ORACLE
Cloud Infrastructure

# Hybrid Data GuardからOracle ExaCCへ

## オンプレミスでの本番データベースとExadata Cloud@Customerでのディザスタ・リカバリ

Oracle ホワイト・ペーパー | 2019 年 9 月

ORACLE

目次.....1

## 概要

Oracle Maximum Availability Architecture（Oracle MAA）は、オラクルの実証済みの高可用性テクノロジー、エンド・ツー・エンドの検証、専門家の推奨事項、およびカスタマー・エクスペリエンスを基盤とするオラクルのベスト・プラクティス構想です。このホワイト・ペーパーでは、Hybrid Data Guard 構成を Exadata オンプレミスと Oracle Exadata Cloud@Customer（Oracle ExaCC）で構成し、維持するためのベスト・プラクティスを紹介します。このホワイト・ペーパーでは、Oracle Exadata Cloud at Customer 構成の基本を理解していることを前提としています。

Oracle Data Guard および Oracle Active Data Guard により、バックアップからのリストアではリカバリ時間目標（RTO）を達成することのできないデータベースのディザスタ・リカバリ（DR）を実現できます。これらのソリューションを利用してデータベース（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 1 つ以上デプロイし、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 Cloud at Customer 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、Oracle Exadata のハードウェアとインフラストラクチャの管理に伴うコストや煩雑さを軽減します。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@Customer 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・クラウド実装と呼ばれます。

## Data GuardおよびActive Data Guardによるクラウドへのディザスタ・リカバリ

Active Data Guard は Cloud at Customer PaaS ライセンスに付属しています。Data Guard の機能を拡張したもので、データ保護および可用性を目的とした高度な機能のほか、読取り専用ワークロードのオフロードや本番データベースからの高速増分バックアップを実行する機能も備えています。オンプレミス・ライセンスの観点から、Active Data Guard は、Extreme Performance Edition 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。

Oracle Maximum Availability Architecture では、次のことを推奨します。

1. ロールの移行後、同じパフォーマンス SLA を確実に満たせるように、ExaCC ターゲット・システムがオンプレミスの Exadata システムと対称になっているか類似の構成になっている。
2. ピーク REDO レートに対処するためのネットワーク帯域幅を十分に確保する。
3. オンプレミスと ExaCC 間のネットワークの信頼性と効率性を確保する。
4. 自動ブロック修復、データ保護、およびオフロードの追加のメリットを得るために、Active Data Guard を使用する。

# クラウドにおけるハイブリッド・スタンバイのメリット

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します。
2. スタンバイをホストする VM のコンピュート・リソースの拡張と縮小など、クラウドでは基本的なシステム・ライフサイクルの操作を実行できます。
3. Oracle Data Guard は、ディザスタ・リカバリ、データ保護、アクティビティをオフロードして利用率と投資回収率の向上に対応する機能を提供します。
4. MAA プラクティスに従って構成を行い、Data Guard ファスト・スタート・フェイルオーバーを使用すると、秒単位のリカバリ時間目標（RTO）を達成でき、ASYN 転送構成では 1 秒未満、SYNC 転送構成または FAR SYNC 転送構成ではゼロのリカバリ・ポイント目標（RPO または潜在的データ損失）を達成できます。

## 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

クラウド上のスタンバイ・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

徹底的な検証のため、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できます。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 Oracle RAC で、ソフトウェアの更新が RAC ローリングである場合、スイッチオーバーは必要ありませんが、それでも検証と保護をさらに強化するため、ソフトウェアを Standby-First で更新することを推奨します。Standby-First 対象のパッチを適用する場合、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。このことについては、パッチの README に記載されています。パッチに関して予期せぬ問題が発生した場合には、パッチが適用されていないバージョンに直ちにフォールバックできるよう、一定期間は異なるパッチ・バージョンのプライマリとスタンバイを稼働させることもできます。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

Oracle Data Guard のもう 1 つの便利なユースケースは、Standby-First ソリューションが適していない場合に、一時ロジカル・スタンバイまたは DBMS_Rolling ソリューションでデータベース・ローリング・アップグレードが利用できることです。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。詳細については、『Oracle Data Guard フィジカル・スタンバイ・データベースの使用によるデータベース・ローリング・アップグレードの簡略化』または Oracle 12c の DBMS_Rolling を参照してください。12.2 以降の Data Guard 環境の場合、スタンバイ・データベースを使っ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・プロセスがあります。Oracle Data Guard のドキュメントの項「Using DBMS_ROLLING to Perform a Rolling Upgrade 」を参照してください。

*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。インスタンス化、管理、またはメンテナンス用のクラウド・コンソールのサポートはありません。*

## サービス・レベル要件

ハイブリッド・クラウド・デプロイメントは、定義上、ユーザー管理の環境です。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計算、メモリ、I/O などの能力がオンプレミスの本番システムよりも劣っていると、データベースの応答時間がフェイルオーバー後に変わる可能性があります。管理者がコストを削減するために意図的に少ないリソースでスタンバイ・システムを構成し、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、この状態が発生します。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。クラウドではコンピュート・バーストが使用可能なため、安定状態では少ない容量でデプロイされ、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高速にスケールアップされるようにする、という中間的な構成が可能です。

## 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で、すべてのデータを暗号化するよう推奨しています。それには、プライマリ・データベース用の TDE ライセンスが必要です。オンプレミスの高度なセキュリティ・オプション（TDE を含む）がある場合、TDE ライセンスはすでに Oracle ExaCC に付属しています。インスタンス化プロセス中にデータを変換することはできますが、もっとも安全な Data Guard 環境を実現するために、移行前に TDE に変換することを強く推奨します。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ていない他のデータベース・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 の暗号化も必要です。

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、元のプレーンテキスト・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 暗号化を使用する場合、REDO 転送で圧縮を使用することは推奨されません。ただし、Oracle Advanced Compression や Hybrid Columnar Compression といったオラクルのデータベース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。

## データベース、OS環境、およびネットワークの前提条件

オンプレミスで TDE が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。

*** プライマリ・データベースとスタンバイ・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。*

### ネットワークの前提条件

ExaCC は TOR スイッチによってオンプレミス・ネットワークに接続されており、オンプレミス・ネットワークから ExaCC ネットワークへのアドレスを解決するように DNS が構成されているため、ExaCC Hybrid Data Guard に必要な追加のネットワークはありません。

### 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。

ExaCC からオンプレミスへのプロンプトレス SSH を構成する必要があります。

» オンプレミス・データベースの Oracle Home は、スタンバイ・データベースの Oracle パッチセットと同じである必要があります。ExaCC 環境がオンプレミス・データベースとは異なるバンドル・パッチ・レベルにある場合、ソース環境を ExaCC 環境のデータベース・ホームと同じデータベース・バンドル・パッチ・レベルにパッチ適用することを推奨します（ソース環境とターゲット環境の両方にインストールされているパッチを確認するには、コマンド"$ORACLE_HOME/OPatch/opatch lspatches"を実行します）。

» オンプレミス・データベースはコンテナ・データベース（CDB）である必要があります。非 CDB は現在、クラウド環境ではサポートされていません。

» 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ブローカ構成が存在する場合は、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(1),0,'NOCONFIG') from v$DG_BROKER_CONFIG;
```

» LISTENER という名前のデフォルトのリスナーを使用します。このドキュメントで概説する手順では、デフォルトのリスナー名（LISTENER）を使用することを前提としています。検証するには、オンプレミスのマシンから次のコマンドを実行します。予期される結果が表示されます。

```
$lsnrctl show current_listener | grep 'Current Listener' Current Listener is
LISTENER
```

» オンプレミスのマシンから次のコマンドを実行してリスナー・ポートを検証します。予期される結果が表示されます。

```
$ 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。インスタンス化の前に、プライマリ・データベースでこのプロセスを完了しておくことを推奨します。インスタンス化プロセス中に複製されるオンラインとスタンバイ双方の REDO ログを構成する場合は特にそのようにしてください。

# 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。ExaCC 上でスタンバイ・データベースをインスタンス化するプロセスは、オンプレミス構成の後続プロセスに非常によく似ています。ExaCC 環境固有の構成については、このプロセスの残りの部分で説明します。

## 手順1：オンプレミス構成上

オンプレミス・データベースが MAA ベスト・プラクティスに従って構成されていることを前提とします。パラメータ設定については、付録 A を参照してください。構成の他の項目のリストを以下に示します。

#### TCPソケットの最大サイズを設定する

次のコマンドを root として実行して、オンプレミス・システムおよびクラウド・インスタンス用のオペレーティング・システムの TCP ソケットの最大サイズを確認します（ExaCC の TCP ソケット・サイズは 128 MB）。

オンプレミス・ホスト上

```
# /sbin/sysctl -a | egrep net.core.[w,r]mem_max
net.core.wmem_max = 4194304
net.core.rmem_max = 4194304
```

ExaCC 上

```
# /sbin/sysctl -a | egrep net.core.[w,r]mem_max
net.core.wmem_max = 4194304
net.core.rmem_max = 4194304
```

必要に応じて、すべてのソケットの最大サイズを 128 MB（134217728）に調整します。オンプレミス・システムの場合のソケットの調整方法について詳しくは、オペレーティング・システム・ガイドを参照してください。クラウド・インスタンスの場合、net.core.wmem_max および net.core.rmem_max の/etc/sysctl.conf ファイル設定を編集します。オンプレミスと ExaCC 間の値が一致しない場合、ネットワーク・プロトコルは、2 つの値のうち、低い方をネゴシエートします。したがって、サイト間の値は一致する必要はありませんが、最適な転送パフォーマンスを達成するには一致させることを推奨します。

```
net.core.rmem_max = 134217728 net.core.wmem_max = 134217728
```

## 手順2：コンソールを使ってExaCCデータベースを作成する

コンソールからデータベースを作成すると、スタンバイ・データベースによって使用される新しい RDBMS ホームが確立されます。ExaCC コンソールを使って、プライマリ・データベースと一致しないダミー・データベース名のデータベースを作成します。スタンバイ・データベースのインスタンス化が完了したら、このダミー・データベースを削除して、Oracle ホームをそのまま残します。『Administering Oracle Database Exadata Cloud Service』ガイドの「Creating a Database Deployment」に従ってください。

## 手順3：Oracle Exadata Cloudに必要な$ORACLE_HOME/network/adminを構成する

ExaCC 環境は、TNS_ADMIN 環境変数をユーザーの環境に設定し、TNS_ADMIN 環境変数をクラスタウェア環境変数として各データベースに設定することで、RDBMS ホーム内の各データベースの tnsnames.ora ファイルと sqlnet.ora ファイルを別々に維持します。この変数が両方の場所で正しく設定されていないと、透過的データ暗号化と SQL*Net 接続で予期しない結果が発生します。

1. TNS_ADMIN ディレクトリ、$ORACLE_HOME/network/admin/<STANDBY db_unique_name>を ExaCC の各ノードに作成します。

```
$ mkdir -p /u02/app/oracle/product/18.0.0.0/dbhome_3/network/admin/<STANDBY
db_unique_name>
```

2. 作成した ExaCC データベースの TNS_ADMIN ディレクトリから既存の sqlnet.ora ファイルを、新たに作成した TNS_ADMIN ディレクトリにコピーします。

```
$ cp /u02/app/oracle/product/18.0.0.0/dbhome_3/network/admin/<db_unique_name of
created db>/sqlnet.ora
/u02/app/oracle/product/18.0.0.0/dbhome_3/network/admin/<STANDBY
db_unique_name>/sqlnet.ora
```

3. コピーした sqlnet.ora ファイルを編集し、ENCRYPTION_WALLET_LOCATION と WALLET_LOCATION を編集します。

```
ENCRYPTION_WALLET_LOCATION =
(SOURCE=
  (METHOD=FILE)
    (METHOD_DATA=
      (DIRECTORY=/var/opt/oracle/dbaas_acfs/<STANDBY db_unique_name>/tde_wallet)))

WALLET_LOCATION =
(SOURCE=(METHOD=FILE)(METHOD_DATA=(DIRECTORY=/u02/app/oracle/admin/<STANDBY
db_unique_name>/db_wallet)))
```

4. 暗号化ウォレット・ディレクトリを作成します。ディレクトリは ACFS 上にあるため、クラスタの全ノードで共有されています。したがって、1 つのノード上だけで作成すれば済みます。

```
$ mkdir -p /var/opt/oracle/dbaas_acfs/<STANDBY db_unique_name>/tde_wallet
```

5. SSL ウォレット・ディレクトリを作成します。このディレクトリはローカル・マウント・ポイント上にあるため、クラスタの全ノード上で作成する必要があります。

```
$ mkdir -p /u02/app/oracle/admin/<STANDBY db_unique_name>/db_wallet
```

## 手順4：TDEウォレットをコピーする

$ORACLE_HOME/network/admin/sqlnet.ora に以下の行が含まれていること、ウォレット・ファイルの場所が sqlnet.ora で ENCRYPTION_WALLET_LOCATION パラメータとして定義されていることを確認します。

**SQLNET.ORA on on-premise host**

```
ENCRYPTION_WALLET_LOCATION =
  (SOURCE=
   (METHOD=FILE)
     (METHOD_DATA=
      (DIRECTORY=/var/opt/oracle/dbaas_acfs/<STANDBY db_unique_name>/tde_wallet)))
```

オンプレミスの ewallet.p12 ファイルと cwallet.sso ファイルを ExaCC ホスト上の上記のディレクトリにコピーします。

*注：プライマリ・オンプレミス・データベースのウォレットの場所を見つけるには、v$encryption_wallet view で問合せます。*

**オンプレミス・ホスト上**

```
scp -i ~/<ssh_key> ewallet.p12 opc@<ExaCC Host>:/tmp
scp -i ~/<ssh_key> cwallet.sso opc@<ExaCC Host>:/tmp
```

**ExaCC ホスト上**

```
$ chmod 777 /tmp/ewallet.p12
$ chmod 777 /tmp/cwallet.sso
$ sudo su - oracle
$ cp /tmp/ewallet.p12 /var/opt/oracle/dbaas_acfs/<STANDBY db_unique_name>/tde_wallet
$ cp /tmp/cwallet.sso /var/opt/oracle/dbaas_acfs/<STANDBY db_unique_name>/tde_wallet
chmod 600 /var/opt/oracle/dbaas_acfs/<STANDBY db_unique_name>/tde_wallet
```

## 手順5：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・データベースはバックアップまたは既存のアクティブ・プライマリ・データベースからインスタンス化することができます。RMAN RESTORE...FROM SERVICE メソッドを使って、RDBMS 12.1 以上のデータベースをアクティブ・プライマリ・データベースからインスタンス化するには、MOS Note 2283978.1 を使用します。この方法は、スタンバイをインスタンス化するもっとも簡単な方法で、MAA で推奨されている方法でもあります。

*注：RMAN リストア・コマンドの間、正しい tnsnames.ora が選択されるように、Oracle ユーザーの環境で TNS_ADMIN を必ず適切に設定してください。*

*注：スタンバイ・データベースの pfile でデータベース・パラメータ、diagnostic_dest='/u02/app/oracle' を必ず設定してください。*

RDBMS バージョン 11.2 の場合、アクティブ・プライマリ・データベースからインスタンス化するには、RMAN DUPLICATE メソッドを使用する必要があります。詳しくは、MOS 1617946.1 を参照してください。

データベースを Oracle クラスタウェアで登録したら、必ず TNS_ADMIN ディレクトリとクラスタウェア環境変数を設定してください。

```
$ srvctl setenv database -d <STANDBY db_unique_name> -env
'TNS_ADMIN=/u02/app/oracle/product/18.0.0.0/dbhome_3/network/admin/<STANDBY
db_unique_name>'
```

### 手順6：'ダミー'データベースを削除する

スタンバイ・データベースを適切にインスタンス化したら、リンク先ドキュメントで定義されているプロセスに従って、手順 2 で作成したダミー・データベースを削除できます。

### インスタンス化の完了

このインスタンス化が完了すると、結果は次のようになります。

» Hybrid DG config が構成済み

» クラウド・インフラストラクチャが管理され、非 DB ライフサイクル操作（バーストなど）が実行可能

» この"スタンバイ・データベース"の DG 操作と DB 管理は手動

## 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。また、（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ベスト・プラクティスです。DR システムは本番システムと同様にサイジングされるので、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 から、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
```

*注：スイッチオーバーを実行する前に、まず、スナップショット・スタンバイ・データベースをフィジカル・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 から、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 のロール移行は、いつでも手動で実行できます。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。

## オンプレミスへのスイッチバック

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。

## 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。

» ORAchk：汎用プラットフォームに適用可能（Database Cloud Service に最適）

» exachk：Oracle Exadata Database Machine に適用可能（Exadata Cloud Service に最適）

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。

### Data Guard 固有の問合せ（Oracle Database 11g 以降で利用可能）

Data Guard 固有の一連の問合せが MOS 2064281.1 に記載されています。

### Data Guard の VALIDATE DATABASE（Oracle Database 12c 以降で利用可能）

Data Guard に固有のヘルス・チェックをもっとも包括的に実施するには、Data Guard Broker の VALIDATE DATABASE コマンドを使用することを強く推奨します。VALIDATE DATABASE により、構成チェックが広範に実施され、構成でスイッチオーバーやフェイルオーバーの準備ができているかどうかが検証されます。

例：

```
DGMGRL> validate database APPDBB;
     Database Role:    Physical standby database
     Primary Database: pri

     Ready for Switchover:     Yes
     Ready for Failover:       Yes (Primary Running)
```

VALIDATE DATABASE コマンドで実行される広範なチェックの詳細については、Data Guard Broker のドキュメントを参照してください。

## クライアント・フェイルオーバー

クライアント・フェイルオーバーの自動化とは、障害の後にアクティブなプライマリ・データベースにクライアントを再接続するプロセスのことです。Data Guard フェイルオーバーの一部として新しいプライマリ・データベースにデータベース・サービスを再配置する処理、TCP のタイムアウトが起こらないよう、障害が発生したことをクライアントに通知する処理、新しいプライマリ・データベースにクライアントをリダイレクトする処理が含まれます。構成の詳細は、継続的な可用性の MAA ベスト・プラクティスに記載されています。詳細を入手するには、リンクにアクセスしてください。

## 結論

Exadata Cloud at Customer システムを使った Hybrid Data Guard は、ディザスタ・リカバリの準備を達成する経済的な方法です。Maximum Availability Architecture のベスト・プラクティスは、データの保護と可用性へのベスト・ソリューションを保証します。

## 付録A：MAAベスト・プラクティスのパラメータ設定

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。以下のパラメータをプライマリ・データベースとスタンバイ・データベースの両方に設定する必要があります。

- ARCHIVELOG 有効化
- Flashback database オン
- FORCE LOGGING 有効化
- SPFILE を使用
- Data Guard Broker を使用
- COMPATIBLE では小数点第 4 位まで使用し、両方のデータベース上で同じ値にすること
- DB_FILES=1024
- オンライン REDO ログの特性
    - 通常の冗長ストレージでは多重化のみ。高冗長性を使用する場合は単一メンバー・グループ
    - スレッド当たり最小 3 つのオンライン・ログ・グループ
    - 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
    - 単一メンバーのみ
    - 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
    - DATA ディスク・グループ上に存在
- LOG_BUFFER = 128M（11.2 の場合）、256M（12.1 以上）
- DB_BLOCK_CHECKING=OFF 注：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てから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=120 以上
- FAST_START_MTTR_TARGET=300
- USE_LARGE_PAGES=HugePages が構成されており、オンプレミス・システムで適切なサイズに調整される場合は ONLY
- CLUSTER_INTERCONNECTS（gv$cluster_interconnects ごとに設定）
- PARALLEL_THREADS_PER_CPU=1
- DB_CREATE_ONLINE_LOG_DEST_1= DATA ディスク・グループ

- DB_CREATE_ONLINE_LOG_DEST_n（DATA が高冗長性ではないときにだけ、1 以外の値を設定すること）
- DB_CREATE_FILE_DEST（DATA ディスク・グループを使用）
- DB_RECOVERY_FILE_DEST（RECO ディスク・グループを使用）
- Recyclebin オン

**Oracle Corporation, World Headquarters**
500 Oracle Parkway
Redwood Shores, CA 94065, USA

**海外からのお問い合わせ窓口**
電話：+1.650.506.7000
ファクシミリ：+1.650.506.7200

CONNECT WITH US

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

Integrated Cloud Applications & Platform Services



ホワイト・ペーパー・タイトル
2019 年 9 月
著者：Kazuhiro Ikeda、Sebastian Solbach 共著者：Ramachandran Pandrapattahil

Oracle is committed to developing practices and products that help protect the environment