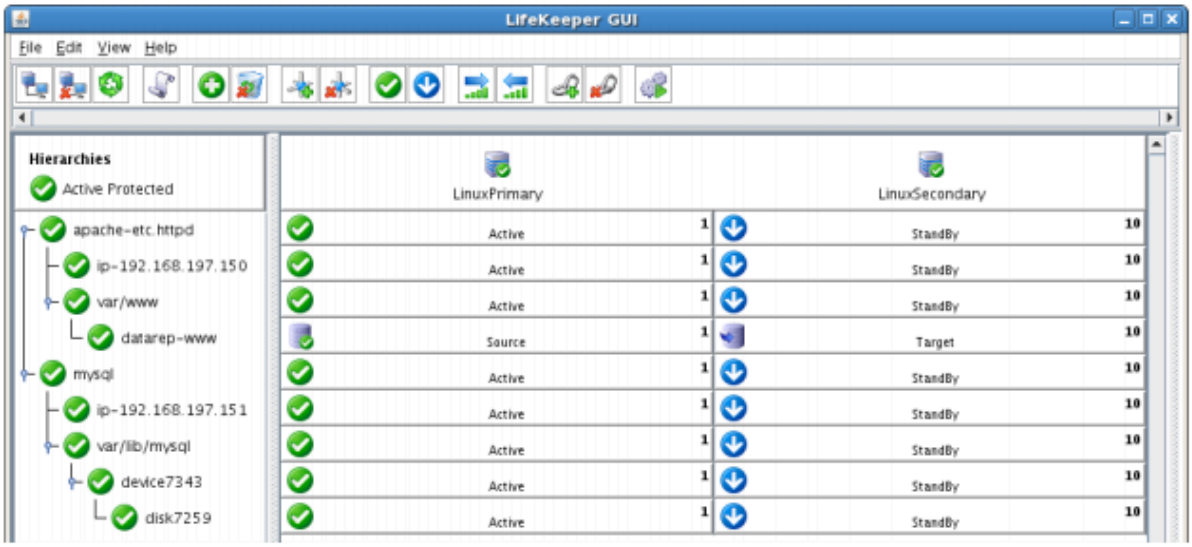
- Plug the power cord back into LINUXSECONDARY and boot it up.

**Expected Result:**

- Using the LifeKeeper GUI, verify that LINUXSECONDARY is coming back up and has become the Standby Server.

**Tests/Verification:**

- Verify the mirror performs a quick partial resync and moves to the Mirroring state
- Verify the Apache and MySQL Hierarchy are in service on LINUXPRIMARY and standby on LINUXSECONDARY.




# Verify Local Recovery of MySQL Server

**Procedure:**

- Kill the PostgreSQL processes via the command line:
  - # ps –ef | grep sql
  - # killall mysqld mysqld_safe
  - run “ps –ef | grep sql once again to verify that the processes no longer exist

**Expected Result: (Assumes Local Recovery for SQL resource is set to YES)**

- The MySQL Server service should stop.
- The MySQL quickcheck process will automatically restart the MySQL Server Service when it runs periodically.
- No failure of MySQL should occur.

**Tests/Verification:**

- Execute “ps –ef | grep sql” once again to verify that the mysql processes have been restored locally on LINUXPRIMARY.
- Verify connectivity to the MySQL database by running:
  - # mysql –S /var/lib/mysql/mysql.sock –u root –p
  - (Enter password “SteelEye”)
- If you inspect the LifeKeeper logs, you will see information indicating that LifeKeeper dectected the failure of the MySQL service and recovered it locally. Run /opt/LifeKeeper/bin/lk_log log for more information.

# 13. LifeKeeper Single Server Protection

LifeKeeper Single Server Protection リリースノート

LifeKeeper Single Server Protection インストレーションガイド

# 13.1. LifeKeeper Single Server Protection for Linux リリースノート

## Version 9.7.0

リリース日： **March 28, 2023**

> **重要** ：SIOS は次の Recovery Kit のサポートを終了します。これらは LifeKeeper for Linux の次のメジャーリリースでサポートされなくなります。
> (なお、新規受注は2022年12月31日をもって終了しています。)
> • NEC iStorage StoragePathSavior
> • Samba
> • Postfix
> • HULFT HUB Recovery Kit

**この製品をインストールまたは使用する前に、このドキュメントをお読みください。このドキュメントには、インストール前、インストール中、およびインストール後に考慮する必要がある最新情報が含まれています。**

## はじめに

このリリースノート（以下本文書）は、 LifeKeeper Single Server Protection for Linux 製品のインストール、設定、管理を行うユーザーを対象として記述されています。本文書には、製品のテクニカルドキュメンテーションには詳細に記述されていない重要な情報（製品の最終テスト時に明らかになったパッケージのバージョン、指示や手順に関する最終段階での変更点、トラブルシューティングセクションへのリンク、製品の制限、トラブル解決のヒントなど）が記載されています。LifeKeeper Single Server Protection をインストールして設定する前に、必ず本文書の内容を確認してください。

### クイックリンク

このリリースの新機能
バグの修正
廃止された機能
コンポーネント
オプションソフトウェア
システム要件
アップグレード
ソフトウェア要件
オープンソースパッケージ
既知の問題

# LifeKeeper Single Server Protection の製品説明

LifeKeeper Single Server Protection は、単一ノード構成におけるアプリケーション監視を可能にします (つまり、クラスターの要件または制約はありません)。単一ノード環境は、物理的なものでも仮想 (vSphere、KVM、Amazon EC2) でも構いません。LifeKeeper Single Server Protection は、実績がある安定した SIOS LifeKeeper アーキテクチャー上に構築されます。LifeKeeper Single Server Protection は優れたアプリケーション監視機能を提供し、障害が発生したアプリケーションおよびシステムインフラストラクチャー項目 (例: NFS 共有、IP アドレス、ファイルシステム) のリカバリーを実行することができます。何らかの理由でアプリケーションをリカバリーできない場合、LifeKeeper Single Server Protection は、システムのリブートまたは VM とアプリケーション監視を設定された VMware 仮想マシンの VMware HA 再起動によって、ノードの再起動を開始します。

# LifeKeeper Single Server Protection for Linux バージョン 9 / サポート対象バージョンの新機能

古いバージョンのドキュメントについては、 https://docs.us.sios.com/alllinux.php を参照してください。

| 製品 | 機能 |
|---|---|
| バージョン **9.7.0** の新機能 | |
| LifeKeeper Core | Red Hat Enterprise Linux 9.1 をサポートしました（2023年6月認定） |
| | Red Hat Enterprise Linux 9.0 をサポートしました。 |
| | Red Hat Enterprise Linux 8.7 をサポートしました。(2023年6月認定) |
| | セットアップ実行時のファイアウォール検出を強化しました。 |
| | Rocky Linux 9.0 をサポートしました。<br>• Rocky Linux はクラウドではサポートされていません。 (例：AWS/Azure/GCP/OCI)<br>• Rocky Linux v9.0 は、LifeKeeper for Linux v9.7.0 でのみサポートされています。<br>• SAP Recovery Kit および SAP HANA Recovery Kit はサポートされていません。 |
| | Rocky Linux 8.7 をサポートしました。(2023年6月認定)<br>**注意** : DataKeeper を使用する場合、LifeKeeper インストール時に特定の手順に従う必要があります。詳細は こちら をご参照ください。<br>• SAP Recovery Kit と SAP HANA Recovery Kit はサポートされていません。 |
| | Miracle Linux 8.6 をサポートしました。 |
| | VMware vSphere 8.0 をサポートしました。(2023年6月認定) |

| | |
|---|---|
| | Oracle Linux 9.1 をサポートしました。（2023年6月認定） |
| | Oracle Linux 9.0 をサポートしました。（2023年6月認定） |
| | Oracle Linux 8.7 をサポートしました。（2023年6月認定）<br>**注意**：DataKeeper を使用する場合、LifeKeeper インストール時に特定の手順に従う必要があります。詳細は こちら をご参照ください。 |
| | Miracle Linux 9.0 をサポートしました。（2023年6月認定）<br>**注意**：カーネルの問題により、シンプロビジョニングディスク上の DataKeeper 非同期モードは使用できません。 |
| PostgreSQL Recivery Kit | PostgreSQL 15 をサポートしました。 |
| NFS | LifeKeeper は、サーバーがファイルシステムをエクスポートし、エクスポートされたファイルシステムを同じシステムにマウントする NFS ループバック構成をサポートしません。 NFS リソースのインストール / 更新または作成中に、サポートされていない構成が検出された場合、エラーメッセージがログに記録されます。 |
| バグの修正 | |
| **バージョン 9.6.2 の新機能** | |
| LifeKeeper Core | Red Hat Enterprise Linux 8.6 をサポートしました。 |
| | Oracle Linux 8.6 をサポートしました。 |
| | Miracle Linux 8.4をサポートしました。 |
| バグの修正 | |
| **バージョン 9.6.1 の新機能** | |
| LifeKeeper Core | Red Hat Enterprise Linux 8.5 をサポートしました。 |
| | Oracle Linux 8.5 をサポートしました。 |
| | バグの修正 |
| Recovery Kit for Oracle Cloud Infrastructure | Oracle Cloud 環境で、セカンダリー IP アドレスの割り当て状態を監視することができるようになりました。 |
| Network Attached Storage Recovery Kit | UDP が無効になっている NFS を保護する際に、「NFS_RPC_PROTOCOL=tcp」を設定する必要がなくなりました。 |
| LifeKeeper GUI、Sybase ASE Recovery Kit | バグの修正 |
| **バージョン 9.6.0 の新機能** | |
| LifeKeeper Core | Oracle Linux 8.4 をサポートしました。 |
| | SUSE Linux Enterprise Server 15 SP3 をサポートしました。 |

| | |
|---|---|
| | ※サポートされるKernel Versionは5.3.18-59.5以降です。 |
| Oracle Recovery Kit | オンプレミス上で動作するOracle 21c(21.3)をサポートしました。 |
| PostgreSQL Recovery Kit | PostgreSQL14 をサポートしました。 |
| | EDB Postgres Advanced Server 14.0をサポート（2022年1月認定）しました。 |
| バグの修正 | |
| **バージョン 9.5.2 の新機能** | |
| LifeKeeper Core | Red Hat Enterprise Linux 8.4 をサポートしました。 |
| | CentOS 8.3 をサポートしました。 |
| | Oracle Linux 8.3 をサポートしました。 |
| | Oracle Linux 7 UEK 6をサポートしました。<br>注意：カーネルバージョンが5.4.17-2102.202.5 以降のみサポートされます。 |
| | Oracle Linux 8 UEK 6をサポートしました。<br>注意：カーネルバージョンが5.4.17-2102.202.5 以降のみサポートされます。 |
| | 以下の ARK で LKCLI の利用をサポートしました。<br>• SAP MaxDB RK<br>• MQ RK |
| | lk group に所属したroot以外のユーザーからでも lkcliを実行できるようになりました。 |
| | lkcli stop でも lkstop 同様に「-i」オプションを使用できるようになりました。 |
| | バグの修正 |
| MySQL | MariaDB 10.5 をサポート。 |
| | バグの修正 |
| MQ | WebSphere MQ 9.2 for RHEL 7.9 をサポート。 |
| | WebSphere MQ 9.2 for RHEL 8.3 をサポート。 |
| Install、NFS、IP、Oracle、Quorum/Witness、Filesystem、Generic、lksupport | バグの修正 |
| **バージョン 9.5.1 の新機能** | |
| LifeKeeper Core | Red Hat Enterprise Linux 7.9 をサポート。（2020年12月認定）<br>注意: DataKeeperを使用する場合、LifeKeeperインストール時に特定の手順に従う必要があります。詳細は こちら をご参照ください。 |
| | CentOS 7.9 をサポート。（2020年12月認定）<br>注意: DataKeeperを使用する場合、LifeKeeperインストール時に特定の手順に従う |

| | |
|---|---|
| | 必要があります。詳細は こちら をご参照ください。 |
| | Oracle Linux 7.9 をサポート。（2020年12月認定）<br>注意: DataKeeperを使用する場合、LifeKeeperインストール時に特定の手順に従う必要があります。詳細は こちら をご参照ください。 |
| | 以下の ARK で LKCLI の利用をサポート。<br>• DB2 RDBMS RK<br>• RAW RK<br>• Postfix RK<br>• VMDK as Shared Storage RK<br>• SAP ASE RK<br>• Samba RK<br>• HULFT RK / HULFT HUB RK<br>• Oracle RK の Oracle PDB 保護機能 |
| | Red Hat Enterprise Linux 8.3 をサポート。（2021年1月認定） |
| | Red Hat Enterprise Linux 8.2 をサポート。 |
| | CentOS 8.2 をサポート。 |
| | Oracle Linux 8.2（UEK6 は除く）をサポート。 |
| | SLES15 SP2 をサポート。 |
| | LifeKeeper 内部で使用している OpenSSL パッケージを 1.1.1g に更新。 |
| | LifeKeeper 内部で使用している cURL パッケージを 7.68.0 に更新。 |
| | lkstop コマンドに -i オプション を追加。このオプションは保護するリソースを停止せずに LifeKeeper のみを停止します。また、停止処理を続行するために「yes/no」の確認を行う必要があります。 |
| | バグの修正 |
| MQ | SIOS Protection Suite for Linux は IBM MQ 9.2 の保護をサポート。 (2021年1月) |
| QSP | Quick Service Protection(QSP) RK によるApache Tomcat の保護をサポート。 |
| install | LifeKeeper の 構成のみ行うセットアップのモード をサポート。その他、セットアップに関する改修。 |
| | バグの修正 |
| PostgreSQL | PostgreSQL 13 をサポート。（2020年12月認定） |
| | EDB Postgres Advanced Server 13.0 をサポート。（2020年12月認定） |
| SAP MaxDB、DB2、Filesystem、GenericApplication、IP | バグの修正 |

# バグの修正

下記は、本バージョンで修正されたバグおよび拡張機能の一覧です。

| バグ | 説明 |
|---|---|
| PL-12327 | ハングした exportfs コマンドが原因で、NFS リソースをインサービスにするプロセスがハングする可能性がある問題を修正しました。 |
| PL-14290 | lkGUIapp のEdit -> Server -> Properties で NullPointerException で落ちてしまう問題を修正しました。 |
| PL-14355 | 同梱する OpenSSL を 3.0.7 に更新しました。これにより TLS 1.1 以前のプロトコルは使用されなくなりました。 |
| PL-15862 | LifeKeeper は、ライセンス キーの有効期限が切れたために起動が失敗した場合にメッセージをログに記録します。 |
| PL-15331 | lifekeeper start 時に nfs-server service も起動してしまう問題を修正しました。 |

# 廃止された機能

| 製品 | 機能 |
|---|---|
| バージョン **9.7.0** で廃止になった機能 | |
| 同じノード上の NFS サーバー/クライアントを使用した新規または既存の構成 (「ループバック構成」とも呼ばれます) のサポートを終了。 | |
| バージョン **9.6.2** で廃止になった機能 | |
| | ありません |
| バージョン **9.6.1** で廃止になった機能 | |
| | ありません |
| バージョン **9.6.0** で廃止になった機能 | |
| | ありません |
| バージョン **9.5.2** で廃止になった機能 | |
| Core | SteelEye 管理コンソールのサポートを終了。 |
| | Red Hat Enterprise Linux 6 のサポートを終了。 |
| | CentOS 6 のサポートを終了。 |
| | Oracle Linux 6 のサポートを終了。 |
| Oracle | Oracle virtual machine (OVM) 3.4.6のサポートを終了。 |

| | |
|---|---|
| PostgreSQL | PostgreSQL 9.5のサポートを終了。 |
| | EnterpriseDB Postgres Plus Advanced Server 9.5のサポートを終了。 |
| VMware | VMware vSphere 5.5 , 6.0のサポートを終了。 |
| バージョン **9.5.1** で廃止になった機能 | |
| | ありません |
| バージョン **9.5.0** で廃止になった機能 | |
| LifeKeeper Core | syslog-ng を用いた system ログ管理のサポートを終了。rsyslog を使用してください。 |
| | SUSE Linux Enterprise Server (SLES) 11.0 ～ SP4 のサポートを終了。 |
| Oracle | Oracle Database Enterprise Edition 11g R2 のサポートを終了。 |
| | Oracle Database Standard Edition 11g R2 のサポートを終了。 |
| | Oracle Database Standard Edition One 11g R2 のサポートを終了。 |
| MySQL | MariaDB 5.5, 10.0 のサポートを終了。 |
| PostgreSQL | PostgreSQL 9.4 のサポートを終了。 |
| | EnterpriseDB Postgres Plus Advanced Server 9.4 のサポートを終了。 |
| バージョン **9.4.1** で廃止になった機能 | |
| | ありません |

## コンポーネント

LifeKeeper Single Server Protection の中核となるソフトウェアです。64bit システム（AMD64、EM64T システム）でのみ稼動します。
コアには次のパッケージが含まれ、サーバーにインストールして使用されます。

| パッケージ | パッケージ名 | 説明 |
|---|---|---|
| LifeKeeper Core | steeleye-lk-9.7.0-7520.x86_64.rpm | メモリー、CPU 、OS 、SCSI ディスクサブシステム 、ファイルシステムなど、のシステムの中核となるコンポーネントの障害回復機能を提供します。 |
| LifeKeeper GUI | steeleye-lkGUI-9.7.0-7520.x86_64.rpm | LifeKeeper および Datakeeper の管理および健全性監視用のグラフィカルユーザーインターフェースです。 |
| LifeKeeper IP Recovery Kit | steeleye-lkIP-9.7.0-7520.noarch.rpm | IP アドレスの障害回復機能を提供します。 |

| | | |
|---|---|---|
| Quick Service Protection | steeleye-lkQSP-9.7.0-7520.noarch.rpm | 様々なサービスに対して、簡易的な障害回復機能を提供します。 |
| LifeKeeper Man Page | steeleye-lkMAN-9.7.0-7520.noarch.rpm | LifeKeeper 製品のリファレンスマニュアルのページが含まれます。 |

## LifeKeeper Single Server Protection オプションソフトウェア

| パッケージ | パッケージ名 | 説明 |
|---|---|---|
| LifeKeeper Apache Web Server Recovery Kit | steeleye-lkAPA-9.7.0-7520.noarch.rpm | Apache Web Server v2.4 |
| LifeKeeper SAP MaxDB Recovery Kit | steeleye-lkSAPDB-9.7.0-7520.noarch.rpm | SAP MaxDB v7.9 |
| LifeKeeper DB2 Recovery Kit | steeleye-lkDB2-9.7.0-7520.noarch.rpm | IBM DB2 Enterprise Server Edition (ESE) v10.5, v11.1 ,v11.5<br><br>IBM DB2 Workgroup Server Edition (WSE) v10.5, v11.1 ,v11.5<br><br>IBM DB2 Express Edition v10.5, v11.1 ,v11.5 |
| LifeKeeper Oracle Recovery Kit | steeleye-lkORA-9.7.0-7520.noarch.rpm | Oracle Database Enterprise Edition v12c, v12c R2, v18c ,v19c (excluding ASM and pluggable database), v21c (excluding ASM and pluggable database features)<br><br>Oracle Database Standard Edition 2 (SE2) v12c, v12c R2, v18c ,v19c (excluding ASM and pluggable database), v21c (excluding ASM and pluggable database features) |
| LifeKeeper MySQL Recovery Kit | steeleye-lkSQL-9.7.0-7520.noarch.rpm | MySQL Community Edition, MySQL Enterprise Edition v5.7 ,v8.0<br><br>MariaDB v10.3, v10.4, v10.5 |

| | | |
|---|---|---|
| LifeKeeper PostgreSQL Recovery Kit | steeleye-lkPGSQL-9.7.0-7520.noarch.rpm | PostgreSQL v9.6, v10, v11, v12, v13 ,v14 ,v15<br><br>EnterpriseDB Postgres Plus Advanced Server/EDB Postgres Advanced Server v9.6, v10.0, v11.0, v12.0, v13.0 ,v14.0 |
| LifeKeeper Sybase ASE Recovery Kit | steeleye-lkSYBASE-9.7.0-7520.noarch.rpm | Sybase ASE 15.7 ,16.0 |
| LifeKeeper Postfix Recovery Kit | steeleye-lkPOSTFIX-9.7.0-7520.noarch.rpm | 対応する Linux ディストリビューションに付属の Postfix ソフトウェアを各サーバーにインストールし、設定する。各サーバーには、同じバージョンの Postfix をインストールする必要があります。 |
| LifeKeeper Samba Recovery Kit | steeleye-lkSMB-9.7.0-7520.noarch.rpm | 対応する Linux ディストリビューションで提供される標準的な Samba ファイルサービス |
| LifeKeeper NFS Server Recovery Kit | steeleye-lkNFS-9.7.0-7520.noarch.rpm | Linux kernel version 2.6 以降<br><br>NFS サーバーおよびクライアントパッケージが SLES システムにインストールされている必要があります。 |
| LifeKeeper Network Attached Storage Recovery Kit | steeleye-lkNAS-9.7.0-7520.noarch.rpm | NFS サーバーまたは Network Attached Storage (NAS) デバイスからマウントされた NFS ファイル システムの NFS バージョン v2、v3、および v4 |
| LifeKeeper WebSphere MQ Recovery Kit | steeleye-lkMQS-9.7.0-7520.noarch.rpm | IBM MQ v8.0, v9.0, v9.1 ,v9.2<br><br>詳細については 既知の問題と制限 > インストールを参照してください。 |
| Recovery Kit for Oracle Cloud Infrastructure | steeleye-lkOCIVIP-9.7.0-7520.noarch.rpm | Oracle Cloud 環境で、セカンダリー IP アドレスの割り当て状態を監視することができます。 |

# システム要件

## LifeKeeper Single Server Protection 製品要件

LifeKeeper Single Server Protection は、以下の表に含まれる最小要件を満たす全ての Linux プラットフォームでサポートされています。

| 説明 | 要件 |
|---|---|
| Linux オペレーティングシステム | 個々のオペレーティングシステム情報については、「Linuxの設定」を参照してください。 |
| 仮想環境 | 仮想マシン内で起動するゲスト OS が「Linuxの設定」に記載されているサポート対象のバージョンのうちの 1 つである必要があります。以下の仮想環境は SIOS Protection Suite for Linux が展開されている場合の例です。<br>サポート対象となる仮想化環境の詳細なバージョンについてはサポートマトリックス を参照ください。LifeKeeper Single Server Protection との組み合わせで利用可能なクラウド環境については こちら を参照してください。<br><br>• KVM<br>• Oracle VM Server for x86<br>• VMware vSphere v6.5、v6.7、v7.0、v8.0<br>• Amazon EC2<br>• Microsoft Azure<br>• Nutanix Acropolis Hypervisor<br>• Google Cloud<br>• Oracle Cloud Infrastructure (**注意 2**)<br><br>VMware vSphere v6.5 以降 では vSAN の利用をサポートします。ただし RDM は、vSAN で非サポートとなるため、利用できません。<br><br>ファイバーチャンネル SAN および共有 SCSI クラスタ設定は、KVM および Oracle VM Server for x86 仮想マシン上で動作する LifeKeeper Single Server Protection をサポートしません。<br><br>**注意 1：** AWSやAzure環境で動作するSLESにおいて、クラウドネットワークプラグインによって仮想 IP アドレスの動的な変更が行われることにより、LifeKeeper クラスターの動作に影響を与えることがあります。詳細や回避策は トラブルシューティング > 既知の問題と制限 >「LifeKeeper Core」をご確認ください。<br><br>**注意 2：** Oracle Cloud Infrastructure 上では LifeKeeper がサポートする OS と ARK について一部制限があります。詳しくは こちら をご確認ください。 |
| メモリー | システムの最小メモリー要件は、OS が必要とする最小要件を満たしてください。システムのメモリーは SPS が保護するシステム上で動作するアプリケーションに対してサイジングする必要があります。 |

| | |
|---|---|
| ディスク容量 | LifeKeeper Single Server Protection の導入に必要なディスク容量は次のとおりです。<br>• /opt – 約 100MB (インストールするキットに依存します)<br>• / – 約 110MB |
| Java Runtime Environment | • OpenJDK 1.8、10以降 |

## アップグレード

LifeKeeper Single Server Protection v9.7.0 へは、v9.5.x および v9.6.x からアップグレードすることができます。これらより前のバージョンからアップグレードする場合、それまでのバージョンをアンインストールし、LifeKeeper Single Server Protection を再インストールする必要があります。それまでのバージョンをアンインストールしないで v9.7.0 へアップグレードしたい場合には、一度 v9.5.x または v9.6.x にアップグレードした後、さらに v9.7.0 にアップグレードする方法をとることもできます。

## サポートソフトウェアの要件

下表のサポートソフトウェアは、VM とアプリケーション監視を設定された VMware VM でのみ必要です。

| 製品 | 要件 | ディスク容量要件 |
|---|---|---|
| VMware | VMware vSphere Client (LifeKeeper Single Server Protection vSphere Client プラグイン機能用)<br><br>保護されるすべての仮想マシンに VMware Tools がインストールされ、実行されている<br><br>VMware アプリケーション HA 監視が有効で、保護されるすべての仮想マシンに対して VM とアプリケーション監視が設定されている | */opt* で約 175 KB (VMware Tools の場合) |

## オープンソースパッケージ

以下のオープンソースパッケージは、Single Server Protection インストールイメージに含まれています。

| 名称 | バージョン | ライセンスのタイプとバージョン |
|---|---|---|

| | | |
|---|---|---|
| curl | 7.86.0 | curl |
| libxml2 | 2.7.8 | MIT |
| lighttpd | 1.4.68 | BSD-3-Clause |
| openssl | 3.0.7 | Apache-2.0 |
| openssl | 1.0.1e | OpenSSL |
| pcre | 4.5 | PCRE |
| pdksh | 5.2.14 | Public Domain and GPL |
| perl | 5.8.8 | GPL-1.0-or-later or Artistic-1.0-Perl |
| Perl Addon Packages | – | Various |
| powercli | 11.5.0 | various license |
| powershell | 6.2.3 | MIT |
| readline | 4.3 | GPL-2.0-or-later |
| runit | 2.0.0 | BSD-3-Clause |
| util-linux | 2.31.1 | GPLv2 and GPLv2+ and LGPLv2+ and BSD with advertising and Public Domain |
| openjdk | 12.0.2 | GPL-2.0-or-later |
| kconfig-frontends | 4.11.0 | GPL-2.0-only |
| ping | – | BSD-4-Clause |
| zlib-static | 1.2.3 | Zlib |
| Perl Config::IniFiles (CPAN module) | 2.27 | GPL/Artistic (Same as Perl) |
| balance | 3.54 | GPL-2.0-only |
| nbd-client | 1.0 | GPL |
| nbd-server | 1.3 | GPL |
| nbd (kernel module) | 2.6.32<br>3.10.0<br>4.18.0 | GPL-2.0-only |
| raid1 (kernel module) | 2.6.32<br>4.18.0 | GPL-2.0-only |

# 既知の問題

既知の問題、回避策、その他のトラブルシューティングについては、 LifeKeeper Single Server Protection

for Linux テクニカルドキュメンテーション の既知の問題と回避策 セクションを参照してください。

**商標について**
「Amazon Web Services」、「Powered by Amazon Web Services」のロゴ、「AWS」、「Amazon EC2」、「EC2」、「Amazon Elastic Compute Cloud」、「Amazon Route 53」、「Amazon Virtual Private Cloud」および「Amazon VPC」は、米国その他の国における Amazon.com, Inc. またはその関連会社の商標です。
「Google Cloud」、「BigQuery」および「Google Compute Engine」は Google LLC の商標です。
「Oracle」および「Java」は、オラクルおよびその関連会社の登録商標です。その他の社名、商品名等は各社の商標または登録商標である場合があります。
なお、本文および図表中では、「™」、「®」は一部を除き明記しておりません。

# 13.2. LifeKeeper Single Server Protection インストレーションガイド

## LifeKeeper Single Server Protection for Linux について

LifeKeeper Single Server Protection (SSP) は、単一ノード構成におけるアプリケーション監視を可能にします (つまり、クラスタの要件または制約はありません)。単一ノード環境は、物理的なものでも仮想 (vSphere、KVM) でも構いません。LifeKeeper SSP は、実績がある安定した SIOS LifeKeeper アーキテクチャ上に構築されます。LifeKeeper SSP は優れたアプリケーション監視機能を提供し、障害が発生したアプリケーションおよびシステムインフラストラクチャ項目 (例: NFS 共有、IP アドレス、ファイルシステム) のリカバリを実行することができます。何らかの理由でアプリケーションをリカバリできない場合、LifeKeeper SSP は、システムのリブートまたは VM とアプリケーション監視を設定された VMware 仮想マシンの VMware HA 再起動によって、ノードの再起動を開始します。

**注記:** LifeKeeper SSP は SIOS LifeKeeper 技術を使用して構築されているため、ドキュメント全体で LifeKeeper を参照します。また、両製品に共通するトピックについては SIOS Protection Suite for Linux ドキュメンテーションの情報を参照します。これらの共通のトピックを参照する場合、LifeKeeper SSP には以下の話題は適用されません。また、サポートもされません。

- クラスタリング
- コミュニケーションパス
- 共有ストレージ (要件、構成、...)
- リソース階層の拡張/拡張解除
- Quorum/Witness
- アプリケーションリカバリーキット
  - Recovery Kit for EC2
  - LVM Recovery Kit
  - MD Recovery Kit
  - Route53 Recovery Kit
  - SAP Recovery Kit
  - SAP HANA Recovery Kit
  - VMDK as Shared Storage Recovery Kit
- ストレージキット (DR、DMMP、HDLM、LVM、MD、PPATH、NEC SPS)

  **注記:** LifeKeeper SSP にベースとなるストレージキットがない場合、保護されるファイルシステムのマウントに必要なデバイスがシステム起動時にアクティベートされるようにするための手順が必要です (例: ファイルシステムが論理ボリューム上でマウントされる場合、LifeKeeper SSP が起動する前にボリュームがアクティブな状態になっていなければなりません)。
- リソース/マシンのフェイルオーバ (LifeKeeper SSP のデフォルトでは、これによってノードが再起

動されます)

- リソースのスイッチオーバ

- 切り替え可能な IP アドレス (LifeKeeper SSP では、保護されるアプリケーションの一部には切り替え可能な IP アドレスが必要ですが、単一ノードしかないため、実際には切り替えは行われません)

LifeKeeper SSP のベースになっている SIOS LifeKeeper 製品の詳細については、共通するリリース番号の SIOS Protection Suite for Linux ドキュメンテーション を参照してください。このドキュメンテーションは、リソース階層の作成、リソースタイプ、状態と関係、グラフィカルユーザインターフェース (GUI)、および共通の作業と高度な作業に関する詳細情報を提供します。

# 13.2.1. はじめに

## LifeKeeper Single Server Protection for Linux について

LifeKeeper Single Server Protection (SSP) は、単一ノード構成におけるアプリケーション監視を可能にします (つまり、クラスタの要件または制約はありません)。単一ノード環境は、物理的なものでも仮想 (vSphere、KVM) でも構いません。LifeKeeper SSP は、実績がある安定した SIOS LifeKeeper アーキテクチャ上に構築されます。LifeKeeper SSP は優れたアプリケーション監視機能を提供し、障害が発生したアプリケーションおよびシステムインフラストラクチャ項目 (例: NFS 共有、IP アドレス、ファイルシステム) のリカバリを実行することができます。何らかの理由でアプリケーションをリカバリできない場合、LifeKeeper SSP は、システムのリブートまたは VM とアプリケーション監視を設定された VMware 仮想マシンの VMware HA 再起動によって、ノードの再起動を開始します。

> 注記: LifeKeeper SSP は SIOS LifeKeeper 技術を使用して構築されているため、ドキュメント全体で LifeKeeper を参照します。また、両製品に共通するトピックについては SIOS Protection Suite for Linux ドキュメンテーションの情報を参照します。これらの共通のトピックを参照する場合、LifeKeeper SSP には以下の話題は適用されません。

- クラスタリング
- コミュニケーションパス
- 共有ストレージ (要件、構成、...)
- リソース階層の拡張/拡張解除
- ストレージキット (DR、DMMP、HDLM、LVM、MD、PPATH、NEC SPS)

> 注記: LifeKeeper SSP にベースとなるストレージキットがない場合、保護されるファイルシステムのマウントに必要なデバイスがシステム起動時にアクティベートされるようにするための手順が必要です (例: ファイルシステムが論理ボリューム上でマウントされる場合、LifeKeeper SSP が起動する前にボリュームがアクティブな状態になっていなければなりません)。

- リソース/マシンのフェイルオーバ (LifeKeeper SSP のデフォルトでは、これによってノードが再起動されます)
- リソースのスイッチオーバ
- 切り替え可能な IP アドレス (LifeKeeper SSP では、保護されるアプリケーションの一部には切り替え可能な IP アドレスが必要ですが、単一ノードしかないため、実際には切り替えは行われません)

> 注記: Amazon EC2上で使用する場合、IP リソースを作成する前に Amazon EC2マネジメ

> ントコンソールを使用して NIC にセカンダリプライベート IP アドレスを割り当ててください。次に、LifeKeeper GUI クライアントを使用している NIC でプライベート IP アドレスとして IP リソースを作成してください。Elastic IP を IP リソースに関連付けることもできます。

LifeKeeper SSP のベースになっている SIOS LifeKeeper 製品の詳細については、共通するリリース番号の SIOS Protection Suite for Linux ドキュメンテーション を参照してください。このドキュメンテーションは、リソース階層の作成、リソースタイプ、状態と関係、グラフィカルユーザインターフェース (GUI)、および共通の作業と高度な作業に関する詳細情報を提供します。

# 13.2.2. LifeKeeper Single Server Protection ソフトウェアのインストール

本ページでは Linux OS について基本的な知識がある方にむけて LifeKeeper Single Server Protection Software (SSP) のインストールについて説明しています。その他詳細については LifeKeeper Single Server Protection テクニカルドキュメンテーション を参照してください。

## インストール前の要件

SSP for Linux をインストールする前に、下記の要件について確認してください。

- **LifeKeeper Single Server Protection リリースノート** -リリースノートにはサポート対象のプラットフォーム、オペレーティングシステム、アプリケーション、ストレージについて記載されています。最新のバグフィックスについても記載されています。
- **TCP/IP 接続と名前解決** – GUI の機能を使用するためにクラスターノードの双方で名前解決ができる必要があります。DNS サービスまたは */etc/hosts* を使用してください。 ローカルホストは 127.0.0.1 となります。
- ファイアーウォール – 以下のポートを使用します。
  - コミュニケーションパス (TCP): 7365/tcp
  - GUI サーバーとの通信: 81/tcp、82/tcp
  - GUI サーバーとクライアントとの RMI 通信: 1024/tcp 以降の全ポート

ファイアーウォールに関して
- GUI サーバー、クライアントの通信に使用するポートは SSP がインストールされるクラスターノードと、GUI クライアントを動作させる全てのシステムでポートが開放されている必要があります。
- GUI サーバー、クライアント間の通信では Java の RMI にて 1024 番以降のポートをランダムに使用します。設定方法等は テクニカルドキュメント をご参照ください。
- 使用しているポート番号に以下を追加します。 WebGUI サーバーのプロセスと lkpolicy コマンドによるポリシー設定で使用 : 778(SSL) /tcp

- **SELinux の設定確認** – SELinux のモードによっては、SSP for Linux はインストールできません。
  - enforcing モード – SSP for Linux をインストールできません。
  - permissive モード – SSP for Linux をインストールできます。(いくつかのテスト済み ARK 環境を除いて推奨できません。)
    - SAP 環境で必要にならない限り、SELinux を permissive モードにすることは推奨されていません。クラスターでの動作が必要となるアプリケーションが permissive モードをサポートしているか確認してください。SELinux の permissive モードは次の ARK でテスト済みです。: SAP MaxDB / Sybase / Oracle / DB2 / NFS / NAS / IP / FileSystem / MQ。必要となるパッケージについては Linux の依存関係 を参照してください。
  - disabled モード – SSP for Linux をインストールできます。
    - SELinux を無効にする方法は、OS 側のドキュメントを確認してください。
  - ご利用のディストリビューションから提供される適切なパッケージをインストールしてください。

- 既知の問題 を確認してください。– ご利用の環境に既知の問題がないか確認してください。

# SSP for Linux のインストール

LifeKeeper Single Server Protection (SSP) 構成内の各サーバーに LifeKeeper Single Server Protection ソフトウェアをインストールしてください。

LifeKeeper のセットアップスクリプトがパッケージマネージャー(yum もしくは zypper)を使用してパッケージをインストールするため、LifeKeeper が依存するパッケージは自動的にインストールされるようになりました。

> **重要** : LifeKeeper を正しくインストールするには、yum または zypper の機能構成が必要です。非機能構成では、インストールが失敗する可能性があります（詳細については、インストールに関する既知の問題 を参照してください）。さらに、インストールがハングする可能性があるため、パッケージマネージャーレポジトリまたは rpm データベースはロックしないでください。依存パッケージをパッケージマネージャーから自動的にインストールできない場合は、事前にLinux の依存関係を参照して必要なパッケージをインストールしてください。

LifeKeeper Single Server Protection のイメージファイル （`lkssp.img`） は、システムに LKSSP をインストールする際に必要な、ユーザ対話型システムセットアップ作業を実行するための一連のインストールスクリプトを提供しています（詳細については、対話モード を参照）。 非対話型インストールも実行できます（詳細については、非対話モード を参照）。

サーバーの Host ID または Entitlement ID を取得および表示するためのユーティリティを提供するライセンスユーティリティパッケージもインストールされます。 LKSSP を実行するための有効なライセンスを取得するために、Host ID または Entitlement ID、あるいはその両方が使用されます。

> **重要**: LifeKeeper Single Server Protection をインストールする前に、Linux の依存関係 トピックを参照してください。

インストール作業はセットアップスクリプトの操作 を参照してください。

アップグレード方法については、SSP のアップグレード を参照してください。

# 13.2.3. セットアップスクリプトの操作

LifeKeeper をインストールするために、セットアップスクリプトを使用して以下の作業を行います。

## セットアップスクリプトの機能

セットアップスクリプトは以下の2つの使用方法があります。

1. 対話型インストール
   メニュー画面からLifeKeeperの構成を設定して、インストールまで行います。
   このとき構成情報を保存すると、後述の非対話型インストールで使用できます。
2. 非対話型インストール
   保存した構成情報を用いて、LifeKeeperをインストールします。
   ユーザーへの問い合わせは発生しないので、構築ツール（Ansible）などから実行することも可能です。

## セットアップの手順

### 対話型インストール

1. root ユーザーでログイン後、次のコマンドを使用して、lkssp.img ファイルをマウントします。

   ```
   mount <PATH/IMAGE_NAME> <MOUNT_POINT> -t iso9660 -o loop
   ```

   ここで、PATH はイメージへのパスです
   IMAGE_NAME はイメージの名前です
   MOUNT_POINT はマウント位置へのパスです

2. lkssp.imgがマウントされたディレクトリに移動して、次のコマンドを入力します。

   ```
   ./setup [-s <response_file>]
   ```

   -s オプションを指定すると、設定した構成情報をresponse_fileに保存できます。このファイルは非対話型インストールで使用します。

   以下は、 **mount** および **setup** の代替コマンドです。

   **Mount Image**: `mount -o loop <PATH/IMAGE_NAME> /mnt/`

   **Run Setup**: `/mnt/setup`

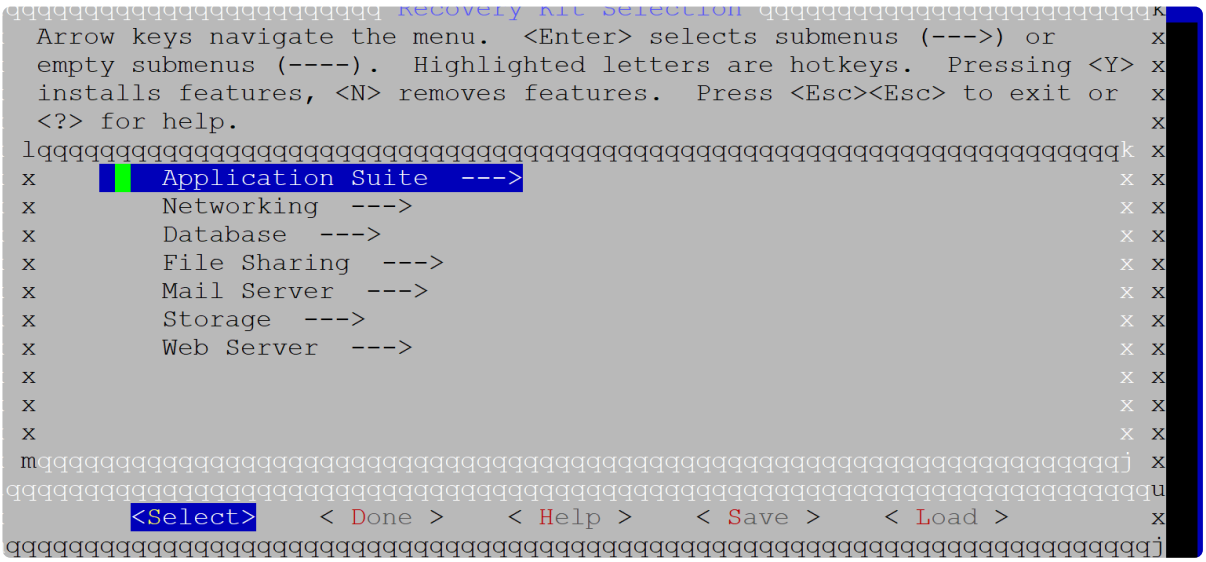
   ```
   [root@node2 ~]# mount -o loop /tmp/sps.img /mnt/
   mount: /mnt: WARNING: device write-protected, mounted read-only.
   [root@node2 ~]# /mnt/setup
   ```

3. スクリプトを実行すると、まず環境情報の収集を行います。

   インストールまたはアップグレードのシステム要件が満たされていない場合は、エラーメッセージが表示され、インストール/アップグレードはキャンセルされます。

   また、何らかの問題が発生した場合や設定の変更が必要な場合は、警告メッセージが表示され、インストールを続行するか中止するかをユーザーに決定するように求めます。

4. メインダイアログ画面からインストールする SSP 機能とアプリケーションリカバリキット (ARK) を選択します。




   詳細については ダイアログ画面の使い方 を参照してください。

5. インストールするSSPの機能やアプリケーションリカバリーキット（ARK）を選択し、 **<Done>** を選択すると、インストールが始まります。

   インストール完了後に何らかの通知が出力された場合は、その指示に従い必要な対応を行ってください。

## 非対話的インストール

1. root ユーザーでログイン後、次のコマンドを使用して、lkssp.img ファイルをマウントします。

   ```
   mount <PATH/IMAGE_NAME> <MOUNT_POINT> -t iso9660 -o loop
   ```

   ここで、PATH はイメージへのパスです
   IMAGE_NAME はイメージの名前です
   MOUNT_POINT はマウント位置へのパスです

2. LifeKeeperのインストールを行うシステムへ、構成情報ファイルをコピーした後、次のコマンドを実行します。

   ```
   ./setup -f <response_file> -q y
   ```

「-q y」オプションを指定すると、警告に対して了解したと渡します。

✻ ‘**sh setup**’ (bourne shell) は使用できません。代わりに **bash** を使用してください。

✻ **注記:** 非対話モードでのインストールで、-sオプションで保存した設定ファイルを使用する場合、そのファイルが使用されるシステムは、そのファイルが生成されたシステムと同じように構成されている必要があります。システムの違いが多すぎると、非対話モードでのインストールは失敗する可能性があります。
create_response_fileスクリプトで作成した設定ファイルはそのような制限がありません。

## 構成情報の作成

非対話型インストールで使用する構成情報ファイルは、`setup -s <response_file>`でセットアップ作業中に作成するか、create_response_fileスクリプトで事前に作成することが出来ます。

1. root ユーザーでログイン後、次のコマンドを使用して、lkssp.img ファイルをマウントします。

   ```
   mount <PATH/IMAGE_NAME> <MOUNT_POINT> -t iso9660 -o loop
   ```

   ここで、PATH はイメージへのパスです
   IMAGE_NAME はイメージの名前です
   MOUNT_POINT はマウント位置へのパスです

2. sps.img がマウントされたディレクトリに移動して、次のコマンドを入力します。

   ```
   ./create_response_file <response_file>
   ```

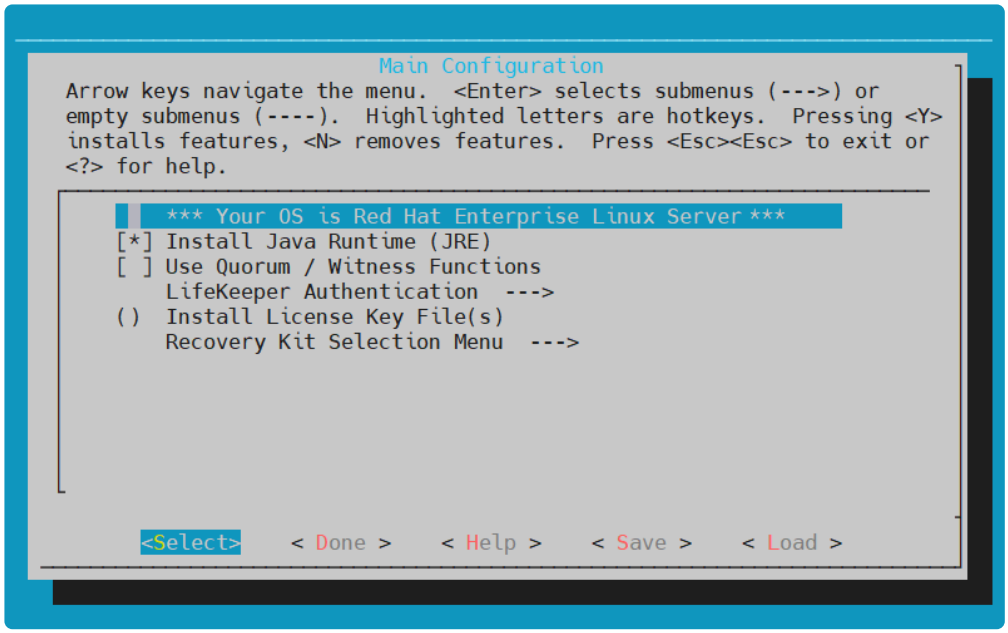
3. メニュー画面にてインストールするSPSの機能やアプリケーションリカバリーキット（ARK）を選択します。詳しくは「対話画面の操作」を参照してください。

4. インストールするSPSの機能やアプリケーションリカバリーキット（ARK）を選択し、 **<Done>** を選択すると、構成をresponse_fileへ保存してスクリプトが終了します。response_fileはインストール先のシステムへコピーして使用します。

   保存された構成情報を変更する場合は、再度手順2～4を実行してください。

## 対話画面の操作

以下のようなメニュー画面が表示されます。




以下のキーで項目選択を行います。

| | |
|---|---|
| ↑↓ | 選択項目の移動 |
| ← → | 最下行ボタンの移動 |
| ENTER | 選択したサブメニューを開く |
| Y / N / Enter | 選択した項目のオン、オフ、または反転 |

最下行ボタンは以下の動作を行います。

| | |
|---|---|
| Select | 選択項目の ON / OFF ・子画面へ移動 |
| Done | 現画面を終了して親画面に戻ります<br>Main Configuration 画面の場合は、メニュー画面を終了して、インストールの確認画面へ移行します |
| Help | 選択項目のヘルプ画面を表示します |
| Save | 設定した構成情報をファイルに保存します<br>保存した構成情報ファイルは非対話モードでのインストールに使用できます<br>(create_response_fileでは無効です) |
| Load | 保存済み構成情報ファイルの読み込み<br>(create_response_fileでは無効です) |

メニューボタン <Save> および <Load> をクリックすると、現在の設定を保存するとき、または保存した設定を読み込むときに使用する設定ファイル名を尋ねるダイアログ画面が表示されます。デフォルトのファイル名を変更したい場合は、[TAB] キーを使用してファイル名フィールドに移動し、新しいファイル名を入力してください。 **注記:** <Save> をクリックした場合、同じ名前でファイルを上書きする前に確認を求められます。

各項目では以下の機能を設定できます。

アップグレード中は、設定可能な項目のみが一覧表示されます。 ホットキー <Z> を使用すると、アップグレード時は変更されない項目が表示されます。

- **Install Java Runtime (JRE)**

  LifeKeeper GUI が使用する Java 実行環境をインストールします。

- **LifeKeeper Authentication**

  LifeKeeper SSP GUI へのログインを許可するユーザーとその権限を指定します。 複数のユーザー名を空白で区切って指定できます。 詳細は、 GUIユーザー設定 を参照してください。

- **Install License Key File(s)**

  LifeKeeper のライセンスファイルを指定します。ファイル名を入力できるダイアログに切り替わるので、ライセンスファイルのフルパス名を入力して下さい。スペース区切りで複数のライセンスファイルを指定することが出来ます。

  ライセンスの取得方法についてはライセンスの取得とインストール を参照してください。

- **Recovery Kit Selection Menu**

  インストールするアプリケーションリカバリーキットを選択します。 アプリケーションリカバリーキットは、機能に基づいていくつかのカテゴリーに分類されています。

  詳細については、アプリケーションリカバリーキットの分類 を参照してください。

- **LifeKeeper Startup After Install**

  インストールもしくはアップデート完了後に LifeKeeper を起動します。

## アプリケーションリカバリーキットの追加と削除

インストール完了後にアプリケーションリカバリーキットを追加するには、setup を実行し、[Recovery Kit] でリカバリーキットを選択し、次にアプリケーションリカバリーキットのカテゴリーを選択して、ご希望のキットを選択します。不要になったアプリケーションリカバリーキットの選択を解除すると、そのキットは削除されます。
ただし、使用中のリソースはキットを削除できないので、事前にリソースを削除してください。

## 修復インストール

LifeKeeper SSP インストールを修復するには、「`--force`」オプションを使用してセットアップを実行します。 修復インストールでは、失われたファイルや破損したファイルを置き換えてインストールを更新します。

# setup スクリプトのオプション

setup スクリプトは以下のオプションを指定して実行できます。

- `-f <response_file>`

  非対話モードでインストールを行います。<response_file>には、インストール中に使用する設定情報が含まれています。

- `-s <response_file>`

  メニューで選択した内容を含む設定ファイルを保存します。このファイルを「-f」オプションと一緒に使用して、同じ LifeKeeper 構成を別のシステムにインストールすることができます。たとえば、次のコマンドを実行します。

  ```
  ./setup -s <response_file>
  ```

  必要なパッケージとオプションを選択し、セットアップを完了します。

  その後、以下のコマンドを実行し、最初にセットアップを実行したときに選択したのと同じオプションで LifeKeeper のサイレントインストールを（別のシステムで）実行します。

  ```
  ./setup -f <response_file> -q y
  ```

- `--force`

  LifeKeeper SSP を強制的に再インストールします。

- `-q <y/n>`

  非対話モードでのインストール中に発生する可能性のある確認への回答を指定します。

# リカバリーキットの分類

| カテゴリー | 説明 |
|---|---|
| Application Suite | IBM MQ アプリケーションを保護するリカバリーキットが所属します |
| Database | Oracle、PostgreSQL、MaxDBなどのデータベースアプリケーションを保護するリカバリーキットが所属します |
| File Sharing | NFSやSambaなどのファイル共有サービスを保護するリカバリーキットが所属します |
| Mail Server | Postfixなどのメールサービスを保護するリカバリーキットが所属します |
| Storage | Network Attached Storage（NAS）など（これらに限らない）のデータ保管方法を保護するリカバリーキットが所属します |

| | |
|---|---|
| Web Server | Apache などのウェブサービスを保護するリカバリーキットが所属します |

# 13.2.4. SSP のアップグレード

LifeKeeper Single Server Protection (SSP) は、既存の階層を保ちながら将来のリリースにアップグレードすることができます。

> **注意:** 最新バージョンの LifeKeeper Single Server Protection にアップグレードできるのは、2 世代前までのバージョンです。それより前のバージョンからアップグレードする場合、古いバージョンをアンインストールし、LifeKeeper Single Server Protection を再インストールする必要があります。 旧バージョンをアンインストールする代わりに、旧バージョンを 1 世代前もしくは 2 世代前のバージョンにアップグレードした上で、最新バージョンにアップグレードする方法もあります。

> **注意:** アップグレード中に `lkbackup` を使用する場合の詳細については、lkbackup の既知の問題 を参照してください。

1. 必要な場合、SSP をアップグレードする前に、Linux オペレーティングシステムをアップグレードしてください。

2. セットアップスクリプトの操作 を参照してLifeKeeperをアップグレードします。

# 13.2.5. LifeKeeper Single Server Protection ライセンスについて

LifeKeeper Single Server Protection for Linux (SSP) では、サーバーごとに一意のライセンスが必要です。

ライセンスは、ランタイムライセンスです。つまり、SSP のインストールはライセンスなしでも可能ですが、正常に製品を起動して実行するためには、起動前にライセンスをインストールする必要があります。

# 13.2.5.1. ライセンスの取得方法

日本の SIOS テクノロジーから購入されたお客様はユーザーポータルの「ライセンス取得方法解説ページ」を参照ください。

# 13.2.5.2. ライセンスのインストール方法

クラスターの各ノードで以下の操作を実行します。

1. ライセンスをインストールします。

```
/opt/LifeKeeper/bin/lkkeyins <path to license file>
```

2. SSP を起動します。
   ライセンス情報を確認して、正しく登録されていることを確認します。
   正規ライセンスを適用している場合は、全てのライセンス情報が UNLICENSED、 または PERMANENT となります。

```
/opt/LifeKeeper/bin/typ_list -l
```

   a. UNLICENSED
      Core Application のライセンスが適用されている状態です。
   b. EVAL
      評価版ライセンスが適用されている状態です。
   c. PERMANENT
      オプション Recovery Kit のライセンスが適用されている状態です。
   d. NEEDED
      オプション Recovery Kit のライセンスが適用されていない状態です。

# 13.2.6. リソースポリシー管理

## 概要

LifeKeeper Single Server Protection (SSP) のリソースポリシー管理では、リソースのローカルリカバリとフェイルオーバの動作管理機能が提供されます。リソースポリシーは、 **lkpolicy** コマンドラインツール (CLI) を使用して管理できます。

## LifeKeeper SSP のリカバリ動作

LifeKeeper SSP には、個々のアプリケーションおよび関連し合うアプリケーションのグループを監視する機能があり、定期的にローカルリカバリを実行したり、保護下のアプリケーションに障害が発生したときに通知したりすることができます。関連し合うアプリケーションの例としては、主アプリケーションが下位のストレージまたはネットワークリソースに依存する階層などがあります。アプリケーションまたはリソースに障害が発生した場合のデフォルトの動作は以下の通りです。

1. **ローカルリカバリ:** 最初に、リソースまたはアプリケーションの **ローカル** でリカバリを試みます。このときは、外部の介入なしにローカルサーバ上でリソースまたはアプリケーションをリストアしようとします。ローカルリカバリが成功した場合、LifeKeeper SSP は追加のアクションを実行しません。

2. **フェイルオーバ:** 次に、ローカルリカバリでリソースまたはアプリケーションのリストアに *失敗* した (またはリソースを監視する *リカバリキット* がローカルリカバリをサポートしていない) 場合、 *フェイルオーバ* が開始されます (下記の標準ポリシーセクションの Failover を参照)。

リカバリ動作の詳細については、LifeKeeper Single Server Protection 障害検出とリカバリのシナリオ を参照してください。

## ポリシーによるカスタム動作およびメンテナンスモード動作

LifeKeeper SSP は、デフォルトのリカバリ動作を変更する追加ポリシーを設定する機能をサポートします。リソース単位またはサーバ単位で、4 つのポリシーが設定可能です (リソース単位のポリシーに関する注意については下のセクションを参照してください)。 **サーバレベルでポリシーを変更する方法を推奨します。**

利用可能なポリシーは以下の通りです。

### 標準ポリシー

- **Failover** - LifeKeeper SSP では、このポリシー設定を使用すると、リソースフェイルオーバを有効 / 無効にできます (これによって再起動されます)。

- **LocalRecovery** - LifeKeeper SSP は、デフォルトでは、フェイルオーバ (再起動) を実行する前に、個々のリソースまたは保護対象アプリケーション全体を再起動することにより、保護対象リソースのリカバリを試みます。このポリシー設定を使用すると、ローカルリカバリを有効 / 無効にできます。

- **TemporalRecovery** - 通常、LifeKeeper SSP は、障害リソースのローカルリカバリを実行します。ローカルリカバリに失敗すると、LifeKeeper SSP は再起動を実行します。ローカルリカバリに成功した場合は、フェイルオーバ (再起動) は実行されません。

ローカルリカバリに成功した場合でも、サーバの何らかの異常によって短時間の間にローカルリカバリが再試行される場合があり、結果として何度も連続してローカルリカバリが試行されることになります。これが発生すると、問題のアプリケーションは可用性が悪化します。

この反復的なローカルリカバリ / 障害サイクルを回避するために、時間的リカバリポリシーを設定できます。時間的リカバリポリシーを使用すると、管理者は指定した時間内に試行するローカルリカバリの回数を (成功かどうかにかかわらず) 制限することができます。

> ソースが試行するローカルリカバリの回数を 30 分間で 3 回に限定するポリシー定義をユーザが設定した場合、30 分以内に 3 回目のローカルリカバリが試行されると、LifeKeeper SSP はフェイルオーバ (再起動) を実行します。

定義した時間的リカバリポリシーは有効または無効にできます。時間的リカバリポリシーが無効の場合、時間的リカバリ処理は継続して実行され、ポリシーが適用されるはずの時間に通知がログに表示されますが、実際のアクションは実行されません。

**注記:** 時間的リカバリポリシーを設定した状態で、フェイルオーバとローカルリカバリの一方または両方を無効にすることは可能です。フェイルオーバまたはローカルリカバリを無効にした場合に、時間的リカバリポリシーは実行されることがないため、この状態は非論理的です。

### メタポリシー

「メタ」ポリシーは、他の複数のポリシーに影響を与える可能性があるポリシーです。通常、これらのポリシーは、標準ポリシーであれば複数個の設定が必要になるような特定のシステム動作を実現するためのショートカットとして使用します。

- **NotificationOnly** - このモードでは、管理者は LifeKeeper SSP を「監視専用」状態にすることができます。 **1 つのリソース (または、サーバ単位のポリシーの場合はすべてのリソース) の** ローカルリカバリ **および** フェイルオーバ (再起動) **の両方が影響を受けます。** 障害が検知されると、ユーザインターフェースには **Failure** 状態が表示されます。ただし、リカバリもフェイルオーバ (再起動) も実行されません。 **注記:** 管理者は、障害の原因となった問題を手動で修正し、障害が起きたリソースを復帰させて通常の LifeKeeper SSP の運用を継続する必要があります。

## リソースレベルのポリシーに関する重要な考慮事項

リソースレベルのポリシーとは、リソース階層全体またはサーバレベルのポリシーとは異なり、特定のリソースにのみ適用されるポリシーです。

> *例:*
>
> アプリケーション
>
> - IP

- file system

上記のリソース階層では、アプリケーションは IP とファイルシステムの両方に依存しています。ポリシーは、特定のリソースのローカルリカバリまたはフェイルオーバを無効にするように設定できます。これは、例えば、IP リソースのローカルリカバリが失敗し、IP リソースのフェイルオーバが無効に設定されていた場合、IP リソースはフェイルオーバを実行せず、他のリソースのフェイルオーバも発生させないことを意味します。ただし、ファイルシステムリソースのローカルリカバリが失敗し、ファイルシステムリソースのポリシーのフェイルオーバが無効化されていない場合、階層全体が再起動を伴うフェイルオーバを実行します。

**注記:** 重要事項として、リソースレベルのポリシーは設定対象の特定のリソースにのみ適用されることに注意してください。

上記は単純な例です。複雑な階層を構成することもできるため、リソースレベルのポリシーを設定するときは注意してください。

既知の問題を参照してください。 *(“フェイルオーバ時に誤った順序でリソースが停止されます“の項目)*

# lkpolicy ツール

`lkpolicy` ツールは、LifeKeeper SSP が稼働するサーバのポリシーを管理 (参照、設定、削除) するためのコマンドラインツールです。`lkpolicy`は、ポリシーの設定および修正、ポリシーの削除、利用可能なポリシーと現在の設定値の表示をサポートします。さらに、設定したポリシーは、有効または無効に設定できるため、リカバリ動作に影響を与えながらリソース / サーバ設定を保持できます。

全体的な使用方法は次の通りです。

```
lkpolicy [--list-policies | --get-policies | --set-policy | --remove-policy]
<name value pair data…>
```

`<name value pair data…>`は、運用方法および対象のポリシーによって異なります (特にポリシーを設定する場合)。 *たとえば、以下のようになります。* 有効 / 無効タイプのポリシーのほとんどでは、必要なのは `-on or --off` のスイッチのみですが、時間的ポリシーの場合は、しきい値を設定するための値も必要です。

## lkpolicy の使用方法の例

### ローカルおよびリモートサーバとの認証

`lkpolicy` ツールは、サーバが公開する API を通じて LifeKeeper SSP サーバと通信します。この API は、`lkpolicy`ツールなどのクライアントに対して認証を要求します。`lkpolicy` ツールで LifeKeeper SSP サーバに最初にアクセスしようとしたときに、そのサーバに対する認証情報がまだ保存されていない場合、ユーザは認証情報を求められます。認証情報はユーザ名とパスワードの形式であり、さらに以下の

条件があります。

1. クライアントには LifeKeeper SSP の管理者権限が必要です。したがって、そのユーザ名は、(PAM による) オペレーティングシステムの認証設定によって `lkadmin` グループに属する必要があります。必ずしも **root** で実行する **必要はありません** が、root ユーザはデフォルトで適切なグループに属しているため、root を使用することもできます。

2. 認証情報は *認証情報ストア* に保存されるため、ツールを使用してこのサーバにアクセスするたびに手動で認証情報を入力する必要はありません。

認証情報ストアと credstore ユーティリティによる管理の詳細については、SIOS Protection Suite の認証情報の設定 を参照してください。

`lkpolicy` によるセッションの例は以下のようになります。

```
[root@thor49 ~]# lkpolicy -l -d v6test4
Please enter your credentials for the system 'v6test4'.
Username: root
Password:
Confirm password:
Failover
LocalRecovery
TemporalRecovery
NotificationOnly
[root@thor49 ~]# lkpolicy -l -d v6test4
Failover
LocalRecovery
TemporalRecovery
NotificationOnly
[root@thor49 ~]#
```

### ポリシーのリスト表示

lkpolicy --list-policy-types

### 現在のポリシーの表示

lkpolicy --get-policies

lkpolicy --get-policies tag=*

lkpolicy --get-policies --verbose tag=mysql* # all resources starting with mysql

lkpolicy --get-policies tag=mytagonly

### ポリシーの設定

lkpolicy --set-policy Failover --off

lkpolicy --set-policy Failover --on tag=myresource

lkpolicy --set-policy Failover --on tag=*

lkpolicy --set-policy LocalRecovery --off tag=myresource

lkpolicy --set-policy NotificationOnly --on

lkpolicy --set-policy TemporalRecovery --on recoverylimit=5 period=15

lkpolicy --set-policy TemporalRecovery --on --force recoverylimit=5 period=10

### ポリシーの削除

lkpolicy --remove-policy Failover tag=steve

**注記:** *NotificationOnly* はポリシーのエイリアスです。 *NotificationOnly* を有効にすることは、対応する LocalRecovery および Failover ポリシーを無効にすることと同じです

# 13.2.7. LifeKeeper Single Server Protection のインストールの検証

LifeKeeper Single Server Protection パッケージが正常にインストールされたことを確認するには、コマンドラインで次のように入力してください。

```
rpm -V <package name>
```

**注記:** パッケージが正しくインストールされている場合、このコマンドは何も出力しません。

コマンドラインから照会を実行するには、次のように入力してください。

```
rpm -qi <package name>
```

**注記:** このコマンドの予想される出力は、パッケージ情報です。

# 13.3. LifeKeeper Single Server Protection テクニカルドキュメンテーション

## LifeKeeper Single Server Protection for Linux について

LifeKeeper Single Server Protection (SSP) は、単一ノード構成におけるアプリケーション監視を可能にします (つまり、クラスタの要件または制約はありません)。単一ノード環境は、物理的なものでも仮想 (vSphere、KVM) でも構いません。LifeKeeper SSP は、実績がある安定した SIOS LifeKeeper アーキテクチャ上に構築されます。LifeKeeper SSP は優れたアプリケーション監視機能を提供し、障害が発生したアプリケーションおよびシステムインフラストラクチャ項目 (例: NFS 共有、IP アドレス、ファイルシステム) のリカバリを実行することができます。何らかの理由でアプリケーションをリカバリできない場合、LifeKeeper SSP は、システムのリブートまたは VM とアプリケーション監視を設定された VMware 仮想マシンの VMware HA 再起動によって、ノードの再起動を開始します。

**注記:** LifeKeeper SSP は SIOS LifeKeeper 技術を使用して構築されているため、ドキュメント全体で LifeKeeper を参照します。また、両製品に共通するトピックについては SIOS Protection Suite for Linux ドキュメンテーションの情報を参照します。これらの共通のトピックを参照する場合、LifeKeeper SSP には以下の話題は適用されません。また、サポートもされません。

- クラスタリング
- コミュニケーションパス
- 共有ストレージ (要件、構成、...)
- リソース階層の拡張/拡張解除
- Quorum/Witness
- アプリケーションリカバリーキット
  - Recovery Kit for EC2
  - LVM Recovery Kit
  - MD Recovery Kit
  - Route53 Recovery Kit
  - SAP Recovery Kit
  - SAP HANA Recovery Kit
  - VMDK as Shared Storage Recovery Kit
- ストレージキット (DR、DMMP、HDLM、LVM、MD、PPATH、NEC SPS)

  **注記:** LifeKeeper SSP にベースとなるストレージキットがない場合、保護されるファイルシステムのマウントに必要なデバイスがシステム起動時にアクティベートされるようにするための手順が必要です (例: ファイルシステムが論理ボリューム上でマウントされる場合、LifeKeeper SSP が起動する前にボリュームがアクティブな状態になっていなければなりません)。

- リソース/マシンのフェイルオーバ (LifeKeeper SSP のデフォルトでは、これによってノードが再起動されます)
- リソースのスイッチオーバ
- 切り替え可能な IP アドレス (LifeKeeper SSP では、保護されるアプリケーションの一部には切り替え可能な IP アドレスが必要ですが、単一ノードしかないため、実際には切り替えは行われません)

LifeKeeper SSP のベースになっている SIOS LifeKeeper 製品の詳細については、共通するリリース番号の SIOS Protection Suite for Linux ドキュメンテーション を参照してください。このドキュメンテーションは、リソース階層の作成、リソースタイプ、状態と関係、グラフィカルユーザインターフェース (GUI)、および共通の作業と高度な作業に関する詳細情報を提供します。

# 13.3.1. ドキュメンテーションとトレーニング

SIOS LifeKeeper Single Server Protection for Linux のインストール、設定、管理、およびトラブルシューティングの方法を説明する関連ドキュメントは、SIOS Technology Corp. ウェブサイト の ドキュメンテーション セクションから参照できます。SIOS LifeKeeper Single Server Protection for Linux のあらゆる側面について、以下のセクションで説明しています。

| セクション | 説明 |
|---|---|
| はじめに および インストール | LifeKeeper Single Server Protection 環境のプランニングと設定、LifeKeeper Single Server Protection のインストールとライセンス、LifeKeeper のグラフィカルユーザインターフェース (GUI) の設定に役立つ情報を提供します。 |
| 管理 | サーバのプロパティの編集やリソースの作成などのサーバレベルの作業、およびリソースの編集、拡張、削除などのリソースレベルの作業について説明します。 |
| ユーザガイド | LifeKeeper GUI で実行できる多数の作業を含めて、LifeKeeper の GUI に関する詳細情報があります。 |
| トラブルシューティング | 既知の問題と制限について説明し、SIOS LifeKeeper Single Server Protection for Linux のインストール、設定、および使用を行うときに発生する可能性がある問題に対する解決策を説明します。 |
| Recovery Kit | LifeKeeper Single Server Protection で特定のアプリケーションを管理および制御するために必要なオプションの Recovery Kit のプランニングおよびインストール手順、管理、設定、およびユーザ情報が含まれます。 |

## トレーニング

LifeKeeper Single Server Protection のトレーニングは、SIOS Technology Corp. または代理店から受講可能です。詳細については、営業担当者にお問い合わせください。

## テクニカルサポート

SIOS Technology Corp. と有効なサポート契約を結んだお客様は、新しい SIOS Technology Corp. のセルフサービスサポートポータル にアクセスできます。

SIOS Technology Corp. のセルフサービスサポートポータル では、以下のことができます。

- 弊社のソリューションナレッジベースから、問題の解決策と質問に対する回答を検索する。
- 次のメニューを選択して、年中無休の SIOS Technology Corp. のサポートチームにアクセスする。
- **Log a Case** - 新しいインシデントを報告する。
- **View Cases** - お客様の未解決と解決済みのインシデントをすべて表示する。
- **Review Top Solutions** - 弊社のお客様が表示した、最も一般的な問題の解決策の情報を表示する。

セルフサービスポータルを設定してアカウントを有効にする方法については、SIOS Technology Corp. のサポート ( support@us.sios.com ) にお問い合わせください。

また、SIOS Technology Corp. のサポートには、以下の方法でも連絡できます。

1-877-457-5113 (通話料無料)

1-803-808-4270 (米国以外のお客様)

電子メール: support@us.sios.com

# 13.3.2. VMware HA との連携

「はじめに」セクションで説明したように、LifeKeeper Single Server Protection は、物理環境と仮想環境の両方で使用できるように設計されています。LifeKeeper SSP を VMware VM にインストールした場合、VMware の HA 機能を LifeKeeper SSP と組み合わせて、保護対象リソースやノードの障害を監視し、復旧することができます。 これらの機能を有効にする方法については、VMware HA と LifeKeeper Single Server Protection の連係の有効化 を参照してください。

# 13.3.3. LifeKeeper Single Server Protection の管理の概要

LifeKeeper Single Server Protection は操作時に管理を必要としません。LifeKeeper Single Server Protection は、保護されたリソースを監視し、障害が発生した場合に指定されたリカバリアクションを実行するように、自動的に機能します。以下のケースでは LifeKeeper Single Server Protection GUI を使用します。

- **リソースおよび階層の定義** 。LifeKeeper Single Server Protection は次のインターフェースオプションを提供します。
    - º LifeKeeper Single Server Protection GUI。
    - º LifeKeeper Single Server Protection コマンドラインインターフェース。
- **リソース監視。** LifeKeeper Single Server Protection GUI は、リソースステータス情報および LifeKeeper Single Server Protection ログへのアクセスを提供します。
- **手動での処理。** メンテナンスやその他の管理アクションのために、サーバまたは特定のリソースを停止することが必要になる場合があります。LifeKeeper Single Server Protection GUI には、特定のリソースを稼動させたり停止させたりすることができるメニュー機能が用意されています。アプリケーションが LifeKeeper Single Server Protection の保護下に置かれると、これらの LifeKeeper Single Server Protection のインターフェースを介してのみアプリケーションを起動および停止させることができます。LifeKeeper Single Server Protection の起動および停止は、コマンドラインを介してのみ行われます。

リソース階層の作成など、管理、設定、およびメンテナンス操作を実行する詳細な手順については、SPS for Linux ドキュメンテーションの管理作業 、GUI の作業 、およびメンテナンス作業 を参照してください。

※LifeKeeperが提供しているコマンド（実行可能なスクリプトやプログラム）を実行するには、スーパーユーザ権限が必要です。
su コマンドや sudo でスーパーユーザ権限を付与したユーザで、LifeKeeper のコマンドを実行することは可能ですが、SIOS Technology Corp では、root ユーザ以外で LifeKeeper のコマンドのテストはしていません。

# 13.3.3.1. VMware HA と LifeKeeper Single Server Protection の連携を有効にする

デフォルトでは、VMware VM 上にインストールした場合、LifeKeeper Single Server Protection と VMware HA の連携は無効になっています。連携を有効にするには、以下の手順が必要です。

1. LifeKeeper Single Server Protection VM に VMware Tools をインストールしてください。

2. */etc/default/LifeKeeper* を編集して、VMware HA 連携を調整する `HA_DISABLE` の値を 1 から 0 に変更します。

3. VMware HA のパラメーター *das.iostatsinterval* を 0 に設定してください。
   0 に設定すると、 仮想マシンの監視感度 に対する I/O 統計間隔が無効になります。(チェックが行われません。)デフォルトは120秒です。

4. LifeKeeper Single Server Protection を再起動します。LifeKeeper Single Server Protection が実行中の場合、 */etc/default/LifeKeeper* の上記の変更内容が検出されるように、停止してから再起動する必要があります。

# 13.3.3.2. VMware HA を有効化した障害検出およびリカバリシナリオ

アプリケーション内の問題を検出して通知する機能は、最適な総合的耐障害性ソリューションを構築する上で非常に重要です。すべての個々のアプリケーションは、障害発生のメカニズムと形式によって異なるため、一般的なメカニズムを示すことはできません。ただし、一般的に、多くのアプリケーションの設定は、LifeKeeper Single Server Protection に用意されている Core システムのエラー検出機能を利用することができます。このトピックでは、LifeKeeper Single Server Protection Core の機能について説明します。

アプリケーションに障害が発生したときに LifeKeeper Single Server Protection が障害を検出しリカバリを実行する仕組みを説明したリカバリシナリオを以下に示します。

1. LifeKeeper Single Server Protection は最初に、アプリケーションを再起動することでリカバリを試みます。

2. リカバリが成功した場合、アプリケーションは正常動作を継続します。

3. リカバリに失敗した場合、以下の処理が実行されます。

    a. LifeKeeper Single Server Protection が HA を有効 (/etc/default/LifeKeeper で HA_DISABLE=0) にした VMware ゲスト OS にインストールされている場合にリカバリに失敗すると、LifeKeeper Single Server Protection がアプリケーション監視インターフェースに送信するハートビートを抑制することで VMware HA がトリガされます。VMware HA はサーバを再起動することで応答します。

    b. LifeKeeper Single Server Protection が VMware ゲスト OS にインストールされていないか、HA を無効 (/etc/default/LifeKeeper で HA_DISABLE=1) にした VMware ゲスト OS にインストールされている場合にリカバリに失敗すると、システムが強制的に再起動されます。

# 13.3.3.2.1. LifeKeeper Single Server Protection で保護するシステムのメンテナンス

LifeKeeper Single Server Protection で保護されているサーバでシステムまたはアプリケーションのメンテナンスを実行するときには、LifeKeeper Single Server Protection による監視を停止するか、保護対象のリソースをメンテナンスモードにしてください。これにより、アプリケーションのリカバリ、および VMware HA の障害イベントのトリガが無効になるので、LifeKeeper Single Server Protection がシステムやアプリケーションのメンテナンス作業に干渉しなくなります。

LifeKeeper Single Server Protection を停止して再起動するには、以下の操作を実行してください。

1. **LifeKeeper Single Server Protection を停止します。 "/opt/LifeKeeper/bin/lkstop -f"** コマンドを使用して、LifeKeeper Single Server Protection を停止してください。リソースは継続して動作しますが、LifeKeeper Single Server Protection からは監視されなくなります。障害は手動で処理する必要があります。

2. **メンテナンスを実行します** 。必要なメンテナンスを実行します。

3. **SIOS LifeKeeper Single Server Protection を起動します。"/opt/LifeKeeper/bin/lkstart"** コマンドを使用して、LifeKeeper Single Server Protection を開始してください。リソースが保護されている状態になります。

**別の方法** - リソースをメンテナンス (「通知のみ」とも呼ばれる) モードにします。以下の操作を実行してください。

1. **リソースをメンテナンスモードにします。 /opt/LifeKeeper/bin/lkpolicy -s NotificationOnly --On** コマンドを使用してください。リソースは復旧されなくなり、VMware HA の障害イベントがトリガされなくなります。

2. **メンテナンスを実行します** 。必要なメンテナンスを実行します。

3. **メンテナンスモードをオフにします。 /opt/LifeKeeper/bin/lkpolicy -s NotificationOnly --Off** コマンドを使用してください。リソースが保護されている状態になります。

# 13.3.3.3. LifeKeeper Single Server Protection ハートビートと VMware HA

LifeKeeper Single Server Protection ハートビートは、保護対象のアプリケーションが正常であることを示すために、(VMware ゲスト OS で動作し、HA が有効である場合、10 秒ごとに) VMware HA に送信される信号です。アプリケーションで障害が発生すると、LifeKeeper Single Server Protection は最初にアプリケーションを復旧しようとします。復旧に失敗すると、LifeKeeper Single Server Protection はハートビートを抑制し、VMware HA に VM の再起動を指示します。

# 13.3.3.4. Quick Service Protection (QSP) Recovery Kit

QSP Recovery Kit は、OSのサービスを簡易的に保護する仕組みを提供します。OSのserviceコマンドで起動 (start) ・停止 (stop) ができるサービスであれば、容易にリソースが作成できます。Generic Applicationを使っても同じことが実現できますが、QSPを使えばコードの開発が不要です。また、依存関係を作成することで、他のリソースで保護しているアプリケーションと連携してサービスを起動･停止させることができます。

ただし、QSPのquickCheckでは簡易なチェックしか行っていません（serviceコマンドのstatusを利用しています）。実際にそのサービスが提供できているか、プロセスが動作しているかは保証されません。複雑な起動や停止処理、堅牢なチェックが必要な場合は、Generic Applicationの利用を検討してください。

詳細は以下のURLをご参照ください。

Quick Service Protection (QSP) Recovery Kit

# 13.3.3.5. LifeKeeper API for Monitoring

LifeKeeper API for Monitoringは、稼働中のLifeKeeperに対して問い合わせを行うことで、LifeKeeperノードの稼働状況や保護しているリソースの状況を取得できます。

本APIはlighttpd上でCGIとして動作し、lcdstatusコマンドで取得できる情報を、API経由で取得できるようにしたものです。本APIで取得できる情報は以下のとおりです。

- LifeKeeperノードの死活
- コミュニケーションパスの死活
- リソースの保護状況

詳細は以下のURLをご参照ください。

LifeKeeper API for Monitoring

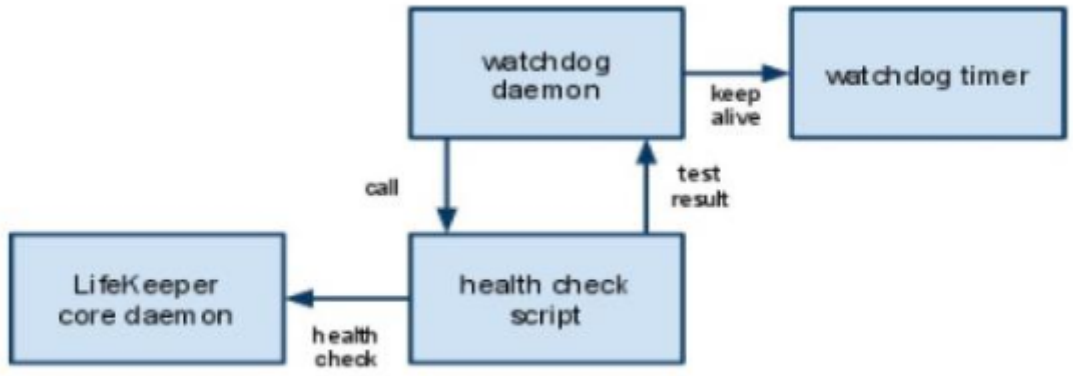
# 13.3.3.6. Watchdog

Watchdog は、サーバーが正常に動作しない場合に、問題の発生を予防する修正処置 (リブート) を確実に実行できるようにサーバーを監視する方法です。 Watchdog は、特別な Watchdog ハードウェアを使用して実装する場合と、ソフトウェアのみのオプションを使用して実装する場合があります。

> **注記:** この構成は、Red Hat Enterprise Linux Versions 7系とRed Hat Enterprise Linux Versions 8系 で検証されています。他のオペレーティングシステムでは検証されていないため、サポートされません。

## コンポーネント

- Watchdog timer - ソフトウェアドライバまたは外部ハードウェアコンポーネント
- Watchdog daemon - 該当する Linux ディストリビューションを通じて rpm が入手可能
- LifeKeeper Core daemon - LifeKeeper のインストールに付随
- health cehck script - LifeKeeper SSP core の動作状況を確認するスクリプト




**LifeKeeper Interoperability with Watchdog**

**LifeKeeper と Watchdog の相互運用性**

次のセクションを注意深く読んでください。デーモンは、エラーからリカバリするように設計されており、注意深く設定しないとデーモンはシステムをリセットします。インストールおよび設定を行う前に慎重に計画してください。このセクションの目的は、 Watchdog についての説明や設定をすることではありません。ここでは、 Watchdog 構成での LifeKeeper SSP との相互運用についての説明や設定のみ行います。

## 設定

以下の手順は、root ユーザ権限を持つ管理者が行う必要があります。管理者は、 Watchdog のリスクおよび問題についてすでに熟知しているものとします。

ヘルスチェックスクリプト (LifeKeeper 監視スクリプト `/opt/LifeKeeper/samples/watchdog/LifeKeeper-watchdog`) は、LifeKeeper の設定と Watchdog の設定を関連付けるコンポーネントです。このスクリプトは、LifeKeeper コアコンポーネントの基本部分を監視できます。

1. 以前に Watchdog を設定していた場合は、次のコマンドを入力して Watchdog を停止します。そうでない場合は、手順 2 に進みます。
   `systemctl stop watchdog`

2. Watchdog ソフトウェアのインストールで供給される Watchdog 設定ファイル (*/etc/watchdog.conf*) を編集します。

   a. test-binary を次のように修正します。

   ```
   test-binary = /opt/LifeKeeper/samples/watchdog/LifeKeeper-watchdog
   ```

   b. test-timeout を次のように修正します。

   ```
   test-timeout = 5
   ```

   c. interval を次のように修正します。

   ```
   interval = 7
   ```

   interval は、test-timeout 以上の値であることが必要です。また、あまりに長い間隔にすると障害検出が遅れてしまうので、5〜10程度を推奨します。

3. LifeKeeper SSP が起動していることを確認します。まだの場合は、LifeKeeper の起動 トピックを参照してください。

4. 次のコマンドを入力して Watchdog を起動します。
   `systemctl start watchdog`

5. 今後の再起動の際に Watchdog を自動的に起動させるには、次のコマンドを入力します。
   `systemctl enable watchdog`

> **注記:** Watchdog を設定すると、予想外のリブートがときどき発生する可能性があります。これは、 Watchdog の仕組みから来る一般的な性質です。正常に応答しないプロセスがあると、 Watchdog 機能は LifeKeeper (またはオペレーティングシステム) がハングしていると判断し、(警告なしに) システムをリブートします。

## アンインストール

LifeKeeper をアンインストールする場合は、慎重に行ってください。以下に列記の通り、上記の手順を逆の順で実行します。

> **警告:** LifeKeeper を構成する RPM パッケージを削除する方法で LifeKeeper をアンインストールする場合は、 **先に Watchdog を停止してください** 。上記の手順 2 では、LifeKeeper の Watchdog スクリプトを呼び出すように Watchdog 設定ファイルを修正しています。したがって、先に Watchdog を停止しておかないと、存在しないスクリプトを呼び出すことになります。リブートを実行するこのスクリプトが見つからない場合は、エラーが発生します。この状態は Watchdog を停止するまで継続します。

1. 次のコマンドを入力して Watchdog を停止します。
   ```
   systemctl stop watchdog
   ```

2. Watchdog ソフトウェアのインストールで供給される Watchdog 設定ファイル (*/etc/watchdog.conf*) を編集します。

   a. test-binary および interval の両エントリをコメントアウトします (各行の先頭に # を追加します)。

   ```
   #test-binary =
   #interval =
   ```

   **注記:** interval が他の機能によって以前から使用されていた場合は、そのままにしておくこともできます。

3. LifeKeeper をアンインストールします。LifeKeeper の削除 トピックを参照してください。

4. これで Watchdog を起動し直すことができます。LifeKeeper のみが Watchdog を使用していた場合は、次のコマンドを入力すると Watchdog を永続的に無効にできます。
   ```
   systemctl disable watchdog
   ```

# 13.3.3.7. LKCLI（LifeKeeper コマンドラインインターフェース）

LKCLI は、LifeKeeper の GUI で操作可能な機能をコマンドラインインターフェースで実現しています。また、GUI には無い機能として、コミュニケーションパスおよびリソース情報の export/import 機能があります。export/import 機能は、作成したシステムの量産や、テスト環境から本番環境のデプロイを容易に行える手段となります。

## サポート対象

次に示す環境での使用をサポート対象としています。

- OS：LifeKeeper 本体でサポートしている OS はすべてサポートされます。
- アプリケーションリカバリーキット： 後述する制限事項に該当するものを除き ARK 一覧 に記載されたアプリケーションリカバリーキットのみで構成された環境のみサポートされます。ノード内にサポート対象外の ARK リソースが作成されている場合、その環境はサポート対象外となります。

サポート対象外の環境でのコマンドライン操作をご検討されているお客様は、「 コマンドラインインターフェース 」の使用をご検討ください。

## 制限事項

Single Server Protection for Linux では、以下関連する機能については適用されません。またサポートもされません。

- 共通オプション
  - [ --remote <str> ]
- コミュニケーションパスの操作
  - lkcli commpath …
  - lkcli import（コミュニケーションパスの作成のみ適用されません。）
- リソースの拡張/拡張解除や切り替えに関する操作
  - lkcli resource extend
  - lkcli resource unextend
  - lkcli resource eqv
  - lkcli resource reorder-priority
  - lkcli resource switchback
  - lkcli server …
- アプリケーションリカバリーキット
  - SIOS DataKeeper for Linux
  - Recovery Kit for EC2
  - LVM Recovery Kit
  - MD Recovery Kit
  - Route53 Recovery Kit
  - SAP Recovery Kit
  - SAP HANA Recovery Kit

    - VMDK as Shared Storage Recovery Kit
    - Multipath Recovery Kit（DMMP / HDLM / PPATH / NECSPS）
- IP Recovery Kit のlkcli export または lkcli resource info コマンド
  lkcli export コマンド、 lkcli resource info コマンドで処理が停止してしまう場合は、ホスト名が正しく設定されていることを確認してください。

Single Server Protection for Linux でも、SIOS Protection Suite for Linux と同様に LKCLI を使用できます。詳細については「 LKCLI（LifeKeeper コマンドラインインターフェース） 」を参照してください。

# 13.3.4. トラブルシューティング

このセクションには、LifeKeeper Single Server Protection の制限または既知の問題が記載されています。

# 13.3.4.1. 既知の問題と回避策

下記に、LifeKeeper Single Server Protection で明らかになっている制限または既知の問題を示します。

## Core

バグ **2257**

**LifeKeeper Single Server Protection** および **SIOS Protection Suite** のノードに

**credstore** 経由でアクセスするときに、正しい **credstore** キーが必要です

**解決方法:** `credstore` を使用して、LifeKeeper Single Server Protection または SIOS Protection Suite のノードの認証情報を保存するときに、credstore の認証情報キーについてホスト名の正しい形式 ( `credstore -k <hostname>`) を使用する必要があります。

LifeKeeper Single Server Protection プラグインの場合、LifeKeeper Single Server Protection プラグイン画面の **[Hostname:]** フィールドに表示されるシステムのホスト名を使用して `credstore` を実行する必要があります。

SIOS Protection Suite の場合、認証情報の保存に使用するホスト名は、コマンドラインツール (`lkpolicy` など) の -d 引数に使用するものと同じである必要があります。例えば、**`lkpolicy -d mynode1`**, を実行する場合、**`credstore -k mynode1`** を使用して認証情報を保存する必要があります。この場合、認証情報の保存に FQDN を使用することはできません。FQDN を使用する場合は、**`lkpolicy -d FQDN`** を実行する必要があります。

**対応策:** LifeKeeper Single Server Protection または SIOS Protection Suite のすべてのノードで機能するデフォルトの認証情報セット (**`credstore -k default`**) を保存した場合、この問題の影響を受けることはありません。

---

バグ **2408**

**HA** ハートビートが不正に有効になります

2 番目のリソースに障害が発生した後、lkvmhad が HA ハートビートを不正に有効にします。

**対応策:** */etc/default/LifeKeeper* の `LKCHECKINTERVAL` に、VMware HA の [VM Monitoring Failure Interval] (VM の障害監視間隔) よりも大きい値を設定してください。 **注記:** `LKCHECKINTERVAL` のデフォルト値は 120 秒です。また、これは、VMware HA の VM 監視の監視感度「low」のデフォルト値でもあります。

リソースを停止し、その後**LifeKeeper**を停止して再起動すると、そのリソースは再開されます。

LifeKeeper Single Server Protection の起動時、LifeKeeper を停止する前にすでに停止していたリソースも再び起動されます。

# GUI

**LifeKeeper Single Server Protection GUI の更新の問題**

GUI のリソースツリーが不規則に表示されることがあります (リソースの依存関係が正しく表示されない)。

**対応策**: GUI の更新を実行してください。

**注記:** lkoper、lkguest でGUI画面を更新できない

# Apache

**Apache リソースの作成に失敗する**

エラーメッセージの例：

Error: valid_http_root: Since "/usr/sbin/httpd" is shareable on "/usr", "/etc/httpd" must be also

原因：

不具合のため、マウントポイント"/"(root)にあるファイルを正しく検索できません。

例えば、マウントポイント"/"と同じファイルシステムに/etc/httpdがある場合、リソースの作成に失敗します。

**対応策:**

次のいずれかのようにマウントすることにより、エラーを回避することができます。

(a) /etc/httpdなどを別のマウントポイント以下に移動する

(b) /etcを/dev/sdb1などにマウントする

# Oracle

バグ **2387**

**LifeKeeper Single Server Protection 環境の root ファイルシステムに、Oracle 階層を作成できません。**

**対応策:** 以下の手順に従って、Oracle を新しいファイルシステムにコピーしてください。

Oracle データを格納できる十分に大きい新しいディスク (例: /dev/sdb) を作成してください。(注記: ディスクの大きさを見積もるために、/oracle ディレクトリの大きさを参考にできます。ログを含めるために、50 % 以上増加してください)。

gdisk を使用して、そのディスクに新しいパーティションを作成してください。

```
gdisk /dev/sdb
```

ファイルシステムを作成してください。

mkfs -t ext3 /dev/sdb1

このファイルシステムをマウントしてください (この例では /mnt/oracle を使用)。

mkdir /mnt/oracle

mount /dev/sdb1 /mnt/oracle

Oracle と Listener を停止してください。

Oracle を新しいファイルシステムにコピーしてください。

cd /oracle

cp -a * /mnt/oracle

(注記: データ量により、この手順には時間がかかることがあります)

新しいファイルシステムをアンマウントしてください。

umount /mnt/oracle

新しいファイルシステムを /oracle にマウントしてください。

mount /dev/sdb1 /oracle

Listener を開始し、次に Oracle を開始してください。

**Oracle Recovery Kitは、AWS EC2上のOracle Database Standard Edition 2 (SE2)をサポートしません。**

AWS EC2上のシステムにおいてOracle Database Standard Edition 2 (SE2)のテストを行った際に、幾つかの奇妙な振る舞いが確認されました。しかしながら、EC2以外のシステムにおいては正常動作が確認できておりますので、EC2以外のシステムにおいてAWS EC2上のOracle Database Standard Edition 2 (SE2)の利用をサポートします。

## SAP

バグ **2388**

**SAP の場合、GUI を使用して階層を作成することはできません。**

**対応策**: コマンドラインオプションを使用して、階層を作成してください。ただし、以下に示すように、コマンドラインの最後に番号 76 を指定してください。

- $LKROOT/lkadm/subsys/appsuite/sap/bin/create <primary sys> <tag> <SAP SID> <SAP Instance> <switchback type> <IP Tag> <Protection Level> <Recovery Level> <Additional SAP Dependents> 76

コマンドラインの詳細については、「コマンドラインによる SAP の設定」を参照してください。

SIOS Protection Suite for Linuxにおける「既知の問題と制限」も参照ください。

# 13.4. アプリケーションリカバリーキット

LifeKeeper Single Server Protection for Linux Application Recovery Kits (ARKs) は特定のアプリケーションをSSP上で管理、制御するためのツールとユーティリティを含んでいます。今バージョンのSSPで利用できるARKは以下の通りです。

> **重要** ：SIOS は次の Recovery Kit のサポートを終了します。これらは LifeKeeper for Linux の次のメジャーリリースでサポートされなくなります。
> (なお、新規受注は2022年12月31日をもって終了しています。)
> • NEC iStorage StoragePathSavior
> • Samba
> • Postfix
> • HULFT HUB Recovery Kit

Apache Recovery Kit 管理ガイド

DB2 Recovery Kit 管理ガイド

IP Recovery Kit 管理ガイド

MySQL Recovery Kit 管理ガイド

WebSphere MQ Recovery Kit 管理ガイド

NAS Recovery Kit 管理ガイド

NFS Recovery Kit 管理ガイド

Recovery Kit for Oracle Cloud Infrastructure 管理ガイド

Oracle Recovery Kit 管理ガイド

PostgreSQL Recovery Kit 管理ガイド

Postfix Recovery Kit 管理ガイド

Samba Recovery Kit 管理ガイド

SAP MaxDB Recovery Kit 管理ガイド

Sybase Recovery Kit 管理ガイド

# 14. プロダクトライフサイクル

LifeKeeper/DataKeeper の各バージョンにおけるプロダクトライフサイクルについては、こちら を参照してください