



Oracle SOA Suite on Oracle Cloud Infrastructure Market の ディザスター・リカバリ

Oracle Cloud Infrastructure (OCI) での本稼働とDR

2024年7月 | バージョン27

Copyright © 2024, Oracle and/or its affiliates 公開

本書の目的

本書には、Oracle SOA Suite Cloud on Marketplaceのディザスター・リカバリ・ソリューションを構成するための説明、要件の要約、セットアップ手順が記載されています。本書の対象読者は、Oracle Cloud、Oracle SOA Suite、Oracle WebLogic、Oracle Database、Oracle Data Guard、Oracle Databaseのバックアップ/リカバリに関する知識を持つ技術者です。

改訂履歴

本書には下記の改訂が行われています。

日付	改訂	コメント
2020年6月	1	初版
2020年9月	2	「ベスト・プラクティス」のポイントを追加 「その他のライフサイクル運用に関する付録C」（スケール・アウト、検証用にスタンバイをオープン）を追加 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Aに、自動Data Guardでのリストアに関する注記を追加
2020年12月	3	“セカンダリSOA Suite on Marketplaceのプロビジョニング”の表を更新
2021年1月	4	“検証のためのセカンダリ・サイトのオープン”に注記を追加 入力ミスを訂正し、いくつかの表現を改善
2021年3月	5	「付録C – その他のライフサイクル運用」に「スタンバイDBシステムの再作成」を追加
2021年4月	6	OCI FSS/rsyncを使用したWLSドメイン構成レプリケーションのための追加のDR方法を含める機能強化。 いくつかのセクションを更新
2021年7月	7	“MFTクラスタ”サービス・タイプのサポートも追加 リージョン間ブロック・ボリューム・レプリケーションを使用したWLSドメイン構成レプリケーションのための追加の DR方法を含める機能強化。「付録D – リージョン間ブロック・ボリューム・レプリケーションに基づくディザスター・ リカバリ」を追加
2021年8月	8	OEL8ボックスから実行できるDRSパッケージ・バージョンを追加
2021年9月	9	5ページに脚注を追加 スイッチオーバーの操作にOCI DNSスイッチオーバーの例を追加
2021年10月	10	図を更新。「前提条件」→「データベース」に情報を追加
2022年1月	11	“RTOおよびRPOの概要”的ポイントを追加
2022年2月	12	Data Guard手動セットアップ・スクリプトを更新。カスタム・リソース名の注記を追加
2022年4月	13	「EXACSに関する考慮事項」の特定箇所を追加
2022年6月	14	「付録C - その他のライフサイクル運用」に「パッチ適用」に関する箇所を追加
2022年6月	15	リンクを更新 Site Guardの非推奨化のため、Oracle Site Guardの参照を削除（Doc ID 2875372.1を参照）
2022年7月	16	必要なホスト・エイリアスのためにDNSビューを使用する方法を追加 リンクを更新
2022年10月	17	「付録E - プライマリでの追加スタンバイ・データベースの使用」を追加 「構成レプリケーションのエンド・ツー・エンド検証」をライフサイクル手順に追加「検証のためのセカンダリ・サイトの オープン」をライフサイクル手順に移動
2023年2月	18	データソース内でTNSエイリアスを使用。複数のセクションを更新
2023年5月	19	ブロック・ボリュームDR手法の考慮事項を更新。
2023年6月	20	“カスタム・ファイル”ポイントの改善
2023年7月	21	開始/停止スクリプトへのリンクを追加

2023年9月	22	SOAMPバージョンに関する注を追加
2023年11月	23	ブロック・ボリュームDRの方法でのボリューム・グループの使用を提示
2023年12月	24	フェイルオーバー後のアクションを追加
2023年12月	25	PDB用のCRS管理サービスの作成手順を追加
2024年1月	26	ドキュメントの再編成を完了。すべての方法をドキュメント本編に記載。
2024年7月	27	SOAMPでのOracle Linux 8バージョンに関する注記を追加

目次

本書の目的	1
改訂履歴	1
はじめに	5
SOA Suite on Marketplaceのディザスター・リカバリ	7
トポロジの説明	7
レプリケーション方法	8
前提条件	15
要件	17
スクリプトのダウンロード	20
ディザスター・リカバリのセットアップの概要	21
ディザスター・リカバリのセットアップの準備	22
1. 仮想フロントエンド名を選択する	22
2. プライマリ中間層での仮想フロントエンドの準備	22
3. プライマリ中間層でのTNSエイリアスの使用の準備	24
4. セカンダリ・サイトのデータベースをセットアップする	25
5. セカンダリ・サイトのSOA Suite on Marketplaceをプロビジョニングする	31
6. セカンダリ中間層での仮想フロントエンドの準備	35
7. セカンダリ中間層でのTNSエイリアスの使用の準備	35
8. 必要な中間層ホスト・エイリアスの構成	36
ディザスター・リカバリのセットアップの完了	38
ロック・ボリューム・レプリケーションを使用して構成する	38
1. スタンバイDBのフィジカル・スタンバイへの変換	38
2. リージョン間ロック・ボリューム・レプリケーションの構成	38
3. 環境固有情報への置換用スクリプトの準備	40
DBFSの方法とFSS/rsyncの方法を使用して構成する	41
1. DBFSの方法でDBFSステージング・マウントを構成する	41
2. FSS/rsyncの方法でFSSステージング・マウントを構成する	41
3. ディザスター・リカバリ・セットアップ・ユーティリティ (DRS) の実行	43
DRセットアップの検証	45
BVLレプリケーションの方法のライフサイクル手順	46
BVLレプリケーションの方法の構成レプリケーション	46
BVLレプリケーションの方法の検証のためのセカンダリ・サイトのオープン	46
BVLレプリケーションの方法のスイッチオーバー	49
BVLレプリケーションの方法のフェイルオーバー	52
BVLレプリケーションの方法のスケール・アウトとスケール・イン	53

DBFSの方法とFSS/rsyncの方法のライフサイクル手順	57
構成レプリケーション	57
検証のためのセカンドリ・サイトのオープン	61
DBFSの方法とFSS/rsyncの方法のスイッチオーバー	62
DBFSの方法とFSS/rsyncの方法のフェイルオーバー	64
DBFSの方法とFSS/rsyncの方法のスケール・アウトとスケール・イン	65
DBFSウォレットの再作成	68
一般的なライフサイクル手順	69
スタンバイ・サイトでコンピュート・インスタンスを停止させる場合	69
プライマリとスタンバイで管理対象サーバー数が異なる場合	69
SOAMP DR環境へのパッチ適用	70
スタンバイDBシステム再作成後のSOAMP DRの再構築	71
RTOおよびRPOの概要	74
予測されるRTO	74
予測されるRPO	75
ベスト・プラクティス	77
結論	78
付録A - 手動構成されたData GuardでのDBシステム・バックアップ[†]	79
付録B - DRセットアップのネットワーク要件のサマリー	81
付録C - プライマリでの追加スタンバイ・データベースの使用	82
ローカル・スタンバイの構成前の追加手順（DBFSの方法とFSS/rsyncの方法）	82
ローカル・スタンバイのリージョン間の追加DR構成（DBFSの方法とFSS/rsyncの方法）	83
ローカル・スタンバイの構成後の追加手順（DBFSの方法とFSS/rsyncの方法）	84
ローカル・スタンバイのローカル・データベースの追加スイッチオーバー（DBFSの方法とFSS/rsyncの方法）	87

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、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、計画停止時間を最小限に減らすことができます。高可用性機能には、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データソース、ロードバランシング、フェイルオーバー、バックアップリカバリ、ローリング・アップグレード、ローリング構成の変更などがあります。

Oracle SOA Suite on Marketplace (SOAMP) は、SOAアプリケーションをクラウド内で実行するためのPlatform as a Service (PaaS) コンピューティング・プラットフォーム・ソリューションを提供します（Oracle SOA Suite、Oracle Service Bus、Oracle B2B、Oracle Managed File Transferなど）。Oracle SOA Suite on Marketplaceは、Oracle Cloud Infrastructureに完全に依存しているPaaSソリューションです。これは、OCI Console Marketplaceを使用してプロビジョニングされ、他のOCIコンポーネント（OCIロードバランサなど）およびOCIインフラストラクチャのライフサイクル手順（バックアップリカバリなど）と完全に統合されています。基本インフラストラクチャとして、Oracle Compute、Oracle Cloud Database、Oracle WebLogicを使用します。SOAMPでは、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のメタデータ、および他のOracle Fusion Middleware（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。SOA Suite on Marketplaceインスタンスでは、Oracle Cloud Infrastructure Database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。

Oracle Cloud Infrastructure内のものも含め、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ます。このディザスター・リカバリによる保護機能では、中間層（Oracle SOA Suite on Marketplace）、データ層（Oracle Cloud Database）、ロードバランサ（LBR）層（OCI LBRまたはサードパーティ）を対象とする必要があります。そのためのソリューションとして、プライマリ本番サイトとは地理的に離れた場所にある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を、本番サイトと比較してそれ以下にすることは可能ですが、オラクルは、同じ容量を備えたミラー構成の実行を推奨しています。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（本番ワーカーロードは維持されません）、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。

Oracle SOA Marketplaceでは、单一データセンター内の高可用性がすでに実現されていることに注意してください。Oracle SOA Marketplaceは、コンピュート・インスタンス・ノードをプロビジョニングするとき、コンピュートにアクティブな高可用性（HA）ポリシーを使用します。つまり、プライマリ・コンピュート・ノードに障害が発生した場合、仮想マシン（VM）は、同じコンピュート・ゾーン内の別の物理コンピュート・ノードに自動的にフェイルオーバーされます。さらに、SOAMPはデフォルトで、クラスタの各コンピュート・インスタンスに異なるOCI Fault Domainを使用します。複数の可用性ドメインを含むOCIリージョンでは、SOAMPのプロビジョニングにより、各コンピュート・インスタンスが別々の可用性ドメイン（AD）に配置されます。これは、リージョン・サブネットがデプロイに使用されている場合は常にデフォルトで実行されます。同様に、SOAMPによって使用されるフロントエンドLBRは、リージョン内にあり、（複数のADがあるリージョンでは）デフォルトでADを横断してフェイルオーバーされます。SOAMPによって使用されるOracle Databaseを（別の可用性ドメインに配置された）Oracle Data Guardのローカル・スタンバイによってADの障害に対して保護することにより、SOAMPシステムには、コンピュート・インスタンス、障害ドメイン、または可用性ドメイン内のあらゆる種類の停止に対して完全なレジリエンスが備わります。

ただし、このすべてでは、SOA Marketplaceシステムを（リージョン全体に影響する）リージョンの障害に対して保護するわけではありません。

本書は、特に**Oracle SOA Suite on Marketplace**のリージョンを横断したディザスタ・リカバリ（DR）に対応することを念頭に作成されています。本書は、廃止されたSOA Cloud Serviceには適用されません。

Oracle SOA Suite on Marketplaceでは、Oracle Fusion MiddlewareとOracle Databaseに搭載された高可用性機能と災害からの保護機能を活用することにより、極めて厳しいリカバリ時間目標（RTO）とリカバリ・ポイント目標（RPO）を達成できます。クラウド・ディザスタ・リカバリ構成に特有の考慮事項もありますが、Oracle Fusion Middleware（Oracle FMW）とOracle Databaseを使用するすべて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のベスト・プラクティスについて詳しく説明し、SOA Suite on Marketplaceの災害からの保護をデプロイする手順の概要を示します。Oracle SOA on Marketplaceディザスタ・リカバリ・ソリューションでは、SOAコンポーネントに必要とされる構成ファイルをレプリケートします。同一のWebLogicクラスタ上にデプロイされたカスタム・アプリケーションでは、他の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。本書では、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で使用するOracle Cloud Databaseは、Oracle Data Guardを使用して災害から保護します。

本書は、Oracle SOA Suite on Marketplaceの“SB & B2Bクラスタを使用するSOA”サービス・タイプおよび“MFTクラスタ”サービス・タイプに適用されます。対象読者は、**Oracle WebLogic Server**、**Oracle FMW SOA**、**Oracle Database**、**Data Guard**、**Oracle Databaseのバックアップとリカバリ**に関する知識を持ち、**Oracle Cloud**¹で提供されるサービスについて基本的に理解している技術者です。

¹<https://cloud.oracle.com/home>

SOA Suite on Marketplaceのディザスタ・リカバリ

トポロジの説明

Oracle SOA Suite on Oracle Cloud Marketplaceのディザスタ・リカバリ・ソリューションは、**アクティブ-パッシブ・モデル**です。a) SOA Suite on Marketplaceデプロイメント、b) ロードバランサ、c) Oracle Cloud Infrastructure DBから構成される**プライマリ・システム**をすべて同じリージョンおよびテナントにデプロイし、SOA Suite on Marketplaceデプロイメント、ロードバランサ、Oracle Cloud Infrastructure DBシステムから構成される**ピア・スタンバイ・システム**を別のリージョン（同じテナント）にデプロイします。

本書で使用する“リージョン”、“データセンター”、“サイト”という用語は、どれもOracle OCIリージョンを指します。“リージョン”、“データセンター”、または“サイト”は、同じ障害に影響を受けないほど**十分遠く地理的に離れている、物理的に実在する場所**です。たとえば、ッシュバーンとフェニックスは、本書において、2つの異なるデータセンターであり、サイトであり、リージョンです。

プライマリとスタンバイのOracle Cloud Infrastructure DBシステムには、Data Guardが構成されます。Data Guardの機能を使用して、プライマリ・データベースに適用されるすべての変更がセカンダリ・データベース（“スタンバイ”データベースとして動作）にレプリケートされます。

スタンバイのSOA Suite on Marketplaceドメインは、**プライマリ・ドメインのレプリカ**です。プライマリ・ドメインと同じドメイン名、スキーマ、パスワードなどを使用しますが、セカンダリ・データベースを参照します。セカンダリ・データセンターのWebLogic Serverのリスナー・アドレスはプライマリ中間層ホスト名で構成されるため、セカンダリでは、セカンダリIPに解決されるよう関連する別名が作成されます。本書では、このスタンバイ・システムを構築および構成する手順を説明します。

フロントエンドでは、システム内で実行されるアプリケーションにアクセスするための一意の名前が設定されます。この“仮想”フロントエンド名は、プライマリ・サイトのOCIロードバランサのIPを参照します。スイッチオーバーするとき、このフロントエンド名は、セカンダリ・サイトのOCIロードバランサのIPを参照するように更新されます。フロントエンド・ホスト名は常に、その時点でプライマリとして機能するサイトのLBRのIPに解決されます。

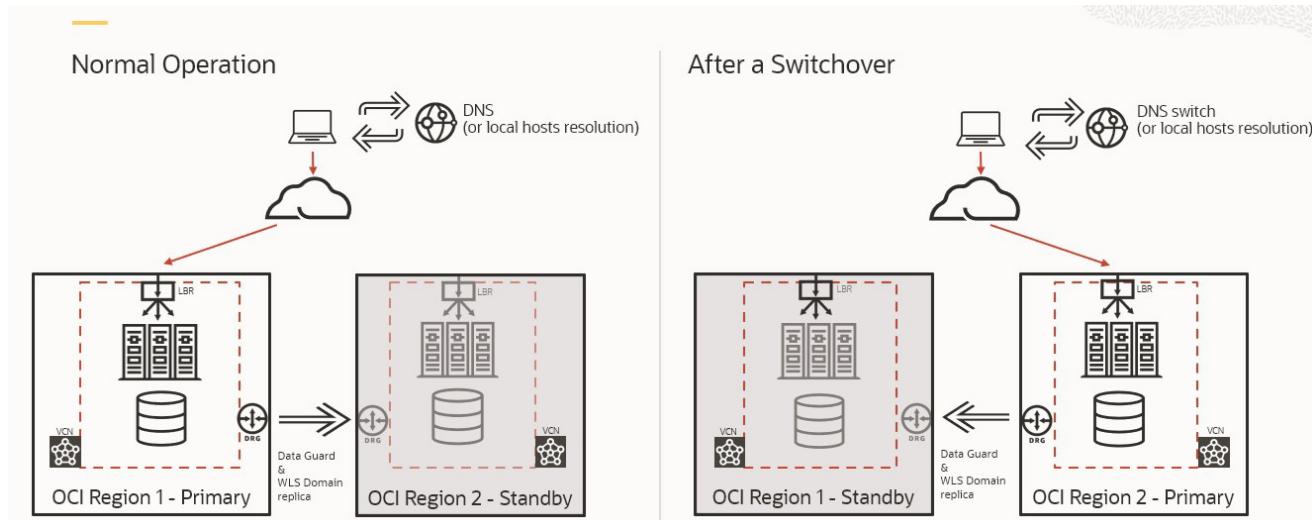


図1：SOA Suite on OCI Marketplaceのディザスタ・リカバリのトポロジ

通常の業務におけるスタンバイ・データベースは**フィジカル・スタンバイ**です。マウント状態か、Active Data Guardを使用する場合は読み取り専用モードで開かれます。スタンバイ・データベースは、プライマリからREDOを受け取り、適用しますが、読み取り/書き込みモードで開くことはできません。このドキュメントで説明する初期DRセットアップや他のライフサイクルの手順など、一部の操作の実行時には、スタンバイ・データベースがスナップショット・スタンバイに変換されます。スナップショット・スタンバイ・モードのデータベースは全体的に更新可能なデータベースです。スナップショット・スタンバイ・データベースは、プライマリ・データベースからREDOデータを受け取ってアーカイブしますが、適用はしません。スナップショット・スタンバイに対して実行されたすべての変更是、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きに破棄されます。

スタンバイ・データベースは、通常の業務を行っている期間にシャットダウン・ステータスになると、プライマリからREDO更新データを受信せず、非同期の状態になります。この状態でスイッチオーバーを実行する必要が生じると、データ損失が発生する可能性があります。そのため、通常の業務の遂行中にスタンバイ・データベースを停止させることは推奨しません。スタンバイの中間層ホストを停止させると発生コストを削減できますが²、スタンバイ・サイトの管理サーバー・ノードが停止している場合、プライマリ・サイトからレプリケートされる構成変更はセカンダリ・ドメイン構成にプッシュされません。

使用されている構成レプリケーション方法に応じて、システムのRPOに及ぶ影響が異なります（以下の「レプリケーション方法」を参照）。つまり、スイッチオーバー・イベントが発生した場合、スタンバイ中間層ホストを起動し、ドメインをプライマリの変更と同期させることが必要になると、RTOの値が増加します。セカンダリの中間層ホストを起動しておくこと（WebLogicプロセスは停止）を推奨します。詳細については「[スタンバイ・サイトでコンピュート・インスタンスを停止させる場合](#)」を参照してください。

レプリケーション方法

このMAAソリューションでは、データベース内に存在するすべての情報が、Oracle Data Guardによって自動的にセカンダリ・サイトにレプリケートされます。これには、SOAスキーマ、OPSS情報、カスタム・スキーマ、TLOG、JDBC永続ストアなどが含まれます。

ただし、ローカル・ファイル・システム内にあるWebLogicドメイン構成についても、プライマリからセカンダリにレプリケートする必要があります。初期レプリケーションは、初期DRセットアップ中に実行されます。このレプリケーションは、システムのライフサイクル中も、理想的にはプライマリ・ドメインで構成変更が実行されるたびに繰り返す必要があります。

両方のロケーションで対応するWebLogicドメイン構成を維持するため、主に2つのアプローチを使用できます。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) WebLogicドメイン構成が**頻繁には変更されない**場合は、本番とスタンバイで1回ずつ、**合計2回、構成変更を手動で適用することを推奨します**。このために、先にセカンダリ・データベースをスナップショットに変換してから、管理サーバーを起動します。WebLogic構成の同期を維持するため、以下の手順に従って、手動によりセカンダリ・サイトで構成変更を繰り返します。

	手順	詳細
1	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WLS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。
2	スタンバイ・データベースをスナップショット・スタンバイに変換する	プライマリ・データベース・ホストでoracleユーザーとして以下を実行します。 [oracle@drdba]\$ dgmgrl sys/your_sys_password@primary_db_unqname DGMGR> 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
3	セカンダリ・サイトでWebLogic管理サーバーを起動する ³	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。管理サーバーのみを起動し、管理対象サーバーは起動しないことが重要です。
4	セカンダリ・サイトで構成変更を繰り返す	セカンダリ・ロケーションのWebLogic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。
5	セカンダリ・サイトのWebLogic管理サーバーを停止する	セカンダリ・サイトのWebLogic管理サーバーを停止します。
6	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・データベース・ホストでoracleユーザーとして以下を実行します。 [oracle@drdbaa ~]\$ dgmgrl sys/your_sys_password@primary_db_unqname

² SOA Marketplaceでは、停止したコンピュート・インスタンスの請求はOCIコンピュート・モデルに従い、コンピュート・シェイプに依存します。

<https://docs.oracle.com/ja-jp/iaas/Content/Compute/restartinginstance.htm#resource-billing>を参照してください。

³ 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い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などを推奨します。

```

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) Weblogicドメイン構成が頻繁に変更される場合、オラクルでは、WebLogicドメイン構成レプリケーションを実行するために、**DBFS (Oracle Database File System)**、**OCI File Storage Service (FSS)** /rsync、リージョン間ブロック・ボリューム・レプリケーションという3つの方法をサポートしています。これら3つの方法は同じトポロジを使用し、その仕組みも似ています。異なるのは、プライマリ・サイトからスタンバイ・サイトへの情報転送方法です。
- DBFSベースの方法**では、ドメイン構成のコピーがDBFSファイル・システムにステージングされ、Data Guardによってセカンダリ・サイトにレプリケートされます。DBFSマウントはデータベース内に存在する情報を公開するファイル・システムであり、NFSボリュームと同様にマウントできます。プライマリ・ドメインの構成がDBFSマウントにコピーされ、それが基盤のData Guard機能によって自動的にスタンバイにレプリケートされます。セカンダリ・サイトでは、中間層ホストで同じDB表からDBFSマウント・ポイントを（スタンバイ・データベースにレプリケートされる）プライマリとしてマウントします。これで、レプリケートされたドメイン構成データが利用できるようになり、DBFSマウントからセカンダリ・ドメインにコピーされます。本書では、プライマリおよびスタンバイでこのプロセスを自動化するスクリプトを提供しています。どちらのサイトでもこのスクリプトをcron単位またはスケジュールで実行し、指定した頻度で構成をレプリケートします。異なるレプリケーション・アプローチを比較するには、以下のセクションを参照してください。

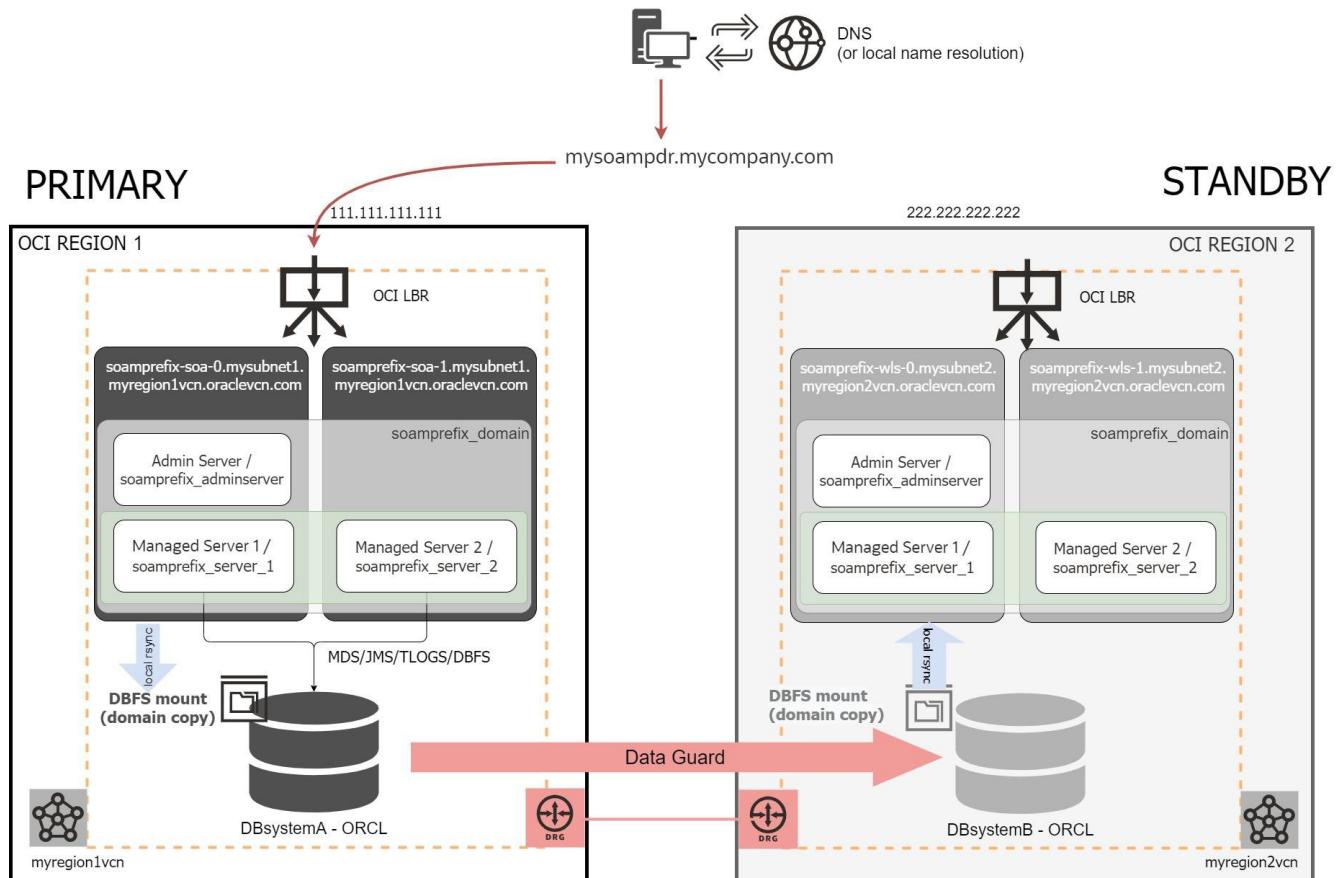


図2：WLSドメイン構成レプリケーションにDBFSの方法を使用する場合の、SOA in Marketplace DRトポロジ

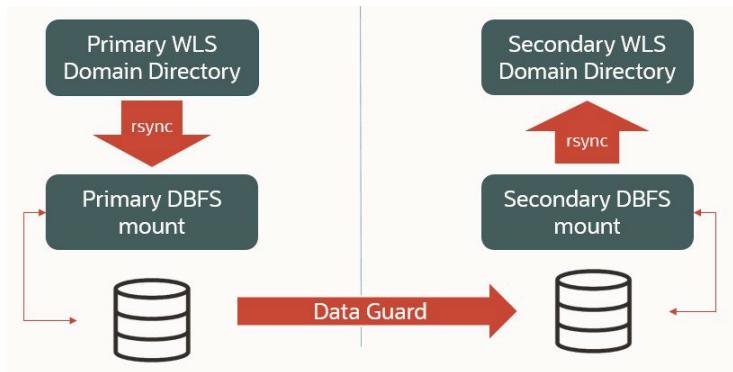


図3：DBFSの方法を使用したWLSドメイン構成レプリケーションの論理フロー・ダイアグラム

- OCI File Storage Service (FSS) /rsyncの方法では、プライマリとセカンダリに1つずつある2つのFSSファイル・システム間でrsyncを直接使用して、セカンダリ・サイトにドメイン構成を転送します。このアプローチでは、DBFSの方法の場合と同様に、中間の“ステージング”ポイントとして共有ファイル・システムを使用します。のために、OCI FSSが各リージョンにマウントされます。プライマリ・ドメイン構成をレプリケートするために、WLSドメイン・フォルダがまずローカルのステージング用OCI FSSマウントにコピーされ、次にrsyncによってリモートのOCI FSSマウントにコピーされます。その後、セカンダリ内で、ドメイン構成がセカンダリ環境のデータセンターのOCI FSSからセカンダリ・ドメイン・ディレクトリにコピーされます。本書では、プライマリおよびスタンバイでこのプロセスを自動化するスクリプトを提供しています。どちらのサイトでもこのスクリプトをcron単位またはスケジュールで実行し、指定した頻度で構成をレプリケートします。FSSはリージョン間レプリケーションを実行しますが、SOAMPドメインはFSS上に直接格納できないことに注意してください。そのため、常にドメインをステージ・ディレクトリに最初にコピーする必要があります。したがって、主なレプリケーションのRTOおよびRPOを促進するのはこのステージ・コピーです。異なるレプリケーション・アプローチを比較するには、以下のセクションを参照してください。

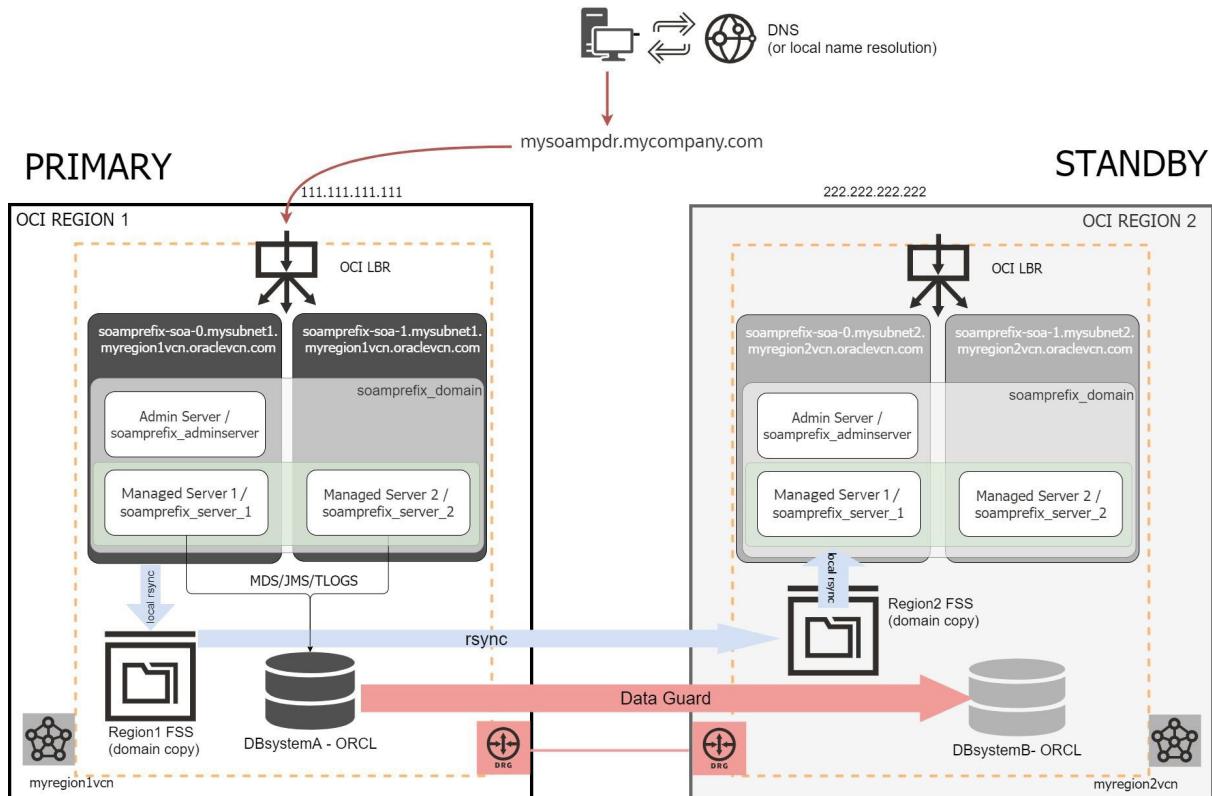


図4：WLSドメイン構成レプリケーションにOCI FSS/rsyncの方法を使用する場合の、SOA Suite in Marketplace DRトポジ
青色の矢印は構成のコピーの論理フローを示しています。rsyncコマンドは、プライマリ・サイトまたはスタンバイ・サイトのWebLogic管理ホストで実行します。つまり、リモート・コピーの場合、rsyncを使用して、プライマリ・サイトのWebLogic管理ホストからスタンバイのWebLogic管理ホストに接続します。

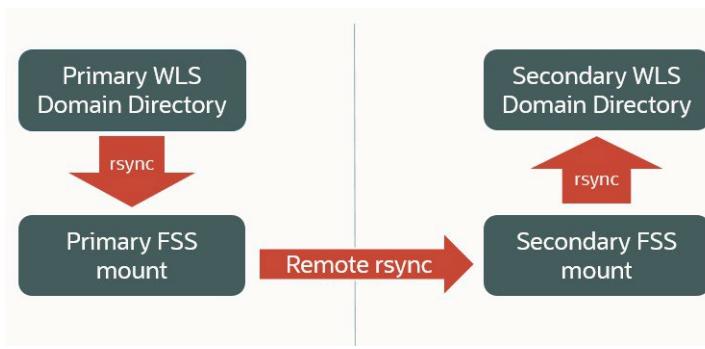


図5：OCI FSS/rsyncの方法によるWLSドメイン構成レプリケーションの論理フロー・ダイアグラム

- リージョン間ブロック・ボリューム・レプリケーション・モデルでは、WebLogicドメイン構成を含む中間層ホストのブロック・ボリューム全体が、OCIのクロス・リージョン・ボリューム・レプリケーション機能を使用してセカンダリ・サイトにレプリケートされます。リージョン間ブロック・ボリューム・レプリケーションによって、ブロック・ストレージ・ボリュームの他リージョンへの非同期レプリケーションが継続的に自動で実行されます。このアプローチでは、構成レプリケーションにステージング・ロケーションを使用しません。

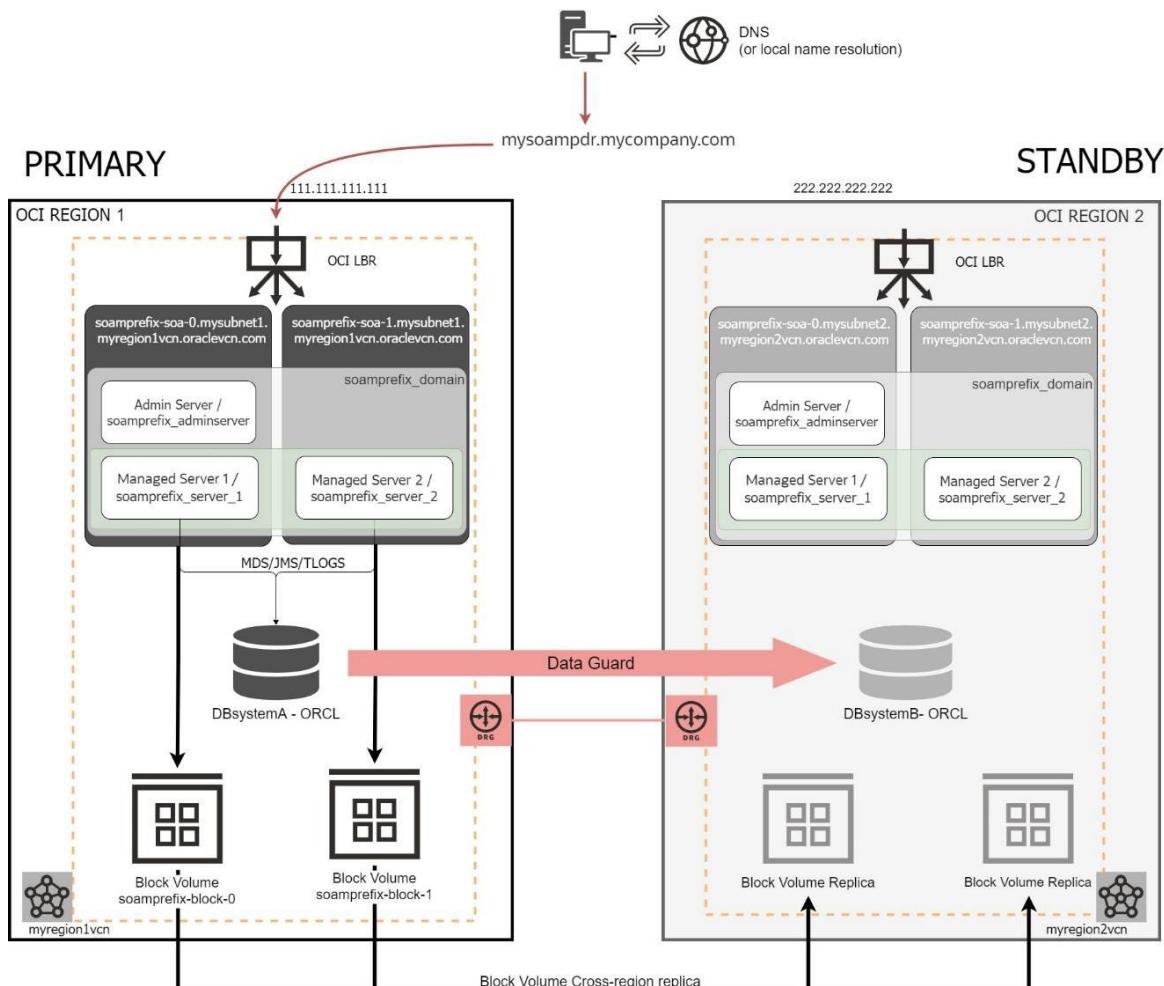


図6：リージョン間ブロック・ボリューム・レプリケーションを使用したSOAMPディザスター・リカバリ・トポロジのダイアグラム

レプリケーション方法の比較

- **総合的な動作 :**

FSS/rsyncまたはBVレプリケーションと比べると、**DBFSの方法**は、Oracle Driverの再試行ロジックによりローカルおよびリモートでの優れた可用性を実現し、動作のレジリエンスも高くなります。RACデータベースを使用してDBFS表をホストする場合、DBFSクライアントはバックエンド・ストレージ障害に対応でき、残っているデータベース・インスタンスに接続をフェイルオーバーできます。この方法は、**Data Guardレプリケーションの堅牢性**を利用したものもあります。この方法はあらゆるシナリオで使用でき、リージョン間の待機時間が長くかかるDRの場合に推奨されます。それは、Dataguardコピーが、他のレプリケーション方法では使用できない、Dataguardを通じたブロックおよびデータ検証を提供するためです。ただし、構成のレプリケートにDBFSを使用すると、セットアップ、データベース・ストレージ、ライフサイクルの観点で影響があります。これには、特定のデータベース保守（表ストレージの消去、圧縮、削減）を実行し、DBFSマウント・ポイントの動作方法を理解することが必要です。

FSS/rsyncの方法は、DBFSの方法よりも保守と構成が容易です。ただし、この方法は、待機時間が短く、動的ルーティング・ゲートウェイを使用するように構成されたOracle OCIデータセンター間でのみ推奨されます。たとえば、パブリック・インターネット経由で通信するデータセンターの場合、FSS/rsyncで信頼性の高い動作を確保できるほど待機時間が短くはない場合があります。また、FSS/rsyncの方法では、FSSの使用とプライマリ/スタンバイの中間層間で必要な接続によって、追加コストが発生する可能性があります（お客様の課金条件は本書の範囲外であるため、詳細についてはオラクルのライセンス・チームにお問い合わせください）。

- **ブロック・ボリューム・レプリケーション**の方法では、セットアップや継続的なレプリケーションのプロセスが容易で、DBFSやFSSの方法とは大きく異なります。このモデルは、継続的、自動的にレプリカを生成できます。このモデルは、ミドルウェアベースのPaaSサービスに加えて、コンピュート・インスタンスにアタッチされたブロック・ボリューム内に存在するすべてのデータに適用可能な汎用ソリューションとして使用できます。スイッチオーバーを準備するには、（DBFSやFSS/rsyncの方法に比べると）いくつかの追加手順が必要です。ただし、これらの操作は、総合的なRTOを向上させるために、スイッチオーバーに先立って実行できます⁴。しかし、クラスタ・ファイル・システムがないと、コンピュート・インスタンス間でブロック・ボリュームを共有できません。したがって、共有ストレージが中間層で必要とされる場合、簡便性と自動化というBVアプローチのメリットは、レプリケーション全体の要件に当てはまりません。

ブロック・ボリューム・レプリケーションは、個々のシステムの構成に依存しにくくなっています。SOAMP以外のシステムにもたいていは適用可能であり、システム内で他の任意のコンピュート・インスタンスのブロック・ボリュームをレプリケートするために使用できます。ここでは、（DBFSやFSSの場合のように手動またはcronによるスクリプトではなく）OCIインフラストラクチャによって実行される、継続的な自動レプリケーション・プロセスが提供されます。

ブロック・ボリューム・レプリケーションでは、データ同期はWeblogicドメイン構成に限定されません。各コンピュート・インスタンスのWebLogicドメイン・フォルダが格納されるブロック・ボリュームは<soampprefix>-data-block-Nです。これは/u01/dataにマウントされています。このブロック・ボリュームは、継続的にスタンバイ・サイトにレプリケートする必要があります。レプリケートされるブロック・ボリュームの情報は、プライマリのブロック・ボリュームをそのままコピーしたものです。つまり、WebLogicドメイン外に保管されたその他のカスタム・ファイルも、（レプリケート対象のブロック・ボリューム内にある限り）追加の操作を必要とせずに自動的にレプリケートされます。最初のスイッチオーバーでは、すべてのノードのデータ・ブロック・ボリュームをレプリケートする必要があります。後続のスイッチオーバーでは、ブロック・ボリュームのレプリケーションを使用するときに、異なる手法を使ってWebLogic構成をレプリケートできます。

1. BVレプリケーションを使用してWeblogic管理サーバーのブロック・ボリュームのみをレプリケートし、その構成をWeblogicインフラストラクチャによって管理対象サーバー・ノードに伝播させます。domain/configディレクトリの下にあるすべての構成は、WLSによって、ドメインのメンバーである他のノードの起動時にコピーされます。この手法は、Weblogic構成ディレクトリの外にはカスタマイズやアーティファクトが置かれていない場合に推奨されます。
2. Weblogic管理対象サーバー・ノードのブロック・ボリュームもレプリケートします。これは、各Weblogicノードで追加のアーティファクトやカスタマイズが使用される場合に便利ですが、より複雑でコストが高くなります。

⁴ ただし、フェイルオーバーの状況では、これらの手順がさらに操作上のオーバーヘッドを引き起こす可能性があります。

- **管理の複雑さ：**

DBFSの方法とFSSの方法における構成レプリケーション手順はかなり似通っており、Weblogicドメインのノード数は関係ありません。これとは逆に、**リージョン間ブロック・ボリューム・レプリケーション**の管理は、レプリケート対象となるブロック・ボリュームの数が多くなるとそれだけ複雑になります。ブロック・ボリュームとレプリカの適切なライフサイクル管理が必要になります。スイッチオーバーとフェイルオーバーの操作も他の方法より複雑になり、スイッチオーバー/フェイルオーバーの実行前と実行後の手順も追加されます。Weblogic管理対象サーバーのブロック・ボリュームが（管理サーバー以外に）レプリケートされると、この複雑さは増します。ただし、フル・スタックDRを使用してシステムのスイッチオーバーとフェイルオーバーを自動化すると、この複雑さを大幅に軽減できます。
- **コストの影響**

DBFSアプローチでは、コストの増加は、DBFS表をホストするデータベースで必要とされるストレージの追加に関係します。割り当てられた領域を効率的にリカバリするためには、一般的なDB表領域およびストレージの保守操作が必要です。通常、リージョン間コピーのネットワーク・オーバーヘッドは、Dataguard全体のトラフィック要件と比較すると無視できるものです。

FSS/rsyncでは、Weblogicドメインのファイル・ストレージ割当ては通常、数ギガバイトであり、rsync経由のコピーはネットワークへの影響も大きくないため、ストレージ要件は低くなります。

ただし、**ブロック・ボリューム・レプリケーション**では、ボリュームのレプリケーションを有効化した後、そのボリュームは指定したリージョンおよび可用性ドメインにレプリケートされるようになります。レプリケート先のリージョンでボリューム・レプリカ分のストレージ・コストが請求対象に加わることになります。レプリケート先のリージョンのボリューム・レプリカは、ソース・リージョンのボリュームの種類にかかわらず、ブロック・ストレージの「より低いコスト」オプション価格を使用して請求されます。Weblogic管理対象サーバーのブロック・ボリュームが（管理サーバー以外に）レプリケートされると、このコストは増します。また、リージョン間のレプリケーション・プロセスに適用されるネットワーク・コストも請求対象に含まれることになります。レプリケーション・プロセスの実行中に、ソース・ボリュームで更新中のすべてのデータがボリューム・レプリカに送信されるため、頻繁に更新されるボリュームではネットワーク・コストが高くなります。

詳細については、Oracleドキュメントの「ボリュームのレプリケート」の「レプリケーションのコストに関する考慮事項」セクションを参照してください。その一方で、DBFSとFSSのシナリオでのステージ・ディレクトリへのコピーはWeblogic管理ノードに影響し、そこではコピーの頻度、Weblogicドメインの大きさ、そして共有ストレージ上の別のファイルを同じレプリケーション・サイクルによりリージョン間でレプリケートする必要があるかどうかに応じて、メモリとCPUリソースの追加が必要になる場合があります。
- **リカバリ時間目標（RTO）**

DBFS、FSS/rsync、およびブロック・ボリューム・レプリケーションの方法におけるスイッチオーバーのRTOはほぼ同じです。しかし、フェイルオーバー操作では、ブロック・ボリューム・レプリケーションに追加手順が必要となります（レプリカのアクティビ化、ブロック・ボリュームのアタッチなど）。これにより、フェイルオーバー時の停止時間が増加します。2ノードWebLogicクラスタの通常のフェイルオーバーの場合、RTOは10分程度増加します。また、さらに時間を必要とする、スイッチオーバー後やフェイルオーバー後のタスク（ブロック・ボリュームのデタッチなど）もあります。ただし、この最後の手順はリカバリ時間の追加なしに実行でき、FSDRを使用して自動化できます。
- **リカバリ・ポイント目標（RPO）** （WebLogic構成について。データベースのRPOはすべてのレプリケーション方法においてまったく同じである）

ブロック・ボリューム・レプリケーション・プロセスは継続的に発生し、通常のリカバリ・ポイント目標（RPO）は1時間未満となります。一方、ソース・ボリュームでのデータの変更率によって、RPOは変化します。たとえば、多数のI/O操作が発生したボリュームについては、RPOが1時間を超える場合があります。詳しくは、<https://docs.oracle.com/ja-jp/iaas/Content/Block/Concepts/volumereplication.htm#volumereplication>を参照してください。

本書で説明した**DBFSおよびFSS/rsync**の方法の場合、WebLogic構成をレプリケートする際のRPOを細かく制御しやすくなります。それは、a) スケジューリングされたスクリプトを使用して情報をレプリケートしており、b) レプリケートされる情報の量がブロック・ボリューム全体よりも少ないためです。一方、DBFSおよびFSS/rsyncの方法では、受信したドメイン構成の“マネージャ”として機能するのは管理サーバーのコンピュート・インスタンスであるため、このノードの可用性と容量によって構成のコピーの速度が決まります。**ブロック・ボリューム・レプリケーション**を使用すると、2次ノードが実行されているかどうかに関係なく、レプリケーションは実行されたままになります。

以下の表は、各レプリケーション方法について上記で説明した点をまとめたものです。

	DBFSの方法	FSS/rsyncの方法	リージョン間ブロック・ボリューム・レプリケーションの方法
ステージ・ストレージの使用	あり	あり	なし
ストレージの設定の複雑さ	中	中	低
ストレージ管理の複雑さ	高	中	低 (*)
DRの設定の複雑さ	中	中	低
スイッヂオーバーとフェイルオーバーの複雑さ	低	低	低 (*)
WLS構成のレプリケーションの複雑さ	中	中	低
スイッヂオーバーのRTO	通常は15~30分	通常は15~30分	通常は15~30分 (ダウンタイムが発生しないスイッヂオーバー・タスクの前後は+約10分、FSDRによる自動化が可能)
フェイルオーバーのRTO	フェイルオーバー時間 (通常、15~30分) + 計画外停止時間	フェイルオーバー時間 (通常、15~30分) + 計画外停止時間	フェイルオーバー前のタスクの時間 (通常、10分) + フェイルオーバー時間 (通常、15~30分) + 計画外停止時間
RPO（実行時データ）	DB永続データの場合は Data Guardに依存 FSSアーティファクト上の 共有ストレージの場合は1時間 “ 予測されるRPO ”を参照	DB永続データの場合は Data Guardに依存 FSSアーティファクト上の 共有ストレージの場合は1時間 “ 予測されるRPO ”を参照	DB永続データの場合はData Guardに依存 FSSアーティファクト上の共有ストレージの場合は1時間 “ 予測されるRPO ”を参照
RPO（WLS構成）	レプリケーション用スクリプトの実行 頻度に依存。“ 予測されるRPO ”を 参照	レプリケーション用スクリプトの実行 頻度に依存。 “ 予測されるRPO ”を参照	通常のRPOは30分よりも大幅に短いですが、 ソース・ボリュームでのデータの変更率によって 変化します。
インフラストラクチャのコスト	低	低	中
(*) Full Stack Disaster Recovery Service (FSDR) が使用されている場合			

Oracle Maximum Availability Architectureのベスト・プラクティスとして、オラクルではブロック・ボリューム・レプリケーションをOCI Full Stack Disaster Recovery Serviceと一緒に使用することを推奨します。OCI Full Stack Disaster Recovery Serviceとブロック・ボリューム・レプリケーションを組み合わせることにより、リカバリ時間目標 (RTO) 、リカバリ・ポイント目標 (RPO) 、総所有コスト (TCO) 、および管理自動化にとって最適な利点が得られます。

前提条件

ロードバランサ

ディザスター・リカバリ・ソリューションは、**SOA Suite in Oracle Cloud Marketplace**スタックがOCIロードバランサと一緒に構成されることを前提としています。ロードバランサは、着信するリクエストを分散できるようにするために、クラスタに複数のサーバーが含まれる場合には必須です。

プロビジョニング中に作成されるデフォルトのロードバランサは、OCIロードバランサです。ネットワーク・トポロジに応じて、パブリックまたはプライベートのいずれかになります。

パブリックOCIロードバランサの有効範囲はリージョンです。パブリックOCIロードバランサは、リージョン・サブネット（推奨）またはおのおのが別々の可用性ドメインにある2つのAD固有のサブネットのいずれかを必要とします。リージョン内に複数の可用性ドメインが存在する場合は、1つのプライマリ・ロードバランサと1つのスタンバイ・ロードバランサが同じリージョン内に作成されますが、それぞれ別々の可用性ドメイン内に配置されます。プライマリ・ロードバランサで障害が発生すると、トラフィックの受信に使用されているパブリックIPアドレスが、同じリージョン内のスタンバイ・ロードバランサに切り替わります。サービスからは、この2つのロードバランサは同じように扱われ、ユーザーがどちらかを"プライマリ"として指定することはできません。このようにして、ロードバランサは、ロードバランサ・レイヤーにローカル（リージョン内）の高可用性を提供します。

セカンダリ・リージョンにも同じトポロジが存在することになります。セカンダリ・ドメイン内のOCI LBRは、セカンダリ・リージョンの可用性ドメインの1つに配置された1つのプライマリ・ロードバランサと、セカンダリ・リージョンの他の可用性ドメイン内の別のロードバランサで構成されます。

プライベートOCIロードバランサでも同様のトポロジが使用されますが、プライマリとスタンバイをホストするためのサブネットを1つだけ使用します。OCIプライベート・ロードバランサは、パブリック・インターネット・アクセスからシステムを分離するために使用されます。そのサブネットは、使用するサブネットのスコープに応じて、スコープ内のリージョンに配置することも、可用性ドメイン固有にすることもできます。サブネットのスコープがリージョンで、そのリージョンに複数の可用性ドメインが含まれる場合、プライマリ・ロードバランサとスタンバイ・ロードバランサはそれぞれ別のAD内に配置されます。サブネットが可用性ドメイン固有の場合、プライマリとスタンバイは同じ可用性ドメインに配置されます（この場合、可用性ドメインの停止に備えるフェイルオーバー機能はありません）。

つまり、OCIロードバランサは、クライアント・アクセスの要件に応じて、パブリックにもプライベートにもできます。マルチ可用性ドメイン・リージョンで暗黙に高可用性機能を使用するには、OCIロードバランサがパブリックとプライベートのどちらであってもリージョン・サブネット内に配置します。

これは、ディザスター・リカバリ構成として十分な構成です。プライマリ・サイトとスタンバイ・サイトのロードバランサに必要なのは、それぞれのローカルWebLogicクラスタへのルーティングだけなので、それらのロードバランサ間での構成レプリケーションは必要ありません。プライマリ・サイトのロードバランサ構成に適用された構成の変更はすべて、セカンダリ・サイトのロードバランサに手動で適用する必要があります。

詳しくは、[OCIロードバランサのドキュメント](#)を参照してください。

データベース

Oracle SOA Suite on Marketplaceでは、Oracle Platform Security Servicesの情報、SOAインスタンス追跡、コンポジットとドキュメントのメタデータ、および他のOracle FMW Infrastructureスキーマを格納するためにデータベースが必要です。また、JMS永続ストアやJTAログなどの、WebLogic Serverドメインによって保存された任意の永続情報にデータベースを使用することも、MAAベスト・プラクティスです。SOA Marketplaceには、このベスト・プラクティスが標準で実装されています。これはディザスター・リカバリのトポロジで特に有用かつ重要な構成です。フェイルオーバーまたはスイッチオーバーの後では、Data Guardレプリケーションによって、スタンバイ・サイトでもこの情報が自動的に利用可能になります。

ディザスター・リカバリ・ソリューションでは、Oracle SOA Suite on OCI Marketplaceが**Oracle Cloud Infrastructure**データベースで構成されていることを前提としています。本書に記載されている例や構成は、データベース・システムVMをOCIで使用する場合に特化したものとなっています。

本書で説明するDRトポロジでサポートされるのは、**1つのプライマリ・データベースごとに1つのスタンバイ・データベースがあるケースのみ**です。同様に、OCIコンソールのData Guard構成も、プライマリ・データベースごとに1つのスタンバイ・データベースのみに制限されています⁵。プライマリ・リージョンで追加のスタンバイを使用する場合（OCIコンソールではサポートされない構成のため、手動での管理が必要）や、後から加える予定がある場合は、「[付録C – プライマリでの追加スタンバイ・データベースの使用](#)」を参照してください。

⁵ <https://docs.oracle.com/ia/cloud/paas/base-database/data-guard/index.html>の「DBシステムでのOracle Data Guardの使用」を参照してください。

本書で説明するDRセットアップ手順は、**Oracle Base Database Service DBシステム（シングル・インスタンスおよびRAC）**と**Oracle Exadata Database Service（EXACS）**で動作保証されています。

SOAMP with Oracle OCI Autonomous Databaseでは、ウォレットと接続文字列の個々の管理が必要です。設定の詳細については、プレイブック[Configure Oracle Fusion Middleware DR on Oracle Cloud with an autonomous database](#)を参照してください。

BVレプリケーションの方法でレプリケートされたブロック・ボリューム

BVレプリケーションDRソリューションでは、**ブロック・ボリュームのホスティング構成のみが他のサイトにレプリケートされます。ブート・ボリュームはレプリケートされません。**

SOA Suite on Marketplaceの各中間層ホストには、**1つのブロック・ボリューム**がアタッチされ、/u01/dataにマウントされています。このボリュームを使用して、各コンピュート・インスタンスの**WebLogicドメイン構成が格納されます**。したがって、そのブロック・ボリュームのコンテンツのみを他のサイトにレプリケートする必要があります。

/u01/dataフォルダの外部にある他のコンテンツはブート・ボリュームの一部であり、**レプリケートはされません**。オペレーティング・システムおよびOracleソフトウェアのホームはブート・ボリューム内に格納されるため、レプリケートされません。プライマリ中間層ホストでOSまたはオラクル製品にパッチが適用された場合、同じパッチ適用手順をセカンダリのSOAMPコンピュート・インスタンスで実行する必要があります。

要件

以下のセクションでは、災害からの保護システムが適切に機能するためのセットアップの要件とライフサイクルについて説明します。一部の要件は、おのののケースで使用されるレプリケーションの方法に固有のものであることに注意してください。これに該当する場合は、セクションのヘッダーに示します。

フロントエンド・アドレス

クライアントからシステムへのアクセスは、プライマリとして使用中のサイトには依存しないようにする必要があります。このため、**システムへのアクセスに使用するフロントエンド・アドレスのホスト名を一意にする必要があり**、この名前は常に、その時点でプライマリとなるシステムのIPにマッピングします。この名前は通常、“**仮想フロントエンド**”または“**バニティURL**”と呼ばれます。

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）は、災害に対する保護用の仮想フロントエンドとして再利用することができます。たとえば、元のシステムで“soampdrs.mycompany.com”をプライマリのバニティURLとして使用していた場合は、スイッチオーバーまたはフェイルオーバー後にこの同じ仮想ホスト名をセカンダリ・サイトのロードバランサのIPに再度マッピングできます。

どちらかのサイトに仮想フロントエンド名をマッピングする場合は、適切なDNSサービス（Oracle Cloud DNS、他の商用DNS、ローカルDNS、またはローカルでのホスト解決）を使用します。本書では、仮想フロントエンド名を使用するようにSOA WebLogicドメインを構成する方法について説明します。

インスタンス名接頭辞

SOA Suite on Marketplaceのプロビジョニング中に、“**インスタンス名接頭辞**”を指定します。このプロパティは、スタックによって使用される多数のリソース名（WebLogic Serverドメイン名、クラスタ名、WebLogicサーバー名、VMのホスト名など）の作成に使用されます。

このプロパティは、**プライマリとセカンダリのSOAシステムで同じ**に設定して、両方のシステムでWebLogicリソースの名前が同じになるようにする必要があります。同じ名前を使用することで一貫性が保証され、JMSメッセージとTLOGをリカバリする際にこの名前が必要となります。名前が同じであれば、両サイトでのカスタマイズや操作も容易になります。

異なるリージョンやコンパートメントで作成されている場合は、同じCloudテナントの複数のインスタンスで同じ“**インスタンス名接頭辞**”を使用することに問題はないことに注意してください。各インスタンスは、特定のリージョンおよびコンパートメント内のみに表示されます。

SOAMPプロビジョニング・プロセスでは、ドメイン、クラスタ、管理サーバー、管理対象サーバーの接頭辞などのカスタム名を構成できるオプション機能が提供されます。その場合、名前は“**インスタンス名接頭辞**”に基づいて作成されません。代わりに、提供される値を使用します。この機能は、**プライマリとスタンバイで同じカスタム名が提供されている限り**、本書で説明されているディザスター・リカバリ・トポロジで使用できます。

サイト間のネットワーク通信

REDO転送のために、プライマリ・データベースとスタンバイ・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。DBFSレプリケーション・モデルでは、セカンダリ中間層のホストも初期セットアップのために、プライマリ・データベースと通信する必要があります。

FSS/rsyncの方法を使用する場合、rsyncコピーを実行するには、それぞれのサイトのWebLogic管理ホストが、ssh（TCP/22）でリモート・ピアWebLogic管理ホストと通信する必要があります。

具体的なネットワーク要件について詳しくは、本書の「[付録B - DRセットアップのネットワーク要件のサマリー](#)」を参照してください。

オラクルでは、**プライマリ・サイトとセカンダリ・サイトの間は**、動的ルーティング・ゲートウェイを使用して、OCIの内部ネットワーク経由で通信することを推奨しています（ネットワーク構成の詳細については、[動的ルーティング・ゲートウェイのドキュメント](#)を参照してください）。可能な場合は、ノード間の接続にプライベートIPを使用してください。サイト間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もできますが（Oracle Netのトラフィックは暗号化されます）、この方法は推奨されません。

場合に応じて適切な受信ルールを有効化します。セキュリティ・ルールは、OCIコンソールの仮想クラウド・ネットワークごとにSecurity Listsで構成します。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。

サイト間でレプリケートされるデータベース・データの量は、プライマリ・データベースによって生成されるREDOに応じて異なります。これは、アプリケーションの負荷、トランザクション能力、同時実行性など直接的な関連があります。DBFSアプローチでは、構成のレプリケーションによって生じるデータベース・オーバーヘッドは一般に、Data Guardが同期する実行時データと比較するとあまり影響しません。REDOログ・ファイルをタイムリーにスタンバイ・データベースに提供できるようにするには、プライマリ・サイトとセカンダリ・サイトの間に適切なネットワーク接続が必要です。Oracle Cloud Infrastructureのリージョンは、フルト・トレントな高帯域幅ネットワークで相互接続されており、99.95 %以上の信頼性（10,000個のパケットのうち、損失は5パケット以下）と一貫した待機時間を実現します。詳しくは、[Oracle Cloud Infrastructureのデータセンター・リージョン](#)を参照してください。

WebLogicのデータソースおよびJPSファイル内でTNSエイリアスを使用

ディザスター・リカバリのトポロジでは、以下の3つのアプローチを使用して、WebLogicデータソースのデータベース接続文字列を構成できます。

- Dataguard対応の（"デュアル"とも呼ばれる） JDBC文字列を使用する。この場合、データベース接続文字列にはプライマリとスタンバイ両方のデータベース接続アドレスが含まれますが、プライマリ・ロールを持つデータベースのみがサービスを提供します。ただし、このアプローチが推奨されるのは、スタンバイ・データベースがプライマリと同じリージョンにあり、中間層がプライマリまたはスタンバイ・データベースに接続できるストレッチ・クラスタに基づくDR環境の場合のみです。これは本書の対象外です。
- 各サイトで異なり、ローカル・データベースのみを参照する非デュアルのJDBC文字列を使用する。この場合は、WebLogicドメイン構成がプライマリからスタンバイへコピーされたたびに置換が必要となります。このアプローチは、バージョン18より前のドキュメントで使用されていました。
- **データソース内でTransparent Network Substrate (TNS) エイリアスを使用する。** TNSエイリアスの名前はプライマリとセカンダリで同一になるため、両方のサイトのデータソースで同じDB接続文字列が使用されることになります。TNSエイリアスは、サイト間でレプリケートされないtnsnames.oraファイルにより解決されます。そのため、tnsnames.oraの内容が各サイトで異なってもWebLogic構成は同一である可能性があります。各サイトでTNSエイリアスを各サイト内の適切な接続文字列へと解決して、ローカル・データベースのみを参照するようになります。この場合は、WebLogicドメイン構成がプライマリからスタンバイへコピーされる際に置換は必要ありません。これが、リモートDRシナリオの推奨アプローチです。

詳しくは、[Oracle Fusion MiddlewareのDisaster Recovery Guide](#)の「Setting Up DataSources in the Middle Tier」セクションを参照してください。

本書では、**TNSエイリアスのアプローチ**を使用します。提供するスクリプトは、WebLogicデータソースおよびJPS構成ファイル内でTNSエイリアスを使用する環境で動作するように設計されています。お使いのWebLogicシステムがまだTNSエイリアスのアプローチを使用していない場合、本書の後半で説明する手順に従って構成できます。

BVレプリケーションの方法のリージョン間ロック・ボリューム・レプリケーションを使用するリージョン

このDRトポロジで使用されるプライマリ・リージョンとスタンバイ・リージョンは異なるリージョンである必要があり、それらのリージョンの間で、リージョン間ロック・ボリューム・レプリケーションの機能が**利用可能である必要があります**。すべてのリージョンがリージョン間ロック・ボリューム・レプリケーション用に相互接続されているわけではありません。レプリケートされるボリュームのあるソース・リージョンで、レプリケート先リージョンとして選択可能なターゲット・リージョンを指定します。ボリューム・レプリケーションを利用可能なソース・リージョンとターゲット・リージョンについては、Oracle Cloudドキュメント「[クロス・リージョン・レプリケーション](#)」の「ソースおよび宛先のリージョン・マッピング」の表を参照してください。

DBFS手法とFSS/rsync手法のWebLogicドメイン構成レプリケーションのためのステージング・ファイル・システム

本書では、ステージ・ディレクトリを使用してサイト間でWebLogicドメイン構成をレプリケートする2つの方法を記載しています（DBFSとFSS/rsync）。どちらの手法でも、補助ファイル・システム（それぞれDBFSまたはFSS）が使用され、WebLogicホストにマウントされます。この後のセクションでは、それぞれのケースでステージング・ファイル・システムを構成するための具体的な手順を示します。

リージョン間ロック・ボリューム・レプリケーションに基づく方法は、ステージング・ファイル・システムを必要としない唯一の方法です。

DBFSの方法とFSS/rsyncの方法でのカスタム・ファイル

WebLogic Serverドメイン構成は、最初にサイト間で同期されます。このとき、以下の考慮事項があります。

- 各SOAシステムは、DRのセットアップの完了後に**ローカルDBへ接続され、同じスキーマを参照します（プライマリ・スキーマ）**。データソースによって使用されるtnsnames.oraを含む**TNS管理フォルダ**は、この技術概要で提供されているスクリプトのコピーから**除外されます**。各リージョンには、そのローカル・データベースをポイントするtnsnames.oraが含まれます。
サイト間の初期同期中に、最初のプライマリ・ノードのWebLogicドメイン・フォルダの内容がすべてのセカンダリ・ノードにコピーされます。
- カスタム・アプリケーションのデプロイメント（ワークフロー・タスクear、カスタムear/warファイル、デプロイメント・プラン、JMSリソースなど）および**管理サーバーのWLSドメイン・ディレクトリ以下**に存在する（一時データを除く）すべてのデータは、本書で説明されている手順を使用してサイト間で**同期されます**。

- **weblogic_domain_name/config**内の構成はすべて、WebLogicのクラスタ機能によって同一サイト内の残りのノードに自動的に配信されます。管理対象サーバーが起動すると、管理サーバーから構成が取得されます。
- weblogic_domain_name/config**ディレクトリ外のドメイン（管理サーバー内のものを除く）の他のノードにあるアーティファクトの後更新は、WebLogicクラスタ機能によっても本書に記載されている手順によってもレプリケートされません。

この後のセクションでは、構成レプリケーションの実行方法について詳しく説明します。

他のノードまたはWebLogic管理サーバーのドメイン・ディレクトリ以外にデータが存在する場合は、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。

SLAの要件

Oracle SOA Suite on Marketplace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、ディザスター・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。詳しくは、このドキュメントのRTOおよびRPOの概要セクションで説明します。

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

詳しくは、このドキュメントのRTOおよびRPOの概要セクションで説明します。

- **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ことがあります。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モデルの間はサービス・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。

さらに、セカンダリ・サイトのホストとサービスのステータスがRTOとRPOに影響を与える可能性があります。前のセクションで説明したように、スタンバイ・データベースは、通常の業務を行っている期間にシャットダウン・ステータスになると、プライマリから更新データを受信せず、非同期の状態になります。この状態でスイッチオーバーを実行する必要が生じると、データ損失（RPOへの影響）が発生する可能性があります。このため、通常業務中にスタンバイ・データベースを停止させることは推奨されません。

スタンバイの中間層ホストは停止させることができます。ただし、停止している間はプライマリ・サイトからレプリケートされる構成変更はセカンダリ・ドメイン・ホストにプッシュされません。スイッチオーバー・イベントの場合は、中間層ホストを起動してプライマリと同期することが必要になるため、RTOの値が増加します。したがって、セカンダリの中間層ホストを起動しておくこと（WebLogicサーバーは停止）をお奨めします。詳細については「スタンバイ・サイトでコンピュート・インスタンスを停止させる場合」を参照してください。

注：DRに関連したサービス・レベルに関係なく、該当するDatabase Cloud Service⁶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。

⁶ <http://www.oracle.com/us/corporate/contracts/paas-iaas-public-cloud-2140609.pdf>

スクリプトのダウンロード

SOAMP DRシステムのセットアップとライフサイクルに必要なスクリプトは、**MAA GitHubリポジトリ**で入手できます。

- a) GitHubのMAAリポジトリ（<https://github.com/oracle-samples/maa>）へ移動します。
- b) **wls_mp_dr**ディレクトリのすべてのスクリプトをダウンロードします。
- c) **app_dr_common**ディレクトリのすべてのスクリプトをダウンロードします。
- d) スクリプトにより、相互に呼出しが作成されます。ある時点で特定の操作が実行されますが、ディレクトリ全体をダウンロードして、**両方のディレクトリのすべてのスクリプトを同じフォルダ内に配置**します。スクリプトは、プライマリおよびセカンダリの両サイトで必要です。
- e) (FSS/rsyncの方法およびDBFSの方法の場合のみ) **drs_mp_soa**ディレクトリに進み、**drs-mp.tar.gz**ファイルをダウンロードします。
これは、FSSの方法およびDBFSの方法でのDRセットアップ手順で使用します。

まだスクリプトを実行しないでください。本書の手順に従って、必要に応じてスクリプトを実行してください。

注：<https://download-directory.github.io/>などのツールは、Githubからのフォルダのダウンロードに使用できます。

ディザスタ・リカバリのセットアップの概要

本書では、開始時点で、プライマリ・サイトがすでに存在し、“稼働中”であることを前提としています。地理的に離れたサイトに存在するセカンダリDR構成は、この既存のプライマリ・システムを基に作成されます。プライマリ・システムはすでに本番で実行されている可能性があるため、DR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（WebLogic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）。

「トポロジの説明」で示したように、WebLogicドメイン構成がどのようにセカンダリ・サイトにレプリケートされるかによって、**DBFS**、**FSS/rsync**、**リージョン間ブロック・ボリューム・レプリケーション**という3種類のDRモデルがあります。

次のフロー・チャートは、本書に記載した3つのモデルでのディザスタ・リカバリ・セットアップ・プロセスの手順を示しています。

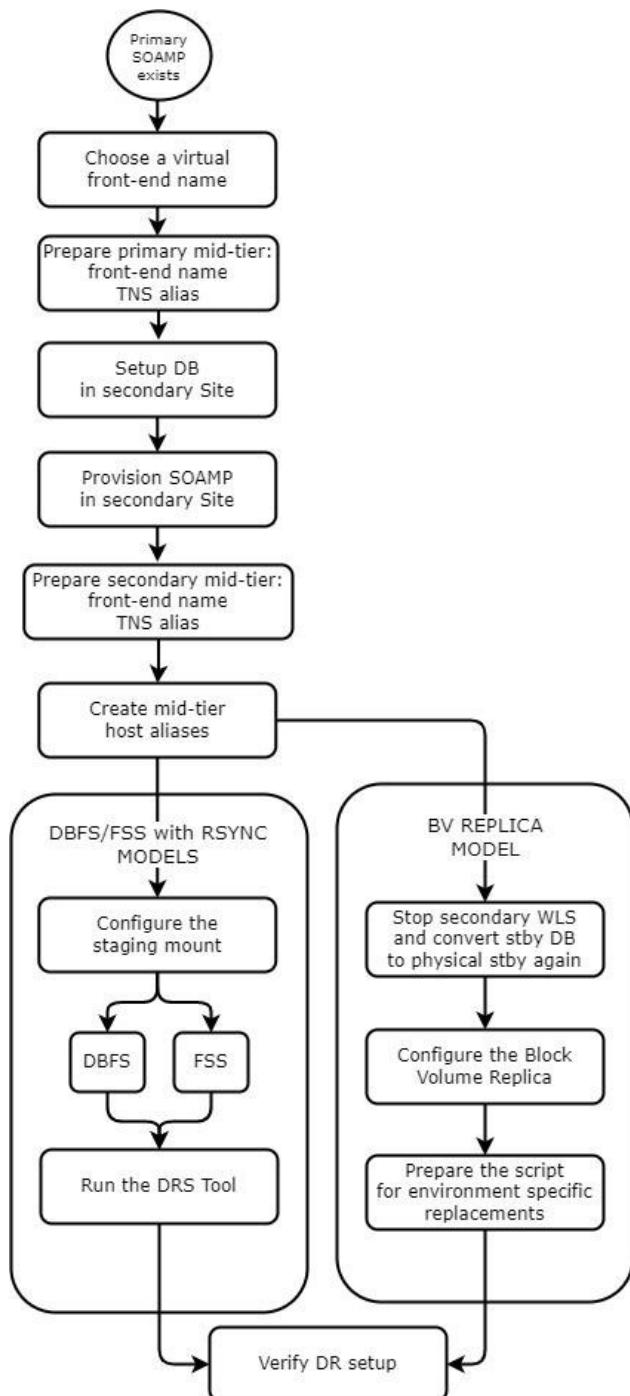


図7：本書に記載したすべてのモデルのDRセットアップ手順を示すフロー・チャート

ディザスタ・リカバリのセットアップの準備

注：以下の手順は、すべてのレプリケーション方法（DBFS、FSS/rsync、リージョン間ブロック・ボリューム・レプリケーション）に適用されます

1. 仮想フロントエンド名を選択する

MarketplaceでSOA Suiteインスタンスが作成されると、プロビジョニングされているロードバランサが特定のフロントエンドIPアドレスをリスニングします。フロントエンドの完全修飾ドメイン名（FQDN）がシステムで入力されることや構成されることはありません。プライマリ・サイトのLBRが1つのフロントエンドIPアドレスをリスニングし、セカンダリ・サイトのLBRが別のフロントエンドIPアドレスをリスニングします。各WLSクラスタの“Frontend Host”プロパティが、対応するロードバランサのフロントエンドIPアドレスとともにデフォルトでプロビジョニングされます。

ディザスタ・リカバリ・トポロジでは、クライアントは、“Oracle Cloudリージョンやデータセンター”に**依存しない**フロントエンドFQDN（通常、“**仮想**”フロントエンド名または“バニティURL”と呼ばれる）を使用して、システムにアクセスする必要があります。この仮想フロントエンド名は、その時点でアクティブなプライマリ・サイトのLBRのIPアドレスに解決されます。システムの**仮想フロントエンド名**はDNSドメイン内（“soampdrs.mycompany.com”など）のものを選択し、**外部から解決できるように**する必要があります。プライマリ・サイトにアクセスするための仮想フロントエンド名がすでに構成されている場合は、これをDR構成にも再利用できます。

この仮想フロントエンド名を外部から解決するには、その名前を任意の公式パブリックDNSに登録する必要があります（あるいは、その名前をクライアントのローカルのホスト・ファイルに追加することもできます）。WebLogicホストのスコープ内で仮想フロントエンド名をローカル解決するには、次の説明に従って、DRSツールの実行前にシステムのホスト・ファイルを手動で構成する必要があります。

システム内のLBRのパブリックIPアドレスを確認するには、OCIコンソールにログインし、適切なリージョンとコンパートメントを選択し、「Load Balancers」セクションに移動し、使用中のLBRをクリックして、そのLBRがリスニングしているパブリックIPアドレスを調べます。

2. プライマリ中間層での仮想フロントエンドの準備

プライマリ中間層をDR構成のために準備するには、プライマリ中間層でいくつかのアクションを実行します。

a) すべてのプライマリ中間層ホストで、/etc/hostsファイルに仮想フロントエンド名とIPアドレスを追加します。

DNS経由でのクライアント向けの解決とは関係なく、各中間層ホストで常にフロントエンド名をローカルLBRへと解決する必要があります。rootユーザーで/etc/hostsファイルを編集し、プライマリLBRのパブリックIPを仮想フロントエンドFQDNにマッピングします。すべてのプライマリ中間層ホストでこの処理を繰り返します。以下に例を示します。

```
[oracle@soampdrs-soa-0 ~]$ more /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
...
# 仮想フロントエンド名
111.111.111.111 soampdrs.mycompany.com
```

注：プライマリ中間層ホストの/etc/hostsファイルを、スイッチオーバーまたはフェイルオーバーの際に変更しないでください。プライマリ・ホストは常に、仮想フロントエンド名を自らのフロントエンドのIPアドレスへと解決します。スイッチオーバーおよびフェイルオーバー手順の実行中に必要となるDNS更新は、DNSまたはSOAクライアントが使用するホスト・ファイル内で実行されます。

b) フロントエンド名をクラスタ・フロントエンドとして構成します。自分のインスタンスのWebLogicコンソールにログインし、次の手順を実行します。

- 「Environment」→「Clusters」に移動し、クラスタを選択します。
- 「Configuration」→「HTTP」に移動します。
- Frontend Hostを仮想フロントエンドのFQDN（“soampdrs.mycompany.com”など）に設定します。
- 保存してアクティブ化します。
- この変更を有効にするには、クラスタを再起動する必要があります。

This page allows you to define the HTTP settings for this cluster. These settings can be overridden by explicitly setting the member servers of this cluster.

Frontend Host:	<input type="text" value="soampdrs.mycompany.ca"/>
Frontend HTTP Port:	<input type="text" value="80"/>
Frontend HTTPS Port:	<input type="text" value="443"/>

Save

図8：クラスタのフロントエンド・ホストの構成

c) **t3/rmi URLをクラスタ構文で更新します（使用する場合）。**

WebLogicクラスタでRMIの呼出しに使用するURLは、各サイトで使用するIPやホスト名に依存しないようにする必要があります。JNDIは、host:port, host:portといったURL構文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ます。クラスタ構文は、cluster:t3://cluster_nameです⁷。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. SOA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」→「Outbound Connection Pools」タブをクリックします。
- e. *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. SOAノード1のデプロイメント・プランを、SOAノード2のまったく同じディレクトリ/場所にコピーします。または、SOAシステムにあるデフォルトの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場所として使用します（SOAクラスタ内のすべてのノードから/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. 「Lock & Edit」をクリックします。
- n. 「Deployments」をクリックし、「JMS Adapter」を選択して「Update」をクリックします。
- o. “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。
- p. 「Finish」をクリックし、変更を有効化します。

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、スイッチオーバー/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が有効な状態にしておく必要があります。

⁷ t3/RMI URLでクラスタ名構文を使用できるのは、**ドメイン内**呼出しの場合のみです。SOAドメイン外のt3/rmiクライアントではこの方法を使用できないため、セカンダリ・サイトに切り替えるときは、host:portリストの適切なDNSマッピングを使用する必要があります。JNDI InitialContextの取得にTCPロードバランサを使用することはできますが、JMSクライアントからの後続リクエストは各host:portに直接接続するため、この場合もセカンダリ・サイトのホストIPへのDNSマッピングが必要になります。

3. プライマリ中間層でのTNSエイリアスの使用の準備

JDBC URLでTNSエイリアスを使用することで、プライマリからスタンバイへのWebLogic構成のレプリカが容易になります。システムがデータソースとJPS構成ファイルでこのアプローチをまだ使用していない場合は、次の手順に従って設定してください。

注：

2023年2月以降に作成されたSOAMPインスタンスでは、データソースとjps-configファイル内のTNSエイリアスが標準で使用されます。これに当てはまる場合は、以下の部分をスキップして次の手順「セカンダリ・サイトのデータベースをセットアップする」に進んでください。

- a) **RACデータベースを使用している場合は、スクリプトfmw_change_to_tns_alias.shを使用して自動的に変更を実行します。**
このスクリプトは、データソースがロング形式の接続文字列（GridLinkデータソースで予測されるものと同様）を使用していることを前提としています。このスクリプトは、既存のTNS文字列をデータソースから取得し、tnsnames.oraを含むtns_adminフォルダを作成して、TNSエントリをtnsnames.oraに書き込みます。これにより、接続文字列がおののデータソースとjps構成ファイルのエイリアスに置き換わります。また、tns_adminプロパティがこれらの各ファイルに追加され、tns_adminフォルダを参照します。

- b) **シングル・インスタンス・データベースを使用している場合は、以下のように手動で構成を行います。**

- すべての中間層ホストにTNSフォルダを作成する

\$DOMAIN_HOME/config/tnsadminフォルダを作成します。

このフォルダは、プライマリからセカンダリへ実行される構成のコピーから除外されます。

- TNSフォルダにtnsnames.oraファイルを作成する

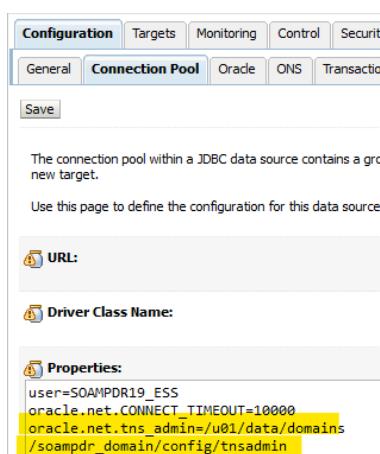
このファイルには、データソースで使用されるデータベースのTNSエントリが含まれる必要があります。例：

```
PDB1 = (DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=mydba-scan.
dbsubnet.vcnash226.oraclevcn.com)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=PDB1.dbsubnet.vcnas
h226.oraclev cn.com)))
```

- WLS管理コンソールを使用してデータソースのURLにTNSエイリアスを設定する

各データソースで2か所変更する必要があります。

- 各データソースにプロパティ“oracle.net.tns_admin”を追加し、「Configuration」→「Connection Pool」→「Properties」を選択します。例：
oracle.net.tns_admin=/u01/data/domains/wlsociprefix_domain/config/tnsadmin



- URLをjdbc:oracle:thin:@tnsaliasの形式に変更します。例：jdbc:oracle:thin:@PDB1
- 保存してアクティブ化します。

- JPS構成ファイルを編集してTNSエイリアスを設定する

WLS管理コンソールを使用してJPS構成ファイルを変更することはできません。

- \$DOMAIN_HOME/config/fmwconfigで、jps-config.xmlファイルおよびjps-config-jse.xmlファイルを編集します。
- <property name="jdbc.url">を見つけます。
- プロパティ値をエイリアス形式の接続文字列に変更します。例：
<property name="jdbc.url" value="jdbc:oracle:thin:@PDB1"/>

- この行の前に、TNS管理フォルダを参照するプロパティ`oracle.net.tns_admin`を追加します。例：
`<property name="oracle.net.tns_admin" value="/u01/data/domains/soampdr_domain/config/tnsadmin"/>`
- c) データソースで複数のデータベースを使用している場合は、必ずデータソースのすべてのデータベースに適切なエイリアスがあり、`tnsnames.ora`ファイルにTNSエントリが含まれているようにしてください。
- d) 以下のようにサーバーを再起動して変更を有効にします。
 - すべての（管理および管理対象）サーバーを停止する
 - 最初に管理サーバーを起動する
 - 管理対象サーバーを起動する

4. セカンダリ・サイトのデータベースをセットアップする

セカンダリ・サイトのデータベースを、**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します。これには2つの方法があり、1つはOCIコンソールを使用してData Guardを有効化する方法です（本書では“自動Data Guard”と呼びます）。もう1つは、`dgmgrl`コマンドを使用して、手動でスタンバイ・データベースを作成し、構成する方法です（本書では“手動Data Guard”と呼びます）。

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。この方法はOCIコンソールのユーザー・インターフェースと統合されているので、コンソールを使用して、Oracle Data Guardを管理できます。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](#)」に従ってください。

何らかの理由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（各DBシステムのフレーバー/エディションでData Guardのリージョン全域の機能の可用性を確認する方法については、DBシステムのドキュメントを参照してください）でも、本書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。その場合は、「[オプション2）手動でのData Guardの構成](#)」の手順に従ってください。

Oracle Exadata Database Serviceのデータベースを使用する場合は、Oracle Exadata Cloud Serviceドキュメント「[Exadata Cloud InfrastructureでのOracle Data Guardの使用](#)」の説明に従ってセカンダリ・データベースをセットアップし、「[4.4 EXACSに関する考慮事項](#)」に続けます。

4.1 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、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システムが自動的にプロビジョニングされ、フィジカル・スタンバイとして構成されます**。これにはいくつかの要件があります。たとえば、両方のDBシステムが同じコンパートメントに含まれる必要がある、両方のDBシステムが同じシェイプ・タイプでなければならない、複数のDBシステムが異なるリージョンに配置される場合はリモートVCNピアリングを介して接続する必要がある、などです。これらの要件について詳しくは、Oracle Cloud Infrastructureのドキュメントの「[Oracle DBシステムでの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 Associations」セクションでData Guardを有効化します。セカンダリDBシステムの構成プロパティの大半（バージョン、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は指定する必要があります。次の表に、指定が必要なプロパティの例と要件を示します。

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	セカンダリDBシステム/例	自動Data Guardのための要件
Oracle Cloudテナンサー	XXXX/paasmaa	YYYY/paasmaa	必ず同じ値
コンパートメント	XXXX/soadr	XXXX/soadr	必ず同じ値
リージョン	XXXX/Ashburn	YYYY/Phoenix	必ず異なる値（DRの場合は異なるリージョンが推奨）
可用性ドメイン	XXXX / efEXT:US-ASBURN-AD1	YYYY / efEXT:PHX-AD-1	必ず異なる値
DBシステム名	XXXX/drdba	YYYY/drdbb	必ず異なる値

シェイプ	XXXX / VM.Standard2.1	XXXX / VM.Standard2.1	必ず同一であること
仮想クラウド・ネットワーク	XXXX/soavcn1ash	YYYY/soavcn1pho	必ず異なる値（異なるリージョンが期待され、リモートVCNピアリングを介して接続）
クライアント・サブネット	XXXX/ashsubnet1	XXXX/phosubnet1	必ず異なる値（異なるリージョンが期待され、リモートVCNピアリングを介して接続）
ホスト名接頭辞	XXXX/drdba	YYYY/drdbb	必ず異なる値
管理者パスワード	XXXX/password	XXXX/password	必ず同じ値

4.2 オプション2) 手動でのData Guardの構成

このアプローチは、プライマリとスタンバイに同じクラウド・テナントを使用できない場合や、OCIコンソールで提供されるData Guard有効化オプションがDR構成に含まれるDBフレーバー/ロケーションに対応していない場合のみ使用します。この場合、セカンダリ・データベースを通常のDBシステムとしてプロビジョニングしてから、Data Guardを手動で構成する必要があります。この手動構成では、本書で提供するData Guardセットアップ・スクリプトを使用できます。次にその手順を説明します。

4.2.1 セカンダリ・データベースのプロビジョニング

注：Data GuardがOCコンソールを使用して有効化されている場合は、これらの手順をスキップする必要があります。

4.3 「[DBシステムでのカスタムPDBサービスの構成](#)」から続行できます。

Data Guardを手動で構成するには、はじめにセカンダリ・データベースをプロビジョニングする必要があります。このとき、データベース名、PDB名、リリース、パッチ・レベル、ノード数、エディションは、プライマリで使用されているのと同じものを使用します。場合によっては、そのために、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必要があります（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。[クラウドDBシステムのドキュメント](#)に記載された手順に従って、スタンバイ・データセンターに必要なデータベース・システムをプロビジョニングします。

次の表に、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示します。

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 / 値の例	セカンダリDBシステム / 値の例	手動DGの要件
Oracle Cloudテナント	XXXX/paasmaa	YYYY / paasmaa	異なってもよい
コンパートメント	XXXX/soadr	YYYY/soadr	異なる値でも可
リージョン	XXXX/Ashburn	YYYY/Phoenix	必ず異なる値（DRの場合は異なるリージョンが推奨）
可用性ドメイン	XXXX / efEXT:US-ASBURN-AD1	YYYY / efXT:PHX-AD-1	必ず異なる値
DBシステム名	XXXX/drdba	YYYY/drdbb	必ず異なる値
シェイプ	XXXX / VM.Standard2.1	XXXX / VM.Standard2.1	必ず同一であること

合計ノード数	N/1	N/1	必ず同一であること
Oracle Databaseのソフトウェア・エディション	EE、EE-HP、またはEE-EP/EE-EP	EE、EE-HP、またはEE-EP/EE-EP	必ず同じ値
使用可能なストレージ	XXXX/256	XXXX/256	必ず同じ値
ライセンス・タイプ	LI、BYOL/BYOL	LI、BYOL/BYOL	異なってもよい
SSH公開鍵	XXXX	YYYY	必ず同一であること
仮想クラウド・ネットワーク	XXXX/soavcn1ash	YYYY/soavcn1pho	必ず異なる値（異なるリージョンが期待され、リモートVCNピアリングを介した接続が推奨）
クライアント・サブネット	XXXX/ashsubnet1	XXXX/phosubnet1	必ず異なる値
ホスト名接頭辞	XXXX/drdba	YYYY/drdbb	必ず異なる値
データベース名	XXXX/ORCL	XXXX/ORCL	必ず同一であること
データベース・バージョン	XXXX/19c	XXXX/19c	必ず同一であること
PDB名	XXXX/PDB1	XXXX/PDB1	必ず同一であること
管理者パスワード	XXXX/password	XXXX/password	必ず同一であること
自動バックアップの有効化	X/チェックを入れる	Y/チェックを外す	スタンバイでは無効にすることが必須 スタンバイ・データベースからバックアップを実行するには、「 Data Guardアソシエーションのスタンバイ・データベースからのバックアップとリストア 」を確認してください。

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。

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、Data Guard構成には、12cバージョンのバグ22611167に対する修正が必要です。プライマリ/セカンダリの両方のDBシステムにパッチが適用されているかどうかを確認し、まだの場合は適用します。最新のOCI 12cR2 DBシステムには、このバグ用のパッチが事前インストールされています。

4.2.2 プライマリ/セカンダリ間のData Guardの構成

注：Data GuardがOCコンソールを使用して有効化されている場合は、これらの手順をスキップし、「[4.3 DBシステムでのカスタムPDBサービスの構成](#)」から続行します。

プライマリ/セカンダリのデータベース間に手動でData Guardを構成するには、以下の手順に従います。

- Data Guardでは、プライマリ・データベースとセカンダリ・データベースがリスナー・ポートで相互通信する必要があります。また、各データベースが適切なリスナー・ポートでそれ自体のIPに到達できる必要があります。このような接続を可能にするために適切な受信ルールが各VCN（プライマリとスタンバイ）に定義されていることを確認します。

Oracle RACデータベースの場合、スキャンIPおよびVIP IPのアドレスにサイト間で到達できなければならないため、プライマリとスタンバイのRACが動的ルーティング・ゲートウェイ経由で通信することが要件になります。

ncコマンドを使用して通信を確認します（使用するネットワーク・トポロジに応じて、パブリック/プライベートIPを使用）。たとえば、次のとおりです。

```
[opc@drdBa ~]$ nc -vw 5 -z <secondary_db_ip> 1521
```

正しい出力例：

```
[opc@drdbb ~]$ nc -vw 5 -z 10.0.0.79
1521 Ncat:Version 7.50
( https://nmap.org/ncat )
Ncat:Connected to 10.0.0.79:1521.
Ncat:0 bytes sent, 0 bytes received in 0.07 seconds.
```

注：パブリックDBシステムのIPを使用するのは、プライマリ・サイトとセカンダリ・サイトがインターネット・ゲートウェイを使用して相互通信している場合のみです。プライマリ/セカンダリVCN間で動的ルーティング・ゲートウェイを使用して内部で通信している場合は、内部DBシステムのIPを使用します（このアプローチが推奨されています）。

- b) 一連のData Guardの手動構成用スクリプト⁸をダウンロードし、ソリューション・プレイブック障害時リカバリのためのスタンバイ・データベースの構成の説明に従ってください。このスクリプトは、シングル・インスタンス・データベースおよびOracle RACシナリオでData Guardを構成するために使用できます。
- c) Data Guardのセットアップが完了したら、プライマリ・システムからData Guard Broker CLIに移動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
Configuration -
ORCL_lhr2bb_ORCL_fra22g Protection
Mode:MaxPerformance Members:
    ORCL_lhr2bb - Primary database
ORCL_fra22g - Physical standby
database Fast-Start Failover:Disabled
Configuration Status:
SUCCESS (status updated 33 seconds ago)
```

4.3 DBシステムでのカスタムPDBサービスの構成

DBシステムがReal Application Cluster (RAC) データベースの場合、SOA MarketplacesはGridLinkデータソースを使用して選択されたOracle Databaseに接続します。GridLinkは、Oracle Database RACの全ノードで動的なロードバランシングとフェイルオーバーを実現するほか、ノードがクラスタから削除またはクラスタに追加された場合にデータベースから通知を受け取ります。これらの機能を完全に活用するために、Cluster Ready Services (CRS) とOracle Notification Service (ONS) に対応したOracle Databaseサービスを作成することが推奨されます。これらのサービスは、データベース・クラスタ内のリソース・ステータスを監視し、ステータス変更時に通知を生成します。

Oracle Base Databaseサービス・システムにおいては、PDB用のCRSアプリケーション・サービスが標準では作成されません。PDBのデフォルトのサービスは、<dbname>.<subnetdomain>.<vcndomain>.oraclevcn.comなどの名前を持つものだけです。これは、CRSマネージド・サービスでありません。CRSサービスを使用してPDBに接続するには、カスタム・サービスを手動で作成する必要があります。これは、プライマリ・データベースとスタンバイ・データベースで作成する必要があります。

- a) **プライマリのPDB用のOracle Databaseカスタム・サービスを作成します。**

プライマリOracle RACデータベースのDBノードに接続し、oracleユーザーとして以下を実行して、サービスを追加、構成、起動します。

⁸ https://github.com/oracle-samples/maa/raw/main/dg_setup_scripts/dg_setup_scripts.zip

```

srvctl add service -db <PRIM_DB_UNIQUE_NAME> -service <NEW_SERVICE_NAME> -preferred
<INSTANCE_NAME1>,<INSTANCE_NAME2> -pdb <PDB_NAME> -role "PRIMARY,SNAPSHOT_STANDBY"
srvctl modify service -db <DB_UNIQUE_NAME> -service <NEW_SERVICE_NAME> -rlbgoal SERVICE_TIME -clbgoal SHORT -pdb
<PDB_NAME>
srvctl start service -db <DB_UNIQUE_NAME> -service <NEW_SERVICE_NAME>

```

サービスを作成するとき、“-role”パラメータに2つのロールを指定することが重要です。こうすることで、データベースのロールがプライマリであっても、スナップショット・スタンバイであっても、サービスが自動的に開始されます。

以下に例を示します。

```

[oracle@piracnode1 ~]$ srvctl add service -db ORCL_lhr3jg -service mypbservice.example.com -preferred
ORCL1,ORCL2 -pdb pdb1 -role "PRIMARY,SNAPSHOT_STANDBY"
[oracle@piracnode1 ~]$ srvctl modify service -db ORCL_lhr3jg -service mypbservice.example.com -rlbgoal
SERVICE_TIME -clbgoal SHORT -pdb pdb1
[oracle@piracnode1 ~]$ srvctl start service -db ORCL_lhr3jg -service mypbservice.example.com
[oracle@piracnode1 ~]$ srvctl config service -db ORCL_lhr3jg -service mypbservice.example.com
Service name: mypbservice.example.com
Server pool:
Cardinality:2
Service role:PRIMARY,SNAPSHOT_STANDBY
Management policy:AUTOMATIC
...
Available
instances:CSS
critical: no

```

b) セカンダリOracle RACデータベースに同じOracle Databaseサービスを作成します。

セカンダリOracle RACデータベースのDBノードに接続し、oracleユーザーとして以下を実行して、サービスを作成、構成します。

```

srvctl add service -db <SECONDARY_DB_UNIQUE_NAME> -service <NEW_SERVICE_NAME> -preferred
<INSTANCE_NAME1>,<INSTANCE_NAME2> -pdb <PDB_NAME> -role "PRIMARY,SNAPSHOT_STANDBY"
srvctl modify service -db <DB_UNIQUE_NAME> -service <NEW_SERVICE_NAME> -rlbgoal SERVICE_TIME -clbgoal
SHORT -pdb <PDB_NAME>

```

以下に例を示します。

```

[oracle@secracnode ~]$ srvctl add service -db ORCL_fra3vb -service mydbservice.example.com -preferred
ORCL1,ORCL2 -pdb pdb1 -role "PRIMARY,SNAPSHOT_STANDBY"
[oracle@secracnode1 ~]$ srvctl config service -db ORCL_fra3vb -service mydbservice.example.com
Service name: mydbservice.example.com
Server pool:
Cardinality:2
Service role:PRIMARY,SNAPSHOT_STANDBY
Management policy:AUTOMATIC
DTP transaction: false
...

```

c) 新しいPDBサービスを表示させます。

プライマリOracle RACのDBノードに接続し、oracleユーザーとして以下を実行します。

```

[oracle@piracnode1 ~]$ srvctl status service -db $ORACLE_UNQNAME
Service mypbservice.example.com is running on instance(s)
ORCL1,ORCL2

```

サービスはプライマリOracle RACで実行されている必要があります。

セカンダリOracle RACのDBノードに接続し、oracleユーザーとして以下を実行します。

```
[oracle@secrecrnode ~]$ srvctl status service -db $ORACLE_UNQNAME  
Service mypbservice.example.com is not running.
```

スタンバイOracle RACのサービスは停止させることができます。

d) **プライマリSOAMPの接続値を更新します。**

セカンダリSOAMPを後でプロビジョニングする場合は、プロビジョニング・ウィザードの“PDB Service Name”プロパティに新しいサービス名を入力します。プロビジョニングでは、PDBサービス名の最初の単語（例：“mypbservice”）をデータソースとjpsファイルのTNSエイリアスとして使用し、入力したPDBサービス名（例：“mypbservice.example.com”）をDOMAIN/config/tnsadmin/tnsnames.oraのtnsエントリのSERVICE_NAMEとして使用します。

プライマリおよびセカンダリSOAMPシステムは、データソースおよびjps構成ファイルの接続文字列で同じTNSエイリアス値を使用する必要があります。デフォルトのPDBサービスを使用してプライマリSOAMPがプロビジョニングされた場合、プライマリSOAMPの値を調整して将来のセカンダリSOAMPと一致するようにする必要があります。

データソースとjps構成ファイルで使用されるTNSエイリアスは、PDB名ではなく、新しく作成されたPDBサービス名の最初の単語である必要があります。例：

```
jdbc:oracle:thin:@mypbservice
```

- \$DOMAIN/config/tnsadmin/tnsnames.oraのTNSエントリは、新しいサービス名を含むTNSエイリアスでなければなりません。以下に例を示します。

```
mypbservice = (DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=mydba-scan.  
dbsubnet.vcnash226.oraclevcn.com)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=mypbservice.e  
xa mple.com)))
```

プライマリSOAMPがすでにカスタムPDBサービスを使用してプロビジョニングされている場合は、プライマリSOAMPに変更を加えないでください。

4.4 EXACSに関する考慮事項

Oracle Exadata Database Serviceを使用する場合は、以下の考慮事項に従ってください。

- Oracle Exadata Database Serviceドキュメント「[Exadata Cloud InfrastructureでのOracle Data Guardの使用](#)」の説明に従って、Data Guardをセットアップします。
- Oracle Exadata Database Serviceを使用してSOAMPをプロビジョニングする場合、使用するPDBサービス名を入力パラメータとして入力できます。SOAMPでは、そのサービスを使用してPDBへ接続することになります。Oracle Exadata Database Serviceのデータベースは標準でCRSデータベース・サービスを使用します。CDBとPDBに対して構成済みのサービスがいくつかあり、srvctlコマンドでそのリストを表示できます。プライマリ・データベースにはPDBサービスがあらかじめ構成されており、その名前は、<DBNAME_PDBNAME>.paas.oracle.comです。以下に例を示します。

```
[oracle@primary-exadb-host1 ~]$ srvctl status service -db $ORACLE_UNQNAME  
Service exadb_dg is running on instance(s) EXADB1,EXADB2  
Service exadb_dg_ro is not running.
```

Service exadb_pdb1.paas.oracle.com is running on instance(s) EXADB1,EXADB2 → this is primary PDB service
Service pdb1_dg is running on instance(s) EXADB1,EXADB2
Service pdb1_dg_ro is not running.

ただし、このサービスはスタンバイ・データベースには構成されていません。デフォルトでスタンバイ・データベースに構成されているのは、“dg”サービスと“read only”サービスのみです。

例：

```
[oracle@standby-exadb-host1 ~]$ srvctl status service -db $ORACLE_UNQNAME  
Service exadb_dg is not running.  
Service exadb_dg_ro is not running.  
Service pdb1_dg is not running.  
Service pdb1_dg_ro is not running.
```


- スタンバイデータベースで、PDBサービスを作成します。一貫性を確保するため、プライマリSOAMPからプライマリPDBに接続する際に使用するのと同じPDBサービス名を使用します。以下に例を示します。

```
[oracle@standby-exadb-host1 ~]$ srvctl add service -db $ORACLE_UNQNAME -service exadb_pdb1.paas.oracle.com -preferred EXADB_PHO1,EXADB_PHO2 -pdb PDB1 -role "PRIMARY,SNAPSHOT_STANDBY"
```

```
[oracle@standby-exadb-host1 ~]$ srvctl modify service -db $ORACLE_UNQNAME -service exadb_pdb1.paas.oracle.com -rlbgoal SERVICE_TIME -clbgoal SHORT -pdb PDB1
```

```
[oracle@standby-exadb-host1 ~]$ srvctl start service -db $ORACLE_UNQNAME -service exadb_pdb1.paas.oracle.com
```

サービスを作成するとき、“-role”パラメータに2つのロールを指定することが重要です。こうすることで、データベースのロールがプライマリであっても、スナップショット・スタンバイ（セットアップ時とライフサイクル運用で必要）であっても、サービスが自動的に開始されます。プライマリのPDBサービスには通常、“primary”ロールのみが構成されているため、これを変更してスタンバイ・ロールを追加します。

```
[oracle@primary-exadb-host1 ~]$ srvctl modify service -db $ORACLE_UNQNAME -service exadb_pdb1.paas.oracle.com -role "PRIMARY,SNAPSHOT_STANDBY"
```

- プライマリSOAMPの接続値を更新します。

プライマリSOAMPシステムがCRSサービスに接続されていることを確認します。データソースおよびjps構成ファイルによって使用される\$DOMAIN/config/tnsadmin/tnsnames.oraのエントリは、デフォルトのPDBサービスではなく、CRSサービスをポイントする必要があります。例：

```
mypdb = (DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=mydba-scan.dbsubnet.vcnash226.oraclevcn.com)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=exadb_pdb1.paas.oracle.com)))
```

プライマリ・データベース・ホストおよびスタンバイ・データベース・ホストのoracleユーザーのプロファイルに、環境変数ファイルをあらかじめロードします。

データベース・ホストでデータベース・コマンドを実行するために必要なORACLE_HOME、ORACLE_UNQNAME、PATHなどの変数は、/home/oracle/<DBNAME>.envという名前のファイル内に設定されます。ただし、デフォルトでは、このファイルはoracleユーザーのプロファイルにあらかじめロードされていません。このファイルを.bashrcから呼び出して、oracleユーザーのプロファイルに追加します。例：

```
[oracle@exadlus1pho-2t2id1 ~]$ more .bashrc
# .bashrc
...
# ユーザー固有のエイリアスと関数
./home/oracle/EXADB.env
```

これが、後でDRSツールを実行するための要件です。DRSツールは、（Dataguardステータスのチェックやロール変換の実行などのために）プライマリおよびスタンバイ・データベース・ホスト上でいくつかのコマンドを実行する必要があり、これにはデータベース環境変数がすでにoracleユーザーのプロファイルにロードされている必要があります。

5. セカンダリ・サイトでのSOA Suite on Marketplaceのプロビジョニング

セカンダリSOAシステムは、スナップショット・スタンバイ・モードで開いている必要があるセカンダリDBシステムを参照して作成されます。

このため、セカンダリSOA Suite on OCI Marketplaceをプロビジョニングする前に、**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**します。これにより、スタンバイ・データベースで、プライマリから取得された変更の適用が停止され、読み取り/書き込みモードで開かれます。これにより、セカンダリSOAが作成されます。このため、**プライマリDBホストでoracleユーザーとして以下のコマンドを実行します。**

```
[oracle@drdba ~]$dgmgrl sys/your_sys_password@primary_db_unqname
DGMGR> 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
```

前の手順でスナップショットに変換されたセカンダリDBシステムを参照してセカンダリ・サイトのSOAシステムを作成するには、[SOA Suite on OCI Marketplaceのドキュメント](#)の手順に従います。

Stack Nameは異なっていても構いませんが、プライマリ・ロケーションで使用したものとまったく同じインスタンス名接頭辞を使用する必要があります。オラクルでは、理想的なフェイルオーバーやスイッチオーバーの動作のため、プライマリとスタンバイの両方のロケーションでまったく同じ容量とコンピュート構成を使用することを推奨しています。

また、セカンダリ・ロケーションにプロビジョニングされるセカンダリSOAMP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。SOAMPのプロビジョニング・メニューでは、最新から8つのバージョンが提供されます。つまり、お客様は同じSOAMPバージョンを12～24か月の時間枠でプロビジョニングできます。セカンダリのプロビジョニング時に、プライマリのSOAMPバージョンをSOAMPのプロビジョニング・メニューで使用できない場合は、新たにプロビジョニングされたセカンダリ・サイトと同じレベルまで、プライマリSOAMPにパッチを適用する必要があります。それぞれのSOAMPバージョンでプロビジョニングされるパッチ・レベルについては、「[What's New in Oracle SOA Suite on Marketplace](#)」で確認してください。

注：

プライマリ・システムとスタンバイ・システムで同じバージョンのオペレーティング・システムを使用する必要があります。SOAMPは、バージョン24.2.2よりも前では、*Oracle Linux 7*のコンピュート・インスタンスを使用しています。SOAMP 24.2.2以降は、*Oracle Linux 8*を使用しています。

プライマリSOAMPシステムが*OEL 7*を使用している場合は、24.2.2よりも前のSOAMPバージョンを使用してセカンダリをプロビジョニングする必要があります。

注：

プライマリSOAMPインスタンスが2023年6月よりも前に（SOAMPリリース23.2.2よりも前に）作成されており、セカンダリが2023年6月以降（リリース23.2.2以降）に作成された場合は、サポートに問い合わせてください。DRセットアップを実行する前に、セカンダリのバイナリ・パスで調整が必要な場合があります。

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。

SOA Suite on Marketplaceのプロパティ	プライマリでの値/例	セカンダリでの値/例	DRの要件
リージョン	XXXX/Ashburn	YYYY / Phoenix	必ず異なる値
バージョン	XXXX/12.2.1.4	XXXX/12.2.1.4	必ず同じ値
スタック名	XXXX/soampdrsPrim	YYYY/soampdrsStby	異なってもよい
インスタンス名接頭辞	XXXX/soampdrs	XXXX/soampdrs	必ず同じ値
サービス・タイプ	XXXX/SB & B2Bクラスタを使用するSOA	XXXX/SB & B2Bクラスタを使用するSOA	必ず同じ値。 本書は、“SB & B2Bクラスタを使用する SOA”サービス・タイプおよび“MFTクラスタ”サービス・タイプをサポートしています
コンピュート・シェイプ	XXXX / VM.Standard2.1	XXXX / VM.Standard2.1	必ず同じ値
SSH公開鍵	XXXX / my_public_key.pub	XXXX / my_public_key.pub	必ず同じ値
クラスタ・ノード数	N/2	N / 2	必ず同じ値
管理ユーザー名	XXXX/weblogic	XXXX / weblogic	必ず同じ値
管理者のパスワード	XXXX/password	XXXX/password	必ず同じパスワード (KMSで暗号化されている場合、暗号化された値は異なる場合があります)
ネットワーク・コンパートメント	XXXX/soadr	YYYY/soadr	異なる値でも可（ただし通常は同じ値）
VCN	XXXX/soadrvcn1ash	YYYY/soadrvcn1pho	必ず異なる値
サブネット	XXXX/ashsubnet1	YYYY/phosubnet1	必ず異なる値
ロードバランサのプロビジョニング	必ずオン	必ずオン	両方ともオンにする
データベース戦略	XXX/データベース・システム	XXX/データベース・システム	必ず同じ値
DBシステム	XXXX/drdba	YYYY/drdbb	必ず異なる値
DBシステム内のデータベース	XXXX/ORCL	XXXX/ORCL	必ず同じ値
PDBサービス名	XXXX/PDB1 または XXXX/mypdbservice.example.com	XXXX/PDB1 または XXXX/mypdbservice.example.com	必ず同じ値* PDB名（その場合はデフォルトのPDBサービスを使用して接続されます）、または完全なPDBサービス名を指定します。 ベスト・プラクティスとしては、デフォルトのPDBサービスではないPDBサービス名を指定します。PDBサービスには、PRIMARYおよびSNAPSHOT_STANDBYロールが含まれる必要があります。

			*PDBサービス名には、最初の単語が同じ場合にのみ異なる値を指定できます。それは、最初の単語はTNSエイリアス用にSOAMPによって使用される値であり、プライマリとセカンダリとで一致している必要があるためです。
データベース管理者	SYS	SYS	必ず同じ値
データベース管理者のパスワード	XXXX/password	XXXX/password	必ず同じパスワード（ただし、KMSで暗号化されている場合、暗号化された値は異なっていても可）
KMS復号化の使用	X/チェックを外す	X/チェックを外す	異なる値でも可。 KMSはオプションであり、プロビジョニング目的にのみ使用されます。
カスタムRCUスキーマ接頭辞を指定する	チェックを入れる/チェックを外す	チェックを入れる/チェックを外す	カスタム・スキーマを指定することも、プロビジョナにランダムなスキーマ接頭辞を作成させることもできます。
(「カスタムRCUスキーマ接頭辞を指定する」にチェックを入れた場合のみ) カスタム・スキーマ接頭辞	XXXX/PREFIXA	YYYY/PREFIXB	必ず異なる値 チェックを入れてカスタムRCUスキーマ接頭辞を指定することにした場合は、プライマリとは別のRCUスキーマ接頭辞を指定する必要があります。これは、すでにスキーマが存在することによるプロビジョニング問題を回避するためです。 セカンダリ・スキーマは後で破棄されます。DRがセットアップされると、プライマリ・スキーマのみが使用されます。
RCUスキーマのカスタム・パスワードを指定する	チェックを入れる/チェックを外す	チェックを入れる/チェックを外す	同一にできる
("RCUスキーマのカスタム・パスワードを指定する"にチェックを入れた場合のみ) カスタムRCUスキーマ・パスワード	XXXX/password	XXXX/password	同一にできる
高度なサービス・インスタンス(オプションのポート)	XXXX	XXXX	必ず同じ値。 デフォルト値を使用していない場合は、必ずプライマリの場合と同じポートを使用してください。
高度なサービス・インスタンス(オプションのカスタム名および接頭辞)	XXXX	XXXX	この機能を使用してドメインやクラスタなどのカスタム名を構成できますが、入力するカスタム名はプライマリの場合と同じである必要があります。

プロビジョニング時に鍵管理サービスを使用するかどうかは任意です。チェックを入れると、KMSサービスは、SOA Suite on Marketplaceのプロビジョニング時のパスワードの暗号化と復号化でのみ使用されます。実行時またはライフサイクルのタスクでは使用されません。プロビジョニング・ウィザードに入力される暗号化されたパスワードの値は異なる場合がありますが、暗号化前のパスワードは同じである必要があります。

プロビジョニング・プロセスが完了したら、SOA Suiteサーバーが正常に動作するか検証できます。

注：Oracle SOA Suite on Marketplace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SOA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サーバーは、プライマリとは異なるスキーマ名を使用するということです。これでは、セカンダリ・ロケーションにある初期状態のSOAMP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ます。

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（DRがセットアップされた後）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・データベースからスナップショットへの変換や複製が発生する前に、プライマリ・サイトでインスタンスがドレインされて完了していることが重要です。

セカンダリのSOAMPインスタンスをプロビジョニングした後、何らかの理由で、DRセットアップの手順を続行するまでに長い時間が経過している場合は、セカンダリでWebLogic管理サーバーと管理対象サーバーを停止して、もう一度スタンバイ・データベースをフィジカル・スタンバイに変換することができます。このようにすると、スタンバイ・データベースとプライマリの間でREDO適用のギャップが大きくなりません。この後、DRセットアップが完了するまでは、セカンダリ・サイトのWebLogicサーバーを起動しないでください。

DRセットアップの前に、セカンダリ・サーバーがデータベース内で元のセカンダリ・スキーマを探しますが、これらはもう存在しません。スナップショット・データベースに対して実行された変更は、そのデータベースが再びフィジカル・スタンバイに変換されるときに失われるため、これは想定される結果です。この種のシナリオでは、セカンダリのWebLogic管理サーバーおよび管理対象サーバーを停止したままで、DRセットアップ・タスクを続行します。DBFSの方法とFSS/rsyncの方法の次のステップでDRSを実行する際は、WLSサーバーの検証が失敗しないように、--skip_checksフラグを使用する必要があります。

6. セカンダリ中間層での仮想フロントエンドの準備

すべてのセカンダリ中間層ホストの/etc/hostsファイルに、フロントエンド名とIPアドレスを追加します。

rootユーザーで/etc/hostsファイルを編集し、セカンダリLBRのIPを仮想フロントエンド名にマッピングします。すべてのセカンダリ中間層ホストでの処理を繰り返します。例：

```
[oracle@soampdrs-soa-0 ~]$ more /etc/hosts
127.0.0.1 localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

...
# 仮想フロントエンド名
222.222.222.222 soampdrs.mycompany.com
```

セカンダリWebLogicクラスタのフロントエンド・アドレスは、プライマリWebLogicドメイン構成からコピーされるため、WebLogicコンソールでその情報を更新する必要はありません。

注：セカンダリ中間層ホストの/etc/hostsファイルは、スイッチオーバーまたはフェイルオーバー時に変更しないでください。セカンダリ・ホストは常に、仮想フロントエンド名を自らのフロントエンドIPへと解決する必要があります。スイッチオーバーおよびフェイルオーバー手順の実行中に必要となるDNS更新は、DNSまたはクライアントが使用するホスト・ファイル内で実行されます。

7. セカンダリ中間層でのTNSエイリアスの使用の準備

プライマリ・システムと同様、\$DOMAIN_HOME/config/jdbcのすべてのデータソース・ファイル、および\$DOMAIN_HOME/config/fmwconfigのJPS構成ファイルでTNSエイリアスを使用する必要があります。

セカンダリ・システムで使用されるエイリアス名は、**プライマリの場合と同一**である必要があります。セカンダリ・システムのtnsnames.oraファイルには、同一でありながらセカンダリ・データベースを参照するTNSエイリアス・エントリが含まれる必要があります。

セカンダリ中間層でTNSエイリアスを使用する準備をするために、データソースとJPS構成ファイルを変更する必要はありません。DRセットアップの間に、プライマリ・システムからコピーされるからです。プライマリ・システムにあるのと同じパスのセカンダリ中間層ホストにTNS管理フォルダが存在することのみを確認してください。セカンダリのtnsnames.oraにプライマリにあるものと同じエイリアスが含まれており、それがセカンダリPDBを参照していることを確認します。tnsnames.oraに、システムに必要な他のTNSエイリアスも含まれるようにします。

注：

2023年2月以降に作成されたSOAMPインスタンスでは、データソースとjps-configファイル内のTNSエイリアスが標準で使用されます。

8. 必要な中間層ホスト・エイリアスの構成

DRセットアップが完了すると、セカンダリのWebLogicドメイン構成はプライマリWebLogicドメインのコピーとなります。このため、プライマリのWebLogicサーバーによってリスニング・アドレスとして使用されるホスト名（つまり、プライマリ中間層ホストのホスト名）が、セカンダリ・ロケーションで有効であり、かつセカンダリIPにマッピングされている必要があります。

逆も同様に、セカンダリ・サーバーのホスト名がプライマリ・ロケーションで有効であり、プライマリIPにマッピングされている必要があります。WebLogic構成で使用されるのは通常はプライマリ・ホスト名のみであるため、この部分は必須ではありません。この処理を実行するのは、セカンダリ・サイトがプライマリ・ロールになっているときに、セカンダリ名に対する何らかの参照が構成に追加された場合に、プライマリでエラーが生じるのを回避するためです。

必要なホスト名のマッピングを構成するには、/etc/hostsファイルにエイリアスとしてホスト名を追加する方法と、OCIのプライベートDNSビューにエントリを追加する方法の2つがあります。

8.1 オプション1) /etc/hostsファイルの使用

一方のサイトのホスト名をエイリアスとして中間層ホストの/etc/hostsファイルに追加します。

この方法は、プライマリ/セカンダリ・サイトで同じDNSサーバーが使用されている場合と、プライマリ/セカンダリ・サイトで別々のDNSサーバーが使用されている場合のどちらのシナリオでも有効です。なぜなら、/etc/hostsファイルのエンティリがDNS解決よりも優先されるからです。どちらが優先されるのかは、/etc/nsswitch.confファイルの“hosts”ディレクティブ内で定義されています。デフォルトでは“files”が設定されており、この場合、/etc/hostsの解決がDNSよりも優先されます。

この方法の短所は、すべてのSOAホストに手動でエントリを追加する必要があることです。セカンダリでのスケール・アウト操作中に新しいノードを追加した場合、/etc/hostsファイルを変更するまでは新規ノードの名前が解決されません。このため、スケール・アウト操作で追加の手動手順が必要になります。詳しくは、本書の「[DBFSの方法とFSS/rsyncの方法のスケール・アウトとスケール・イン](#)」および「[BVレプリケーションの方法のスケール・アウトとスケール・イン](#)」のセクションを参照してください。

注：DBFSの方法とFSS/rsyncの方法におけるDisaster Recovery Setup（DRS）ユーティリティでは、プライマリおよびセカンダリSOAコンピュート・インスタンスの/etc/hostsで、これらの変更を自動で実行できます。DRSでこの構成を行うかどうかをオプションのフラグで選択できます。DRSを使用して/etc/hostsファイルにエイリアスを作成しようとしている場合は、これらの手順をスキップしてください。

必要なエイリアスを構成する手順は以下のとおりです。

- すべての中間層ホスト（プライマリとスタンバイ）で、rootユーザーとして/etc/oci-hostname.confファイルを編集し、PRESERVE_HOSTINFO=3を設定します。こうすることで、次のステップで/etc/hostsに加える変更が、ノードの再起動後も維持されます。
- サーバーがリスニングするプライマリとセカンダリの各WebLogicホストの**ホスト名を特定します**。ホスト名は、ドメイン構成ファイルのリスニング・アドレスで確認できます。たとえば、プライマリ・ドメインでは以下のとおりです。

```
[oracle@soampdrs-soa-0 config]$ cd $DOMAIN_HOME/config
[oracle@soampdrs-soa-0 config]$ grep listen-address config.xml
<listen-address>soampdrs-soa-0.mysubnet1.myregion1vcn.oraclevcn.com </listen-address>
<listen-address>soampdrs-soa-0.mysubnet1.myregion1vcn.oraclevcn.com </listen-address>
<listen-address>soampdrs-soa-1.mysubnet1.myregion1vcn.oraclevcn.com </listen-address>
...
```

セカンダリ・ドメインでは以下のとおりです。

```
[oracle@soampdrs-soa-0 config]$ cd $DOMAIN_HOME/config
[oracle@soampdrs-soa-0 config]$ grep listen-address
```

```
<listen-address>soampdrs-soa-0.mysubnet2.myregion2vcn.oraclevcn.com </listen-address>
<listen-address>soampdrs-soa-0.mysubnet2.myregion2vcn.oraclevcn.com</listen-address>
<listen-address>soampdrs-soa-1.mysubnet2.myregion2vcn.oraclevcn.com</listen-address>
..
```

各プライマリ中間層ホストでコマンド“**hostname --fqdn**”を実行して、ホスト名を取得することもできます。

- c) すべてのプライマリ中間層ノードで（rootとして）**/etc/hosts**を編集し、スタンバイのホスト名をプライマリ・ホストのエイリアスとして追加します。各ホストでエントリは次のようにになります。

```
<IP_prim_node1> <long_and_short_hostnames_primary_node1> <long_and_short_hostnames_secondary_node1>
<IP_prim_node2> <long_and_short_hostnames_primary_node2>
```

（短縮名は、プライマリとスタンバイで同じになることが予想されます）以下の例は、プライマリ中間層ホストの/etc/hostsの入力結果です。

```
[oracle@soampdrs-soa-0 config]$ more /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6
localhost6.localdomain6 # フロントエンド
111.111.111.111 mysoampdr.mycompany.com

# プライマリ・サイトのSOAMP DRのエイリアス
10.0.2.10 soampdrs-soa-0.mysubnet1.myregion1vcn.oraclevcn.com soampdrs-soa-0 soampdrs-soa-
0.mysubnet2.myregion2vcn.oraclevcn.com
10.0.2.11 soampdrs-soa-1.mysubnet1.myregion1vcn.oraclevcn.com soampdrs-soa-1 soampdrs-soa-
1..mysubnet2.myregion2vcn.oraclevcn.com
```

- d) 同じように、すべてのセカンダリ中間層ノードで（rootとして）**/etc/hosts**を編集し、プライマリのホスト名をセカンダリ・ホストのエイリアスとして追加します。各ホストでエントリは次のようにになります。

```
<IP_secondary_node1> <long_and_short_hostnames_secondary_node1>
<long_and_short_hostnames_prim_node1>
<IP_secondary_node2> <long_and_short_hostnames_secondary_node2>
<long_and_short_hostnames_prim_node2>
```

以下の例は、セカンダリ中間層ホストの/etc/hostsの入力結果です。

```
[oracle@soampdrs-soa-0 ~]$ more /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6
localhost6.localdomain6 # フロントエンド
222.222.222.222 mywsoampdr.mycompany.com

# セカンダリ・サイトのSOAMP DRのエイリアス
10.1.2.5 soampdrs-soa-0.mysubnet2.myregion2vcn.oraclevcn.com soampdrs-soa-0 soampdrs-soa-
0.mysubnet1.myregion1vcn.oraclevcn.com
10.1.2.4 soampdrs-soa -1..mysubnet2.myregion2vcn.oraclevcn.com soampdrs-soa-1 soampdrs-soa-
1..mysubnet1.myregion1vcn.oraclevcn.com
```

8.2 オプション2) OCIのプライベートDNSビューの使用

すべての/etc/hostsファイルにエントリを追加する代わりに、各VCNの**OCIプライベートDNSビュー**に必要なエントリを追加することができます。プライベートDNSビューへのエントリの追加には、スケール・アウト操作で利点があります。セカンダリ・サイトの新規ノードで、追加設定しなくてもプライマリ名を解決できます。このため、スケール・アウト操作が簡単になります。

詳しい手順とterraformスクリプトについては、Githubリポジトリ（https://github.com/oracle-samples/maa/tree/main/private_dns_views_for_dr）を参照してください。

ディザスタ・リカバリのセットアップの完了

ブロック・ボリューム・レプリケーションを使用して構成する

注：以下の手順は、リージョン間ブロック・ボリューム・レプリケーション・モデルのみに適用されます。DBFSまたはFSSのアプローチを使用している場合は、このセクションをスキップし、「[DBFSの方法とFSS/rsyncの方法を使用して構成する](#)」を参照してください。

リージョン間ブロック・ボリューム・レプリケーション・モデルを使用している場合は、次の手順を実行してDRのセットアップを完了します。

1. スタンバイDBのフィジカル・スタンバイへの変換

この時点で、セカンダリのWebLogicサーバーは停止している必要があるため、スタンバイ・データベースをもう一度フィジカル・スタンバイに変換できます。

a) セカンダリ中間層ホストのOracleプロセスを停止する。

すべてのセカンダリ中間層ホストのWebLogic管理対象サーバー、管理サーバー、ノード・マネージャのプロセスを停止します。
また、すでにマウントされている場合は、すべてのセカンダリ中間層ホストで**DBFSをアンマウント**します。

b) スタンバイ・データベースをフィジカル・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@drdbaa ~]$ 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
```

2. リージョン間ブロック・ボリューム・レプリケーションの構成

中間層ホストのブロック・ボリュームをプライマリ・リージョンからセカンダリ・リージョンにレプリケートするには、以下の手順に従います。

a) プライマリ中間層ホストのブロック・ボリュームを特定する

プライマリ・リージョンのブロック・ボリュームを特定するには、次の手順を実行します。

- OCIコンソールに移動し、使用中のプライマリ・リージョンを選択します。
- 「Storage」→「Block Volumes」に移動します。
- プライマリSOA Marketplaceのコンパートメントを選択し、ブロック・ボリュームを探します。
- SOA Marketplaceコンピュート・インスタンスにアタッチされているブロック・ボリュームは<soamp_prefix>-block-Nという名前になっています。Nは0、1、2などで、コンピュート・インスタンスの<soamp_prefix>-soa-0、<soamp_prefix>-soa-1などに対応しています。
- ブロック・ボリュームの名前や配置されているADをメモしておきます。たとえば、次のとおりです。soampdrs-block-0、AD1 soampdrs-block-1、AD2

これらのブロック・ボリュームは、各SOAホストの/u01/dataにマウントされます。例：

[oracle@soampdr s-soa-0 ~]\$	df -h			
Filesystem	Size	Used	Avail	Use% Mounted on
devtmpfs 7.2G	0	7.2G	0%	/dev
tmpfs	7.2G	0	7.2G	0% /dev/shm
tmpfs	7.2G	65M	7.2G	1% /run
tmpfs	7.2G	0	7.2G	0% /sys/fs/cgroup
/dev/sda3	39G	9.0G	30G	24% → これはブート・ボリューム、レプリケート対象ではない
/dev/sda1	200M	8.6M	192M	5% /boot/efi → これはブート・ボリューム、レプリケート対象ではない
tmpfs	1.5G	0.1.	5G	0% /run/user/0
tmpfs	1.5G	0.1.	5G	0% /run/user/994
tmpfs	1.5G	0.1.	5G	0% /run/user/1000
/dev/sdb	49G	1.5G	46G	4% /u01/data このブロック・ボリュームがレプリケート対象となる
dbfs-@ORCL:/	200G	128K	200G	1% /u01/soacs/dbfs_directory
dbfs-@ORCL:/	200G	128K	200G	1% /u01/soacs/dbfs

b) セカンダリ中間層ホストのブロック・ボリュームを特定する

a)の手順を繰り返して、セカンダリ中間層ホストのブロック・ボリュームの名前、可用性ドメインの情報を取得します。

c) プライマリでブロック・ボリューム・グループを作成し、リージョン間レプリケーションを有効化する

プライマリにブロック・ボリューム・グループを作成し、レプリケートされるブロック・ボリュームをすべてグループ化します。レプリケーションはボリューム・グループに対して有効化されるため、そのグループのすべてのブロック・ボリュームに適用されます。ボリューム・グループには同じADにあるブロック・ボリュームのみを含めることができます。複数のADにコンピュート・インスタンスがある場合は、ADごとにブロック・ボリューム・グループを作成してください。

ブロック・ボリューム・グループを作成し、リージョン間レプリケーションを有効化するには、次の手順を実行します。

- プライマリ・リージョンのOracle Cloud Infrastructureコンソールにログオンします。
- 「Storage」→「Volume Groups」に移動します。
- コンピュート・インスタンスと同じ可用性ドメイン（AD）にブロック・ボリューム・グループを作成します。以下に例を示します。
<soaprefix>-BVGGroup-AD1
- レプリケートするブロック・ボリュームをボリューム・グループに追加します。

注：ブート・ボリュームは追加しないでください。これらはレプリケートされません。

- ボリューム・グループのリージョン間レプリケーションを有効化します。
 - ターゲット・リージョン：セカンダリ・リージョンを選択します。
 - 可用性ドメイン：これらのボリュームをマウントするコンピュートが存在するセカンダリ・リージョンのADを設定します。
 - ボリューム・グループ・レプリカ名：ブロック・ボリューム・レプリケーション用の名前。
- ブロック・ボリュームですでにリージョン間レプリケーションが個別にONに設定されている場合は、OFFに切り替えます。レプリケーションはボリューム・グループで構成される必要があります。
- 変更を保存した後、セカンダリ・リージョンにレプリカが作成されていることを確認します。OCIコンソールでセカンダリ・リージョンを選択し、「Storage」→「Block Storage」→「Volume Group Replicas」に移動します。

プライマリ・コンピュート・インスタンスが複数のADにある場合は、同じ手順を繰り返して別のADにブロック・ボリューム・グループを作成します。

d) セカンダリ中間層ホストから元のブロック・ボリュームをデタッチする

注：ブート・ボリュームはアンマウントもデタッチもしないでください。

セカンダリの各中間層ホストで以下を実行します。

- /u01/dataにマウントされているブロック・ボリュームをアンマウントします。

[opc@soampdrs-soa-0 opc]# sudo umount /u01/data

実行中のOracleプロセスがないことを確認してください。この時点ではOracleプロセスは停止されているはずですが、このフォルダでまだ実行中のプロセスがある場合、アンマウントが失敗します。

- アンマウントされたら、このブロック・ボリュームをOCIコンソールからデタッチします。
それぞれのブロック・ボリュームで、「Attached Instances」→「Detach from Instance」を選択します。
OCIコンソールで、デタッチを完了する前に、いくつかのiscsciコマンドを実行するよう求められます。
- rootユーザーで、/etc/fstabファイルを編集し、/u01/dataに関するエントリを削除します。こうすることで、次の再起動時に元のBVのマウントが試行されなくなります。例：

```
..  
#このエントリを削除：  
#UUID=765185db-a10e-4da4-bffd-0437348b3cf6 /u01/data ext4 auto,defaults,_netdev,nofail 0 2
```

セカンダリの残りの中間層ノードで上記ステップを繰り返します。

e) セカンダリでデタッチしたブロック・ボリュームを削除または名前変更する

前の手順でセカンダリ中間層ホストからデタッチした元のブロック・ボリュームは今後使用しません。削除するか、名前を変更して後で削除することができます。

f) セカンダリ・コンピュート・インスタンスでsystemdデーモンを再起動し、以前マウントしたデバイス（rootユーザーによる“systemctl daemon-reload”）ヘキヤッシュされた参照をリフレッシュします。

3. 環境固有情報への置換用スクリプトの準備

スイッチオーバー/フェイルオーバー操作中、レプリケートされたブロック・ボリュームをセカンダリ中間層ホストでマウントした後、WebLogicドメイン構成を置換する必要があります。これは、WebLogicドメイン構成はプライマリのコピーであるため、tnsnames.oraのTNSエントリはプライマリ・データベースを参照しているからです。これをセカンダリ・データベースの接続の詳細で置き換えます。

この置換操作業を自動化するために、replacement_script_BVmodel.shスクリプトを使用します。接続文字列の置換と同様に、このスクリプトによってもWeblogicサーバーの一部のステート・ファイル（.lckおよび.state）がクリーンな起動のためにクリーンアップされます。

- 次からスクリプトをダウンロードします。https://github.com/oracle-samples/maa/tree/main/wls_mp_dr/Block_Volume_Replica_Method
- すべての中間層ホスト（プライマリとセカンダリ）にアップロードします。
- レプリケート対象のブロック・ボリュームには含まれないフォルダにスクリプトを保存します。たとえば、oracleユーザーのホーム内のフォルダ（例：/home/oracle/scripts）に保存します。
- ファイルの所有者をoracleユーザーに変更します（このスクリプトはoracleユーザーで実行します）。
- スクリプトを編集し、各ホスト内で適切な値を設定してカスタマイズします。その際に、各サイトのデータベースに合わせたローカルおよびリモートの値を指定します。

この時点ではスクリプトを実行しないでください。このスクリプトは、次にスイッチオーバーまたはフェイルオーバーが実行されるときに使用されます。SOAMP DRセットアップの準備ができました。[DRセットアップの検証](#)に続きます。

DBFSの方法とFSS/rsyncの方法を使用して構成する

注：以下の手順は、DBFSの方法およびFSS/rsyncの方法のみに適用されます。

DBFSまたはFSS/rsyncモデルを使用している場合は、次の手順を実行してDRのセットアップを完了します。

1. DBFSの方法でDBFSステージング・マウントを構成する

SOA Suite on OCI Marketplaceには、構成済みのOracle Database File System (Oracle DBFS) が事前にマウントされています。DBFSファイル・システムは、データベース表に格納されるファイルおよびディレクトリの上位にある標準ファイル・システム・インターフェースです。DBFSファイル・システムはデータベースに格納されているので、すべての中間層ホストからアクセスできる共有ファイル・システムとして使用できます。したがって、SOA Suite on OCI Marketplaceで構成されたDBFSファイル・システム（直接I/Oアクセスの場合は/u01/soacs/dbfsまたは/u01/soacs/dbfs_directory）により、インスタンスの中間層ノード間でファイルを共有できます（例：デプロイメント・プランのxmlファイル）。

本書で説明するディザスタ・リカバリ・ソリューションは、**このDBFSファイル・システムがSOA Suite on OCI Marketplaceインスタンスで動作していることを前提**としています。このソリューションは、ディザスタ・リカバリの初期セットアップでプライマリからスタンバイに変更を同期するための補助ファイル・システムとして使用され、システムのライフサイクル期間における構成変更のレプリケートでも使用されます。DBFSマウントは、ディザスタ・リカバリに関連したその他のWebLogic実行時操作では使用されないため、サービスにとって重要ではなく、システムのパフォーマンスに大きな影響を与えることもありません。

SOAMPホストをプロビジョニングすると、標準でDBFSファイル・システムが構成されマウントされた状態になります。

```
[oracle@soampdr14-soa-1 ~]$ df -h | grep dbfs
dbfs@ORCL:/ 229G 128K 229G 1% /u01/soacs/dbfs
dbfs@ORCL:/ 229G 128K 229G 1% /u01/soacs/dbfs_directory
```

マウントされていない場合は、スクリプト\$DOMAIN_HOME/dbfs/dbfsMount.shを使用してマウントできます。

SOAMP DRは、このDBFSマウント（/u01/soacs/dbfs）を利用して、プライマリWebLogicドメイン構成をセカンダリ・サイトに転送します。

2. FSS/rsyncの方法でFSSステージング・マウントを構成する

FSS/rsyncの方法を使用してWebLogic構成をレプリケートする場合、プライマリ・サイト用とセカンダリ・サイト用の2つのFSSファイル・システムを作成します。これらのファイル・システムのマウント先はローカル・ホストのみになります。セキュリティとパフォーマンスの理由から、このトポロジでリージョンを横断して直接NFSマウントを実行することはありません。これらのファイル・システムのマウントは、サイト間でrsyncコマンドを使用してデータをレプリケートするためのステージング領域として使用されます。この領域にドメイン・フォルダのコピーが保存されます。これらが実行時に使用されることはありません。

初期のDRセットアップでは、プライマリFSSボリュームは**プライマリ・サイトのWLS管理ホスト**にマウントされ、セカンダリFSSボリュームは**すべてのセカンダリ・サイトのSOAホスト**にマウントされます。セカンダリ・サイトのマウントは、DRセットアップ・フェーズで実行される、レプリケートされたドメインの初期コピーのソースとして使用されるため、すべてのセカンダリ中間層ホストで使用できる必要があります。

注：初回の構成の同期（初期DRセットアップ中に実施）が完了した後、FSSマウントが必要になるのはプライマリとスタンバイのWLS管理ホストのみです。残りのWLSノードからはFSSをアンマウントできます。ただし、FSSを使用して保管している追加のアーティファクトがあり、すべてのノードでFSSがマウントされている必要がある場合を除きます。

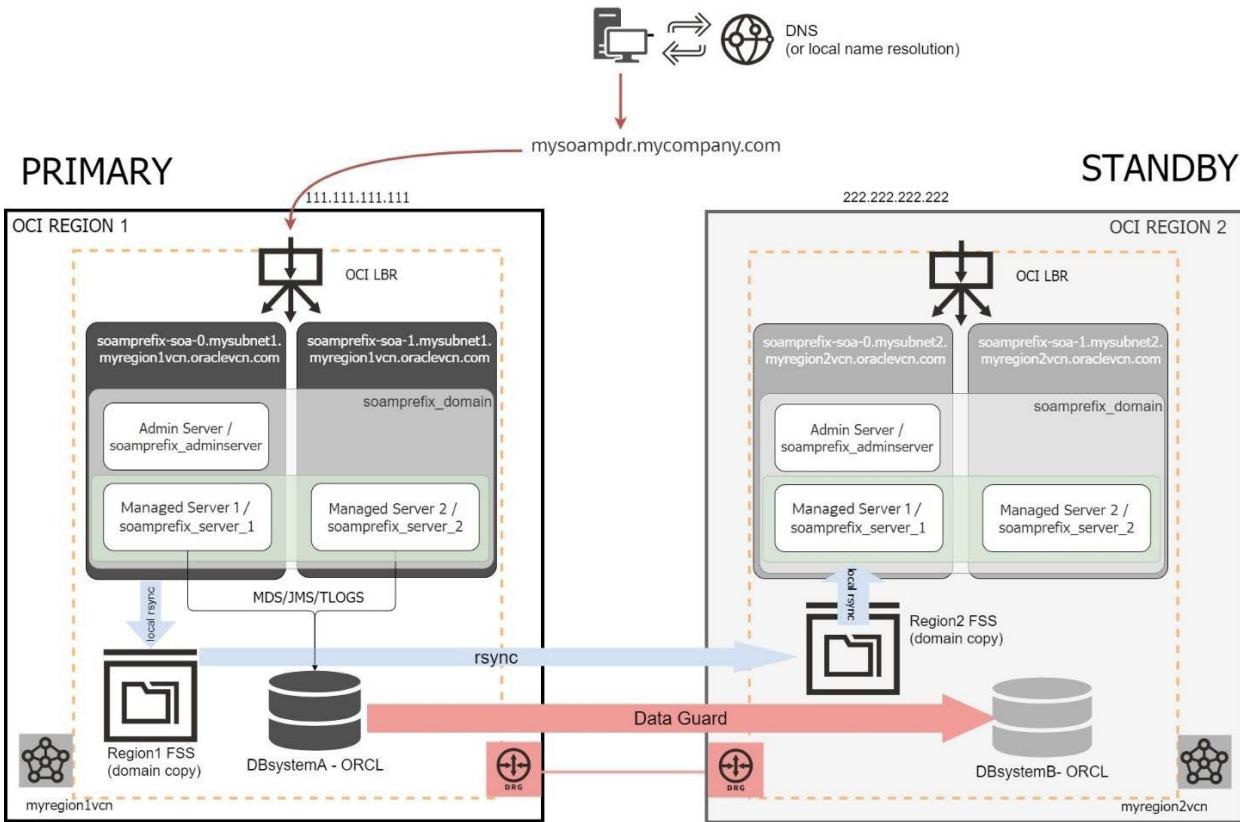


図9：補助FSSファイル・システムとrsyncを使用したWLSドメイン構成のレプリケーション青い矢印は構成コピーの論理フローを表しています。rsyncコマンドは、プライマリ・サイトまたはスタンバイ・サイトのWebLogic管理ホストで実行します。つまり、リモート・コピーの場合、rsyncを使用して、プライマリ・サイトのWebLogic管理ホストからスタンバイのWebLogic管理ホストに接続します。

OCIのFSSマウントを構成してマウントするには、以下の手順を実行します。

FSSの構成とマウントの手順	詳細	プライマリのサンプル値	セカンダリのサンプル値
1 各リージョンでマウント・ターゲットを作成する（まだ存在していない場合）	<ul style="list-style-type: none"> -OCIコンソールに接続する -適切なリージョンとコンパートメントを選択します（プライマリまたはセカンダリ） -Go to "File storage" > "Mount target". -「Create Mount target」をクリックします -作成後、それぞれのIPアドレスをメモしておく 	新しいマウント・ターゲットの名前： primary_mt 可用性ドメイン： <プライマリSOAと同じ> 仮想クラウド・ネットワーク： <プライマリSOAと同じ> サブネット： <プライマリSOAと同じ>	新しいマウント・ターゲットの名前： secondary_mt 可用性ドメイン： <セカンダリSOAと同じ> 仮想クラウド・ネットワーク： <セカンダリSOAと同じ> サブネット： <セカンダリSOAと同じ>
2 各リージョンでファイル・システムを作成する	<ul style="list-style-type: none"> -OCIコンソールに接続する -適切なリージョン（プライマリまたはセカンダリ）を選択する -「File storage」→「File System」に移動する -「Create File System」をクリックする -それそれで適切なマウント・ターゲットを選択します 	名前： primary_fs 可用性ドメイン： <プライマリSOAと同じ> エクスポート： /primaryfs マウント・ターゲット： primary_mt (以前に作成済み)	名前： secondary_fs 可用性ドメイン： <セカンダリSOAと同じ> エクスポート： /secondary fs マウント・ターゲット： secondary_mt (以前に作成済み)
3 FSSマウントに必要なネットワーク・セキュリティ・ルールを確認/設定する	ホストとマウント・ターゲット間のNFSトラフィックを許可するために、各サブネットでいくつかのネットワーク・ルールが必要になります。「 ファイル・ストレージに対するVCNセキュリティ・ルールの構成 」の手順に従って、ファイル・システムのセキュリティ・ルールを適切に設定します。		

4	<p>SOAホストにファイル・システムをマウントする</p> <p>すべてのSOAホストで以下を実行します。</p> <ul style="list-style-type: none"> - ローカルのマウント・ポイントを作成します。 例 : # sudo mkdir /u01/shared - rootユーザーとして/etc/fstabを編集し、リージョン固有の適切なマウント・ターゲットIP（プライマリ・サイトではプライマリのマウント・ターゲットIP、セカンダリ・サイトではセカンダリのマウント・ターゲットIP）を使用してマウントを追加します。 <pre><mount_target_ip_address>:<export_name> <your_local_mount_point> nfs defaults,nofail,nosuid,resvport 0 0</pre> <ul style="list-style-type: none"> - 新しいファイル・システムをマウントします。 # sudo mount -a <p>参照：Unixスタイルのインスタンスからのファイル・システムのマウント</p>	<p>プライマリ・ホストの/etc/fstabに追加する行の例（必ず1行にすること）</p> <pre>10.1.1.1:/primaryfs /u01/shared nfs defaults,nofail,nosuid,resvport 0 0</pre> <p>セカンダリ・ホストの/etc/fstabに追加する行の例（必ず1行にすること）：</p> <pre>10.2.2.2:/secondaryfs /u01/shared nfs defaults,nofail,nosuid,resvport 0 0</pre>	
5	<p>マウントされたファイル・システムを確認する</p> <p>すべてのSOAホストで以下を実行します。</p> <pre># df -h grep /u01/shared # ls -la /u01/shared</pre>	<p>プライマリSOAホストでの実行結果のサンプル：</p> <pre>[root@wlscociprefix-wls-0 opc]# df -h grep primaryfs 10.1.1.1:/primaryfs 8.0E 0 8.0E 0% /u01/shared</pre> <p>セカンダリSOAホストでの実行結果のサンプル：</p> <pre>[root@wlscociprefix-wls-0 opc]# df -h grep secondaryfs 10.2.2.2:/ secondaryfs 8.0E 0 8.0E 0% /u01/shared</pre>	
6	<p>マウントされたファイル・システムの所有者をoracleユーザーに変更します。</p> <p>- FSSボリュームがマウントされたら、oracleユーザーをマウント・ポイント・フォルダの所有者にします。</p> <pre># sudo chown oracle:oracle /u01/shared # sudo su - oracle</pre>	<p>プライマリの管理ホストでこれを実行します（マウントは残りのプライマリWLSホストと共有されているため、繰り返しは不要です）。</p>	<p>セカンダリの管理ホストでこれを実行します（マウントは残りのセカンダリWLSホストと共有されているため、繰り返しは不要です）。</p>

3. ディザスタ・リカバリ・セットアップ・ユーティリティ (DRS) の実行

ディザスタ・リカバリ・セットアップ・ユーティリティ (DRS) は、DBFSの方法およびFSS/rsyncの方法によるSOA Suite on Marketplaceのディザスタ・リカバリ・セットアップの構成手順を編成し、実行する一連のスクリプトです。

a) 必須のサイト間の接続性を確認する

DRSツールでは現在、サイト間で次の通信が必要です。

- セカンダリ中間層ホストからプライマリDBのIP、ポート1521**（および、RACデータベースを使用する場合はプライマリ・スキヤンIP）。

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OCI内部のネットワーク・インターフェクトを使用して接続する場合、セカンダリ中間層ホストはプライマリDBホストの**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・スキヤンIP）に接続します。これが**推奨アプローチ**です。

プライマリ・データベースとセカンダリ・データベースがパブリックIPを介して接続されている場合（サイト間でリモート・ピアリング/DRGが使用されないため）、セカンダリ中間層ホストは、プライマリDBホストの**パブリックIP**に接続します。これは一般に**推奨アプローチではなく、Oracle RAC DGに適していません**。

DRSを実行する前に、すべてのセカンダリ中間層でoracleユーザーを使用して簡単なチェックを実行します。これにより、ネットワーク・シナリオに応じて、プライマリ・データベースのプライベート/パブリックIPへの接続を検証できます。

```
java -classpath /u01/app/oracle/middleware/wlserver/server/lib/weblogic.jar utils.dbping  
ORACLE_THIN system
```

- プライマリWLS管理ホストからセカンダリWLS管理ホストのIP、ポート22**

これは、WebLogicドメインのプライマリからセカンダリへのrsyncコピーのために、**FSS/rsyncの方法でのみ**必要です。

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サイトがOCI内部のネットワークを使用して接続する場合、プライマリWLS管理ホストはセカンダリWLS管理ホストのプライベートIPに接続します。

これが**推奨アプローチ**です。

プライマリ・サイトとセカンダリ・サイトがパブリックIPを使用して接続されている場合（サイト間でリモート・ピアリング/DRGが使用されないため）、プライマリWLS管理ホストは、セカンダリWLS管理ホストのパブリックIPに接続します。これは**推奨アプローチではありません**。

具体的なネットワーク要件について詳しくは、本書の「[付録B - DRセットアップのネットワーク要件のサマリー](#)」を参照してください。

b) DRSを実行するホストを選択する

このツールは、両方のサイトにわたるDRトポロジ内にあるSOAおよびDBホストへのSSH接続が可能なすべてのホスト（OEL 7またはOEL 8のオペレーティング・システムを搭載）から実行できます。また、インターネットへ接続し、DRSに必要なPythonパッケージをダウンロードする必要があります。次のいずれかの方法を使用できます。

- SOAノードのいずれかからDRSを実行します。
- クラウド・テナント内の別のコンピュート・インスタンス（OEL 7またはOEL 8）からDRSを実行します。このコンピュート・インスタンスはDRSツールの実行用に使用でき、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。

DRSを実行するホストを選択する際には、次のDRS SSHアクセス要件を考慮してください。

- パブリック・ネットワークがSOAMP中間層ホストとDBホストによって使用されており、そのホストがパブリックIPアドレス経由でSSH接続可能である場合、**DRSは、SSH経由でこれらのパブリックIPアドレスに接続可能な任意のホストで実行できます**。DRSを実行する前にDRSプロパティ・ファイルを構成する場合は、ホストのパブリックIPを指定します。
- プライベートのネットワークを使用しており、ホストにパブリックIPが設定されない場合は、**DRSが実行されているホストと同じネットワーク・インフラストラクチャに併置して、サイト間通信用にすでに構成されているサイト間接続を使用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**。DRSを実行する前にDRSプロパティ・ファイルを構成する場合は、ホストのプライベートIPを指定します。

DRS実行時のSSHアクセスにパブリック・ネットワークを使用するかプライベート・ネットワークを使用するかは、DRを目的としたプライマリ・サイトとセカンダリ・サイトの通信方法とは別の考慮事項であることに注意してください。

c) DRSをダウンロードして実行する

DRSの実行手順：

- **drs-mp.tar.gz**を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-mp.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_mp_soa’ディレクトリに移動します。
- **README.md**を開いて、手順と推奨事項を確認してください。正常に実行して環境を適切に構成するには、すべての仕様を満たしていることが重要です。
- **drs_user_config.yaml**ファイルのプロパティを構成します。
- 適切なパラメータを指定してDRSツールを実行します。

DRSツールは、セカンダリSOAMPをスタンバイSOAMP 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。概要は次のとおりです。

- **初期チェック**を実行して、環境がDRセットアップ用に準備されていることを確認します。
- オプションとして、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**構成を追加します。セカンダリ中間層ホストの名前がプライマリ中間層の/etc/hostsにエイリアスとして追加され、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。
セ カン ダ リ ・ ド メ イ イ 構 成 の 変 更 前 の バ ッ ク ア ッ プ を 実 行 し ま す
(/u01/data/domains/soampdrs_domain_backup_<timestamp>)。
- **プライマリ・ドメイン構成のコピー**を**セカンダリ・サイト**にコピーします。プライマリのドメイン構成をステージング・マウント（DBFSまたはFSS）にコピーした後、セカンダリ・サイトで、そのステージング・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。一部のフォルダはコピーから明示的に除外されます。tnsnames.oraファイル、tmpフォルダ、lckファイルを含むフォルダなどです。

- 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します。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ますセカンダリ・フロントエンドのsoa-infra URLへの接続をチェックします。この確認は、DRSの実行時に"--do_not_start"フラグを指定することで任意にスキップできます。
- このプロセスの間、ツールによって、セカンダリ・データベースでデータベース・ロールの変換がいくつか実行されます（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、DRS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。セットアップの進捗状況は、このファイルとプロセスの標準出力を用いて監視できます。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・ロールのままで、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となります。

SOAMP DRセットアップの準備ができました。[DRセットアップの検証](#)に続きます。

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なら、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る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をドレインして完了させておくことが重要です。

DRセットアップの検証

オラクルは、DBFS、FSS、BVセットアップ手順の実行後、DRセットアップが正しいことをすぐに検証することを推奨します。検証するには、完全なスイッチオーバーを実行します（次ページ以降のスイッチオーバーに関する手順を参照）。または、ダウントIMEを回避するために、検証用にセカンダリ・サイトを開くことができます（次ページ以降の、検証のためのセカンダリ・サイトのオープンに関する手順を参照）。システムの適切なレプリケーション・メソッドの手順に従っていることを確認してください。

BVレプリケーションの方法のライフサイクル手順

BVレプリケーションの方法の構成レプリケーション

WebLogicドメイン構成は、リージョン間ブロック・ボリューム・レプリケーション機能を使用してスタンバイ・データベースにレプリケートされます。このレプリケーションは**自動**で行われ、Oracle Cloud Infrastructureによって管理されます。

セットアップ手順「リージョン間ブロック・ボリューム・レプリケーションの構成」で説明したように、WebLogicドメイン構成を含むブロック・ボリュームは、1つのボリューム・グループ（コンピュート・インスタンスが複数のADにある場合は複数のボリューム・グループ）に分類されます。システムのライフサイクル中は、プライマリ・ロールを持つシステムのこれらのボリューム・グループでリージョン間レプリケーションが有効化されていることを確認してください。以下に例を示します。

Volume Groups in soacsdr Compartment								
Create volume group								
Name	State	Number of volumes	Total size	Availability domain	Source volume group	Cross region replication	Backup policy	Created
soamod20-VolumeGroup-AD2	Available	1	50 GB	eXtUS-ASHBURN-AD-2	-	On	-	Mon, Nov 27, 2023, 14:20:29 UTC
soamod20-VolumeGroup-AD1	Available	1	50 GB	eXtUS-ASHBURN-AD-1	-	On	-	Mon, Nov 27, 2023, 14:20:29 UTC

BVレプリケーションの方法の検証のためのセカンダリ・サイトのオープン

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ます。このため、セカンダリ・サイトの検証によってプライマリ・データに影響が及ぶことはありません。

この操作は注意して実行する必要があります。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・サイトのSOAサーバーで起動時にそれらが処理されるからです。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してください。または、スナップショット・スタンバイ・データベースに変換された後、**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・サイトのSOAサーバーを起動してください（「[Removing 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（例ではSite2）を検証する手順は以下のとおりです。

検証のためにスタンバイ・サイトを開く手順		詳細
1	セカンダリ環境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SOAサーバーで起動時にそれらの処理が試行されます。 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。「 表削除なしでの実行時表からのレコードの削除 」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。
2	Site2でレプリカをアクティビ化する	この時点まで、ブロック・ボリュームがSite1からSite2へ継続的にレプリケートされています。これらをSite2でマウントするには、Site2のレプリカをアクティビ化する必要があります。 BVレプリカをアクティビ化すると、「アタッチ可能なBVがレプリケートされたBVのクローンとして作成されます」。 その後、これらのクローンBVをコンピュート・インスタンスにアタッチできます。Site2でレプリカをアクティビ化するには、OCIコンソールに接続し、次の手順を実行します。 - Site2で、「Block Storage」→「Volume Group Replicas」に移動します。 - ボリューム・グループ・レプリカをクリックして、「Activate」を選択します。 - ボリューム・グループ名としては、リージョンにかかわらず、同じ名前を使用することを推奨します。たとえば、次のとおりです。“<wlsciprefix>-BVGroup-AD1” - Site2のすべてのボリューム・グループ・レプリカに対して、同じ操作を繰り返します。
3	レプリケートされたブロック・ボリュームをSite2の中間層ホストにアタッチする	Site2でOCIコンソールの「Storage」→「Block Volume」に移動すると、アクティビ化により作成されたアタッチ可能なブロック・ボリュームが表示されます。 Site2でアクティビ化されたブロック・ボリュームをアタッチするには、次のようにします。

		<p>- ホストに適切なブロック・ボリュームをアタッチします。</p> <p>「Block Volume」で「Block Volume」をクリックして「Attached Instances」→「Attach to Instance」を選択します。手順を簡素化するには、“Oracle Cloud Agent”を使用して、iSCSIでアタッチされたボリュームに自動的に接続します” フラグのチェックを選択します。Oracle Cloud Agentによって自動的にiSCSIコマンドが実行されるため、コマンドを手動で実行する必要はありません。これらのコマンドをエージェントがコンピュート・インスタンス上で実行できるようにするには、https://docs.oracle.com/ja-ip/iias/Content/Block/Tasks/enablingblockvolumemanagementplugin.htm#blockplugin-prereq_permsで要件を確認します。</p> <p>Oracle Cloud Agentを使用しない場合は、iSCSIコマンドを手動で実行します。アタッチされたブロック・ボリュームで「iSCSI Commands & Information」をクリックし、中間層ホストの“Commands for connecting”に表示される iscsci</p> <p>コマンドを実行します。</p> <p>- 新しくアタッチされたBVのUUIDを取得します。</p> <pre>[root@soampdrs-soa-0 opc]# sudo blkid /dev/sda3:UUID="974147f5-d731-41de-bba8-56ff78ed1c9c" TYPE="xfs" PARTUUID="4a95c68a-bc70-4be9-bce8-b15e995fcf46" /dev/sda1:SEC_TYPE="msdos" UUID="593B-B893" TYPE="vfat" PARTLABEL="EFI System Partition" PARTUUID="c5ac3089-6a91-40e0-bcc1-212ba0b43418" /dev/sda2:UUID="9ca12daa-d7ea-44a2-8680-5b676488b054" TYPE="swap" PARTUUID="682a63d1-d3ec-4019-b372-43720aae717" /dev/sdb:UUID="35e72262-979a-4d84-85ce-a6f91e3b1250" TYPE="ext4"</pre> <p>- まだ完了していない場合は、マウントし、再起動後もマウントを持続できるように、適切なUUIDのエントリを /etc/fstabに追加します。</p> <pre>UUID=35e72262-979a-4d84-85ce-a6f91e3b1250 /u01/data ext4 auto,defaults,_netdev,nofail</pre> <p>注：最初のスイッチオーバーの実行後は、レプリケートされた各ブロック・ボリュームのUUIDは変更されなくなります。 /etc/fstabのエントリをコメントアウトするか、または維持することができます。ただし、コメントアウトしたままにすることが推奨されます。次にブロック・ボリュームがアタッチされたときに、systemdデーモンが自動的にブロック・ボリュームをマウントします。</p> <p>- デバイスがアタッチされたときに適切なエントリがすでに/etc/fstabに存在している場合、ブロック・ボリュームはアタッチされた後に自動的にマウントされます。それ以外の場合は、新しくアタッチしたブロック・ボリュームを/u01/dataにマウントして確認します。</p> <pre>[root@soampdrs-soa-0 opc]# mount -a [root@soampdrs-soa-0 opc]# df -h grep /u01/data /dev/sdb 49G 1.4G 46G 3% /u01/data</pre> <p>これらの手順を繰り返して、すべてのアクティブ化されたブロック・ボリュームをアタッチします。</p>
4	Site2の中間層で置換スクリプトを実行する	<p>Site2の管理ホストで、replacement_script_BVmodel.shスクリプトを実行します。</p> <p>すべてのノードのデータ・ブロック・ボリュームをレプリケートする場合でも、このスクリプトは管理ホストでしか実行できません。tnsnames.oraはDOMAIN_HOME/configフォルダの下にあるので、管理対象サーバーが起動すると、残りのノードは更新されたtnsnames.oraをダウンロードします。</p> <p>DBFSの場合の注記：SOAMPに付属のDBFSマウントを使用し、すべてのノードに対してデータ・ブロック・ボリュームをレプリケートしている場合は、Site2のすべてのノードでこのスクリプトを実行する必要があります。 DBFSアーティファクトにはそれ自体のtnsnames.oraが含まれ、それはDOMAIN_HOME/configフォルダにないためです。</p>
5	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	<p>プライマリDBホストでDGプローカを使用して、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。As user oracle:</p> <pre>[oracle@drdbA ~]\$ dgmgrl sys/your_sys_password@primary_db_unqname DGMGR> convert database "secondary_db_unqname" to snapshot standby "show configuration"を使用して変換が正常に実行されたことを確認します。</pre>
6	Site2でサーバーを起動する	<p>すべてのセカンダリ・サーバーでノード・マネージャを起動します。例：</p> <pre>\$ cd \$DOMAIN_HOME/bin/ \$ nohup ./startNodeManager.sh > \$DOMAIN_HOME/nodemanager/nodemanager.out 2>&1 &</pre> <p>セカンダリ管理サーバーを起動します。次に例を示します。</p> <pre>\$ cd /u01/app/oracle/middleware/oracle_common/common/bin \$./wlst.sh wlst> nmConnect ('weblogic', 'password','soampdrs-soa-</pre>

		O','5556','soampdrs_domain','/u01/data/domains/soampdrs_domain','SSL') wlst> nmStart('soampdrs_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）。
7	検証する	<p>これはスイッチオーバーではなく、プライマリ・サイトは引き続きアクティブ状態であるため、仮想フロントエンド名がプライマリ・サイトのLBR IPアドレスに解決されます。デフォルトでは、ブラウザからアクセスするとアクティブなプライマリ・サイトにリダイレクトされます。</p> <p>セカンダリ・サイトのSOAサービスに直接アクセスするには、管理下のクライアント（ラップトップなど）で/etc/hostsファイルを更新し、仮想フロントエンド名がセカンダリ・サイトのフロントエンドLBRのIPアドレスに解決されるように設定する必要があります。その後、このクライアントから検証を実行します。</p> <p>注：検証に使用されるクライアントがHTTPプロキシ経由でSOAMPシステムにアクセスしていないことを確認してください。HTTPプロキシによって、どちらの名前がクライアントの/etc/hostsに設定されているかにかかわらず、そのまま仮想フロントエンド名がプライマリ・サイトのLBRのIPアドレスに解決される可能性があるためです。</p> <p>注：Linux以外のクライアントで、カスタマイズしたホスト・ファイルのエントリを使用してブラウザでIPアドレスを解決するには、事前にローカルDNSキャッシュのリセットが必要になる場合があります。</p>

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・ロールで定義されたため、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・システムの検証が済んだら、以下の手順でスタンバイ・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	セカンダリのSite2のプロセスを停止する	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。ノード・マネージャのプロセスも停止し、DBFSがマウントされている場合はアンマウントします。
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プローカを使用して、セカンダリを再びフィジカル・スタンバイに変換します。ユーザーoracleで次を実行します。 <pre>[oracle@drdbA ~]\$ dgmgrl sys/your_sys_password@primary_db_unqname DGMGR> convert database "secondary_db_unqname" to physical standby "show configuration"を使用して変換が正常に実行されたことを確認します。</pre>
3	クライアントで更新された/etc/hostsを元に戻す	クライアントの/etc/hostsファイルの仮想フロントエンド名を、セカンダリ・サイトを参照するように更新した場合、これを元に戻して、仮想フロントエンド名がプライマリ・フロントエンドIPを再び参照するようにします。
4	Site2のブロック・ボリュームをデタッチする	レプリケートされたすべてのブロック・ボリュームに対して、以下を実行します。 <ul style="list-style-type: none"> - /u01/dataにマウントされているブロック・ボリュームをアンマウントします。 <pre>[opc@soampdrs-soa-0 opc]# sudo umount /u01/data</pre> <p>実行中のOracleプロセスがないことを確認してください。この時点ではOracleプロセスは停止されているはずですが、このフォルダでまだ実行中のプロセスがある場合、アンマウントが失敗します。</p> <ul style="list-style-type: none"> - アンマウントされたら、このブロック・ボリュームをOCIコンソールからデタッチします。それぞれのブロック・ボリュームで、「Attached Instances」→「Detach from Instance」を選択します。 <p>Oracle Cloud Agentを使用しないでブロック・ボリュームをアタッチした場合、OCIコンソールで、デタッチを完了する前にいくつかのiscscコマンドを実行するよう求められます。</p> <p>セカンダリのレプリケートされたブロック・ボリュームの残りで上記手順を繰り返します。</p>

5	Site2でデタッチしたボリュームを削除するか名前を変更して、誤ってマウントされないようにする	OCIコンソールを使用して、前の手順でSite2の中間層ホストからデタッチしたブロック・ボリュームを削除（または名前を変更）します。ボリューム・グループも削除します。これらは今後使用されません。
---	---	---

BVレプリケーションの方法のスイッヂオーバー

スイッヂオーバーは、管理者が2つのサイトのロールを元に戻す、計画された操作です。プライマリはスタンバイになり、スタンバイはプライマリになります。手動のスイッヂオーバーでは多数の操作が必要になり、RTOや運用のオーバーヘッド全体が増加する可能性があります。Oracleフル・スタックDRサービスを使用すると、これらのタスクのほとんどを自動化でき、必要な操作を簡素化できます。詳しくは、「[Oracle WebLogic ServerドメインでのOCI Full Stack Disaster Recovery Serviceの使用について学習](#)」を参照してください。

リージョン間ブロック・ボリューム・レプリケーションに基づくSOA Suite on OCI MarketplaceのDR構成で、[Site1からSite2への手動スイッヂオーバー](#)を実行するには、次の手順を実行します。

a) スイッヂオーバー前のタスク：

これらの手順では停止時間は発生しません。

スイッヂオーバー前の手順	詳細
1	以前に使用した、 Site2 のブロック・ボリュームをデタッチする 以前に使用されていた元のブロック・ボリュームがまだアタッチされている場合は、Site2の中間層ホストからこれをデタッチします（アンマウント後にデタッチ）。 次に、Site2でデタッチしたボリュームおよびボリューム・グループを削除するか名前を変更して、誤ってマウントされないようにします。これらは今後使用されません。
3	Site2 でレプリカをアクティブ化する この時点まで、ブロック・ボリュームがSite1からSite2へ継続的にレプリケートされています。それらをSite2でマウントするには、Site2のレプリカをアクティブ化する必要があります。 BVレプリカをアクティブ化すると、“アタッチ可能な” BVがレプリケートされたBVのクローンとして作成されます 。その後、これらのクローンBVをコンピュート・インスタンスにアタッチできます。Site2でレプリカをアクティブ化するには、OCIコンソールに接続し、次の手順を実行します。 - Site2 で、「 Block Storage 」→「 Volume Group Replicas 」に移動します。 - ボリューム・グループ・レプリカをクリックして、「 Activate 」を選択します。 - ボリューム・グループ名としては、リージョンにかかわらず、同じ名前を使用することを推奨します。以下に例を示します。 “<wlsciprefix>-BVGroup-AD1” - Site2のすべてのボリューム・グループ・レプリカに対して、同じ操作を繰り返します。
4	レプリケートされたBVをSite2の中間層ホストにアタッチする Site2でアクティブ化により作成されたアタッチ可能なブロック・ボリュームがリストに表示されます。Site2でアクティブ化されたブロック・ボリュームをアタッチするには、次のようにします。 - OCIコンソールの「 Block Storage 」→「 Block Volume 」で、「 Block Volume 」→「 Attached Instances 」→「 Attach to Instance 」をクリックします。 手順を簡素化するには、“ <i>Oracle Cloud Agent</i> を使用して、iSCSIでアタッチされたボリュームに自動的に接続します”フラグのチェックを選択します。 <i>Oracle Cloud Agent</i> によって自動的にiSCSIコマンドが実行されるため、コマンドを手動で実行する必要はありません。これらのコマンドをエージェントがコンピュート・インスタンス上で実行できるようにするには、 https://docs.oracle.com/ja-ip/iaas/Content/Block/Tasks/enablingblockvolumemanagementplugin.htm#blockplugin-prereq_perms で要件を確認します。 Oracle Cloud Agentを使用しない場合は、iSCSIコマンドを手動で実行します。アタッチされたブロック・ボリュームで「iSCSI Commands & Information」をクリックし、中間層ホストの“Commands for connecting”に表示されるiscsciコマンドを実行します。 - 新しくアタッチされたブロック・ボリュームの UUID を取得します。 [root@soampdrs-soa-0 opc]# sudo blkid /dev/sda3:UUID="974147f5-d731-41de-bba8-56ff78ed1c9c" TYPE="xfs" PARTUUID="4a95c68a-bc70-4be9-bce8-b15e995fcf46" /dev/sda1:SEC_TYPE="msdos" UUID="593B-B893" TYPE="vfat" PARTLABEL="EFI System Partition" PARTUUID="c5ac3089-6a91-40e0-bcc1-212ba0b43418" /dev/sda2:UUID="9ca12daa-d7ea-44a2-8680-5b676488b054" TYPE="swap" PARTUUID="682a63d1-d3ec-4019-b372-43720aaee717" /dev/sdb:UUID="35e72262-979a-4d84-85ce-a6f91e3b1250" TYPE="ext4"

	<p>- マウントし、再起動後もマウントを持続できるように、適切なUUIDのエントリを/etc/fstabに追加します。以下に例を示します。</p> <pre>UUID=35e72262-979a-4d84-85ce-a6f91e3b1250 /u01/data ext4 auto,defaults,_netdev,nofail</pre> <p>注：最初のスイッチオーバーの実行後、レプリケートされた各ブロック・ボリュームのUUIDは変更されなくなります。後続のスイッチオーバーでは、コメントアウトは外したままにすることが推奨されます。次にブロック・ボリュームがアタッチされたときに、systemd デーモンが自動的にブロック・ボリュームをマウントします。</p> <p>- デバイスがアタッチされたときに適切なエントリがすでに/etc/fstabに存在している場合、ブロック・ボリュームはアタッチされた後に自動的にマウントされます。それ以外の場合は、新しくアタッチしたブロック・ボリュームを/u01/dataにマウントして確認します。</p> <pre>[root@soampdrs-soa-0 opc]# mount -a [root@soampdrs-soa-0 opc]# df -h grep /u01/data /dev/sdb 49G 1.4G 46G 3% /u01/data</pre> <p>Site2のすべてのアクティブ化されたブロック・ボリュームで、同じ手順を繰り返します。</p>
5	<p>Site2で置換用スクリプトを実行する</p> <p>Site2の管理者ホストで、replacement_script_BVmodel.shスクリプトを実行します。</p> <p>すべてのノードのデータ・ブロック・ボリュームをレプリケートする場合でも、このスクリプトは管理ホストでしか実行できません。tnsnames.oraはDOMAIN_HOME/configフォルダの下にあるので、管理対象サーバーが起動すると、残りのノードは更新されたtnsnames.oraをダウンロードします。</p> <p>DBFSの場合の注記：SOAMPに付属のDBFSマウントを使用し、すべてのノードに対してデータ・ブロック・ボリュームをレプリケートしている場合は、Site2のすべてのノードで置換用スクリプトを実行する必要があります。</p> <p>DBFSアーティファクトにはそれ自体のtnsnames.oraが含まれ、それはDOMAIN_HOME/configフォルダにないためです。</p>

b) スイッチオーバー

実際のスイッチオーバーの手順はこの時点から開始します。

	スイッチオーバーの手順	詳細
1	プライマリ・サイトのサーバーを停止する	<p>Site1のWebLogic管理サーバー、管理対象サーバー、ノード・マネージャを停止します。</p> <p>注：SOAMPホストには、プロセスを開始/停止するスクリプトが標準で付属しています。ただし、https://github.com/oracle-samples/maa/tree/main/maa_wls.lifecycle_scriptsでMAAによって提供される開始/停止スクリプトを使用できます。</p> <p>これらのスクリプトにより、高い粒度でシャットダウン手順の向上を実現できます。</p>
2	仮想フロントエンドのDNS名をスイッチオーバーする	システムで使用される名前をホストしているDNSサーバーで必要なDNSプッシュを実行するか、クライアントでのファイル・ホスト解決を変更して、システムのフロントエンド・アドレスがSite2のLBRで使用されるパブリックIPを参照するようにします。
3	データベースをスイッチオーバーする	プライマリDBホストでDGブローカを使用してスイッチオーバーを実行します。oracleユーザーとして、次のコマンドを実行します。 [oracle@drdbwlmp1a ~]\$ dgmgrl sys/your_sys_password@primary_db_unqname DGMGR> switchover to "secondary_db_unqname"
4	Site2（新しいプライマリ）でサーバーを起動する	<p>すべてのセカンダリ・サーバーでノード・マネージャを起動します。例：</p> <pre>\$ cd \$DOMAIN_HOME/bin/ \$ nohup ./startNodeManager.sh > \$DOMAIN_HOME/nodemanager/nodemanager.out 2>&1 &</pre> <p>セカンダリ管理サーバーを起動します。例：</p> <pre>\$ cd /u01/app/oracle/middleware/oracle_common/common/bin \$./wlst.sh wlst> nmConnect ('weblogic', 'password','soampdrs-soa-0','5556','soampdrs_domain','/u01/data/domains/soampdrs_domain','SSL') wlst> nmStart('soampdrs_adminserver')</pre> <p>管理対象サーバーを起動します（セカンダリWebLogicコンソールまたはスクリプトを使用）。</p>

注：SOAMPホストには、プロセスを開始/停止するスクリプトが標準で付属しています。ただし、
https://github.com/oracle-samples/maa/tree/main/maa_wls.lifecycle_scriptsで
MAAによって提供される開始/停止スクリプトを使用できます。
これらのスクリプトにより、高い粒度とシャットダウン手順の向上を実現できます。

c) **スイッチオーバー後のタスク：**

この時点では、サービスが**Site2でアクティブ状態**であるため、追加の停止時間は必要ありません。ただし、スイッチオーバー手順を完了し、システムをロールが逆転した適切な状態のままにするには、追加タスクが必要です。次のタスクをすぐに実行することを推奨します。

- Site2（新しいプライマリ）のボリューム・グループで、逆方向のブロック・ボリューム・レプリケーションを有効にします。適切な可用性ドメインをこのレプリケーションに指定します。
- Site1（新しいスタンバイ）のボリューム・グループでレプリケーションを無効化します。
- 新しいプライマリからレプリケートされた、Site1（新しいスタンバイ）のブロック・ボリュームをアンマウントします。
- 今後のために、Site1の中間層ホストからアンマウントされたブロック・ボリュームをデタッチします。Oracle Cloud Agentを使用してブロック・ボリュームをアタッチした場合、エージェントがiSCSIコマンドを実行してiSCSIターゲットをログオフします。
- /etc/fstabのエントリをコメントアウトするか、または維持することができます。オラクルでは維持することを推奨します。次にブロック・ボリュームがアタッチされたときに、systemd демонが自動的にブロック・ボリュームをマウントします。
- Site1の中間層ホストからデタッチしたボリュームを削除するか名前を変更して、誤ってマウントされないようにします。Site1の未使用のボリューム・グループも削除します。これらは今後使用されません。

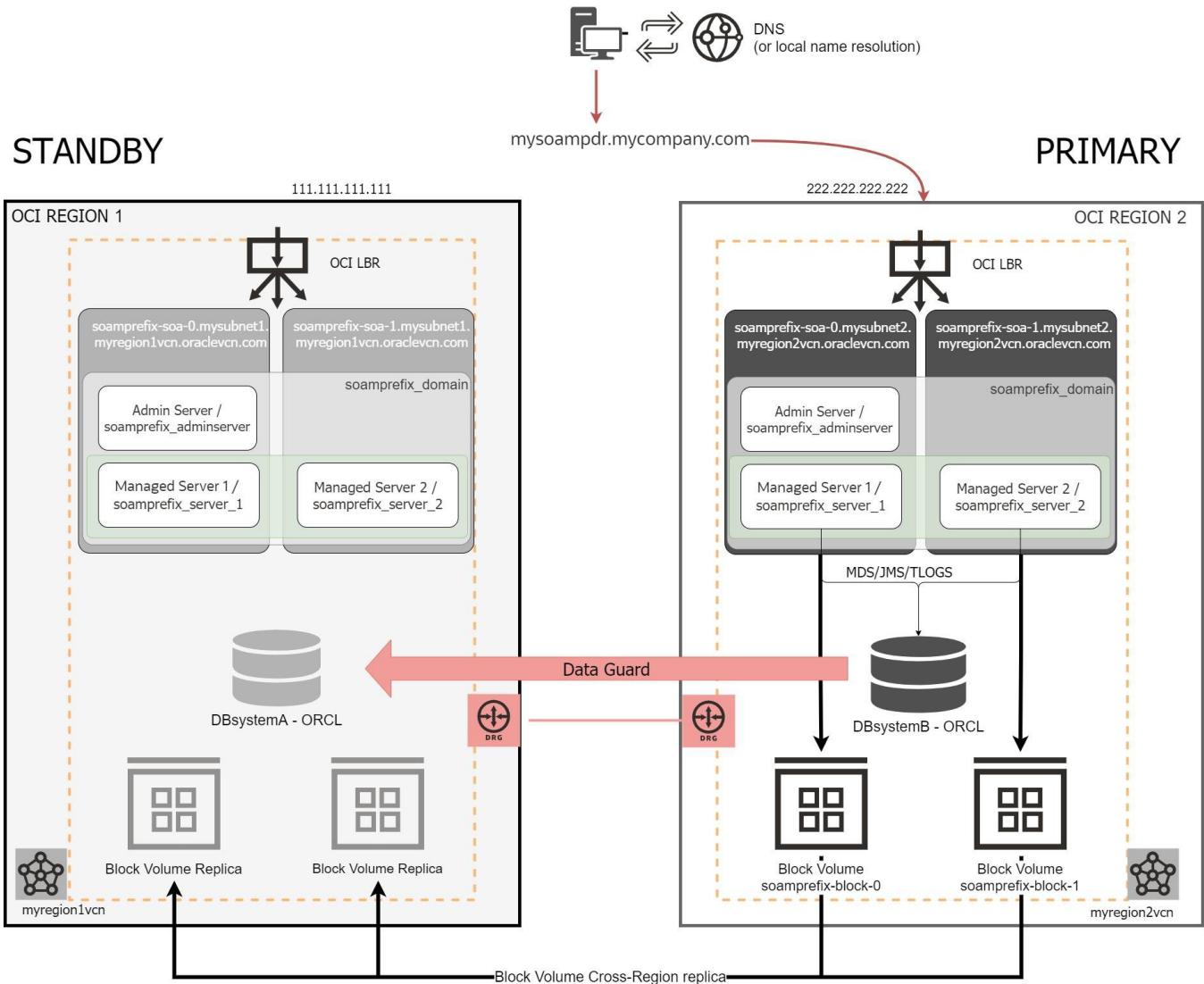


図10：スイッチオーバー後、逆方向のブロック・ボリューム・レプリケーションを有効化するための実行後手順

注（22.4.1以前のバージョンのSOAMPイメージの場合）

コンピュート・インスタンスを再起動するときに、新しいブロック・ボリュームのマウントにかかる時間は、元のブロック・ボリュームのマウントにかかった時間よりも少し長くなります。そのため、コンピュート・インスタンスの起動時にWLS起動スクリプト（`/opt/scripts/restart/restart_12c_servers.sh`）が実行されるまでに/u01/dataのマウントが間に合わず、ノード・マネージャとWLSプロセスが自動的には開始されない状況になる可能性があります。その場合は、
WLS起動

スクリプトを手動で実行できます。回避策として、`restart_12c_servers.sh`の実行前に待機時間を設けることができます。`/etc/rc.local`で、このスクリプトを実行する行の前に、`"sleep 90"`のコマンドを追加してください。参考資料：バグ33997638

BVLレプリケーションの方法のフェイルオーバー

フェイルオーバー操作は、プライマリ・サイトが使用できなくなったときに実行される、通常は計画外の操作です。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。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・スタンバイ・データベースで一貫性が維持されるかどうかに応じて、データ損失が発生する場合と発生しない場合があります。

手動のフェイルオーバーでは多数の操作が必要になり、RTOや運用のオーバーヘッド全体が増加する可能性があります。Oracleフル・スタックDRサービスを使用すると、これらのタスクのほとんどを自動化でき、必要な操作を簡素化できます。詳しくは、[Oracle WebLogic ServerドメインでのOCIフル・スタック・ディザスター・リカバリ・サービスの使用について学習を参照してください。](#)

手動フェイルオーバーの手順はスイッチオーバーの場合と同じであり、前述したスイッチオーバー前のタスクを総合的なRTOに追加する必要があるだけです（計画外の操作の前にBVのアクティブ化とアタッチメントのための準備をすることはできません）。

その他の違いは、データベース・ロールの変更方法です。フェイルオーバーでは、スタンバイDBに接続して、スイッチオーバー・コマンドの代わりに次のフェイルオーバー・コマンドを実行する必要があります。

```
[oracle@drdbwlmp1b ~]$ dgmgrl sys/your_sys_password@secondary_db_unqname
DGMGRL> failover to "secondary_db_unqname"
```

残りの手順は、実行前と実行後の手順を含めて、スイッチオーバーの手順と同じです。

通常、フェイルオーバー操作は、停止がプライマリ・リージョンに影響する場合に実行されます。そのため、一部のタスクはプライマリでは実行できない場合があります。たとえば、ホストにアクセスできないため、WLSプロセスはプライマリで停止できません。フェイルオーバー操作が終了して以前のプライマリ・サイトに再びアクセスできるようになると、手動のタスクをいくつか実行して、将来のスイッチバックのためにシステムを準備する必要があります。必要なタスクは次のとおりです。

- 障害が発生したサイトの**WebLogicプロセスを停止**します。フェイルオーバー時に停止していない場合、プロセスがハンギングする場合があります。プロセスが停止していることを確認してください。
- Data Guardの復旧操作**を実行します。フェイルオーバー後、障害が発生したプライマリは“Disabled Standby”として表示されます。復旧操作中、障害が発生したサイトのデータベースはフラッシュバックされ、フィジカル・スタンバイ・データベースとして変換されます。この操作はなるべくOCIコンソールを使用して実行してください。そうすれば、OCIコンソールでステータスが適宜更新されます。
- フェイルオーバー操作中に実行されていない場合は、スイッチオーバー操作で説明した**実行後の手順**を実行して**リージョン間レプリケーション**を切り替え、障害が発生したサイトのボリューム・グループとブロック・ボリュームをアンマウント、**デタッチ**、および**終了**します。

BVレプリケーションの方法のスケール・アウトとスケール・イン

SOA Marketplaceシステムは、SOA Marketplaceのドキュメント「[Oracle SOA Suite on Marketplaceインスタンス・クラスタのスケール・アウトまたはスケール・イン](#)」で説明されている手順に従って、スケール・アウトおよびスケール・インすることが可能です。

SOA MarketplaceのDR環境でスケール・アウトまたはスケール・インを実行する場合、DR環境に特有の性質を考慮する必要があります。つまり、2つのSOA Marketplaceインスタンス（プライマリとセカンダリ）があり、セカンダリのドメイン構成はプライマリ構成のコピーであるため、リスニング・アドレスとしてプライマリ・ホスト名が使用されています。

リスニング・アドレスのホスト名が、中間層の/etc/hostsにエイリアスとして追加されている場合、デフォルトでは、スケール・アウト操作中にプロビジョニングされた新規ノードの/etc/hostsファイル内にこれらのエイリアスが含まれません。この場合、新規ノードからWebLogic管理サーバーに接続できないため、セカンダリ・ロケーションでスケール・アウト手順が失敗する可能性があります。DR環境でスケール・アウトするとき、この問題の発生を回避するために必要な手順をここに示します。

https://github.com/oracle-samples/maa/tree/main/private_dns_views_for_drの説明に従って、セカンダリのDNSプライベート・ビューにプライマリ・ホスト名エントリを追加している場合、新規ノードは作成されるとすぐにプライマリ・ホスト名を解決できるため、スケール・アウト手順が容易になります。

詳しい手順については、以下を参照してください。

スケール・アウト

リージョン間ブロック・ボリューム・レプリケーション・モデルにおけるSOAMP DR環境のスケール・アウトの手順は次のとおりです。

- まず、スケールアウトを進める前に、「[BVレプリケーションの方法の検証のためのセカンダリ・サイトのオープン](#)」に示された手順に従ってセカンダリ・サイトを開きます。ただし、この時点では、まだスタンバイ・データベースをナップショットに変換せず、管理サーバーと管理対象サーバーも起動しません。スケール操作の実行前にセカンダリ・ホストでブロック・ボリュームの1つのバージョンをマウントするだけです。そうすることで、プライマリとセカンダリで個別にスケール操作を実行できます。
- ここで、**プライマリSOA Marketplaceインスタンスをスケールアウト**できます。
 - プライマリ・スタックの「[Oracle SOA Suite on Marketplaceインスタンス・クラスタのスケール・アウト](#)」で説明されている手順に従います。
 - スケール・アウトが正常に終了したら、SSHを**新しいノード**に接続して以下を実行します。
 - /etc/hostsを編集して、フロントエンドFQDNを追加し、プライマリ・フロントエンドLBRのIPアドレスを指定します。以下に例を示します。

```
# DR用の仮想フロントエンド名。プライマリ・フロントエンドのIPを指しています
111.111.111.111 soampdrs.mycompany.com
```

b. **(ホスト名のエイリアス設定にDNSプライベート・ビューを使用している場合は不要)**

新しいノードで/etc/hostsを編集して、残りのプライマリ・ノードにすでに存在しており、セカンダリ名を含むエイリアスを追加します。以下に例を示します。

```
10.0.0.82 <prim_middtier1_fqdn> <prim_middtier1_hostname> <sec_middtier1_fqdn> <sec_middtier1_hostname>
10.0.0.81 <prim_middtier2_fqdn> <prim_middtier2_hostname> <sec_middtier2_fqdn> <sec_middtier2_hostname>
```

c. **(ホスト名のエイリアス設定にDNSプライベート・ビューを使用している場合は不要)**

/etc/oci-hostname.confを編集し、PRESERVE_HOSTINFOを3に設定します。これにより、再起動しても上記の変更が維持されます。

3. 新しい管理対象サーバーを再起動します。

注：SOAMPによってスケール・アウトのために使用されるTerraformスクリプトによって、既存のノードに冗長／不要なブロック・ボリュームが作成されます。この重複したブロック・ボリュームには既存のブロック・ボリュームと同じ名前が設定され、既存のノードにアタッチされていますが、マウントはされていません。これらの重複したブロック・ボリュームは不要であるため、誤って正しくないブロック・ボリュームをマウントしないように、スケール・アウト後すぐにデタッチして削除することを強く推奨します。スケール・アウト・ジョブによって作成された重複するブロック・ボリュームは必ず削除するようしてください。

c) **セカンダリSOA Marketplaceインスタンスのスケール・アウト：**

セカンダリをスケール・アウトするには、特にプライマリ・アドレスのエイリアスをDNSプライベート・ビューではなく、/etc/hostsに追加した場合は、スケール・アウトの前に操作する必要があります。詳しい手順は以下のとおりです。

1. 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ます。

2. **(ホスト名のエイリアス設定にDNSプライベート・ビューを使用している場合は不要)**

スタンバイのWebLogicドメイン構成はプライマリのコピーであり、サーバーのリスニング・アドレスとしてプライマリ・ホスト名が使用されています。スケール・アウトの際にセカンダリに追加された新しいノードでは、プライマリ・ホスト名を認識しません（デフォルトではプライマリ名のエイリアスは新しいノードの/etc/hostsファイルに含まれません）。セカンダリでのスケール・アウトが問題なく完了できるようにするには、スケール・アウトを進める前にセカンダリ・ドメインでリスニング・アドレスを変更し、そこでセカンダリSOAホスト名を設定します。これにより、問題なくスケール・アウト手順を実行できます。手順：

- **プライマリSOAホストのFQDN**を特定します（スケール・アウト前の既存のノード）。以下に例を示します。

```
soampdr6-soa-
0.mysubnet1.myregion1vcn.oraclevcn.com
```

プライマリ中間層1のFQDNはsoampdr6-soa-0.mysubnet1.myregion1vcn.oraclevcn.comで、そのホスト名はsoampdr6-soa-0です。

プライマリ中間層2のFQDNはsoampdr6-soa-1.mysubnet1.myregion1vcn.oraclevcn.comで、そのホスト名はsoampdr6-soa-1です。

- **セカンダリSOAホストのFQDN**を特定します（現在の既存のノード）。以下に例を示します。

```
soampdr6-soa-0.mysubnet2.myregion2vcn.oraclevcn.com
```

```
soampdr6-soa-1.mysubnet2.myregion2vcn.oraclevcn.com
```

注：ホスト名はプライマリとセカンダリのSOAホストで同じになると想定され、FQDN値のみが異なります。

セカンダリ・サイトの管理サーバー・ノードで、<DOMAIN_HOME>/config/config.xmlファイル内のプライマリ・インスタンスのFQDNをセカンダリ・インスタンスのFQDNに置き換えます。

```
cd <DOMAIN_HOME>/config/
cp config.xml config.xml_backup_pre_scale-out
sed -i 's/primary_middtier1_fqdn/secondary_middtier1_fqdn/g' config.xml
sed -i 's/primary_middtier2_fqdn /secondary_middtier2_fqdn/g' config.xml
```

以下に例を示します。

```
sed -i 's/soampdr6-soa-0.mysubnet1.myregion1vcn.oraclevcn.com/soampdr6-soa-
0.mysubnet2.myregion2vcn.oraclevcn.com/g' config.xml
```

```
sed -i 's/soampdr6-soa-1.mysubnet1.myregion1vcn.oraclevcn.com/soampdr6-soa-
1.mysubnet2.myregion2vcn.oraclevcn.com/g' config.xml
```

- 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。

セカンダリ管理対象サーバーの起動は注意深く実行する必要があります。スタンバイ・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は、サーバーがそれらを処理する可能性があります。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・サーバーを起動してください。⁹

- セカンダリ・スタックの「[Oracle SOA Suite on Marketplaceインスタンス・クラスタのスケール・アウト](#)」で説明されている手順に従ってノードを追加します。
- スケール・アウト・プロセスが終了したら、新しく追加されたノードで必要なエイリアスを追加します。
 - 新しいノードで /etc/hosts を編集して、残りのセカンダリ・ノードと同様に、セカンダリ・フロントエンド LBR の IP アドレスに対してフロントエンド FQDN を追加します。

```
# DR用の仮想フロントエンド名。セカンダリ・フロントエンドのIPを指しています 222.222.222.222
soampdrs.mycompany.com
```

- (ホスト名のエイリアス設定にDNSプライベート・ビューを使用している場合は不要)**

新しいノードで /etc/hosts を編集し、セカンダリ中間層ノードにすでに含まれている既存のエイリアスを追加します。ここでは、プライマリ・ノードの FQDN がセカンダリ・ローカル IP アドレスのエイリアスとなっています。

```
10.2.0.12 <sec_middtier1_fqdn> <sec_middtier1_hostname> <prim_middtier1_fqdn>
<prim_middtier1_hostname>
```

/etc/oci-hostname.conf を編集し、PRESERVE_HOSTINFO を 3 に設定します。これにより、再起動しても上記の変更が維持されます。

- セカンダリ・サイトでサーバー（管理対象サーバーと管理サーバー）を停止します。
- スタンバイ・データベースを **フィジカル・スタンバイ** に変換します。

- (ホスト名のエイリアス設定にDNSプライベート・ビューを使用している場合は不要)**

オプションとして、手順 2 で行った変更を元に戻し、ドメイン構成のリスニング・アドレスでプライマリ FQDN を再設定することができます。または、後でプライマリから構成をレプリケートする際にこれは自動的にオーバーライドされます。

- d) **プライマリとスタンバイの両方がスケール・アウトされたら、新規ノードのエイリアスをすべての中間層ホスト（既存ノードと新規ノード）に追加して、構成を完了します。**

ホスト名のエイリアスを /etc/hosts に設定している場合、以下を実行します。

- プライマリで、これをすべての既存のプライマリ中間層ノード（および新しいノード）に追加します。例（必ず 1 行で記述）：

```
<primary_newnode_IP> <primary_newnode_fqdn> <primary_newnode_hostname> <secondary_newnode_fqdn>
<secondary_newnode_hostname>
```

- セカンダリで、すべての既存の中間層ノード（および新しいノード）にエイリアスを追加します。例（必ず 1 行で記述）：

```
<secondary_newnode_IP> <secondary_newnode_fqdn> <secondary_newnode_hostname> <primary_newnode_fqdn>
<primary_newnode_hostname>
```

DNS プライベート・ビューで設定している場合、以下を実行します。

/etc/hosts ではなく適切な DNS プライベート・ビューに新規ノードのホスト名を追加します。つまり、新規セカンダリ・ノードの名前をプライマリのプライベート・ビューに追加し、新規プライマリ・ノードの名前をセカンダリのプライベート・ビューに追加します。

- e) 「[BVレプリケーションの方法の検証のためのセカンダリ・サイトのオープン](#)」の説明に従い、ボリュームをデタッチしてセカンダリをスタンバイ・ロールに戻します。
- f) プライマリの新しいノードのブロック・ボリュームに対して、既存のプライマリ・ノードで適切なボリューム・グループに追加することによって構成したときと同じ方法で、リージョン間レプリケーションを有効化します。

⁹ 表削除なしでの実行時表からのレコードの削除

注：SOAMPIによってスケール・アウトのために使用されるTerraformスクリプトによって、既存のノードに冗長／不要なブロック・ボリュームが作成されます。この重複したブロック・ボリュームには既存のブロック・ボリュームと同じ名前が設定され、既存のノードにアタッチされていますが、マウントはされていません。これらの重複したブロック・ボリュームは不要であるため、誤って正しくないブロック・ボリュームをマウントしないように、スケール・アウト後すぐにデタッチして削除することを強く推奨します。スケール・アウト・ジョブによって作成された重複するブロック・ボリュームは必ず削除するようにしてください。

スケール・イン

リージョン間ブロック・ボリューム・レプリケーション・モデルにおけるSOAMP DR環境のスケール・インの手順は次のとおりです。

- a) はじめに、前述の「[BVレプリケーションの方法の検証のためのセカンダリ・サイトのオープン](#)」に示された手順に従って、セカンダリ・サイトを開きます。ただし、この時点では、まだスタンバイ・データベースをスナップショットに変換せず、管理サーバーと管理対象サーバーも起動しません。スケール操作の実行前にセカンダリ・ホストでブロック・ボリュームの1つのバージョンをマウントするだけです。そうすることで、プライマリとセカンダリで個別にスケール操作を実行できます。
- b) ボリューム・グループからブロック・ボリュームを削除することにより、これから削除するプライマリ・ノードのブロック・ボリュームでリージョン間レプリケーションを無効化します。スケール・イン・ジョブは、リージョン間レプリケーションが有効化されているブロック・ボリュームを削除できません。
- c) **プライマリSOA Marketplaceインスタンスをスケール・インします。**
 1. プライマリ・スタックの「[Oracle SOA Suite on Marketplaceインスタンス・クラスタのスケール・イン](#)」で説明されている手順に従います。
- d) **セカンダリSOA Marketplaceインスタンスをスケール・インします。**
 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ます。
 3. 管理サーバーのみを起動します（管理対象サーバーの起動は不要です）。
 4. セカンダリ・スタックの「[Oracle SOA Suite on Marketplaceインスタンス・クラスタのスケール・イン](#)」で説明されている手順に従います。
 5. 終了したら、実行中のWebLogicプロセスを停止し、セカンダリ・データベースをフィジカル・スタンバイに変換します。
- e) プライマリとセカンダリの中間層ホストで、削除したノードのエイリアスを/etc/hostsから削除します。DNSプライベート・ビューを使用している場合は、適切なビューから削除します。
- f) 次に、前述の「[BVレプリケーションの方法の検証のためのセカンダリ・サイトのオープン](#)」の説明に従い、ボリュームをデタッチしてセカンダリをスタンバイ・ロールに戻します。

注：SOAMPIによってスケール・インのために使用されるTerraformスクリプトによって、既存のノードに冗長／不要なブロック・ボリュームが作成されます。これらの重複したブロック・ボリュームは既存のブロック・ボリュームと同じ名前であり、ノードにアタッチされていますが、マウントされていません。これらは必要ないため、間違いが発生したり、マウントが正しくないブロック・ボリュームを参照したりしないようにするために、スケール・イン後すぐにデタッチして削除することを強く推奨します。スケール・イン・ジョブで作成された重複するブロック・ボリュームは必ず削除してください。

DBFSの方法とFSS/rsyncの方法のライフサイクル手順

構成レプリケーション

前のセクションで説明したように、データベース内のデータはすべて、Data Guard経由で自動的にスタンバイ・サイトにレプリケートされます。そのデータには、SOA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データ、SOAランタイム・データ、JMSとTLOG（JDBC永続ストアを使用している場合）、およびお客様データが含まれます。

ただし、**WebLogicドメインの構成のほとんどは、WebLogicドメイン・フォルダのファイルに存在します**。この（DBFSおよびFSSに基づく）レプリケーション・プロセスは、config_replica.shスクリプトを使用して自動化できます。このスクリプトは、DRトポロジで選択された方法に応じてDBFSまたはFSS/rsyncを介して、プライマリからスタンバイへWebLogic構成をレプリケートします。同じスクリプトがプライマリとスタンバイで使用され、これはDBFSとFSSの両方の方法に対して有効です。スクリプトには、レプリケーションを正しい方向で実行するために、どちらのサイトがプライマリとして機能し、どちらのサイトがセカンダリとして機能するかを決定するロジックが含まれます。

オプション1) DBFSの方法

このアプローチでは、プライマリ・サイトのドメイン構成のコピーを格納する**補助ファイル・システム**としてDBFSファイル・システムが使用されます。各SOA MPインスタンス（/u01/soacs/dbfs）で標準で自動構成されるDBFSファイル・システムが使用されます。

注：WebLogicサーバーのドメイン構成を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、FUSEライブラリ、マウント・ポイントなども依存することになります）。

このファイル・システムにある情報は、Data Guardを介しスタンバイ・ロケーションに自動的にレプリケートされます。スタンバイ・サイトでは、スタンバイ・データベースが読み取り専用モードで開いていなければ利用できません（Active Data Guardを使用する場合）が、データベースがスナップショット・スタンバイに変換される場合はDBFSファイル・システムもマウント可能です。

これを行う手順は、以下のとおりです。

- プライマリのWebLogicドメイン構成ディレクトリの内容が、**プライマリDBFSファイル・システムにコピー**されます。必要ないか無関係なファイル/フォルダ（一時フォルダ、TNS管理フォルダなど）は除外されます。
- DBFSにコピーされたファイルがデータベース内に格納されると、**Data Guardによって自動的にスタンバイ・データベースに転送**されます。
- スタンバイ・サイトでは、**データベースはスナップショット・スタンバイに変換され、DBFSマウントはスタンバイ中間層ホストでマウント**されます。
- スタンバイ・サイトでは、WLSドメイン構成ファイルが**DBFSマウントからスタンバイ・ドメインのフォルダにコピー**されます。

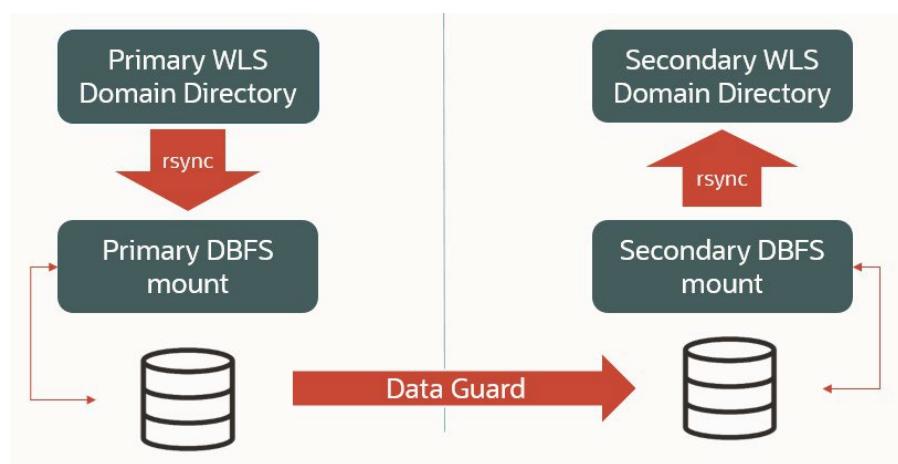


図11：DBFSの方法を使用した、スタンバイSOAへのドメイン構成変更のレプリケート

次の手順で、このレプリケーション方法をすばやく簡単に検証できます。

- DBFSマウントがプライマリSOAノードで使用可能なことを検証します。

```
[oracle@soampdrs-soa-0 ~]$ df -h | grep dbfs
dbfs-@ORCL:/ 244G 388M 243G 1% /u01/soacs/dbfs_directory
dbfs-@ORCL:/ 244G 388M 243G 1% /u01/soacs/dbfs
```

```
[oracle@soampdrs-soa-0 ~]$ ls  
/u01/soacs/dbfs share
```

- プライマリSOAノード1マウントにサンプル・ファイルを書き込みます。
[oracle@soampdrs-soa-0 ~]\$ echo "test" > /u01/soacs/dbfs/share/test.txt
- DBFSマウントがセカンダリで使用可能なことを検証します。これには、スタンバイ・データベースが読み取り専用でオープンされているか（Active Data Guardの使用時に可能）、スナップショット・スタンバイに変換されている必要があります。DBが読み取り専用モードかスナップショット・モードになったときにセカンダリにDBFSファイル・システムが存在しない場合は、dbfsMount.shスクリプトを使用してマウントできます。

```
[oracle@soampdrs-soa-0 ~]$ cd  
$DOMAIN_HOME/dbfs [oracle@soampdrs-soa-  
0 dbfs]$ ./dbfsMount.sh
```

- ファイルがセカンダリ・サイトに表示されるかどうかを確認します。
[oracle@soampdrs-soa-0 ~]\$ ls -la /u01/soacs/dbfs/share/test.txt
-rw-rw-r--.1 oracle 5 Mar 27 16:09 /u01/soacs/dbfs/share/test.txt

注：中間層は、tnsエイリアスを使用してローカルPDBデータベースに接続することで、DBFSをマウントします。このエイリアスは、\$DOMAIN_HOME/dbfs/tnsnames.oraファイルにあります。このエイリアスは、再試行のパラメータ化を使用して作成されているため、DBFSマウントからのコピー中、またはDBFSマウントへのコピー中にデータベース接続に問題が発生した場合は、この再試行が役立ちます。固有の要件を満たすために、必要に応じて、デフォルト設定値（合計10分であり、最小限のDBホストのリブートをサポート）を調整したり、小さくしたりすることができます。PDBのサービスに到達できない場合に、DBFSファイル・システムから情報を取得するオペレーティング・システム・コマンド（"df -h"や、DBFSマウント・フォルダでの"ls"など）でリトライが発生して、実行完了までに長い時間がかかる可能性がある点に注意してください。

オプション2) FSS/rsyncの方法

この方法では、rsyncを使用して、プライマリ・サイトのWLSドメイン構成をセカンダリ・サイトに定期的にレプリケートします。これを行う手順は、以下のとおりです。

- プライマリ・サイトで、ドメイン構成がローカルFSSファイル・システムにコピーされます。その後、プライマリFSSファイル・システムの内容がリモート・サイトのFSSファイル・システムにコピーされます。
- セカンダリ・サイトで、ドメイン構成がローカルFSSファイル・システムからWLSドメイン・ディレクトリにコピーされます。

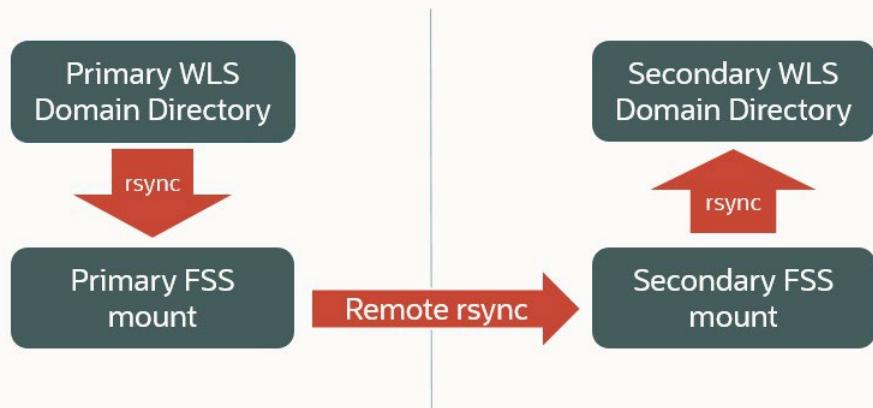


図12：FSS/rsyncの方法を使用した、WebLogicドメイン構成のレプリケート

WebLogic構成のレプリケートで**config_replica.sh**スクリプトを使用するには、次の手順を実行します。

1. スクリプトには以下の通信要件があります。
 - a. DBFSの方法の場合、DBロールを変更するために、それぞれのWebLogic管理ホストからリモート・データベースのリスナー・ポートにアクセスできる必要があります。それは、スクリプトがスタンバイ・ロールで実行される場合、DBFSをマウントするためにスタンバイ・データベースをスナップショットに変換するためです。
 - b. rsyncの方法の場合、それぞれのWebLogic管理ホストからリモートのWebLogic管理ホスト（sshポート）にSSHでアクセスできる必要があります。リモートrsyncコピーを実行するためには、リモート管理ホストへ接続します。

この通信を可能にするための適切なルールを作成するようにしてください。この通信は、パブリックIP経由（サイト間の接続にインターネット・ゲートウェイを使用している場合）でも、内部IP経由（サイト間の接続に動的ルーティング・ゲートウェイを使用している場合 - この方法を推奨）でも構いません。

2. まだ完了していない場合は、**config_replica.sh**スクリプトを**プライマリWebLogic管理ホスト**と**セカンダリWebLogic管理ホスト**にアップロードします。このスクリプトは他のスクリプトと同じフォルダに配置し、すべてのスクリプトに実行権限が付与されるようにしてください。**プライマリWebLogic管理ホスト**で、**config_replica.sh**スクリプトを開きます。カスタマイズ可能なパラメータのセクションを編集します。プライマリに適切な変数を指定するようにしてください。
3. **セカンダリWebLogic管理ホスト**で、**config_replica.sh**スクリプトを開きます。カスタマイズ可能なパラメータのセクションを編集します。セカンダリに適切な変数を指定するようにしてください。
4. はじめに**プライマリWebLogic管理ホスト**で、（oracleユーザーを使用して）**config_replica.sh**スクリプトを実行します。実行状況を監視し、エラーがないか確認します。このスクリプトは現在のサイトのロールを検証して、**プライマリWebLogic**ドメインからセカンダリ・サイトに（DBFSまたはFSS/rsyncの方法で）ドメイン構成をコピーします。
5. 完了したら、**セカンダリ・サイトのWebLogic管理サーバー・ホスト**で（oracleユーザーとして）**config_replica.sh**スクリプトを実行します。カスタマイズされたパラメータでは必ず適切な値を使用してください。スクリプトにより、データベース・ロールが検証されます。スタンバイの状態であるため、セカンダリのステージング・ファイル・システムからセカンダリWebLogicドメインにドメイン構成がコピーされます。

セカンダリWebLogic管理サーバーは通常、変更がレプリケートされるときに停止しています。次にサーバーを起動するときに（スイッチオーバー/フェイルオーバー時、または検証用にセカンダリ・サイトをオープンするとき）、この変更が有効になります。

WebLogic管理サーバーがセカンダリ・ロケーションで起動している場合は、再起動して変更を有効にする必要があります。セカンダリWebLogic管理サーバーを起動するには、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ることに注意してください。オラクルでは、古い構成が実行されることを避けるため、スタンバイWebLogic管理サーバーを停止したままにしておくことを推奨します。スタンバイ・サイトを検証する場合、またはスイッチオーバー/フェイルオーバー手順の間のみ、このサーバーを起動してください。

注：`<domain_home>/config`内の構成は、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`などにある起動スクリプトのカスタマイズが含まれます。

さらに、スクリプトによって変更が転送されるのは、ドメイン内のファイルのみです。*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、**config_replica.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。

注：アプリケーションのデプロイ操作では、*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”を使用することを推奨します。こうすることで、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が配置され、構成レプリケーション・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ます。

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（またはお客様が使用する他のスケジューリング・ツール）にスクリプトを追加して、定期的に、またはプライマリ・システムでの構成変更後に実行されるようにすることができます。このスクリプトは常にプライマリとスタンバイの両方で実行する必要があります。最初に**プライマリのWebLogic管理ホスト**で実行し（ドメイン構成をステージング・フォルダにコピー）、次に**スタンバイのWebLogic管理ホスト**で実行します（ドメイン構成のコピーをステージング・フォルダからドメイン・フォルダにコピー）。

ロールに変更がある場合は、スクリプトによって自動的に新しいロールへの実行が適応されます。それは、スクリプトがアクションを起こすためにサイトの実際のロールを確認するからです。

なお、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、次のような影響に注意してください。

同期は、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を追加して実行される可能性があります。つまり、それぞれ30分間隔で実行されるようにcronジョブが設定されている場合、プライマリでの処理時間枠とセカンダリ・ロケーションでの処理時間枠が重複していると、変更が有効になるまでに60分かかる可能性があります。

- スイッチオーバーを実行する前に、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ください。そうしないと、スタンバイに変更がコピーされないいうちにスイッチオーバーし、ロールの切替えで適用された元の変更が上書きされてしまう恐れがあります。
- 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。そうでない場合、コピー・ジョブが重複する可能性があります。

構成レプリケーションの検証

config_replica.shスクリプトは、ロールに応じて処理を変更します。サイトの現在のロールをチェックするために、ローカル・データベースのロールを取得します。サイトのロールがプライマリの場合、ドメイン構成のコンテンツをステージング・フォルダ（DBFSまたはFSS）にコピーします。サイトのロールがスタンバイの場合は、ステージング・フォルダ（DBFSまたはFSS）からドメイン・フォルダにコンテンツをコピーします。プライマリからスタンバイへの完全なレプリケーションを行うには、スクリプトを必ず両方のサイトで実行する必要があります。先にプライマリ・ロールを持つサイトで実行し、次にスタンバイ・ロールを持つサイトで実行します。

スイッチオーバーまたはフェイルオーバーの実行後に、構成レプリケーションが正しく動作することを必ず確認してください。構成レプリケーションを全面的に検証するには、以下の手順に従います。

- はじめに、プライマリからセカンダリへの構成レプリケーションを検証します。この作業ではプライマリの停止時間は発生しません。
 - プライマリWebLogicドメインで、何らかの構成変更を行います。たとえば、あるデータソースで接続プールのサイズを変更するか、またはその他の影響の小さい変更をプライマリWebLogicに適用します。
 - config_replica.shスクリプトを使用して、プライマリからスタンバイに構成をレプリケートします。このプロセスにはステップが2つあります。プライマリWebLogic管理ホストでconfig_replica.shを実行してから、スタンバイWebLogic管理ホストでconfig_replica.shを実行します。
 - プライマリで行われた構成変更が、セカンダリ・ドメインのディレクトリ内にあることを確認します。
 - 「検証のためのセカンダリ・サイトのオープン」で説明したように、スタンバイ・データベースをスナップショット・スタンバイに変換します。
 - WebLogic管理サーバーをセカンダリ中間層で起動し、構成が正しいことを検証します。
 - セカンダリのWebLogic管理サーバーを停止します。
 - スタンバイ・データベースをスナップショット・スタンバイからフィジカル・スタンバイに戻します。
 - プライマリで行った構成変更はテスト用であるため、ここで元に戻すことができます。
- 逆方向の構成レプリケーションを検証します（セカンダリへのスイッチオーバーまたはフェイルオーバー後）。
 この検証ではスイッチオーバーが必要になるため、プライマリの停止時間が発生します。メンテナンス時間内に以下の手順を実行します。
 - 本書の「DBFSの方法とFSS/rsyncの方法のスイッチオーバー」の説明に従って、システム全体をセカンダリにスイッチオーバーします。
 - 新しいプライマリWebLogicドメイン（以前のスタンバイ）で、何らかの構成変更を行います。
 - config_replica.shスクリプトを使用して、新しいプライマリから新しいスタンバイに構成をレプリケートします。新しいプライマリ管理ホストでconfig_replica.shを実行してから、新しいスタンバイ管理ホストでこのスクリプトを実行します。
 - 新しいスタンバイ・サイトにWebLogic構成が正しくレプリケートされたことを確認します。
 - システムをスイッチバックして元の状態に戻します。

検証のためのセカンダリ・サイトのオープン

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、セカンダリのSOAサーバーがスタンバイ・サイトで起動されるため、セカンダリで検証を実行できます。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ます。このため、セカンダリ・サイトの検証によってプライマリ・データに影響が及ぶことはありません。

この操作は注意して実行してください。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・サイトのSOAサーバーで起動時にそれらが処理されるからです。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してください。または、スナップショット・スタンバイ・データベースに変換された後、**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・サイトのSOAサーバーを起動してください（「[表削除なしでの実行時表からのレコードの削除](#)」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。

検証のためにスタンバイ・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プローカを使用して、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。 <pre>[oracle@drdbA ~]\$ dgmgrl sys/your_sys_password@primary_db_unqname DGMGRL> convert database "secondary_db_unqname" to snapshot standby "show configuration"を使用して変換が正常に実行されたことを確認します。</pre>
2 セカンダリ環境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SOAサーバーで起動時にそれらの処理が試行されます。 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。次に例を示します。 <pre>\$ cd /u01/app/oracle/middleware/oracle_common/common/bin \$./wlst.sh wlst> nmConnect ('weblogic', 'password','soampdrs-soa-0','5556','soampdrs_domain','/u01/data/domains/soampdrs_domain','SSL') wlst> nmStart('soampdrs_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）。</pre> 注：SOAMPホストには、プロセスを開始/停止するスクリプトが標準で付属しています。ただし、 https://github.com/oracle-samples/maa/tree/main/maa_wls_lifecycle_scripts でMAAによって提供される開始/停止スクリプトを使用できます。 これらのスクリプトにより、高い粒度とシャットダウン手順の向上を実現できます。
4 検証する	注：今回はスイッチオーバーではなく、プライマリ・サイトがアクティブ状態であるため、仮想フロントエンド名がプライマリ・サイトのLBR IPアドレスに解決されます。そのため、ブラウザからアクセスすると、デフォルトでアクティブなプライマリ・サイトにリダイレクトされます。 セカンダリ・サイトのSOAサービスに直接アクセスするには、管理下のクライアント（ラップトップなど）で /etc/hostsファイルを更新し、仮想フロントエンド名がセカンダリ・サイトのフロントエンドIPに解決されるように設定します。 その後、このクライアントから検証を実行します。 注：検証に使用されるクライアントがHTTPプロキシ経由でSOAMPシステムにアクセスしていないことを確認してください。HTTPプロキシによって、どちらの名前がクライアントの/etc/hostsに設定されているかにかかわらず、そのまま仮想フロントエンド名がプライマリ・サイトのLBRのIPアドレスに解決される可能性があるためです。 注：Linux以外のクライアントでは、カスタマイズしたホスト・ファイルのエントリを使用してブラウザでIPアドレスを解決するには、事前にローカルDNSキャッシュのリセットが必要になる場合があります。

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・サイトの検証が完了したら、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プローカを使用して、セカンダリを再びフィジカル・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"を使用して変換が正常に実行されたことを確認します。
3	クライアントの更新された/etc/hostsを元に戻す	クライアントの/etc/hostsファイルで、セカンダリ・サイトを参照するように仮想フロントエンド名を更新した場合、これを元に戻して、仮想フロントエンド名が再びプライマリのフロントエンドIPを参照するようにします。

DBFSの方法とFSS/rsyncの方法のスイッチオーバー

スイッチオーバーは、管理者が2つのサイトのロールを元に戻す、計画された操作です。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。

Oracleフル・スタックDRサービスを使用すると、これらのタスクのほとんどを自動化でき、必要な操作を簡素化できます。詳しくは、「[Oracle WebLogic ServerドメインでのOCI Full Stack Disaster Recovery Serviceの使用について学習](#)」を参照してください。

SOA Suite on OCI MarketplaceのDR構成で**手動スイッチオーバー**を実行するには、次の手順を実行します。

a) スイッチオーバー前のタスク

このスイッチオーバー前の手順では停止時間は発生しません。

	スイッチオーバー前の手順	詳細
1	保留中のすべての構成変更を伝播する	レプリケートすべき保留中の変更がある場合は、本書の「 構成レプリケーション 」のセクションを参照して、変更をセカンダリ・サイトにレプリケートしてください。 レプリケートした後、スケジューリングされているあらゆるレプリケーションを無効化して、スイッチオーバー時に実行されないようにします。

a) スイッチオーバー

実際のスイッチオーバーの手順はこの時点から開始します。

	スイッチオーバーの手順	詳細
1	プライマリ・サイトのサーバーを停止する	WebLogic管理サーバー・コンソールまたはスクリプトを使用して、プライマリ・サイトの管理対象サーバーを停止します。管理サーバーは実行中のままにもできますが、併せて停止することを推奨します。

		<p>注 : SOAMPホストには、プロセスを開始/停止するスクリプトが標準で付属しています。ただし、https://github.com/oracle-samples/maa/tree/main/maa_wls_lifecycle_scriptsでMAAによって提供される開始/停止スクリプトを使用できます。</p> <p>これらのスクリプトにより、高い粒度とシャットダウン手順の向上を実現できます。</p>
2	DNS名をスイッチオーバーする	<p>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LBR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。</p> <p>外部フロントエンドの解決にDNSを使用するシナリオでは（OCI DNS、商用DNSなど）、適切なAPIを使用して変更をプッシュすることができます。OCI DNSでこの変更をプッシュする例についてはこちらを参照してください。</p> <p>DNSエントリのTTL値はスイッチオーバーの実効RTOに影響を及ぼします。TTLの値が大きい場合（20分など）、DNSの変更がクライアントで有効になるまでにその指定した時間がかかります。この時間を短縮するには、TTL値を小さくします。ただし、TTLに小さい値を設定すると、クライアントがDNSをチェックする頻度が高くなるため、オーバーヘッドが生じる可能性があります。解決策の1つとして、DNSの変更前にTTLに一時的に小さい値（1分など）を設定します。その後、変更を実行してスイッチオーバーの手順が完了したら、TTLをもう一度通常の値に設定します。</p>
3	データベースをスイッチオーバーする	<p>プライマリDBホストでDGプローカを使用してスイッチオーバーを実行します。oracleユーザーとして、次のコマンドを実行します。[oracle@drdbwlmp1a ~]\$ dgmgrl sys/your_sys_password@primary_db_unqname DGMGR> switchover to “secondary_db_unqname”</p>
4	セカンダリ・サイト（新しいプライマリ）でサーバーを起動する	<p>セカンダリ管理サーバーを起動（起動済みの場合は再起動）して、スタンバイ中にレプリケートされた構成変更が有効になるようにします。</p> <p>セカンダリ管理対象サーバーを起動します（WebLogicコンソールかスクリプトを使用）。</p> <p>注 : SOAMPホストには、プロセスを開始/停止するスクリプトが標準で付属しています。ただし、https://github.com/oracle-samples/maa/tree/main/maa_wls_lifecycle_scriptsでMAAによって提供される開始/停止スクリプトを使用できます。</p> <p>これらのスクリプトにより、高い粒度とシャットダウン手順の向上を実現できます。</p>

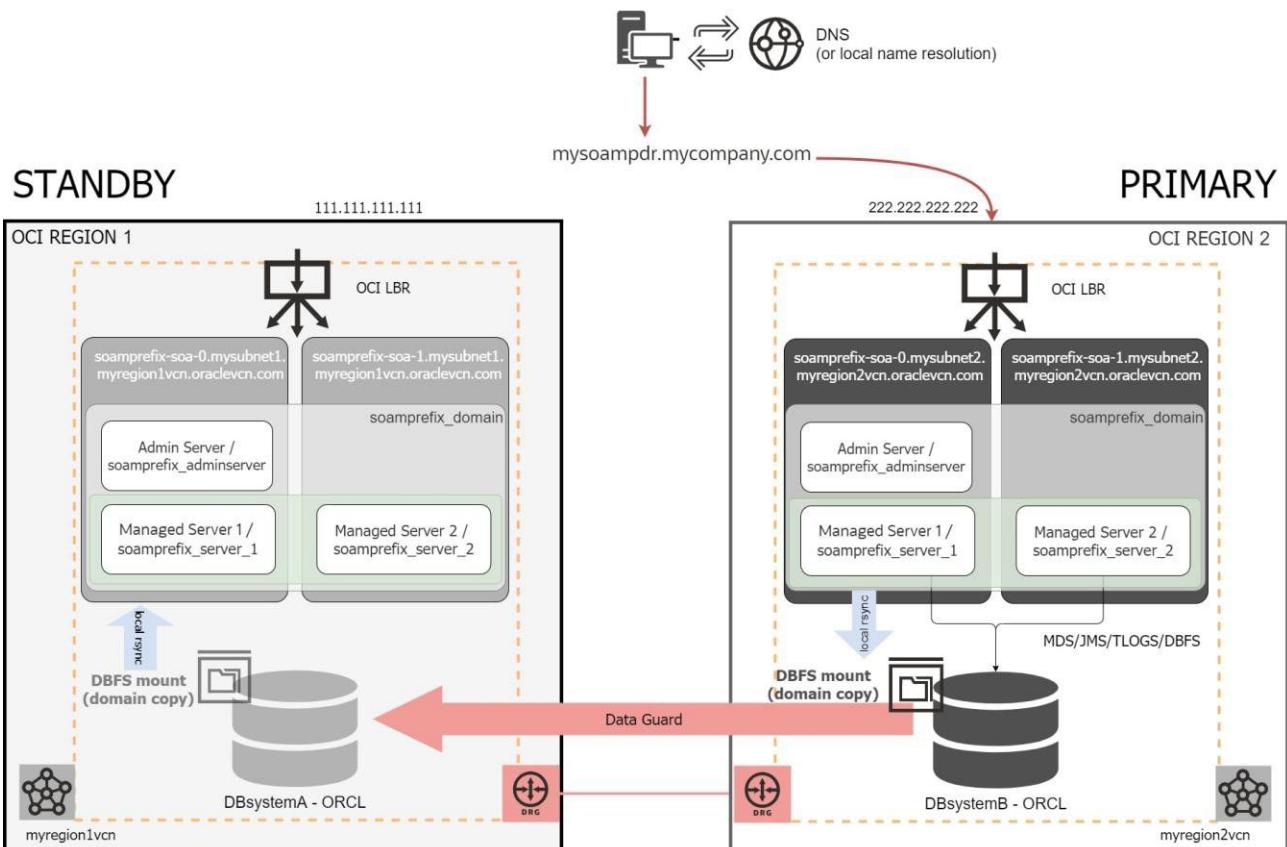


図13：スイッチオーバー後のSOA Suite on Marketplaceのディザスター・リカバリ（DBFSベースの方法）

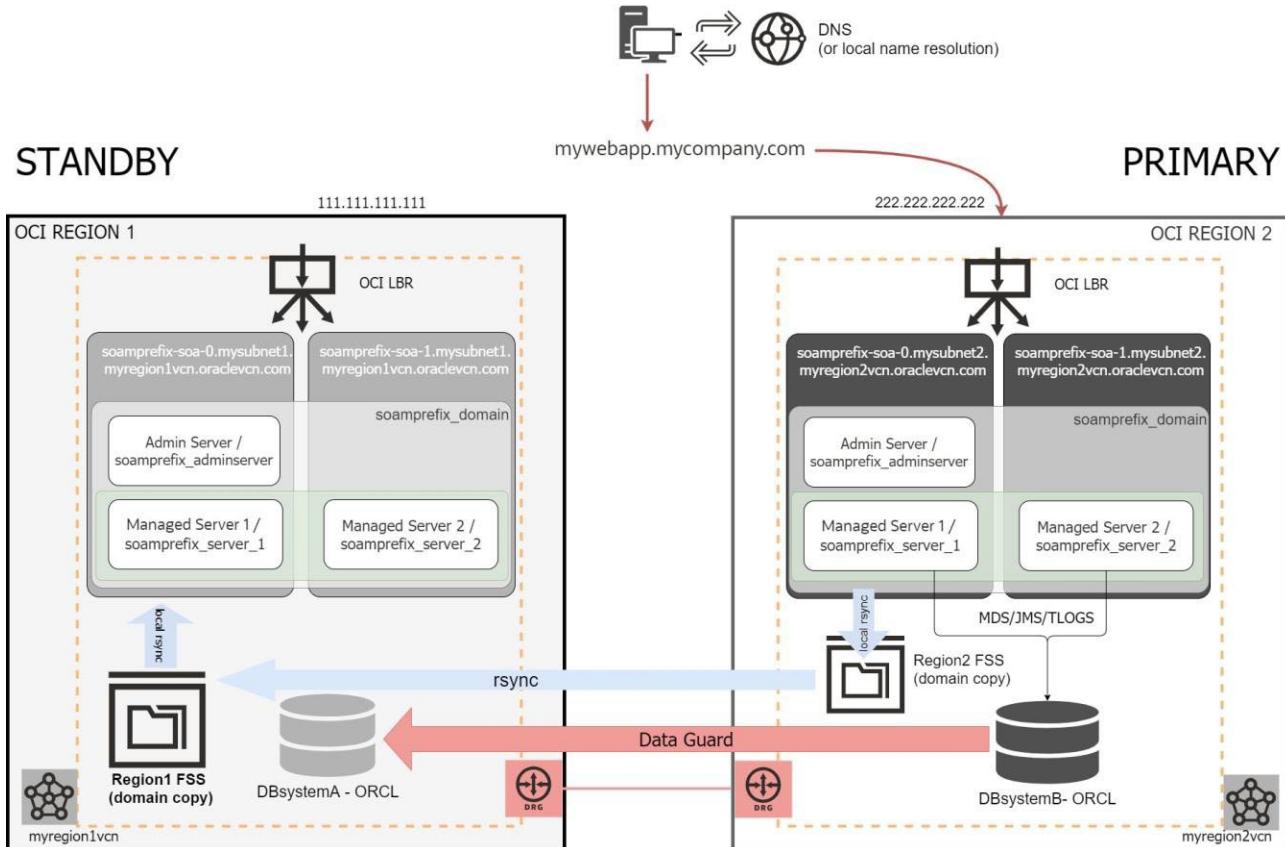


図14：スイッチオーバー後のSOA Suite on Marketplaceのディザスター・リカバリ（FSS/RSYNCの方法）

DBFSの方法とFSS/rsyncの方法のフェイルオーバー

フェイルオーバー操作は、プライマリ・サイトが使用できなくなったときに実行される、通常は計画外の操作です。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。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・スタンバイ・データベースで一貫性が維持されるかどうかに応じて、データ損失が発生する場合と発生しない場合があります。

Oracleフル・スタックDRサービスを使用すると、これらのタスクのほとんどを自動化でき、必要な操作を簡素化できます。詳しくは、「[Oracle WebLogic ServerドメインでのOCI Full Stack Disaster Recovery Serviceの使用について学習](#)」を参照してください。

SOA Suite on OCI MarketplaceのDR構成で手動フェイルオーバーを実行するには、次の手順を実行します。

	フェイルオーバーの手順	詳細
1	DNS名をスイッチオーバーする	システムで使用される名前をホストしている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LBR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。 外部フロントエンドの解決にDNSを使用するシナリオでは（OCI DNS、商用DNSなど）、適切なAPIを使用して変更をプッシュすることができます。OCI DNSでこの変更をプッシュする例については こちら を参照してください。
2	データベースをフェイルオーバーする	セカンダリDBホストでDGブローカを使用して、フェイルオーバーを実行します。oracleユーザーとして、次のコマンドを実行します。 [oracle@drdbwlmp1b ~]\$ dgmgrl sys/your_sys_password@secondary_db_unqname DGMGRL> failover to “secondary_db_unqname”

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（起動済みの場合は再起動）して、スタンバイ中にレプリケートされた構成変更が有効になるようにします。
セカンダリ管理対象サーバーを起動します（WebLogicコンソールかスクリプトを使用）。

通常、フェイルオーバー操作は、停止がプライマリ・リージョンに影響する場合に実行されます。そのため、一部のタスクはプライマリでは実行できない場合があります。たとえば、ホストにアクセスできないため、WLSプロセスはプライマリで停止できません。フェイルオーバー操作が終了して以前のプライマリ・サイトに再びアクセスできるようになると、手動のタスクをいくつか実行して、将来のスイッチバックのためにシステムを準備する必要があります。必要なタスクは次のとおりです。

- 障害が発生したサイトの**WebLogicプロセスを停止**します。フェイルオーバー時に停止していない場合、プロセスがハングする場合があります。プロセスが停止していることを確認してください。
- Data Guardの復旧操作**を実行します。フェイルオーバー後、障害が発生したプライマリは“Disabled Standby”として表示されます。復旧操作中、障害が発生したサイトのデータベースはラッシュバックされ、フィジカル・スタンバイ・データベースとして変換されます。この操作はなるべくOCIコンソールを使用して実行してください。そうすれば、OCIコンソールでステータスが適宜更新されます。
- 構成レプリケーション**（新しいプライマリから回復済みのセカンダリへ）が正しく実行されていることを**検証します**。

DBFSの方法とFSS/rsyncの方法のスケール・アウトとスケール・イン

SOA Marketplaceシステムは、SOA Marketplaceのドキュメント「[Oracle SOA Suite on Marketplaceインスタンス・クラスタのスケール・アウトまたはスケール・イン](#)」で説明されている手順に従って、スケール・アウトおよびスケール・インすることが可能です。

SOA MarketplaceのDR環境でスケール・アウトまたはスケール・インを実行する場合、DR環境に特有の性質を考慮する必要があります。つまり、2つのSOA Marketplaceインスタンス（プライマリとセカンダリ）があり、セカンダリのメイン構成はプライマリ構成のコピーであるため、リスニング・アドレスとしてプライマリ・ホスト名が使用されています。

リスニング・アドレスのホスト名が、中間層の/etc/hostsにエイリアスとして追加されている場合、デフォルトでは、スケール・アウト操作中にプロビジョニングされた新規ノードの/etc/hostsファイル内にこれらのエイリアスが含まれません。この場合、新規ノードからWLS管理サーバーに接続できないため、セカンダリ・ロケーションでスケール・アウト手順が失敗する可能性があります。DR環境でスケール・アウトするとき、この問題の発生を回避するために必要な手順をここに示します。

https://github.com/oracle-samples/maa/tree/main/private_dns_views_for_dr の説明に従って、セカンダリのDNSプライベート・ビューにプロトコル名エントリを追加している場合、新規ノードは作成されるとすぐにプライマリ・ホスト名を解決できるため、スケール・アウト手順が容易になります。

詳しい手順については、以下を参照してください。

スケール・アウト

注：データソースでTNSエイリアスのアプローチを使用している環境をスケール・アウトするには、パッチ34988073が含まれる必要があります。2023年2月以降に作成されたSOAMPインスタンスには、このパッチがあらかじめインストールされています。

SOA MP DR環境のスケール・アウトを実行するために推奨される手順は次のとおりです。

a) プライマリSOA Marketplaceインスタンスをスケール・アウトします。

- config_replica.shの定期実行をスケジューリングしている場合は停止します。
重要：セカンダリにもスケール・アウトを実行するまで、config_replica.shレプリケーションをセカンダリに実行しないでください。セカンダリSOAシステム構成に含まれるWebLogicサーバー・ノードで、セカンダリ・サーバーに認識されないものがある場合（セカンダリSOAがスケール・アウトされるまで、セカンダリには対応するノードが含まれません）、セカンダリでsoa-infraの起動が失敗する可能性があります。「プライマリとスタンバイでノード数が異なる場合」を参照してください。
- プライマリ・スタックの「[Oracle SOA Suite on Marketplaceインスタンス・クラスタのスケール・アウト](#)」で説明されている手順に従います。
- スケール・アウトが正常に終了したら、SSHを新しいノードに接続して以下を実行します。
 - /etc/hostsを編集して、フロントエンドFQDNを追加し、プライマリ・フロントエンドLBRのIPアドレスを指定します。
以下に例を示します。

DR用の仮想フロントエンド名。プライマリ・フロントエンドのIPを指しています
111.111.111.111 soampdrs.mycompany.com

b. (ホスト名のエイリアス設定にDNSプライベート・ビューを使用している場合は不要)

新しいノードで/etc/hostsを編集して、残りのプライマリ・ノードにすでに存在しており、セカンダリ名を含むエイリアスを追加します。以下に例を示します。

```
10.0.0.82 <prim_midtier1_fqdn> <prim_midtier1_hostname> <sec_midtier1_fqdn>
<sec_midtier1_hostname>
10.0.0.81 <prim_midtier2_fqdn> <prim_midtier2_hostname> <sec_midtier2_fqdn>
```

c. (ホスト名のエイリアス設定にDNSプライベート・ビューを使用している場合は不要)

/etc/oci-hostname.confを編集し、PRESERVE_HOSTINFOを3に設定します。これにより、再起動しても上記の変更が維持されます。

7. 新しい管理対象サーバーを再起動します。

b) セカンダリSOA Marketplaceインスタンスのスケール・アウト：

セカンダリにスケール・アウトを実行するには、スケール・アウトの前に介入する必要があります。スタンバイのWebLogicドメイン構成はプライマリのコピーであり、サーバーのリスニング・アドレスとしてプライマリ・ホスト名が使用されています。プライマリ名を解決するために/etc/hostsを使用する場合、スケール・アウト中にセカンダリに追加される新規ノードが、プライマリ名を認識しません（プライマリ名のエイリアスは、デフォルトでは新規ノードの/etc/hostsファイルに含まれていません）。セカンダリでのスケール・アウトを正しく完了させるため、実際のスケール・アウトに進む前に、セカンダリ・ドメインのリスニング・アドレスにセカンダリのホスト名を設定します。これにより、問題なくスケール・アウト手順を実行できます。

ホスト名のエイリアス設定にDNSプライベート・ビューを使用している場合、以下の手動作業の一部をスキップできます。詳しい手順は以下のとおりです。

1. 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ます。

2. (ホスト名のエイリアス設定にDNSプライベート・ビューを使用している場合は不要)

プライマリ・インスタンスの完全修飾ドメイン名ではなくセカンダリ・インスタンスの完全修飾ドメイン名を使用するように、セカンダリ・ドメインのWebLogicサーバーのリスニング・アドレスを変更します。この変更は後で元に戻します。ここで変更が必要な理由は、新しく追加されるノードでは、config.xmlに設定されたプライマリ・サーバーのエイリアスが/etc/hostsに含まれないからです。これには、次を実行します。

- プライマリSOAホストのFQDNを特定します（スケール・アウト前の既存のノード）。以下に例を示します。

```
soampdr6-soa-0.mysubnet1.myregion1vcn.oraclevcn.com
soampdr6-soa-1.mysubnet1.myregion1vcn.oraclevcn.com
```

プライマリ中間層1のFQDNはsoampdr6-soa-0.mysubnet1.myregion1vcn.oraclevcn.comで、そのホスト名はsoampdr6-soa-0です。

プライマリ中間層2のFQDNはsoampdr6-soa-1.mysubnet1.myregion1vcn.oraclevcn.comで、そのホスト名はsoampdr6-soa-1です。

- セカンダリSOAホストのFQDNを特定します（現在の既存のノード）。以下に例を示します。

```
soampdr6-soa-0.mysubnet2.myregion2vcn.oraclevcn.com
soampdr6-soa-1.mysubnet2.myregion2vcn.oraclevcn.com
```

注：ホスト名はプライマリとセカンダリのSOAホストで同じになると想定され、FQDN値のみが異なります。

セカンダリ・サイトの管理サーバー・ノードで、<DOMAIN_HOME>/config/config.xmlファイル内のプライマリ・インスタンスのFQDNをセカンダリ・インスタンスのFQDNに置き換えます。

```
cd <DOMAIN_HOME>/config/
cp config.xml config.xml_backup_pre_scale-out
sed -i 's/primary_midtier1_fqdn/secondary_midtier1_fqdn/g' config.xml
sed -i 's/primary_midtier2_fqdn /secondary_midtier2_fqdn/g' config.xml
```

以下に例を示します。

```
sed -i 's/soampdr6-soa-0.mysubnet1.myregion1vcn.oraclevcn.com/soampdr6-soa-
0.mysubnet2.myregion2vcn.oraclevcn.com/g' config.xml
sed -i 's/soampdr6-soa-1.mysubnet1.myregion1vcn.oraclevcn.com/soampdr6-soa-
1.mysubnet2.myregion2vcn.oraclevcn.com/g' config.xml
```

3. 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。

セカンダリ管理対象サーバーの起動は注意深く実行する必要があります。スタンバイ・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は、サーバーがそれらを処理する可能性があります。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・サーバーを起動してください。¹⁰

4. セカンダリ・スタックの「Oracle SOA Suite on Marketplaceインスタンス・クラスタのスケール・アウト」で説明されている手順に従ってノードを追加します。

5. スケール・アウト・プロセスが終了したら、新しく追加されたノードで必要なエイリアスを追加します。

- 新しいノードで /etc/hosts を編集して、残りのセカンダリ・ノードと同様に、セカンダリ・フロントエンド LBR の IP アドレスに対してフロントエンド FQDN を追加します。

DR用の仮想フロントエンド名。セカンダリ・フロントエンドのIPを指しています

222.222.222.222 soampdrs.mycompany.com

- (ホスト名のエイリアス設定にDNSプライベート・ビューを使用している場合は不要)

新しいノードで /etc/hosts を編集し、セカンダリ中間層ノードにすでに含まれている既存のエイリアスを追加します。

ここでは、プライマリ・ノードの FQDN がセカンダリ・ローカル IP アドレスのエイリアスとなっています。

10.2.0.12 <sec_middtier1_fqdn> <sec_middtier1_hostname> <prim_middtier1_fqdn>
<prim_middtier1_hostname>

/etc/oci-hostname.conf を編集し、PRESERVE_HOSTINFO を 3 に設定します。これにより、再起動しても上記の変更が維持されます。

6. セカンダリ・サイトでサーバー（管理対象サーバーと管理サーバー）を停止します。

7. スタンバイ・データベースをフィジカル・スタンバイに変換します。

8. (ホスト名のエイリアス設定にDNSプライベート・ビューを使用している場合は不要)

オプションとして、手順 2 で行った変更を元に戻し、リスニング・アドレスでプライマリ FQDN を再設定することができます。または、プライマリから構成をレプリケートする際に config_replica.sh を使用して後で自動的に実行することができます。

c) プライマリとスタンバイの両方がスケール・アウトされたら、新規ノードのエイリアスをすべての中間層ホスト（既存ノードと新規ノード）に追加して、構成を完了します。

ホスト名のエイリアスを /etc/hosts に設定している場合、以下を実行します。

- プライマリで、これをすべての既存のプライマリ中間層ノード（および新しいノード）に追加します。例（必ず 1 行で記述）：

<primary_newnode_IP> <primary_newnode_fqdn> <primary_newnode_hostname><secondary_newnode_fqdn><secondary_newnode_hostname>

- セカンダリで、すべての既存の中間層ノード（および新しいノード）にエイリアスを追加します。例（必ず 1 行で記述）：

<secondary_newnode_IP> <secondary_newnode_fqdn> <secondary_newnode_hostname><primary_newnode_fqdn><primary_newnode_hostname>

DNS プライベート・ビューで設定している場合は、/etc/hosts ではなく適切な DNS プライベート・ビューに新規ノードのホスト名を追加します。つまり、新規セカンダリ・ノードの名前をプライマリのプライベート・ビューに追加し、新規プライマリ・ノードの名前をセカンダリのプライベート・ビューに追加します。

d) ここですぐに config_replica.sh を実行し（通常どおり、プライマリで実行した後にセカンダリで実行）、プライマリからスタンバイに構成を伝播させます。これで、config_replica.sh の定期実行のスケジューリングを有効化できます。

スケール・イン

SOA MP の DR 環境をスケール・インするための推奨手順は以下のとおりです。

a) config_replica.sh の定期実行をスケジューリングしている場合は停止します。

重要：セカンダリにもスケール・インを実行するまで、config_replica.sh レプリケーションをセカンダリに実行しないでください。

b) プライマリ SOA Marketplace インスタンスをスケール・インします。

- プライマリ・スタックの「Oracle SOA Suite on Marketplace インスタンス・クラスタのスケール・イン」で説明されている手順に従います。

¹⁰ [Removing Records from the Runtime Tables Without Dropping the Tables](#)

- c) セカンダリSOA Marketplaceインスタンスをスケール・インします。
7. 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ます。
 8. 管理サーバーのみを起動します（管理対象サーバーの起動は不要です）。
 9. セカンダリ・スタックの「[Oracle SOA Suite on Marketplaceインスタンス・クラスタのスケール・イン](#)」で説明されている手順に従います。
 10. 処理が終わったら、セカンダリ・データベースをフジカル・スタンバイに変換します。
- d) プライマリとセカンダリの中間層ホストで、削除したノードのエイリアスを/etc/hostsから削除します。DNSプライベート・ビューを使用している場合は、適切なビューから削除します。
- e) （必要な場合）config_replica.shを実行し（プライマリで実行した後セカンダリで実行）、プライマリからスタンバイに構成を伝播させ、セカンダリを検証します。これで、config_replica.shの定期実行のスケジューリングを有効化できます。

DBFSウォレットの再作成

注：これはDBFSベースの方法のみに適用されます。

データベース内でSchemaPrefix_DBFSユーザーのパスワードが変更された場合、新しいパスワードでDBFSウォレットを再作成する必要があります。SOA Marketplaceのドキュメントのデータベース・スキーマおよびウォレット・パスワードの変更の「ウォレット・パスワードの手動更新」で説明されている手順を実行できます。必ず、DBFSアーティファクトによって使用される一貫性のある正しいTNSエイリアスを使用してください。以下の手順を実行します。

- DOMAIN_HOME/dbfsフォルダのdbfsMount.shスクリプトを確認して、dbfsマウントによって使用されるTNSエイリアスを特定します。
例：

```
$ORACLE_HOME/bin/dbfs_client -o wallet @<TNS_ALIAS> -o direct_io $MOUNT_PATH_DIRECTIO &>dbfs.log
&
$ORACLE_HOME/bin/dbfs_client -o wallet @<TNS_ALIAS> -o mount_path=$MOUNT_PATH
```
- DOMAIN_HOME/dbfs/tnsnames.oraを確認して、TNS ALIASが含まれており、ローカルPDBを参照していることを確認します。
- そのTNS_ALIAS名を使用して、新しい資格証明を作成します。例：

```
$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
```
- 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推奨します。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。

プライマリとスタンバイのすべての中間層ホストでこの処理を繰り返します。フォルダ\$DOMAIN_HOME/dbfs/のコンテンツは、DBFSの方法ではプライマリとスタンバイの間でレプリケートされません（されるべきではありません）。

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・ドメインにおいて「*Change the Database Schema Password Manually*」で説明されている手順を実行し、config_replica.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します。ドメイン構成内にあるデータソースおよびその他のファイルのパスワード変更は、すべてセカンダリにレプリケートされます。

一般的なライフサイクル手順

スタンバイ・サイトでコンピュート・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、プライマリとスタンバイ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ト・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、ディザスター・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：スタンバイ管理サーバー・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、DBFSの方法およびFSSの方法の場合はスタンバイ・ドメイン構成にプッシュされません。このため、フェイルオーバーの発生時、スタンバイ・ドメインにプライマリの構成が同期されない可能性があります。レプリケーションはWLS管理ノードに依存せずに内部的に実行されるため、この影響はBVアプローチには当てはまりません。
- RTOへの影響：セカンダリ中間層ホストが停止しており、起動が必要な場合、リカバリ時間は長くなります。スイッチオーバーまたはフェイルオーバーの前に、プライマリ・サイトのドメイン構成の変更を適用するためにセカンダリ・ドメインでドメイン構成の同期を実行する必要があるため、リカバリ時間はさらに長くなります。

これらの影響や、保守対応の時間と労力を最小限に抑えるため、いくつかのセカンダリSOAのコンピュート・インスタンスを停止させる場合には、少なくとも、セカンダリ・サイトのWebLogic管理コンピュート・インスタンスを起動したままの状態で維持し、他のWLS管理対象サーバーのコンピュート・インスタンスのみを停止します。

注：

お客様の課金条件は本書の範囲外です。一部のサーバーを停止することによる課金への影響を確かめるには、オラクルのライセンス・チームに問い合わせて、お客様の課金条件を確認してください。

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ト・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。SOAMPでは、すべてのコンピュート・シェイプがサポートされているため、停止したインスタンスの請求はコンピュート・シェイプに依存します。

プライマリとスタンバイで管理対象サーバー数が異なる場合

オラクルは、プライマリとスタンバイのSOAシステム間でリソース（ノード数、メモリなど）を正確に一致させることを強く推奨しています。プライマリ・ロケーションでスケール・アウト/スケール・インを実行する場合は、前述の説明に従って、セカンダリでも同じ処理を実行してください。ノード数が異なる場合、機能レベルやパフォーマンス・レベルで問題が生じる可能性があります。たとえば、プライマリがスケール・アウトされて2ノードから3ノードになり、その構成が、2ノードしかないスタンバイにレプリケートされた場合、セカンダリでsoa-infraが起動しません。これは、セカンダリ・ロケーションが認識していない新規ノードがあるからです（セカンダリ・サイトには対応するノードがないため、解決できません）。次のようなエラーが生じる可能性があります。

<May 18, 2020 10:55:48,394 AM GMT> <Error> <Deployer> <BEA-149231> <Unable to set the activation state to true for the application "soa-infra".>

`weblogic.application.ModuleException: java.net.UnknownHostException: soampdr6-soa-2.mysubnet1.myregion1vcn.oraclevcn.com`

人為的エラーやリカバリの発生によってこの状況に直面した場合、回避策として、プライマリに存在するがセカンダリに存在しないノードのために“偽の”エイリアスをセカンダリSOAホストに追加すると、既存のサーバーを起動できます。偽のエイリアスは存在しないIPアドレス（またはセカンダリDBのIPアドレスが使用可能）を参照します。それにより、既存のセカンダリ・サーバーでsoa-infraを起動できます。“新しい”ノードはセカンダリ内に存在せず、アクセスもできませんが、“unknownhostexception”エラーは発生しなくなり、既存ノードでsoa-infraアプリケーションを起動できるようになります。これは一貫した状況ではないため、このセカンダリ・ドメインにスケール・アウトを実行して新しいノードを追加しようといいください（構成の中に新しいサーバーはありますが、そのための実際のホストがありません）。この一貫性のない状況を解決するための正しい方法は、元のプライマリにスイッチオーバーして戻り、セカンダリのノード数と再び一致するようにスケール・インを実行してから、config_replica.shレプリケーションを実行して、ノード数が同じになったセカンダリにプライマリ構成をレプリケートすることです。

要約すると、プライマリとセカンダリで異なる数のサーバーを構成すると不整合が生じる可能性があるため、オラクルはこれを推奨しません。

SOAMP DR環境へのパッチ適用

SOAMP DRシステムで、Oracleソフトウェアにパッチを適用する際のガイドラインを以下に示します。ディザスタ・リカバリ・トポロジを使用することで、(場合によっては) パッチ適用の停止時間を短縮できます。

- データベース・パッチ

SOAMP DRトポロジはData Guardを使用します。プライマリ・データベースだけを使用するのではなくData Guardを利用するメリットは、順番にサイトにパッチを適用できることです。ただし、すべてのデータベース・パッチにこの方法を使用できるとは限りません。データベースにパッチを適用する際の停止時間と手順は、パッチのタイプによって異なります。データベース・パッチには次の種類があります。

- Data Guardスタンバイ・ファースト・パッチ：このタイプのパッチは、はじめにスタンバイに適用してからプライマリに適用できます。さまざまなオプションがあります。詳しくは、『*Oracle Patch Assurance - Data Guard Standby-First Patch Apply*』(Doc ID 1265700.1) を参照してください。
- 非Data Guardスタンバイ・ファースト・パッチ：このタイプのパッチは、プライマリ・データベースとスタンバイ・データベースの両方に同時に適用する必要があり、シャットダウンが必要です。

つまり、パッチがスタンバイ・ファーストに対応している場合、最小限の停止時間で処理を完了できるか、またはスイッチオーバーの時間まで短縮することができます。それ以外の場合は、プライマリとスタンバイをシャットダウンし、両方にパッチを適用しなければなりません。

- 中間層のみのパッチ（中間層ビットのみに適用）

FMWパッチの一部は、ReadmeでFMW_ROLLING_ORACLE_HOMEとしてマーク付けされます。このタイプのパッチでは、DRの使用の有無に関係なく、停止時間は発生しません。

ただし、他のパッチではFMW_ROLLING_ORACLE_HOMEに対応していないため、中間層のシャットダウンが必要になります。そのようなケースでは、ディザスタ・リカバリ・トポロジが役に立ちます。以下の手順に従います。

1. セカンダリ・データベースをスナップショット・スタンバイに変換します。
2. はじめに、セカンダリ中間層ドメインにパッチを適用します。
3. パッチを適用したセカンダリ・ドメインをテストします。
4. セカンダリですべての検証が完了したら、セカンダリ・データベースをフィジカル・スタンバイに戻します。
5. セカンダリにスイッチオーバーします（この時点でセカンダリ・リージョンがプライマリになり、ビジネスに必要な処理を実行します）。
6. 以前のプライマリ・データベースをスナップショットに変換します。
7. 古いプライマリにパッチを適用してテストします。
8. データベースをフィジカル・スタンバイに戻します。
9. 元のサイトにスイッチバックします。

この場合、停止時間はスイッチオーバー操作にかかる時間のみになります。

スタンバイ・システムがない場合、停止時間にパッチ適用時間が追加され、さらにシステムを停止して起動する時間がかかります。

- 中間層パッチ（DBスキーマ変更を含む）

パッチがFMW_ROLLING_ORACLE_HOMEに対応していない場合、前の手順とは少し異なり、DB変更を失わないようにする必要があります（DBスキーマ変更では中間層とDBへの同時パッチ適用が必要）。スタンバイ・システムでは、以下を実行できます。

1. セカンダリ・データベースをスナップショット・スタンバイに変換します。
2. はじめに、セカンダリ中間層ドメインにパッチを適用します。
3. パッチを適用したセカンダリ・ドメインをテストします。
4. セカンダリですべての検証が完了したら、セカンダリ・データベースをフィジカル・スタンバイに戻します。この時点で、セカンダリ WebLogicドメインの整合性は取れていません。スキーマのバージョンが、中間層のバージョンよりも古くなっています。
5. プライマリにパッチを適用します。

この場合の停止時間はDRなしの場合と同じですが、パッチ適用の手順とシステムの動作をスタンバイで最初に検証できるというメリットがあります。

スタンバイDBシステム再作成後のSOAMP DRの再構築

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、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。プライマリ・データベースをバックアップからリストアするには、（スタンバイDBシステムを停止することで）Data Guardの関連付けを削除し、プライマリ・データベースのリストア後にもう一度有効化する必要があります。この操作により、新しいスタンバイDBシステムが作成されます。

SOAMP 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システムに対してもう一度有効化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）。こうすることで、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SOAMP DRシステムに必要な変更を最小限に抑えることができます。

新しいスタンバイDBシステムを使用してSOAMP DRを再構築するには、以下の手順に従います。

- a) これから終了する元のスタンバイDBシステムの一意のDB名（\$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システムが停止したら、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) プライマリDBシステムで、OCIコンソールを使用してData Guardを再有効化する場合、**以前のスタンバイDBシステムが使用していたのと同じVCN、同じサブネット、同じホスト名プリフィックスを必ず指定してください**。これにより、新しいスタンバイDBシステムと前のスタンバイDBシステムで異なる値は、一意のDB名、プライベートIP、およびパブリックIPのみになります。
- d) 新しいDBシステムが問題なく作成され、OCIコンソールでのData Guard構成が完了したら、**新しいスタンバイDBシステムの一意のDB名およびプライベートIPの値をメモします**。

リージョン間ブロック・ボリューム・レプリケーションの方法の場合は、以下の手順を追加して実行します。

- a) 接続文字列の置換を実行するスクリプト（**replacement_script_BVmodel.sh**）のカスタム・プロパティが新しいスタンバイ・データベースと一致していることを検証します。新しいスタンバイに同じVCN、サブネット、およびホスト名接頭辞の値を再利用した場合、同じPDBサービス名を使用してデータベースに接続するようにします。何も変更する必要はありません。再利用しない場合は、スクリプトのカスタム値を更新し、次のスイッチオーバーで置換が正しく実行されるようにする必要があります。

DBFSの方法とFSS/rsyncの方法の場合は、以下の手順を追加して実行します。

- a) セカンダリ・ロケーションのconfig_replica.shのカスタム値を更新します。具体的には、再作成されたスタンバイDBの値でローカルCDBサービス名を更新します。
- b) **(TNSエイリアスのアプローチを使用している場合は不要)**
この手順は、TNSエイリアスのアプローチの前に構成されたSOAMP DR環境でのみ必要とされます。**スタンバイSOAMPホスト**で以下を実行します。
 - ファイル/**u01/data/domains/local_CDB_jdbcurl.nodelete**を編集し、スタンバイDBのunameを新しいスタンバイDBの一意の名前で更新します。
- c) **(TNSエイリアスのアプローチを使用している場合は不要)**
この手順は、TNSエイリアスのアプローチの前に構成されたSOAMP DR環境でのみ必要とされます。DBFSの方法を使用している場合は、**スタンバイSOAMPホスト**で以下を実行します。
 - **\$DOMAIN_HOME/dbfs/localdb.log**ファイルを編集します。
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。
\$DOMAIN_HOME/dbfs/tnsnames.oraファイルを編集します。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。そのうちの1つは、元のスタンバイDBシステムの一意の名前です。エイリアスとエイリアスのサービス名で、元のスタンバイDBの一意の名前を新しいスタンバイDBのそれで置き換えます。

DBFSの方法を使用している場合は、**プライマリSOAMPホスト**で以下を実行します。

- ファイル**\$DOMAIN_HOME/dbfs/tnsnames.ora**を編集します。このファイルには、エントリがいくつか含まれています。そのうちの1つは、元のスタンバイDBシステムの一意の名前です。元のスタンバイDBの一意の名前を新しいスタンバイDBのそれで置き換え（エイリアスとサービス名で）、元のスタンバイIPを新しいスタンバイIPで置き換えます。
tnsnames.ora内に指定されたスタンバイCDBのエイリアスは、プライマリとスタンバイのSOAホスト間で異なる場合があります。

- その場合、プライマリでは、スタンバイIP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ます。リージョン間でDNS解決は想定されていないため、これは想定どおりの動作です。
- プライマリSOAホストのlocaldb.log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。

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>

以上から、**スタンバイSOAMPホスト**では以下のようになります。

更新するファイル	元のコンテンツ	新しいコンテンツ
/u01/data/domains/ local_CDB_jdbcurl.nodelete	drdb6b- scan.mysubnet.region2vcn.oraclevcn.com:1521/ ORCL_phx1kg.mysubnet.region2vcn.oraclevcn.com	drdb6b- scan.mysubnet.region2vcn.oraclevcn.com:1521/ ORCL_phx1c3.mysubnet.region2vcn.oraclevcn.com
(DBFSの方法の場合のみ) \$DOMAIN_HOME/ dbfs/localdb.log	ORCL6_phx1kg	ORCL6_phx1c3
(DBFSの方法の場合のみ) \$DOMAIN_HOME/ dbfs/tnsnames.ora	... 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))) 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))) ...

プライマリSOAMPホストでは以下のようになります。

更新するファイル	元のコンテンツ	新しいコンテンツ
(DBFSの方法の場合のみ) ... \$DOMAIN_HOME/dbfs/ tnsnames.ora	... ORCL6_phx1kg = (DESCRIPTION=(SDU=65535)(SEND_BUF_SIZE=1048 5760)(RECV_BUF_SIZE=10485760)(ADDRESS=(PROT OCOL=TCP)(HOST=10.2.0.2)(PORT=1521))(CONNECT_ DATA=(SERVER=DEDICATED)(SERVICE_NAME=ORCL 6_phx1kg. ORCL6_phx1c3 = (DESCRIPTION=(SDU=65535)(SEND_BUF_SIZE=1048 5760)(RECV_BUF_SIZE=10485760)(ADDRESS=(PROT OCOL=TCP)(HOST=10.2.0.5)(PORT=1521))(CONNECT_ DATA=(SERVER=DEDICATED)(SERVICE_ NAME=ORCL6_phx1c3

	mysubnet.region2vcn.oraclevcn.com)(UR=A)))mysubnet.region2vcn.oraclevcn.com)(UR=A)))
--	--	---

- d) 元のスタンバイDBシステムに固有のIPを使用して作成された既存のOCIセキュリティ・ルールがある場合、新しいスタンバイDBシステムのIPを使用するように更新されていることを確認します（これが必要なのは、ルールがCIDRではなくIPに固有の場合のみ）。

これで、SOAMP 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。

RTDおよびRPOの概要

注：以下の値は、参考の目的のみに提供されるサンプル値であり、契約上の値として使用することはできません。これらの時間は、おのののシステムにおいて、多数の要素（アプリケーション、接続プール構成、ホスト・シェイプ、ロード、チューニングなど）によって異なる可能性があります。Oracle Cloud ピラー・ドキュメントには、オラクルによる可用性の観点での実際の契約上の義務となるSLA/SLOの正式な値が記載されています。ドキュメントはこちら（<https://www.oracle.com/assets/paas-iaas-pub-cld-srvs-pillar-4021422.pdf>）から確認できます。

予測されるRTD

リカバリ時間目標（RTD）は、特定のシステムで停止した場合の最大許容停止時間を表します。フェイルオーバーは通常、システムに影響する重大な問題によって引き起こされる計画外イベントであるため、その際発生する停止時間は、複数の“制御不能”な要因によって変わります。しかし、計画イベントであるスイッチオーバーに必要な停止時間は見積もりが可能です。

次の表に、サンプルのSOAMPシステムとMFTMPシステムでスイッチオーバーの各ステップにかかる標準的な時間を示します。例に挙げたこれらの特定のシステムでは、SOA/MFTホストでVM.Standard2.1シェイプ、SOA MPの場合はWLSサーバー向けに8Gサイズのヒープ・メモリ、MFT MPの場合は1 Gのヒープを使用しています。これらは、WebLogicサーバーの接続プールで標準提供される構成を使用しています。SOA MPシステムには“Fusion Order Demo”アプリケーションがデプロイされ（3コンポジット + 3アプリケーション）、MFT MPシステムには20を超える送信インスタンスがデプロイされています。

	スイッチオーバー手順	SOA MPにおける時間の例	MFT MPにおける時間の例
1	スイッチオーバー前のタスク	この処理では停止時間は発生しません	
停止時間の開始....			
2	プライマリ・サイトのサーバーを停止する		
	2.1 管理対象サーバーを停止する	約30秒（強制） / 約2分（正常）	
	2.2 管理サーバーを停止する	約8秒（強制） / 約2分（正常）	
3	DNS名をスイッチオーバーする	この部分はお客様ごとに異なります。たとえば、OCI DNSを使用している場合は30秒で済むことがあります。使用するDNSプロバイダによっては何時間もかかる可能性があります。この処理は、残りのステップと並行して実施できます。	
4	データベースをスイッチオーバーする	約3分	
5	セカンダリ・サイトでサーバーを起動する		
	5.1 管理サーバーを起動する	約3分	約2分
	5.2 管理対象サーバーを（並列で）起動する	約5分	約3分
... 停止時間の終了			

ステップ間で生じる必然的な遅延や、その他の追加検証にかかる時間は、スイッチオーバー手順の実行方法（手動処理、カスタム・スクリプトによる自動化、カスタム・オーケストレーション・ツールの使用、Oracle Full Stack DRの使用など）によって異なるため、上記に含まれていません。そのため、合計時間には、単に時間を合計するだけでなく、追加の時間を考慮する必要があります。DNSスイッチオーバーの時間も、お客様固有のものであるために除外されています。

通常、スイッチオーバー時間の合計は15~30分の範囲内と予測されます。ここで、スイッチオーバー操作中の停止時間を最小限に抑えるためのヒントを挙げます。

- スイッチオーバー関連の作業のうち、停止時間を必要としない作業はすべて、プライマリ・サーバーの停止前に実行します。たとえば、config_replica.shスクリプトに基づくWebLogic構成レプリケーションは停止時間を必要としないため、プライマリ・システムが稼働している間に実行できます。別の例として、停止したホストをスタンバイ・サイトで起動することが挙げられます。
- 可能な場合は、管理対象サーバーと管理サーバーをパラレルで停止します。
- アプリケーションおよびビジネス上で許容される場合は、WebLogicサーバーの停止に強制シャットダウンを使用します。
- WLSサーバーのシャットダウンにかかる最長時間は、"server lifecycle timeout"パラメータ（通常30秒に設定）と"graceful shutdown"パラメータ（通常120秒に設定）によって制限されます。最長シャットダウン時間を制限するために、これらのパラメータが設定されていることを確認します。
- DNSのフロントエンド更新作業はお客様ごとに異なります。（少なくともスイッチオーバー操作の間）適切なDNSエントリで低いTTL値を使用して、更新のための時間を削減します。スイッチオーバーが終了したら、TTLを元の値に戻すことができます。
- データベースのスイッチオーバーにData Guard Brokerコマンド（dgmgrl）を使用するほうが、OCIコンソールを使用するよりも高速です。RTOは2分にすることができます。ただし、OCIコンソールのUIでは、それぞれのDBシステムのロールは自動的にリフレッシュされません¹¹。OCIコンソールを使用してデータベースをスイッチオーバーすると、OCIコンソール内のロールが自動リフレッシュされますが、DBのスイッチオーバーにかかる時間が長くなります。
- OCI LBRがサーバーの起動を認識し、リクエストを送信し始めるまでにも時間が少しかかります。OCI LBRヘルス・チェックの頻度によりますが、通常は数秒です。チェックの間隔を短くすると、迅速にサーバーの起動を認識します。ただし、ヘルス・チェックの負荷が高い場合、間隔を短くしすぎるとバックエンドが過負荷になる可能性があるため、慎重に設定する必要があります。

予測されるRPO

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。データベースとWebLogic構成は異なる仕組みでレプリケートされるため、**ランタイム・データのRPOとWebLogic構成のRPO**は区別できます。

ランタイム・データのRPOとして実際に達成可能な値は、データベースのRPOに依存します。これは、ランタイム・データ（コンポジット・インスタンス、JMSメッセージ、TLOG、お客様データなど）がデータベースに保管されているからです。ランタイム・アーティファクトはファイル・システムにも保管されている場合があります（MFTによって使用されるファイルなど）。このため、**ランタイム・データのRPO**は以下によって変わります。

- プライマリとスタンバイ間で利用可能なネットワーク帯域幅とネットワーク信頼性。動的ルーティング・ゲートウェイおよびリモートVCNピアリングがプライマリとスタンバイの相互接続に使用されている場合、Oracle Cloud Infrastructureバックボーン・ネットワークが使用されています。OCIのバックボーン・ネットワークは、プライベートでルーティングされるリージョン間接続を提供しているため、帯域幅、待機時間、ジッターに関して、パブリック・インターネットよりも安定した性能を実現します（リージョン間のネットワーク待機時間について、詳しくはコンソールの「リージョン間のレイテンシ」ダッシュボードを参照してください）。OCIバックボーンの使用。DBシステムのData Guardが有効化されてOCIネットワーク・バックボーンが使用されている場合、RPOは最大5分間になります。最適な動作のために、ファスト・スタート・フェイルオーバー・オブザーバの手動構成が必要な場合があります。オブザーバの設定については、[Oracle DBシステムのドキュメント](#)を参照してください。
- Data Guardで使用される保護モードは、Maximum Availability（最大可用性）、Maximum Protection（最大保護）、またはMaximum Performance（最大パフォーマンス、デフォルト）のいずれかです。
最大可用性モードでは、特定の二重障害（スタンバイ・データベース障害の後にプライマリ・データベース障害が発生するなど）のケースを除いて、データ損失は生じません。

¹¹ dgmgrlコマンドによるスイッチオーバー後、OCIコンソールでDBシステムのロールが自動リフレッシュされない場合は、My Oracle Supportでサービス・リクエストをオープンしてください。

- **最大パフォーマンス・モード**で提供されるデータ保護レベルは、最大可用性モードよりも若干低くありますが、プライマリ・データベースのパフォーマンスへの影響が最小限に抑えられます。
- **最大保護モード**では、プライマリ・データベースで障害が起きてても、データ損失は発生しません。データ損失がないことを保証するため、REDOストリームを少なくとも1つの同期されたスタンバイ・データベースに書き込みできない場合、プライマリ・データベースはトランザクションの処理を続行せずに停止します。
あるシステムに最適なData Guard保護モードは、ビジネス要件によって異なります。企業は、状況にかかわらずデータを失うわけにはいかない場合があります。確率の低い多重障害発生時の潜在的なデータ損失よりもデータベース可用性の方が重要になるケースもあるでしょう。また、常に最大のデータベース・パフォーマンスが求められるため、何らかのコンポーネント障害が発生した場合のわずかなデータ損失が許容されるアプリケーションもあります。詳細については、Oracle Data Guardドキュメントの「[Oracle Data Guard Protection Modes](#)」を参照してください。

- c) さらに、データベースにないファイル・システムに保存されているランタイム・アーティファクト（MFTまたはFile/FTPアダプタによって使用または生成される、カスタムのファイル・ストレージ・サービスに保存されるファイルなど）がある場合、このデータのRPOは、セカンダリ・ロケーションに同期される頻度によって異なります。どのコンテンツを、いつ、どのような方法で同期するのかは、ビジネス要件によって決まります。たとえば、これらのランタイム・ファイルの持続性が非常に低い（短時間で作成/使用される）場合、ファイルの同期は必要なく、やり過ぎになる可能性があります。しかし、もっと静的なコンテンツであれば、DRイベント発生時にはセカンダリに保持されている必要があります。そのコピー頻度はシステムの想定されるRPOに準拠する必要があり、RPOは、コンテンツがレプリケートされる間に生成されるデータ量になります。
また、これらのランタイム・ファイルは、DBFSファイル・システム（/u01/soacs/dbfs/share/mftに保存されるMFTランタイム・ファイルなど）に置くことができます。この場合、基盤のData Guardレプリケーションによってファイルがスタンバイにレプリケートされるため、RPOはData Guard保護モードに依存します。

実際に達成可能なWebLogic構成のRPOは次の要因によって異なります。

- d) **WebLogic構成が変更される頻度。** WebLogic構成は、ランタイム・データほど動的に変更されません。システムの初期段階にもかかわらず、構成が継続的に変更されるのは一般的ではありません。構成が頻繁に変更されるほど、障害発生時に損失する可能性のある構成変更の量が多くなります。
- e) **WebLogic構成がスタンバイに同期される頻度。**
FSSおよびDBFSのレプリケーション方法では、WebLogic構成は、手動で、またはconfig_replica.shスクリプトにより自動的にレプリケートできます。1つのアプローチは、プライマリで実行されるすべての構成変更の後に構成をレプリケートすることです。これにより、セカンダリWebLogic構成には常にプライマリの最新状態が適用されますが、プライマリで変更が行われるたびにレプリケーション・プロセスを含める必要があります。もう1つの方法は、定期的（毎晩など）にレプリケーションをスケジュール設定することです。この場合、DRの計画外イベントが発生すると、最新のレプリケーション以降にプライマリで実行された構成変更は失われます。
リージョン間ブロック・ボリューム・レプリケーション・モデルでは、通常のRPOは30分よりも大幅に短いですが、ソース・ボリュームでのデータの変更率によって変化します。
- f) **WebLogic構成のレプリケーションに使用される手順の信頼性。** すべてのレプリケーション方法が信頼性に優れていますが、言うまでもなく、基盤インフラストラクチャに障害が発生した場合は（ステージング・フォルダを使用できない、接続障害など）、RPOに影響する可能性があります。このため、レプリケーション手順が正しく機能していることを確認し、セカンダリ・サイトの検証を定期的に実施することを推奨します。

ベスト・プラクティス

ディザスター・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。

- カスタム・リソースに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。カスタムの永続ストアを作成する場合は、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経由でスタンバイ・サイトにレプリケートされます。
- 新しいデータソースを作成する場合は、URL接続文字列でTNSエイリアスを使用する。適切なTNS文字列がプライマリ中間層システムとスタンバイ中間層システムの両方でtnsnames.oraに必ず含まれるようにしてください。tnsnames.oraファイルは各サイトに固有のもので、レプリケートされません。
- プライマリ・サイトとスタンバイ・サイトの間で同じパッチ・レベルを維持する。どの層のソフトウェアも、自動的にはセカンダリ・サイト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、Data Guardトポロジにパッチを適用する方法については、当該パッチのドキュメントを参照してください。
- 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、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。例としては、プライマリ・ロードバランサの変更や、オペレーティング・システムの変更などがあります。
定期的にスイッチオーバーを実行してセカンダリ・サイトの健全性を検証する。または、完全なスイッチオーバーを実行せずに、検証用にセカンダリ・サイトを開くことができます。
- アプリケーションのデプロイ操作では、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が配置されるようにすることを推奨します。このようにすると、構成レプリケーション・スクリプトまたはBVコピー（DBFS、FSS、BVのいずれが使用されるかによって変わります）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。
- デフォルトでは、SOAホストが再起動すると、WLS管理サーバーと管理対象サーバーは“自動”で起動します。ただし、これはスタンバイ・サイトでは望ましくありません。スタンバイ・サイトでは、<https://docs.oracle.com/en/cloud/paas/soa-cloud/soa-marketplace/soamp-disable-server-restart-instance-reboot.html>に記載されている機能を使用してこの自動再起動を無効化することが推奨されます。（すべてのスタンバイSOAホストで）"start_server_on_reboot"をfalseに設定すると、マシンの起動時にノード・マネージャのみが起動します。
- そのプロパティ（soampRebootEnv.sh）を設定するためのファイルは、構成をレプリケートする際に上書きされないため、プライマリ・システム（trueの状態を想定）とスタンバイ・システム（falseに設定可能）とで異なる値を設定できます。その後、スイッチオーバーを実行し、セカンダリ・サイトをプライマリとして長期間使用する予定がある場合は、それに応じて各サイトの値を変更できます。
- プライマリとスタンバイの両方において、SOAMPホストによって使用されるブロック・ボリュームについて、ブロック・ボリュームのバックアップを定期的に実行するか、またはブロック・ボリュームの自動バックアップを構成する。詳しくは、SOA Marketplaceドキュメントの[Back Up a Block Volume](#)を参照してください。

結論

SOA Suite on OCI Marketplace構成でのディザスター・リカバリは、Oracle Data Guardによって同期される本番データベースとスタンバイ・データベース、ローカル・データベースを参照する2つの中間層構成、および最小限必要なファイル・レプリケーションを管理するためのソリューションで構成されます。Oracle Cloudでこのディザスター・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、サード・パーティ・レプリケーション・ソフトウェア、さらにリモート・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、同時に業界屈指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をディザスター・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、ディザスター・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、異なる構成同期アプローチを使用して動作を最適にすることができます。

付録A – 手動構成されたData GuardでのDBシステム・バックアップ[†]

DBシステムのバックアップは、どのようなOracleデータベース環境でも重要な側面です。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。

Data Guardが手動で構成されている場合（「[オプション2 手動でのData Guardの構成](#)」）、Data Guard環境の構成を最適化するため、バックアップも手動で構成する必要があります。データベースのどちらか（プライマリまたはスタンバイ）でバックアップを実行し、他方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必要です。

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。以前に自動バックアップを有効化したことがない場合は、「[RMANを使用したオブジェクト・ストレージへのデータベースのバックアップ](#)」の手順に従い、プライマリDBにバックアップ・モジュールをインストールして構成します。
- リンクの推奨事項に従って、RMANを設定します。また、Data Guardに推奨されているアーカイブ・ログ削除ポリシーを含めます。

```
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。全体バックアップを実行する例は次のとおりです。

```
# RMANを実行します
export ORACLE_HOME=/u01/app/oracle/product/18.0.0.0/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
```

スタンバイでアーカイブ・ログの増加を管理するには、以下の手順に従います。

- 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、次のコマンドを実行して適切なアーカイブ・ログ削除ポリシーを構成します。

```
RMAN> CONFIGURE ARCHIVELOG DELETION POLICY TO APPLIED ON ALL STANDBY;
```

- 適切なアーカイブ・ログ削除ポリシーを設定すると、ファスト・リカバリ領域（FRA）でのアーカイブ・ログの増加を十分管理できますが、クリーンアップ・スクリプトを作成して古いアーカイブ・ログを削除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を使用します # 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・ロールの場合にアーカイブ・ログが不要に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"
#####
```

CloudコンソールUIを使ってOracle Data Guardが構成されている場合、CloudコンソールUIを使ってプライマリまたはスタンバイ・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。これが推奨される方法です。このようなケースでは、デフォルトのRMAN構成で、Data Guardシナリオに推奨されるアーカイブ・ログ削除ポリシーが使用されていると考えられます。ただし、前述の説明に従って、セカンダリ・データベースでもアーカイブ・ログの増加を管理できます。

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。こういったまれなシナリオでは、スタンバイ・データベースの再作成も必要です。スタンバイ・データベースを再作成するには、次のようにします。

手動Data Guardの場合、プライマリ・データベースのリストア後に、「[オプション2\) 手動でのData Guardの構成](#)」の手順で提供されたスクリプトを再実行してスタンバイ・データベースを再作成し、*Data Guard*を再構成します。

ただし、**自動Data Guard**では、OC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システムでDRを再構築するために、SOAMP中間層で一部のプロパティを更新する必要があります。「[スタンバイDBシステム再作成後のSOAMP DRの再構築](#)」を参照してください。

付録B - 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、SOA Marketplace DRの具体的なネットワーク要件を示します。

操作	SSH	SQLNET (1521)	HTTPS
DRセットアップ (DRSを使用)	<p>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）。</p> <p>ファイル・レプリケーションにFSS/rsyncの方法を使用する場合は、プライマリ・サイトのWLS管理サーバー・ホストからセカンダリ・サイトのWLS管理サーバー・ホスト（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プライベートIP、インターネットを介して通信する場合はパブリックIP¹²）への接続性も必要。</p>	<p>セカンダリ・サイトのすべての中間層ホストからプライマリ・サイトのDBプライベートIP（Oracle RACの場合はスキャンIP）（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）。</p> <p>または</p> <p>セカンダリ・サイトのすべての中間層ホストからプライマリ・サイトの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）。¹¹</p>	<p>DRSが実行されているホストからプライマリ・サイトのフロントエンドIP。DRSが実行されているホストからセカンダリ・サイトのフロントエンドIP。DRSが実行されているホストからインターネット。</p>
config_replica.shを介したWLSドメイン構成レプリケーション (FSS/rsyncの方法)	各サイトのWLS管理サーバー・ホストから他のサイトのWLS管理サーバー・ホスト（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プライベートIP、インターネットを介して通信する場合はパブリックIP）。 ¹¹		
config_replica.shを介したWLSドメイン構成レプリケーション (DBFSベースの方法)		<p>各サイトのWLS管理サーバー・ホストからリモートDBプライベートIP（Oracle RACの場合はスキャンIP）（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）。</p> <p>または</p> <p>各サイトのWLS管理サーバー・ホストからリモートDB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）。¹¹</p>	
通常実行時		プライマリ・サイトとセカンダリ・サイトのデータベース間（これはData Guardの要件です）。	

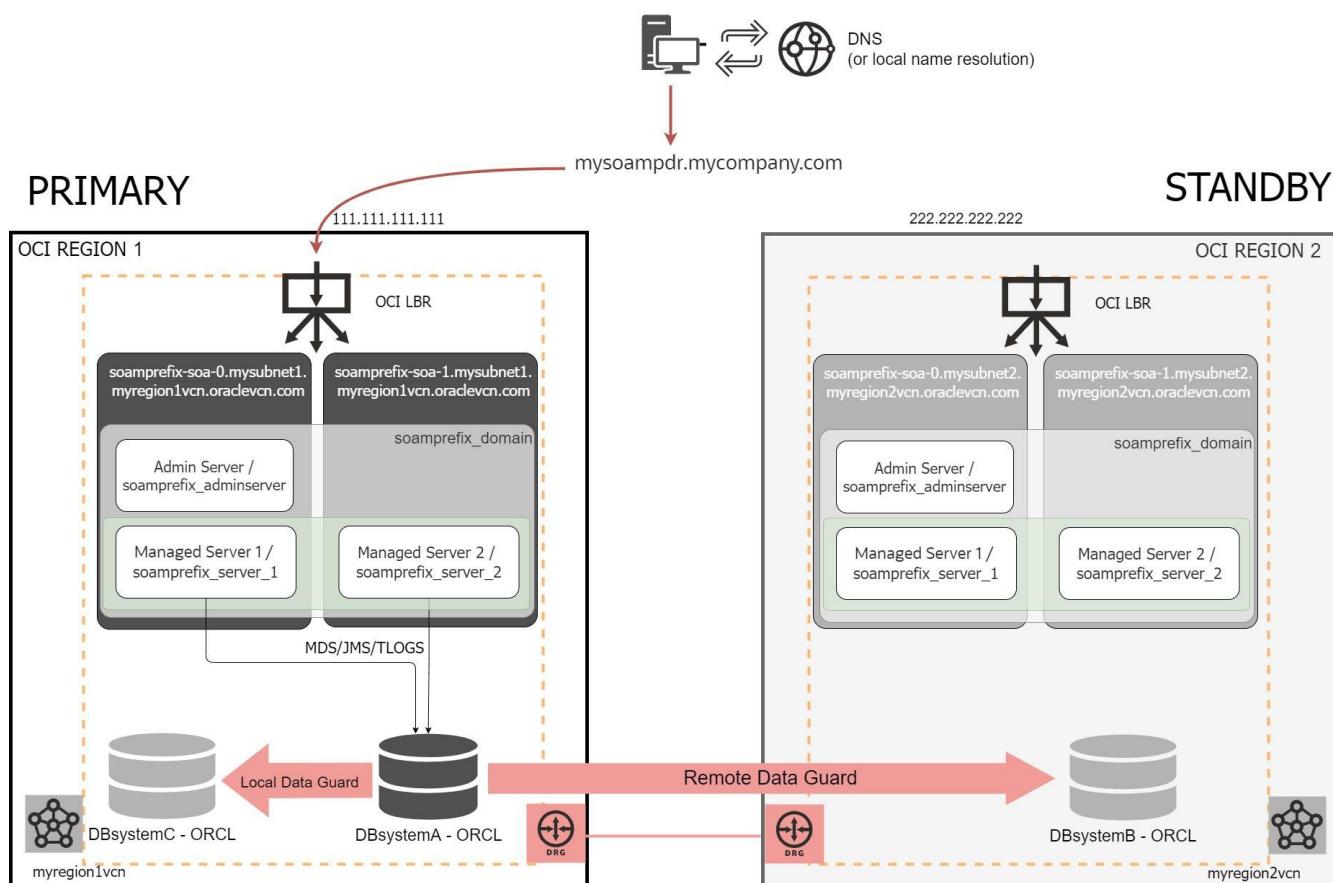
¹² OCIでは、異なるリージョンに配置されたVCNネットワーク間のプライベート・トライフィックに動的ルーティング・ゲートウェイを使用できるため、インターネットを介したサイト間通信は最新バージョンでは推奨されていません。

付録C - プライマリでの追加スタンバイ・データベースの使用

OCIコンソールでサポートされるのは、**1つのプライマリ・データベースごとに1つのスタンバイ・データベースがあるケースのみ**です。本書でも、セカンダリ・リージョンにあるスタンバイは1つだけであることを前提としています。

プライマリ・サイトに手動でさらにスタンバイ・データベースを追加しているシナリオでは、以下を考慮することでリージョン間DRのセットアップを実行できます。

1. リージョン間ブロック・ボリューム・レプリケーションの方法を使用する場合は、通常どおりにDRのセットアップを実行できます。確認する必要があるのは、各サイトで適切な接続文字列を更新することだけです。
2. DBFSまたはFSS/rsyncを使用した場合は、**DRセットアップ・スクリプトの実行前に、Data Guard構成からローカル・スタンバイを削除する必要があります**。DRSの実行が終わったら、後でもう一度追加できます。以下に詳細を示します。



ローカル・スタンバイの構成前の追加手順（DBFSの方法とFSS/rsyncの方法）

リージョン間DRSユーティリティをセットアップするためのスクリプトを実行する前に、以下を実行します。

1. Data Guard構成でローカル・スタンバイを無効化し、削除します。DRSの実行前に、リージョン間のスタンバイのみがData Guard構成に存在する状態になっている必要があります。以下に例を示します。

```

DGMGRL> show configuration
Configuration - ORCL_DG_CONF
Protection
Mode:MaxPerformance
Members:
  ORCL_london1 - Primary database
  ORCL_frankfurt - Physical standby
  database ORCL_london2 - Physical standby
  database
Fast-Start Failover:Disabled
Configuration Status:
SUCCESS (status updated 130 seconds ago)

DGMGRL> disable database
ORCL_london2 Disabled.

DGMGRL> show configuration
Configuration - ORCL_DG_CONF
Protection
Mode:MaxPerformance
Members:
  ORCL_london1 - Primary database
  ORCL_frankfurt - Physical standby
  database
  ORCL_london2 - Physical standby database
(disabled) ORA-16749:The member was disabled
manually.
Fast-Start Failover:Disabled
Configuration Status:
SUCCESS (status updated 1 seconds
ago) DGMGRL> remove database
ORCL_london2
Removed database "orcl_london2" from the configuration

DGMGRL> show configuration
Configuration - ORCL_DG_CONF
Protection
Mode:MaxPerformance
Members:
  ORCL_london1 - Primary database

```

2. **(TNSエイリアスのアプローチを使用している場合は不要)** この手順は、TNSエイリアスのアプローチの前に構成されたSOAMP DR環境でのみ必要とされます。プライマリ・システムで、デュアル文字列接続を使用していないことを確認します。DRセットアップの完了後はデュアル文字列接続を設定できますが、DRをセットアップするまでは、**データソースとjps-config.xml**ファイル内の構文は推奨される形式に従う必要があります。
- データベースがシングル・インスタンスの場合、推奨されるDB接続文字列は次のようにになります。
`jdbc:oracle:thin:@//<db-scan-address>:<port>/< pdb_service_name>`
 データベースがOracle RACの場合、データソースはGridLinkデータソースでなければなりません。また、推奨されるDB接続文字列は次のようにになります。
`jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=<db-scan-address>)(PORT=<port>)))(CONNECT_DATA=(SERVICE_NAME=< pdb_service_name>)))`

ローカル・スタンバイのリージョン間の追加DR構成（DBFSの方法とFSS/rsyncの方法）

構成前の手順が完了したら、本書の説明に従ってDRセットアップを実行します。DRのセットアップ中、ローカル・スタンバイは処理に関係しません。

ローカル・スタンバイの構成後の追加手順（DBFSの方法とFSS/rsyncの方法）

DRセットアップの完了後、もう一度ローカル・スタンバイをトポロジに追加して、必要なシステムの調整を実行できます。次のステップを実行します。

1. Data Guard構成にローカル・スタンバイ・データベースを再度追加して有効化します。以下に例を示します。

```
DGMGRL> add database 'ORCL_london2' as connect identifier is 'ORCL_london2' maintained as
physical; Database "ORCL_london2" added

DGMGRL> show configuration
Configuration - ORCL_DG_CONF
Protection
Mode:MaxPerformance
Members:
  ORCL_london1 - Primary database
  ORCL_frankfurt - Physical standby
  database
  ORCL_london2 - Physical standby database
    (disabled) ORA-16905:The member was not
      enabled yet.
Fast-Start Failover:Disabled
Configuration Status:
SUCCESS (status updated 29 seconds ago)

DGMGRL> enable database
ORCL_london2; Enabled.

DGMGRL> show configuration
Configuration - ORCL_DG_CONF
Protection
Mode:MaxPerformance
Members:
  ORCL_london1 - Primary database
  ORCL_frankfurt - Physical standby
  database ORCL_london2 - Physical standby
  database
  Warning:ORA-16857: member disconnected from redo source for longer than
```

新しく追加されたメンバーが保留中のREDOを受け取って適用するまでには、少し時間がかかります。以下に例を示します。

```
DGMGRL> validate database ORCL_london2
Database Role:Physical standby database
Primary Database:ORCL_london1
Ready for Switchover:Yes
Ready for Failover:Yes (Primary
Running) Managed by Clusterware:
  ORCL_london1:YES
  ORCL_london2:YES

Standby Apply-Related Information:
  Apply State:Running
  Apply Lag:19 hours 45 minutes 12 seconds (computed 1 second
ago) Apply Delay:0 minutes

DGMGRL> /
Database Role:Physical standby
database Primary
Database:ORCL_london1
Ready for Switchover:Yes
Ready for Failover:Yes (Primary
Running) Managed by Clusterware:
  ORCL_london1:YES
  ORCL_london2:YES
```

```

DGMGRL> show configuration
Configuration - ORCL_DG_CONF
Protection
Mode:MaxPerformance
Members:
  ORCL_london1 - Primary database
  ORCL_frankfurt - Physical standby
  database ORCL_london2 - Physical standby
  database
Fast-Start Failover:Disabled
Configuration Status:
SUCCESS (status updated 55 seconds ago)

```

必要なアーカイブがすでに使用できなくなっている場合は、手動による調整が必要になります。保留中のアーカイブを見つけるか、“restore from service”を使用してプライマリからローカル・スタンバイをリフレッシュする必要があります。

- PDBのサービス名は、プライマリとローカル・スタンバイで一致させる必要があります。こうすることで、プライマリWebLogicシステムでデュアル文字列を使用できます。まだ作成していない場合は、次のとおりに新しいCRSサービスをプライマリ・システムとローカル・スタンバイ・システムで作成します。
プライマリ・データベース・システムで、プライマリ・ロールとスナップショット・スタンバイ・ロールを持つPDBのサーバーを作成します。次に例を示します。

```

srvctl add service -db $ORACLE_UNQNAME -service mypbservice.example.com -preferred ORCL1,ORCL2 -pdb
PDB1 -role "PRIMARY,SNAPSHOT_STANDBY"
srvctl modify service -db $ORACLE_UNQNAME -service mypbservice.example.com -rlbgoal SERVICE_TIME -clbgoal SHORT
srvctl config service -db $ORACLE_UNQNAME -service mypbservice.example.com

```

ローカル・スタンバイ・データベース・システムで、上記を繰り返します。以下に例を示します。

```

srvctl add service -db $ORACLE_UNQNAME -service mypbservice.example.com -preferred ORCL1,ORCL2 -pdb
PDB1 -role "PRIMARY,SNAPSHOT_STANDBY"
srvctl modify service -db $ORACLE_UNQNAME -service mypbservice.example.com -rlbgoal SERVICE_TIME -clbgoal SHORT
srvctl config service -db $ORACLE_UNQNAME -service mypbservice.example.com

```

その時点でプライマリ・ロールを持つシステムで、新しいサービスを開始します（次回、このサービスはDGプローラによって自動的に停止/開始されます）。以下に例を示します。

```

srvctl start service -db $ORACLE_UNQNAME -service mypbservice.example.com

```

- ローカル・スタンバイ・データベースを使用するサイトの中間層ホストでは、TNS管理フォルダのtnsnames.oraのTNSエイリアスはデュアルである必要があります。これには、プライマリおよびローカル・スタンバイのアドレスが含まれます。次に、tnsnames.oraのTNSエイリアス・エントリの例を示します。

```

MYALIAS =
(DESCRIPTION=
(CONNECT_TIMEOUT=15)(RETRY_COUNT=5)(RETRY_DELAY=5)
(ADDRESS_LIST=(LOAD_BALANCE=on)(ADDRESS=(PROTOCOL=TCP)(HOST= drdba-
scan.mysubnet.region1vcn.oraclevcn.com)(PORT=1521)))
(ADDRESS_LIST=(LOAD_BALANCE=on)(ADDRESS=(PROTOCOL=TCP)(HOST= drdbc-
scan.mysubnet.region1vcn.oraclevcn.com)(PORT=1521)))
(CONNECT_DATA=(SERVICE_NAME=mypbservice.example.com)))

```

4. (TNSエイリアスのアプローチを使用している場合は不要)

この手順は、TNSエイリアスのアプローチの前に構成されたSOAMP DR環境でのみ必要とされます。プライマリ・サイトのWebLogic構成でデュアル文字列による接続を設定します。この設定は、データソースとjps-config.xmlファイルの両方で行います。データソースとjps-config.xmlファイルで文字列を一致させる必要があります。次に、ローカル・スタンバイを含むデュアルのデータソース接続文字列の例を示します。

```
jdbc:oracle:thin:@(DESCRIPTION=
  (CONNECT_TIMEOUT=15)(RETRY_COUNT=5)(RETRY_DELAY=5)
  (ADDRESS_LIST=(LOAD_BALANCE=on)(ADDRESS=(PROTOCOL=TCP)(HOST= drdba-
  scan.mysubnet.region1vcn.oraclevcn.com)(PORT=1521)))
  (ADDRESS_LIST=(LOAD_BALANCE=on)(ADDRESS=(PROTOCOL=TCP)(HOST= drdbc-
  scan.mysubnet.region1vcn.oraclevcn.com)(PORT=1521)))
  (CONNECT_DATA=(SERVICE_NAME=mypdbservice.example.com)))
```

注 :この例では、分かりやすくするために改行を使用しています。実際の接続文字列には、**改行や空白文字を追加しないでください。**データソースとjps-config.xmlファイルには、上記を1行で追加します。

セカンダリ・サイトでは、デュアルの接続文字列は使用しません。設定を変更しないでください。セカンダリ中間層はセカンダリのローカル・データベースのみに接続します。config_replica.shスクリプトをスタンバイ・サイトで実行すると、コピーされた構成内で必要な接続文字列の置換が自動的に行われます。

- 必ず最新バージョンのconfig_replica.shスクリプトを使用してください。これは適宜更新してください。ローカル・スタンバイ・データベースのCDB接続文字列を指定する必要があります。

警告 :最新バージョンのconfig_replica.shスクリプトは、TNSエイリアスのアプローチにのみ有効です。

- (TNSエイリアスのアプローチを使用している場合は不要)

この手順は、TNSエイリアスのアプローチの前に構成されたSOAMP DR環境でのみ必要とされます。プライマリ・サイトの中間層ホストで、**/u01/data/domains/local_CDB_jdbcurl.nodelete**ファイルを更新します。ここに、ローカル・スタンバイのCDBサービスへのJDBC URL接続の行を追加します。

変更前のファイルには、ローカル・プライマリ・データベースを参照する1行のみが含まれています。以下に例を示します。

```
drdba-scan.mysubnet.region1vcn.oraclevcn.com:1521/ORCL_london1.mysubnet.region1vcn.oraclevcn.com
```

ローカル・スタンバイの追加後は、ローカル・プライマリ・データベースとローカル・スタンバイ・データベースの2行が含まれます。以下に例を示します。

```
drdba-scan.mysubnet.region1vcn.oraclevcn.com:1521/ORCL_london1.mysubnet.region1vcn.oraclevcn.com
drdbc-scan.mysubnet.region1vcn.oraclevcn.com:1521/ORCL_london2.mysubnet.region1vcn.oraclevcn.com
```

重要 :このファイルにはPDBサービス名ではなくCDBサービス名を指定する必要があります。

このファイルは、DR設定中に、DRセットアップ・スクリプトによって作成されています。ライフサイクル中、config_replica.shスクリプトはこの情報を使用してローカルCDBに接続し、現在のサイト・ロールを取得します。ローカル・スタンバイ・データベースを使用する場合、ファイルにはスタンバイのローカルCDB接続文字列も含める必要があります。それにより、ローカルでスイッチオーバーが発生した場合も、config_replica.shで正しいサイト・ロールを取得できます。

- (TNSエイリアスのアプローチを使用している場合は不要)

この手順は、TNSエイリアスのアプローチの前に構成されたSOAMP DR環境でのみ必要とされます。**レプリケーションにDBFSの方法が使用されている場合は、中間層ホストのtnsnames.oraに、追加のスタンバイへのエイリアスを追加します。**このエイリアスには**ローカル・スタンバイ・データベースの一意の名前**を使用し、参照先にはCDBのデフォルト・サービス名を指定する必要があります。追加は、プライマリとスタンバイの両方の中間層ホストで行います。これは、そのサイトがスタンバイ・ロールにあるとき、config_replica.shによって使用されます。以下に例を示します。

- それぞれの中間層ホストで、\$DOMAIN_HOME/dbfs/tnsnames.oraファイルを編集します。
- プライマリ・サイト（追加のローカル・スタンバイがあるサイト）の中間層ホストで、ローカル・スタンバイCDBへのエイリアスを追加します。次に例を示します。

```
..  
ORCL_london2 =  
(DESCRIPTION =  
(SDU=65536)(RECV_BUF_SIZE=10485760)(SEND_BUF_SIZE=10485760)  
(ADDRESS = (PROTOCOL = TCP)(HOST = drdbc-scan.mysubnet.region1vcn.oraclevcn.com)(PORT =1521))  
(CONNECT_DATA=(SERVER = DEDICATED)(SERVICE_NAME = ORCL_london2.mysubnet.region1vcn.oraclevcn.com)))  
...
```

上の例ではそれぞれ次のとおりになります。

“ORCL_london2”は、ローカル・スタンバイの一意のDB名です。

“drdbc-scan.mysubnet.region1vcn.oraclevcn.com”は、ローカル・スタンバイのSCANアドレスです。

“ORCL_london2.mysubnet.region1vcn.oraclevcn.com”は、ローカル・スタンバイCDBのデフォルト・サービス名です。

- セカンダリ・サイトではおそらくSCAN名が解決されないため、SCAN名ではなくIPを使用します。以下に例を示します。

```
...  
ORCL_london2=  
(DESCRIPTION=  
(SDU=65535)(SEND_BUF_SIZE=10485760)(RECV_BUF_SIZE=10485760)  
(ADDRESS_LIST=  
(ADDRESS=(PROTOCOL=TCP)(HOST=10.0.2.42)(PORT=1521))  
(ADDRESS=(PROTOCOL=TCP)(HOST=10.0.2.43)(PORT=1521))  
(ADDRESS=(PROTOCOL=TCP)(HOST=10.0.2.44)(PORT=1521)))  
(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME= ORCL_london2.  
mysubnet.region1vcn.oraclevcn.com)))  
..
```

上の例ではそれぞれ次のとおりになります。

“ORCL_london2”は、ローカル・スタンバイの一意のDB名です。“10.0.2.42”、“10.0.2.43”、“10.0.2.44”は、ローカル・スタンバイのSCAN IPです。

“ORCL_london2.mysubnet.region1vcn.oraclevcn.com”は、ローカル・スタンバイCDBのデフォルト・サービス名です。

8. DBFSマウントを使用している場合（構成レプリケーションまたはその他の目的で）、\$DOMAIN_HOME/dbfs/tnsnames.oraファイル内でPDBを参照するエントリには、デュアル接続文字列を使用する必要があります。ローカル・スイッチオーバーの発生時、DBFSがローカル
スタンバイに接続できなければなりません。
 - DBFSマウントによって使用されるtnsエイリアスを特定します。通常は、“ORCL”またはPDB名です。
 - \$DOMAIN_HOME/dbfs/tnsnames.oraファイルで目的のエイリアスを見つけます。
 - このtnsエントリを、ローカル・スタンバイPDBを含むデュアル文字列形式に変更します。次の例のようになります。

```
PDB1 = (DESCRIPTION =  
(CONNECT_TIMEOUT= 10)(RETRY_COUNT=10) (RETRY_DELAY=10)  
(ADDRESS_LIST=(LOAD_BALANCE=on)(ADDRESS=(PROTOCOL=TCP)(HOST=drdba-  
scan.mysubnet.region1vcn.oraclevcn.com)(PORT=1521)))  
(ADDRESS_LIST=(LOAD_BALANCE=on)(ADDRESS=(PROTOCOL=TCP)(HOST=drdbrc-  
scan.mysubnet.region1vcn.oraclevcn.com)(PORT=1521)))  
(CONNECT_DATA =  
(SERVER = DEDICATED)  
(SERVICE_NAME = mypdbservice.example.com)))
```

ローカル・スタンバイのローカル・データベースの追加スイッチオーバー（DBFSの方法とFSS/rsyncの方法）

このシナリオでは、使用できる追加のライフサイクル運用があります。プライマリ・データベースからローカル・スタンバイ・データベースにスイッチオーバーできます。この操作では、中間層はスイッチオーバーされません。ローカル・データベースのスイッチオーバーが完了したら、WebLogicサーバーは自動的に新しいローカル・プライマリ・データベースに再接続します。このためには、データソースとjps構成ファイルに、デュアルのデータベース接続文字列を設定して使用します。

このシナリオでは、構成レプリケーション手順を検証することを推奨します。以下の検証手順を使用します。

1. プライマリ・データベースからローカル・スタンバイ・データベースにスイッチオーバーします。スイッチオーバーの進行中はデータベースを使用できないため、この操作によりシステムで停止が発生します。アプリケーションへの影響を最小限に抑えるため、このステップはメンテナンス時間内に実行します。
2. プライマリWebLogic構成で、何らかの構成変更を行います。たとえば、いざれかのデータソースで接続プール・サイズを変更します。
3. WebLogic構成をセカンダリ・サイトにレプリケートします。通常どおり、はじめにプライマリWebLogic管理ホストでconfig_replica.shを実行してから、セカンダリWebLogic管理ホストでconfig_replica.shを実行します。
4. セカンダリ・サイトにWebLogic構成が正しくレプリケートされたことを確認します。セカンダリ・ドメインのファイルに変更が反映されていることを確認します。
5. さらに徹底的に検証する場合、スタンバイ・データベース（セカンダリ・リージョンにあるリモート・スタンバイ）をスナップショット・スタンバイに変換して、セカンダリWebLogic管理サーバーを起動します。
6. WebLogic構成コンソールで変更が反映されているかどうかを確認します。セカンダリWebLogic管理サーバーを停止し、スタンバイ・データベースをもう一度スナップショット・スタンバイからフィジカル・スタンバイに戻します。
7. プライマリ・データベースを元のプライマリにスイッチオーバーし、システムをスイッチバックして元の状態に戻します。

CONNECT WITH US

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

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



Copyright © 2024, Oracle and/or its affiliates. All rights reserved.本文書は情報提供のみを目的として提供されており、ここに記載されている内容は予告なく変更されることがあります。本文書は、その内容に誤りがないことを保証するものではなく、また、口頭による明示的保証や法律による默示的保証を含め、商品性ないし特定目的適合性に関する默示的保証および条件などのいかなる保証および条件も提供するものではありません。オラクルは本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクルの書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

OracleおよびJavaはOracleおよびその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標です。

IntelおよびIntel XeonはIntel Corporationの商標または登録商標です。すべてのSPARC商標はライセンスに基づいて使用されるSPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMDロゴおよびAMD Opteronロゴは、Advanced Micro Devicesの商標または登録商標です。UNIXは、The Open Groupの登録商標です。0120

Oracle SOA Suite on Oracle Cloud Infrastructure Marketplaceのデイザスター・リカバリ（2024年7月）

