Oracle Maximum
Availability Architecture

OCI上のSOA Cloud Serviceのディザスタ・リカバリ

クラウドでの本番環境とDR

Oracleホワイト・ペーパー | 2021年3月

ORACLE

本書の目的

本書には、Oracle SOA Cloud Service on OCIのディザスタ・リカバリ・ソリューションの構成についての説明、要件の要約、セットアップ手順が記載されています。このホワイト・ペーパーの対象読者は、Oracle Cloud、Oracle SOA Suite、Oracle WebLogic、Oracle Database、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。

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、実装および記載されている製品機能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。

改訂履歴

このホワイト・ペーパーには下記の改訂が行われてきました。

日付	改訂	コメント
2020年6月	1	以前の改訂履歴なし
2020年9月	2	「ベスト・プラクティス」セクションを追加 新しい「付録E - その他のライフサイクル操作」（検証でスタンバイを開く、サーバーを停止させる） DRSツールのためのOracle RACのサポートとOracle RACのための手動DG構成を追加 付録Cに、自動化されたData Guardでのリストアに関する注意点を追加
2021年1月	3	「検証のためのセカンダリ・サイトを開く」に注意点を追加 入力ミスの訂正と表現の改善
2021年3月	4	付録Eに「スタンバイDBシステム再作成」の要点を追加

目次

はじめに

Oracle Maximum Availability Architecture（Oracle MAA）は、オンプレミス、プライベート・クラウド、パブリック・クラウド、またはハイブリッド・クラウドにデプロイされるオラクル製品（データベース、Oracle Fusion Middleware、アプリケーション）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 Maximum Availability Architectureのベスト・プラクティスは、Oracleデプロイメントの重要な要件の1つとなっています。Oracle Fusion MiddlewareとOracle Databaseには、プロセス停止の検出と再起動、クラスタ化、サーバーの移行、クラスタウェアの統合、GridLink、ロードバランシング、フェイルオーバー、バックアップとリカバリ、ローリング・アップグレード、ローリング構成の変更など、豊富な高可用性機能が含まれています。これらは、アプリケーション環境を計画外停止時間から保護し、計画停止時間の必要性を最小限に減らすことができます。

Oracle Service-Oriented Architecture Cloud Service（SOACS）は、Oracle SOA Suiteをクラウド内で実行するためのPaaS（Platform as a Service）コンピューティング・プラットフォーム・ソリューションです。Oracle Managed File Transfer Cloud Service（Oracle MFTCS）は、標準に基づく高パフォーマンスなエンド・ツー・エンドのマネージド・ファイル・ゲートウェイです。これら2つのコンピューティング・プラットフォームの基本インフラストラクチャとして、Oracle Compute Cloud Service、Oracle Database Cloud Service、Oracle Java Cloud Serviceが使用されます。いずれのプラットフォームでも、Oracle Platform Security Servicesの情報、インスタンス追跡、コンポジットとドキュメント・メタデータ、および他のOracle FMW Infrastructureスキーマを格納するためにOracle Databaseが必要です。標準的なOracle SOAデプロイメントでは、トランザクション一貫性の保持と管理の簡素化を目的として、アプリケーション・データ（アプリケーション固有のスキーマ、jmsストアなど）とSOA固有のスキーマを同じデータベースに格納します。SOACS/MFTCSインスタンス内では、Oracle Database Cloud Serviceインスタンスを使用してこれらのスキーマを格納します。どのOracle SOAデプロイメントも予期せぬ障害や自然災害から保護する必要があり、Oracle Cloudにデプロイされたシステムも同じように保護する必要があります。しかも、中間層（Oracle SOA Suite Cloud Service）とデータ層（Oracle Database Cloud Service）の両方への対処が必要です。そのためのソリューションとしては、本番サイトとは地理的に異なる場所にあるOracleクラウド・データセンターにスタンバイ・システムをセットアップする必要があります。スタンバイ・システムのサービスとリソースは、本番サイトと同じにすることも、少なくすることもできます。スタンバイ・システムは通常パッシブ・モードになっており、プライマリ・サイトが使用できなくなるとアクティブ化されます。このデプロイメント・モデルは、アクティブ-パッシブ・モデルとも呼ばれます。通常、このモデルを採用するのは、2つのサイトがWAN経由で接続されていて、ネットワーク待機時間の関係で2つのサイトをまたぐクラスタ化ができない場合です。

Oracle SOA Cloud Service（Oracle SOACS）では、Oracle Fusion MiddlewareとOracle Databaseに搭載されている高可用性機能とディザスタ・リカバリ機能を活用してリカバリ時間目標（RTO）とリカバリ・ポイント目標（RPO）を最適化します。クラウド・ディザスタ・リカバリ（DR）構成に特有の考慮事項もありますが、Oracle Fusion Middleware（FMW）とOracle Databaseを使用する他のデプロイメントの場合と同じOracle MAAベスト・プラクティスがこの構成にも適用されます。このOracle MAA構想では、Oracle MAAベスト・プラクティスについて詳しく説明し、Oracle SOA Cloud Serviceに対応したDRをデプロイする手順の概要を示します。Oracle SOA Cloud Serviceディザスタ・リカバリ・ソリューションでは、SOAコンポーネントのブートストラップに必要なごく一部の構成ファイルをレプリケートします。アプリケーションによっては、他の構成ファイルをレプリケートすることが必要な場合もあります。このホワイト・ペーパーでは、さまざまなアプリケーション・パラダイムに適合するように複数のオプションを紹介します。Oracle SOA Cloud Serviceで使用するOracle Database Cloud Serviceは、Oracle Data Guardを使用して災害から保護します。

このホワイト・ペーパーでは、Oracle Cloud Infrastructure上のOracle SOA Cloud ServiceとDBシステムを使用して説明します。構成画面やセットアップ手順がさまざまな項で出てきますが、それらはこの新しいIaaSプラットフォームの機能に基づいています。旧バージョンのIaaSでディザスタ・リカバリを有効化する方法については、OCI ClassicでのSOACSのディザスタ・リカバリに関するホワイト・ペーパーを参照してください。SOA on Marketplaceの場合は、『SOA Suite on Oracle Cloud Infrastructure Marketplaceのディザスタ・リカバリ』ホワイト・ペーパーを参照してください。

このホワイト・ペーパーの対象読者は、Oracle WebLogic Server、Oracle FMW SOA、Oracle Database、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で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[1]。

1 https://www.oracle.com/jp/index.html

Oracle SOA Cloud Serviceのディザスタ・リカバリに関する考慮事項

Oracle SOA Cloud ServiceはOracle Database Cloud Service（Oracle DBCS）を使用してメタデータとSOAインスタンスの情報をホスティングします。Oracle Cloudでさまざまなデータベース・モデルとデータベース・サービスを使用できます。このホワイト・ペーパーでは、使用される例および構成として、データベース・システムVMをOCIで使用する場合に特化して焦点を当て、使用します。Oracle Database Cloud Serviceを災害から保護する目的で、次の2つの異なるソフトウェア・エディションを使用できます。

- Enterprise Edition ServiceまたはHigh Performance Serviceを利用するData Guard。
- Extreme Performance Serviceを利用するActive Data Guard。

Oracle Active Data Guardでは、Data Guardに組み込まれているすべての機能を使用できるほか、読取り操作にスタンバイ・データベースを使用することができます。Data GuardおよびActive Data Guardについて詳しくは、Oracle Technology NetworkのData Guardホームページおよび『Oracle Active Data Guard』ホワイト・ペーパー[2]を参照してください。

Oracle FMW SOA Suiteでは、自動ブロック修復、高速増分バックアップ、高速ローリング・アップグレード、Far Syncなど、Active Data Guardのほとんどの機能を利用できます。ただし、SOAサーバーではOracle Active Data Guardの読取り専用問合せ機能を使用できません。Oracle SOA Serverは読取り専用モードで実行できないためです。SOA Serverを起動するとただちにサーバーからデータベースへの接続が確立され、ビジネス・フローの処理が始まります（つまり、データベースへの"書込み"が即座に開始されます）。スタンバイ・サイトのSOAサーバーを起動するには、1）データベースのスイッチオーバーまたはフェイルオーバーを実行するか、2）スタンバイ・データベースをスナップショット・スタンバイに変換してスタンバイをテスト用として一時的に読取り/書込み可能にするかのいずれかしかありません。プライマリから受信したREDOは格納されますが、適用されません。スタンバイで実行した変更はテストの最後で破棄され、プライマリ・データベースから取得したREDOの適用はこの時点で再開できるようになります。ただし、注意しなければいけない点として、スタンバイをスナップショット・モードにするとファスト・リカバリ領域に必要なストレージが増えるということがあります。プライマリの本番データベースから受信したアーカイブREDOを後の使用に備えて保持する必要があり、スナップショット・スタンバイで生成される最新のREDOログとフラッシュバック・ログも保持する必要があるためです。

SOACSのディザスタ・リカバリ（SOACS DR）をセットアップする場合、WebLogic Server（WLS）ドメインの構成変更がさほど頻繁でなければ、本番サイトに適用した構成変更のレプリケーションにスナップショット・スタンバイを使用できます。手順としては、スタンバイ・データベースをスナップショット・スタンバイに変換した後、セカンダリ・ロケーションのWLS管理コンソールで変更を適用し直します（手順については、後で詳しく説明します）。これにより、ファイル・システムのアーティファクト（earファイル、デプロイ・プランなど）が更新され、プライマリ・サイトと同じ状態になります。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。ただし、SOAサーバーでスナップショット・スタンバイを使用する場合は注意が必要です。SOAサーバーからデハイドレーション・ストアにアクセスできるようになっていると、保留しているインスタンスの処理が意に反して実行される可能性があるからです。DRシステムは本番システムと同様にサイジングされることが多いため、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。Data Guardスナップショット・スタンバイ[3]について詳しくは、オラクルのドキュメントを参照してください。さらにOracle Cloudは、プライマリ・データベースが何らかの理由で使用できなくなった場合のディザスタ・リカバリで必要とされるバックエンド・インフラストラクチャと機能をすべて備えています。これには、次のものが含まれます。

1. スタンバイ・データベースを監視し、大きな問題の発生時にアラートを発信する機能。
2. スタンバイ・データベースをアクティブ化してDRの準備状況を検証し、再度変換して同期されたスタンバイ・データベースに戻す機能。
3. オンプレミスのデプロイメントと本質的に同じOracle MAAのベスト・プラクティスの利用。このホワイト・ペーパーでは、クラウド・デプロイメントに固有の考慮事項をいくつか説明します。

2 http://www.oracle.com/technetwork/jp/database/availability/active-data-guard-wp-12c-1896127-ja.pdf

3 http://www.oracle.com/pls/topic/lookup?ctx=en/database/oracle/oracle-database/12.2/sbydb&id=GUID-B140C38B-DE01-4252-8422-7154018DDFEC

4. 計画メンテナンスや計画外停止の期間中に、本番データベースをクラウド内のスタンバイ・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。障害が発生したプライマリ・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新しい本番データベースと再同期させ、ロールを元のステータスにスイッチバックすることができます。

5. サイトが完全に停止した場合に、SOA/中間層とデータベースの両方をフェイルオーバーし、本番アプリケーションをOracle Cloudで実行できるようにする機能。

中間層でSOACS/MFTCS DRシステムのアーキテクチャを構成するのは、Oracle Database Serviceインスタンスと同じサイトにデプロイされる2つのSOACSインスタンスです。それぞれのSOACSインスタンスでは、SOAクラスタと管理サーバーを使用し、フロントエンドのロードバランサとしてOTD/OCILBR[4]を使用します。また、単一のデータベース（Oracle Database Cloud Service）インスタンスを使用して、両方のSOACSスキーマ（MDS、コンポジット・デプロイメント、SOAINFRAスキーマなど）、JMS永続ストア、トランザクション・ログ永続ストア、およびアプリケーション固有のデータをホスティングします。図1にこのアーキテクチャを示します。

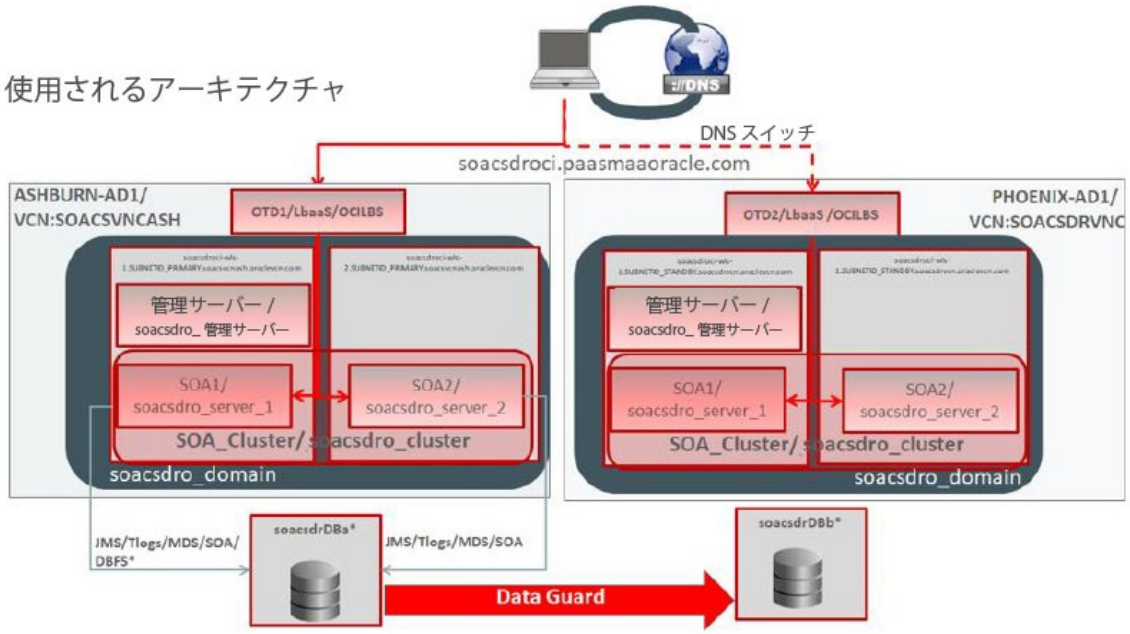


図1：SOACS/MFTCS DRアーキテクチャ

このトポロジはさまざまな方法で作成できます。アプローチごとにいくつかの意味があります。このホワイト・ペーパーでは、可用性、プロビジョニング・オーバーヘッド、ライフサイクルの観点からもっともメリットが大きいと考えられているセットアップ・プロセスについて説明します。推奨されているアプローチについては、この後の項で説明します。

4 このホワイト・ペーパーの例では、Oracle Traffic Director（OTD）をフロントエンド・ロードバランサとして使用し、テストもこれで行っていますが、まったく同じWeb層コンポーネントがSOACSでサポートされていれば、このドキュメントで説明しているものと同じ仮想サーバー、証明書およびルーティング変更がLoad Balancer Classic（LbaaS）およびOracle Cloud Infrastructure Load Balancingサービス（OCILBR）に適用されます。

サービス・レベル要件

Oracle SOA Cloud Serviceはユーザー管理の環境です。ユーザーは、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。

- **可用性**：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、停止の検出に要する時間と、データベース、Web層、SOAサーバーをフェイルオーバーしてサービスを再開するまでに要する時間が含まれます。

- **データ保護**：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。SOAの場合でRPOが特に関係するのは、トランザクション・ログ、JMSメッセージ、SOAインスタンス情報です。これらはすべて同じデータベースに格納されているためです。実際に達成可能なRPOは次の要因によって異なります。

  » 使用可能なネットワーク帯域幅。

  » ネットワークの信頼性。

  » 使用するData Guard転送方式：*非同期*（データ損失ほぼゼロの保護を実現）または*同期*（データ損失ゼロの保護を実現）のいずれか。

- **パフォーマンス**：スタンバイ・システムにプロビジョニングされている容量（コンピュート、メモリ、I/Oなど）がプライマリ・システムより少ないと、データベースと中間層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。これは、コストを削減するためにユーザーが意図的にスタンバイ・リソースを控えめに構成し、DRモードの間はサービス・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、Web、アプリケーション、データベースの各層で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。クラウドでは迅速なプロビジョニングが可能なため、当初は容量を少なめにデプロイし、フェイルオーバーが必要な場合は新しいプライマリを迅速にスケール・アップするという妥協案が可能です。

注：Oracle Cloudでは、DR関連のサービス・レベルに関係なく、該当するDatabase Cloud Serviceで定義されているサービス記述に従ってすべてのデータベース・インスタンスが作成されます[5]。

セキュリティ要件

Oracle MAAのベスト・プラクティスでは、保管中のプライマリ・データベースとスタンバイ・データベースをオラクルの透過的データ暗号化（TDE）機能で暗号化することを推奨しています。TDEに変換すると、保管中のすべてのDATA/INDEX表領域の自動暗号化と、ユーザー・データのREDO変更をクラウドにレプリケーションするときの移動時暗号化を有効化できます。TDEによって暗号化されない他のREDO変更（SYSTEM、SYSAUXといった暗号化されていない表領域のデータなど）の移動時暗号化には、Oracle Netの暗号化も必要です。DBFSを使用してサイト間でWLSドメイン構成をレプリケートする場合は、DBレベルで適切な暗号化を使用することにより、サイト間の構成伝播のセキュリティも保証されます。

5 http://www.oracle.com/us/corporate/contracts/paas-iaas-public-cloud-2140609.pdf

注：Data GuardおよびActive Data Guardでは、REDOベースのレプリケーションと呼ばれるプロセスが使用されます。このプロセスでは、プライマリ・データベースで生成されたREDOがスタンバイ・データベースに転送され、その変更内容が連続メディア・リカバリによってスタンバイ・データベースに適用されます。つまり、プライマリ・データベースとスタンバイ・データベースはブロック同士が完全に一致する相互コピーです。TDEを使用してスタンバイ・データベースを暗号化する場合は、プライマリ・データベースもTDEで暗号化する必要があります。

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。TDEの場合は、データベースのキャッシング機能を利用し、IntelやSPARCのCPUで処理されるAES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、オーバーヘッドを最小限に抑えます。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。詳しくは、『Oracle Database Advanced Securityガイド[6]』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はREDO転送圧縮を有効化しないようにする必要があります。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。数多くの暗号化データベースを使用する場合は、オンプレミスでOracle Key Vault[7]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。

構成要件

SOACS/MFTCS DRをセットアップする場合は、上に示した実行時関連の側面以外にも以下の点を考慮することが重要です。

- SOACSインスタンスの名前は、プライマリとスタンバイの両ドメイン/ロケーションで同一にする必要があります。インスタンス名は、Oracle SOACSを実行するホスト名、WLSサーバー名、ドメイン名を作成するときに使用されます。同じインスタンス名を使用することで一貫性が保証されます。JMSメッセージとTLOGをリカバリするには、インスタンス名が同じである必要があります。インスタンス名が同じであれば、両サイトでのカスタマイズや操作も容易になります。ただし、Oracle SOA Cloud Serviceでは、同じサービス・タイプの場合、単一テナンシーで同じインスタンス名を何度も使用することはできません。つまり、同じアカウントでは、同じ名前の2つの異なるSOACSインスタンスを使用できません。システムをゼロから作成する場合に、プライマリSOACSインスタンスに使用する名前を自由に決められるのであれば、プライマリとスタンバイのインスタンスで最初の8文字が共通の1つのインスタンス名を使用することにより、この問題を解決できます。（SOACSのWebLogicドメインとWebLogic Serverの名前は最大8文字です。ただし、SOACS Cloudのインスタンス名にはこのような制限がありません）。単一テナンシーを使用してSOACS DRシステムを構成可能な場合は、インスタンス名の例と2つの異なるテナンシーを使用しなくてもよくするために使用できる手法については、「付録B - 単一テナンシーを使用したDRシステムの作成」を参照してください。

6 http://www.oracle.com/pls/topic/lookup?ctx=en/database/oracle/oracle-database/12.2/asoag&id=GUID-81BF34FA-6044-47D4-BF31-12DEF178BA6B
7 https://www.oracle.com/jp/security/database-security/key-vault/

その他のすべての場合（つまり、プライマリ・システムで8文字のインスタンス名を使用している場合）は、以下を適用してSOACS/MFTCS DRが構成されます。

1. 2つの異なるテナントを使用する必要がある
2. したがって、OCIコンソールで提供される自動化されたData Guard構成オプションは使用できない（テナンシーが異なる場合は使用不可）

- SOACSに必要なOracle Database Cloud Serviceを作成すると、データベース・インスタンスが自動的に作成されます。スタンバイ・サイトのData Guardを手動構成する場合、スタンバイ・サイトで最初に作成されたデータベースをData Guardスタンバイ・データベースとして使用することができません。このデータベースはDRのセットアップ・プロセス中に削除します。
- このホワイト・ペーパーのバージョン2以降、説明されているDRのセットアップ手順はOracle RACデータベースでの動作が保証されています。

  Oracle Autonomous Processing（ATP）は、このドキュメントでは扱いません。ATPはまだクロスリージョンのディザスタ・リカバリ保護に対応していないため、Oracle SOACSのディザスタ・リカバリ・トポロジで使用することはできません。
- MDS、コンポジット・デプロイメント、ポリシーはデータベースに格納されているため、これらのサイト間同期はData Guardによって自動的に行われます。
- SOACSで使用するWLSドメイン構成のほとんどは、DBFSを使用して最初にサイト間で同期します。このとき、以下の点を考慮します。

  1. DRのセットアップが完了した後も、ローカルDBへの接続に使用する元のJDBC URLはSOACSシステムごとに管理されます。両方の場所から同じスキーマが参照されるようにするために、スキーマ接頭辞のみを変更します。
  2. *weblogic_domain_name/config*の下の構成はすべて、WLSインフラストラクチャによって同じサイト内の他のノードに配布されます。
  3. カスタム・アプリケーション・デプロイメント（ワークフロー・タスクear、カスタムearファイル、デプロイメント・プランの更新、JMSリソース、再デプロイメント）および管理サーバーのWLSドメイン・ディレクトリの下にある（一時データ以外の）あらゆるものが、サイト間で最初に同期されます。他のノードまたはWebLogic管理サーバーのドメイン・ディレクトリ外に存在するデータは、セカンダリ・ロケーションに手動でコピーする必要があります。
- SOACS/MFTCS DR構成は、サービスにアクセスしているエンドユーザーやシステムに認識されずにフェイルオーバーやスイッチオーバーが行われるように、使用中のサイトがSOAエンド・ポイントに依存しないように構成します。これを可能にするために、構成が完了したら、既存のSOA/MFTクラスタのフロントエンド・アドレスを変更し、いずれのDRロケーションのフロントエンド・ロードバランサにも割当て可能な仮想ホスト名を使用します。いずれかのサイトにフロントエンドURLが割り当てられるようにするには、適切なDNSサービス（Oracle Cloud DNS、商用DNS、ローカルDNSサーバーまたは手動ファイルによるホスト名解決）を使用する必要があります。このフロントエンド・アドレス変更を行う前にデプロイまたは使用されたコンポジットがある場合は、この新しい仮想ホスト名と一致するようにエンドポイント・アドレスを変更することが必要になる場合があります（注：デフォルトでは、SOAクラスタまたはWLSクラスタの*HTTPフロントエンド*・パラメータに基づいてエンドポイント・アドレスが構成されます）。

- 既存システムのフロントエンド・ホスト名アドレス（すでに存在する場合）をディザスタ・リカバリ用の仮想ホスト・アドレスとして使用することが可能です。つまり、元のシステムで*mysoacs.example.com*がプライマリの仮想URLとして使用されていた場合は、スイッチオーバー後にこの同じ仮想ホスト名をセカンダリ・システムにマッピングできます。
- OTD/OCILBR層は、初期構成でも十分に、SOACSサービスに必要なデフォルト・ルーティング、仮想サーバー、プールを維持できます。OTD/OCILBRのカスタム構成はサイト間でレプリケートされません。一方のサイトでロードバランサの構成変更を適用した場合は、他方のサイトで同じことを繰り返す必要があります。
- プライマリとセカンダリのSOACSシステムで使用するデータベースのData Guard構成が完了したら、"フィジカル・スタンバイ"から"スナップショット・スタンバイ"への変換は、スタンバイ・サイトのSOAサーバーを起動せずに行うか、SOAデータベースの"ドレイン/切捨て"後にのみ行う必要があります。これは、再実行される可能性がある保留中のメッセージを除去するために必要です。[8]
- REDO転送のために、プライマリ・データベースとセカンダリ・データベースがそれぞれのリスナー・ポート経由で相互に通信する必要があります。初期セットアップのために、中間層もリモート・データベースと通信する必要があります。この通信はインターネット・ゲートウェイを介して行うことができます（Oracle Netのトラフィックは暗号化されます）。より優れたアプローチとして、プライマリ・サイトとスタンバイ・サイト間の通信は動的ルーティング・ゲートウェイを使用した内部ネットワークによって実現でき、この方法が推奨されます（ネットワーク構成についてさらに詳しくは、「動的ルーティング・ゲートウェイ」に関するドキュメントを参照）。使用する方式に応じて、該当する受信ルールを有効にする必要があります。これについてさらに詳しくは、OCIドキュメントの「セキュリティ・ルール」セクションで説明されています。

SOACS/MFTCS DRのデプロイメント

このホワイト・ペーパーでは、単一インスタンスのDBシステム、SOACS Cluster、Oracle Traffic Director（OTD）で構成されるプライマリ・サイトがすでに稼働している状態を出発点とします。この既存のプライマリ・システムに対するセカンダリDR構成を、地理的に離れたサイトに作成します。初期セットアップのプロビジョニング方法について詳しくは、Oracle SOACSのドキュメントを参照してください。まだプライマリ・システムが作成されていない場合は、付録Bを参照して、単一テナンシーでのDR構成の作成を可能にするインスタンス名の使用方法を確認してください。すでにプライマリ・システムが存在し、9文字以上のSOACSインスタンス名を使用している場合も、付録Bを参照してください。同じテナンシーにスタンバイを作成する方法について説明しています。上記に該当しない（プライマリ・システムが存在し、9文字以上のサービス名を使用していない）場合は、次に示す手順を実行します。

プライマリのSOACSですでに本番ワークロードが実行されている可能性があるため、DRの構成プロセスは停止時間が最小限になるように設計されています（実際、SOAサーバーの再起動が必要なのは、フロントエンド・アドレスの変更時のみです）。

次に示すのは、セットアップ・プロセスの手順の概略です。

1. 解決可能な仮想ホスト名をプライマリ・システムのフロントエンドとして割り当てる（または、既存のSOACSシステムですでに使用しているフレンドリーURLか架空のURL、またはDNS名があれば、それを再利用する）
2. プライマリSOACSクラスタのフロントエンド・アドレスを更新する
3. t3/rmi URLを更新する（使用する場合）
4. セカンダリ・データベースをセットアップする（OCIコンソールまたは手動でのData Guardの構成）
5. 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・データベースに変換する
6. 異なるテナンシーのスナップショットDBを参照するOracle SOACSをセカンダリ・ロケーションにプロビジョニングする

8 SOAサーバーはSOAインスタンスの処理を開始し、すでにプライマリ・サイトで処理が済んでいる可能性がある保留中の作業（コールバック、非同期呼出しなど）を完了させようとします。スナップショット・データベースのドレインが完了していない場合、すでに本番稼働しているシステムについては、管理サーバーのみをスタンバイ・ドメインで起動する必要があります。

7. ディザスタ・リカバリ・セットアップ（DRS）ツールを実行してSOACS DRを構成する
8. （任意）スタンバイにスイッチオーバーしてDRシナリオ全体を検証する

ここまでの手順については、以降の各項で詳しく説明します。

セットアップの詳細

SOACSシステムをスタンバイに作成する前に、プライマリ・システムの中間層およびデータベースのパッチとPSUを最新のものにすることを推奨します。より正確には、異なるDBドメインをまたぐDBシステムのData Guard構成にはバグ22611167の修正を適用する必要があります。opatch出力にチェックを入れることによってプライマリとセカンダリの両方のDBシステムでこのバグのパッチが適用されているかどうか確認し、未適用の場合には適用してください（最新のOCI 12cR2 DBシステムにはこのバグのパッチが事前インストールされており、上位バージョンではバグが修正されています）。

前のセクション「構成要件」で説明されている要件を再確認し、特にプライマリ・インスタンス（すでに存在することを前提）のインスタンス名を書き留めます。また、セカンダリ・ロケーションにプロビジョニングするSOACSのバージョンとパッチ・レベルが、プライマリ・サイトで稼働中のものと一致していることを確認してください。[9]

プライマリ・ロケーションのDBシステムとSOACSインスタンスにパッチを適用し、正しく動作することを確認したら、次の手順を順番に実行します。

1. フロントエンドOTD/OCILBRに仮想ホスト名を割り当て、プライマリ・サイトのフロントエンド・アドレスを更新する

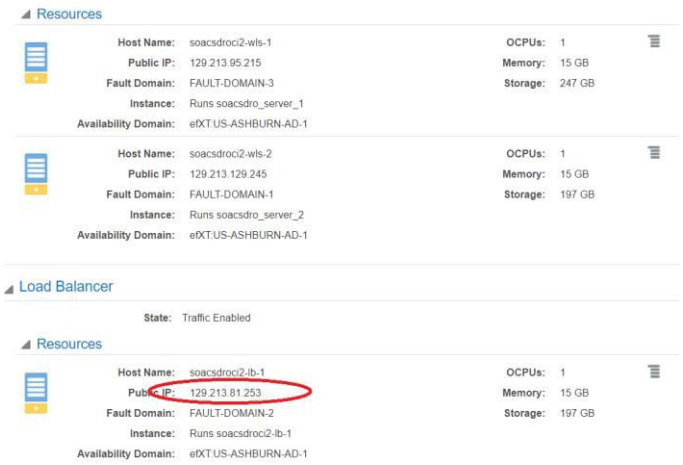
デフォルトでは、Oracle SOACSをプロビジョニングすると、SOAクラスタのフロントエンド・アドレスの値はOTD's//OCILBRのリスニング・アドレスのIPに設定されます。このIPは仮想ホスト名に置き換える必要があり、このホスト名は外部クライアントでも解決でき、OTD/OCILBRからリクエストをルーティングする宛先のWLSサーバーでも解決できる必要があります。仮想ホスト名を外部的に解決する場合は、ローカル・ホスト・ファイルまたは任意の公式なDNS解決を使用できます。仮想ホスト名をSOACSインスタンスの有効範囲内で解決する場合は、ローカル・ホスト・ファイル解決またはOracle DNS Cloud Serviceのいずれかを使用できます。この最後のケースでは、仮想ホスト名のレコードをホスティングする新しいDNSサーバーを使用してSOAノードのresolv.confファイルを更新する必要があります。

A. SOACSクラスタの仮想ホスト名に対応するIPを特定するために、次の手順を順番に実行します。

- SOACSインスタンスのOracle WebLogic Server管理コンソールにログインします。
- 左ペインのDomain Structureウィンドウで「Environment」を選択し、「Clusters」を選択します。Summary of Clustersページが表示されます。クラスタ「SOA_Cluster」を選択します。
- 「HTTP」を選択します。
- *Frontend Host*フィールドで使用されているIPをメモします。

注：このIPは、SOACSインスタンスの画面に表示される情報から取得することもできます。次に示すように、このIPはロードバランサのパブリックIPとしてリストされます。

9 プライマリSOACSインスタンスがかなり以前に作成されている場合は、SOACSホストの基盤のOSイメージ・バージョンが、新しいセカンダリSOACSインスタンスにプロビジョニングされているOSイメージより古い可能性があります。その場合は、My Oracle Supportにお問い合わせください。



B. 仮想ホスト名にマッピングするIPを各ノードのDNS解決に追加します。SOACSノードのコマンド・シェル・プロンプトでsudoを実行してrootユーザーになり、/etc/hostsを編集してこのIPを仮想ホスト名にマップします（このIPの外部解決にDNSを使用する場合は、このホスト名をここで使用する必要があります。ファイルベース（/etc/hosts）のホスト名解決を使用する場合は独自に選択できます）。次に例を示します。

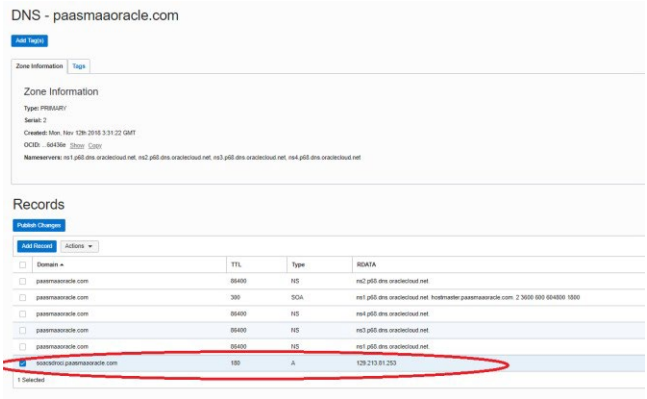
```
[oracle@soacsdroci-wls-1  ~]$ cat /etc/hosts
127.0.0.1          localhost localhost.localdomain localhost4 localhost4.localdomain4
::1                localhost localhost.localdomain localhost6 localhost6.localdomain6
10.0.0.11 drDb2a.sub10171440110.soacsvcnash.oraclevcn.com  drDb2a
129.213.81.253 soacsdroci.paasmaaoracle.com  soacsdroci
```

SOACSクラスタのメンバーをホスティングする他のすべてのSOACSサーバーでこの手順を繰り返します。

- 外部クライアントで仮想ホスト名を使用できるようにするには、DNSサーバーを更新するか、外部クライアントのhostsファイルを上記のとおり変更します。たとえば、オンプレミスのWindowsクライアントでは、C:\Windows\System32\drivers\etc\hostsファイルを上記と同じように編集します。

注：外部クライアント用のこの構成が機能するのは、インターネットからの直接接続を使用する場合です。カスタムのイントラネットからの接続では、ブラウザまたはhttpクライアントで使用される必要なプロキシ・サーバーにこのホスト名を追加する必要があります。

他に考えられる方法としては、Oracle DNS Zone管理を使用してフロントエンドのホスト名のマッピングを管理することもできます。この例では、適切なゾーンのレコードがフロントエンド・アドレスのホスト名となります。クライアント側では、/etc/resolv.confの名前サーバーのエントリを更新し、適切なDNSサーバーが使用されるようにする必要があります。



C. 上の手順に沿って適切なホスト名解決を整備したら、SOACSクラスタのフロントエンド・アドレスを変更します。

- SOACSインスタンスのOracle WebLogic Server管理コンソールにログインします。
- 左ペインのDomain Structureウィンドウで「Environment」を選択し、「Clusters」を選択します。Summary of Clustersページが表示されます。クラスタ「SOA_Cluster」を選択します。
- 「HTTP」を選択します。
- Frontend Host=virtual_hostname_created の値を設定します。上の例では *soacsdroci.paasmaaoracle.com*となります。
- 「Save」をクリックします。
- 変更を有効にするために、管理コンソールのChange Centerセクションで「Activate Changes」をクリックします。
- WebLogic管理コンソールを使用してSOACSクラスタを再起動し、フロントエンドの変更を有効にします。

2. t3/RMI URLを更新する（使用する場合）

SOACSクラスタでRMIを起動するときに使用するURLは、SOACS/MFTCS DR構成の各サイトで使用されるIPまたはホスト名に依存しない必要があります。JNDI URLは、通常の*host:port,host:port*形式を使用するのではなく、クラスタ構文を使用するように変更してください[10]。クラスタ構文は、*cluster:t3://cluster_name*です。たとえば、JMSアダプタのファクトリ・プロパティをこの構文を使用するものに変更する場合は、次の手順を実行します。

A. SOACSインスタンスのOracle WebLogic Server管理コンソールにログインします。

B. Domain Structureの左ペインで「Deployments」をクリックします。

C. 右側のペインのSummary of Deploymentsの下にある「JmsAdapter」をクリックします。

D. 「Configuration」タブをクリックします。

10 言うまでもなく、このアプローチを使用できるのはドメイン間の呼出しの場合のみです。SOAドメインの外部にあるt3/rmiクライアントではこのアプローチを使用することができず、セカンダリ・サイトへの切替えでは、host:portリスト形式の適切なDNSマッピングを使用する必要があります。JNDI InitialContext取得ではTCPロード・バランスを使用可能ですが、JMSクライアントからの後続のリクエストはhost:portに直接接続するため、この場合にはセカンダリ・サイトIPへのDNSマッピングも必要になります。

E. 「Outbound Connection Pools」タブをクリックしてoracle.tip.adapter.jms.IJmsConnectionFactoryを展開し、構成済みのコネクション・ファクトリを表示します。

F. 使用している特定のインスタンス（例：eis/wls/Queue）をクリックします。コネクション・ファクトリのOutbound Connection Propertiesが開きます。

G. 「Lock & Edit」をクリックします。

H. FactoryPropertiesフィールド（Property値の下の対応するセルをクリック）で、java.naming.provider.urlフィールドを、次のようなクラスタ構文を使用したものに変更します（残りのフィールドはそのままにしておきます）。

java.naming.provider.url= cluster:t3://*cluster_name*;

I. プロパティを更新したら「Save」をクリックします。Save Deployment Planページが表示されます。

J. デプロイメント・プランのロケーションを入力します。

K. SOACSノード1のデプロイメント・プランをSOACSノード2のまったく同じディレクトリ/ロケーションにコピーするか、SOACSシステムに存在するデフォルトのDBFSマウント・ポイントを、デプロイメント・プランをホスティングするロケーションとして使用します（SOACSクラスタ内のすべてのノードが /u01/soacs/dbfs/shareにアクセスできます）。

L. 「Save and Activate」をクリックします。

M. 「Deployments」をクリックします。

N. 「Lock & Edit」をクリックします。

O. 「JMS Adapter」を選択します。

P. 「Update」をクリックします。

Q. 「Update this application in place with new deployment plan changes」を選択し（このオプションにはデプロイメント・プランを指定する必要があります）、共有ストレージの場所に保存したデプロイメント・プランを選択します。クラスタに含まれるすべてのサーバーからこのプランにアクセスできる必要があります。

R. 「Finish」をクリックします。

S. 変更を有効化します。

SOACSシステムで使用するカスタムJNDIのURLが他にもあればそれらもすべて同様に更新し、SOACS DRシステムでスイッチオーバーまたはフェイルオーバーが発生したときにセカンダリ・サイトでもURLが有効であるようにしておく必要があります。

3. セカンダリ・データベースをセットアップする

セカンダリSOACSシステムは、プライマリ・システムで使用されているデータベースのData Guardスタンバイとして構成される予定のデータベースを参照することにより、通常の手順を使用してプロビジョニングされます。セカンダリSOACSをプロビジョニングする場合のコンテナとしてスタンバイ・データベースを使用するには、このフィジカル・スタンバイをスナップショット・スタンバイに変換する必要があります。

したがって、セカンダリSOACSシステムのプロビジョニング前に、セカンダリ・データベースをセットアップする必要があります。セカンダリ・データベースは、プライマリ・データベースのData Guardフィジカル・スタンバイとして作成されます。これを実現する方法のオプションには、OCIコンソールを使用してData Guardを有効化する方法（このドキュメントでは、"自動化Data Guard"と呼びます）と、dgmgrlコマンドを使用してスタンバイ・データベースを手動で構成する（このドキュメントでは、"手動Data Guard"と呼びます）方法の2つがあります。

推奨アプローチは、OCIコンソールを使用してData Guardを構成する方法（オプション1）です。この方法では、OCIコンソール・ユーザー・インタフェースと統合され、このコンソールを使用してDBシステム内のOracle Data Guard対応付けを管理することができます。また、Data Guardでのバックアップの標準構成を備えています。OCIコンソールを使用してData Guardを有効化するには、「オプション1）OCIコンソールを使用したData Guardの構成」の指示に従ってください。

何らかの理由（たとえば、プライマリとセカンダリで2つの異なるテナンシーを使用している）で使用環境においてData Guardを有効化する機能を使用できない場合（特定のDBシステムのフレーバー/エディションでのData Guardのリージョン横断機能の可用性を確認する方法については、Cloud DBシステムのドキュメントの「Oracle Data Guardの使用」セクションを参照）でも、このホワイト・ペーパーで示すスクリプトを使用してData Guardを手動で構成できます。これについては、「オプション2）Data Guardの手動構成」の手順を参照してください。

3.1 オプション1）OCIコンソールを使用したData Guardの構成

OCIコンソールを使用してData Guardを有効化する場合は、プライマリDBシステムで「Enable Data Guard 」をクリックすると、セカンダリDBシステムがフィジカル・スタンバイとして自動的にプロビジョニングおよび構成されます。このための要件は、両方のDBシステムが同じテナンシーとコンパートメント内に存在すること、両方のDBシステムが同じシェイプ・タイプであること、DBシステムが異なるリージョンに存在する場合はそれらがリモートVCNピアリングを介して接続されていることなどです。これらの要件について詳しくは、Oracle Cloud Infrastructureのドキュメントで「Oracle Data Guardの使用」を参照してください。

プライマリ・データベースに対してData Guardを有効にするには、OCIコンソールにログインし、プライマリDBシステムに移動して、プライマリ・データベースをクリックします。Data Guardは、"Data Guard Associations"セクションで有効化できます。セカンダリDBシステムの構成プロパティのほとんど（バージョンやDB名など）は、プライマリから継承されるため事前定義されていますが、一部の構成プロパティについては指定する必要があります。次の表に、これらのプロパティの例と要件を示します。

クラウド・アカウント構成プロパティ	既存のプライマリDBシステム/例	作成されるセカンダリDBシステム/例
Oracle Cloudテナンシー	XXXX/paasmaa	XXXX/paasmaa（必ずOCIコンソールを使用してData Guardを構成する場合と同じ値にする）
Oracle Cloudアカウント・ユーザー	XXXX/joe@acme.com	XXXX/joe@acme.com （同じ）
Oracle Cloudアカウント・パスワード	XXXX/Acme1234#	XXXX/acme1234#（同じ）

DBシステム構成プロパティ	既存のプライマリDBシステム/例	作成されるセカンダリDBシステム/例
7	XXXX/soacsdr	XXXX/soacsdr（必ず同じ値）
リージョン	XXXX/Ashburn	YYYY/Phoenix（異なる値でも可、DRの場合は異なるリージョンにすることを推奨）
可用性ドメイン	XXXX/efEXT:US-ASBURN-AD1	YYYY/ efXT:PHX-AD-3（各リージョンは異なることが想定されるため、プライマリとは別にすることが必要）
ピアDBのシステム名	XXXX/soacsdrDBa	YYYY/soacsdrDBb（プライマリと異なっていても可）
シェイプ	XXXX/VM.Standard2.1	XXXX VM.Standard2.1（プライマリと同じ）

仮想クラウド・ネットワーク	XXXX/soacsdrASH	YYYY/soacsdrPH（システムは遠隔地にあると想定しているため、プライマリとは異なる）[11]
クライアント・サブネット	XXXX/パブリック・サブネット efXT:US-ASBURN- AD1	YYYY/ パブリック・サブネット efXT:PHX-AD3（システムは遠隔地にある想定であるため、プライマリとは異なる）
ホスト名のプリフィックス	XXXX/soacsdrASH	YYYY/soacsdrPH（プライマリと異なっていても可）
データベース管理者のパスワード	XXXX/Acme1234#	XXXX/Acme1234#（プライマリと同じ）

3.2 オプション2）Data Guardの手動構成

注：3.1で説明しているようにData GuardがOCIコンソールを使用して有効化されている場合は、こちらの手順をスキップし、セクション4「セカンダリSOACSシステムをプロビジョニングする」から続行できます。

プライマリとスタンバイで異なるテナンシーが使用されている場合、またはOCIコンソールで提供される自動化Data GuardオプションがDR構成に関係するDBフレーバーやロケーションに関して有効化されていない場合は、Data Guardを手動で構成する必要があります。

この場合、セカンダリDBシステムは通常DBシステムとしてプロビジョニングされる必要があり、その後、このドキュメントで示すいくつかのセットアップ・スクリプトを実行することによってスタンバイとして構成されます。「付録A - Data Guardの手動セットアップ」で説明されている手順に従い、セカンダリDBシステムをプロビジョニングし、Oracle Data Guardを手動で構成してください。

4. セカンダリSOACSシステムをプロビジョニングする

通常の手順に沿ってセカンダリSOACSシステムをプロビジョニングします。ただし、前のセクションで説明したように、プライマリと同じインスタンス名を使用する必要があるため、同じテナンシーには作成できません（プライマリ・システムもゼロから作成し、使用するインスタンス名を自由に決められる場合は除きます。その場合は、付録Bを参照してください）。以下は、このセットアップを行ううえでのおもな考慮事項です。

- プライマリ・システムに使用されているデータベースのData Guardスタンバイとしてこれから構成するデータベースを参照するSOACSサービス作成します。セカンダリSOACSをプロビジョニングする場合のコンテナとしてスタンバイ・データベースを使用するには、このフィジカル・スタンバイをスナップショット・スタンバイに変換する必要があります。
- Oracle SOACSでは、SOAスキーマをプロビジョニングするときに、それぞれのクラウド・サービスまたはクラウド・インスタンスに固有の接頭辞を使用します。つまり、プロビジョニングの初期状態では、セカンダリ・ロケーションにあるSOACSサーバーから参照されるデータベースは同じなのに、使用されるスキーマは異なるということです。これでは、セカンダリ・ロケーションにある初期状態のSOACSドメインからコンポジットやフローを実行できなくなるため、すでに稼働しているシステムにとっては重大な問題です。使用可能なデータベースを参照するアクティブなSOAサーバーは、どの時点でも1つのサイトにのみ存在する必要があります。そうしないと、メッセージやコールバックの重複が発生し、SOAシステムの一貫性が損なわれることになりかねません。

11 構成には、プライマリとスタンバイのリモートVCNの間のリモート・ピアリングが必要です。

重要：セカンダリ・ロケーションのJDBC文字列を更新し、本番と同じスキーマが参照されるようにすると、スナップショットへの変換を行った場合は、本番サーバーから参照されていたデータと同じデータがセカンダリ・ロケーションにあるSOAサーバーから参照されるようになります。SOAのフロー、コールバックなどに保留中のものがあれば、セカンダリ・ロケーションにあるサーバーは処理を完了させようとします。そのため、スタンバイ・データベースをスナップショットに変換する前にプライマリ・サイトでインスタンスをドレインして完了させ、重複が発生しないようにしておくことが重要です。別のやり方としては、プライマリ・ロケーションのSOAサーバーを停止して、データベースをセカンダリ・ロケーションにスイッチオーバーする方法もあります（停止時間が長くなります）。

以下の手順に沿ってセカンダリSOACSシステムをプロビジョニングします。

A. 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・データベースに変換します。プライマリのデータベース・ノードで、oracleユーザーとして次のコマンドを実行します（*your_sys_password*と*secondary_db_unqname*は、それぞれのケースに合わせて適切な値に置き換えてください）。

[oracle@soacsdrDBa ~]$ dgmgrl sys/*your_sys_password*@*primary_db_unqname* DGMGRL> CONVERT DATABASE *secondary_db_unqname* to SNAPSHOT STANDBY;
Converting database "*secondary_db_unqname* " to a Snapshot Standby database, please wait...
Database "*secondary_db_unqname*" converted successfully

B. SOACSプロビジョニング・ウィザードを使用してセカンダリSOACSインスタンスをプロビジョニングします。

Oracle SOACSのドキュメントの手順に従ってセカンダリ・サイトのSOACSシステムを作成します。このとき、参照先は前の手順でスナップショットに変換したセカンダリDBシステムにします。SOACSサービスに使用する名前は、プライマリ・ロケーションで使用しているものとまったく同じにします。以下の表に、プロビジョニング・ウィザードで設定するオプションをまとめます。

クラウド・アカウント構成プロパティ	既存のプライマリSOACSシステム/例	作成されるセカンダリSOACSシステム/例
OracleCloudテナンシー	XXXX/paasmaa	YYYY/paasmaa2（プライマリとは異なる名前にする必要がある）
OracleCloudアカウント・ユーザー	XXXX/joe@acme.com	YYYY/joe@acme.com （ユーザー名は異なっていても構いません。テナンシーは異なるものにする必要があります）
OracleCloudアカウント・パスワード	XXXX/Acme1234#	XXXX/ Acme1234#（プライマリと異なっていても可）

SOACSの構成	既存のプライマリSOACSシステム/例	作成されるセカンダリSOACSシステム/例
SOACSサービス名	XXXX/soacsdroci	XXXX/soacsdroci（プライマリと同じ）
リージョン	XXXX/us-ashburn-1	YYYY/us-phoenix-1（システムは遠隔地にある想定であるため、プライマリ・サイトとは異なる）
可用性ドメイン	XXXX/efXT-ASHBURN-AD-1	YYYY/ efXT:PHX-AD-3（プライマリとは異なるもの、かつセカンダリDBシステムと同じものにする必要がある）

サブネット	XXXX/パブリック・サブネットefXT:US-ASBURN-AD1	YYYY/パブリック・サブネットefXT:PHX-AD3（システムは遠隔地にある想定であるため、プライマリとは異なる）
SSH公開鍵	XXXX	YYYY（プライマリと異なっていても可）
ライセンス・タイプ	LI、BYOL/BYOL	LI、BYOL/BYOL（プライマリと異なっていても可）
ソフトウェア・リリース	XXXX/12.2.1.4	XXXX/12.2.1.4（プライマリと同じ）
サービス・タイプ	SB & B2Bクラスタを使用するSOAまたはSB & B2Bクラスタを使用するMFT/SOA	SB & B2Bクラスタを使用するSOA（プライマリと同じ）
コンピュート・シェイプ	XXXX/VM.Standard2.1	XXXX VM.Standard2.1（プライマリと同じ）
クラスタ・サイズ	N/2	N/2（プライマリと同じ）
WebLogicユーザー名	XXXX/weblogic	XXXX/weblogic（プライマリと同じ）
WebLogicパスワード	XXXX/Acme1234#	XXXX/Acme1234#（プライマリと同じ）
データベース・タイプ	Oracle Cloud Infrastructure	Oracle Cloud Infrastructure
データベース・コンパートメント名	XXXX/soacsdr	XXXX/soacsdrB（プライマリと異なっていても可）
データベース名	XXXX/ORCL	XXXX/ORCL（プライマリと同じ）
データベースPDB名	XXXX/PDB1	XXXX/PDB1（プライマリと同じ）
データベース管理者ユーザー名	XXXX/SYS	XXXX/SYS（プライマリと同じ）
データベース管理者のパスワード	XXXX/Acme1234#	XXXX/Acme1234#（プライマリと同じ）
ロードバランサのプロビジョニング	Yes	Yes
ロードバランサのポリシー	Least Connection Count	Least Connection Count
ロードバランサのコンピュート・シェイプ	XXXX/VM.Standard2.1	XXXX/VM.Standard2.1（プライマリと同じ）
バックアップとリカバリの構成	XXXX	YYYY（プライマリとは異なる）

オラクルでは、フェイルオーバーやスイッチオーバーの動作を最適化するために、プライマリとスタンバイの両方のロケーションで使用する容量とコンピュート構成をまったく同じにすることを推奨します（そうしないと、OTD/OCILBRでリクエスト調整が必要となり、セカンダリ・ロケーションでSOACSノードのサイジングを実行することが必要になります）。プロビジョニング・プロセスが完了したら、SOAサーバーが正常に動作するか検証できます。

5. セカンダリ・サイトのフロントエンドOTD/OCILBRの仮想ホスト名を割り当てる

セカンダリ・サイトのOTD/OCILBRインスタンスのパブリックIPを使用して、前述した「*フロントエンドOTD/OCILBRに仮想ホスト名を割り当て、プライマリ・サイトのフロントエンド・アドレスを更新する*」の項と同じ手順従い、セカンダリSOAホストの仮想ホストをマッピングします。SOACSクラスタのフロントエンド・アドレスはプライマリのWebLogicドメイン構成からコピーされるため、これを更新する必要はありません。必要なのは、/etc/hostsまたはDNSのエントリをスタンバイで更新することだけです。

注：両方のサイトでホスト名のエイリアスが同じになりますが、必要な信頼ストアや証明書はスタンバイ・ロケーションで作成し直す必要があり、Oracle Cloudのサーバーからフロントエンドのロードバランサに対してSSL呼出しを実行する場合は、スタンバイのOTD/OCILBRの証明書を適切な信頼ストアにインポートする必要があります。必要な手順については、『Oracle SOA Suiteエンタープライズ・デプロイメント・ガイド』を参照してください。

6. ディザスタ・リカバリ・セットアップ（DRS）ツールをダウンロードして実行する

ディザスタ・リカバリ・セットアップ（DRS）ツールは、SOACSディザスタ・リカバリ・セットアップの構成手順を編成し、実行するフレームワークです。このツールは、SOACSにssh接続されているあらゆるホスト（OEL 7のオペレーティング・システムを搭載）、およびDRセットアップが済んでいるDBホストから実行できます。

現行のDRSツールでは、次の通信を許可する必要があります。

- セカンダリ中間層ホストからプライマリDBのプライベートIP（および、RACデータベースを使用する場合はプライマリ・スキャンIP）ポート1521。
  リモート・ピアリングと動的ルーティング・ゲートウェイを介し、プライマリとセカンダリのデータベースがプライベート・ネットワークを使用して接続する場合に必要です。これは、DRのセットアップ時と、構成レプリケーションのようなライフサイクル操作の場合にも使用されます。これは、推奨アプローチです。
- セカンダリ中間層ホストからプライマリDBのパブリックIPポート1521。
  プライマリとセカンダリのデータベースがそれぞれのパブリックIPを介して接続する場合に必要とされます（サイト間でリモート・ピアリングが使用されないため）。これは、DRのセットアップ時と、構成レプリケーションのようなライフサイクル操作の場合にも使用されます。これは一般には推奨されるアプローチではなく、Oracle RAC DGに適していません。

ネットワーク・シナリオに応じて、DRSを実行する前に、oracleユーザーとしてすべてのセカンダリ中間層ホストでクイック・チェックを実行して、プライベート/パブリック・プライマリ・データベースIPへの接続を検証することができます。

java -classpath /u01/app/oracle/middleware/wlserver/server/lib/weblogic.jar utils.dbping ORACLE_THIN system <system_password> <primary_ip_to_check>:1521/<primary_db_service>

DRセットアップのネットワーク要件について詳しくは、「付録E - DRセットアップのネットワーク要件のサマリー」の表を参照してください。

DRSツールの実行手順：

- DRSツールを実行するホストを選択します。このホストには、DRに関係するすべてのホスト（中間層ホストとDBホスト）にssh接続する機能が必要です。ホストへのssh接続に使用されるIPは、その他の入力パラメータと一緒に、DRSの実行前にyamlプロパティ・ファイルによってDRSに指定されます。中間層とDBホストでパブリックIPネットワークが使用される場合は、それぞれのパブリックIPでSSH接続が可能なため、それらのパブリックIPでDRSプロパティ・ファイルを構成します。DRSは、SSHを介してそれらのパブリックIPに接続可能な任意のホストで実行できます。プライベート・ネットワークを使用する場合は、中間層とDBホストのプライベートIPアドレスでDRSプロパティ・ファイルを構成する必要があります。この場合は、プライベート・ネットワークのサイト間接続がご使用のデータベースと中間層コンピューティング・インスタンスに合わせて構成されることが想定され、DRSが実行されているインスタンスを同じネットワーク・インフラストラクチャに併置して、すべてのホストにプライベート接続で到達できるようにする必要があります。

  ヒント：ツールを実行するクラウド・テナンシーでコンピュート・インスタンス（OEL 7）を作成します。このコンピュート・インスタンスは、DR構成が終わってDRSが不要になったら、後で削除できます。代わりに、SOAノードの1つからDRSを実行できます。

- SOACSのDRSツールをこちらからダウンロードし、ツールの実行先ホストにアップロードします。
- コマンド'tar -xzf drs.tar.gz'を使ってこのファイルの内容を展開し、作成された'drs_psm_soa'ディレクトリに移動します。
- README.mdを開いて手順に従います。DRSのREADME.mdファイルで詳細な手順と推奨事項を必ず確認してください。ツールの実行に必要なconfigファイルを確認し、ツールをセットアップで適切に動作させるための要件を満たすことが重要です。

DRSツールは、セカンダリSOACSをスタンバイSOACS DRサイトとして構成するために必要な手順を自動的に実行します。手順は次のとおりです。

- 環境でDRセットアップの準備ができていることを確認するための初期チェック。
- プライマリとセカンダリのSOAサーバーで、/etc/hostsファイルに必要なホスト・エイリアス構成を追加。セカンダリ中間層ホストの名前は、エイリアスとしてプライマリ中間層の/etc/hostsに追加され、一方でまた、プライマリ中間層ホストの名前がセカンダリ中間層の/etc/hostsファイルにエイリアスとして追加されます。

- SOA DBFSマウント用のDBFSウォレット再作成、および中間層ホストのtnsnames.oraファイルへの必要なエイリアスの追加（リモートおよびローカル・データベースのエイリアスは、その後のドメイン構成レプリケーションで使用されます）。dbfsマウントは、DRセットアップ・プロセス中に、プライマリ管理ノードとすべてのセカンダリ・ノードに再マウントされます。
- セカンダリ・ドメイン構成のバックアップは、DRのセットアップ中に変更される前にDRSによって実行されます（つまり、/u01/data/domains/soacsdrs_domain_backup_<timestamp>）。
- プライマリ・ドメイン構成をセカンダリ・サイトにコピー。プライマリのドメイン構成をdbfsマウントにコピーした後、dbfsマウントからセカンダリ・ホストのドメイン・フォルダにコピーします。
- プライマリ・ドメインからセカンダリ・ドメインにドメイン構成をコピーした後、セカンダリ構成ファイルでデータベース接続文字列を適切に置換。プライマリ・データベースの接続文字列はセカンダリ・ドメイン構成でのセカンダリ・データベースの接続文字列と置き換えられます。プライマリ・ドメインとセカンダリ・ドメインの間ではDB接続文字列のみが異なります。これは、いったんDRが構成されると、セカンダリ・ドメインはプライマリと同じスキーマ名を参照するためです。
- セカンダリ・ドメインがDR用に正しく構成されていることを確認。スナップショット・モードのデータベースを使用して、DRの構成後にセカンダリ管理対象サーバーをローリング方式で起動し、セカンダリ・フロントエンドのsoa-infra URLをチェックします。

このプロセスの間、ツールによって、セカンダリ・データベースでいくつかのデータベース・ロールの変換が実行されます（スナップショット・スタンバイに変換し、フィジカル・スタンバイに再び変換）。実行中、フレームワークにより、"logfile_<date-time-stamp>.log"という名前のログ・ファイルに記録されます。セットアップ・プロセスは、このファイルの内容とプロセスの出力を監視することによって監視できます。終了すると、セカンダリ・データベースはフィジカル・スタンバイ・ロールのままになり、セカンダリ管理サーバーとセカンダリ管理対象サーバーは停止したままになります。

重要：この時点まで、セカンダリ・サイトのSOAサーバーから参照されていたのは、デプロイされているコンポジットがなく、実行が保留になっているポリシーやフローもない"空の"SOAINFRAスキーマです。上記の手順に沿ってセカンダリ・ロケーションのJDBC文字列を更新し、本番と同じスキーマが参照されるようにしたら、セカンダリ・ロケーションにあるSOAサーバーから参照されるデータは本番のサーバーから参照されていたものと同じになります。実行が保留されていたフローやコールバックなどがある場合は、セカンダリ・ロケーションにあるサーバーを起動すると、処理を完了させるための実行がこの時点で再開されます。そのため、すでに指摘したように、スタンバイ・データベースをスナップショットに変換する前にプライマリ・サイトでインスタンスの処理をすべて完了させておくことが重要です。

7. **任意：スイッチオーバー・プロセス全体を検証する**

システムはこの時点でスイッチオーバーできる状態になっている必要があります。データベースとサーバーが正しく動作することを確認するために、スイッチオーバー全体の通しテストを必ず実行することをお薦めします（プライマリがすでに"本番"稼働している場合は、メンテナンス用の時間枠をスケジュールする必要があります）。次の手順に沿って、最初のスイッチオーバー・テストを実行します。

1. プライマリSOAプロセスを停止する

プライマリWebLogic管理コンソールに接続し、プライマリ・ロケーションでサーバーを停止します。

2. 必要なDNS変更を実行し、コンシューマが新しいプライマリOTDを参照するようにする

システムで使用される名前をホスティングしているDNSサーバーで必要なDNSプッシュを実行するか、システムのフロントエンド・アドレスがsite2のOTD/OCILBRで使用されるパブリックIPを参照するようにクライアントでのファイル・ホスト解決を変更します。

3. データベースをスイッチオーバーする

プライマリのデータベース・ホストで*oracle*ユーザーとして次の手順を実行します。

[oracle@soacsdrDBa ~]$ dgmgrl sys/*your_sys_password*@*primary_db_unqname*

DGMGRL> switchover to *secondary_db_unqname*
Performing switchover NOW, please wait...
Operation requires a connection to instance "ORCL" on database "*primary_db_unqname*"ORCL
Connecting to instance "ORCL"...
Connected as SYSDBA.
New primary database "*secondary_db_unqname*" is opening...
Oracle Clusterware is restarting database "orcl" ...
Switchover succeeded, new primary is "*secondary_db_unqname* "

4. セカンダリ・サイトのSOACSを検証する

この時点では、両方のサイトのSOAサーバーから同じスキーマが参照され、セカンダリ・ロケーションのデータベースがオープンされています。サーバーを起動し、新しいアクティブ・サイトのコンポジット・デプロイメントとコンポジット・インスタンス（存在する場合）を検証します。

a. セカンダリ・サイトで管理サーバーと管理対象サーバーを起動します。

a. セカンダリのSOACSインスタンスのEnterprise Manager FMW Controlコンソールにログオンし、クラスタに含まれる両方のサーバーのsoa-infraシステムを検証します。エンドポイント・テストを実行して、システムが正しく動作することの確認をお薦めします。

5. 元のサイトにスイッチバックする

システムを元の状態に戻す場合は、データベースのスイッチバックを実行し、プライマリ・ロケーションにあるSOAサーバーを再起動します。

SOACS/MFTCS DRライフサイクル手順

スイッチオーバー

計画的な運用として管理者が2つのサイトのロールを元に戻す操作をスイッチオーバーといいます。ロールはプライマリからスタンバイに、またスタンバイからプライマリに変更されます。スイッチオーバーに必要なタスクをOracle Site Guardを使用して自動化し、スイッチオーバー時のRTOを短縮することができます（詳しくは、付録Dを参照）。SOACS/MFTCS DR構成で手動スイッチオーバーを実行する場合は、次の手順を実行します。

a. SOACSプライマリ・サイトの管理対象サーバーを停止します。

SOACSのドキュメントを使用して、プライマリ・サイトの管理サーバーと管理対象サーバーを停止します。

b. 最近、プライマリSOA/WebLogicドメインに構成変更を適用した場合は、変更を必ずスタンバイ・ロケーションに伝播してください（下の「ドメイン構成の変更をシステムに適用する」の項を参照してください）。

c. 新しいプライマリOTDがコンシューマから参照されるようにするために、必要なDNS変更を行います。

システムで使用される名前をホスティングしているDNSサーバーで必要なDNSプッシュを実行するか、site2のOTD/LBaaS/OCILBRで使用されるパブリックIPがシステムのフロントエンド・アドレスから参照されるようにファイルのホスト解決を変更します。

d. Data Guard Brokerを使用してデータベースのスイッチオーバーを実行します。

プライマリ・サイトのDBシステム・ノードからoracleユーザーとして以下を実行します。

[oracle@soacsdrDBa ~]$ dgmgrl sys/*your_sys_password*@*primary_db_unqname*

DGMGRL> switchover to *secondary_db_unqname*

Performing switchover NOW, please wait...

Operation requires a connection to instance "ORCL" on database "*secondary_db_unqname*"
Connecting to instance "ORCL"...
Connected as SYSDBA.

New primary database "*secondary_db_unqname*" is opening...

Oracle Clusterware is restarting database "*primary_db_unqname* " ...
Switchover succeeded, new primary is "*secondary_db_unqname*"

e. 新しいSOACSプライマリ・サイトの管理対象サーバーを起動します。

Oracle SOACSのドキュメントを使用して、セカンダリ（現在の新しいプライマリ）サイトの管理対象サーバーを起動します（管理サーバーとノード・マネージャはスタンバイで起動したままにしておいて構いません）。

フェイルオーバー

プライマリ・サイトが使用できなくなり、管理者がデータベースをフェイルオーバーし、セカンダリ・サイトで管理対象サーバーを起動する操作がフェイルオーバーです。元のプライマリ・データベースで障害が発生し、すみやかにプライマリ・データベースをリカバリできない場合は、スタンバイ・データベースをプライマリ・データベースにロール移行することができます。これは手動フェイルオーバーと呼ばれます。データが失われるか失われないかは、プライマリ・データベースで障害が発生したときにプライマリ・データベースとターゲットのスタンバイ・データベースにトランザクション一貫性があったかどうかによって異なります。フェイルオーバーに必要なタスクをOracle Site Guardを使用して自動化し、フェイルオーバー時のRTOを短縮することができます（詳しくは、付録Dを参照）。SOACS/MFTCS DR構成で手動フェイルオーバーを実行する場合は、次の手順を実行します。

a. 新しいプライマリOTDがコンシューマから参照されるようにするために、必要なDNS変更を行います。

システムで使用される名前をホスティングしているDNSサーバーで必要なDNSプッシュを実行するか、システムのフロントエンド・アドレスがsite2のOTD/OCILBRで使用されるパブリックIPを参照するようにクライアントでのファイル・ホスト解決を変更します。

b. Data Guard Brokerを使用してデータベースのフェイルオーバーを実行します。セカンダリのDBシステム・ノードでData Guard Brokerを起動します。oracleユーザーとして次の手順を実行します。

[oracle@soacsdrDBb ~]$ dgmgrl sys/your_sys_password@primary_db_unqname
DGMGRL> failover to *secondary_db_unqname*
Performing failover NOW, please wait...
Failover succeeded, new primary is "*secondary_db_unqname*"

c. 新しいSOACSプライマリ・サイトの管理対象サーバーを起動します。

Oracle SOACSのドキュメントを使用して、セカンダリ（現在の新しいプライマリ）サイトの管理対象サーバーを起動します。

ドメイン構成の変更をシステムに適用する

クラウド・データセンターの仮想マシン間でファイル・システムを直接同期することはできないため、WLSドメインの構成変更は自動的にセカンダリ・サイトに伝播されません（標準的なオンプレミスのアクティブ-パッシブDRデプロイメントでは自動的に伝播されます）。両方のロケーションで同じ構成（earデプロイメント、WLSドメイン構成、デプロイメント・プランなど）を維持するための手段として使用できるおもな方法は次の2つです。それぞれを適用できるかどうかは、この"ファイル・システムの中身"の構成が変更される頻度によって決まります。

a) ドメイン構成が頻繁に変更されないOracle SOACSの場合（なお、コンポジット・デプロイメント、ドメインとWSMのポリシー、MDSはデータベースに格納されるため、これらの更新はドメイン構成の変更には該当しません）は、構成変更を手動で2回（本番で1回、スタンバイで1回）適用することを推奨します。

b) 定期的にファイル・システムの構成が変更されるOracle SOACSの場合は、Data Guardで構成を同期するときにOracle Database File System（Oracle DBFS）を使用できます（Oracle DBFSは、データベースに格納されているデータのファイル・システム・ビューを提供します）。構成のレプリケートにOracle DBFSを使用すると、セットアップ、ディスク領域の割当て、ライフサイクルの観点で影響があるため、構成変更を頻繁にレプリケートすることが必要な場合にOracle DBFSを使用することを推奨します。Oracle DBFS以外にも、サイト間で直接*rsync*を行うなどの代替手段がありますが、その場合は、コピー時にトランザクションを制御できない、コピー操作中に障害が発生した場合にドメイン構造が壊れる可能性がある、といった別のリスクが現出します。

2つの方法については、以下に説明します。

a) 両方のサイトでの構成変更の繰返し

ファイル・システムの構成が同期された状態に維持するには、以下の手順に従って両方のサイトで構成変更を繰り返します。

A. プライマリ・サイトで通常どおり構成変更を適用する

プライマリ・ロケーションのWLS管理コンソールを使用して構成変更を適用します。変更を有効化し、必要に応じて必要なSOACSサーバーを再起動して、想定どおり変更が機能していることを確認します。

B. 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 プライマリのデータベース・ホストで*oracle*ユーザーとして次の手順を実行します。

[oracle@soacsdrDBa ~]$dgmgrl sys/*your_sys_password@primary_db_unqname*
DGMGRL> CONVERT DATABASE *secondary_db_unqname* to SNAPSHOT STANDBY;

Converting database " *secondary_db_unqname*" to a Snapshot Standby database, please wait...
Database " *secondary_db_unqname*" converted successfully

C. セカンダリ・サイトの管理サーバーを起動する（起動していなかった場合）

Oracle Cloudのドキュメントの手順に沿って管理サーバーを起動します。セカンダリ・ロケーションでは、管理サーバーのみが起動し、管理対象サーバーは起動していないことが重要です[12]。

D. セカンダリ・サイトで構成変更を繰り返す

プライマリ・ロケーションのWLS管理コンソールを使用して構成変更を適用します。変更を有効化し、想定どおり変更が機能していることを確認します。この変更では、データベースの構成は一切変更せず、WLSドメイン構成またはアプリケーション・デプロイメントのみを変更する必要があります。データベースの変更は、次の手順でデータベースをフィジカル・スタンバイに戻したときにプライマリによって上書きされます。

E. データベースをフィジカル・スタンバイに戻す。プライマリのデータベース・ホストでoracleユーザーとして次の手順を実行します。

[oracle@soacsdrDBa ~]$dgmgrl sys/*your_sys_password@primary_db_unqname*
DGMGRL> CONVERT DATABASE *secondary_db_unqname* to PHYSICAL STANDBY;
Converting database " *secondary_db_unqname*" to a Physical Standby database, please wait...
Oracle Clusterware is restarting database "orclb" ...
Continuing to convert database " *secondary_db_unqname*" ...
Database " *secondary_db_unqname*" converted successfully

b) Oracle DBFSを使用した構成の伝播

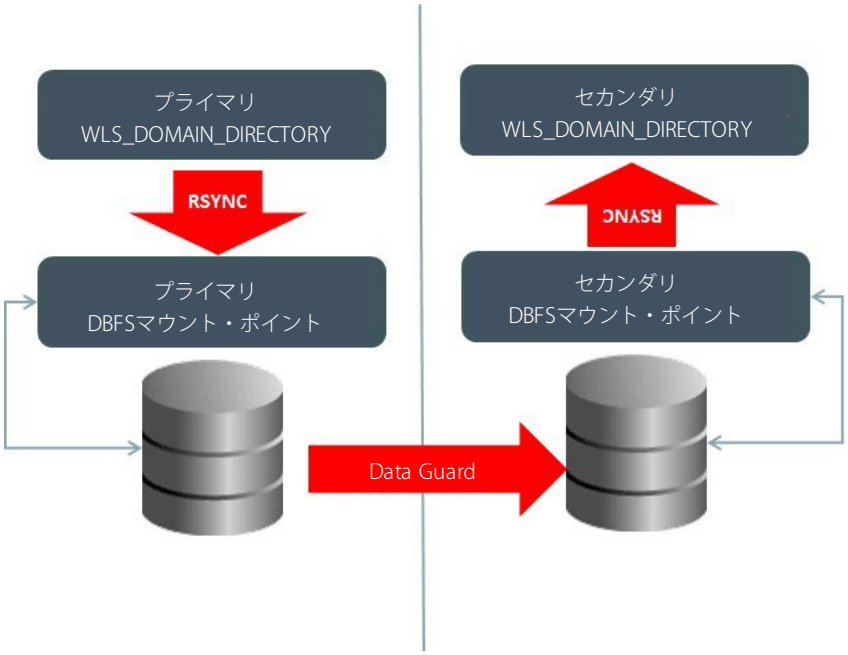
ファイルの格納やリレーショナル・データの管理にデータベースの機能を使用して、どのオペレーティング・システムからでもアクセスできる標準のファイル・システムとして公開するのが、Database File System（DBFS）です。Oracle Database File System（Oracle DBFS）は、データベース表に格納されているファイルおよびディレクトリの上位に標準のファイル・システム・インタフェースを作成します。Oracle DBFSは、ローカル・ファイル・システムのように見える共有ネットワーク・ファイル・システムを提供する点でNFSに似ています。NFSと同様に、Oracle DBFSを実行するにはサーバー・コンポーネントとクライアント・コンポーネントの両方が必要です。

Oracle Cloudデータセンター間のファイル・レプリケーションには制限があるため、Oracle DBFSをOracle Data Guardと併用してプライマリ・サイトからセカンダリ・サイトにファイルをコピーすることができます。システムのライフサイクル中にドメイン・ファイル・システムの更新が頻繁にある場合は、Oracle DBFSを使用して定期的にWLSドメイン構成のサイト間レプリケーションを実行できます。ただし、SOACSドメインの構成をDBFSマウント上に直接格納することはできません。中間層の起動がDBFSインフラストラクチャに依存することになるためです（データベースだけでなく、FUSEライブラリやマウント・ポイントなども依存することになります）。また、SOACS/MFTCS DRソリューションの各サイトにはクラウド識別ドメインへの参照とJDBC接続文字列のローカルDBサービスへの参照が含まれるため、このホワイト・ペーパーの設計ではSOACSのWebLogicドメイン構成を"そのまま"コピーできないことにも注意してください。各サイトに構成をコピーした後で、構成を変更する必要があります。この方法の場合、プライマリ・サイトのドメイン構成のコピーがあるディレクトリは、スタンバイ・ロケーションのData Guardで最新の状態に維持されます。このディレクトリは、DBが読取り専用モードでオープンになっているか（Active Data Guardを使用している場合）、スナップショット・スタンバイへの変換が実行されるか、スイッチオーバーまたはフェイルオーバーが実行されない限り、スタンバイ・サイトでは使用できません。スナップショットへの変換を行うと、DBFSファイル・システムを書込み用としてもマウントできるようになりますが、セカンダリ・ロケーションのSOAサーバーが（偶発的に、または再起動により）起動されなくなるわけではないため、重複が発生し、プライマリ・ロケーションですでに処理された作業がもう一度実行される可能性がある点に注意が必要です。

12 SOAとOracle Secure Backupの構成アーティファクトの数を減らす変更の場合は、サーバーが稼働していなければ変更を適用できないこともあります。その場合は、管理対象サーバーの起動が必要になります。該当するアーティファクトは、それぞれの製品ドキュメントで確認してください。こうしたケースで、データベースに保留中のメッセージがある場合は、スタンバイ・ロケーションでメッセージが再実行されることもあります。そのようなシナリオでは、SOA WLSサーバーを起動する前に、スナップショット・データベースのSOAデータベース・スキーマをSOAの標準の手順に沿ってドレインするか切り捨てることを推奨します。

とはいえ、セカンダリ・ロケーションのサーバー全体の起動とテストを定期的に実行し、構成とデプロイメントが正しくレプリケートされること、およびクラウド識別ドメインとJDBC接続文字列が正しく置き換えられることを確認することを推奨します。これは、スイッチオーバーを実行するか、セカンダリ・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ことで実行できます。この最後のケースでは、処理が重複して実行されないようにするために、まずプライマリ・データベースから保留中のSOA作業（これには、保留中の非同期呼出しとJMSメッセージが含まれます）を"ドレイン"することを推奨します。ドレインするということは、新たに着信するリクエストがブロックされ、保留中のメッセージがすべて処理されるか完了するまで待機することになる場合があります。次のダイアグラムは、WebLogicドメインの構成変更をプライマリ・ロケーションからセカンダリ・ロケーションにレプリケートするときに使用されるモデルを示しています。

アプリケーション・デプロイメント操作の場合は、WebLogic管理コンソールでWebLogicのデプロイメント・オプション



として"Upload your files"を使用し、管理サーバーのアップロード・ディレクトリの下（*ドメイン・ディレクトリ/servers/admin_server_name/upload*の下）に、デプロイしたファイルを配置することを推奨します。このようにすると、DBFSコピー・スクリプトによってこれらのファイルがスタンバイに同期されるようになります。

要するに、Oracle DBFSを使用してドメイン構成を同期する場合は次の手順が必要です。

1. SOACSインスタンスにOracle DBFS構成がすでに作成されていて、DR用の適切なマウント・ポイントがあることを確認する
2. ドメインをコピーするスクリプトと構成を置き換えるスクリプトを作成する
3. ドメインのコピーを有効化する（方法は、手動で実行するか、cronジョブを構成するかのいずれかです）[13]

以下のサブセクションで、各手順について詳しく説明します。

13 SOACSクラスタ内の別のノードにあるファイルのうちdomain/configディレクトリの下に存在しないファイルに適用した変更は、該当するノードに手動で同期する必要があります。このホワイト・ペーパーで説明したOracle DBFSを使用する方法では、管理サーバー・ノードにのみドメイン構成がレプリケートされます。

SOACSインスタンスにすでに作成されているOracle DBFS構成を使用し、DR用の適切なマウント・ポイントがあることを確認する

SOACSシステムにはDBFSがすでに構成されているため、特に設定は必要ありません。DBFSマウント・ポイントにアクセスできるように事前構成された各SOACSコンピュータで、dbfs_clientと呼ばれるクライアント・コマンドライン・インタフェースを実行します。dbfs_clientを使用すると、ネットワーク上の任意のホストとデータベースとの間でファイルを相互にコピーすることができます。DBFSはOracle Databaseの一部としてインストールされるもので、マウント・ポイントにアクセスするノードには*fuse*（FileSystem in Userspace）が必要です。これにより、リスト表示やコピーのような単純なファイル・システム・コマンドが、シェル・ユーティリティと同様の方法で実装されます。SOACS 12.2.1.2以降のVMにはdbfs_clientとfuseがすでにインストールされています。SOACS VMにデフォルトで作成されるDBFSマウント・ポイントとクライアント構成は、MFT、B2Bおよびファイル・アダプタの場合に（共通のマウント・ポイントをSOAクラスタのメンバー間で共有するために）使用されます。パフォーマンスと管理の独立性を目的として、サイト間のドメイン・レプリケーションに使用するdbfsマウント・ポイントは、*/u01/soacs*配下のデフォルトのDBFSマウント・ポイントとは別に構成してマウントします。こうすると、システムの実行時動作が構成操作に妨げられることが確実になくなります（たとえば、このようにマウント・ポイントを別々にしておくと、クラスタ内でのファイル・アダプタによるファイル処理に影響を及ぼすことなく、DBFS構成ディレクトリをアンマウントできます）。"構成"用のDBFSマウント・ポイントは（下の手順で構成する）*/u01/soacs/dbfs*の下に、"実行時"用のDBFSマウント・ポイントは*/u01/soacs/dbfs_directio*の下に作成されます。両方ともSOACSインスタンスによって自動的に作成/構成されます。

DBFS内のファイル・システムを確認する

プライマリとスタンバイの中間層ノードのそれぞれにマウント・ポイントと/u01/soacs/dbfs/share/ディレクトリが存在することを*oracle*ユーザーで確認します。

```
[oracle@soacsdroci-wls-1~]$ mount | grep dbfs
dbfs-@ORCLB:/ on /u01/soacs/dbfs type fuse
  (rw,nosuid,nodev,max_read=1048576,default_permissions,user=oracle)
dbfs-@ORCLB:/ on /u01/soacs/dbfs_directio type fuse
  (rw,nosuid,nodev,max_read=1048576,default_permissions,user=oracle)
[oracle@soacsdroci-wls-1~]$ ls -lart /u01/soacs/dbfs/share/
total 0
drwxr-x---.4 oracle oracle 0 Nov 21 18:09 ..
drwxr-x---.2 oracle oracle 0 Nov 23 15:45 .
```

プライマリにテスト・ファイルを作成し、そのファイルをスタンバイで読み取れることを確認します。プライマリの中間層ノードのいずれかで、次のコマンドを実行します。

```
[oracle@soacsdroci-wls-1~]$ echo "test" > /u01/soacs/dbfs/share/test.txt
```

Active Data Guardが有効化されているデータベース・エディション（Extreme Performance Editionが必要です）の場合は、スタンバイ・データベースが自動的に読取りでオープンされるため、プライマリのdbfsマウントに適用された変更はすぐにスタンバイで読み取ることができます。Active Data Guardが有効化されていないデータベース・エディションの場合は、スタンバイのデータベースをスナップショットに変換しなければ、プライマリで作成されたファイルを読み取れるようになりません。Active Data Guardを使用していないシステムの場合は、スタンバイ・データベースをスナップショットに変換します（Active Data Guardを使用している場合は、その作成されたファイルに何の手順も踏まずにスタンバイのファイル・システムからアクセスできるはずです）。

```
[oracle@soacsdrDBa ~]$dgmgrl sys/your_sys_password@primary_db_unqname
DGMGRL> CONVERT DATABASE secondary_db_unqname to SNAPSHOT STANDBY;
Converting database " secondary_db_unqname" to a Snapshot Standby database, please wait...
Database " secondary_db_unqname" converted successfully
```

セカンダリのWebLogic管理ノード上のファイルを確認します。

[oracle@soacsdroci-wls-1~]$ cat /u01/soacs/dbfs/share/test.txt test

DBFSファイル・システムの基本的な伝播の検証が済んだら、プライマリからスタンバイにドメイン・ディレクトリを同期する*dbfscopy.sh*スクリプトのカスタマイズに着手できます。次に、DBFSを使用したレプリケーションを構成する場合の考慮事項の一部を示します。

- 手動スイッチオーバーを実行する前に（または適切な頻度でcronジョブを実行して）デプロイメントをWebLogicドメイン・ディレクトリにプッシュすることが非常に重要です。そうしないと、スタンバイ・ドメイン・ディレクトリを最後に更新してからロールが切り替えられるまでの間に空白期間ができ、データ損失が発生する恐れがあります。データ損失が発生する期間は、プライマリとスタンバイで実行されるcronジョブの頻度で決まります。
- WebLogic管理サーバー・ノードのドメイン・ディレクトリ外で作成されるデータやファイルは、*dbfscopy.sh*スクリプトでは処理されないため、別途同期する必要があります。

dbfscopy.shスクリプトをダウンロードし、カスタマイズして実行する。任意でcronジョブを有効化する

ドメイン構成をコピーして置き換えるこのスクリプトの目的は、WebLogicドメイン構成をプライマリ・ロケーションのDBFSステージ・ディレクトリにコピー（rsyncを使用）することと、ステージ・ディレクトリの内容をセカンダリ/スタンバイ・システムのWebLogicドメイン・ディレクトリにコピー（rsyncを使用）することです。このスクリプトでは、各サイトのデータソースの接続文字列とCloud識別ドメインの名前の"マッピング"（置換）も行われます。スクリプトは両方のサイトで使用されるため、いずれのサイトもプライマリ・ロールを担って他方に構成をレプリケートすることができます。このスクリプトは、制御された方法で手動実行することも、cronジョブを使用して定期的に自動実行することもできます。手動と"cron化"のどちらのアプローチが各ケースに適しているかは、適用される構成変更の頻度と規模によります。推奨される頻度は、各システムのライフサイクル（新しいデプロイメントや構成変更が適用される頻度）によって変わります。構成のコピーが"cron化"されていて、大きなファイルが変更に含まれない場合は、rsyncとData Guardで伝播が迅速に行われ、あらゆるものがわずかな時間差でプライマリ・ロケーションからセカンダリ・ロケーションに送信されます。ただし、（一部のファイルが非常に大きいことや、関係するファイルが多数あることが原因で）rsyncコピーに長時間かかると、構成変更の一部しかプライマリのDBFSステージ・ディレクトリに"適用"されていない時点で、スタンバイのcronによってセカンダリ・ドメインのロケーションへの移動が行われている、という事態が発生する恐れがあります。この時点でプライマリ・ノードがクラッシュすると、スタンバイ・ドメインの構成が無効または一貫性のない状態になる可能性があります。
このような事態は異なる方法で回避できます。

1) --delay-updatesを使用してrsyncコピー操作をより"アトミック"にする（この方法の場合は、更新されたファイルごとに一時ファイルを作成し、送信が終了するまで保持用ディレクトリに配置し、終了したらすべてのファイルを次々に名前変更して所定のディレクトリに配置します）。

2) rsyncを使用して別の中間フォルダにコピーし、そこからtarファイルを作成し、このtarファイルをOracle DBFSの場所にコピーしてからスタンバイに抽出する。いずれの方法の場合も、必要な領域が増加します。

これらの方法を利用できるかどうかは、構成変更の種類とデプロイメントの頻度によります。標準的な変更頻度（1日に数個の新しいデプロイメント）でコンポジット・ファイルとearファイルのサイズが普通（500 KB未満）の場合は、*rsync*を使用するこの方法を使用する必要はありません。ただし、用意されているサンプルの*dbfscopy.sh*スクリプトは、拡張してtarを使用する方法に使用することもできます。

また、このスクリプトでは、ステータス情報を取得するために、各中間層サイトからリモート・データベースにアクセスする必要があります。デフォルトでは、SOACSで使用されるOCI DBシステム・インスタンスはパブリック・ネットワーク内に存在する必要があり、その1521ポートへのアクセスは、適切なインターネット・ゲートウェイを使用して外部から有効化できる必要があります。DBへのアクセスがVPNトンネル経由または動的ルーティング・ゲートウェイ経由で構成されていた場合は、プライマリとスタンバイの中間層から相応のアクセスができるようにすることが必要になります。OCI上のSOACSのネットワーク要件について詳しくは、SOACSとOCIのドキュメントを参照してください。

DBFSコピー・スクリプトは次の手順に沿って使用します。

A. OTNからスクリプトをダウンロードしてプライマリWebLogic管理サーバー・ノードにコピーします。スクリプトはこちらから入手できます。

B. *dbfscopy.sh*スクリプトを開いて手順を読みます。このスクリプトの最新バージョンでは変数のカスタマイズは不要で、sysのパスワードの入力だけが求められます。

C. 最初にプライマリWebLogic管理サーバー・ノードでスクリプトを実行し、その後プライマリでcronを作成し、一定間隔で実行します。実行状況を監視し、エラーがないか確認します。このスクリプトでは、Data Guardのステータスの検証と、プライマリWebLogicドメインからDBFSマウント・ポイントへのドメイン構成のコピーが行われます。

D. 完了したら、*dbfscopy.sh*スクリプトをダウンロードしてセカンダリWebLogic管理サーバー・ノードにコピーします。

E. スクリプトを開いて手順を読みます。このスクリプトの最新バージョンでは変数のカスタマイズは必要ではなく、sysのパスワードの入力だけが求められます。

F. セカンダリの中間層のWebLogic管理サーバー・ノードでスクリプトを実行し、その後スタンバイでcronを作成し、一定間隔で実行します。実行状況を監視し、エラーがないか確認します。このスクリプトでは、Data Guardのステータスの検証と、DBFSマウント・ポイントからセカンダリWebLogicドメインへのドメイン構成のコピーが行われ、スタンバイで必要な構成に必要な置き換えが行われます。

G. セカンダリ・ロケーションの管理サーバーを再起動して、変更を有効にします（セカンダリDBをスナップショット・スタンバイ・モードにするか、Active Data Guardを使用する必要があります）。

重要：domain_home/config配下の構成は、管理対象サーバーを再起動して管理サーバーに接続すると、WebLogicドメインを構成している他のすべてのノードに自動的にコピーされます。domain_home/configディレクトリ以外の場所にあるその他の構成は最初のノードにしかコピーされません。管理対象サーバー・ノードには1つ1つ手動でレプリケートする必要があります。これには、domain_home/bin、domain_home/securityなどの下にある起動スクリプトが含まれます。

H. プライマリとセカンダリでの最初の実行が完了したら、システムのcronリストにスクリプトを追加して、定期的に実行されるようにすることができます。

重要：コピー・スクリプトを"cron化"すると同期が自動化されますが、次のような影響もありますので注意してください。

1) 同期を行うと、両方のロケーションのcronジョブの頻度と同程度の待機時間が追加発生する可能性があります。つまり、cronジョブをそれぞれ30分間隔で実行するように設定すると、プライマリ・ロケーションとセカンダリ・ロケーションで処理時間枠が重複している場合は、変更が有効になるまで60分かかる可能性があります。スイッチオーバーは、前回の構成変更からそれだけの時間が過ぎていることを確認してから実行してください。そうしないと、スタンバイ上に変更があるうちにスイッチオーバーしてしまい、元々適用されていた変更がロールの切替えで上書きされる恐れがあります。

2) cronの頻度は、ドメイン・ディレクトリからDBFSステージ・ディレクトリにデプロイメントまたは構成変更をコピーするのに要する最大時間以上の間隔で設定する必要があります。さもなければ、コピー・ジョブが重複する可能性があります。

その他のライフサイクル操作については、「付録E – その他のライフサイクル操作」を参照

ベスト・プラクティス

ディザスタ・リカバリ・トポロジのライフサイクルにおいて、オラクルでは以下のベスト・プラクティスを推奨しています。

- JDBC永続ストアを使用する。デフォルトでは、SOAサーバーによって使用されるJMS永続ストアは、JDBCストアです。カスタム永続ストアを作成する場合、それらは必ずJDBC永続ストアとしても作成してください。この方法ではJMSメッセージがデータベース表に保存されるため、この情報はData Guard経由でセカンダリ・サイトにレプリケートされます。
- プライマリ・サイトとスタンバイ・サイトで同じパッチ・レベルを維持する。ソフトウェアはいずれの層においてもセカンダリ・システムに自動的にレプリケートされません。パッチをプライマリにインストールする場合は、同じパッチをスタンバイ・ロケーションにインストールする必要があります。データベースにパッチを適用する際は、Data Guardトポロジのパッチの適用方法に関する個別のパッチのドキュメントをチェックしてください。
- プライマリ・サイトとスタンバイ・サイトで同じ構成を維持する。WebLogic構成の一部ではない（つまり、DBFSスクリプトを使用してレプリケートされない）プライマリ・システムに適用される変更はすべて、セカンダリ・システムでも実行する必要があるため、プライマリ・システムとセカンダリ・システムには同じ構成が含まれます。例：プライマリOTD/LBRでの変更、オペレーティング・システムへの変更など。
- 定期的にスイッチオーバーを実行してセカンダリ・システムの健全性を検証する。「付録D 検証のためにセカンダリ・サイトを開く」で説明するように、完全なスイッチオーバーを実行せずに、検証のためのセカンダリ・サイトを開くこともできます。

結論

SOA Cloud Service構成でのディザスタ・リカバリは、本番データベースと、Oracle Data Guardによって同期されるスタンバイ・データベースで構成されます。独立した中間層構成を2つ作成し、それぞれ独自のローカル・データベースを参照することで、データセンター間でのファイル同期の必要性を最小限に抑えます。Oracle Cloudでこのディザスタ・リカバリ・ソリューションを使用すれば、スタンバイのハードウェアとソフトウェア、さらにはリモート・データセンターを所有、管理するコストや手間が不要になり、同時に最大限のリカバリ時間目標とリカバリ・ポイント目標を達成できます。

Oracle Data Guardをディザスタ・リカバリに使用すると、リモート・バックアップをリストアするよりも優れたRTOとRPOが得られます。つまり、本番環境は、Oracle Cloud上ですでに稼働している同期済みの本番データベースのコピーに即座にフェイルオーバーされます。クラウド上のスタンバイ・データベースは、ディザスタ・リカバリで利用できるだけでなく、開発およびテスト用のクローン・データベースをシードする目的でも使用可能です。

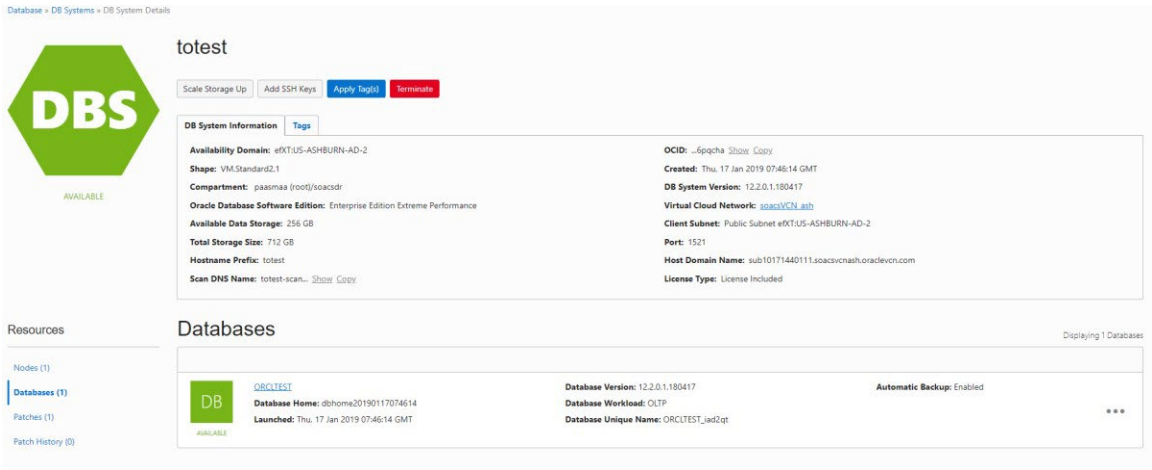
中間層を使用して効率的に構成をレプリケーションすることで保守が容易になり、構成を絶え間なくレプリケーションする方法の場合に発生するオーバーヘッドが減少します。ただし、いつでもリカバリができる状態にしておくために、適切な方法で定期的にスタンバイの検証を行う必要があります。各システムのライフサイクルに応じ、異なる同期方法を使用して動作を最適にすることができます。

付録A - Data Guardの手動セットアップ

プライマリとスタンバイに単一テナンシーを使用できない場合や、OCIで提供される自動Data Guard構成オプションがDR構成に含まれるDB"フレーバー"や場所に対応していない場合は、Data Guardを手動で構成する必要があります。この項で説明する手動プロセスは、プライマリ・サイトにデータベースがすでにプロビジョニングされていることを前提としています。

シングル・インスタンス・データベースの場合

1. セカンダリ・データベースはプライマリと同じ特定のデータベース名、シェイプ、バージョンで作成する必要があります。OCIデータベースに関する必要データを取得するには、使用中のデータベース・システムをOCIコンソールでクリックします。



以下の情報をメモします。

- シェイプ：（例）VM.Standard2.1
- ソフトウェア・エディション：（例）Enterprise Edition Extreme Performance
- 使用可能なデータ・ストレージ：（例）256 GB
- DBシステム・バージョン：（例）12.2.0.1.180417
- データベース名：（例）ORCLTEST
- ポート：1521

プライマリで使用されているPDBの名前も確認します。名前を確認するには、プライマリDBノードで次のSQL問合せを実行します。

```
SQL> SELECT NAME, CON_ID FROM V$CONTAINERS;

NAME                           CON_ID
------------------------------ ----------
CDB$ROOT                            1
```

PDB$SEED	2
PDBTEST	3

2. セカンダリ・データベースをプロビジョニングします。これには、OCIコンソールでリージョンを変更し、スタンバイを配置する予定の適切なロケーションを設定します。「launch DB System」ボタンをクリックしてセカンダリDBを作成します。セカンダリDBインスタンスで使用するデータベースの名前、リリース、パッチ・レベル、PDBの名前は、プライマリの作成時に使用したものと必ず同じにしてください。そのためには、スタンバイを作成する前にプライマリ・システムにパッチを適用することが必要になる場合があります（特に、プライマリ・システムの使用期間が長い場合）。オラクルでは、コンピュート・シェイプとストレージ・サイズもプライマリと同じにすることを推奨しています。スタンバイ・データセンターに必要なデータベースを、SOACSのドキュメントの手順に沿ってプロビジョニングします。スタンバイDBシステムの作成プロセスで使用する必要があるプロパティの例と要件を次の表に示します。

   次の表に、例を使用して、セカンダリ・システムのプロビジョニング画面で使用する必要があるパラメータを示します。

クラウド・アカウント構成プロパティ	既存のプライマリDBシステム/例	作成されるセカンダリDBシステム/例
Oracle Cloudテナンシー	XXXX/paasmaa	YYYY/paasmaa2（プライマリと異なっていても可。付録Bを参照）
Oracle Cloudアカウント・ユーザー	XXXX/joe@acme.com	YYYY/joe@acme.com（プライマリと同じでも異なっていても可。付録Bを参照）
Oracle Cloudアカウント・パスワード	XXXX/Acme1234#	XXXX/Acme1234#（プライマリと同じでも異なっていても可。付録Bを参照）

DBシステム構成	既存のプライマリDBシステム/例	作成されるセカンダリDBシステム/例
表示名	XXXX/soacsdrDBa	YYYY/soacsdrDBb（プライマリと異なっていても可）
可用性ドメイン	XXXX/efEXT:US-ASBURN-AD1	YYYY/efXT:PHX-AD-3（システムは遠隔地にある想定であるため、プライマリとは異なりますが、セカンダリSOACSサービスが作成されるサイトと同じである必要があります）
シェイプ	XXXX/仮想マシンVM Standard 2.1	XXXX/仮想マシンVM Standard 2.1（プライマリと同じ）
合計ノード数	N/1	N/1（プライマリと同じ）
Oracle Databaseのエディション	Enterprise Edition High PerformanceまたはEnterprise Edition Extreme Performance/Enterprise Edition Extreme Performance	XXXX/ Enterprise Edition Extreme Performance（プライマリと同じ）
使用可能なストレージ・サイズ	XXX/256	XXX/256（プライマリと同じ）
SSH公開鍵	XXXX	YYYY（プライマリと異なっていても可）
ライセンス・タイプ	LI、BYOL/BYOL	LI、BYOL/BYOL（プライマリと異なっていても可）
SSH公開鍵	XXXX	YYYY（プライマリと異なっていても可）
仮想クラウド・ネットワーク	XXXX/soacsdrASH	YYYY/soacsdrPH（システムは遠隔地にあると想定しているため、プライマリとは異なる）[14]

14 プライマリDBとスタンバイDBとの間にSQLNet接続が必要です。Data Guardのトラフィックは暗号化されているため、パブリック・インターネットを使用してこの接続性を確保できます。DBシステムを使用するSOACSシステムでサポートされる限り、動的ルーティング・ゲートウェイのような他のオプションを使用でき、推奨されています。デフォルトでは、SOACSで使用されるDBシステム・インスタンスはパブリック・ネットワーク内に存在する必要があり、適切なインターネット・ゲートウェイを使用して外部からその1521ポートにアクセスできるようになっている必要があります。DBへのアクセスがVPNトンネル経由または動的ルーティング・ゲートウェイ経由で構成されていた場合は、プライマリとスタンバイの中間層から相応のアクセスができるようにすることが必要になります。OCI上のSOACSのネットワーク要件について詳しくは、SOACSとOCIのドキュメントを参照してください。

サブネット	XXXX/パブリック・サブネットefXT:US-ASBURN-AD1	YYYY/パブリック・サブネットefXT:PHX-AD3（システムは遠隔地にある想定であるため、プライマリとは異なる）
ホスト名のプリフィックス	XXXX/soacsASH	YYYY/soacsPH（プライマリと異なっていても可）
データベース名	XXXX/ORCL	XXXX/ORCL（プライマリと同じ）
データベースのバージョン	XXXX/12.2.0.1.180417	XXX/12.2.0.1.180417 （「Display All available versions」にチェックを入れ、プライマリと同じものが選択されていることを確認します）
PDBの名前	XXXX/PDBTEST	XXXX/PDBTEST（プライマリと同じ）
データベース管理者のパスワード	XXXX/Acme1234#	XXXX/Acme1234#（プライマリと同じ）
データベース・ワークロード	オンライン・トランザクション処理	オンライン・トランザクション処理
自動バックアップ	X/チェックを入れる	X/チェックを入れない（スタンバイではバックアップを無効にします）

注：セカンダリ・サイトに作成されるデフォルト・データベース・インスタンスは、Data Guardスタンバイ・データベースとして使用できないため後で削除します。必要なライフサイクル・スクリプトがプライマリDBと同じ構成でシステムにシードされるようにするために、スタンバイ・データベースはプライマリと同じ名前で作成します。

セカンダリDBの作成が完了したら、必要なパッチを両方のロケーション（プライマリとセカンダリ）のDBに適用し、両方のパッチ・レベルが同じになるようにしてください。より正確には、異なるDBドメインをまたぐData Guard構成には、データベース12cバージョンのバグ22611167の修正が必要です。opatch出力にチェックを入れることによってプライマリとセカンダリの両方のDBシステムでこのバグのパッチが適用されているかどうか確認し、未適用の場合には適用してください。使用している正確なデータベース・バージョンに関係のあるパッチがプライマリ・システムとスタンバイ・システムに適用されていることを確認してください。最新のOCI 12cR2 DBシステムは、このバグのパッチが事前インストールされています。このパッチはリリース18c以降では必要ありません。

3. 以下の手順を実行してData Guardを構成する前に、OCIコンソールで適切な受信ルールが定義され、プライマリとセカンダリのデータベース間接続ができるようになっていることを確認してください。また、それぞれのデータベースが、適切なリスナー・ポートでそれ自体の"パブリック"IP[15]に到達できるようになっている必要もあります。
4. OTNから*dataguardit_primary*.shスクリプトをダウンロードしてプライマリ・データベース・ノードにコピーします。スクリプトはこちらから入手できます。
5. *oracle*ユーザーでスクリプトを開き、説明を読み、最初のセクションで説明されている変数をカスタマイズします。

15 このIPはREDO転送に使用されるIPであり、dataguard構成スクリプトで入力として指定されます。パブリックIP（プライマリ・データベースとスタンバイ・データベースがインターネット・ゲートウェイを使用して通信する場合）または内部IP（プライマリ・データベースとスタンバイ・データベースが動的ルーティング・ゲートウェイを使用して通信する場合。こちらを推奨）を使用できます。

6. カスタマイズが済んだら、*oracle*ユーザーでスクリプトを実行します。スクリプトの説明にあるように、このスクリプトを実行すると、プライマリからスタンバイへのコピーが必要なtarファイルが作成されます。tarファイルが作成される場所は、スクリプトの変数を使用してカスタマイズできます。

7. 出力tarファイルをスタンバイ・データベース・ノードにコピーします。プライマリDBノードでopcユーザーとして次のコマンドを実行します。

```
[oracle@soacsdrDBa ~]$ scp -i cloud_private_sshkey.ppk /tmp/ORCL.GZ opc@standby_DB System_public_ip:/tmp/
```

たとえば、次のように指定します。

```
[oracle@soacsdrDBa ~]$ scp -i /u02/install/fca-toshare.ssh.ppk /tmp/ORCL.GZ opc@129.146.141.24:/tmp/
```

8. スタンバイで*opc*ユーザーを使用して、ファイルに対する権限を変更し、*oracle*ユーザーによる読取りを可能にします。

```
[opc@soacsdrDB2 ~]$chmod o+r /tmp/ORCL.GZ
```

9. OTNから*dataguardit_standby_root*.shスクリプトをダウンロードして、セカンダリ・データベース・ノードにコピーします。スクリプトはこちらから入手できます。

10. *root*ユーザーでスクリプトを開き、説明を読み、最初のセクションで説明されている変数をカスタマイズします。

注：Oracle Public Cloudのドキュメントに書かれているように、プロビジョニング済みのクラウド・インスタンスへのrootアクセス権は、opcユーザーに接続して次のsudoコマンドを実行することで取得できます。

```
[opc@soacsdb ~]$ sudo su -
```

---

11. カスタマイズが済んだら、rootユーザーでスクリプトを実行します。既存のデータベース・インスタンスが削除され、プライマリを複製した新しいデータベース・インスタンスが作成されます。また、Oracle Data Guard Brokerに必要なデータベース・リスナーと構成もセットアップされます。スクリプトの実行状況を監視し、エラーがないか確認します。このスクリプトからログ・ファイル（*/tmp/dataguardit_date.log*）が作成されます。トラブルシューティングのための情報はこのログでチェックします。スクリプトが失敗した場合は再実行することができます（前に失敗したときの内容はすべてクリーンアップされます）。

12. スクリプトが完了したら、プライマリ・システムからData Guard Broker CLIを入力して構成を確認します（REDOを適用して同期が取れるまで時間がかかる場合があります）。

```
DGMGRL> show configuration verbose

Configuration - ORCL_ORCLB_12:55:19-22-11-18

  Protection Mode: MaxPerformance
  Members:
  orcl - Primary database

    orclb - Physical standby database

  Properties:
```

FastStartFailoverThreshold = '30'

OperationTimeout = '30'

TraceLevel = 'USER'

FastStartFailoverLagLimit = '30'

CommunicationTimeout = '180'

ObserverReconnect = '0'

FastStartFailoverAutoReinstate = 'TRUE'

FastStartFailoverPmyShutdown = 'TRUE'

BystandersFollowRoleChange = 'ALL'

ObserverOverride = 'FALSE'

ExternalDestination1 = ''

ExternalDestination2 = ''

PrimaryLostWriteAction = 'CONTINUE'

ConfigurationWideServiceName = 'ORCL_CFG'

Fast-Start Failover:DISABLED

Configuration Status:

SUCCESS

13. **任意：プライマリとセカンダリのTCPソケットの最大サイズをData Guardの推奨値に設定します。**

[root@soacsdb ~]$ sysctl -w net.core.rmem_max=10485760
[root@soacsdb ~]$ sysctl -w net.core.wmem_max=10485760

インスタンスを再起動しても変更内容が保持されるようにするために、/etc/sysctl.confファイルを編集して変更を反映させることも必要です。

注：dataguardit_standby_root.shスクリプトでオンラインREDOログのデフォルト・サイジングを行います。Database Cloud Serviceで作成されるデフォルトのREDOログはサイズが1 GBでしたが、オラクルが実施したテストではこれで十分でした。オンラインREDOログのサイズは次の計算式で求めますが、1 GB未満にならないようにする必要があります。

1分あたりのピークREDOレート x 20

REDOレートは、バッチ処理、四半期末処理、年末処理といったピーク・ワークロード期間中のAWRレポートから取得できます。平均ではなく、ピーク・ワークロードを使用することが非常に重要です（平均を使用するとピークREDOレートがはっきりせず、小さすぎるREDOログがプロビジョニングされる可能性があります）。

MAAのベスト・プラクティスでは、オンラインREDOログ（ORL）のグループ数より1つ多い数のスタンバイREDOログ（SRL）を作成することを推奨しています。Oracle RACの場合は各スレッドについてこれを行う必要があります。各スレッドのREDOログ・グループの数は次の問合せで確認できます。

select thread#, count(group#) from v$log group by thread#;

SRLは、もっとも大きいORLと同じサイズで作成する必要があります。グループ番号はORLと共有であるため、SRLは大きいグループ番号を使用して作成します。もっとも大きいORLのサイズと現在のもっとも大きいグループ番号を求めるには、次の問合せを実行します。

```
select max(bytes), max(group#) from v$log;
```

MAAベスト・プラクティスでは、SRLを二重化しません。

Oracle RACデータベースの場合

Oracle RACデータベースの場合は、OCIコンソールを使用してData Guardを構成するよう強く推奨します。特定のシナリオにおいてこれが実現不可の場合にのみ、Oracle RAC DBシステム用にOracle Data Guardを手動で構成するため、RAC-DG-setup- scripts.zipのスクリプトを代わりに使用できます（これらのスクリプトは、シングル・インスタンス・データベースでOracle DGを手動構成するためのスクリプトとは異なる点に注意してください）。この場合は、RAC- DG-setup-scripts.zipをダウンロードし、README.txtの指示に従ってください。これらのスクリプトでは、『*Creating a Physical Standby using RMAN Duplicate (RAC or Non-RAC)(Doc ID 1617946.1)*』で説明されている、2つのOCI Oracle RAC DBシステム間でData Guardを構成するための手順を自動化しています。
プライマリとスタンバイのOracle RACが（同じバージョン、エディション、ノード数などで）すでにプロビジョニングされており、動的ルーティング・ゲートウェイを介して通信可能であることが要件です。

付録B – 単一テナンシーを使用したDRシステムの作成

このホワイト・ペーパーの「構成要件」の項で説明したように、単一テナンシーを使用してSOACS/MFTCS DR構成を作成することが可能です。SOACS/MFTCSシステムで使用されるWebLogicドメインの名前とWebLogicサーバーの名前は、プロビジョニング・プロセス中に指定されるSOACS/MFTCS クラウド・インスタンスの名前に基づきます。そのため、SOACSクラウド・インスタンスのサービス名として*soacsdroci*（例）を使用している場合は、*soacsdro_cluster*という名前の1つのクラスタと*socsdro_server_1*と*soacsdro_server_2*という名前の2つのサーバーで構成される*soacsdro_domain*という名前のWebLogicドメインがプロビジョニング・ツールによって作成されます。一意のテナンシーの下でこうしたWebLogic名の一貫性をプライマリとスタンバイとで維持するには、プライマリ・クラウド・サービス・インスタンスで使用する名前を8文字以上にする必要があります（WebLogicドメイン、クラスタ、サーバーの名前に使用できる文字数の上限は8文字です）。そうすると、この同じテナンシー内でセカンダリ・インスタンス名（たとえば*soacsdrocistandby*）を作成して、まったく同じWebLogicアーティファクト名（*soacsdro_domain*、*soacsdro_cluster*、*soacsdro_server_1*、*soacsdro_server_2*）を取得することができます。この場合はOracle Cloud Serviceインスタンス名が競合していないため、単一テナンシーを使用してプライマリとスタンバイを作成できるというわけです。要するに、プライマリとスタンバイのSOACS/MFTCSインスタンスの名前は最初の8文字を同じにする必要があるということです（例：*soacsdroci/soacsdrocistandby*、*soacsdmycompany/socsdrmycompanystandby*、*mycompanyprimary/mycompanysecondary*など）

単一テナンシーを使用することになったら、次の考慮事項を踏まえて、このホワイト・ペーパーの「SOACS/MFTCS DRのデプロイメント」の項で説明したのと同じ手順を実行します。

1. 単一テナンシーを使用する場合は、DBシステムに使用できるOCIコンソールの構成オプションを使用してプライマリ・データベースのData Guardを構成できます（Oracle Data Guardのセットアップについては、Database Cloud Serviceのドキュメントを参照してください）。つまり、付録Aに書かれている手順に従ってData Guardを手動で構成する必要はありません。ただし、次の考慮事項を念頭に置いてください。
    a. 自動Data Guard構成に対応したデータベース"フレーバー"（データベース・バージョン、RAC、デプロイメント・プラットフォームなど）がSOACSとMFTCSでサポートされている必要があります。
    b. SOACS DRシステムに使用する2つの正確なロケーションでこのData Guard構成オプションを使用できる必要があります。
    c. このホワイト・ペーパーのさまざまなセットアップの項（フィジカル・スタンバイからフィジカル・スナップショットへの変換、フィジカル・スタンバイに戻す変換など）で説明したように、Data Guard構成を管理するためにデータベース・ノードにアクセスすることが必要になります。
    d. Oracle Data Guard構成が自動的に作成される場合でも、DBソフトウェアに最新のパッチが適用されていることを確認してください（異なるDBドメインをまたぐData Guard構成の場合はbug バグ22611167の修正を適用する必要があります）。使用している正確なデータベース・バージョンに関係のあるパッチがプライマリ・システムとスタンバイ・システムに適用されていることを確認してください。
2. 自動Data Guard構成を使用してSOACS/MFTCSをセットアップする場合は、このドキュメントの「セットアップの詳細」に書かれているのと同じ手順を実行します。ただし、データベース・サービスを作成し、そのデータベースのData Guardを構成するプロセスは、Cloudコンソールの自動Data Guard構成プロセスにそのまま置き換わります。これについて詳しくは、Oracle Database Cloud Serviceのドキュメントを参照してください。
3. Data Guardの構成が完了し、適切なパッチを適用したら、前述の項でテナンシーが2つのケースについて説明したときと同様に、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・スタンバイに変換し、セカンダリ・データベース・ノードにログインします。

4. 前述したように、SOACSセカンダリ・サービスの作成時は、最初の8文字がプライマリのサービス名と共通する適切な9文字（以上）のサービス名を使用します。
5. 2つのテナンシーの場合について前の項で説明したように、DRSツールを実行する必要があります。
6. ライフサイクルの手順は同じで、Data Guardシステムを手動で作成したかCloudコンソールを使用して作成したかは関係ありません。

付録C – DBシステムでのクラウド・バックアップ

DBシステムのバックアップは、あらゆるOracleデータベース環境にとって重要な側面です。Oracle Cloudはさまざまなアプローチを提供します。バックアップをローカルまたはクラウド・ストレージに保存したり、RMANまたはDBCLIを使って自動化、カスタマイズしたりすることができます。DRシナリオの場合、データベースはOracle Data Guardで構成されるため、特別な考慮点がいくつかあります。

Oracle Data Guardを（前の付録Aで説明した手順に従って）手動で構成する場合、最適なOracle Data Guard環境を実現するには、バックアップを手動で構成する必要があります。
データベースの1つで（プライマリまたはスタンバイ）バックアップを実行し、別のデータベースでアーカイブ・ログの増大を制御することができます。
プライマリDBシステムで手動バックアップを構成する方法：

- このシステムのOCIコンソールで自動バックアップを有効にした場合、バックアップ・モジュールがすでに自動バックアップによって構成されています。その場合、カスタマイズできるように自動バックアップを無効にします。これまでに自動バックアップを有効にしたことがない場合は、「RMANによるオブジェクト・ストレージへのデータベースのバックアップ」の手順に従って、プライマリDBでバックアップ・モジュールをインストールし、構成してください。
- リンク先の推奨事項に従ってRMAN設定を構成してください。また、Oracle Data Guardsに推奨されているアーカイブ・ログ削除ポリシーも必ず含めてください。

  ```
  RMAN> CONFIGURE ARCHIVELOG DELETION POLICY TO BACKED UP 1 TIMES TO 'SBT_TAPE'
  APPLIED ON ALL STANDBY;
  ```

- バックアップ要件に従ってRMANバックアップ・スクリプトを作成し、crontabに含めます。以下はあくまでも、全体バックアップを実行する例です。

  ```
  # RMANを実行します

  export ORACLE_HOME=/u01/app/oracle/product/12.2.0.1/dbhome_1
  export ORACLE_SID=ORCL
  $ORACLE_HOME/bin/rman <<RMAN
          connect target /
          SET ENCRYPTION ON;

          BACKUP DATABASE PLUS ARCHIVELOG TAG "FULL_BACKUP";

          exit;

  RMAN

      echo "Completed full backup for" $ORACLE_SID
  ```

スタンバイでのアーカイブ・ログの増大を制御する方法：

- このシステムで自動バックアップを有効にしていた場合は無効にし、アーカイブ・ログ削除ポリシーがまだスタンバイに適用されていない場合に削除されないように適切なアーカイブ・ログ削除ポリシーを構成します（アーカイブ・ログ削除ポリシーを"applied on all standby"に設定）。
- FRAでのアーカイブ・ログの増大を制御するには、適切なアーカイブ・ログ削除ポリシーを設定すれば十分なはずですが、古いアーカイブ・ログを作成するクリーンアップ・スクリプトを作成することもできます。以下は、古いアーカイブ・ログを消去する例で、アーカイブ・ログの誤削除を回避するために、アーカイブ・ログ削除ポリシーを使用しています。

  ```
  ########################################################
  # データベースがスタンバイ・ロールで、バックアップが実行されない場合

  # ディスクから古いアーカイブ・ログを消去するにはこのスクリプト使用します
  # RMANを実行します
  export ORACLE_HOME=/u01/app/oracle/product/12.2.0/dbhome_1
  export ORACLE_SID=ORCL
  $ORACLE_HOME/bin/rman <<RMAN
          connect target /
  ```

```
        # このDBがプライマリ・ロールの場合に不要なアーカイブ・ログが削除されないようにする場合
        CONFIGURE ARCHIVELOG DELETION POLICY TO APPLIED ON ALL STANDBY;
        # 20日以上経過したアーカイブ・ログを削除します
        delete noprompt archivelog all completed before 'SYSDATE-20';
        exit;
RMAN
    echo "deleted applied old archivelogs on $ORACLE_SID"
#########################################################
```

OCIコンソールを使ってOracle Data Guardが構成されている場合、プライマリ・データベースで自動バックアップを有効にすることができます。このアプローチをお奨めします。このような場合、デフォルトのRMAN構成では、Oracle Data Guardのシナリオで推奨されるアーカイブ・ログ削除ポリシーを使用しています。ただし、前述したように、セカンダリ・データベースでのアーカイブ・ログの増大を制御する場合もあります。

注：トポロジでのOracle Data Guard構成により、ほとんどのデータベース障害シナリオに対する保護機能が提供されます。言い換えると、プライマリ・データベースで障害が発生するほとんどの場合において、スタンバイへのスイッチオーバーによって操作を再開させることができます。プライマリに障害が発生し、スタンバイへのスイッチオーバーが不可能という極端な場合では、プライマリをバックアップからリストアする必要が生じる可能性があります。そのようなまれなシナリオでは、スタンバイ・データベースも再作成する必要があります。

手動Data Guardでは、プライマリ・データベースでのリストア後にスタンバイ・データベースを再作成してOracle Data Guardを再構成するために、付録Aに記載されているスクリプトを再実行できます。

ただし、自動Data Guardでは、OCIコンソールはUIコンソールからスタンバイ・データベースを再作成するための機能をまだ備えていません。バックアップからプライマリ・データベースをリストアするには、Data Guardの関連付けを削除して（スタンバイDBシステムを終了することで実行）、プライマリ・データベースをリストアしたらその関連付けを再び有効化することが必要です。これによって新しいスタンバイDBシステムが作成されます。この新しいスタンバイDBシステムを使用してSOACS DRを再構築するには、「スタンバイDBシステム再作成」に記載されている手順を実行します。

付録D – Enterprise Manager Site Guardを使用したSOACS DRのスイッチオーバーの管理

Oracle Site Guardは、管理者によるサイト全体のスイッチオーバーやフェイルオーバーの自動化を可能にするディザスタ・リカバリ・ソリューションです。Oracle Fusion Middleware、Oracle Fusion Application、Oracle Databaseを組織化して連係を取りながらフェイルオーバーさせることができます。Oracle Site Guardsには次のメリットがあります。

- ディザスタ・リカバリ処理を完全に自動化して1回のクリックで起動
- ディザスタ・リカバリに要する時間の最小化
- 人為的エラーの削減
- 柔軟でカスタマイズ可能
- 特殊なスキルが不要
- 1つの画面でディザスタ・リカバリを管理
- オンデマンドの、またはスケジューリングされたディザスタ・リカバリ・ドリルを使用してディザスタ・リカバリに確実に対応

このホワイト・ペーパーですでに説明したように、SOA Cloud Serviceのディザスタ・リカバリのスイッチオーバーはOracle Site Guardを使用して調整することができます。ホワイト・ペーパー『Oracle Site Guardを使用したOCI PaaSシステムのディザスタ・リカバリの管理』での手順に従って、SOA Cloud Serviceのディザスタ・リカバリ環境用にOracle Site Guardを構成し、これを使用してスイッチオーバーとフェイルオーバーを実行してください。

付録E – その他のライフサイクル操作

SOACS DRシステムのスケーリング

現行のCloudバージョンではSOACS DRサービスでスケールアウト操作を実行することはできません。

検証のためにセカンダリ・サイトを開く

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ことで、完全なスイッチオーバーを実行しなくてもスタンバイ・サイトを検証できます。これにより、セカンダリSOAサーバーをスタンバイ・サイトで起動し、セカンダリ・システムを検証することができます。スナップショット・スタンバイ・モードの間にスタンバイ・サイト・データベースで実行された変更はすべて、そのスタンバイが再びフィジカル・スタンバイに変換されると破棄されるため、プライマリ・データがセカンダリの検証によって影響を受けることはありません。

ただしこの操作は注意して行う必要があります。これは、スナップショットへの変換時にデータベース内に保留中のメッセージやコンポジットが存在する場合には、スタンバイSOAサーバーで起動時にそれらを処理できるからです。スナップショット・スタンバイへの変換時にプライマリ・データベースに保留中のアクションが存在しないことを確認するか、スナップショット・スタンバイ・データベースの実行時SOA表からレコードを削除してから、セカンダリ管理対象サーバーを起動します
（「https://docs.oracle.com/en/middleware/soa-suite/soa/12.2.1.4/administer/managing-database-growth.html - GUID-3AE61279-BC23-4389-93CC-8D9F7E8B4B2ERemoving Records from the Runtime Tables Without Dropping the Tables」を参照）

スイッチオーバーを実行せずにスタンバイ・サイトを検証する手順は以下のとおりです。

	検証のためのセカンダリ・サイトを開く手順	詳細
1	スタンバイDBをスナップショット・スタンバイに変換する	プライマリDBホストでDGブローカを使用して、セカンダリをスナップショット・スタンバイに変換します。ユーザーoracleで次を実行します。 [oracle@drdbA ~]$ dgmgrl sys/your_sys_password@primary_db_unqname DGMGRL> convert database “secondary_db_unqname” to snapshot standby “show configuration”コマンドを使用して変換が正常に実行されたことを確認します。

2	セカンダリに保留中のアクションがないことを確認する	スタンバイがスナップショットに変換されるときにプライマリDBに保留中のアクション（トランザクション、メッセージ）が存在した場合は、セカンダリ・サーバーで起動時に処理できます。 SOAのtruncateスクリプトを使用してセカンダリ・データベース（スナップショット・スタンバイになった後）のSOA実行時表からレコードを削除し、セカンダリ・サーバーを起動する前に実行時データを消去することができます。「Removing Records from the Runtime Tables Without Dropping the Tables」を参照してください。 プライマリDBの表を切捨てないよう、このアクションは注意して実行してください。
3	セカンダリ・サイトでサーバーを起動する	セカンダリ管理サーバーを起動します。例： cd /u01/app/oracle/middleware/oracle_common/common/bin ./wlst.sh wlst> nmConnect ('weblogic','acme1234#','soacsdrs-soa-0','5556','soacsdrs_domain','/u01/data/domains/soacsdrs_domain','SSL') wlst> nmStart('soacsdrs_adminserver') セカンダリ管理対象サーバーを起動します（セカンダリWebLogicコンソールかスクリプトを使用）。
4	検証する	これはスイッチオーバーではなく、プライマリには依然としてプライマリ・ロールがあるため、フロントエンド名はプライマリを指します。従って、すべてのアクセスはプライマリにリダイレクトされます。 セカンダリのOracle SOAに直接アクセスするには、制御されたクライアント（ラップトップなど）で/etc/hostsファイルを更新し、セカンダリ・フロントエンドIPによって解決されたフロントエンド名を設定します。

注：

*ORA-01403: no data found ORA-06512エラー。ここで説明するように、（完全なスイッチオーバーを実行せずに、すなわちスタンバイをスナップショット・スタンバイ・モードで開いて）セカンダリ・サイトを検証する間、スタンバイSOAサーバーのログに"ORA-01403: no data found ORA-06512"エラーが記録されることがあります。これらのエラーは、SOA自動消去ジョブに関連しています。これらのエラーが発生するのは、データベース内のジョブにDBロールの依存関係があるからです（それらのジョブは、データベースがプライマリ・ロールの場合にのみ有効化されるように定義されています）。これは、期待される望ましい挙動で、これによりジョブが2回実行される（プライマリで1回、スタンバイで1回）のを防止します。SOAの自動消去ジョブはプライマリ・ロールで定義されるため、データベースがスナップショット・スタンバイ・モードの場合にはDBA_SCHEDULER_JOBSビューに表示されません。ジョブごとに定義されるdatabase_roleは、DBA_SCHEDULER_JOB_ROLEビューで確認できます。まとめると、これらのエラーは、スタンバイ・システムに表示される限り無視して構いません。*SOA自動消去のスケジューラ・ジョブは、インスタンスがそのロールをPRIMARYに変更する場合にのみ、DBで実行されます。

セカンダリ・サイトが検証されたら、以下の手順に従ってスタンバイ・ロールに戻します。

	スタンバイをスタンバイ・ロールに戻す手順	詳細
1	セカンダリ・サイトで管理対象サーバーと管理サーバーを停止する	セカンダリWebLogicコンソールに接続し、セカンダリ・サイトの管理対象サーバーと管理サーバーをシャットダウンします。
2	スタンバイDBを再びフィジカル・スタンバイに変換する	プライマリDBホストでDGブローカを使用して、セカンダリDBを再びフィジカル・スタンバイに変換します。ユーザーoracleで次を実行します。 [oracle@drdbA ~]$ dgmgrl sys/your_sys_password@primary_db_unqname DGMGRL> convert database "secondary_db_unqname" to physical standby "show configuration"コマンドを使用して変換が正常に実行されたことを確認します。
4	クライアントの/etc/hostsの任意の更新を元に戻す	セカンダリ・サイトを指すように、任意のクライアントの/etc/hostsファイルを更新した場合、元に戻して、フロントエンド名がプライマリ・フロントエンドIPを再び指すようにします。

DBFSウォレットの再作成方法

中間層ホストの<domain_home>/dbfsフォルダ内dbfsウォレット、tnsnames.ora、dbfsMount.shは、DRのセットアップ時に更新されます[16]。SchemaPrefix_DBFSユーザーのパスワードが変更されているためにウォレットを更新する必要がある場合は、SOA Cloud Serviceのドキュメントの「DBFSウォレットのパスワードを更新する」で説明されている手順を実行できます。ただし、dbfsのマウントに使用されたエイリアスは（デフォルトの"ORCL"エイリアスではなく）PDB名であることを考慮に入れてください。エイリアスを生成するコマンドでは、次のようにPDB名を使用する必要があります。

```
$middleware_home/oracle_common/bin/mkstore -wrl /u01/data/domains/domain_name/dbfs/wallet
-create < /var/tmp/dbfsp
$middleware_home/oracle_common/bin/mkstore -wrl /u01/data/domains/domain_name/dbfs/wallet
-createCredential <PDB_NAME> SchemaPrefix_DBFS < /var/tmp/dbfsp
```

<domain_home>/dbfs/フォルダのdbfsMount.shスクリプトでは、dbfsクライアントのマウント・コマンドの<PDB_NAME>エイリアスを必ず使用してください。例：

```
$ORACLE_HOME/bin/dbfs_client -o wallet /@<PDB_NAME> -o direct_io $MOUNT_PATH_DIRECTIO &>dbfs.log &
$ORACLE_HOME/bin/dbfs_client -o wallet /@<PDB_NAME> $MOUNT_PATH &>dbfs.log &
```

また、<domain_home>/dbfs/tnsnames.oraに、ローカルPDBを参照するエイリアス<PDB_NAME>が含まれていることも確認してください。

dbfsをマウントするには、dbfs_clientコマンドを直接使用する代わりに、<domain_home>/dbfs/dbfsMount.shスクリプトを使用することをお奨めします。dbfs_clientコマンドを使用する場合は、必ず正しいエイリアスを使用してください。SchemaPrefix_DBFSユーザーでのパスワード変更の後には、プライマリとスタンバイ両方のホスト中間層でウォレットの再作成を実行する必要があります。これは、<domain_home>/dbfs/フォルダが各ドメインに固有のものだからです（また、プライマリからスタンバイにレプリケートしないようにする必要があります）。

注：残りのスキーマ・パスワード（SOAINFRA、STBなど）を更新するには、プライマリ・ドメインで「Change the Database Schema Password Manually」で説明されている手順を実行し、dbfscopy.shを使用してセカンダリ・ドメインに変更をレプリケートすることができます。ドメイン構成時のデータソースおよびその他のファイルでのパスワード変更は、セカンダリにレプリケートされます。

スタンバイ・サイトでコンピューティング・インスタンスを停止させる場合

スタンバイ・データベースは、プライマリから更新データを受信せず、非同期の状態になるため、通常の業務を行っている期間にはシャットダウンしないようにする必要があります。シャットダウンすると、スイッチオーバーを実行することが必要になった場合にデータを損失する可能性があります。さらに、プライマリとセカンダリのデータベース間のREDOでの解決不能なギャップでは、物理スタンバイの完全再インスタンス化と構成が必要になる場合があります。したがって、プライマリとセカンダリのデータベース間の接続が長時間切断されないようにすることが推奨されます。これには、セカンダリが停止するシナリオや、通常の業務中に2つのサイト間での通信が阻止される可能性があるネットワーク・レベルの問題などが含まれます。

スタンバイ中間層のコンピューティング・インスタンスは、プライマリに影響を及ぼすことなく停止することができますが、ディザスタ・リカバリにおいて次のような影響があります。

- RPOでの影響：セカンダリ管理サーバー・ホストが停止すると、プライマリ・サイトからレプリケートされるドメイン構成の変更は、セカンダリ・ドメイン構成にプッシュされません。フェイルオーバーの場合、セカンダリ・ドメインはプライマリ構成と同期が取れなくなる可能性があります。これを回避するには、少なくともセカンダリ管理サーバー・ホストを起動したままの状態で維持し、他の管理対象サーバーのコンピューティング・インスタンスのみを停止する必要があります。
- RTOでの影響：セカンダリ中間層ホストが停止し、それらのホストを開始し、スイッチオーバーやフェイルオーバーの前にプライマリの変更とセカンダリ・ドメインを同期する必要がある場合には、RTOが増加します。

16 これらの更新は、プライマリ管理ノード・ホストとセカンダリ中間層ホストでのDRセットアップ時に実行されます。それ以外のプライマリ中間層ホストは<domain_home>/dbfsフォルダに元のdbfsウォレット、tnsnames.ora、dbfsMount.shを維持します。これは、dbfsマウントへのプライマリ構成のコピーにはプライマリ管理ノードのみが使用されるためです。DRセットアップが完了したら、プライマリ管理ノード・ホストから残りのプライマリ管理対象サーバー・ホストにこれらのファイルをコピーすることによって均等化できます。

- バックアップ/リストアでの影響：セカンダリSOACSインスタンスのPSMによって実行されるSOACSバックアップには、リストアの場合に問題を起こす可能性がある停止したインスタンスのバックアップは含められません。

これらの影響を最小限に抑えるため、いくつかのセカンダリSOAのコンピューティング・インスタンスを停止させる場合には、セカンダリ管理サーバー・ホストを起動したままの状態で維持し、管理対象サーバーのコンピューティング・インスタンスのみを停止できます。

注：

顧客の請求状態はこのドキュメントの対象外です。いくつかのサーバーを停止させることの請求への影響を確認するには、請求状態に関する確認を得るため、オラクルのライセンス・チームにお問い合わせください。
すべての場合において、インスタンスのOSを使用してインスタンスを停止しても、そのインスタンスの請求は停止されません。この方法でインスタンスを停止する場合は、必ずコンソールまたはAPIからも停止してください。停止したコンピューティング・インスタンスの請求は、通常、OCIコンピュート・モデルに従います。

スタンバイDBシステム再作成

いくつかのシナリオでは、スタンバイDBシステムを完全に再作成する必要があります。たとえば、自動化DGでは、プライマリDBシステムがバックアップからリストアされた場合、OCIコンソールにはUIコンソールからスタンバイ・データベースを再作成するための機能がまだありません。バックアップからプライマリ・データベースをリストアするには、Data Guardの関連付けを削除して（スタンバイDBシステムを終了することで実行）、プライマリ・データベースをリストアしたらその関連付けを再び有効化することが必要です。これによって新しいスタンバイDBシステムが作成されます。

SOACS DR環境では、プライマリDBシステムでData Guardを再び有効化してスタンバイDBシステムを再作成するときに、スタンバイDBシステムに対して以前と同じ値を指定することを推奨します（同じVCN、同じサブネット、同じホスト名接頭辞）。そうすることで、この新しいDBシステムをスタンバイDBとして使用するためにSOACS DRシステムで必要になる変更が最小限になります。

新しいスタンバイDBシステムを使用してSOACS DRを再構築するには、以下の手順を実行します。

a) これから終了する元のスタンバイDBシステムのDBの一意の名前（$ORACLE_UNQNAME）、プライベートIPとパブリックIP、VCN、サブネット、ホスト名接頭辞をメモしておきます。
b) スタンバイDBシステムが終了したら、プライマリDBシステム・ホストの/etc/hostsファイルを確認します。終了したスタンバイDBホストに関するエントリがある場合は、削除するかコメントアウトします。スタンバイDBホスト用の新しいエントリがその作成時に自動的に追加されます。
c) OCIコンソールを使用してプライマリDBシステムでData Guardを再び有効化するときに、以前のスタンバイDBシステムで使用していたものと同じVCN、同じサブネット、同じホスト名接頭辞を指定するようにしてください。そうすると、新しいスタンバイDBシステムと以前のスタンバイDBシステムで異なる値は、DBの一意の名前、プライベートIP、パブリックIPだけになります。
d) 新しいDBシステムが正常に作成され、OCIコンソールでData Guard構成が完了したら、新しいスタンバイDBシステムのDBの一意の名前、パブリックIP、プライベートIPの値をメモしておきます。
e) スタンバイSOACSホストで実行：
    - ファイル$DOMAIN_HOME/dbfs/localdb.logを編集します。
      このファイルには、元のスタンバイ・システムのDBの一意の名前が含まれています。この値を、新しいスタンバイDBシステムのDBの一意の名前に置き換えます。
    - ファイル$DOMAIN_HOME/dbfs/tnsnames.oraを編集します。このファイルには、エイリアスがいくつか含まれています。その1つが元のスタンバイDBシステムの一意の名前になっています。エイリアスとエイリアスのサービス名について、この元のスタンバイDBの一意の名前を、新しいスタンバイDBの一意の名前に置き換えます。
f) プライマリSOACSホストで実行：
    - ファイル$DOMAIN_HOME/dbfs/tnsnames.oraを編集します。このファイルには、エントリがいくつか含まれています。その1つが元のスタンバイDBシステムの一意の名前になっています。この元のスタンバイDBの一意の名前を、新しいスタンバイDBの一意の名前に置き換えて（エイリアスとサービス名で）、元のスタンバイIPアドレスを新しいスタンバイIPアドレスに置き換えます。
      スタンバイCDBのtnsnames.oraのエイリアスは、プライマリとセカンダリのSOAホストで異なる可能性があります。その場合、プライマリでは、スタンバイIPアドレスを使用してセカンダリCDBを参照しており、スタンバイSOAホストでは、スタンバイのホスト名が使用されています。この状態は想定範囲内です。リージョンを横断したDNS解決は想定されていないためです。

- プライマリSOAホストのlocaldb.logファイルを更新する必要はありません。このファイルにはプライマリの一意の名前が含まれており、この名前は変更されていないためです。

g) さらに、元のスタンバイDBシステム固有のIPアドレスに対して作成された既存のOCIセキュリティ・ルールのすべてが、新しいスタンバイDBシステムのIPアドレスを使用するように更新されているかを確認します（この作業は、CIDRではなくIPアドレス固有のルールとなっていた場合に限り必要です）。

これで、SOACS DR環境が新しいスタンバイDBシステムを使用できるようになりました。ここでは例として以下の値を使用します。

	元のスタンバイDBシステム	新しいスタンバイDBシステム
DBの一意の名前（$ORACLE_UNQNAME）	ORCL6_phx1kg	ORCL6_phx1c3
DBシステムのプライベートIP	10.2.0.2	10.2.0.5
DBシステムのホスト名	drdb6b.mysubnet.region2vcn.oraclevcn.com	<同じ値>
DBシステムのスキャン名	drdb6b-scan.mysubnet.region2vcn.oraclevcn.com	<同じ値>

- この場合、スタンバイSOACSホストでは以下のようになります。

更新対象のファイル	元のコンテンツ	新しいコンテンツ
$DOMAIN_HOME/dbfs/localdb.log	ORCL6_phx1kg	ORCL6_phx1c3
$DOMAIN_HOME/dbfs/tnsnames.ora	（多くのエントリ） ... ORCL6_phx1kg＝ (DESCRIPTION＝ (SDU=65536) (RECV_BUF_SIZE=10485760) (SEND_BUF_SIZE=10485760) (ADDRESS=(PROTOCOL=TCP)(HOST＝drdb6b-scan.mysubnet.region2vcn.oraclevcn.com)(PORT=1521)) (CONNECT_DATA＝ (SERVER=DEDICATED) (SERVICE_NAME＝ ORCL6_phx1kg. mysubnet.region2vcn.oraclevcn.com))) ...	（多くのエントリ） ... ORCL6_phx1c3＝(DESCRIPTION ＝(SDU=65536) (RECV_BUF_SIZE=10485760) (SEND_BUF_SIZE=10485760) (ADDRESS＝(PROTOCOL＝TCP)(HOST＝drdb6b-scan.mysubnet.region2vcn.oraclevcn.com)(PORT＝1521)) (CONNECT_DATA＝(SERVER ＝DEDICATED) (SERVICE_NAME＝ORCL6_phx1c3. mysubnet.region2vcn.oraclevcn.com))) ...

- 次に、プライマリSOACSホストでは以下のようになります。

更新対象のファイル	元のコンテンツ	新しいコンテンツ
$DOMAIN_HOME/dbfs/tnsnames.ora	（多くのエントリ） ... ORCL6_phx1kg＝ (DESCRIPTION＝ (SDU=65536) (RECV_BUF_SIZE=10485760) (SEND_BUF_SIZE=10485760) (ADDRESS＝(PROTOCOL＝TCP)(HOST＝ 10.2.0.2)(PORT＝1521)) (CONNECT_DATA＝ (SERVER＝DEDICATED) (SERVICE_NAME＝ORCL6_phx1kg. mysubnet.region2vcn.oraclevcn.com)) ） ...	（多くのエントリ） ... ORCL6_phx1c3＝ (DESCRIPTION＝ (SDU=65536) (RECV_BUF_SIZE=10485760) (SEND_BUF_SIZE=10485760) (ADDRESS＝(PROTOCOL＝TCP)(HOST ＝10.2.0.5)(PORT＝1521)) (CONNECT_DATA＝ (SERVER＝DEDICATED) (SERVICE_NAME＝ORCL6_phx1c3. mysubnet.region2vcn.oraclevcn.com)) ） ...

付録F - DRセットアップのネットワーク要件のサマリー

次の表に、SOACS DRに特有のネットワーク要件を示します。

アクション	SSH	SQLNET（1521）	HTTPS
DRセットアップ（DRSを使用）	DRSが実行されているホストから、すべてのDBホストと中間層ホスト、およびyaml構成ファイルで設定されているIP。 （通常はパブリックIPですが、DRSが要塞または内部サブネットを介してノードに接続可能な場合はプライベートIPに設定可能）。	プライマリ・リージョンとセカンダリ・リージョンが動的ルーティング・ゲートウェイを介して通信する場合は、すべてのセカンダリ中間層ホストからプライマリDBのプライベートIP（および、Oracle RACの場合はスキャンIP）。 または すべてのセカンダリ中間層ホストからプライマリDBのパブリックIP（プライマリ・リージョンとセカンダリ・リージョンがインターネットを介して通信する場合）[17]。	DRSが実行されているホストからプライマリ・フロントエンドIP。 DRSが実行されているホストからセカンダリ・フロントエンドIP。
構成レプリケーション（dbfscopy.sh）	スクリプトを実行するユーザーは、SOA管理ノードでスクリプトを直接実行するため、自分のロケーションからSSHアクセスする必要があります。	プライマリ・リージョンとセカンダリ・リージョンが動的ルーティング・ゲートウェイを介して通信する場合は、各中間層管理ホストからリモートDBのプライベートIP（および、Oracle RACの場合はスキャンIP）。 または 各中間層管理ホストからリモートDBのパブリックIP（プライマリ・リージョンとセカンダリ・リージョンがインターネットを介して通信する場合）[17]。	
通常実行時		プライマリとセカンダリのデータベース間（これはData Guardの場合の要件です）。	

17 OCIでは、異なるリージョンに配置されたVCNネットワーク間のプライベート・トラフィックに動的ルーティング・ゲートウェイを使用できるため、最新バージョンでは推奨されていません。

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

ORACLE

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：Oracle SOA Cloud Serviceのディザスタ・リカバリ - クラウドでの本番環境とDR
2021年3月

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