ORACLE

Oracle Exadata Cloudを使用したディザスタ・リカバリ

Oracle Exadata Cloud ServiceまたはGen 2 Exadata Cloud at Customerにおけるオンプレミスのプライマリからスタンバイまで

2020年3月6日

機密情報：公開ドキュメント

免責事項

本文書には、ソフトウェアや印刷物など、いかなる形式のものも含め、オラクルの独占的な所有物である占有情報が含まれます。この機密文書へのアクセスと使用は、締結および遵守に同意したOracle Software License and Service Agreementの諸条件に従うものとします。本文書と本文書に含まれる情報は、オラクルの事前の書面による同意なしに、公開、複製、再作成、またはオラクルの外部に配布することはできません。本文書は、ライセンス契約の一部ではありません。また、オラクル、オラクルの子会社または関連会社との契約に組み込むことはできません。

本書は情報提供のみを目的としており、記載した製品機能の実装およびアップグレードの計画を支援することのみを意図しています。マテリアルやコード、機能の提供をコミットメント（確約）するものではなく、購買を決定する際の判断材料になさらないでください。本書に記載されている機能の開発、リリース、および時期については、弊社の裁量により決定されます。

製品アーキテクチャの性質上、コードが大幅に不安定化するリスクなしに、本書に記載されているすべての機能を安全に含めることができない場合があります。

目次

はじめに

Oracle Maximum Availability Architecture（Oracle MAA）は、プライベート・クラウドやパブリック・クラウドにデプロイされたOracleデータベースのデータを保護し、可用性を高めるためのベスト・プラクティス構想です。Oracle Data GuardおよびOracle Active Data Guardにより、単なるバックアップからのリストアでは達成することのできないリカバリ時間目標（RTO）とリカバリ・ポイント目標（RPO）が設定されたデータベースのディザスタ・リカバリ（DR）に対応できます。これらのソリューションを利用して本番データベース（プライマリ・データベース）の同期レプリカ（スタンバイ・データベース）を物理的に離れた場所に1つ以上デプロイし、ミッション・クリティカルなデータの高可用性、包括的なデータ保護、およびディザスタ・リカバリを実現します。

有効なディザスタ・リカバリ計画には、リモート・データセンターの設置、装置の整備、および管理が関係しており、かなりのコストがかかる可能性があります。Oracle Exadata Cloud（Exadata Cloud ServiceとExadata Cloud at Customerの両方を含む）はスタンバイ・データベースをホスティングするための優れた代替手段であり、DRサイトがなかったり、リモート・データセンターの管理に伴うコストや煩雑さを避けたいお客様のために用意されました。既存の本番データベースはオンプレミスに残し、DR用のスタンバイ・データベースをOracle Exadata Cloud上にデプロイします。このデプロイメント・モードは、一般にハイブリッドData Guard実装と呼ばれます。

お客様は、要件に応じてOracle Exadata CloudにData GuardまたはActive Data Guardのどちらかのスタンバイをデプロイする方法を選択できます。ハイブリッドData Guard構成に関しては特有の考慮事項がいくつかありますが、他のData Guardデプロイメントの場合と同じOracle MAAのベスト・プラクティスに従っています。このOracle MAA構想では、Oracle MAAのベスト・プラクティスについて詳しく述べ、Exadata Cloud Service（ExaCS）とExadata Cloud at Customer Gen2（ExaCC）を使用してOracle Cloud上にDRをデプロイする場合の手順の概要を示します。このホワイト・ペーパーの対象読者は、Oracle Database、Data GuardまたはActive Data Guard、およびOracle Databaseのバックアップとリカバリに関する知識を持つ技術者です。またこのホワイト・ペーパーでは、Oracle Cloudによって提供されるサービスについての基本的な理解があることも前提としています。

Data GuardおよびActive Data GuardによるExadataCloudへのディザスタ・リカバリ

Oracle Cloudは、お客様の特定の要件に合わせて調整された、Platform as a Service（PaaS）、Infrastructure as a Service（IaaS）、Software as a Service（SaaS）などの広範なクラウド・サービスを提供します。オンプレミス・システムのディザスタ・リカバリ（DR）は、Exadata Cloud Service（ExaCS）またはExadata Cloud at Customer（ExaCC）を使用して構成できます。

Data Guardは、Oracle Database Enterprise Editionに付属しており（オンプレミス・システムの場合は、別個のライセンスが不要）、Exadata Cloud ServiceとExadata Cloud at Customerのすべてのエディション（Enterprise、High Performance、Extreme Performance）でサポートされています。Oracle Database Standard EditionはData Guardをサポートしていないことに留意してください。Enterprise EditionとStandard Editionでサポートされる機能の理解に役立つサポート対応マトリックスを確認するには、選択したバージョンのOracle Databaseドキュメントで“ライセンス情報”を検索してください。

Active Data GuardはData Guardの機能を拡張したもので、自動ブロック修復などのデータ保護と可用性のための高度な機能に加えて、読取り専用ワークロードのオフロード機能と本番データベースからの高速増分バックアップ機能も搭載しています。本書で"Data Guard"と述べる場合、この用語は、Active Data GuardとともにData Guardに適用されることをご了承ください。Active Data Guardは、Oracle Database Extreme Performance Edition、Exadata Cloud at Customer、Exadata Cloud Serviceに組み込まれています。ハイブリッド構成で使用する場合は、オンプレミス・システムでもActive Data Guardのライセンスを取得する必要があります。

本書で説明するプロセスは、Exadata Cloud Service（ExaCS）とExadata Cloud at Customer（ExaCC）Gen2の場合に有効です。ExaCC Gen 1でのHybrid Data Guardのデプロイ・プロセスについては、「Hybrid Data Guard to ExaCC」で説明されています。

クラウドでのHybrid Data Guardの利点

Oracle CloudでハイブリッドData Guard構成を使用することには多くの利点があります。おもな利点を以下に示します。

1. クラウド・データセンターとインフラストラクチャはオラクルが管理する。
2. CPUリソースのスケーリング、パッチ適用、バックアップ/リカバリなどの基本的なシステム・ライフサイクル操作をOracle Cloudで実行できる。
3. Oracle Data Guardには、ディザスタ・リカバリ機能、データ保護機能、アクティビティをオフロードして利用率と投資回収率を向上させる機能がある。
4. ハイブリッドData Guard構成により、計画メンテナンスや計画外停止の期間に、本番データベースをクラウドのスタンバイ・データベースにスイッチオーバー（計画イベント）またはフェイルオーバー（計画外イベント）する機能がある。障害が発生したオンプレミス・データベースの修復が完了したら、このデータベースをクラウド内の現在の本番データベースと再同期させ、その後本番データベースをオンプレミス・データベースにスイッチバックすることができます。
5. オンプレミスのデプロイメントと同じOracle MAAのベスト・プラクティスを利用する。このホワイト・ペーパーで説明されている、ハイブリッドData Guardのデプロイメントに固有のOracle MAAの追加ベスト・プラクティスを使用します。
    a. MAAプラクティスに従って構成されている場合、ハイブリッドData Guard構成により以下を実現できます。
        i. Data Guardファスト・スタート・フェイルオーバーで構成されている場合に、自動フェイルオーバーにより秒単位のリカバリ時間目標（RTO）、および
        ii. ASYNC転送が可能なData Guardの場合は1秒未満のリカバリ・ポイント目標（RPO）、または
        iii. SYNCまたはFAR SYNC構成のData Guardの場合はRPOゼロ

Hybrid Data Guard構成の場合、Data Guardライフサイクル管理のスイッチオーバー、フェイルオーバー、復旧などは手動プロセスです。

ディザスタ・リカバリ用にExadata Cloudをデプロイする場合、MAAでは以下が推奨されます。

1. オンプレミスのプライマリ・データベースと対称かまたは類似のDBシステム・ターゲットを作成し、ロールの移行後にパフォーマンスSLAが確実に満たされるようにします。RACにはRACを、ExadataにはExadataを使用するなどです。
2. ネットワーク帯域幅を、既存のネットワーク・トラフィックに加えてピークREDO速度を確実に処理できる容量にします。My Oracle Supportドキュメント2064368.1には、Data GuardおよびRMANのネットワーク・パフォーマンスを評価および調整するための補足的なネットワーク帯域幅トラブルシューティング指針が記載されています
3. オンプレミスとExadata Cloud環境間のネットワークの信頼性とセキュリティを確保します。
4. 自動ブロック修復、データ保護、オフロードによる追加の利点を得るため、Active Data Guardを使用します。
5. オラクルの透過的データ暗号化（TDE）をプライマリとスタンバイの両方のデータベースで使用します。My Oracle Supportドキュメント 2359020.1には、クラウド構成におけるTDEの動作についての補足的な詳細が記載されています。

オブジェクト・ストアへのバックアップは、公開時点のスタンバイ・データベースではサポートされていませんでした。ただし、クラウド・データベースがプライマリになった場合に、スタンバイ・データベースのバックアップ機能が使用可能になるのに備えて、クラウド・データベースでバックアップを構成することをお奨めします。

スタンバイ・データベースの使用による計画メンテナンス中の停止時間の短縮

Exadata Cloud上のスタンバイ・データベースを使用し、いくつかの方法でプライマリ本番データベースの計画停止時間を短縮できます。

Standby-First Patch Apply

徹底的に検証するため、多くのパッチをまずフィジカル・スタンバイ・データベースに適用する場合があります。停止時間を最小限に抑えるために、まずスタンバイにパッチを適用し、次に本番データベースをスタンバイ・データベースに切り替え、それから元のプライマリ・データベースにパッチを適用する作業が行われることがよくあります。プライマリとスタンバイがReal Application Cluster（RAC）用に有効化されており、ソフトウェア更新がRACローリングである場合にはスイッチオーバーが不要ですが、

それでもStandby-Firstのソフトウェアを更新して検証および保護機能を強化することをお奨めします。Standby-First対象のパッチ（パッチのreadmeに記載）を適用する場合、異なるパッチ・バージョンで稼働しているプライマリとスタンバイの間のData Guardによる物理レプリケーションがサポートされます。お客様は、プライマリとスタンバイの間で一定期間に渡り異なるバージョンのパッチを混在させ、パッチに関して予期せぬ問題が発生した場合に、パッチ未適用のバージョンに高速でフォールバックできるようにする方法を選択することも可能です。Standby-Firstプロセスを対象とするパッチの詳細については、My Oracle Support Note 1265700.1『Oracle Patch Assurance - Data Guard Standby-First Patch Apply』を参照してください。

データベースのローリング・アップグレード

Exadata Cloudでのスタンバイのための別の有益なユースケースはデータベースのローリング・アップグレードの場合で、Standby-First互換ではない新しいOracle Databaseへのアップグレード時に停止時間を短縮できます。Oracle 11gとOracle 12cで使用される一時ロジカル・プロセスでは、フィジカル・スタンバイ・データベースを一時的にロジカル・スタンバイに変換し、ロジカル・スタンバイを新しいバージョンにアップグレードして検証し、準備ができたらData Guardスイッチオーバーを実行します。スイッチオーバーが完了すると、元のプライマリ・データベースは、同様に新しいリリースで動作している同期されたフィジカル・スタンバイに変換されます。12.1.0.1以前のリリースからローリング・アップグレードについて詳しくは、「データベース・ローリング・アップグレード」を参照してください。12.1.0.2リリース以後は、Data Guard環境をアップグレードするために、スタンバイ・データベースを使用したさらに効率的なデータベース・ローリング・アップグレード・プロセスが用意されています。Oracle Data Guardのドキュメントの「Automated Database Upgrades using Oracle Active Data Guard and DBMS_ROLLING」および「Using DBMS_ROLLING to Perform a Rolling Upgrade」のセクションを参照してください。

サービス・レベル要件

Hybrid Data Guardデプロイメントは、定義上はユーザー管理環境です。管理者は、可用性、データ保護、およびパフォーマンスについて、所定の構成およびアプリケーションの場合に期待する実際的なサービス・レベルを決める必要があります。サービス・レベルは、すべてのData Guard構成に適用される、ディザスタ・リカバリに関係した次の3つの次元ごとに設定する必要があります。

- **リカバリ時間目標（RTO）**は、停止した場合の最大許容停止時間を表します。これには、停止を検出し、データベースとアプリケーションの両方の接続をフェイルオーバーしてサービスが再開されるまでに必要な時間が含まれます。
- **リカバリ・ポイント目標（RPO）**は、許容可能な最大データ損失量を表します。望ましいRPOを達成できるかどうかは、次の要因に左右されます。
  - ネットワークのデータ量に対して使用可能な帯域幅
  - 信頼性が高く中断することのない送信を実現するネットワークの能力
  - Data Guardで使用される転送方式（データ損失をほぼゼロに抑える非同期方式、またはデータ損失をゼロに抑える同期方式のいずれか）
- **データ保護：**Active Data GuardとMAA構成により、ユーザーはもっとも包括的なブロック破損の検出、防止、自動修復を構成できます。
- **パフォーマンス：**スタンバイ・システムでプロビジョニングされているコンピュート、メモリ、I/Oなどの能力がオンプレミスの本番システムと比較して劣っていると、データベースの応答時間がフェイルオーバー後に違ってくる可能性があります。この状態は、管理者がコストを削減するためにスタンバイのリソースを意図的に不十分に構成し、DRモード中はサービス・レベルが低下することを容認している場合に発生します。Oracle MAAのベスト・プラクティスでは、プライマリとスタンバイの両方を同容量のリソースで構成し、フェイルオーバー後も応答時間が変わらないようにすることを推奨しています。クラウドで使用可能な高速プロビジョニングにより、安定状態中はデプロイされる容量が減らされても、フェイルオーバーが必要になった場合には新しいプライマリが急速にスケールアップされる、中間的な構成が可能です。

高速プロビジョニング・アプローチで安定状態の期間に減らされるリソースは、スタンバイ・データベースを最新の状態で維持するリカバリの能力に影響し、そのため適用ラグが作成され、RTOに影響を与える可能性があります。このアプローチは、徹底的なテストの後にのみ考慮してください。

セキュリティ要件

Oracle MAAのベスト・プラクティスでは、オラクルの透過的データ暗号化（TDE）を使用してプライマリ・データベースとスタンバイ・データベースの両方を暗号化することにより、静止状態でデータを確実に暗号化するよう推奨しています。データは移行プロセス中に変換可能ですが、移行前にTDEに変換してもっとも安全なData Guard環境を実現することを推奨されています。詳しくは、『Oracle Database Tablespace Encryption Behavior in Oracle Cloud』（ドキュメントID：2359020.1）を参照してください。TDEによって暗号化されない任意の他のデータベース・ペイロード（データファイルやREDOヘッダーなど）の移動時暗号化には、VPN接続またはOracle Net暗号化も必要です。

TDEを使用してデータを保護することは、システムのセキュリティを強化する上で重要です。ただしユーザーは、いずれの暗号化ソリューションを使用する場合であっても、以下に示す特定の考慮事項があることを認識している必要があります。

- CPUオーバーヘッドの増大：暗号化では、暗号化された値と復号された値を計算するための追加のCPUサイクルが必要となります。ただしTDEは、データベースのキャッシング機能を利用し、Oracle Exadata内のハードウェア・アクセラレーションを利用することによって、オーバーヘッドが最小限に抑えられるように最適化されています。大半のTDEユーザーは、TDEを有効化した後の本番システムのパフォーマンスへの影響がほとんどないことを認めています。パフォーマンスのオーバーヘッドに関して詳しくは、『Oracle Database Advanced Securityガイド』を参照してください。
- データ圧縮率の低下：暗号化されたデータでは、元の平文データの情報が一切開示されないようにする必要があるため、圧縮率が低くなります。そのため、TDEで暗号化されたデータの圧縮率は、どのような圧縮方式を適用した場合でも低くなります。したがって、TDE暗号化を使用している場合、REDO転送圧縮を使用することは奨められません。ただし、Oracle Advanced CompressionやHybrid Columnar CompressionといったOracle Database圧縮テクノロジーとTDEを併用する場合は、暗号化の前に圧縮が実行されるため、圧縮と暗号化の両方のメリットが得られます。
- 鍵の管理：暗号化の強度は、暗号化に使用される鍵の強度によって決まります。さらに、暗号化鍵を失うことは、その鍵によって保護されているすべてのデータを失うことと同じです。暗号化が有効になっているデータベースの数が少ない場合は、鍵とそのライフサイクルを比較的簡単に追跡できます。しかし、暗号化されるデータベースの数が増えれば、鍵の管理もより難しくなります。暗号化データベースを多数使用する場合は、オンプレミスでOracle Key Vault7を使用してTDEマスター鍵を保管および管理することを推奨します。

  ベスト・プラクティスが適用されていると仮定するとき、オンプレミス・データベースがTDEでまだ有効になっていない場合は、マスター・ノート『Master Note For Transparent Data Encryption（TDE）』（ドキュメントID：1228046.1）に従って、TDEを有効にし、ウォレット・ファイルを作成してください。

TDEがオンプレミスで不要な場合は、Data Guard構成における暗号化および非暗号化データベースの暗号化動作の詳細について、My Oracle Support Note 2359020.1 を参照してください。

データベース、OS環境、およびネットワークの前提条件

表1：前提条件

	オンプレミス	Oracle Cloud（ExaCSおよびExaCC Gen 2）
オペレーティング・システム	Linux、Windows、またはSolaris X86（Data Guardのクロス・プラットフォームの互換性については、My Oracle Support Note 413484.1を参照）	Oracle Enterprise Linux（64ビット）
Oracle Database*	» *Oracle Database Enterprise Edition 11.2.0.4（64ビット）* » *Oracle Database Enterprise Edition 12.1.0.2（64ビット）* » *Oracle Database Enterprise Edition 12.2.0.1（64ビット）* » *Oracle Database Enterprise Edition 18c（64ビット）* » *Oracle Database Enterprise Edition 19c（64ビット）*	卓越したパフォーマンス/BYOL
Oracle RAC	RACまたは非RAC	RACまたは非RAC
Oracle Multitenant	12.1以上では、プライマリ・データベースはCDB/PDBデータベースとする必要がある。	マルチテナント・データベース 非CDB
物理/仮想	物理または仮想	Exadata Virtual
データベース・サイズ	あらゆるサイズ	あらゆるサイズ。シェイプの制限については、Exadata Cloudのドキュメントを参照
TDE暗号化	推奨	クラウド・データベースでは必須

** プライマリおよびスタンバイのデータベース上のOracle Databaseバージョンは、初期インスタンス化の間、一致している必要があります。Standby-Firstと互換性があるデータベース・ソフトウェアの更新の場合、プライマリ・データベースとスタンバイ・データベースのOracleホーム・ソフトウェアは同じである必要はありません。『Oracle Patch Assurance - Data Guard Standby-First Patch Apply』（ドキュメントID：1265700.1）を参照してください。*

クラウド・ネットワークの前提条件

オンプレミスからOracle Cloud Infrastructure（OCI）へのデータ転送では、Oracle FastConnectによって提供されるパブリック・ネットワーク、VPN、高帯域オプションが使用されます。

Data Guard構成では、プライマリとスタンバイが双方向で通信できる必要があります。そのため、システム間のポートにアクセスするために、ネットワーク構成を追加する必要が生じます。

Exadata Cloud at Customerの場合は、オンプレミス・ネットワークにデプロイされるため、ネットワーク接続構成が不要です。ExaCCを使用する場合は、「オンプレミス・ネットワークの構成」セクションに移動してください。

セキュアな接続

Exadata Cloud Serviceの場合（Exadata Cloud at Customerの場合は不要）

クラウド・ネットワークをオンプレミス・ネットワークにプライベート接続する場合のために、FastConnectとIPSec VPNの2つのオプションが用意されています。いずれの方法でも、仮想クラウド・ネットワーク（VCN）に接続するための動的ルーティング・ゲートウェイ（DRG）が必要です。DRGの作成について詳しくは、このリンクのドキュメントを参照してください。

Oracle FastConnect

OCI FastConnectによって、データセンターとOCIの間に、専用のプライベート接続を作成できます。FastConnectには高帯域幅のオプションが用意されており、インターネットベースの接続に比べて、より信頼性の高い一貫したネットワーク接続を体験できます。FastConnectについて詳しくは、こちらを参照してください。

IPSec VPN

IPSecはInternet Protocol SecurityまたはIP Securityの略です。IPSecは、パケットが送信元から宛先に送られる前に、IPトラフィック全体を暗号化する、プロトコル・スイートです。OCIでのIPSecの概要については、こちらのドキュメントを参照してください。

パブリック・インターネット接続

OCIとオンプレミスとの間の接続には、パブリック・インターネットを使用することもできます。この方法は、デフォルトの状態では安全ではなく、送信のセキュリティを確保するために追加の作業が必要になります。このホワイト・ペーパーのこの後の部分は、パブリック・インターネット接続の利用を前提としています。

デフォルトでは、ポート1521のクラウド・セキュリティは無効になっています。また、クラウド内に存在する、仮想マシン（VM）またはベア・メタル（BM）のどちらか用の構成済みデフォルト・ポートには、パブリック・インターネットからオープンにアクセスできます。

1. スタンバイ・データベースの仮想クラウド・ネットワーク（VCN）にインターネット・ゲートウェイがない場合は、追加する必要があります。以下のリンク先には、インターネット・ゲートウェイの作成方法が掲載されています。
   https://docs.us-phoenix-1.oraclecloud.com/Content/Network/Tasks/managingIGs.htm
2. オンプレミス・データベースと接続するには、VCNセキュリティ・リストで受信ルールと送信ルールを構成する必要があります。以下のリンクから追加情報を参照できます。
   https://docs.cloud.oracle.com/en-us/iaas/Content/Network/Concepts/securitylists.htm

オンプレミス・ネットワークの構成

Data Guard構成では、プライマリとスタンバイのデータベース間で情報が双方向に伝送されます。これには、基本設定を行い、ネットワークをチューニングし、プライマリとスタンバイの両方のデータベースでポートを開く必要があります。

特に、プライマリ・データベースのREDO生成速度を処理できる帯域幅を確保することが重要です。My Oracle Support Note 2064368.1「Assessing and Tuning Network Performance for Data Guard and RMAN」の説明に従って、オンプレミス環境とクラウド環境の間のネットワーク・リンクを評価および調整してください。

オンプレミスの前提条件

スタンバイ・データベースをインスタンス化するには、以下の前提条件を満たす必要があります。

- 名前解決を設定する
  - ExaCCの場合は、クラスタがオンプレミス・ネットワーク上に存在するため、オンプレミスDNSで各クラスタが解決され、それ以上の構成は必要ありません。
  - Exadata Cloud Serviceの場合は、クラスタ間の名前解決を構成する必要があります。この設定は、/etc/hostsのような静的ファイルを使用するか、OCIインスタンスのパブリックIPアドレスが正しく解決されるようにオンプレミスDNSを設定することによって行うことができます。

    また、オンプレミスのファイアウォールでは、オンプレミス・システムとOCIとの間のSSHおよびOracle Netアクセスを許可するように適切に設定された、アクセス制御リストが必要になります。
- DRシチュエーションのData Guardでは、クラウド・インスタンスからオンプレミスのデータベースにアクセスする必要があります。そのためプライマリ・データベースのリスナー・ポートは、iptablesのような機能を使用して、クラウドのIPアドレスからのアクセスを制限して開いておく必要があります。ネットワーク・セキュリティ・ポリシーは企業ごとに異なるため、ネットワーク管理者は、以下の項に示すクラウド側のネットワーク構成に類似した操作を実行することが必要になります。
- Exadata Cloudからオンプレミス・マシンへのプロンプトレス SSH。これは、プロビジョニング・プロセス中に、オンプレミスからExadata CloudとExadata Cloudからオンプレミスの両方で構成します。
- Exadata Cloud Serviceからオンプレミス・マシンへのインバウンドSSH接続を許可する、オンプレミス・ファイアウォールの構成。
- 前のセクション「オンプレミス・ネットワークの構成」でネットワーク評価を完了しておくことを強くお奨めします。ASYNC REDO転送の場合には、適切なTCPソケット・バッファ・サイズを設定することが特に重要です。
- RDBMSソフトウェアは、インスタンス化のためにプライマリとスタンバイで同じものを使用する必要があります。現在のオンプレミスOracle DatabaseバージョンをExadata Cloudで使用できない場合は、プライマリ・データベースにパッチを適用するか、または使用可能なクラウド・バンドル・パッチにアップグレードする必要があります。クラウドで使用可能なバンドル・パッチの一覧は、以下のコマンドで表示できます。ソフトウェアのインストールについては、「デプロイメント・プロセス」セクションで説明しています。

  rootとして：
  ```
  # dbaascli cswlib list
  ```

- 個別パッチとマージ・パッチもプライマリとスタンバイのデータベース間で一致している必要があります。適用された個別パッチを以下のコマンドで検索し、パッチのドキュメントに従ってすべてのオンプレミス個別パッチをクラウド・データベース・ソフトウェアに適用します。

  oracleとして：
  ```
  $ORACLE_HOME/OPatch/opatch lspatches
  ```

- このドキュメントで概要を示す手順では、オンプレミスのプライマリ・データベースがまだ既存のData Guard Broker構成の一部になっていないことを前提としています。オンプレミス・データベースの既存の構成がある場合は、管理者がブローカについての情報を事前に得ており、既存のブローカ構成に新しいスタンバイ・データベースを追加する方法を知っていることを前提とします。次の問合せに対する戻り値が'NOCONFIG'以外の場合は、既存のブローカ構成を意味します。
  ```
  SQL> select decode(count(*),0,'NOCONFIG') from v$DG_BROKER_CONFIG;
  ```

- オンプレミスのマシンから次のコマンドを実行して、オンプレミス・リスナーのリスナー・ポートを検証します。このポートは、REDO転送の構成時に必要となり、デプロイメント・プロセス時にtns記述子に入力されます。
  ```
  $lsnrctl stat| grep 'Connecting to'
  Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=(1521)))
  ```

プライマリ・データベースでMAAベスト・プラクティスのパラメータ設定を実装する

ベスト・プラクティスのリストについては、付録Aを参照してください。このドキュメントで説明するプロセスでは、インスタンス化前にプライマリ・データベースが構成されていることを前提としています。インスタンス化前にREDOログを構成することは特に重要です。

オンプレミス・ホストとOracle Exadata Cloudホストとの間の接続の検証

すべてのネットワーク接続手順を正常に実施し終えたら、以下のコマンドを実行して、全ソースから全ターゲットへ、および全ターゲットから全ソースへの接続に問題がないことを確認します。telnetに成功したら、次の手順に進みます。

オンプレミス・ホスト上

```
[root@onpremise1 ~]# telnet <TARGET HOST IP ADDRESS> <PORT> Trying
xxx.xxx.xxx.xxx...
Connected to xxx.xxx.xxx.xxx.Escape
character is '^]'.
^C^]q
telnet> q
Connection closed.
```

クラウド・ホスト上

```
[root@oci2 ~]# telnet <TARGET HOST IP ADDRESS> <PORT> Trying
xxx.xxx.xxx.xxx...
Connected to xxx.xxx.xxx.xxx.
Escape character is '^]'.
^]q
telnet> q
Connection closed.
```

netcat (nc -zv <IP> <PORT>) can be used in place of telnet

デプロイメント・プロセス

以下のデプロイメント・プロセスでは、前提条件が満たされていることを前提としています。各環境間でネットワークが構成されると、Hybrid Data Guardスタンバイ・データベースをインスタンス化するプロセスは、オンプレミス構成でのプロセスと類似したプロセスになります。完了すると、クラウド・データベースは、クラウド・コンソールに表示され、クラウド・ツール、パッチ適用、バックアップ機能で使用できるように登録されます。スイッチオーバー、フェイルオーバー、復帰などのData Guard操作は、オンプレミス環境の場合のように、Data Guard Brokerを介して手動で実行されることになります。

特に記載がない限り、クラウド・システムがExadata Cloud ServiceなのかExadata Cloud at Customer Gen 2なのかに関係なく、コマンドは同じです。

このプロセスでは、プライマリ・データベースでASMとOracle Managed Files（OMF）が構成されていることを前提としています。非OMFおよび非ASM構成でのHybrid Data Guardの構成プロセスは、このドキュメントの適用範囲外です。

前提条件：クラウド・ツールを更新する

このプロセスでは、dbaastools rpmバージョン18.2.3.2.0_190618以上が必要です。最新のdbaastools rpmに更新することが常に推奨されています。

以下のようにして、最新のツールのrpmを適用します。

(as root) # dbaascli patch tools apply LATEST

手順1：クラウド・データベースを作成する

クラウド・コンソールのデータベース作成機能を使用して、スタンバイ・データベースとなるデータベースを作成します。コンソールを介してデータベースを作成する（続いてファイルを削除し、インスタンス化する）ことにより、確実にデータベースがコンソールに表示され、クラウド・ツールで使用できるよう正しく登録されるようにします。データベースの作成前に、プライマリ・データベースのバンドル・パッチとともにすべての個別パッチについて知らせておく必要があります。

必要なRDBMSバージョンとバンドル・パッチを選択およびダウンロードする

スペースに制限があるため、Exadata Cloudは、クラスタのACFSストレージ領域に、バージョンごとに1つのバンドル・パッチ（BP）イメージをローカルに保存します。どちらのBPであれ、コンソールを介してデータベースが作成されるときにダウンロードされるBPが、新規データベースが使用するBPとなります。使用するバージョンに必要なバンドル・パッチをダウンロードするには、以下の操作を行います。

最初に、使用可能なバージョンとバンドル・パッチのリストを表示します。

rootとして：

```
# dbaascli cswlib list

DBAAS CLI version 19.4.1.0.0
Executing command cswlib list
############ List of Available BP #############
-APR2017    (For DB  Versions      12201  12102 11204)
-JAN2018    (For DB  Versions      12201  12102 11204)
-APR2018    (For DB  Versions      12201  12102 11204)
-JUL2018    (For DB  Versions      18000  12201 12102 11204)
-OCT2018    (For DB  Versions      18000  12201 12102 11204)
-JAN2019    (For DB  Versions      18000  12201 12102 11204)
-APR2019    (For DB  Versions      18000  12201 12102 11204 19000)
-JUL2019    (For DB  Versions      18000  12201 12102 11204 19000)

#### List of Available NONCDB BP ####
-APR2018 (For DB Versions  12201 12102)
-JAN2019 (For DB Versions  12201 12102)
-APR2019 (For DB Versions  12201 12102)
-JUL2019 (For DB Versions  12201 12102)
```

次に、必要なバンドル・パッチをダウンロードします。

rootとして：

```
# dbaascli cswlib download --version 12201 --bp JUL2019

DBAAS CLI version 19.4.1.0.0
Executing command cswlib download --version 12201 -bp JUL2019 INFO:CSWLIB
update db image bits
INFO:Log file is: /var/opt/oracle/log/misc/cswlib/cswlib_<date>_<timestamp>.log
INFO:CSWLIB update_bits of 12201 succeded !
```

リスト表示されたバージョンごとのバンドル・パッチは、Exadata Cloudで使用可能なイメージのみです。オンプレミス・データベースのバンドル・パッチをクラウド内で使用できない場合は、インスタンス化の前に、プライマリ・データベースにパッチを適用するか、使用可能なバンドル・パッチの1つにアップグレードしてください。

クラウド・コンソールを使用してデータベースを作成する

クラウド・コンソールを使用し、前の手順でダウンロードしたバージョンを選択して正しいバンドル・パッチでデータベースを作成します。バンドル・パッチはリスト表示されず、バージョンのみが表示されます。コンソールを介してデータベースを作成することに慣れていない場合は、ExaCSとExaCCのドキュメントを参照してください。

Data Guardの要件に従い、スタンバイのデータベース名（db_name）はプライマリ・データベースと同じにする必要があります。ただし、db_unique_nameは別にする必要あります。ExaCCのデータベース作成画面には一意の名前用の別個のフィールドがあり、デフォルトでプライマリと異なる文字列になるように設定されています。ExaCSの場合は、一意の名前のための明示的な設定はありません。ツールにより、db_unique_nameは自動的にdb_nameと異なるように設定されます。

Exadata Cloudデータベースのデプロイ時には、バックアップ構成を含めるようお奨めします。オブジェクト・ストアへのバックアップは、公開時点のスタンバイ・データベースではサポートされていませんでした。ただし、スタンバイ・データベースがバックアップ可能になった場合や、クラウド・データベースがプライマリになる場合に備えて、バックアップを構成することをお奨めします。

手順2：APIで作成されたデータベースを手動で削除する

データベースがプロビジョニングされて準備ができたら、以下の手順を実行して、RMANを使用してオンプレミス・データベースからデータベース・ファイルを削除し、スタンバイをインスタンス化します。RMAN複製を開始する手順については、このドキュメントの以降の部分で説明します。

プロビジョニングされたデータベースを削除するには、ASMディスク・グループからデータベース・ファイルを手動で削除する方法を使用します。srvctl登録とともに、スタンバイ用に保持する必要があるoratabエントリも削除されるため、DBCAは使用しないでください。

Exadata Cloud Serviceホストのデータベースを手動で削除するには、以下の手順を実行します。

1. このプロセスでは、オブジェクト・ストアへのバックアップ用のRMAN設定を保存するcontrolfileを置き換えます。したがって、これらの設定はスタンバイ・データベースのインスタンス化後に保存および置換する必要があります。

2. スクリプトを作成して、すべてのデータベース・ファイルを削除します。

```
SQL> set heading off linesize 999 pagesize 0 feedback off trimspool on
SQL> spool /tmp/delete_ASM_files.sh
SQL> select 'asmcmd rm '||name from v$datafile
union all
select 'asmcmd rm '||name from v$tempfile
union all
select 'asmcmd rm '||member from v$logfile;
SQL> spool off
SQL> create pfile='/tmp/<standby DB_UNIQUE_NAME>.pfile' from spfile; #Backup of spfile
$chmod 777 /tmp/delete_ASM_files.sh
```

3. インスタンス化後に再適用するRMAN設定を保存します。

コマンドからの出力はログ・ファイルにのみプリントされます。画面には表示されません。

```
$ rman target / log='/tmp/rman_setting.log'

RMAN> show all;
RMAN> exit
```

重要：インスタンス化プロセスでは、controlfileがプライマリ・データベースのcontrolfileと置換されるため、Exadata Cloudツールによってデプロイされ たRMAN構成が置換されることになります。この手順では、インスタンス化後に置換される構成を保存します。これは、バックアップが構成されている場合には特に重要です。

4. データベースをシャットダウンします。
まず、後で参照できるよう、データベースのクラスタウェア構成を収集します。

```
$ srvctl config database -d <db_unique_name> > /tmp/<standby db_unique_name>.config
```

最後に、データベースを停止します。

```
$ srvctl stop database -d <db_unique_name> -o immediate
```

5. データベース・ファイルを削除します。

既存のデータファイル、ログ・ファイル、tempfileを削除します。パスワード・ファイルは置換され、spfileは再使用されます。

gridユーザーとして（opcユーザーからgridユーザーへsudo）

前に作成した/tmp/delete_ASM_files.sh を編集して、sqlplusからすべての不要な行を削除し、'asmcmd'で始まる行のみを残します。
[grid@<host> ~]$ vi /tmp/delete_ASM_files.sh

次に、スクリプトを保存して実行します。
[grid@<host> ~]$. /tmp/delete_ASM_files.sh
これで、初期データベースのすべてのファイルが削除されました。

手順3：パスワード・ファイルをExadata Cloudにコピーする

Exadata Cloudデータベースのパスワード・ファイルは、オンプレミスのプライマリ・データベースのパスワード・ファイルで置き換える必要があります。

パスワード・ファイルの場所を確認する

Oracle Clusterwareがオンプレミス・ホストで稼働している場合は、パスワード・ファイルの場所を確認します。

$ srvctl config database -db testdbname
Database unique name: testdbname
Database name:
Oracle home: /u02/app/oracle/product/12.2.0.0/dbhome_2
Oracle user: oracle
Spfile: +DATA/testdbname/spfiledbtestdbname.ora
Password file: +DATA/testdbname/PASSWORD/orapwd<sid> <======== password file location
Domain: domainname.xxxx.xxxx

オンプレミスのパスワード・ファイルをCloud Exadataにコピーする

パスワード・ファイルをすべてのExadata Cloudノードにコピーします。

オンプレミス・パスワード・ファイルの場所がASM以外の場合は、以下のようにしてそのファイルをコピーします。

$ scp -i <ssh key> $ORACLE_HOME/dbs/orapw<SID> opc@<Public-IP-OCI-HOST>:/tmp

パスワード・ファイルの場所がASMの場合は、ユーザーを"grid"またはASM所有者に切り替え、環境変数を指定してから、以下のようにしてパスワード・ファイルをコピーします。

オンプレミス
$ sudo su – grid
$ export ORACLE_SID=<ASM ORACLE_SID>
$ export ORACLE_HOME=<GRID_HOME>
$ asmcmd
ASMCMD> cd +<DISKGROUP_NAME>/<DB_UNIQUE_NAME>/PASSWORD
ASMCMD> cp orapw<SID> /tmp
copying +DATA/testdbname/PASSWORD/orapw<sid> -> /tmp/orapw<sid>
scp -i <ssh key> /tmp/orapw<SID> opc@<Public-IP-OCI-HOST>:/tmp

パスワード・ファイルをスタンバイ・データベースのExadata Cloudの適切な場所に格納する

Exadata Cloudホストのopcユーザーとして

$ chmod 777 /tmp/<password file name>
$ sudo su – grid

パスワード・ファイルをASMに格納します。

srvctl config database -db <stadndby DB_UNIQUE_NAME> を使用して、現在のパスワード・ファイルの場所を検索します。

gridユーザーとして：

$ asmcmd pwcopy --dbuniquename <standby DB_UNIQUE_NAME> /tmp/<password file name> <current standby password file> –f

以前に登録されたパスワード・ファイルの場所を再使用している場合、Error ASMCMD-9453: failed to register password file as a CRS resourceは無視してかまいません。

手順4：Exadata Cloudへのウォレット・ファイルのコピー

以下の問合せで、オンプレミス・データベースからTDEウォレットの場所を取得します。
SQL> select WRL_PARAMETER from v$encryption_wallet; WRL_PARAMETER

/u01/app/oracle/admin/<db_unique_name>/wallet/

オンプレミス・ディレクトリのewallet.p12ファイルとcwallet.ssoファイルを、Exadata Cloudノード1上の/tmpディレクトリにコピーします。プライマリでは、パスワード・ファイルの場合と同様に、ASMから/tmpにファイルをコピーすることが必要になる場合があります。

オンプレミス・ホスト上

scp -i ~/<ssh_key> <PATH>/ewallet.p12 opc@<Public-IP-OCI-HOST1>:/tmp
scp -i ~/<ssh_key> <PATH>/cwallet.sso opc@<Public-IP-OCI-HOST1>:/tmp

/var/opt/oracle/dbaas_acfs/<standby db_name>/tde_walletの古いウォレット・ファイルを削除します。ノード1でのみコマンドを実行します（Exadata Cloudノードは、ACFSを使用して/var/opt/oracle/dbaas_acfsのストレージを共有）。

クラウド・ホスト・ノード1上

opcユーザーとして

$ chmod 777 /tmp/ewallet.p12
$ chmod 777 /tmp/cwallet.sso
$ sudo su – oracle
$ cp /tmp/ewallet.p12 /var/opt/oracle/dbaas_acfs/<standby db_name>/tde_wallet/
$ cp /tmp/cwallet.sso /var/opt/oracle/dbaas_acfs/<standby db_name>/tde_wallet/
$ chmod 600 /var/opt/oracle/dbaas_acfs/<standby db_name>/tde_wallet/*wallet*

プライマリのウォレット・ファイルがASMに保存される場合、ASMキーストアは、ローカル・キーストア・ファイルをクラウドにコピーする前に、ローカル・ファイル・システムのキーストアにマージする必要があります。詳しくは、My Oracle Support のドキュメントID 2193264.1を参照してください。

手順5：（11gのみ）静的リスナーを構成する

この手順は、11.2データベースの場合にのみ必要とされます。» 静的リスナーは、スタンバイ・データベースの最初のインスタンス化のほかに、Data Guard Brokerの通信で必要とされます。静的リスナーにより、インスタンスを開始できるようにするためにデータベースが停止している間でも、インスタンスにリモート接続することができます。詳しくは、My Oracle Support Doc ID 1387859.1を参照してください。

gridユーザーまたはソフトウェア・オーナーとして、変数の置換後にExadata Cloudノード1のlistener.oraに次のエントリを追加します。listener.oraは<grid home>/network/adminにあります。

スタンバイの最初のノードのlistener.ora

```
SID_LIST_LISTENER =
   (SID_LIST =
      (SID_DESC =
         (GLOBAL_DBNAME = <DB_UNIQUE_NAME>_DGMGRL)
         (ORACLE_HOME = <Local Oracle Home>)
         (SID_NAME = <ORACLE SID of the local instance>)
      ) (SID_DESC =
         (GLOBAL_DBNAME = <DB_UNIQUE_NAME of the Cloud database>)
         (ORACLE_HOME = <Local Oracle Home>)
         (SID_NAME = <ORACLE SID of the local instance>)
      )
 )
```

構成内の残りのノードすべてのlistener.ora

```
SID_LIST_LISTENER =
   (SID_LIST =
      (SID_DESC =
         (GLOBAL_DBNAME = <DB_UNIQUE_NAME>_DGMGRL)
         (ORACLE_HOME = <Local Oracle Home>)
         (SID_NAME = <ORACLE SID of the local instance>)
      )
)
```

最後にリスナーを再ロードします（gridユーザーまたはクラスタウェア・ソフトウェア所有者として）。

```
$ORACLE_HOME/bin/lsnrctl reload
```

手順6：REDO転送用のオラクルのネットワーク暗号化とTNSエントリ

プレーンテキストまたは暗号化されていない表領域のREDOがWAN上で可視状態にならないよう保護するには、オンプレミスのすべてとクラウドにおいてsqlnet.oraファイルに次のエントリを入力します。

クラウドのデプロイメントでは、TNS_ADMIN変数を利用して、共有データベース・ホームでtnsnames.oraとsqlnet.oraを分離します。したがって、所定のデータベースのクラウドsqlnet.ora、および延長線上のtnsnames.oraは、$ORACLE_HOME/network/admin/<db_name>に格納されます。これらの値は、クラウド構成においてデプロイメント・ツールによってすでに設定されています。

オンプレミス・ホスト上のsqlnet.ora

```
SQLNET.ENCRYPTION_SERVER=REQUIRED
SQLNET.CRYPTO_CHECKSUM_SERVER=REQUIRED
SQLNET.ENCRYPTION_TYPES_SERVER=(AES256,AES192,AES128)
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER=(SHA1)
SQLNET.ENCRYPTION_CLIENT=REQUIRED
SQLNET.CRYPTO_CHECKSUM_CLIENT=REQUIRED
SQLNET.ENCRYPTION_TYPES_CLIENT=(AES256,AES192,AES128)
```

SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT=(SHA1)

REDO転送が正常に行われるようにするには、プライマリとスタンバイの両方のtnsnames.oraファイルに各データベースのエントリが含まれている必要があります。以下の例の太字の値を、構成に沿った値で置き換えて使用してください。

データベースのTNS記述子は、他のシステムからスキャン・リスナーが解決可能かどうかに応じて異なります。

以下の説明では、スキャン名が解決可能で、TNS記述子で使用可能なことを前提としています。スキャン名を解決できない場合は、ADDRESS_LISTを使用して、付録BでTNS記述子の例を参照してください。

オンプレミス・ホスト上のtnsnames.ora

必要な置換操作を行った後に、次の記述子をオンプレミスのtnsnames.oraファイルに追加します。

```
<standby db_unique_name> =
  (DESCRIPTION =
    (SDU=65536) (RECV_BUF_SIZE=134217728)
    (SEND_BUF_SIZE=134217728)
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = <standby scan name>)(PORT = {1521|<port#>}))
    )
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = <service name of the standby database>)
    ))
```

クラウド・ホスト上のtnsnames.ora

必要な変更を加えた後に、プライマリ・データベースの記述子をクラウドのtnsnames.oraファイルに追加します。

クラウド・スタンバイの記述子名を、db_nameではなくdb_unique_nameを使用するように変更してください。

```
<primary db_unique_name> =
  (DESCRIPTION =
    (SDU=65536) (RECV_BUF_SIZE=134217728)
    (SEND_BUF_SIZE=134217728)
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = <primary scan name>)(PORT = {1521|<port#>}))
    )
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = <primary database service name>])
      (UR=A)
    ))
```

手順7：スタンバイ・データベースをインスタンス化する

スタンバイ・データベースは、アクティブなプライマリ・データベースから、またはプライマリ・データベースのバックアップから作成できます。この項では、Oracle 12.1以降の機能であるRMAN 'RESTORE…FROM SERVICE'を使用して、アクティブなプライマリ・データベースから複製を実行する方法を説明します。

RDBMS 11.2では、RMAN RESTORE FROM SERVICEをサポートしません。バックアップ・ベースの複製またはRMAN DUPLICATEを使用する必要があります。詳しくは、ドキュメントを参照してください。

スタンバイのインスタンス化について詳しくは、MOS 2275154.1「Creating a Physical Standby Database in an 11.2, 12.1, 12.2 or later environment.」を参照してください。

プライマリ・データベースをコピーする前に、特にスタンバイREDOログの作成時など、プライマリ・データベースでベスト・プラクティスが確実に実施されるようにしてください。SRLは、コピー前に作成されない場合、プライマリとスタンバイのデータベースに別々に追加することが必要になります。

スタンバイ・インスタンスを起動します（RACで1インスタンス）。

$ srvctl stop database -d <standby DB_UNIQUE_NAME> -o immediate

$ rman target /
RMAN> startup nomount

RMAN> restore standby controlfile from service '<primary tns descriptor>';

RMAN> alter database mount;

RMAN> CONFIGURE DEVICE TYPE DISK PARALLELISM 4; ← 並列度（チャネルの数）を設定します。帯域幅に応じて変更できます。

RMAN> restore database from service '<primary tns descriptor>' section size 64G;

コピーを完了するのに必要な時間は、データベースのサイズと使用可能な帯域幅に応じて異なります。たとえば、小規模なデータベースの場合には分単位の時間が予想され、大規模なデータベースの場合には何時間もの時間を要することがあります。

RMAN> shutdown immediate

スタンバイ・データベースを再起動します。

$ srvctl start database -d <standby DB_UNIQUE_NAME> -o mount

すべてのオンラインおよびスタンバイREDOログを消去します。ログを消去する前に検証します。

- DB_CREATE_ONLINE_LOG_DEST_1= <DATA disk group>.必要に応じて修正します。
- DB_CREATE_ONLINE_LOG_DEST_*n* is not set other than n=1.

すべてのlogfileを消去します。
$ sqlplus "/ as sysdba"
SQL> set pagesize 0 feedback off linesize 120 trimspool on
SQL> spool /tmp/clearlogs.sql
SQL> select distinct 'alter database clear logfile group '||group#||';' from v$logfile;
SQL> spool off
SQL> @/tmp/clearlogs.sql

SQL> select member from v$logfile;

すべてのREDOログは、スタンバイDB_UNIQUE_NAMEディレクトリのDATAディスク・グループに格納されることになっています。

手順8：Data Guard Brokerを構成する

プライマリ・データベースとスタンバイ・データベースで、dg_broker_config_fileパラメータを有効にします。

ASMの場合は、ブローカ構成ファイルを別々のディスク・グループに配置します。RACの場合は、ブローカ構成ファイルを共有ストレージに格納する必要があります。

オンプレミス上とクラウド・ホスト上

SQL> alter system set dg_broker_config_file1='+DATAC1/<db_unique_name/dr1.dat';

SQL> alter system set dg_broker_config_file2='+RECOC1/<db_unique_name/dr2.dat';

プライマリ・データベースとスタンバイ・データベースでData Guard Brokerのプロセスを開始します。

オンプレミス上とクラウド・ホスト上

SQL> alter system set dg_broker_start=true;

SQL> show parameter dg_broker_start

NAME	TYPE	VALUE
dg_broker_start	boolean	TRUE

SQL> select pname from gv$process where pname like 'DMON%';

PNAME
DMON
DMON

プライマリ・サイトでDGMGRLを使用してデータベースを登録します。

オンプレミス・ホスト上

$ dgmgrl sys/<sys password>@<net service name for primary database>

DGMGRL> CREATE CONFIGURATION <configuration_name> AS PRIMARY DATABASE IS <primary db_unique_name>

CONNECT IDENTIFIER IS <Net Service name for primary database>;

DGMGRL> ADD DATABASE <standby db_unique_name> AS CONNECT IDENTIFIER IS <Net Service name for standby database> MAINTAINED AS PHYSICAL;

DGMGRL> enable configuration;

スタンバイでフラッシュバック・データベースを有効化します。

スタンバイ・データベース上

DGMGRL> edit database <standby> set state=apply-off;

SQL> alter database flashback on;

DGMGRL> edit database <standby> set state=apply-on;

手順9：（11gのみ）静的リスナーを削除する

Exadata Cloudノード1のgridユーザーとして、インスタンス化用に作成した静的リスナーのSID_DESCを削除し、リスナーを再ロードします。

```
(SID_DESC =
    (GLOBAL_DBNAME = <DB_UNIQUE_NAME of the Cloud database>)
    (ORACLE_HOME = <Local Oracle Home>)
    (SID_NAME = <ORACLE SID of the local instance>)
)
```

<DB_UNIQUE_NAME>_DGMGRLのSID_DESCを削除しないでください

手順10：RMANパラメータを設定する

元のRMAN構成を、前に作成した/tmp/rman_setting.log の内容で置き換えます。これにより、もっとも重要な点として、バックアップ構成が、スナップショットcontrolfileの場所に加えて、暗号化、圧縮、保存で置換されます。これらの値は、スタンバイのcontrolfileがプライマリ・データベースからコピーされたときに失われています。

ヘルス・チェックと監視

スタンバイをインスタンス化したら、ヘルス・チェックを実行して、Data Guardデータベース（プライマリとスタンバイ）がOracle MAAのベスト・プラクティスに準拠していることを確認します。ヘルス・チェックは、毎月実施するとともに、データベースのメンテナンスの前後にも実施するようお奨めします。Data Guard構成のヘルス・チェックは、いくつかの方法で実施できます。

Oracle MAAスコアカード

オラクルでは、複数の自動化されたヘルス・チェック・ツールを提供しており、ハードウェア・プラットフォームのタイプごとにMy Oracle Supportからダウンロードすることができます。

- exachk：Oracle Exadata Database Machineに適用可能（Exadata Cloud Serviceに最適）
- すべてのOracleスタックに対するORAchkアプリケーション

自動化された各チェック機能には、Data Guard構成の多数の主要なベスト・プラクティスおよびその他多くのチェック項目について報告するOracle MAAスコアカードが組み込まれています。

これらの自動化ツールは、Data Guard構成だけでなく、システム全体の包括的なヘルス・チェックでも使用することを強く推奨します。ヘルス・チェックの結果は、最新の情報に基づいて定期的に更新されます。必ず、ご使用のプラットフォームに該当する最新バージョンのヘルス・チェック・ツールをダウンロードしてください。

監視

Data Guard構成の標準の監視は、Hybrid Data Guard構成では行われないため、手動で実行する必要があります。監視に関するMAAベスト・プラクティスの推奨事項については、MOS Note『Monitoring a Data Guard Configuration（ドキュメントID 2064281.1）』を参照してください。

DRの準備状況の検証

ベスト・プラクティスは、Active Data Guardを使用して読取り専用ワークロードをスタンバイ・データベースにオフロードし、スタンバイで本番稼働の準備ができていることをアプリケーションレベルで検証し続けることです。これにより、Data Guardの適用プロセスで実行される継続的なOracleブロックレベルの検証に加えて、一定レベルの保証が得られます。ベスト・プラクティスとしては、（Data Guardスナップショット・スタンバイを使用して）スタンバイを定期的に読取り/書込みモードにし、読取り/書込みの本番ワークロードにいつでも対応できるようになっているか検証することも必要です。DRシステムは本番システムと同様にサイジングされるため、パッチやアップグレードの本番前機能テストやパフォーマンス・テストの最終段階でスナップショット・スタンバイを使用することもできます。

スナップショット・スタンバイはプライマリ・データベースからREDOを受信し続け、後で使用するためにREDOをアーカイブし、それにより常時データを保護します。ただし、テストの進行中にフェイルオーバーが必要になった場合のリカバリ時間（RTO）は、スナップショット・スタンバイを元のスタンバイ・データベースに変換するために必要とされる時間だけ長くなります。スタンバイがスナップショット・モードになっている場合は、（プライマリ本番データベースから受信し後で使用するためにアーカイブされたREDOログと、スナップショット・スタンバイによって生成される最新のREDOログおよびフラッシュバック・ログを保持するため、）ファスト・リカバリ領域として追加のストレージが必要になります。スタンバイをスナップショット・スタンバイに変換し、戻すための手順は、以下の項に記載されています。Data Guardスナップショット・スタンバイの詳細については、Oracleドキュメントを参照してください。
必要に応じて、完全なエンド・ツー・エンドのDRテストとして、クラウドへの実際のスイッチオーバーまたはフェイルオーバー操作を実行することもできます。詳しくは、「クラウドへのフェイルオーバー/スイッチオーバー」を参照してください。

スタンバイ・データベースからスナップショット・スタンバイへの変換

スナップショット・スタンバイは、フィジカル・スタンバイ・データベースから作成される、すべてを更新可能なスタンバイ・データベースです。スナップショット・スタンバイ・データベースでは、REDOデータを受信しますが、スナップショット・スタンバイ・データベースが変換されてフィジカル・スタンバイ・データベースに戻されるまで適用されません。

スナップショット・スタンバイ・データベースを使用することには、次のような利点があります。

1. 常時データが保護された状態を維持しつつ、開発およびテスト目的で本番データベースの正確なレプリカとして機能します。Oracle Real Application Testingオプションを使用して、プライマリ・データベースのワークロードを捕捉し、スナップショット・スタンバイでテスト目的で再生することができます。
2. フィジカル・スタンバイへの変換と再同期化により、容易にリフレッシュして現在の本番データを含めることができます。
フィジカル・スタンバイ・データベースをスナップショット・スタンバイに変換するには、以下の手順に従います。

スタンバイをスナップショット・スタンバイに変換し、検証します。

Data Guard Brokerを介して、次のコマンドを発行します。

```
DGMGRL> convert database 'stby' to snapshot standby;
DGMGRL> SHOW CONFIGURATION;
Configuration - DRSolution
Protection Mode:MaxPerformance Databases:
prmy - Primary database
stby - Snapshot standby database
Fast-Start Failover:DISABLED
Configuration Status:SUCCESS
```

スナップショット・スタンバイ・データベースをスイッチオーバーまたはフェイルオーバーのターゲットにすることはできません。スナップショット・スタンバイ・データベースは、そのデータベースへのロール移行を実行する前に、まず変換してフィジカル・スタンバイ・データベースに戻す必要があります。

スナップショット・スタンバイから元のフィジカル・スタンバイ・データベースに変換します。

Data Guard Brokerを介して、次のコマンドを発行します

```
DGMGRL> CONVERT DATABASE 'stby' to PHYSICAL STANDBY;
```

クラウドへのフェイルオーバー/スイッチオーバー

Data Guardのスイッチオーバー（計画イベント）やフェイルオーバー（計画外イベント）は、いつでも手動で実行することができます。また、ファスト・スタート・フェイルオーバーを設定することによるData Guardのフェイルオーバーの自動化も選択できます。スイッチオーバーとフェイルオーバーにより、Data Guard構成におけるデータベースのロールが逆転し、クラウドのスタンバイがプライマリになり、元のオンプレミスのプライマリがスタンバイ・データベースになります。Data Guardのロール移行について詳しくは、Oracle MAAのベスト・プラクティスを参照してください。

スイッチオーバーは常に、データを失わないことが保証された計画イベントとなります。スイッチオーバーを実行するには、Data Guard Brokerで以下のコマンドを実行します。

DGMGRL> validate database stby;

Database Role:Physical standby database Primary Database: pri

Ready for Switchover:Yes
Ready for Failover:Yes (Primary Running)
DGMGRL> switchover to <target standby>;

フェイルオーバーは、プライマリ・データベースで障害が発生することを想定した計画外イベントです。使用可能なプライマリのすべてのREDOが適用された後、スタンバイ・データベースが即時にプライマリ・データベースに変換されます。フェイルオーバーの後、古いプライマリ・データベースはフィジカル・スタンバイとして復旧する必要があります。これは、フラッシュバック・データベースとData Guard Brokerを有効にすることによって簡単に行うことができます。フェイルオーバーと復旧を実行するには、Data Guard Brokerで以下のコマンドを実行します。

DGMGRL> failover to stby;

Performing failover NOW, please wait...

Failover succeeded, new primary is "stby"

復旧前に、古いプライマリの1つのインスタンスで、startup mountを実行します。

SQL> shutdown abort

SQL> startup mount

DGMGRL> reinstate database pri

Reinstating database "pri", please wait...

Data Guard Brokerを使用したロール移行について詳しくは、Oracle Database 11gまたは12c用のBrokerのドキュメントを参照してください。

オンプレミスにスイッチバックする

本番データベースをオンプレミス・データベースに移行する準備ができたら、フェイルオーバー/スイッチオーバーのプロセスで述べられているのと同じロール移行手順を再び適用します。

ソフトウェア更新 - パッチ適用とアップグレード

説明に従ってプロセスを完了させたら、クラウド・ツールを使用してスタンバイ・データにパッチを適用し、アップグレードすることができます。Hybrid Data Guard構成のパッチ適用とアップグレードのプロセスについては、My Oracle Support Note 2626861.1：「Patching and upgrading a Hybrid Data Guard Standby Database」で説明されています。

結論

Exadata Cloudを使用するHybrid Data Guardは、ディザスタ・リカバリに備えるための経済的な方法です。Maximum Availability Architectureのベスト・プラクティスを適用することにより、データの保護と可用性のためのベスト・ソリューションを提供します。

付録A：MAAのベスト・プラクティスとパラメータ設定

MAAベスト・プラクティスに従ってデータの最大限の可用性と保護を達成するには、以下の設定を推奨します。以下のパラメータと特性は、プライマリ・データベースとスタンバイ・データベースの両方で設定する必要があります。

- ARCHIVELOG mode enabled
- Flashback database on
- FORCE LOGGING enabled
- Use SPFILE
- Use Data Guard Broker
- オンラインREDOログの特性
    - Exadata以外のデータベースではサイズが1 G以上、Exadataでは4 G以上
    - スレッド当たり最小3グループ
    - 高冗長性ストレージを使用する場合は単一メンバー・グループ
    - グループが単一メンバーの場合にはDATAディスク・グループ上に存在
- スタンバイREDOログの特性
    - オンラインREDOログと同一サイズ
    - Oracle RACの場合、SRLグループをスレッドに割り当て
    - スレッド当たり、オンラインREDOログ・グループと同じ数のグループ
    - 単一メンバー・グループのみ
    - DATAディスク・グループ上に存在
- LOG_BUFFER = 128M for 11.2; 256M for 12.1+
- プライマリの場合はDB_BLOCK_CHECKING=NONE、スタンバイの場合はMEDIUMまたはFULL

  注：MEDIUMの値はプライマリ用に提案されていますが、この設定はパフォーマンスに影響する可能性があり、アプリケーションを適切にテストした後にのみ有効にする必要があります。
- DB_BLOCK_CHECKSUM=TYPICAL
- STANDBY_FILE_MANAGEMENT=AUTO
- DB_LOST_WRITE_PROTECT=TYPICAL
- DB_FLASHBACK_RETENTION_TARGET=最小値120
- FAST_START_MTTR_TARGET=300
- USE_LARGE_PAGES=ONLY（オンプレミス・システムでホームページが構成され、適格にサイジングされている場合）
- CLUSTER_INTERCONNECTS（gv$cluster_interconnectsに従って設定）　***#Exadataのみ***
- DB_CREATE_ONLINE_LOG_DEST_1= DATAディスク・グループ
- DB_CREATE_ONLINE_LOG_DEST_n（n=1以外はDATAディスク・グループが高冗長性ではない場合にのみ設定。この場合、設定はRECO）（クラウドでは高冗長性ディスク・グループを使用）
- DB_CREATE_FILE_DESTではDATAディスク・グループを使用
- DB_RECOVERY_FILE_DESTではRECOディスク・グループを使用
- Recyclebinはオン

付録B：ADDRESS_LISTを使用するtnsnames.oraサンプル

ADDRESS_LISTを使用する以下のTNS記述子の例は、Cloudクラスタとオンプレミス・クラスタの間でスキャン名を解決できない状況で使用可能です。

スタンバイ・データベース

オンプレミスのtnsnames.oraのオンプレミス・データベース記述子に加える必要のある変更はありません

```
<standby db_unique_name> =
   (DESCRIPTION=
   (SDU=65536) (RECV_BUF_SIZE=134217728) (SEND_BUF_SIZE=134217728)
      (ADDRESS_LIST= (FAILOVER=on)
        (CONNECT_TIMEOUT=3)(RETRY_COUNT=3)
        (ADDRESS = (PROTOCOL = TCP)(HOST = <standby node1 VIP address>)(PORT = {1521|<port#>}))
   )
        (ADDRESS = (PROTOCOL = TCP)(HOST = <standby node2 VIP address>)(PORT = {1521|<port#>}))
   )
      )
      (CONNECT_DATA=
         (SERVER=DEDICATED)
         (SERVICE_NAME= <standby database service name>)
      )
   )
```

プライマリ・データベース

*Cloudデータベースの接続記述子の名前を変更して、db_nameの代わりにdb_unique_nameを使用します。*それ以外の変更はありません。

```
<primary db_unique_name> =
   (DESCRIPTION=
   (SDU=65536) (RECV_BUF_SIZE=134217728) (SEND_BUF_SIZE=134217728)
     (ADDRESS_LIST= (FAILOVER=on)
       (CONNECT_TIMEOUT=3)(RETRY_COUNT=3)
       (ADDRESS = (PROTOCOL = TCP)(HOST = <primary node1 VIP address>)(PORT =
{1521|<port#>}))
   )
       (ADDRESS = (PROTOCOL = TCP)(HOST = <primary node2 VIP address>)(PORT =
{1521|<port#>}))
   ) (CONNECT_DATA =
       (SERVER = DEDICATED)
       (SERVICE_NAME = <primary database service name>)
   ))
```

オラクルの情報を発信しています

+1.800.ORACLE1までご連絡いただくか、oracle.comをご覧ください。

北米以外の地域では、oracle.com/contactで最寄りの営業所をご確認いただけます。

blogs.oracle.com facebook.com/oracle twitter.com/oracle

Oracle Exadata Cloudを使用したディザスタ・リカバリ 2020年3月

著者：Andrew Steinorth

共著者：Glen Hawkins